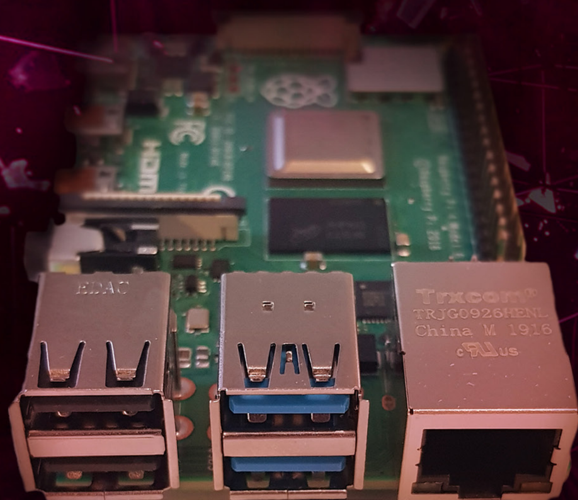
RaspberryTips

MASTER RASPBERRY PI

In 30 Days

FROM BEGINNER TO EXPERT IN RASPBERRY PI.
LEARN USEFUL LINUX SKILLS AND PRACTICE
MULTIPLES PROJECT WITH STEP-BY-STEP GUIDES



PATRICK FROMAGET

Table of contents

Foreword

As tradition wants, I will start with an introduction for people needing clarifications about me and this book.

What is the book's purpose?

As mentioned in the title, the goal of this book is to improve your Raspberry Pi skills (and probably your Linux skills as well).
No matter what your current level is.
By the end of the book, if you work seriously on each part, you'll know more than the other 90% Raspberry Pi owners.

This book's purpose is to turn you into a Raspberry Pi expert in 30 days
This will highly depend on your available time, but it is completely attainable with decent work every day.

Every chapter is different and will require more or less time to achieve it, but one to two hours a day is a good start.

Don't rush the book and try to respect the timeline, even if you have more time available.
And of course, if you don't have that much time, you can progress at your own pace and finish it over a longer period of time.

The main goal is to understand everything at the end, the schedule is not that important.

What is in the book?

In this book, you'll find a step by step way to learn Raspberry Pi.
I worked on a natural way to start from zero and increase your skills.

As you will see, this book has lots of pictures, schemas and several illustrations to help you visualize the new things I will teach you (I promise this introduction is the last boring part of the book ^^)

I don't like text-only books to learn new skills, especially in computing, therefore I hope you will appreciate this format.

Hardware notice

I use my Raspberry Pi 4 to create all the tutorials on this book, but any Raspberry Pi model would be fine. Recent models are more powerful so it's better to use them if you can, but it's not mandatory.

Who am I?

Yes, it's always a good idea to check if your teacher knows anything about the topic you want to learn :)

Let's continue with the traditional introduction.
My name is Patrick, I live in France and I went to a developer school a long time ago :)
I have worked in different companies for 20 years now, first as a web developer and now as a system administrator, mainly dealing with Linux systems.

In 2018, I started a Raspberry Pi hobby to apply my Linux knowledge in home projects.
In the same year, I started the RaspberryTips.com website to share my experiences.
Today, there are over 300 tutorials on the website and 100 videos on the YouTube channel. Every week, I publish at least one new tutorial.

I try to help people learn new skills on Linux and Raspberry Pi through the use of how-to guides and I also try to give them new ideas with inspirational posts.

Why did I write this book?

If I already teach Raspberry Pi on my website and it's free, so why do you need this book?

On my website, I help people with very specific things using short articles.
Each guide is about a tiny part of what you can do with a Raspberry Pi, and most of the time I have to guess the readers level.

So it can be either too hard to follow for newbies or too detailed for experienced people.

With this book, you will start at the very beginning, with a gradual difficulty increase.
No matter what your current level is. You can move quickly through the first parts if you already have a Raspberry Pi working, but at the end everyone will have the same knowledge.
That’s the best way to learn, in a step by step process and not by merging all the tiny skills together in a random order.

Day 1:
Hardware selection

Introduction

Your first step in the Raspberry Pi world will be to select a device that fits your specific needs.
It's not that complicated, but it can become overwhelming when you start browsing your favorite retailer website.

Between different versions of Raspberry Pi, you will also find different RAM quantities; along with dozens of kits and accessories to further confuse the decision making process!
In this chapter, I will clarify everything to make your choice easier.

Even if you already have a Raspberry Pi, I recommend reading this informational chapter, as you will probably still learn a few things.

Models list

At the time of writing, since 2012 the Raspberry Pi Foundation has released 14 models through 3 series.
Don't worry, you won't need to know everything about these 14 models to make your choice.
In each series or family, you will find a few versions, and most of the time the latest version is the best and the only one you'll find on the market.

For your information I will introduce each family, and then give you a short summary of each version.
Then, I will summarize everything to clarify your options, and when you need to choose each series and version.

Raspberry Pi series A

Presentation

The Raspberry Pi "A" family offers the lowest-cost models.
You can absolutely use it if you have a very low budget, or if you need to really limit your power consumption.

You just need to know that you will only get the minimum, vital components.
For example, Raspberry Pi A models don't have an Ethernet port; USB is limited to one port; and the amount of RAM is very low compared to the more powerful ones available.
But, other things are very similar to what you will get in the B series.

Models A list

Version	Raspberry Pi 1 Model A	Raspberry Pi 1 Model A+	Raspberry Pi 3 Model A+
Release Date	February 2012	November 2014	November 2018
CPU	Single Core 700Mhz (ARMv6)	Single Core 700Mhz (ARMv6)	Quad Core 1.4GHz (ARMv8)
RAM (MB)	256	512	512
USB ports	1	1	1
Network	No	No	Wi-Fi / Bluetooth
Dimensions	85 x 56 x 10 (mm)	65 x 56 x 10 (mm)	65 x 56 x 10 (mm)
Idling power consumption	300mA	200mA	300mA

Raspberry Pi series B

Presentation

The model B series offers the best performance possible for a slightly higher price.
You also get all of the components included, like Ethernet port and more USB ports.

The table below shows you exactly what you will get with each Raspberry Pi version.

Models B list

Version	Raspberry Pi Model B	Raspberry Pi Model B+	Raspberry Pi 2 Model B
Release Date	March 2012	July 2014	February 2015
CPU	700Mhz ARMv6	700Mhz ARMv6	900Mhz Quad Core ARMv7
RAM (MB)	512	512	1024
USB ports	2	4	4
Network	Ethernet	Ethernet	Ethernet
Dimensions	85 x 56 x 17 (mm)	85 x 56 x 17 (mm)	85 x 56 x 17 (mm)
Idling power consumption	700mA	200mA	200mA

Version	Raspberry Pi 3 Model B	Raspberry Pi 3 Model B+	Raspberry Pi 4 Model B
Release Date	February 2016	March 2018	June 2019
CPU	1.2Ghz Quad Core ARMv8	1.4Ghz Quad Core ARMv8	1.5Ghz Quad Core ARMv8
RAM (MB)	1024	1024	1024, 2048, 4096
USB ports	4	4	4
Network	Ethernet, Wifi, Bluetooth	Ethernet, Wifi, Bluetooth	Ethernet, Wifi, Bluetooth
Dimensions	85 x 56 x 17 (mm)	85 x 56 x 17 (mm)	85 x 56 x 17 (mm)
Idling power consumption	300mA	450mA	600mA

Version	Raspberry Pi 5
Release Date	October 2023
CPU	2.4Ghz Quad Core
RAM (MB)	4096, 8192
USB ports	4
Network	Ethernet, Wifi, Bluetooth
Dimensions	85 x 56 x 17 (mm)
Idling power consumption	540mA

Raspberry Pi 400

Presentation

The Raspberry Pi 400 is a brand new model. I don't know if we can consider it as "a series" for the moment, but it's very different from the other ones.

The main difference is that it's not a motherboard only, as it's embedded in a desktop keyboard. It's closer to a desktop computer, a bit like a classic Atari, Commodore or Amiga if you want.

Model 400 specifications

Version	Raspberry Pi 400
Release Date	November 2020
CPU	1.8Ghz Quad Core ARMv8
RAM (MB)	4096
USB ports	3
Network	Ethernet, Wifi, Bluetooth
Dimensions	286 x 113 x 23 (mm)
Idling power consumption	500mA

Raspberry Pi series Zero

Presentation

The last Raspberry Pi family is the Zero family.
The Raspberry Pi Zero is the smallest and cheapest Raspberry Pi available.

At about half the size of other models, it offers a new set of possibilities.
It's the perfect option if you need something discreet or need to be powered constantly.
Projects ideas that use the Zero include things like: a small security camera, a DIY Game Boy or even a weather station.

Models Zero list

Version	Raspberry Pi Zero v1.2	Raspberry Pi Zero v1.3	Raspberry Pi Zero W / WH*
Release Date	November 2015	May 2016	February 2017
CPU	1Ghz Single Core (ARMv6)	1Ghz Single Core (ARMv6)	1Ghz Single Core (ARMv6)
RAM (MB)	512	512	512
Micro USB ports	1	1	1
Network	No	No	Wifi / Bluetooth
Dimensions	65 x 30 x 5 (mm)	65 x 30 x 5 (mm)	65 x 30 x 5 (mm)
Idling power consumption	100mA	100mA	100mA

Version	Raspberry Pi Zero 2 W
Release Date	October 2021
CPU	1Ghz Quad Core (ARMv8)
RAM (MB)	512
Micro USB ports	1
Network	Wifi / Bluetooth
Dimensions	65 x 30 x 5 (mm)
Idling power consumption	100mA

The difference between W and WH is the GPIO header. For the WH version it's already soldered onto the board.

Raspberry Pi Pico

Just a quick word about the Raspberry Pi Pico, as you might wonder why it's not mentioned in this list.

The main difference between the Raspberry Pico and other models is that it is a microcontroller and not a single-board computer. That is why this model is smaller and cheaper. It cannot run an operating system, but can still be used in many electronic projects.

This book is not written for this model, most projects and tutorials included won't work for the Raspberry Pi Pico.

You can find **more info about the Raspberry Pi Pico on the website** if you want to learn more about it. But this book is not a good fit if you goal is to build projects with it.

Model selection

Ok, now that you are aware of all of the models available, how do you make your choice?

If you want to buy an unused model, the first thing to know is that you can almost always choose the latest version.
You'll get all the latest improvements for a similar price.

So, basically, you have four choices at the time of writing:

- Raspberry Pi Model 5
- Raspberry Pi Model 4 or 400
- Raspberry Pi Model Zero 2 W

If you don't care about a $15 difference and don't have any space limitations, you should go for the Raspberry Pi 5. It's the best one on the market. It's available with different amounts of RAM; choose the one that best fits your needs.

In the same case, you can prefer the Raspberry Pi 400 if you'll use it mainly as a desktop computer. Basically, it's almost a Pi 4 : a slightly better CPU, one USB port missing (but no need for a keyboard). Also, GPIO ports are less convenient to use.

If you want a tiny Raspberry Pi, your choice is easy: take the Zero 2.
You'll get fewer features, but it's good enough to do almost everything.

As there are often new models released, I recommend checking my resources pages on RaspberryTips.com regularly, to be sure that nothing else would suit you better:
https://raspberrytips.com/go/resources

Kits

What I haven't told you yet is that the Raspberry Pi is essentially just a motherboard.
Do you only take the main circuit board when you buy a new computer? Nope!

Here is a list of what you'll probably need to use your new Raspberry Pi:

- An SD Card (mandatory)
- A power input (mandatory)
- A screen cable or adapter
- A case
- Heatsinks or a fan
- A mouse and keyboard

Even if you can use your existing accessories, there is a better option: order your Raspberry Pi as part of a kit.

A kit offers all the stuff you need, including the Raspberry Pi. Also, most of the time it will be cheaper than buying everything separately.

The same thing applies for the Raspberry Pi main board. I can't give you the best deals at the time of reading, but you'll get them on my resources pages.

If you prefer, you can also find my recommended accessories on the same page. Aside from the basic stuff, you can also find the specific things you may need; such as audio devices, gaming controllers or Pi extensions.

Where to buy?

Buying a Raspberry Pi is easy as you'll find them everywhere.
The Raspberry Pi official website gives you a list of official resellers in your country, but you can also find them on Amazon, especially if you are looking for a kit or accessories
(all the links are in my resources pages linked above).

And finally, you can even find them at Target or Walmart.
My advice is to compare prices between all of the different options.

While Amazon is often the fastest and most convenient, prices can explode quickly, especially during times of low inventory. Always check the official price before buying on Amazon. While paying a few dollars more can be worth it, you shouldn't have to pay 3 times the normal price, like in 2022.

You can find more details on how to get the Raspberry Pi model you want at the best price on the website:
https://raspberrytips.com/why-are-raspberry-pis-so-expensive/.

Homework

In this book, I'll often give you homework to make sure you understand everything, and you will be ready to continue to progress the next day without being held back or confused by the previous steps.

Today, I want you to read this chapter carefully (again if needed) and make sure you have everything you need.
Tomorrow, we'll plug everything in to your Raspberry Pi; so it's mandatory to get your hands on the Raspberry Pi and accessories before continuing.

Day 2:
Prepare your Raspberry Pi Hardware

Introduction

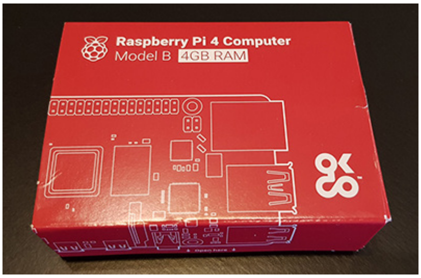
The goal of today is still easy.
I will start off with the basics at a slow and manageable pace to be sure that you learn the essential things to know about your device.
You will discover everything about the hardware you have bought, the components and how to plug everything to be ready for tomorrow.

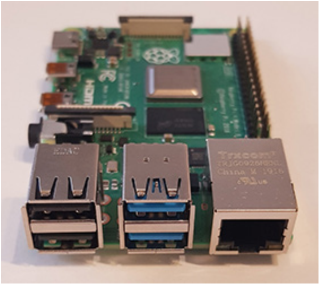
Don't rush this step.
It's easy to do but it's important to know exactly what I'm writing about.
You'll play with the small Raspberry Pi in your hand for the next month (at least); so take the time to really get to know it.

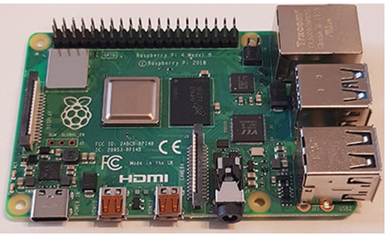
Unboxing

If you haven't already, carefully remove the device from the packaging.
If you bought a kit, do this on a clear table to make sure you don't lose anything.

Here is what it looks like with just the main board:







Raspberry Pi components

Let's take a look at the Raspberry Pi main board.
Raspberry Pi is a very small computer, but it's still a computer; with everything you need to use it as one (processor, memory, storage, wireless card, …).

Can you find all these components on the electronic circuit board?

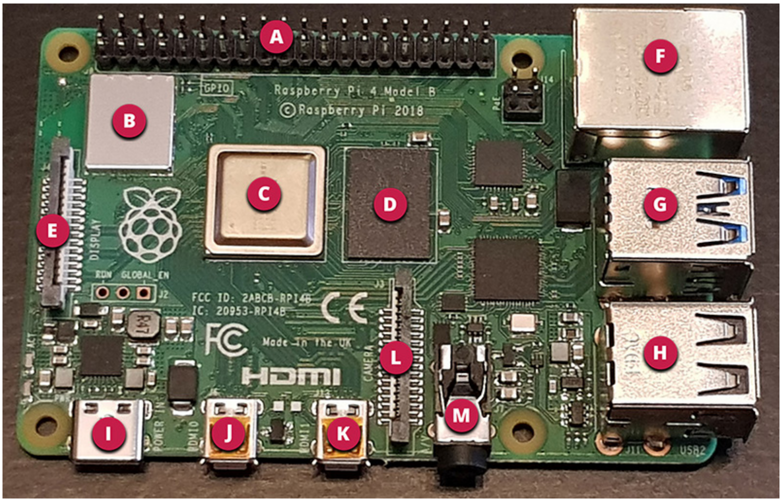
Hold the Raspberry Pi in your hands and try to identify each major element of the circuit board.

Can you find the USB and network ports? The display ports? Maybe the power and GPIO pins?
Harder: What do you think those big squares in the middle of the board are (Processor? Memory? Storage?)?

I'll give you the details for the Raspberry Pi 4 on the next page. As mentioned in the last chapter, not all models look the same. You'll find similar components on B-series models, but if you have a Raspberry Pi A or Zero, it won't have every component on it.

The Raspberry Pi 5 also comes with a new connector called the RP1. It replaces the camera port and provides most of the I/O capabilities for the Raspberry Pi.

Here is a picture of the Raspberry Pi 4 board:



Here are the main components of the Raspberry Pi 4:

- **A**: GPIO Pins
- **B**: Wireless Card
- **C**: Processor (CPU)
- **D**: Memory (RAM)
- **E**: Display connector (DSI)
- **F**: Ethernet port
- **G**: 2x USB3 ports
- **H**: 2x USB2 ports
- **I**: USB-C power input
- **J**: Micro HDMI display output (screen 1)
- **K**: Micro HDMI display output (screen 2)
- **L**: Camera connector
- **M**: Jack 3.5 mm output

The GPIO pins (A) are here to allow the Raspberry Pi to create a link between the Pi and an external component.
You can plug extensions into the board here, or just wire it up to create your own electronic circuit.
We'll see more about that later in the book.

The display connector is used to plug a specific screen into the Raspberry Pi.

The camera connector is similar.
Yes, you can use a USB camera, but you can also find specific devices for Raspberry Pi, you'll plug in this connector too.
You will also find a tutorial in this book to test it :)

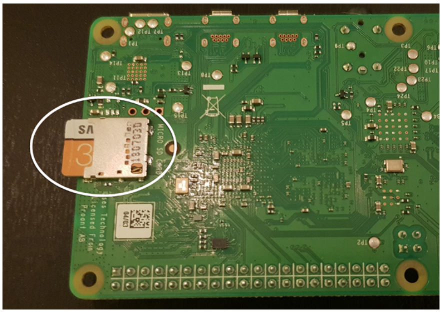
The other components are classic computer components; so you probably already know them.

Plug in your Raspberry Pi

Let's start practicing!

Now that you know what each component is, you can start to plug things into your Raspberry Pi:

- If you have a case, insert the Raspberry Pi inside and close it.
  Even if it's not mandatory to start, I recommend getting one as soon as possible
- Plug your screen into the HDMI port (normal, mini or micro depending on your Pi model).
  On a Raspberry Pi 4, you need to plug the screen on the first connector (J).
  On other models, the port is located in the same place
- If needed, plug an Ethernet cable into the network port (F)
- Plug your mouse and keyboard into the USB ports (USB2 is perfect if you have a Raspberry Pi 4)
- If you already have one, insert the SD card into the dedicated slot.
- I don't have it on the previous picture with components because it's on the back of the board
  Here is what it looks like:



- Don't plug in the power input for now as I have a few things to teach you before starting the Raspberry Pi.

We'll see more about that in the next chapter :)

If needed, you can find each piece of hardware on my website (recommended case, adapters, keyboard, SD card, …)
The direct URL is **https://raspberrytips.com/go/resources**

Raspberry Pi cooling?

Depending on what you want to do with your Raspberry Pi, cooling may have to be considered.
It's not mandatory, but there are a few simple things you can do to make sure your device is cool enough:

- **Monitor the temperature**: You can check the current temperature of your Raspberry Pi easily, either in the desktop bar or with a command. As it's probably too soon to talk about that, I will not teach you this for now. But if you're interested, I will give you a link with everything you need at the end of this paragraph
- **Install heat sinks**: Heat sinks are a small piece of metal you can place on the hottest components of the Pi to help them to cool down (CPU and RAM).
  You can plug the fan into the GPIO pins and fix it to a specific case.
  The case I recommend for the Raspberry Pi 4 is in fact a giant heat sink; you can find it here: **https://raspberrytips.com/beginner-resources**
- **A fan**: For more extreme usage of the Raspberry Pi, a fan is recommended to keep the components at a defined standard temperature.

By the way, I recommend having at least heat sinks or a fan for any Raspberry Pi 4, as it heats up pretty quickly.

As promised, if you want to know more about how to install heat sinks and monitor the temperature, you'll find a complete guide here:
https://raspberrytips.com/go/heatsink

Day 3:
Going Further

Introduction

Before you start using Raspberry Pi, I have a few things to share with you.
We will discuss a bit more about the Raspberry Pi, particularly on topics such as storage and operating systems.

This is the last part of the introduction, so you may already be familiar with most of the material in this chapter. The learning process will really begin in the next chapter.
Slowly but surely you'll master your device! :)

SD Card

On a standard computer you usually have a hard disk to store your work.
There are several formats, but the general idea is the same. The operating system runs on a secure device which is physically attached within your computer. You can use external devices (USBs, DVDs or even SD cards), but the main storage is internal.

On Raspberry Pi, we don't have this.

The only thing you have is a micro SD card, and it looks like this:

You will install your operating system and store all of the data on this small piece of plastic.
It's not the best way to keep data safe, but it works pretty well.

The model that I recommend on my website can store between 32 and 2 TB of data, and works at a maximum speed of 120 MB/s (read). The speed and size are not too different from a standard hard drive, but an SSD will be 2 to 5 times faster.

The main advantage of using a micro SD card is the very small size that it takes on the board.
You can also have several cards, with a different system or project installed on it, and change it as necessary.

Operating systems

The last thing you need to know about Raspberry Pi, would be the systems you can use on it.
It's not the same choices as on your computer (Windows/Mac/Linux). Most of the time it's Linux.
There are a few others, but for the most part it would be Linux distributions.

The main Linux distribution on Raspberry Pi is Raspberry Pi OS (ex Raspbian), and I will use it to teach you how to master your Raspberry Pi in this book.

The name Raspbian comes from combining Raspberry + Debian. It's based on Debian, and adapted to the Raspberry Pi hardware. We will have approximately 15 days in the process to work on this system, so in less than a month you will know everything you need to about Raspberry Pi OS.

You also need to know that it's possible to install other distributions and operating systems on your Raspberry Pi. Here are a few examples:

- **Ubuntu**, Fedora and CentOS: similar to the desktop version if you already used them on PC
- **Retropie**: dedicated to retro-gaming
- **OSMC**: to build a media center easily
- **Windows 10 IoT**: a specific version of Windows for the Raspberry Pi

But the good news is that most of them are also based on Debian, so many of the commands and tips I will give you, may also work on other systems. Debian and Ubuntu are also the most used systems in the computer world, so the skills you'll learn here can help you find a job in this field.

You have no practice or homework for today, but enjoy it because it will progressively become more difficult! :)

Day 4:
Raspberry Pi OS Desktop Installation

Introduction

Today is the day that you have probably been waiting for since the beginning of this book, we'll start up the Raspberry Pi and install Raspberry Pi OS Desktop on it!

I have already written about Raspberry Pi OS, but today we're finally installing and testing it!

Prerequisites

Ok, as it is the first time you will be using it, you'll need a few things to complete this tutorial.
Nothing complicated, you just need:

- **A computer**: to download and flash the Raspberry Pi OS image
  Any operating system with any specs is fine
- **A minimum 8 GB SD card** to flash Raspberry Pi OS on it
- **An SD card reader** on your computer
  If you don't have one integrated into the computer, you will need to buy a USB adapter
- You will probably also need **a Micro-SD / SD card adapter**
- **Your Raspberry Pi** (obviously!)
- **A mouse and keyboard** to use the Raspberry Pi (check out the one in my recommended products)
- **A screen**: a computer screen via HDMI is probably the easiest for beginners.
  I also have the adapters you will need for Pi 4 on my website

If you need to buy some of the items, start by checking my updated resources pages on RaspberryTips, here:
https://raspberrytips.com/go/resources

Raspberry Pi OS Desktop presentation

As you know, Raspberry Pi OS is the most common operating system on Raspberry Pi. To be more precise, it's a Linux distribution.
You can find it in three versions:

- **Lite**: with a terminal interface only, no graphics at all
- **Desktop**: same base with an intuitive graphic interface
- **Full**: similar to the Desktop version, but with more software directly installed

In the next few chapters, we'll use Raspberry Pi OS Desktop.
It's the easiest one to start with, as you don't really need to learn Linux commands for now.

You can also choose the Full version if you want, it will still be compatible with the tutorials.

Download Raspberry Pi OS Desktop

Once your choice made, you can download Raspberry Pi OS from the official website.

In fact, you have two ways of doing this:

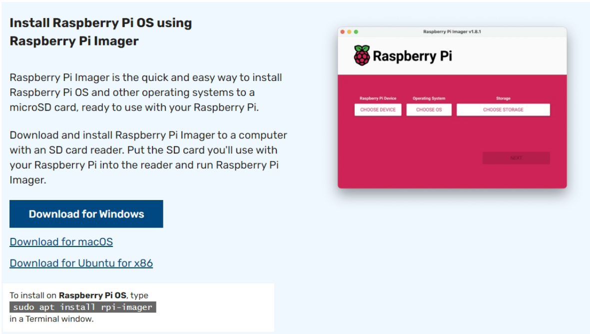
- **Raspberry Pi Imager**: it's a tool developed by the Raspberry Pi Foundation, that will do everything for you (download the image and flash it on your SD card)
- **Balena Etcher**: it's another tool that will only flash the image on your SD card. You have to download the file manually before using it.

Whatever the solution you choose it'll be fine.
I used Etcher a lot in the past because I was used to it (and it is slightly faster when you already have the image file). But I use it less and less now, Imager is my default option in most cases..
I'll show you both in this chapter. The choice is yours.

Option 1: Use Raspberry Pi Imager

Install the tool

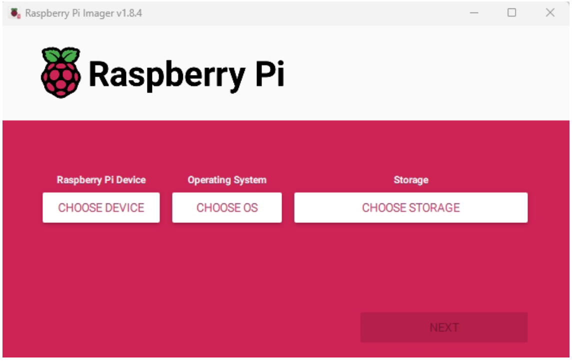
- **Go to** https://raspberrytips.com/go/dl
- **Download the Raspberry Pi Imager** for your operating system
  It's available on Windows, macOS and Ubuntu



- Install it on your computer as any other application

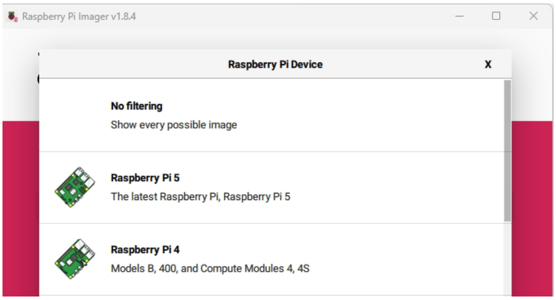
Flash your SD card

- **Start the Raspberry Pi Imager software**
- The default interface looks like this:

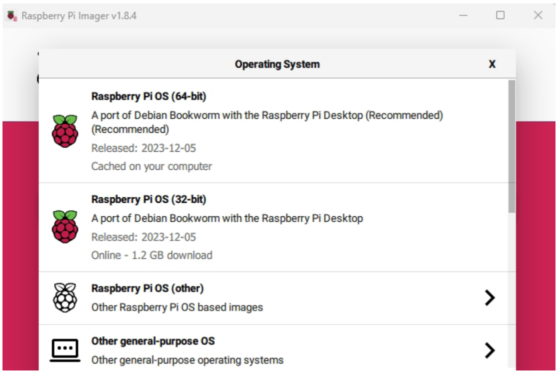


- **There arre three main steps:**
  - "Choose Device", to filter the list of systems when you click on the next button.
  - "Choose OS", where you'll get the OS list I'm referring to.
  - "Choose Storage", to select the support you'll use for your new setup.

- **Click on "Choose Device"**, and select your Raspberry Pi model.



- **Click on "Choose OS"**, and a menu like this will show up:



- Insert your SD card into your computer and **click on "Choose Storage"**.
- Select the corresponding SD card in the list, and **click on "Next"**.

When you click on "Next" for one of the supported operating systems (Raspberry Pi OS and Debian-based distributions, typically), you get access to additional configuration options. This bonus step is named "OS Customisation".

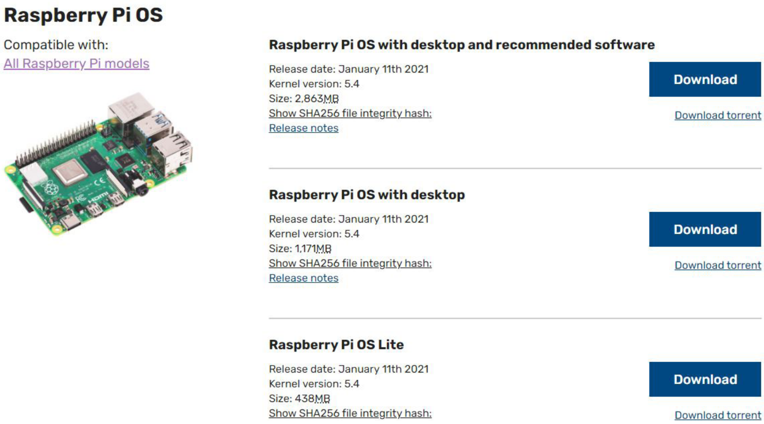
By clicking on "Edit Settings", you can change the default settings for your new system. For example, you can create the first user and password (which is now a mandatory step on new installations), enable SSH or change the localization options (keyboard layout and time zone).

When you click on "Save", options will be kept in memory for future installations. Each time you install a new system, you can decide if you apply the same settings, edit them or just use the default settings.

Option 2: Use Balena Etcher

Download the image

- **Go to https://raspberrytips.com/go/dl**
- Skip the Raspberry Pi Imager section, and **click on "See all download options"**
- On the next page, you can choose the image your want to download:



- Then **click on "Download"** to download the one you want to install

That's all! You'll get a .zip image file saved to your computer and you can use it directly in the next step, you don't need to extract it.

Flash the image on your SD card

To copy the image on an SD card, we can't simply copy and paste the file to it; it doesn't work like that.

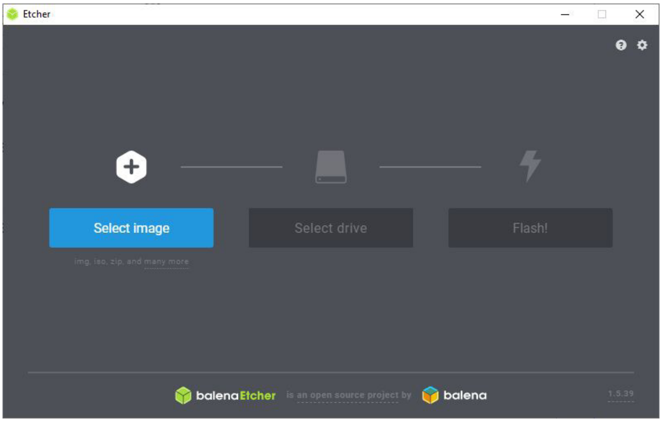
The Raspberry Pi is looking for a specific disk format, with specific partitions and files in it to start the operating system.
Don't stress; there is a software that will do everything for you: Balena Etcher

- **Download and install Etcher** from the official website: **https://raspberrytips.com/go/etcher**

It's a free tool available for Windows, Linux and macOS.
Click the download button under the animated image

- **Install it on your computer** just like any other software.
  For example, on Windows, you have to double-click on the downloaded file to start the setup wizard
  Then, **start it** like any other application.
  Sticking with our Windows example, it's available in the Start menu and probably on the Desktop too

- **A window like this will appear**:



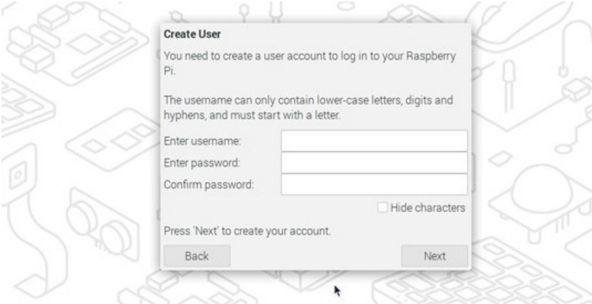
- On the left, **click on "Select Image"** and browse the Raspberry Pi image location to select it.
- In the middle, **check that your SD card is selected** or insert it into your computer
- Finally, **click on "Flash!"** to start the SD card preparation

After a few seconds or minutes, the SD card is ready, and you can remove it from the computer.
We can now move to the first boot of the Raspberry Pi!

First Raspberry Pi OS boot

- Ensure that the power cable is not plugged in
- **Insert the SD card into your Raspberry Pi**
- **Plug in the power input cable** again and wait a few seconds
- Raspberry Pi OS will do a few things automatically and start up

- The system automatically logs in and introduces you to a welcome wizard:



Here, you can configure the basic options for your Pi:

- Select your country and language from the lists
- Create a new user and set a password
- Connect to the Wi-Fi network, if you have one
- Select the default browser between Chromium and Firefox. You can also uninstall the one you don't keep as default.
- Start system updates

Wait for the updates to finish and restart the Raspberry Pi.
It may take a few times depending on your Internet connection speed.
After this, the Raspberry Pi is ready for your next day :)

By the way, if you use Raspberry Pi Imager and complete the OS settings correctly (including username and password), this step is skipped completely on the first boot. You will be taken directly to the main interface.

Homework

Nothing complicated for today, just take a few minutes to reread quickly what you have learned so far.

Day 5:
Raspberry Pi OS Desktop Overview

Introduction

Today, you will take control of Raspberry Pi OS by discovering everything you can do just after the installation.

We'll start with the basic configuration, and then look at all the software that would already be there and what they are useful for.
Finally, I'll focus on three main softwares, and explain to you in more detail what you can do with them.

Prerequisites:

You don't need anything additional here, just the basics:

- Your Raspberry Pi with Raspberry Pi OS Desktop installed (with recommended software if you want)
- Your mouse and keyboard to control the Raspberry Pi.
- Your screen to see what you're doing :)

Raspberry Pi OS desktop basic configuration

Let's start with the Raspberry Pi OS configuration

- On the top left of the screen you have the Raspberry Pi logo.
- **Click on it to open the Main Menu** (I will refer to it like this now).
- **Go into Preferences and click on Raspberry Pi configuration**.
- A window will show up with five tabs: System, Display, Interface, Performance and Localisation.

You can configure from here the main options.

In System you have:

- **Password**: You can change the password for the current user.
- **Hostname**: It's the name of the computer. Not useful if you have only one Raspberry Pi at home.
- **Boot**: You can choose here to boot like now to desktop, or stop at the CLI login prompt.
- **Auto login**: Open the session on boot (you can disable it for security reasons if you want).
- **Browser**: Choose the default web browser for your system.
- **Splash screen**: It's the image during the boot, you can disable it if you want.

In "Display" you have:

- **Screen Blanking**: The system will blank the screen after being inactive for 10 minutes. You can turn this off here.
- **Headless resolution**: The resolution the system will use if you don't plug a monitor.

The following tab is "Interfaces". Here you can enable or disable the main Raspberry Pi services:

- **SSH**: It's a secure protocol to access the Raspberry Pi command line interface from your computer. We'll use it soon.
- **VNC**: similar service, but this time you can have access to the desktop from your computer
- **SPI and I2C** are protocols we'll see later, to use the GPIO pins.
- **Serial Port** is an old port in the computing industry. It doesn't exist anymore, but Raspberry Pi can talk to a serial port via GPIO pins.
- **Serial Console**: Enable this service to use the serial console.

- **1-Wire**: It's also a low-speed data bus you can use through GPIO pins.
- **Remote GPIO**: This option gives you the opportunity to control GPIO ports through the network, for example to enable or disable an external device from a script on another computer.

It's a big list of services and ports, but you'll use the 3 or 4 of the first ones most of the time, so don't stress too much about this.

The next one is "Performance" and includes advanced configuration options like the overlay file system or an option to disable the USB current limit. I recommend leaving the defaults for now.

In the last tab you have the Localisation options.
Basically, it's the same thing you have seen in the welcome wizard:

- **Locale**: Select the interface language.
- **Timezone**: Select your current timezone.
- **Keyboard**: Adjust the keyboard configuration.
- **Wireless LAN Country**: Select your current country to connect to a wireless network.

Once you have set the configuration settings that you want, you are ready to use the Raspberry Pi OS system. Read the next two sections to clearly understand what you can do from here.

Default software overview

Raspberry Pi OS Desktop comes with a little software already installed for you and "Raspberry Pi OS Desktop with recommended software" has a few more.

I will list here all the software you'll find in the main menu, and explain to you briefly what you can do with it. It's everything I have in the Raspberry Pi OS Desktop with recommended software, and it may differ a little depending on the version you have.

Here we go:

- **Programming**
  - Geany Programmer's Editor: A text editor for developers
  - Mathematica: You'll probably never use it if you don't know what it is. Mathematica is a technical computing system.
  - mu: A Python editor for beginners
  - Scratch: The editors for the famous graphical programming language
  - Thonny: An editor to write and run Python code

- **Education**
  - SmartSim: Simulate logic circuit in this simulation program

- **Office**
  - LibreOffice (Base, Calc, Draw, Impress, Math and Writer): An open-source solution to replace Microsoft Office.

- **Internet**
  - Chromium Web Browser: An alternative to Google Chrome.
  - Firefox: another web browser, now available by default on Raspberry Pi OS.
  - Claws Mail: The default email client on your Pi.
  - VNC Viewer: Connect to another PC with this tool.

- **Sound & Video**
  - VLC Media Player: Default media player for Raspberry Pi OS

- **Graphics**
  - Image Viewer: A basic tool to view image files

- **Games**
  - Just a few free games

- **Other**: Including a suite of tools for CAD.
  - KiCad
  - KiCad Gerber Viewer
  - KiCad Imager Converter
  - KiCad PCB Calculator
  - KiCad PCB Editor (Standalone)
  - KiCad Schematic Editor (Standalone)

- **Accessories**
    - Archiver: The default tool to open and create archive files (zip, tar, …)
    - Calculator
    - Document Viewer
    - File Manager: File explorer for Raspberry Pi OS
    - Raspberry Pi Imager
    - Raspberry Pi Diagnostics
    - SD Card Copier
    - Task Manager: Check what's running and debug performance issues
    - Terminal: to run Linux commands directly
    - Text Editor

- **Help**: A few links to useful websites and documentation pages
    - Bookshelf: Free access to some magazines
    - Debian Reference
    - Get Started
    - Help
    - Projects
    - The MagPi

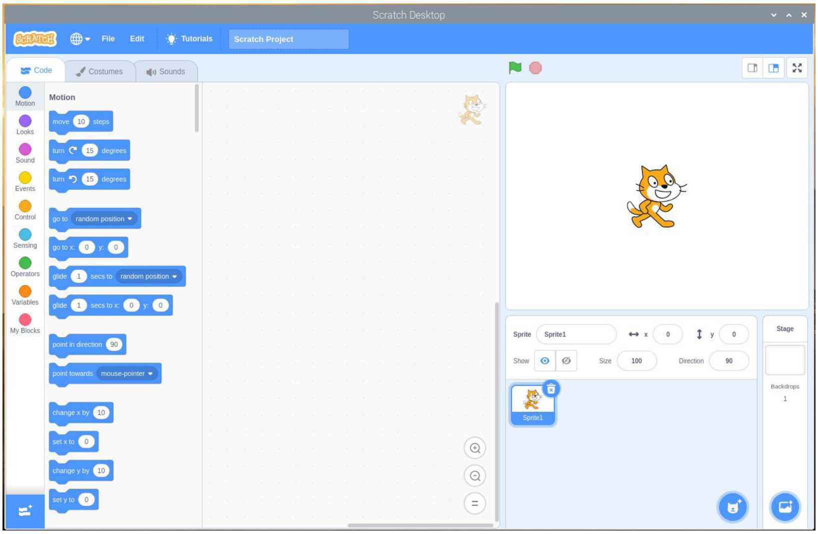
- **Preferences**: I think all of this is transparent. Open them to see exactly what you can configure from here.
    - Add/Remove Software
    - Appearance Settings
    - Main Menu Editor
    - Keyboard and Mouse
    - Print Settings
    - Raspberry Pi Configuration
    - Recommended Software
    - Screen Configuration

- **Run**: start a command without going to the terminal
- **Logout**: stop, reboot or logout

Focus on three software

I will now take some time to introduce three software you probably don't know before using a Raspberry Pi. We'll come back to them in the next chapters, but here is a quick overview

Scratch



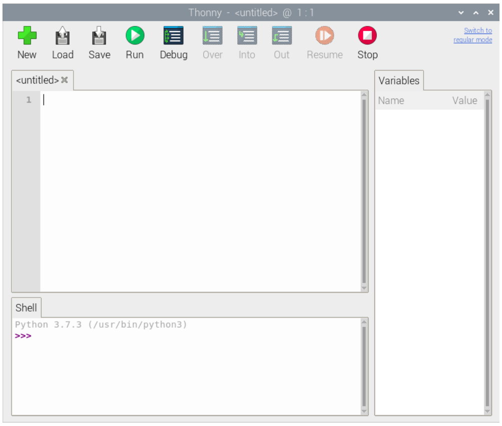
Scratch is a visual language, to learn to program at school, but you can still use it even if you have never done any programming classes and you want to see the logic behind the code.

If you remember the beginning of this book, the goal of the Raspberry Pi is to provide a cheap computer to schools for them to teach programming to their students. Scratch is an essential software on Raspberry Pi OS to do this.

As you can see on the picture, you can create an adventure for the heroes, and make it move depending on your code. But you can also go even further and add text, sounds, conditions and even use GPIO pins to interact with other accessories.

Whatever your age may be, I would really recommend you to test this software. :)

Thonny

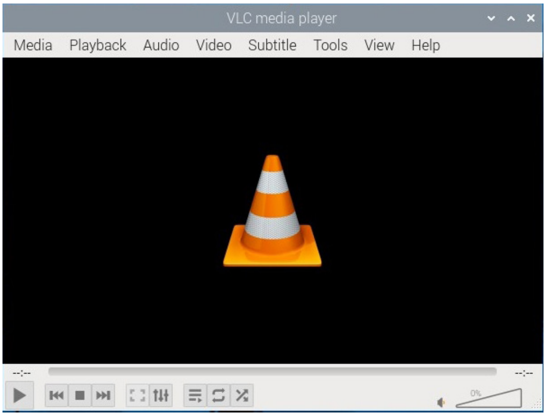


Thonny Python IDE is an editor to create, debug and run Python scripts. I really like this interface because it's simple, compared to other big editors that you can use to code in Python.

In the top bar, you have an easy access to all the main features (new file, load and save, run the code or debug it, etc.). Just under that, you have the script part, where you can type the code you want and at the bottom, you have the shell console which sees what happens when you run your script.

It's easy and it's the perfect interface to start coding in Python.

VLC



You probably already know VLC. It's the most popular media player that natively supports most audio and video formats. It's open source, and whether you use it to listen to music, watch movies, or anything else, it's the perfect companion for your Raspberry Pi.

It's now installed by default on all Raspberry Pi OS versions with a desktop interface and will launch when you double-click on a supported file.

Day 6:
Software management on Raspberry Pi OS Desktop

Introduction

Today we'll start to customize the operating system to your preferences.
That is to say, we will install or uninstall any software to fit your needs.
I'll also give you a shortlist of the best software to install on your new Raspberry Pi OS.

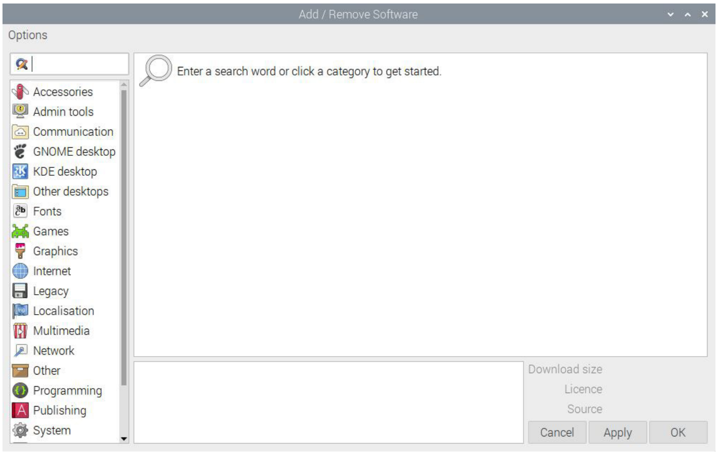
Install and uninstall software from the Desktop

You have several ways to manage software on Raspberry Pi OS Desktop.
The goal here is to show you how to install new software with the default package manager, and how to remove it.

Installing a new software

Firstly, let's see how to install something new on your Raspberry Pi.
With any Desktop version of Raspberry Pi OS, you get a package manager already installed to do this, so it's easy:

- Open the main menu
- Click on "Preferences" and then "Add / Remove Software"
- A window like this will show up:



- On the left you have a menu where you can search for something, or browse a specific category

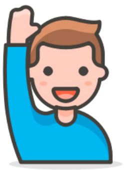
Do a quick search:

- So, the first way is to directly look for the software you want to install
- For example, **type "Thunderbird" in the search engine**
- On the right, the tool will show you the package corresponding to "Thunderbird"
- **Check the box and click on OK to install it**, or Apply to confirm and look for another software
- The software installs itself in a few seconds
- You can close the window

Browse a category

- If you don't know exactly which software you want to install, you can click on a category on the left and browse the list until you find one that fits your needs
- For example, Thunderbird is in the "Internet" category
- When you browse the list, you will notice that all of the software is only sorted by name, so good luck if you have no idea of the name :)
- It's probably easier to find a software name for Raspberry Pi on Google (or RaspberryTips.com), and then search for the software name on this once you know it

Once installed, the software appears in the main menu, in the same category.



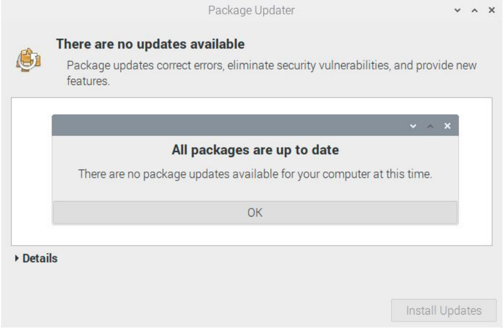
Extra tip

You can also use the "Recommended Software" tool in Main Menu > Preferences, to install or remove main software, recommended by the Raspberry Pi team.

Update software

From the same tool, you can also update your software

- **Open the "Add / Remove Software" tool** from the Main Menu > Preferences
- On the top you have a menu with **"Options"**
- Click on it and then **"Check for Updates"**
- A new window shows up with a list of software to update:



- Click on "Install Updates" to install all of them

That's it, all of your software is now updated.
If there were a lot to update, it's a good idea to reboot now.

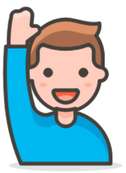
Note: On new Raspberry Pi OS versions, you'll also get a notification in the top-right corner when new updates are available.

Uninstall a software

Ok, now the last thing I want to show you here is how to remove software.
We'll use the same tool to do this:

- From the main menu, go to **"Preferences"** and **"Add / Remove Software"**

- Search for any software you don't use and want to remove.
- In my case, I'll remove Thunderbird
- Just **uncheck the box and press "OK"** to uninstall the program
- It may ask you to remove other dependencies.
  For example, for Thunderbird it asks me to remove Lightning, the calendar extension.
  Just confirm by clicking on "Continue"



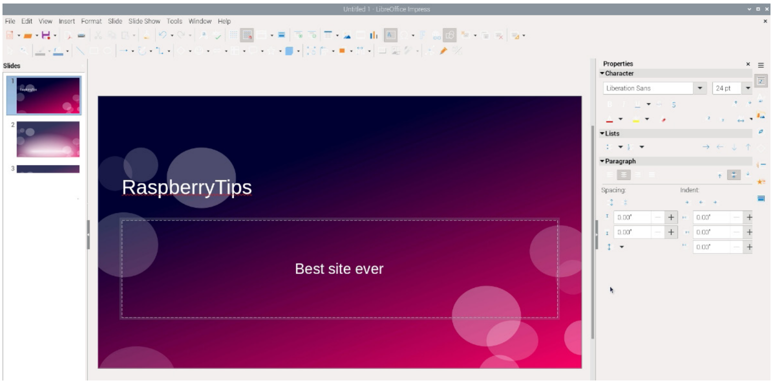
Extra tip

If you want another tool to manage your packages, I like Synaptic.
It offers more features to make your searches easier
You can try it by installing it with the "Add / Remove Software" tool :)

Top 5 software to install on Raspberry Pi OS Desktop

Before closing this chapter, I want to share with you 5 software that you definitely need to try on your Raspberry Pi to use it as a Desktop PC

1 - LibreOffice



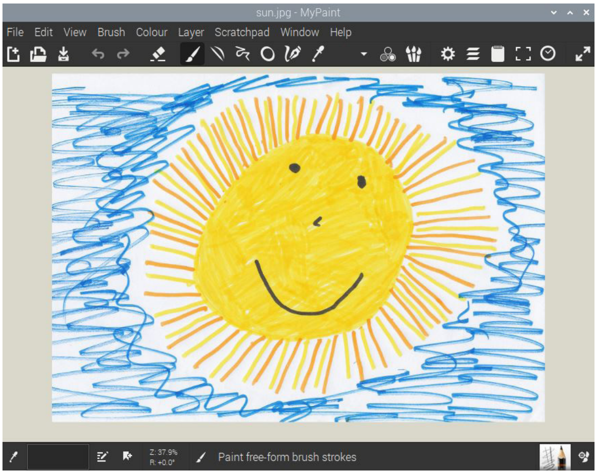
LibreOffice is almost the only decent alternative to Microsoft Office on Linux.

LibreOffice is an entire suite of tools, similar Microsoft:
• Microsoft Word => LibreOffice Writer
• Microsoft Excel => LibreOffice Calc
• Microsoft PowerPoint => LibreOffice Impress

The interface changes a bit, but overall, you should be able to quickly find the main tools
on LibreOffice.

By the way, LibreOffice is one of the applications listed under the "Recommended Software" tool in the main menu. So, it's even easier to install it than other apps mentioned here. Open the tool, find "LibreOffice" in it and check the corresponding box.

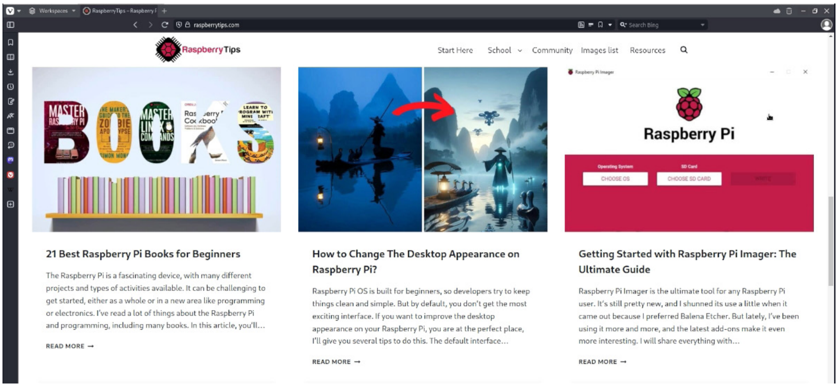
2 - MyPaint



If you know Paint on Windows, it's an alternative for Debian, and it's available on Raspberry Pi OS.

I don't like using GIMP, but you can use it if you do like it.
My favorite for quick image editing on Raspberry Pi OS is MyPaint.
In a small interface, you'll get all the stuff you need in the main toolbar.

3 - Vivaldi



The next one I want to show you is really useful if you are not using a recent Raspberry Pi model. While Raspberry Pi OS now comes with two web browsers, that doesn't mean you can't try other options.

Vivaldi is a good web browser that is fully compatible with your Raspberry Pi.

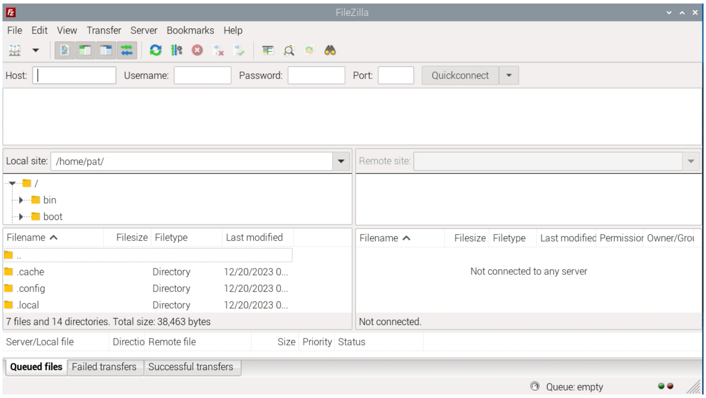
It can be lighter than other browsers, so you'll appreciate it on an old Raspberry Pi model. But I'll also use it as an excuse to introduce a new concept in software installation.

Sometimes you won't be able to find the application you want using the previous methods I mentioned ("Add/Remove Software" or "Recommended Software"). You may have to install it manually by downloading the file from the official website.

This is the case with Vivaldi. It's only available from their website here: https://raspberrytips.com/dl/vivaldi.
You will need to download the "Linux DEB" file that is compatible with your architecture (ARM or ARM64).

Once downloaded, open the Downloads folder, right-click on the DEB file and select "Package Install". Confirm the installation and a few seconds later Vivaldi will be available in the main menu.

4 - FileZilla



FileZilla is an interesting application to try if you work with Linux servers, FTP servers or other devices with which you want to exchange files.

It's best known as an FTP client. FTP stands for "File Transfer Protocol" and is typically used to send files to a remote server (for example, to update the source code of a website).

But FileZilla also supports other protocols, such as SFTP. SFTP works the same way as FTP, but over SSH (remember? It's a secure protocol we can use to access the Raspberry Pi's command line).

So I often use it to transfer files between my computer and my Raspberry Pi. And you can, too. The interface is intuitive, and the application is available in the standard Raspberry Pi OS repository. It's also free for all operating systems, so you can install it on your computer, no matter what system you use.

5 - Sonic Pi

Looking for something a bit more original for your new desktop system? Try Sonic Pi.

Sonic Pi can be seen as a music creation tool, but it's also used as an interesting way to learn how to code.

In short, you write simple lines of code, and these lines tell the computer what sounds to make, like beats or melodies.

It's popular in schools for teaching programming and music because it's fun and interactive. You can experiment with different sounds and create your own songs by typing out code!

Sonic Pi is no longer installed by default, but you can easily find it via the "Add/Remove Software" tool.

Homework

Once you have read this, try to install and remove some software.
Then, browse the different categories to find the software you want to use, and install it.

You can also check this post on RaspberryTips.com to get more software ideas and details: **https://raspberrytips.com/go/apps**

Before going further, you need to be comfortable with this process. I won't explain this again at any other point of the book.

Day 7:
Pimp your Raspberry Pi

Introduction

If you've read the previous chapters, you've probably noticed that, by default, Raspberry Pi OS doesn't have the prettiest operating system interface.

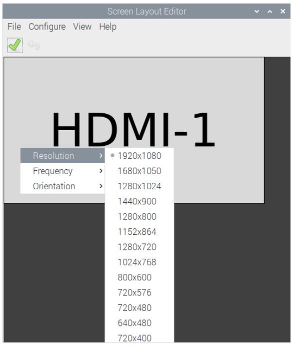
As with any Linux system, you can change almost anything: display options, wallpaper, general theme, and even the boot screen.

This chapter is all about making your new system look better.

Display settings

In this first part, you'll configure the screen and resolution to look exactly as you want it.
Depending on your hardware you may want to adjust this accordingly.
For example, on a 5-inch screen or a 43-inch TV you won't configure it the same way.

- On Raspberry Pi OS, open the **Main Menu**
- Then go to **Preferences > Screen configuration**
- A new window shows up, with the screen configuration interface

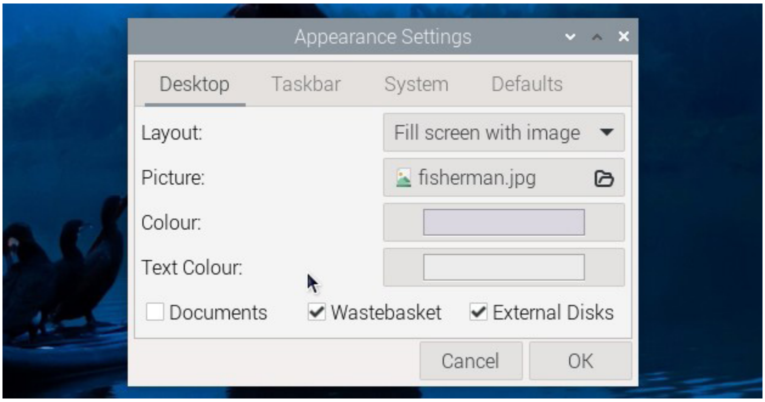


- From here you can configure the screens dispositions (if you have two screens).
- Just as you would on any other operating system, move the squares corresponding to each screen, to dispose them like in reality (side by side for example)
- Resolution, frequency and orientation can be adjusted by right clicking on the square, or by going into **Configure > Screens**
- **Click on the green tick** (or CTRL+Enter) to save your changes

Change the background

The second thing you can do is change the default background.
You may still like it for now, but at the end of the 30-days of reading this book, you'll probably want to try something else :)

The easiest way to do this is by right clicking on the desktop
If needed, the tool is available in the main menu, under "Preferences" > "Appearance Settings".



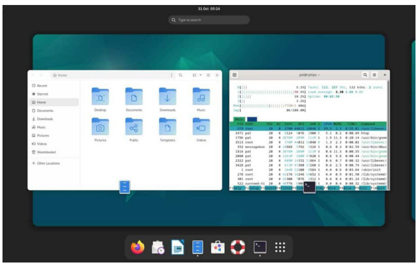
On the first table, click on the "Picture" field to choose one of the default wallpapers.
You can also download and choose a custom wallpaper.

New desktop interface

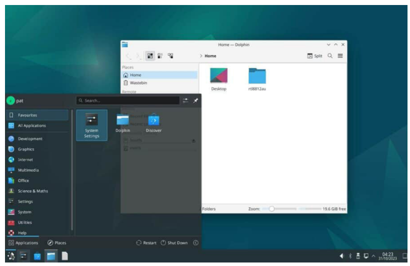
There are different desktop interfaces, named "desktop environments", that can be used on Linux and Raspberry Pi. A desktop environment is a set of display options that allows you to manage most of the graphical aspects of your desktop differently.

Here are some examples of graphical environments that you might know:

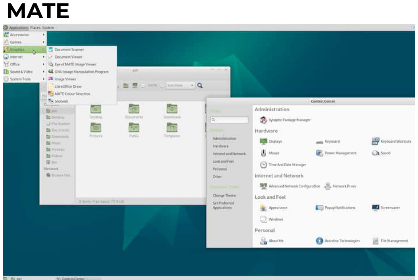
GNOME



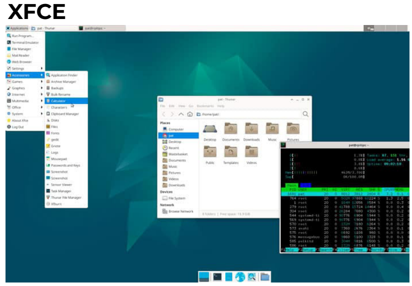
KDE



MATE



XFCE



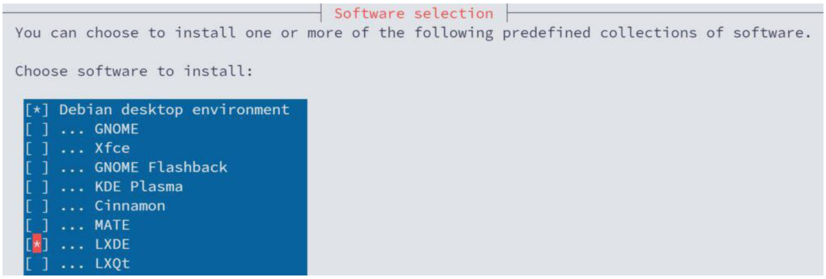
Each of them has some specific characteristics: KDE looks a bit like Windows, with the Start menu on the bottom left. Gnome is the default on Ubuntu.

Raspberry Pi uses its own desktop environment (PiXeL) based on LXDE (another option). It's built and optimized by the Raspberry Pi developers, so it's generally a good idea to keep it. But for the sake of experimentation, I'll show you how to install a different one.

It's possible to manually install the packages needed for your preferred desktop environment. However, the easiest way is to use the "tasksel" command instead.

Open a terminal and type this:
sudo tasksel

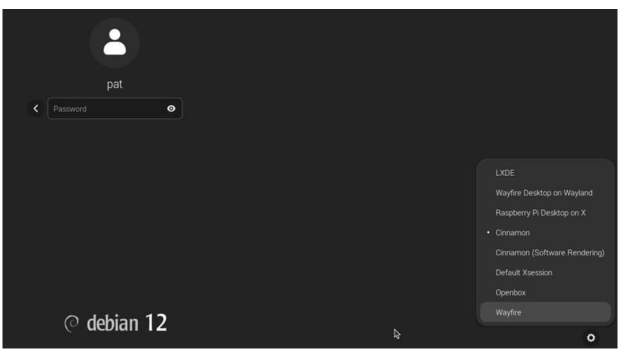
You'll get a window looking like this, where you can select a new desktop environment:



Use the arrows on your keyboard to move the cursor, and the space bar to select/unselect an option. Then, press TAB to switch to the OK/Cancel buttons at the bottom. Select "OK" and press "Enter" to start the installation. The system will install all the required packages automatically.

Once the new desktop environment is installed, we also need a session manager that will ask for the login and password on boot, and leave us the option to choose which desktop environment we want to use. For example, you can install GDM for this:
sudo apt install gdm3

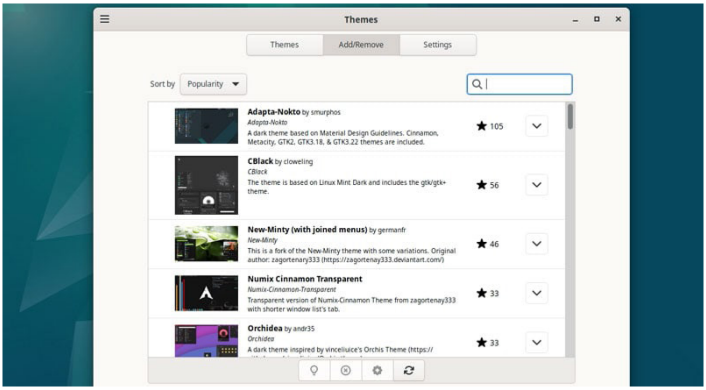
You can now restart the Raspberry Pi. On the next startup, you'll get an interface looking like this to log in, with an option to decide which desktop interface you want to use:



New desktop theme

Something else you can now try is experimenting with different themes for your interface. Using a different theme for your system is less extreme, but not necessarily easy. Each desktop environment has different themes available and different ways to install them.

For example, Cinnamon has one of the best solutions to help you find, download and apply new themes. There is a “Themes” section in the system preferences, where you can see, install and apply many themes in a few clicks:



But in most other environments, there is no great tool to help you with this, so the only way is to use the APT command or the package manager.

Another option is to find a theme online. Many sites will give you a link to download more themes via .deb files or PPA repositories. Feel free to search online for the ideal theme, and in most cases, you will be able to see a preview.

Custom splash screen

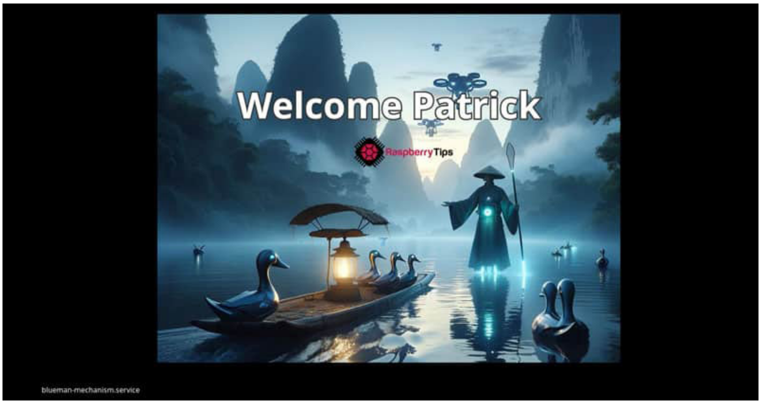
The last thing I want to show you today is how to change the default splash screen.
The splash screen is the picture that shows for a few seconds at the beginning of the boot, with the Raspberry Pi logo and the text “Welcome to the Raspberry Pi Desktop”.

Let's change this:

- **Find a file you like** and resize it to 1024×768.
  A wallpaper, a game you like or your baby's face :)
  You can use this website to resize it: **https://imageresizer.com/**
- Then, **download the file on to your Raspberry Pi**, wherever you want.
- If you know how to do it, you can prepare the file on your computer and transfer it via SFTP.
- Or, even easier, you can host it on an image hosting website and download it from your Pi (ex: **https://imgbb.com/**).
- Finally, you have to enter a few commands to change it.
  Open a terminal and type these three commands in a row:

  ```
  cd /usr/share/plymouth/themes/pix
  cp splash.png splash.png.backup
  cp /home/$USER/Pictures/yourfile.png splash.png
  ```

  The second command creates a backup of the original file, so you can restore it if needed. Make sure to edit the file name and path in the last command.
- You can now **reboot your Raspberry Pi** and see the result :-).



Homework

That's it, you've learned many new things today.
Before the next chapter, I want you to try out different wallpapers, themes and graphical environments until you find the perfect one for you.
So, you can enjoy each time you start your Pi :)

We are almost finished learning all of the Raspberry Pi OS Desktop knowledge you need to have to use your Raspberry Pi as a Desktop PC; only one chapter left!

Day 8:
Remote Access
your Raspberry Pi

Introduction

Today is the last step of our journey on our way to mastering Raspberry Pi OS Desktop.
Even though you can use it for each project and tutorial I'll show you in the following pages, Raspberry Pi OS Desktop is not required.
We'll go into more depth with commands, servers, other operating systems and many other fun things :)

OK, let's get back to this chapter.
Here, you'll learn how to connect remotely to your Raspberry Pi.
It's useful if you don't have a screen on the Raspberry Pi, or if you can't access it physically.

There are many solutions, but we'll focus on the 3 most interesting here so you can try them and choose which one works best for you.

VNC

VNC is one of my favorite tools to remote access a Raspberry Pi (or any computer), but it doesn't work if your Raspberry Pi is on a different network (if it is you'll need to do port forwarding).

We'll keep this simple for now; I assume your Raspberry Pi is on the same network for your first try.

Installation on your Raspberry Pi

- VNC is already installed on Raspberry Pi OS.
- Go to the **"Raspberry Pi Configuration"** tool
- In the "Interfaces" tab, **look for "VNC" and enable it**.
  Apply changes by clicking on "OK"

Once VNC is enabled, you can connect to the Raspberry Pi from another computer, no matter what system you are using on that computer. The only thing you'll need is the Raspberry Pi's IP address (its identifier on the network).

The easiest way to get the IP address on Raspberry Pi OS Desktop is to hover over the network icon (top-right corner) and write down the numbers (usually something like 192.168.1.X). You'll need them later.

Remote access from Linux

On several operating systems you will already have a tool for remote access. For example, **on Ubuntu you have "Remmina"** that can do this.

It's also possible to download and install the TigerVNC client if you prefer. It's available via the package manager on most distributions, or you can download it from their GitHub project: **https://tigervnc.org/**

Find the app in your main menu and start it up.
Fill in the host IP address and connect!
The user and password are those from the default user you created during the installation.

Remote access from Windows

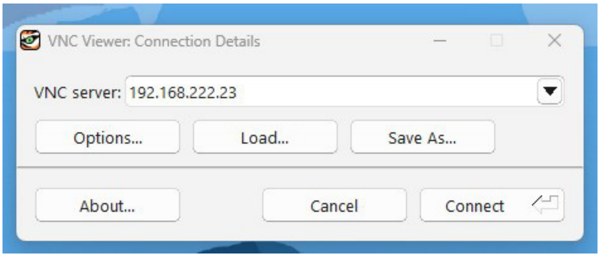
On Windows, it's a bit less intuitive. The official website is not that user-friendly, and we can easily get lost in the Sourceforge folders.

Here are the steps to install TigerVNC binaries manually:

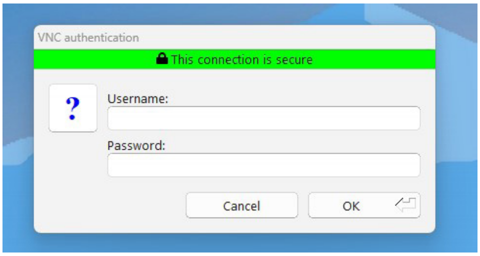
- **Open the TigerVNC page on SourceForge:**
  https://sourceforge.net/projects/tigervnc/files/stable/
  You'll see one folder for each version.
- Find the latest version available and click on it.
- Now click on the file corresponding to your system and architecture.
  For example: "tigervnc64-winwvnc-<version>.exe".
- Once the file is downloaded, double-click on it to install it or start it, like any other executable.

Once the installation is completed, it will be available in the start menu, and you can use TigerVNC client to access your Raspberry Pi remotely.

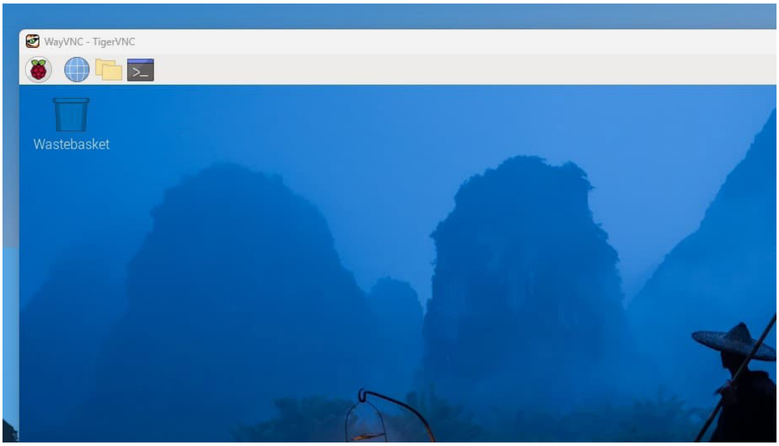
- The first step is to fill the IP address:



- Then enter the user name and password (the same you use on the Raspberry Pi directly).



- And you'll finally get access to the full interface from your computer:



NoMachine

The second one you can try is NoMachine. NoMachine is based on the NX protocol, which was created to improve the performance of the classic X display. Using it is very similar to using VNC.

If used behind a firewall, please note that NoMachine uses port 4000 for connections from the client.

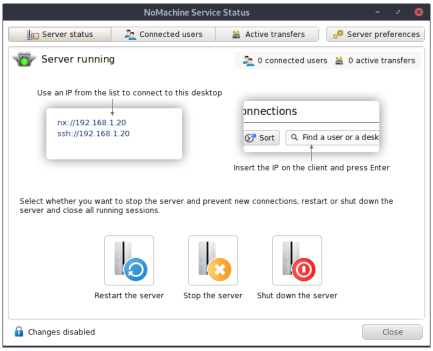
Note: At the time of my last edit, NoMachine was not yet compatible with Wayland, so you probably won't be able to use it on the latest version of Raspberry Pi OS.

Installation on Raspberry Pi

NoMachine is not available in the default repositories, so you have to install it manually

- **Download the file** corresponding to your Raspberry Pi model here: **https://raspberrytips.com/dl/nomachine**
  I Prefer the DEB version, as it's easier to install from Raspberry Pi OS Desktop
- Once downloaded, **double-click on the file to install it**, as we saw in "Day 6". If an archive manager opens the file when you double-click on it, try right-click then open with "Package Install"

Once installed, an icon will also appear near the clock.
You can click on it and choose "Show the service status" to get the current IP address



Access from your computer or smartphone

For any operating system that you want to use to control your Raspberry Pi, NoMachine offers a solution: Windows, macOS, Linux, iOS, Android and even from another Raspberry Pi or ARM device

Just download and install the corresponding file here:
https://raspberrytips.com/dl/nomachine2
Then, start the program from the main menu and enter an IP address to connect to your Raspberry Pi (it will probably find it directly on the first launch).

XRDP

This one is probably my personal favorite.
Xrdp is the Linux alternative to Windows TSE/RDP.
As the server is available natively on Raspberry Pi, and the client already installed on most operating systems, it's very simple to set up.

To install this software, look for "xrdp" in your package manager and install it.

Then, from your computer, find the corresponding tool for your operating system:

- On Windows, it's the **"Remote Desktop Connection"** tool (in the start menu).
- On Linux, if you don't already have an alternative, I recommend using Remmina.
  You can use it with VNC or Xrdp.
- On macOS and smartphone, you have to install "Microsoft Remote Desktop" from the App Store.

Start the app, enter the Raspberry Pi IP address and connect.
Choose your session manager in the list and enter your login and password.

Homework

Well done, you already know a lot of things on Raspberry Pi, and particularly on Raspberry Pi OS Desktop.
In the next few chapters, you'll learn how to use Raspberry Pi OS in command line, so you'll have an even greater knowledge about Raspberry Pi and Linux
But for now, I want you to try all this software and choose the one you prefer. Install it on any computer or smartphone you want to use as a client.
It can be useful daily, if you need to debug your Pi or access it without plugging in a screen.

Teamviewer and X11 forwarding are another two that you can try if you don't like the options already discussed.
If you want to give it a try, check my post on Raspberry Tips about this:
https://raspberrytips.com/go/remote

Day 9:
Discover Raspberry Pi OS Lite

Introduction

On day 4, I told you that Raspberry Pi OS exists in three versions: Full, Desktop and Lite.
Full and desktop are almost the same, both being graphic interfaces like on Windows or macOS, they just have more or less preinstalled software.
Lite is the minimal installation, without desktop; just the terminal version of Raspberry Pi OS.

It's really important to master Raspberry Pi OS with the command line.
If you use SSH to connect to your Raspberry Pi, you'll only get this interface.
Also, if you are using or will use other Linux systems (at work for example), you'll already know how to do the most common things without this interface.

Presentation

So, what is the "Lite" version exactly, and why do you need it?

Raspberry Pi OS Lite is the base version of Raspberry Pi OS Desktop.
The interface is just here to make things easier and more intuitive.
But, everything can be done without interface, by using Linux commands.

As I said before, Raspberry Pi OS Lite is what you get when connecting to your Raspberry Pi via SSH, so you need to know how to configure it or fix issues from there.

In addition, the Lite version is lighter than the Desktop one, meaning it can work smoothly on any Raspberry Pi model.
For example, if you have a Raspberry Pi Zero or an older model, Raspberry Pi OS Lite will work perfectly.

And finally, you'll also build projects where you don't need any interface at all.
If you integrate the Raspberry Pi in a robot or any system without a screen, why would you want to install a desktop on it? To slow it down?

Let's try it, you'll like it :)

Note

You can absolutely keep your Desktop installation and follow the tutorials by using only the terminal tool on it.
But, I highly recommend installing Raspberry Pi OS Lite (on another SD card maybe).
If you are still reading this book, you are probably motivated enough to do it, and it's the best thing to do if your goal is to master Raspberry Pi and Linux.
By staying on Desktop, you'll always think first at the graphic solution to do the same thing, and never remember the basic commands you need to know.

Create your first Raspberry Pi OS Lite SD Card

The procedure is almost the same as for the Desktop version, but here are the main steps again:

- **Open Raspberry Pi Imager.**
- Click on "**Choose Device**", and select your Raspberry Pi model.
- Click on "**Choose OS**", then "Raspberry Pi OS (Other)".
- Click on "**Raspberry Pi OS Lite**".
  You may see two options in this list (64-bit or 32-bit).
  Pick 64-bit if your Raspberry Pi model supports it.
  It might boost the performance slightly, and there is no reason to use 32-bit unless you find incompatibility with your projects later on. For our tests, it's safe to install the 64-bit version.
- Insert your SD card into your computer and click on "**Choose Storage**".
  Select the corresponding SD card in the list, and click on "Next".
- If you filled in the "OS Customisation" options when you flashed the Desktop version, you can keep them. You can also do it now or edit them (e.g. use a different hostname for the Lite version).
  When you're ready, click "Save" and start the writing process.

Wait a few seconds and the SD card will be ready.
You can then eject it from your computer and insert it into the Raspberry Pi.
We'll see how to complete the installation in the next section.

Note: It's also possible to manually download the image file from the official website, and then use another tool to do the same thing (like Balena Etcher), but it's generally easier to use Raspberry Pi Imager.

First boot with Raspberry Pi OS Lite

Insert the SD card into your Raspberry Pi, and start it up.
Wait a few seconds for the boot sequence to finish.

As with the Desktop edition, there is a short wizard on the first boot, where you can set the keyboard layout and create the username and password.

This step is done automatically if you already set these options in the “OS Customisation” tab with Raspberry Pi Imager.

Conclusion

That’s it, you have nothing else to do for the moment.
In the next chapter, I’ll show you how to do the basic configuration and really start to use Raspberry Pi OS Lite.

No homework for today as you can’t do anything more for now.
So, enjoy your day off, and I will see you tomorrow for the next part :)

Day 10:
Raspberry Pi OS Lite Configuration

Introduction

Today, you will learn many important things.
In the last chapter, I showed you how to install Raspberry Pi OS Lite. It was a short and easy chapter, but in this one you'll learn all of the basics about it in just one day!

Firstly, we'll configure the operating system the way you want it.
Then I'll give you other useful tips about passwords management, SSH access and packages management.
Let's go!

Basic configuration

Make sure that the Raspberry Pi OS Lite SD card is in your Raspberry Pi and start it up.
Then log in with the login and password you created in the last chapter.

Now I'll show you how to configure each part of Raspberry Pi OS Lite.
They all use the same tool: ***raspi-config***.
This tool exists only on Raspberry Pi OS, and allows you to do the basic configuration that you need to start (network, keyboard, language, etc.)

Localization options

If needed, you can adjust the keyboard layout at anytime with this tool.
If you didn't pick the right one during the first boot, here is how to change it after that.

- Start raspi-config

```
sudo raspi-config
```

It looks like this:

```
┤ Raspberry Pi Software Configuration Tool (raspi-config) ├

1 System Options       Configure system settings
2 Display Options      Configure display settings
3 Interface Options    Configure connections to peripherals
4 Performance Options  Configure performance settings
5 Localisation Options Configure language and regional settings
6 Advanced Options     Configure advanced settings
8 Update               Update this tool to the latest version
9 About raspi-config   Information about this configuration tool
```

- Go to **"Localisation Options"**
- Then **"Keyboard"**

  - The wizard now asks you to answer a few questions to find the corresponding keyboard:
  - **Keyboard model**: probably the default one
  - **Keyboard layout**: use the arrows to choose "Other"
    - Then select your country from the list
    - Now select the keyboard layout from these country options
    - Finally, select a few other options about the keyboard configuration (AltGr key, Compose key)
      You can keep the default options, as we don't need these keys for Raspberry Pi OS Lite
  - That's it, your keyboard is now properly configured!
- Back to the "Localisation options" menu in *`raspi-config`*, you can also change other options if you need to: Locale, Timezone and WLAN country.
  They have intuitive submenus, so I think you'll be able to configure them fairly easily.

Network options

The second thing you'll probably need to configure before going further are the network options
If you want to use a wireless connection, you have to configure it now

- **Open the Raspberry Pi OS configuration tool**

  ```
  sudo raspi-config
  ```

- Go to **"System Options"**
- Then select **"Wireless LAN"**

- If not already done, you may need to **select your country** (mandatory for the wireless configuration)
- Then, **enter your wireless network SSID (name) and password**
- You can now exit raspi-config

After a few seconds, the network should be up and ready to use.

If you need to use a static IP address, the easiest way is to fix it in your network router interface.
But generally you don't need it, and you can use the host name to access it later (the default host name is "raspberrypi" but you can change it in *raspi-config*).

I won't show you how to do this directly on Raspberry Pi, as it requires a few commands and concepts that we haven't yet come across.
If you need to, you can take a look at this post on RaspberryTips.com:
https://raspberrytips.com/go/ip

That's it for *raspi-config*.
This tool contains a lot more configuration options, but for now we don't need to use them. In the next three sections, I'll show you other configurations you have to do, and how you can do them with or without *raspi-config*.

Password management

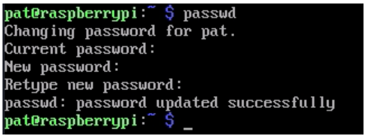
With new versions of Raspberry Pi OS, you now create a unique user and password on the first boot or directly with Raspberry Pi Imager. So it isn't a major concern anymore, but it's still useful to know how to change your password if needed.

You can do this with a command, or in *raspi-config*.
As you are here to learn, I'll start with the command:

- In the Raspberry Pi OS Lite terminal, **type the following command**:

```
passwd
```

- Hit "Enter" to start the tool:



- Enter the current password
- Enter the new password
- And once again to confirm
- That's it, your new Raspberry Pi password is set

If you don't remember the command to do this, you can use ***raspi-config*** to start it

- Open raspi-config

    sudo raspi-config

- Select **"System options"** (first option), then **"Password"**
- The same tool will start in the terminal

SSH remote access

When I started playing with a Raspberry Pi, I was using it on my TV with an old keyboard and mouse, while reading tutorials on my smartphone.
So, I had to decrypt mysterious commands and type them, making lots of mistakes.

But there is a way to be more comfortable while using your Raspberry Pi; you can connect to it from your computer. With Raspberry Pi OS Lite it doesn't change anything as you'll get the same terminal screen, and this will allow you to copy and paste commands directly from a web page, without errors.

SSH server

The most common way to do this is to use SSH.
SSH stands for Secure SHell.
It's a type of software called a secured protocol used to access remote computers.
SSH is installed by default on Raspberry Pi OS Lite, so there is almost nothing that you need to do, you just need to start the service.

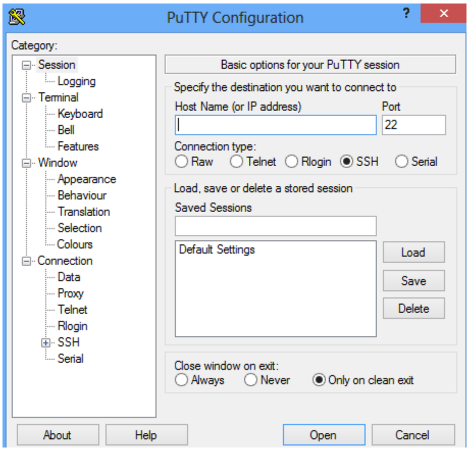
From your Raspberry Pi OS Lite terminal, **run the following command**:

```
sudo service ssh start
```

And that's it, you have nothing else to do.
You can now connect to your Raspberry Pi from your computer.

SSH client

- **For Windows users:**
  - You need to install software to use SSH connections
  - I'm using **Termius** (**https://termius.com/**) because I have several servers and Raspberry Pi to manage.
    But the easiest way is probably to use **Putty** (**https://putty.org/**)
    On recent Windows versions, you can also type a command line to use SSH (*ssh user@IP*).
  - Download Putty for Windows from the official website
  - Once done, start it (double-click on it)
  - A window like this shows up:



  - Enter the Raspberry Pi host name ("raspberrypi" by default)
  - Then click on the "Open" button
  - A black screen shows up, enter the username and password to connect
  - That's it, you now have access to the Raspberry Pi OS Lite terminal from your computer

- **From a Linux or macOS operating system:**
  - On Linux systems (Debian, Ubuntu, macOS, other Raspberry Pi, etc.), SSH is preinstalled by default, so you have nothing more to install
  - To connect to another system, you need to use a command line
  - **Open a terminal** on your computer (it depends on your exact system, often it's in the Accessories or System submenu)
  - Then **use this command** to connect to your Raspberry Pi:

    ```
    ssh <USER>@raspberrypi
    ```

  - Press "Enter" and enter your password to connect
  - You are now connected to the Raspberry Pi, and can use this terminal to follow the next sections and chapters

Auto-start the SSH server

There is an issue with the command I gave you to start the SSH service.
As soon as you stop or reboot the Raspberry Pi, the service stops and won't restart automatically.

If you have only one thing to do, it's OK, as it's safer than leaving the service open all the time.
But you'll probably use it every day to try the commands I'll give you in this book.
You have two solutions to make it start automatically.

SSH auto-start with command lines

In your terminal (SSH or directly on your Raspberry pi), **enter these two commands:**

```
sudo update-rc.d ssh defaults
sudo update-rc.d ssh enable
```

After the next reboot, the SSH service will start automatically.
You can use these commands to auto-start any service (just replace ssh by the service name), so it can be useful to know them.

SSH auto-start with raspi-config

And if you don't remember them, there is also a way to do this in *raspi-config*:

- Start raspi-config

  ```
  sudo raspi-config
  ```

- Go to **“Interface options”** and select **“SSH”**
- Answer yes to the question “Would you like the SSH server to be enabled?”

Close ***raspi-config***, and that’s it; SSH service will start automatically on next boot.

Packages management

On Raspberry Pi OS Desktop, we saw some tools to update, install or remove packages.
It’s almost the same on the Lite version, without the interface :)

You can do everything with the “apt” command.
Here is the syntax:

```
sudo apt <action> <search string>
```

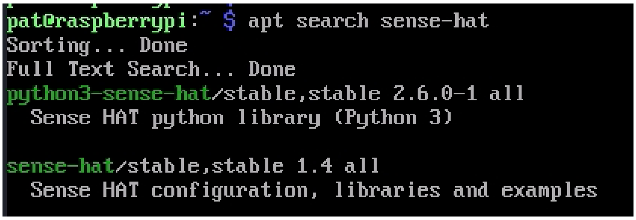
Then you have five actions to know:

- **Update**
  - Goal: Raspberry Pi OS works with a repository server that hosts all packages. Before using apt you need to download the last files inventory from the server. Use this command each time you want to use another apt action
  - Syntax: `sudo apt update`
  - There is no other option to add, just ask to update the repository files

- **Upgrade**
  - Goal: once apt knows the updated list of available files, it can suggest upgrades for your system. This action will download and install all the updated version of your software
  - Syntax: `sudo apt upgrade`
  - There is also no option to add

- **Search**
  - Goal: as you may have seen on Raspberry Pi OS Desktop, package names are not always obvious. The search action allows you to find the package name from a search string
  - Syntax: `apt search <string>`

- Example: *apt search sense-hat*



- This command displays a list of available packages containing the string you search for. You can then use the package name to install it.

- **Install**
  - Goal: as the name suggests, this action allows you to install a specific package
  - Syntax: *sudo apt install <package name>*
  - Example: *sudo apt install python3-sense-hat*
  - Make sure to use the exact package name you get with apt search

- **Remove**
  - Goal: you can use the remove action to uninstall a package, it works almost like the install action
  - Syntax: *sudo apt remove <package name>*
  - Example: *sudo apt remove python3-sense-hat*
  - After using the remove action, you can use the autoremove to also uninstall dependencies that are no longer needed

That's it, you now know how to manage packages on Raspbian (and any Debian-like operating systems).
I often use these commands, even on a desktop OS, as I don't like the slow speed of the graphical tools.

Homework

We have seen many things today, and for your homework I want you to reread this chapter and try everything you can.

These are the basics on Raspberry Pi, and we'll use them all the time in the next chapters:

- The *`raspi-config`* tool
- Remote access with SSH
- Packages management

Make sure you understood everything, try to use SSH to install new packages etc.

In the next chapter I'll teach you many more useful commands.
So, you must really know how to use and understand those we have seen today before moving on to the next chapter.

Day 11:
Raspberry Pi OS Lite
Basic Commands

Introduction

This chapter is probably the biggest one in the book, and also the most complicated.
I'll give you more than 50 commands that you need to know to master a Raspberry Pi, or any other Linux system.

If you are starting on Linux, it won't be a simple mission, but it's important to remember at least the command name for each goal.
Take your time, make sure you understand each command, test it, etc.

At the end of the chapter, I'll give you a cheat sheet that you can download and print to keep all these commands close to you when working on your Raspberry Pi.

Table of contents

I choose to group commands by category, so you can come back to this chapter and easily find the command you are looking for.
Here are the 8 categories we'll come across, with the most useful commands:

1. **Files management**
2. **Network commands**
3. **System updates**
4. **Packages management**
5. **System management**
6. **Raspberry Pi OS commands**
7. **Misc commands**
8. **Warrior commands**

This list is based on my experience on Raspberry and Linux generally.
Everyone uses their system differently and can also use other commands that are not present here.

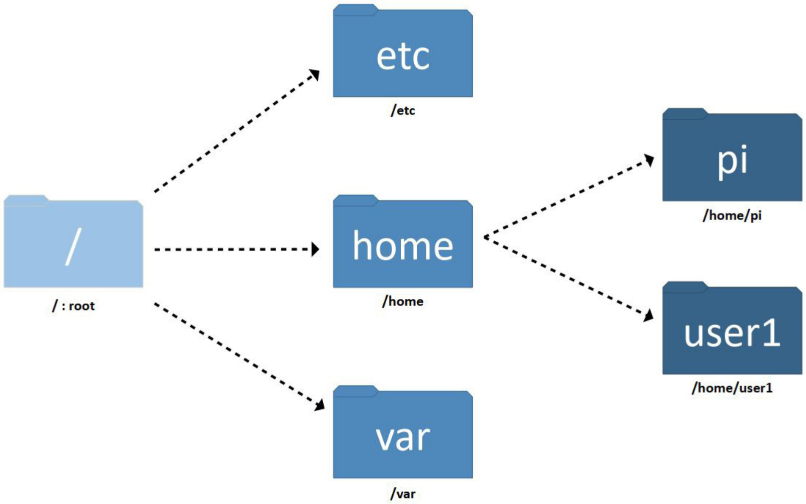
The goal is really to introduce the essential commands to start, and not to make an exhaustive list.

Files management

Linux tree

Before I show you these commands, I need to start with an introduction about the Linux directories tree.

The Linux files organization is a tree, starting at the root: /
Each subfolder created under this is accessible with /
Ex: /home/pi => pi is a subfolder of /home, home is a subfolder in /



Don't forget to use *sudo* if you are not allowed to access the file or directory (*sudo* gives you administrator privileges for the command that follows)

Files management commands

- **cd <path>**: Changes directory, go to the specified folder in the files tree

  ```
  cd /home/pi
  ```

- **ls**: Lists the files and directory in the current or specified folder

```
ls
ls /home/pi
ls -latr /home/pi
```

- **mkdir <folder>**: Creates a new subfolder in the current or specified folder

```
mkdir myfolder
mkdir /home/pi/myfolder
```

- ***cp <file> <destination>***: Copies a file or a directory to another location (to copy a complete directory you need to add the -r parameter for "recursive")

```
cp test.txt /home/pi/Documents/
cp /home/pi/test.txt /home/pi/Documents/
cp -r /home/pi/test/ /home/pi/Documents/
```

- **mv <source> <destination>**: Moves a file or a directory to another location

```
mv /home/pi/test.txt /home/Documents/
mv /home/pi/test/ /home/Documents/
```

- **cat <file>**: Displays all the content of the specified file

```
cat /home/pi/README.txt
```

- **more <file>**: Displays the content of the specified file, page per page (enter or space to continue, q to quit)

```
more /var/log/syslog
```

- **tail <file>**: `Tail` allows you to display the end of a file, it's useful to check new entries in a log file

```
tail /var/log/syslog
```

You can specify the number of lines to display with -n

```
tail -n20 /var/log/syslog
```

And finally, my favorite is the option -f to displays new lines in real-time

```
tail -f /var/log/syslog
```

- **head <file>**: It's the same as `tail` but to display the beginning of a file

```
head /home/pi/file.txt
head -n20 /home/pi/file.txt
```

- **grep <string>**: *Grep* is a powerful tool to search for strings in a text. You can use it to search for something in a file or to filter the output of another command or script
  Basic usage:

  ```
  grep dhcp /var/log/syslog
  ```

  As I say, you can use it on a file or a script output:

  ```
  cat /var/log/syslog | grep dhcp
  /home/pi/myscript.sh | grep error
  ls -latr | grep php
  ```

  And finally, there are many options to use with *grep*, like regular expressions or options to display lines before (-B), after (-A) or around (-C) the search string
  You can also use -v to display everything except the input string

  ```
  grep ‘dhcp\|dns’ /var/log/syslog
  grep -A2 -B4 ‘Fatal error’ /var/log/apache/error.log
  grep -v ‘Notice’ /var/log/apache/error.log
  ```

  If you like this tool, I recommend you read the man page to know exactly what you can do with it

  ```
  man grep
  ```

- **nano <file>**: Nano is text editor. The next chapter is all about Nano options, commands and shortcuts
  But quickly, it allows you to edit a file, and save your changes with (CTRL + O, Enter, CTRL + X)

  ```
  nano /home/pi/myscript.sh
  ```

  You’ll find all available actions at the bottom of the screen

- **rm <file>**: Deletes a file. For a folder add option -rf (recursive and force)

  ```
  rm monscript.sh
  rm -rf /home/pi/scripts/
  ```

- **tar -c**: You can use *tar* to store files into an archive. It’s often used with *gzip* to compress files

  ```
  tar -cvfz archive.tar.gz /home/pi/Documents/mydirectory
  ```

-c: create an archive
-v: verbose
-f: filename of the archive follow
-z: compress files with gzip

- **tar -x**: It's the same command but to extract files

  ```
  tar -xvfz archive.tar.gz
  ```

  -x: extract an archive

- **find**: As the name suggests, *find* is useful to locate files on your Raspberry Pi

  ```
  find /home/pi -iname *.tar.gz
  ```

  There are many options to help you find the file you are looking for (size, last modification date, …), check the man page for more information

- **pwd**: *pwd* lets you see in which directory you are.

  ```
  pwd
  ```

- **tree**: Another great tool to analyze your current location in the files tree. It will show you the entire lower tree (see the example below)

  ```
  tree
  ```

  ```
  pi@raspberrypi:/var/log $ tree
  .
  |-- alternatives.log
  |-- alternatives.log.1
  |-- alternatives.log.2.gz
  |-- apt
  |   |-- eipp.log.xz
  |   |-- history.log
  |   |-- history.log.1.gz
  |   |-- history.log.2.gz
  |   |-- term.log
  |   |-- term.log.1.gz
  |   `-- term.log.2.gz
  ```

Network commands

The recent Raspberry Pi models come with 2 interfaces (Ethernet and Wi-Fi).
The Ethernet is called eth0 and the Wi-Fi is wlan0.
You have to use these names with some commands below.

Configuration

- **ifconfig**: Displays your current network configuration, mainly your IP address if connected

    ```
    ifconfig
    ```

    This command is now deprecated, and will soon be replaced by:

    ```
    ip a
    ```

- **ping <ip>**: Sends a ping packet to another IP on the network to check if the host is alive

    ```
    ping 192.168.1.1
    ```

- **ifup <interface>**: Enables the specified interface

    ```
    sudo ifup eth0
    ```

- **ifdown <interface>**: Disables the specified interface. Can be useful to disable Wi-Fi if you are already connected by cable for example

    ```
    sudo ifdown wlan0
    ```

File transfer and remote access

- **wget <url>**: This command allows you to download a file from the Internet

    ```
    wget https://wordpress.org/latest.zip
    ```

- **ssh <user>@<ip>**: SSH is a network protocol that provides you a way to connect securely to a remote computer

    ```
    ssh root@192.168.1.201
    ```

- **scp <file> <user>@<ip>:<path>**: scp can transfer a file to a remote computer over SSH

    ```
    scp test.txt root@192.168.1.201:/root/
    ```

- **rsync <file> <user>@<ip>:<path>**: rsync does almost the same thing but with a delta comparison algorithm and some optimizations to transfer files faster

    ```
    rsync test.txt root@192.168.1.201:/root/
    rsync -auzr /home/pi/Documents/* /home/pi/backups/Documents/
    ```

    As you can see, you can also use rsync for local files synchronization

Packages management

We have already seen this in the previous chapters, but just as a quick reminder I add them here again.

On Raspberry Pi OS, and generally on any Linux distribution, you'll have a package for each app or command you install.
A list of all available packages is called a repository.
Once installed, you need to update this repository and all your packages regularly to keep your system safe.
These commands explain how to do this, and we'll need sudo for all of these commands.

- **apt update**: Downloads the last repository version for each one you have in your configuration (/etc/apt/sources.list)

  ```
  sudo apt update
  ```

- **apt upgrade**: Updates all installed packages if needed

  ```
  sudo apt upgrade
  ```

- **rpi-update**: Only use this if you know what you do.
  This command will update everything on your Raspberry Pi (firmware, packages, …) and can potentially break something

  ```
  rpi-update
  ```

Install or uninstall packages

- **apt install <package>**: Installs the specified package(s)

  ```
  sudo apt install phpmyadmin
  sudo apt install vim htop
  ```

- **apt remove <package>**: Removes a previously selected package

  ```
  sudo apt remove vim
  ```

- **apt search <search>**: Searches for a package name in the packages list (repository)

  ```
  sudo apt search myadmin
  sudo apt search php
  ```

- **dpkg -l**: Lists all installed packages on your system. You can use grep to find a specific package

```
dpkg -l
dpkg -l | grep myadmin
```

System management

Here are the commands you'll often use to manage your Raspberry Pi system::

- **reboot**: As the name says, this command will restart the Raspberry Pi immediately

```
sudo reboot
```

- **shutdown -h now**: This is to stop the Raspberry Pi immediately

```
sudo shutdown -h now
```

  You can replace "now" by a specific time (`shutdown -h 12:05`)

- **service <servicename> <action>**: This command allows you to start or stop services

```
service apache2 start
service apache2 stop
```

  Sometimes there are other options, depending on the service, for example:

```
service apache2 reload
service apache2 restart
```

  Don't type any action to see all those available:

```
service apache2
```

- **update-rc.d <service> <action>**: On Debian, this command allows you to manage the service start or stop on the system boot.
  To start a service on boot:

```
sudo update-rc.d ssh enable
```

  To disable start of the service:

```
sudo update-rc.d -f ssh remove
```

  The -f option is here to force the symbolic link deletion.

This command is only for service.To start other scripts or commands on boot, you have to edit the /etc/rc.local file

```
sudo nano /etc/rc.local
```

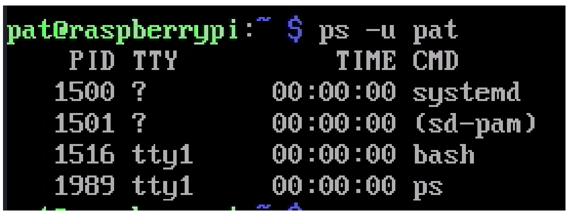
- **ps**: This command displays all running process on your Raspberry Pi.
The basic command is this one to display everything:

```
ps aux
```

You can also display process started by a specific user:

```
ps -u pi
```

This will give you a list like this:



The process ID (PID) can be useful for other commands, to stop it for example (next command)

- **kill <pid>**: The `kill` command allows you to terminate a process. You'll need the process ID to do this (see the previous command)

```
kill 12345
```

Sometimes you may need to use the -9 option to force all related commands to stop.
For example, if you run 20 commands in a script and kill it, it'll continue to the next line, not exit the program, except if you use the -9 option:

```
kill -9 12345
```

You can also use `killall` to stop all occurrences of a program:

```
killall php
```

This command will stop all PHP scripts.
Be aware that this command will immediately stop the process you ask it to, no matter what was going on. It's not a clean stop.
You don't know what the script is doing so it can damage data or corrupt files. This should be used as a last resort, and if possible, on the non-critical process

- **htop**: This tool is an alternative to *top*. It's more user-friendly than *top*, with colors and dynamic load bars:

  ```
  htop
  ```

- **df**: Displays the partitions list, with the disk space used and available for each one:

  ```
  df
  df -h
  ```

  -h option is for the human-readable format

- **vcgencmd measure_temp**: You may not remember it, but this command displays the current CPU temperature:

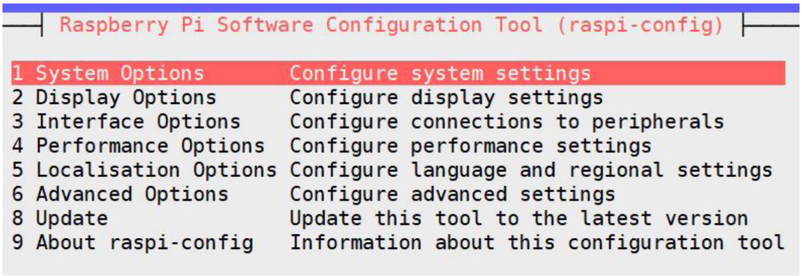
  ```
  vcgencmd measure_temp
  ```

Raspberry Pi OS commands

Most of the commands from this chapter are basically Linux commands.
But Raspberry Pi OS has some exclusive ones that I will introduce first.
These are not all essentials, but you may not know them even if you are good with Linux:

- **raspi-config**: We used it in the previous chapter. This tool allows you to manage all the configuration from a terminal or an SSH connection:

  ```
  sudo raspi-config
  ```



- **libcamera-still**: If you have a camera plugged in to the camera module, this command takes a shot and saves it to an image file:

  ```
  libcamera-still -o image.jpg
  ```

- **libcamera-vid**: It's the same command but for video capture from the camera:

  ```
  libcamera-vid -o video.h264 -t 10000
  ```

  -t parameter is the time of the capture in milliseconds

- **raspi-gpio**: This command allows you to manage the GPIO pins of the Raspberry Pi. You can either set or get a value:

  ```
  raspi-gpio get
  raspi-gpio get 20
  raspi-gpio set 20 a5
  raspi-gpio set 20 op pn dh
  ```

- **raspividyuv** or **raspiyuv**: This command is similar to `libcamera-vid` but for a raw YUV video stream:

  ```
  raspividyuv -o video.yuv
  ```

- **rpi-update**: Only use this if you know what you do. This will update everything on your Raspberry Pi (firmware, packages, …) and can potentially break something:

  ```
  sudo rpi-update
  ```

Note: Since Raspberry Pi OS Bullseye, commands related to the camera have changed. They are no longer "raspistill", "raspivid", etc. I recommend checking my article on how to use a camera on Raspberry Pi for the latest information: https://raspberrytips.com/install-camera-raspberry-pi/

Misc

Here are some other useful commands that I haven't managed to place in the other categories :)

- **history**: Linux stores any command you type in an archive file. History is the command to use to display this list:

  ```
  history
  ```

  You can also clear all the history:

  ```
  history -c
  ```

Or clear one specific entry:

```
history -d 123
```

- **crontab**: Cron is a tool used to schedule tasks on a Raspberry Pi. Crontab is the file where you enter lines for each task to run:

  ```
  crontab -l
  crontab -e
  ```

  -l option to display lines
  -e option to edit lines
  You can use *sudo* before to schedule tasks to run with root privileges

- **screen**: This tool allows you to let something run in the background even if you close your session:

  ```
  screen -s <name>
  screen -r <name>
  ```

  -s option to start a new screen with the following name
  -r option to resume a running screen with this name
  You can forget the name if you want, an ID will be generated, use `screen -r` to find it and `screen -r <ID>` to resume it.
  With only one screen running, `screen -r` will resume it directly

Warrior commands

In this last part, I'll introduce some powerful commands to really help you master your Raspberry Pi.
If you start on Linux, you may not need to know these ones, but if you want to save time or go further on Raspberry Pi, you should know these commands:

- **awk**: *awk* is almost a programming language; it allows you to search for a string and transform them to display it differently.
  So, it'll be difficult to summarize all of the possibilities in a few lines, but I'll try to give you some examples to help you understand it.
  The basic syntax of *awk* is this one:

  ```
  awk [-F] [-v var=value] 'program' file
  ```

  -F is the field separator string (":" or "," for example)
  -v allows you to define variables to use later in the process.
  What I called program is the expression used to process the file, we'll see this part soon.

And finally, the “file” parameter is the file name you want to process.

Here is a basic example:

```
awk -F”:” ‘{print $1}’ /etc/passwd
```

/etc/passwd is the file to parse. The field separator is “:” so we use it in the -F option.Then, in the program string, we ask to display only the first column.
This command will display only a list of usernames.
This is the simplest way to use it but if you want to know more, I recommend reading a dedicated tutorial like this one:
https://raspberrytips.com/go/awk

- **sed**: *sed* allows you to do similar things to awk; it will transform text to what you want.
  As with *awk*, it’s a complex command to master, and I’ll only introduce it here.
  The basic syntax looks like this:

  ```
  sed <option> <script> <file>
  ```

  So it’s very close to *awk* on this.
  Let’s look at an example:

  ```
  sed ‘/^#/d’ /etc/apache2/apache2.conf
  ```

  Generally, in each configuration file, you’ll find a lot of comments to explain what each line is.
  This command will display the apache configuration file without comments.
  We use a regular expression to delete lines starting with #.

  You have to redirect the output to another file to save it:

  ```
  sed ‘/^#/d’ apache2.conf > apache2-nocomment.conf
  ```

  Like for *awk*, this is just a glimpse of what *sed* can do.
  If you want to know more, there is also a good *sed* tutorial on the same website: **https://raspberrytips.com/go/sed**

- **cut**: *cut* is the last way to transform text that I’ll introduce here. It’s less powerful but it’s easier to use, so if *cut* can do it, you’ll probably prefer to use it rather than *awk* or *sed*.
  As the name suggests, *cut* allows you to extract part of a piece of text or file.
  The basic syntax is:

  ```
  cut <options> <file>
  ```

```
echo <string> | cut <options>
```

This will display only "bcd"
-c option is for the character, so basically it'll extract character 2 to 4

Here are other options with a file:

```
cut -d : -f 1 /etc/passwd
```

This will do the same thing as the first example of the `awk` command.
/etc/password is a file with ":" use as a delimiter
-d option is to give the delimiter character (":")
-f option is to indicate the column to extract (f stands for the field).
This will display only the first column and you'll get a list of usernames

- **wc**: *wc* stands for Word Count, it allows you to count everything in a file or stream.
  There are three main options: -l for lines, -w for words and -m for characters.
  There is also the -c option to get the file size.
  Wc without option will give you all of this:

```
wc .bash_history
668  1977 17979 .bash_history
```

  The first column is line count, the second is word count and the last column is the file size in bytes.
  Here are some examples of options:

```
wc -l .bash_history
ls -latr | wc -l
wc -w myfile.txt
```

- **lsof**: *lsof* stands for "List open files". This command displays all files open on your Raspberry Pi.
  This can be useful to know why you can't edit a file, or which file locks the unmount process:

```
lsof
```

- **watch**: If you are waiting for something, in a file or directory, the *`watch`* command can help you to monitor what happens. This will execute the same command every two seconds:

  ```
  watch date
  watch ls -latr
  watch cat output.txt
  ```

  You can also change the refresh rate with the -n option:

  ```
  watch -n10 date
  ```

  This will display the current date every ten seconds

- **netstat**: *`Netstat`* is a powerful tool to monitor what your Raspberry Pi is doing with the network. For example, you can see every port open and every traffic flow.
  But *`netstat`* is a complex tool that I can't explain in detail in a few lines.
  I will only introduce some basic usages to display all listening connections you can use:

  ```
  netstat -l
  ```

  -p option will add the process id (PID)

  ```
  netstat -lp
  ```

  -c option allows you to refresh data continuously

  ```
  netstat -lpc
  ```

  You can find all options in the man page of *`netstat`*

- **dmesg**: This command is useful for troubleshooting your Raspberry Pi boot.
  It will show you every event that happened in the start sequence.
  Here you can see errors with drivers or services and understand why something doesn't work the way you want it to:

  ```
  dmesg
  ```

  You will get a column with the time elapsed since the beginning of the boot and a text explaining what happened.
  There are also normal messages when everything is fine.

Conclusion

Here we are.
Now you should have a better idea of commands to learn to get the most out of your Raspberry Pi.
Depending on your skill level before reading this book, this chapter might be easier or more difficult to read and understand.
Take all the time you need; don't rush!
It's OK if you need two or three days to assimilate how to use these commands.

Homework

As I told you in the introduction, it's one of the hardest chapters in the book.
But, if your goal is to master your Raspberry Pi and any Linux system, you have to go through and work on this chapter, take your time.

So, for homework you can read this chapter again, maybe take some notes about commands you want to use or those where you have difficulties understanding/remembering.
Try them, the best way to remember them is to play about with them on your Raspberry Pi.

If you start on Linux, I highly recommend taking one or two days to use these commands.

As promised in the introduction, you can download a free cheat sheet here with all the commands we just saw (plus a few more):
https://raspberrytips.com/dl/cheatsheet

Feel free to print it and keep it near your computer while working on your Raspberry Pi.

Day 12: Nano Commands and Shortcuts

Introduction

I hope you now have digested the last chapter :)

Today, I'll show you the most useful Nano commands and shortcuts.
Nano is a text file editor, which you'll use all the time at the end of the book and in your future projects.
On Raspberry Pi OS Lite you don't have access to a graphical editor, so you have to use Nano to do everything.

Nano is a powerful tool when you master the main commands and shortcuts.
But, for a beginner, it can be hard to start, that's why I chose to dedicate an entire chapter to this tool.
The goal today is that you start to see how it works, remember a few useful shortcuts, and you'll be ready to use them in the next chapters.

Don't stress, this chapter is smaller and easier than the previous one :)

The Nano command

Before you open a file and see all the shortcuts available, you have to know that Nano offers many options when opening a file.

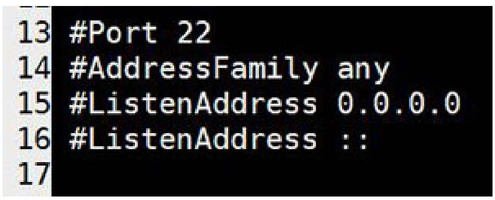
In this first part, I'll show you what you can do directly in the command line and some general shortcuts:

- **Opening Nano**
  nano
  nano <filename>

  The filename can be an existing file or the new filename you want to create
  It can be just the name in the current folder, or a path to the file

Don't forget to use sudo if you need root access to edit the file:
sudo nano /etc/ssh/sshd_config

- **Nano options**
  You can start Nano with many useful options I will show you here:
  - +line,column: open the file and place the cursor at the specified position
  - -B: create a backup file when saving (backup file starts with a ~)
  - -C <dir>: keep each version of the file in the specified backup directory
  - -Y <language>: force the syntax highlighting for the specified language
    Available languages are located in /usr/share/nano/*.nanorc
  - -l: Shows line numbers in the left column



  - -v: Read only mode

- **Exit Nano**
  To exit Nano at any time, you can use CTRL+X

- **Shortcut cancel**
  Nano provides many shortcuts and if you type one by mistake, you can exit the contextual menu with CTRL+C

Files

For file management you don't have many commands to know, but to be sure, here are the two you need to use:

- **Create a new file**
  To create a new file, the easiest way is to tell Nano the filename directly when starting it, like this:
  nano /dir/file.txt
  If you just use Nano to open it, it will ask you a filename when saving

- **Save a file**
  To save a file at any time, you can use CTRL+O
  Don't ask me how they chose the shortcut key ^^
  CTRL+S ? Nooo, it would have been too easy for us!

Search and in file navigation

Another feature I used a lot with vim and never took the time to master on Nano was the search in file feature.

Here is everything you need to know about this:

- **Search for a string**
  To locate a string in a file, you can use CTRL+W
  Then enter the string (or regular expression if you want), and confirm with Enter
  While typing the string, you can use a few options to filter your search results (like case-sensitive search)

- **Next item in the search results**
  To move to the next matching result, use ALT+W

- **Go back to the first line of the file**
  If you are lost in the file, you can move back to the beginning with ALT+\

- **Go to the end of file**
  Similarly, to move directly to the end of the file, use ALT+/
  Please, don't use the down arrow 58 times to do this :)

- **Move one screen up**
  For fast scrolling, use CTRL+Y to move to the previous page of the file

- **Move one screen down**
  And the opposite, CTRL+V is to move to the next page

Navigation through lines and columns

To move more precisely in a file, here are the keys and shortcut to use:

- **Move on a line or from line to line**
  Arrows :)
  I don't really understand why there are so many shortcuts to move from one character or line to another

For me, arrows are the easiest way to do the same
Arrow up: previous line
Arrow down: next line
Arrow left: previous character on the same line
Arrow right: next character on the same line
I think it's a good idea to keep all of this as easy as possible

- **Go directly to the beginning of the line**
  To go to the first character of the line you can use CTRL+A

- **Go to the end of the line**
  And for the last character on the line, use CTRL+E

Content edition

Ok, in this part, we move to the file-edition process.
You can save time by remembering a few shortcuts:

- **Undo**
  To cancel your last action you can use ALT+U

- **Redo**
  If you change your mind, use ALT+E to redo it

- **Copy and paste strings or lines**
  Another thing to keep simple, don't use shortcuts for this
  When you use Nano on your Raspberry Pi you're probably using SSH, so use your client features to manage copy and paste
  Copy: Select a line or piece of text with the mouse
  Paste: Move to the paste spot and right-click to paste

- **Cut an entire line**
  If you want to cut or remove a line, you can use CTRL+K to do this.
  I often use this to remove useless lines

- **Paste the line**
  To paste the line you just cut, use CTRL+U

- **Editing**
  For everything else, use Nano like any other text editor
  Its Nano strength allows you to edit a file directly

Working keys are, for example, Backspace to delete a character, TAB to indent, Enter to add a new line etc.

Nano options

And finally, here are a few other useful shortcuts for Nano I didn't give you in the previous parts:

- **Remove the help**
  To stay focus on your file, you can hide the bottom shortcuts help with ALT+X
  This way you get one or two extra lines of your files on the same screen
  It's the same shortcut to show it again

- **Disable syntax highlighting**
  When syntax highlighting doesn't help, you can disable it while editing with ALT+Y
  Did you never manage to get the ugly dark blue color exactly where you need to focus?

- **Enable backup**
  If you didn't set it when starting Nano, you can enable backup at any time while in nano with ALT+B

- **Display the help for all shortcuts**
  If one shortcut I give you here was not clear, or if you want to see the long list of not so useful shortcuts, use CTRL+G to display the Nano help

Homework

Create a new file and paste or type a random piece of text inside it.
Then, read this chapter again and try each command to see how it works.
Try to be creative and use useful combinations, as this is the best way to remember them.

Feel free to create a cheat sheet on a piece of paper, so you can use it later

If you have five commands or shortcuts to remember, they are here:

- Open a file
- Find a string in the file
- Replace it by something else
- Save
- Exit

You'll use them all the time in the projects we'll build together in the following chapters, so it must become natural if it is not already.
The other commands and shortcuts are useful to save your time, but not required.

Day 13:
Secure your Raspberry Pi

Introduction

Security has always been a major topic in the IT world and still is today.
Almost any network can be hacked, it's just a matter of time.

You may think that you just use your Raspberry Pi for a small project or to make tests, but as soon as it is on the network, someone can use it to access another computer or to use your IP address for bad reasons.

I know that everyone reading this book will have varying levels of computer security knowledge.
If you are at home trying Linux for the first time, or in a big company using a Raspberry Pi to host a web service, you don't have to check all of the same things.

In this chapter I will keep it simple, giving you just the essential information and a few links to find out how to do more if your situation requires more advanced stuff.

System updates

Linux is probably safer than other systems, but it mostly depends on volunteer developers to maintain the packages.
So, we can occasionally see security breaches in old packages.
Generally, they are quickly fixed, but you need to update your system to benefit from the changes.

The good news is that this is simple to do, no matter if you are on Raspberry Pi OS Desktop or Lite.
You have to do this regularly, maybe once a month at least, to keep your system safe.
I will now show you how to do this on Raspberry Pi OS Desktop and Lite.

Raspberry Pi OS Desktop

- Go to the main menu > **Preferences**
- Start the **"Add / Remove Software"** tool
- On the top menu, click on **"Options > Check for Updates"**
- A window shows up, after a few seconds listing the updates you have to do
- If there are some, click on **"Install Updates"**

Raspberry Pi OS Lite

On Raspberry Pi OS Lite, you have two commands to know to do the same thing:

- Update the packages repository:

  ```
  sudo apt update
  ```

- Install the new updates available:

  ```
  sudo apt upgrade
  ```

That's it!
You can also create a script or schedule this task if you know how to do this.

Password change

There is no point to implement security measures if you use a weak password.
It's not longer a generic default password, but still make sure you use something strong enough.

Even at home, your wireless network can be hacked, and your Raspberry Pi running 24/7 can be used to host malicious software.
Yes, the probability is lower than that of a company, but it may happen.

Before showing you how to change it, here is a quick recommendation for a strong password.
The main difficulty for hackers is the password length.

Hard to remember passwords with special characters and random letters are not so useful (especially if you note it somewhere).
I would instead recommend using a long password, 15 characters minimum, using a phrase to remember it easily (but avoid kids names and birth dates ^^).

For example “myfavoritewebsiteisraspberrytips”, is a better password than “8*#SLIPu” for me.
No idea what to use? You can use websites to generate passwords with words inside, here is one: **https://raspberrytips.com/go/pwd**

On Raspberry Pi OS Desktop, you have probably already seen how to do this:

- Go to the **Main Menu > Preferences > Raspberry Pi Configuration**
- In the system tab, click on **“Change Password”**
- Enter the new password you want to use twice, and confirm
- I also recommend disabling the “Auto Login” feature in the same tab if you use a screen on your Raspberry Pi

If anyone can access it without a password, the password you choose doesn’t matter so much :)

On Raspberry Pi OS Lite, one command is enough:

```
passwd
```

Enter the new password twice, and it’s done!

Secure SSH

The easiest way to have a secure SSH service, is to disable it :)
But if you want to use it, you have to follow my tips carefully, in order to keep your Raspberry Pi safe, even with remote access.

SSH means “Secure Shell”, i.e. the link is encrypted, so nobody on the network can view what you’re doing.
But if you keep the default configuration, it’s not secure.

You have already seen how to change the default password, and use a stronger one
We’ll go even further now.
They are not ultimate tips to keep a system secure, but they will help to protect it against low-level hackers (at home, the eventual attacks will mostly come from these kinds of people).

Create new users

Changing the default password is the very minimum that you should do.
But you should also avoid using generic usernames, like “root”, “pi” or “admin”.

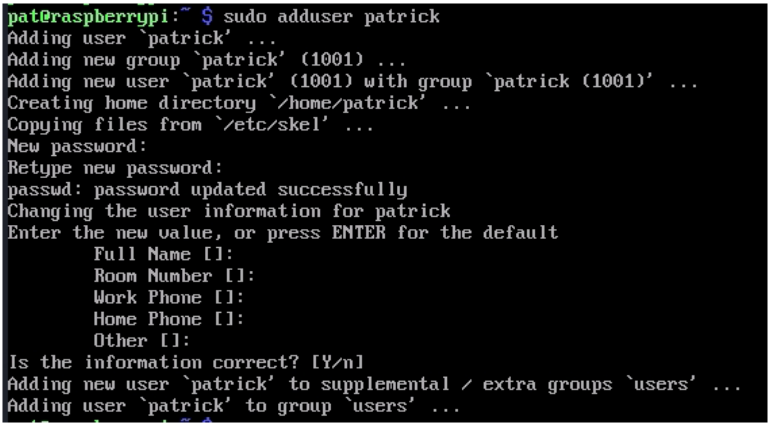
You can create other users to make the attackers take more time.

It's not possible to create new users via the desktop interface, but there is a simple way to it from a terminal:

- Type the following command:

  sudo adduser <username>

  And follow the wizard to set the password and optional details.
  Don't forget to use a strong password, as I explained in the previous part, or this will be useless
- Here is an example:



```
pat@raspberrypi:~ $ sudo adduser patrick
Adding user `patrick' ...
Adding new group `patrick' (1001) ...
Adding new user `patrick' (1001) with group `patrick (1001)' ...
Creating home directory `/home/patrick' ...
Copying files from `/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for patrick
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n]
Adding new user `patrick' to supplemental / extra groups `users' ...
Adding user `patrick' to group `users' ...
```

Now you have a new account with an SSH remote access.
But, you can't use *sudo* with it by default.
You have to add it to the right group like this:

sudo usermod -a -G sudo patrick

Users from the "sudo" group are allowed to use the sudo command

Check that the new user can connect to the Raspberry Pi and use *sudo*.
Now, we need to check that users like "pi" or "root" are not allowed to log in.
Root is disabled by default, so there is nothing to do there.

For generic users (like "pi"), you have two options:

- The first one is to delete it.
  If you have nothing yet on your Raspberry Pi, it's the easiest way:
- Use the command:

```
sudo deluser <username>
```

- That's all you have to do
- If you have many files on the session and want to keep it, you can just block the SSH access:
- Open the SSH configuration file:

```
sudo nano /etc/ssh/sshd_config
```

- Add the following line:

```
AllowUsers <username>
```

- Save and exit
- Reload the SSH configuration:

```
sudo service ssh reload
```

Now you can't connect with this user anymore, so you have to use the new username. This way, even if a hacker tried to crack passwords of generic users for years, he would never get access to it :)

Change the default port

Another thing you can do is to change the default SSH port.
By default, SSH is open on the port 22.
Low levels hackers will just scan your network to look for common open ports, but if you change them, there is a good chance they will ignore you.

- Open the file:

```
sudo nano /etc/ssh/sshd_config
```

- Find the line:

```
#Port 22
```

- Replace with:

```
Port 2222
```

Use the number you want, let's say between 1000 and 30000

- You need to remove the # at the beginning of the line
- Save and exit
- Reload the SSH configuration:

```
sudo service ssh reload
```

That's it!
The next time you want to connect with SSH, don't forget to change the port in Putty or your SSH software.
On Linux or macOS, you have to add an option with the port, for example:

```
ssh username@ip -p2222
```

Don't use passwords

And the last thing you can change in the SSH configuration is to stop using passwords altogether.
If passwords are forgotten, how can hackers find and use them?

How can you do this?
You can use keys to connect to a device with SSH without a password.
A key is a secure identifier for your computer, that will be the only accepted way to connect to the Raspberry Pi.
It works the same way as fingerprint; the system needs to know who you are to grant you access.

This is the most complex part of this chapter but it can be done in three steps:

- **Create a new key for your computer**
  - Linux / macOS:
    - You have only one command to type:

      ```
      ssh-keygen -t rsa
      ```

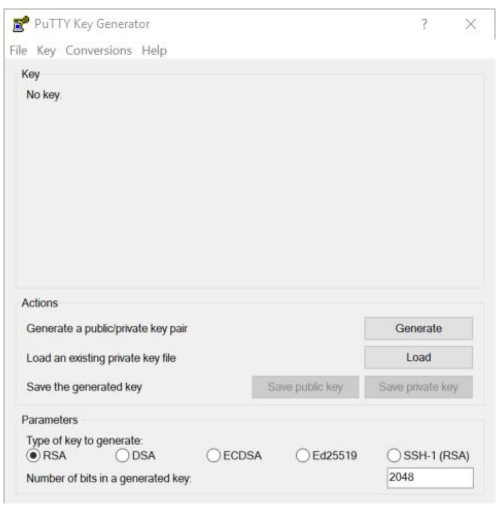
      RSA is an encryption system.
      Keep the default values (everything empty) to generate your keys at the default location without any password (a password isn't mandatory for a home usage, but in a company it's better to use them, with one different password for each person).
      You can change it if you prefer

- Here is how it should look:

```
pat@raspberrypi:~ $ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/pat/.ssh/id_rsa):
Created directory '/home/pat/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/pat/.ssh/id_rsa
Your public key has been saved in /home/pat/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:ESUXtBTcq5zgNG11w9CWaKnoePfdDJ8JChE+rIOns0c pat@raspberrypi
The key's randomart image is:
+---[RSA 3072]----+
|        o+B+.* . |
|         *..* B  |
|        +oo+ + . |
|        =*+ .    |
|      .=S=oo     |
|     ..E+.=  ..  |
|      +.......=o|
|     o .    .. .o+|
|     .+          |
+----[SHA256]-----+
```

The public key is now available in the specified location

- Windows:
  - On Windows, you have to use a specific software to generate your keys, I recommend PuttyGen
    If you installed Putty, you already have it
  - Open the Start Menu, and look for "PuttyGen", then click on it
  - A window like this shows up:



- Click on the “Generate” button
  You need to move your mouse to generate random data
- Once done, you can save the keys with the corresponding buttons
  The public key is the one you need on your Raspberry Pi.
  Keep the private key on your computer

- **Add it to the allowed keys on your Raspberry Pi**
  - Transfer the public keys to your Raspberry Pi
  - Use the easiest way for you: SFTP with Filezilla, USB keys, network share, etc.
  - Then, copy the file content into the authorized_keys file, for example:

    ```
    cat /home/$USER/ssh_key_mycomputer.pub >>
    /home/$USER/.ssh/authorized_keys
    ```

    Replace “pi” with your username if you chose to create a new one

- **Use it in your SSH software**
  - It depends on the software you use
  - On Linux and macOS, your key is automatically used. Or, you can use the -i option
  - On Windows, you can upload it in the MobaXterm options, or use Pageant with Putty
  - Then connect the same way as usual, and you’ll get access to your Raspberry Pi without a password

Don’t forget to deny access with password in your SSH configuration once you confirmed that everything is ok:

- Open the SSH configuration with Nano:

  ```
  sudo nano /etc/ssh/sshd_config
  ```

- Find the line:

  ```
  #PasswordAuthentication yes
  ```

- Comment it out (remove the #) and replace “yes” with “no”:

  ```
  PasswordAuthentication no
  ```

- Save and exit

- Reload the SSH configuration:

```
sudo service ssh reload
```

That's it; your SSH configuration is now optimal for a minimum-security level. You can always go further with security, and I will show you a final way to do so in the next paragraph.

Fail2ban

Fail2ban is mainly an intrusion prevention tool.
It works on the services log file and uses patterns to detect malicious activities.

For example, you can check the SSH log file to list bad login and password attempts, and then block the IP address in your firewall.
Everything is done automatically, with a list of things to monitor and actions to take depending on how many log lines there are in the file.

But you can also monitor other services like Apache or Postfix.

So, it's really a must-have if you want a secure system.
Here is how to install it:

```
sudo apt install fail2ban
```

That's it. Fail2ban will start automatically with the default configuration.
The default configuration is enough to protect you against attacks on SSH.
If you want to go further, you need to edit the configuration
They are all in /etc/fail2ban:

- **fail2ban.conf**: It's the main configuration file with default options for the fail2ban service. You'll probably never change this
- **jail.conf**: In this file, you'll find the things you want to monitor (fail2ban calls this "jails"). Basically, you define here the main configuration for a future test, like the service port and log file
- **jail.d folder**: Here, you'll create the new files for each log file to monitor, with specific options like IP whitelist and ban duration
- **filter.d folder**: Finally, in this folder you can create or edit a filter for each service to monitor. A filter is the definition of what you are looking for in the service log file.

The goal here is just to introduce you to this tool.

It’s a complex tool with many options, and an entire book could be written to explain everything you can do with it
So, I won’t give you any more information here.

If you’re interested, you can check the official website here to learn more about it: **https://raspberrytips.com/go/f2b**

I will also give you one of my articles on RaspberryTips as homework, with one simple example.

Homework

We saw many things today, and you may need time to try them, make cheat sheets and finally remember them.
As usual, I recommend reading this chapter again, taking your time to go through each point. You can take notes if that helps too.

Then, you can also read my posts on this topic to learn more on how to improve the security level of your Raspberry Pi:

- 17 security tips for Raspberry Pi: **https://raspberrytips.com/go/security**
- Fail2ban post: **https://raspberrytips.com/go/fail2ban**

This will also increase your current level of understanding about security in computing.

Tomorrow (in the next chapter), you’ll learn how to do backups of your Raspberry Pi.
You already know many things about Raspberry Pi and Raspberry Pi OS, and after this chapter on backups, we can start trying exciting projects to do more with your device.
Stay motivated, this is just the beginning of your adventure :)

Day 14:
Backup and Restore your Raspberry Pi

Introduction

Before going any further, you need to know how to back up your Raspberry Pi.
As I told you, after this chapter, you'll have all the required knowledge to start trying several amazing projects.
You'll probably really like some and be less interested in other ones. So, if you want to back up the Raspberry Pi state at some point, to come back to it later, this chapter is crucial
(the other way is to use 20 SD cards, one for each project ^^).

There are several ways to back up your Raspberry Pi, I'll try to give you all the useful solutions here:

- Backup specific files only (configuration files for example)
- Backup only MySQL data
- Create an entire image of the SD card to use it again later
- And even create a clone of your Raspberry Pi while running

For each solution, you have to think about how to restore it (and try it if possible) because an hourly backup that you don't know how to restore is useless.

After reading and testing yourself on this chapter, you'll have all the required information, and will be able to make your choice of the best way to use it all in your specific situations in the future.

Partial backup

Presentation

If you are a complete beginner, this part may be long and hard.
I'll introduce a few important things we haven't talked about, so stay focused and take notes :)

The first method you can use if you have a simple installation is to back up only the essential files.
For example, if you use your Raspberry Pi for a security camera, once you back up the configuration file, it's ok, you don't need more.
So, we'll see step by step:

- How to create a script to do that
- How to schedule it if the files are modified
- How to send the file on another computer (in the script or manually)
- And finally, how to restore the files

Creating a basic script

A script is a file containing several commands that you'll run sequentially.
By doing this, you can use only one command that will run all the commands included in the script, without having to remember exactly what each does and why.

Before starting to create the script, you have to identify the files you need to save.
In my script I'll use a file and a folder that you can adapt to your needs:

- /etc/app1/file1.conf
- /etc/app2

Then we need to create a backup folder to store the files:

```
mkdir /home/$USER/backups
```

Use Nano to create the script:

```
nano /usr/local/bin/backup.sh
```

And finally, the first script could look like this

```
#!/bin/bash
/bin/cp /etc/app1/file1.conf /home/$USER/backups
/bin/cp /etc/app2 /home/$USER/backups/ -r
```

As you can see it's a basic script that will overwrite the old backup each time.
Now we'll improve this script at several points.

If you want to try the script at each step, you need to add execution directly to it:

```
sudo chmod +x /usr/local/bin/backup.sh
```

And then run it like this:

```
/usr/local/bin/backup.sh
```

Improve the script

Use variables

The first good practice we'll use to improve the scripts is to add variables.
For example, you have 200 files to backups to /home/pi/backups and tomorrow you want to save them in /media/nas/pi/.
You'll have to edit 200 lines in your script to change the destination folder.

A better way to do this is to create a variable at the beginning with the destination folder path and use it for each line.
I'll also do the same for the "*cp*" command, so if you want to change it to use *rsync* or another command, you'll have only one line to edit.

Here's a better script:

```
#!/bin/bash
DEST_FOLDER='/home/$USER/backups/'
BACKUP_CMD='/bin/cp'
$BACKUP_CMD /etc/app1/file1.conf $DEST_FOLDER
$BACKUP_CMD /etc/app2 $DEST_FOLDER -r
```

The result will be the same, but it will be easier to update.

Compress files

Most of the time we use compression for backup or at least archive files.
You have several ways to do this, but on Linux, *tar* and *gzip* are the most used solutions.
I'll use *tar* to archive all files in one and *gzip* to compress this file.

Here is the new script:

```
#!/bin/bash
DEST_FOLDER='/home/$USER/backups/'
DEST_FILE='backup.tar'
BACKUP_CMD='/bin/tar -rvf'
/bin/rm $DEST_FOLDER/$DEST_FILE.gz
$BACKUP_CMD $DEST_FOLDER/$DEST_FILE /etc/app1/file1.conf
$BACKUP_CMD $DEST_FOLDER/$DEST_FILE /etc/app2
/bin/gzip $DEST_FOLDER/$DEST_FILE
```

I add a new variable DEST_FILE to store the file name of the backup.
"`tar -rvf`" allows you to append several files to one tar file.
"`gzip`" allows you to compress the whole tar file.

Create a different backup file for each day

There is one last step to improve your basic script.
As you can see, with each version, we delete the previous backup each time.
This is not a good thing to do. If there is an issue with the backup, you'll not be able to get an older version (the same thing happens if you run your script daily and a file was deleted 3 days ago).

The good practice is to name the file with the current date and time, like this:

```
#!/bin/bash
DEST_FOLDER='/home/$USER/backups/'
DEST_FILE='backup-$(date +%F_%R).tar'
BACKUP_CMD='/bin/tar -rvf'
$BACKUP_CMD $DEST_FOLDER/$DEST_FILE /etc/app1/file1.conf
$BACKUP_CMD $DEST_FOLDER/$DEST_FILE /etc/app2
/bin/gzip $DEST_FOLDER/$DEST_FILE
```

Nothing changed except we added the date in the DEST_FILE variable, so that we stopped deleting the previous backup.
Each time the script runs, it will now create a new file and keep all the previous versions.

Schedule backups

You now know how to create a script, with variables, etc.
The second big thing you need to learn if you start on Linux is how to schedule a task, and basically a script.

Create a “cron”

Now that your script is created, most of the work is done

All that remains is to program our script so that it starts, for example, every day. For that, we will use the crontab:

- Open the user’s crontab

```
crontab -e
```

- Paste this line at the end of the file

```
0 0 * * * /usr/local/bin/backup.sh
```

  This cron will run your backup script each day at midnight, but you can change it if you want

- Save and quit (CTRL+O, Enter, CTRL+X)

If you back files up with privilege access needed, don’t forget to schedule the script in the root crontab (`sudo crontab -e`).

If you are not comfortable with this, do not hesitate to read my tutorial on scheduling tasks on Raspberry Pi: **https://raspberrytips.com/go/cron**

Delete old files

As you can see, each day a new file will be created, and we never delete it.
What you do to free up some space is delete files older than eight days (or more if needed).

To do this, you can add this command at the end of the script:

```
/usr/bin/find $DEST_FOLDER -mtime +8 -delete
```

By doing this, you’ll have at any time only eight backup files of history in your destination folder.
Feel free to change the number of files you want to keep.

It’s always a good idea to keep several days of history.
You can’t be sure that you will see the issue with your files on the first day.

Copy backup files somewhere else on the network

We now have a backup with some days of history on the SD card, but that's not enough.
If, come tomorrow, your SD card doesn't work anymore, it will not help to have backup files on it.

We need to put the files safely on to another computer to avoid this kind of problem.
You can download it on your computer with WinSCP or FileZilla, but the better way is to schedule a copy on a NAS or any Linux computer, by using "*rsync*" to copy the files.

If you have another computer on Linux (or a NAS), you can transfer backup files on it to keep them safe.
This method will work for macOS too, and the other device can be another Raspberry Pi too :)

Manually

The first way to do this is manually.
If your files never change or if they are not so critical, you can do this once a week or once a month to be safe.

To transfer the files from the Raspberry Pi to the Linux computer, we'll use rsync.
Rsync is a tool commonly used to transfer files between two Linux computers.
So, you need to install it on both machines:

```
sudo apt install rsync
```

Then create a backup folder on your computer:

```
mkdir /home/<USER1>/backups_pi
```

Then you have two ways to transfer files:

- From your computer (in one line):

```
rsync -auzr pi@<RASPBERRY_IP>:/home/<USER2>/backups/*
        /home/<USER1>/backups_pi/
```

- From the Raspberry Pi (in one line):

```
rsync -auzr /home/<USER2>/backups/*
        <USER1>@<COMPUTER_IP>:/home/<USER1>/backups_pi/
```

Don't forget to replace the variables with the Raspberry Pi IP address and your username.
Feel free to delete old files before or after the transfer on your computer.

Automatically

As when you are saving anything on the Raspberry Pi, it's always a good idea to automate this kind of task.
The problem you might have seen is that a password is required when the transfer starts.
To avoid this, you must first make an exchange of SSH keys between the two machines.

For this step, I'll assume it's your computer that triggers the transfer.
If you prefer to transfer in the other direction, you have to do the opposite (generate the keys on the raspberry and allow them on the pc).

Key exchange

We already saw in the last chapter how to create a key exchange, so you can refer to it if need be to do this between the Raspberry Pi and the computer/device you want to use for the backup destination.

Script example

Once everything is ready, we can create a script to do this automatically.
The script should look like this:

```
#!/bin/bash
rsync -auzr pi@[RASPBERRY_IP]:/home/[USER2]/backups/*
        /home/[USER1]/backups_pi/
find /home/[USER1]/backups_pi -mtime -8 -delete
```

Then, schedule the script by adding it in your crontab as explained previously.

Restore

This backup is the easiest one to restore.
You only need to extract files from the archive and send them back to the original folder.

Use the default archive software from your operating system or distribution and extract all the files in a new folder.
Then transfer them to the Raspberry Pi with FileZilla or WinSCP.
You need to transfer them exactly at the good location (an SSH configuration file in /home/pi is useless).

Just read these lines if you are actually doing it, you need to try it.
Most of the backups in the world are useless because the restoration procedure is not written and not even tested.
You can be sure that it works only if you know how to do it and have tested it.

You can even create a script on the destination computer, that will check periodically if the backup file is present as expected.
Something like that for example:

```
#!/bin/bash
if [[ $(find /home/$USER/backups -type f -mtime -1) ]]; then
    echo "Backup OK"
    //DO NOTHING
else
    echo "Backup KO"
    //DO SOMETHING (E-MAIL ?)
fi
```

Export MySQL databases

The second thing you may want to back up is your MySQL database.
The procedure is a bit different, as you can't simply copy files to another location.
I'll explain to you how to do this.

Backup

The best practice for MySQL data backup is to first export and then move this export file to another location.

Export database with this command line:

```
mysqldump -uroot -p[PASSWORD] [DATABASE-NAME] > [DATABASE-NAME].sql
```

This command will create a file with all SQL queries needed to recreate the database from scratch.

You can follow the same steps as seen previously to:

- create a script
- schedule it
- transfer files to another computer

If you have more than one database to back up, put as many lines as needed in your script and change the database name and the file name on each line.

Restore

To recover a lost database, you have to follow the same steps as for files backup.

Once the .sql file back on the Raspberry Pi, you can use this command to import data to a fresh database:

```
mysql -uroot -p[PASSWORD] [DATABASE-NAME] < [BACKUP-FILE].sql
```

The database must be empty to start the import.
So depending on the situation, you must follow one of these two methods:

- import the backup into a database with another name (and then copy only what you are interested in)
- OR rename the corrupt database, recreate it (empty) and then import the backup file

As with file backup, consider testing this procedure at least once

Create an image of the SD card

For more complex systems, you may want to back up the entire SD card.
If you have multiple services and configuration files to save, it's probably the best way to do it.
So in case of a crash, you just have to restore the SD card, and start at the same point.

Backup

The goal of this backup method is to create an image with all your Raspberry Pi files in it.
First you need to stop the Raspberry Pi, and insert the SD card into your computer
The following depends on your operating system:

- **Linux / macOS**
  - Find the SD card device name:

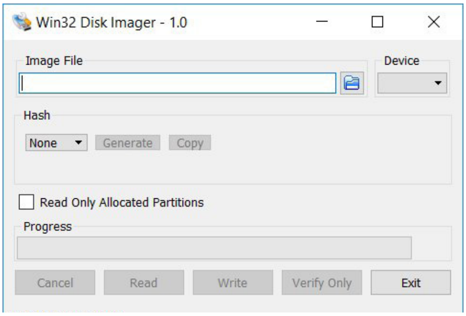
    ```
    sudo fdisk -l
    ```

    This command will display all the storage devices on your computer
    You should see your hard drive, and somewhere at the end of the list your SD card
    Note the name of the device (generally /dev/sdX or /dev/mmcblk0)
  - Create the image:

    ```
    sudo dd bs=4M if=/dev/mmcblk0 of=backup.img
    ```

    Replace the device name and file name if needed
    After a few minutes, you'll get the full backup image from your Raspberry Pi on your computer

- **Windows**
  - The best software on Windows is Win32 Disk Imager
    You can download it here: **https://raspberrytips.com/dl/w32di**
  - You should get a window like this:



    Enter the image destination folder and file name (.img)
  - Select the SD card letter in the device list
  - Click on "Read"
  - This tool will create the image in the selected folder

Restore

The restore process is easy, as it's the same thing as for creating a new Raspberry Pi OS SD card (Day 5)
You can use Etcher to do this:

- Start Etcher
- Choose your backup image on the left
- Select the SD card if needed
- Click on the "Flash" button

You'll get a new SD card with the system in the same state as during the backup

On Linux / Mac you can also use ***dd*** to flash the SD card (same command, just reverse if and of):

```
sudo dd bs=4M if=backup.img of=/dev/mmcblk0
```

If you have an empty SD card, feel free to try this procedure to be sure it's working in your case.

Homework

This chapter can be easy for Linux advanced users, but very hard to understand for complete beginners.
So take the time you need, depending on your level of experience, to integrate everything in this chapter.

If you are already a Linux expert and just here to look for Raspberry Pi skills, you can continue and refer to this chapter when you need it.
But for beginners on Linux, you have to take your time here.
You learn many new notions here, like scripts and crons, that you'll need to use often in this book and in the next few years if you work with a Raspberry Pi or any Linux system.

If you have a critical installation in the future, that you can't stop, but that you want to back up entirely (like the last method), you can read the guide on my website here: **https://raspberrytips.com/go/backup/**

Day 15:
3 servers projects
to start with

Introduction

As I told you in the last chapter, you should now have a strong knowledge about basic usage of your Raspberry Pi.
What you are missing is experience, to put everything into practice, and knowing exactly in which situation you'll use your new skills.

That's why the second part of this book is here to share with you many different projects ideas.
You'll get an overview of what you can do with your device, before being able to fly on your own and try other things!

This chapter is longer than the previous ones, as we'll go through three classic server projects that you can run on Raspberry Pi:

- File server
- Ad blocker
- VPN server

Take the time you need, and please try everything to really make the most of this lesson.

Samba: store and share files on your Raspberry Pi

Let's start with the file server.

Why do you need a file server?

File server usage

A file server is used to store and share files on the local network.
This server allows you to, for example, create automatic backups of your computer.

You can also store files that take up space on your computer but are not critical:

- movies
- software
- ISO files

Raspberry Pi advantages

A standard NAS server will cost you more than $300 for the entry level, make noise and consume a lot of electricity.
A Raspberry Pi will cost you less than $50 and waste almost no power.
You will eventually need to add the price of a hard drive if you need a lot of space, but it is still very reasonable.

This allows you to add a small backup storage server at a lower cost while learning how it works.

How to have more disk space on a Raspberry Pi

If you follow this guide only for testing, you do not necessarily need to read this paragraph.
It is more for people who do not have enough space on their SD card and who want to have more free disk space.

A larger SD card

The first possibility is to buy a larger capacity SD card.

You may only have an SD card of 8 or 16 GB at the moment, so it's going to be a bit limited.
But there are SD cards that go up to 256 GB or more and for a reasonable price.
It may be worthwhile investing in a larger SD card before following this tutorial.

Feel free to check out my recommended products page if you want to know which SD card I currently recommend:
https://raspberrytips.com/go/basic

If you choose this, there is nothing else to do technically other than replacing the SD card in the drive. After you have done this, you can then read the rest of this chapter.

USB hard drive

Another solution is to connect an external USB drive to your Raspberry Pi.
With the 2.5" format you should be able to find disks up to at least 2 TB, which is a much better configuration for file sharing.
It also gives you the advantage of being able to bring it with you even if you are not taking your Raspberry Pi and to read the content of any computer.

Most of the time your hard drive will be recognized and mounted directly by Raspberry Pi OS.
A mount point will be created in: /media/pi/NAME
where NAME depends on the name of your hard drive.

If you have any issues, I advise you to format your disk in a file format readable under Linux (EXT4 for example) and try again.
The fdisk and mount commands could also serve you just as well in the worst cases.

A SATA hard drive?

If you want something professional, you can add a card to connect SATA hard drives to your Raspberry Pi.
This card will allow you to get closer to a classic NAS with faster, better-integrated disks and the ability to manage security with RAID software.

This implementation is another project, so I'll not give you more details about this right now, you just need to know that this is another possibility.

Install Raspberry Pi OS

So, we have all of our prerequisites:
- an SD card
- possibly additional storage

So, let's move on to the installation of Raspberry Pi OS which will be the basis of our file server.

Raspberry Pi OS Lite is the best choice for this project.
If you can remember, I told you that Raspberry Pi OS Lite is a good choice when you don't need a screen.
As you'll probably let your Raspberry Pi in a corner without any screen, mouse and keyboard, Raspberry Pi OS Lite is the option I would keep using for this.

If you want to use it for other things, and just make a try with an existing SD card, you can absolutely do this on Raspberry Pi OS Desktop.

For the installation part, you can refer to Day 9/10.
Here is a quick reminder:

- **Flash Raspberry PI OS again** with Raspberry Pi Imager (on a new SD card if possible).
- **Boot the Raspberry Pi** with the new SD card, and configure the basic settings if needed (keyboard, password, Wi-Fi, etc.).
- **Update the system** with APT.
- **Enable SSH** via raspi-config.

Once your system is ready, you can move on to the file server installation.

Install the file server

We'll now install and configure the file server service: Samba.
Samba is an open-source software, mostly used to share files on Linux systems. It's also able to communicate with Windows and macOS clients, and is available in the default repositories, so it's the perfect tool to do this.

Organization of the file system

Depending on what you initially chose as a storage type, the paths will change a little later.
So I'm going to assume that our available storage space is in /media/share.
Replace this with the right folder you have created.

If you have chosen to share your files directly on the SD card, create a new folder of your choice (/media/share for example):

```
sudo mkdir /media/share
```

In other cases, you should create a mount point in /media/pi/share or something like that:

- Install Samba
- Start by checking for updates:

```
sudo apt-get update
sudo apt-get upgrade
```

- Then install samba:

```
sudo apt-get install samba
```

Once samba is installed you should be able to access the Raspberry Pi with the file explorer:

- **Windows**: \\X.X.X.X
- **Linux/Mac**: smb://X.X.X.X

Replace X.X.X.X with the IP address of the Raspberry Pi

But for now, we don't see a shared folder, and can't create files.
Let's see how to fix this.

Configure Samba for a guest share

Samba is now available, so let's go to the configuration.
The configuration will depend on what you want to do, but I'll explain how to create a basic sharing one:

- Give all permissions to the folder

```
sudo chmod 777 /media/share
```

- Edit the /etc/samba/smb.conf file

```
sudo nano /etc/samba/smb.conf
```

- Add these lines at the end of the file

```
[SharePi]
comment = RaspberryPi
public = yes
writeable = yes
browsable = yes
path = /media/share
create mask = 0777
directory mask = 0777
```

- Save and leave (CTRL+X)
- Restart Samba to apply changes:

```
sudo service smbd restart
```

Test access

We now have a shared folder accessible to all without authentication.

To verify that this works, connect to the network drive:

- **Windows**: Open the file explorer (Win + E) and type the following address:
  \\X.X.X.X\SharePi
- **Linux/Mac**: Open the file explorer and type the following address:
  smb://X.X.X.X/SharePi

Make sure the folder opens, and that you can create files or folders, and then delete them without permission issues.
If so, we can move on.

Permissions

For now, our share is accessible to anyone, even without a password.
If you do not want your little sister to delete your backup history, you will have to forbid guest access!

Configuration

To request a password when connecting, edit the public option in /etc/samba/smb.conf

```
[SharePi]
comment = RaspberryPi
public = no
writeable = yes
browsable = yes
path = /media/share
create mask = 0777
directory mask = 0777
```

You must also tell samba that the user is allowed to connect to shares.

```
sudo smbpasswd -a $USER
```

Then restart Samba:

```
sudo service smbd restart
```

If you try to connect again, it should open an authentication window.
If this is not the case check that you are not already connected in guest mode (first you must disconnect, it depends on your OS, so restart your computer if you can't find this).

On Ubuntu, I had to click on the “Eject” button in front of the mount in the file explorer.

Add another user

If necessary, you can add other user accounts with different passwords.

Create a new user:

```
sudo adduser <username>
```

Enable for samba:

```
sudo smbpasswd -a <username>
```

You can now connect with any of the users

But remember that we put everyone’s rights on the /media/share folder, so all users you create will have the same permissions.

If, for example, you want only the main user to have access to the SharePi share, you will have to change it like this:

```
chown <username> /media/share -R
chmod 700 /media/share -R
```

You can then create multiple shares by changing user permissions for each one.

Going further

From here, you can add other features and improvements to enjoy your new file server the way you want.

I have a guide on this topic on the RaspberryTips website, where I introduce the possibility to use it as a Torrent or DLNA server. You can also look at it to discover a NAS distribution available on Raspberry Pi, and how to use RAID and backups. Feel free to visit this link to learn more:
https://raspberrytips.com/go/mediaserver

Pi-Hole: whole network ad-blocker

The second project I'll show you is an ad-blocker.
You may use AdBlock or an alternative on your computer.
This is something similar, but which applies to the entire network (including any computer, smartphone or tablet).

Pi-Hole presentation

What is Pi-Hole?

Pi-Hole is a free and open-source ad-blocker.
It's different from AdBlock and other browser extensions because it's directly on the network; it's a DNS ad-blocker.

Pi-Hole exists since 2014 and it works on most Linux distributions.
It's available for Debian-like distributions (Debian, Raspberry Pi OS, Ubuntu) and also Fedora/CentOS.

What is a DNS ad-blocker?

You probably use AdBlock to get the same results.
However, AdBlock depends on your browser.

If you have AdBlock on Google Chrome and switch to Firefox, ads are back.
Also, it's not possible to use it on mobile (or at least with any system/browser).

A DNS ad-blocker is easier to manage, as you install Pi-Hole once, and you can use it directly with any device on the same network.
Just set your DNS server with the Raspberry Pi IP, and that's it, you don't have ads anymore!

Pi-Hole installation

Let's move now to the installation process; it's fairly straightforward.

Hardware

Pi-Hole is a lightweight software, and it's designed especially for Raspberry Pi. So, you can use it with any Raspberry Pi model.

In my case, I'm even using it on my Raspberry Pi Zero for this, and it's working very well!

If you don't have a Raspberry Pi yet, you can follow the same procedure with a Linux computer, or even a virtual machine running on Linux.
But a Raspberry Pi is the best choice, as you can keep it running 24/7.

Operating system

Pi-Hole is working on Raspberry Pi OS, so there is no need to use a specific SD card or buy a Raspberry Pi just for this.
You can use any Raspberry Pi you have.

If you want to install Raspberry Pi OS, check chapters 9 and 10 on how to install Raspberry Pi OS on Raspberry Pi.
Raspberry Pi OS Lite is fine if you don't need the graphical interface for something else.

You can absolutely continue with the file server SD card if you want to use both simultaneously.

Before continuing, make sure you have:

- Raspberry Pi OS installed
- A working network
  Wired or wireless is ok on a small network, it's just for DNS requests
- An up-to-date system with the latest packages

  ```
  sudo apt update && sudo apt upgrade -y
  ```

- Enabled SSH access

  ```
  sudo service ssh start
  ```

And that's it, you're ready to go!

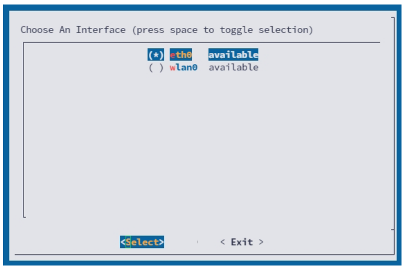
Download and install

Pi-Hole is available on a GitHub repository, and they have a one line command to install it easily: **https://raspberrytips.com/dl/pihole**

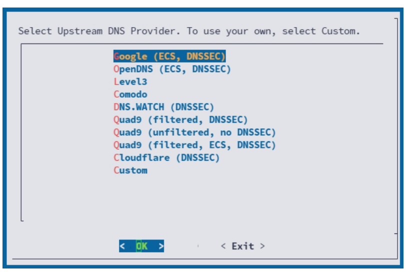
- **Connect to your Raspberry Pi** with SSH
- Login with pi and your password (raspberry by default)
- Copy and enter this command

```
curl -sSL https://install.pi-hole.net | sudo bash
```

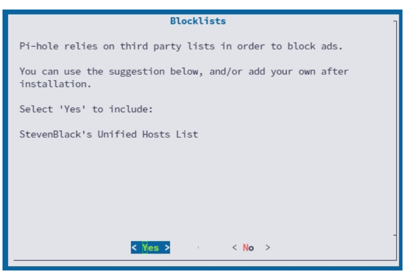
- The installation starts with an ASCII art :)
- Then a standard Debian wizard will start
- Answer Ok to the three firsts questions, there are no other choices anyway.
- Then you may need to choose the network interface (eth0 = wired, wlan0 = Wi-Fi)



- Now pick which DNS provider you want to use. Pi-Hole will forward all requests to this DNS server after filtering them.

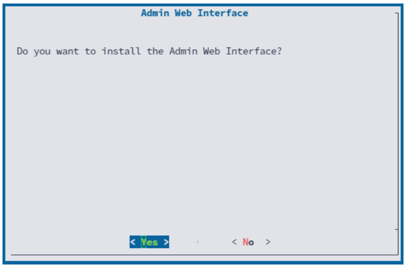


- The next step lets you know which blocklist will be used, but there is only one at this point, so just press Enter to continue.



You can always change this later if needed.

- Then the wizard asks if you want to enable the web admin interface.

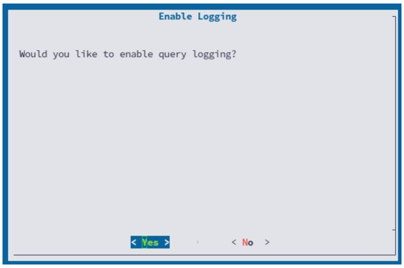


I highly recommend it, it's a major feature of Pi-hole and you'll love it.

- The web interface needs a web server, which is the next step.

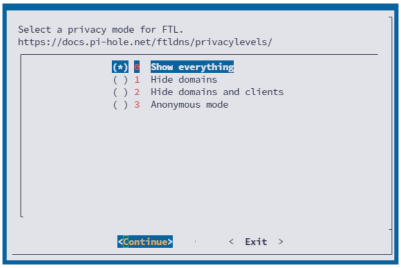
Press "Enter" to install lighttpd and the required modules.

- Now, you need to decide if you want to enable queries log.



If you enable it, the server will log all DNS requests, so you can get statistics about which domains are used the most. I suggest leaving it on for now, and disabling it later if it's not useful to you.

- There are then different privacy modes available:



It's basically how many details you'll get in the web interface about the web usage from all users.
Make sure you have permission to enable this if you're not the only user.
On a company network, it's probably something to check with the legal department, as it can be a major problem if you use this without the consent of the other users.

Ok, it was a long wizard, but basically, you can keep the default value for each screen. And you can change everything later with the web interface anyway.

After this, the installation process continues by downloading, installing and configuring all of the packages, based on your choices.

Don't miss the last screen, as it will give you the web interface URL and password:

I'll explain the web interface in the next section.

But before that, we need to tell the clients to use the new DNS server.

Client Configuration

DHCP configuration

The easiest way to configure all devices at once is to change your DHCP server configuration.
At home, it's probably your Internet router that takes this role.

I won't explain this in detail as it's different on each router.
It can be in the DHCP settings or in the DNS settings.

For me it's in the DNS settings.

The only thing you have to do is remove all values and set the primary DNS server with your Raspberry Pi IP address (for example 192.168.1.17 in my case). Don't set a secondary IPv4 DNS server, unless you have two Raspberry Pi with Pi-Hole on your network.

This way, any device using DHCP will now use the Pi-Hole ad-blocker from your network as DNS server.
It can take a few hours to update the settings on all devices, so be patient or disconnect/reconnect them manually.

Static configuration
If you don't have access to these settings, or want to try them first, you can only edit your computer settings.

On Windows 10:

- Do a right click on the **"Start Menu"** and choose **"Network Connections"**
- Then click on **"Change adapter settings"**
- Do a right click on your current connection and choose **"Properties"**

- Double click on **"Internet Protocol Version 4 (TCP/IPv4)"**
- **Set the DNS server** to static and enter your Raspberry Pi IP Address
- Keep the secondary DNS server empty

On Linux and Mac OS:

If you have a graphical interface, you'll find the network settings in the System Preferences.
If not, you can edit the /etc/resolv.conf file and replace the current DNS server with the Raspberry Pi IP address.

Pi-Hole web interface

Pi-Hole comes with a great web interface, let's check this now.

First login

To access the web interface, open your browser and go to
http://<RASPBERRYPI_IP>/admin
In my case, it's http://192.168.1.17/admin

If you didn't note the IP address during the installation wizard, you can find it with "ifconfig"

Once on the web page, click Login and enter the password you set previously.
If everything is correct, you'll get something like this:

On this page, you can see all the statistics about your Internet usage:

- **Total queries**: how many requests you made
- **Queries blocked**: how many requests it blocked
- **Percent Blocked**: queries blocked / total queries
- **Domains on Blacklist**: how many domain names you have in the blacklists

You also have graphs about queries and client evolution in the last few hours.
Then there are two pie charts about queries types.
And below (we don't see it on the picture), you have top domains and top blocked domains, and the same thing for clients.

That's pretty cool, and it's only the first page :)

Menu sections

In the main menu you'll find links to other pages and sections:

- **Dashboard**: the page we just saw
- **Query Log**: if enabled, all the queries you made and if Pi-Hole blocked them or not
- **Long-term data**: dashboard is showing data about the last 24 hours, in this section you can specify the date range
- **Whitelist**: here you can add domains to whitelist (if they are in the blacklist and you don't want to block them). For example, I'm using Google Analytics for my website, and it's in the blacklist, so I added the URL here
- **Disable**: Sometimes you may need to disable Pi-Hole for a few minutes or permanently to do something specific on your computer. This is where you can do it
- **Tools**: a bunch of tools to debug Pi-Hole, I'll let you discover this part on your own
- **Settings**: all the configuration menus; we'll come to this in the next paragraph
- **Log out / Donate / Help**: these should be clear enough :)

Settings

Let's look specifically at the settings part now.
When you click on the "Settings" menu item, you'll get 7 tabs with settings forms:

- **System**: In this first tab, you can check the information about the network and the Pi-Hole version and usage. You have also set a “Danger Zone” to restart services, reboot Raspberry Pi or clear logs
- **Adlists**: In the adlists, you can manage blacklisted domains. You can disable or delete default blacklists, or add a new one
- **DNS**: The DNS tab is where you can change your DNS providers and configure more advanced DNS settings
- **DHCP**: This tab allows you to enable a DHCP server on Pi-Hole. That’s another way to configure clients if your current router doesn’t do this. Don’t forget to disable the old DNS server if you want to try this
- **API / Web interface**: The next tab is for various settings about the interface. You can mask domains for the dashboard and even customize the interface display
- **Privacy**: In this tab, you can change the privacy level we saw in the installation wizard to increase or decrease the privacy level on the interface
- **Teleporter**: Finally, what they call “Teleporter” is just an Import/Export tool, used to move the configuration from one Raspberry Pi to another

Pi-Hole tips

Before ending this part, there are a few other things you need to know about Pi-Hole.

Update the blacklist

As you can guess, domain blacklists are changing every day.
Like an antivirus software, you need to update them regularly to stay well protected.

To do this, go to Tools > Update Gravity and click on Update.

Pi-Hole will download each blacklist file from the specified sources and reload the configuration.

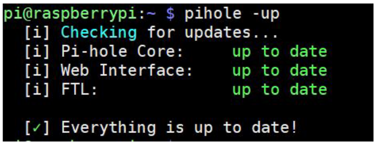
Update software

The other thing you need to update is the software.
Pi-Hole is releasing updates and fixes regularly, so you need to update it regularly too.
The procedure for this is simple:

- **Connect to Raspberry Pi OS** via SSH
- **Type this command:**

sudo pihole -up

- You’ll get something like this:



- In my case I’m already up to date, but if there is an update, you’ll see it here

OpenVPN: access your home network from somewhere else

The last project I want to show you today is how to install your own VPN server on a Raspberry Pi.
A VPN can work in both directions, but the goal here is to give you access to your home network from anywhere in the world.

This can be useful if you have devices at home that you want to access remotely (for example to download a file or check something).
Don’t panic if you are a beginner, I’ll explain right now what a VPN is and how it works.

What’s a VPN?

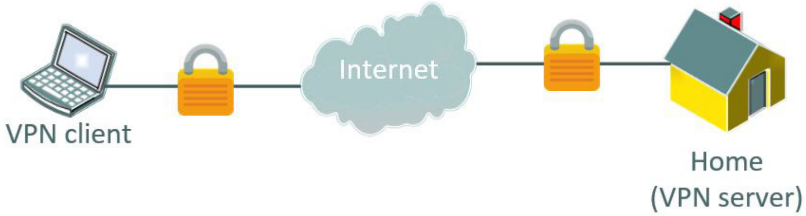
Before going any further, let’s start with a few reminders about VPNs.

Introduction

VPN stands for Virtual Private Network, and that’s exactly what it is. When connected to a VPN, it’s as if you were on a private network between you and the VPN server.The main goal of a VPN is to encapsulate your data in a secure tunnel between you and the VPN server.

Let’s look at an example.
If you share a web server at home with port forwarding (public_ip:80 => local_ip:80), data could be accessible to hackers, as data flows in clear on the network (man in the middle attacks are possible).

If you use a VPN server on your Raspberry Pi, data flows in the secure tunnel, so nobody can decrypt them.
The goal of this tutorial is to create a secure tunnel between you (from anywhere in the world) and your local network at home



How it works

I won't bother you with details concerning data encryption technology, but here is what you need to know:

- We need to **install a new software on the client** computer to encrypt data
- On this client, we also **have keys coming from the server** to encrypt data in a way that only the VPN server can understand
- In the client configuration, we'll **tell the software to connect to the VPN** server public IP address
- When the encrypted data arrives at the VPN server, the server software will decrypt it and know what to do with it
- The same thing applies for packets coming from the home network to the VPN client

So, we don't need to do too many things, just install software on each side of the secure tunnel.

OpenVPN introduction

OpenVPN is the free software that we'll use to do this.
It provides client and server parts for any operating system.
More precisely, we need to install:

- **OpenVPN server** on our Raspberry Pi at home
- **OpenVPN client** on our laptop computer or smartphone, to access home resources from anywhere

How to install OpenVPN on your Raspberry Pi?

You now understand how it works and what we need to do.
Let's get to the technical part!

Raspberry Pi configuration

Here is what you need to start this guide:

- A Raspberry Pi (tested on Zero, so any model should work)
- Raspberry Pi OS installed (refer to Day 4 if needed, you can continue with your current installation if you followed my tutorial about file server and/or Pi-Hole)
- Administrator access to your Internet router or firewall (for port forwarding)
- A static public IP address if possible or a dynamic host.
- I don't have a static IP, so I'm using No-IP: **https://www.noip.com/**
  If you are in the same case, you can check my tutorial about No-IP here: **https://raspberrytips.com/noip**

Installation

Once everything is ready you can follow this procedure to install OpenVPN:

- We'll use the installation script from angristan on GitHub.
  You can check the code on this link before installing it if you want: **https://github.com/angristan/openvpn-install**
- Copy and paste this command in your terminal to download it:

  ```
  wget https://raw.githubusercontent.com/angristan/openvpn-install/master/openvpn-install.sh -O openvpn-install.sh
  ```

- Then run the script with:

  ```
  sudo bash openvpn-install.sh
  ```

- The script will show you your local IP address (you'll need it in the next step), your public IP address and ask you a few questions:

```
Do you want to enable IPv6 support (NAT)? [y/n]: n

What port do you want OpenVPN to listen to?
   1) Default: 1194
   2) Custom
   3) Random [49152-65535]
Port choice [1-3]: 1

What protocol do you want OpenVPN to use?
UDP is faster. Unless it is not available, you shouldn't use TCP.
   1) UDP
   2) TCP
Protocol [1-2]: 1

What DNS resolvers do you want to use with the VPN?
   1) Current system resolvers (from /etc/resolv.conf)
   2) Self-hosted DNS Resolver (Unbound)
   3) Cloudflare (Anycast: worldwide)
   4) Quad9 (Anycast: worldwide)
   5) Quad9 uncensored (Anycast: worldwide)
   6) FDN (France)
   7) DNS.WATCH (Germany)
   8) OpenDNS (Anycast: worldwide)
   9) Google (Anycast: worldwide)
   10) Yandex Basic (Russia)
   11) AdGuard DNS (Anycast: worldwide)
   12) NextDNS (Anycast: worldwide)
   13) Custom
DNS [1-12]: 11
```

- Most of the time, you'll keep the default values, so just press "Enter" for each question if you don't know.
  The OpenVPN installation starts.
- A few seconds later, you will be asked some information about the first user to create. Give it a name (your name or the device you'll connect with for example), and set a password if needed (not mandatory).
- Once done, the scripts ends and give you the path to the first configuration file:

```
The configuration file has been written to /home/pi/android.ovpn.
Download the .ovpn file and import it in your OpenVPN client.
```

That's it, the VPN server is running and your first client is ready to use.

I recommend restarting the Raspberry Pi after the installation. I don't know why, but for me, it was not working before the reboot, even after starting the service manually.

Port forwarding

The OpenVPN server is listening on the port 1194.
To access it from another location, you need to enable IP forwarding in your Internet router configuration.
That's to say, redirect <your_public_ip>:1194 to <your_raspberry_ip>:1194

I can't provide any more help with this because it all depends on your router software.
You'll often find a NAT configuration page in the advanced options.
Ask your Internet provider's support team if you don't know how to do this.

Client installation

OpenVPN is available for all devices with any operating system, even smartphones.
Don't forget, you need to be out of the local network to test the connection.
Use a mobile connection while testing.

- **For Windows**, you can download the OpenVPN software for free here:
  https://raspberrytips.com/dl/ovpn
  Import the configuration and connect

- **For Linux users**, OpenVPN is generally available in your distribution repository.
  Use the graphical tool or the command line to start the connection
  For example:

  ```
  sudo openvpn client.ovpn
  ```

- **And for Android / iOS**, you'll find it in the corresponding app store.
  You can send the configuration to your phone by email and download it.
  Then, import it from your downloads folder.

If you have any difficulties, you can find more help on the official website:
https://raspberrytips.com/go/ovpn

Congratulations if you made everything work as expected!

Homework

Wow, what a big chapter!
And this is only the first one like this, as I'll try to show you many things you can do with your Pi.
You will start to see how the basics you learned at the beginning are useful in real life situations.

As usual, take your time, try all projects and take notes if you need to.
The goal now is no longer to remember everything, but to learn the way, so you can be at ease with any other project you'll try in the future.
You may also have learned new global concepts, like how a VPN or a file server really works.
Depending on your skill level, you may need more or less time here.

In the next chapter, I'll show you three other projects.
They can be easier for beginners, but don't rush this part.
Enjoy the process :)

Day 16:
3 projects ideas
to use your Raspberry Pi
at home

Introduction

This chapter is similar to the previous one: 3 projects to try to improve your knowledge and skills.
The goal here is to learn how to use your Raspberry Pi at home to make your life easier, more convenient, and often cheaper.

I'll show you 3 projects I like that typical examples of what you can do at home:

- How to plug your Raspberry Pi into your TV and use it as a media center
- How you can protect your house by turning your Pi into an IP camera
- Finally, how you can use it as a smart stereo system, and play Spotify on it, with your smartphone as a remote control.

Project 1: Your Raspberry Pi as a media center (Kodi/OSMC)

The best qualities of the Raspberry are its size, its low-energy consumption and its absolutely silent operation.
What better list of things do we need to build a media center?

Yes, that's it. The Raspberry Pi is the perfect device to turn your old TV into a smart TV or to upgrade your current smart TV with the latest apps for a media center.
You can also travel with it, and then use it at home, at your parents' home or while traveling, with the same content, whatever TV you find on your way.

To do this, you'll need only two things:

- A new operating system, dedicated to this goal: OSMC
- Good software to use on your TV: Kodi

I will show you how to install both in this part, let's go!

Note: At the time of my last edit, OSMC was not yet supported on Raspberry Pi 5. Before following this project, make sure it's been updated and is now supported. You can also check out LibreElec (an alternative distribution), which generally updates faster and works almost the same once installed.

OSMC Installation

What is OSMC?

As you know, the Raspberry Pi runs mainly Linux distributions.
Linux is an operating system (like Windows or Mac OS) which exists in several variants (called distributions).

To create our media center, we'll use OSMC as our Linux distribution for this project.
OSMC is a free and open-source distribution, dedicated to media center hosting. It's a lightweight distribution, making it easy to install, which will do the job perfectly.

Download

All downloads for the OSMC distribution are available on this page:
https://osmc.tv/download/

OSMC offers you two ways to create an SD card:

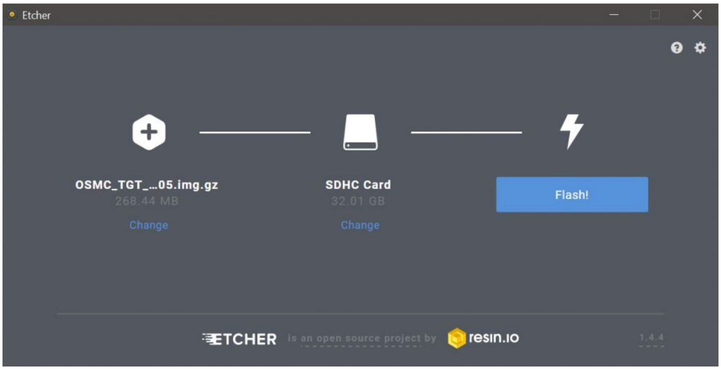
- With an installer; **download the version corresponding to your computer system**
- **By downloading an image**; download the version corresponding to your Raspberry Pi version

As you already have Etcher on your computer, and already know how to use it, we'll use this method.
But if you don't have DHCP on your network, the installer will help you to configure that.

Note: OSMC is one option, but you can use LibreElec instead. The end of this tutorial will be almost the same. Read the article to learn more about these two options: **https://raspberrytips.com/osmc-vs-libreelec/**

Flash the SD card

- **Download the OSMC image** corresponding to your device (click on the "Disk Images" button under Windows and macOS)
- **Start Etcher**
- As usual, load the image file with the left button and select your SD card:



- **Click on the Flash button as soon as you are ready**
  The SD card creation was relatively fast for me, as the image takes up less than 300 MB

At the end of this procedure, you're ready to start OSMC for the first time.

First start

Insert the SD card into your Raspberry Pi and start it.
I'll explain to you what happens in the next steps.

Splash screen

This first step is automatic. You don't have to do anything.
OSMC will carry out the required tasks to make your system work (resizing the SD card, Installing files, etc.)

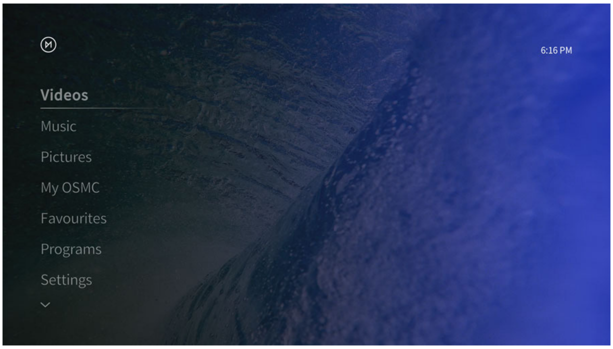
At the end of the progress bar, you get a success message and the Raspberry Pi reboots.

Welcome menu

After a brief preview of the Kodi menu, the Welcome menu appears.
In this menu, you'll need to enter your preferences:

- Language
  - Select your preferred language for the Kodi interface
- Timezone
  - Select the city corresponding to your current time zone
  - I don't find this menu very intuitive
  - On my second try, I used the touch pad from my Bluetooth keyboard, and it became easier
- Hostname
  - You can leave this by default or choose a custom name
- SSH
  - I recommend you enable this directly
  - It could be easier for some things we'll see later on
  - If you don't use it after this, you can disable it later
- License
  - Accept the license and continue
- Look
  - The welcome menu gives you a choice between two themes
  - Take the one you prefer but you can change this later
- Sign Up
  - This is not mandatory, skip it

Once you complete this menu, the Kodi interface comes back:



I'll call this Kodi, even if I know that the line is blurred between OSMC and Kodi. You'll find OSMC and Kodi stuff in this menu, which is not important as long as we understand each one.

Other things to know about OSMC

SSH

If you enabled the SSH service in the welcome menu, it's possible to connect OSMC via SSH remotely.
You can connect the same way we did previously:

- Use putty (Windows) or the ssh command (Linux/macOS)
- Enter the IP address
- Log in with the default user credentials:
  - **Login**: osmc
  - **Password**: osmc
- Once connected, I recommend you change this default password with the command:

```
passwd
```

You are now in the OSMC system, basically a Debian-like system, where you can use your Debian skills to do what you need to do :)

SFTP

It comes with SSH, but many people don't think about this.
As you have an SSH connection to your Raspberry Pi, you can use software like WinSCP, FileZilla or rsync/scp to transfer a file to and from it.

You can even open a port in your router (NAT) to transfer files remotely from another network on the Internet.

Kodi Usage

What is Kodi?

People are often confused about the difference between OSMC and Kodi.
That's why I'm making this short introduction.

Kodi is the software we'll use to browse through our media center (like Spotify for music).
OSMC is the operating system running on the Raspberry Pi (like Windows on a PC).

Kodi allows you to play videos, music and other digital media files from local and network storage or the internet.

We'll use Kodi over OSMC because the main goal of this project is to have a dedicated Raspberry Pi for our media center.
But if you already have a Raspberry Pi on Raspberry Pi OS and want to try Kodi, it's possible.
Kodi is now available directly in the Raspberry Pi OS repository.

Kodi basics: add media files

For the moment, media entries in the menu are empty (Videos, Music, and Pictures).
I will now explain to you how to add your files to it, and how to test the player. There are several ways.

USB Drive
The most natural way to read a file in Kodi is to use a USB key or external drive (especially if you have all your photos or movies on it).

In this case, plug the drive into the Raspberry Pi and all your files will be available directly.
OSMC and Kodi will automount your disk and display all available files.

For example, if I go to Music > Files, I see my USB key name on the list.
Click on it and you will get your files:

Everything is synchronized with the web interface so, if you prefer, you can use the web interface to find and start your files on the TV.
The USB drive will be found in the "Browser" menu in the web interface.

Copy files into the Raspberry Pi

You can't directly save your files from the USB drive to your SD card from the file menus.

You can't directly save your files from the USB drive to your SD card from the file menus.

To do this, you need to go to the Settings Menu > File Manager.
Then find your file in the left panel and the destination folder on the right panel.
Finally, right-click on the file and choose "Copy" to save it in your Raspberry Pi.

There are other interesting options here, but I'll let you discover them for yourself.

Network share

If you have a share available on your network (like a NAS or a shared folder on another PC), you could access it from Kodi, to play media files over the network.

To do this go to Videos > Add videos.
Enter your media source URL and choose a name.

Kodi can manage many different servers (smb, nfs, http etc.)
You'll find help about the syntax to use in each case on the Kodi wiki:
https://raspberrytips.com/go/kodiservers

Once your media source properly added, you can access all media on this server directly from the Kodi interface (or the web interface, I'll show you how later).

Install add-ons on Kodi

Ok. You now know how to read media files on your Raspberry Pi with Kodi.
But it's not that different to what you could do on Raspberry Pi OS or your game console.

One of the powerful features of Kodi is the possibility to install add-ons.
An addon is similar to an app on your smartphone. A smartphone without an app store is not so fun.

There are also many add-ons in Kodi. I will introduce you some of them here.
The problem is that you will find anything and everything on the Internet.
I will classify those that I present to you by legality but beware of what you will find elsewhere!

Official Kodi add-ons

In this first part, I'll only show you add-ons from the official repository.
So, everything here is legal, don't worry.

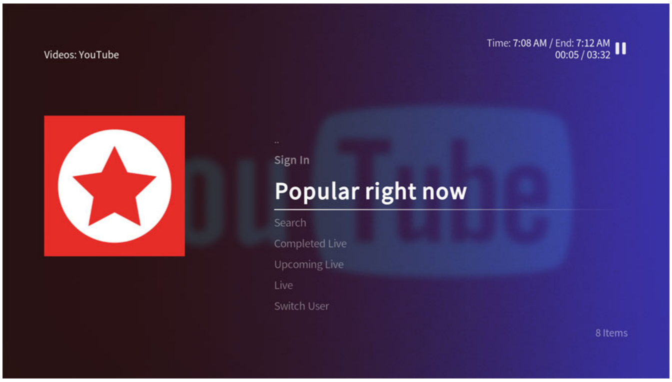
YouTube

The first essential add-on is Youtube.
To install it, follow these steps:

- Go in **Settings > Add-on browser**
- Then **"Install from repository"**
- Select **Video Add-ons**
- **Find YouTube** on the list (Yes, at the bottom)
- Click on it and then choose **"Install"**

The add-on will start downloading and will be installed quickly.
To use it, go back to the main menu and then go to "Videos"
Choose "Video add-ons" and then YouTube.

On the first start, you may be asked some configuration questions like language and country.
The interface is not extraordinary but will allow you to find and watch your favourite YouTube videos.



Soundcloud

Another example is Soundcloud, which you can get the same way as YouTube.

- Go to **Settings > Add-ons browser > Install from repository > Music add-ons**
- **Find and install Soundcloud**
  Go back into your main menu > Music > Music add-ons, you'll see the Soundcloud app.

Other default apps

You should now understand that any add-on you find in the main repository is easy to install and use and that you can use it completely legally.
So, I will let you browse this list and install any add-ons you are interested in before moving on to the next paragraph.

If you need more advice about VPN and add-ons to avoid, check the end of the corresponding post on RaspberryTips: **https://raspberrytips.com/go/kodiaddons**

Web Interface

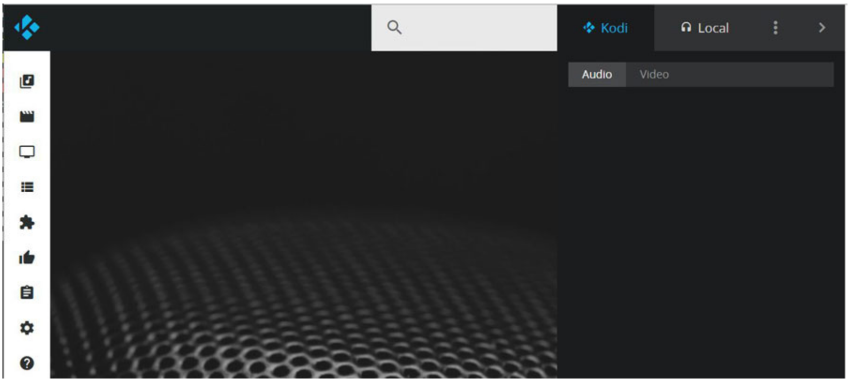
You may have missed it, but straight after the installation, a web interface for Kodi becomes available.
You can access it by entering your Raspberry Pi IP Address: http://X.X.X.X

This interface is responsive so that you can use it from your tablet or smartphone.

Before this, you need to know your IP address.
I'll give you four ways:

- If you have set your IP address in the OSMC Wizard:
  - You already know your IP address; it'll be the same as you entered in the wizard
- If you are in the OSMC menu, it's easy:
  - Go to **"My OSMC"**
  - Then **"Network"**
  - Then choose your network connection (Wired or Wireless)
  - All of your network settings will be displayed here, including your IP address

- If you are on another computer on the network, are there are two other ways:
  - Go to your router web interface, most of the time it'll give you all connected devices with their IP address
  - Follow my article about how to find your Raspberry Pi IP address - use the network scanner: **https://raspberrytips.com/go/ipscan**



Project 2: Use the Pi with a camera

The second project you can try at home is to use a Raspberry Pi to host an IP camera.
The base is for a security camera, but it doesn't change too much if you prefer to use it as a baby monitor, a live camera on the web or to create exterior time lapses.

To do one of these project ideas, you need to:

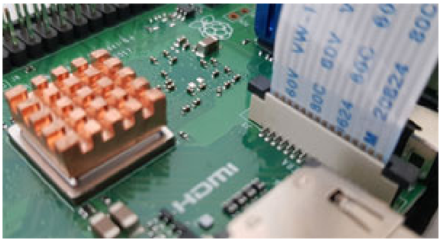
- Have a Raspberry Pi and a camera for it
  I hope you are not on chapter 16 without a Raspberry Pi, but if you need help to choose your camera, check my recommended products page here:
  https://raspberrytips.com/resources/intermediate/
- Install Raspberry Pi OS (see chapter 4 if you still need a guide)
  Raspberry Pi OS Lite is OK for this project.
  Raspberry Pi OS Desktop can help to check that everything is configured correctly, but you can do this remotely
- Plug the camera in
- Know the basic commands for using it (photo, video, etc.)
- Install software to get a web interface and notifications

I will now give you the step-by-step process to do this.

Plug and install the camera on your Raspberry Pi

The camera installation on the Raspberry Pi is straightforward once you find the port location:

- Take the Raspberry Pi out of its case
- Find the camera port on the Raspberry Pi (between the HDMI and jack port) It's the only one that fits the cable width, and it may be written as "CAMERA" on the main board; you'll find it easily.
- Before plugging the cable in, you may need to remove the plastic film and lightly pull the black plastic
- Plug the cable in and push the black plastic to hold the cable inside
- Make sure to align both connectors on the same side (cable connectors on the HDMI port



- Then I recommend starting with a rapid test before holding everything in the case and/or in the holder, just to be sure that it's working

Note: On Raspberry Pi 5, the camera installation is slightly different. It's a different cable (the same as on Raspberry Pi Zero) and must be connected to one of the two connectors near the Ethernet port. Since the port is fixed in the other direction, the pins on the cable must also face the Ethernet port, not the HDMI ports.

Plug all cables back in (power, HDMI, network, USB ...) and start the Raspberry Pi.

If you're using an outdated version of Raspberry Pi OS (Buster and previous ones), you may also need to enable the camera in raspi-config and reboot the system. This is no longer needed in recent releases.

Commands to test the camera

I already gave you most of them in chapter 11, but I will add more details on how to use them here.

> **Note**: *The camera commands are different on Raspberry Pi OS Legacy, you'll use "raspistill" instead of "libcamera-still" for example. But the idea is similar..*

Take a photo

The first thing you can try is to take a simple picture of the image seen by the camera.
"*libcamera-still*" is the corresponding command on Raspberry Pi OS.
It's already installed on your system.

To use it, the basic command line is:

```
libcamera-still -o image.jpg
```

With -o you define the target file name (where the pictures will be saved).
It's possible to use a file name including the path, for example:

```
libcamera-still -o ~/Pictures/mypicture.jpg
```

By using *libcamera-still -h* you'll get the command help screen, with all possible options.

Record a video

To record a video, the command is almost the same.
"*libcamera-vid*" is the corresponding command name.

So, similar to taking pictures, to record a video use:

```
libcamera-vid -o video.h264
```

Use the CTRL+C shortcut to stop the recording.
.h264 files are compatibles with VLC.
On Raspberry Pi OS Desktop: *sudo apt install vlc*

It's the same procedure as before to see all available parameters;
use libcamera-vid -h to get all options with a short description.

Useful options are:

- –t: to choose the video duration in ms (ex: 6000 for a 6s video). This way you don't need to use CTRL+C and can schedule the video capture with a script or cron
- -w and -h: video size (width and height)

If you are on Raspberry Pi OS Lite, you can use a remote access to check if the files are OK.
We have already seen many solutions in the previous chapters: SFTP is probably the easiest one to use, but you can also install a remote desktop or anything else to access the files from your computer.

And obviously, you can use these commands in a script, for example if you want to schedule the shot.
Python also provides a library to do advanced scripts with a camera, but that's not the goal today (I'll talk about Python later in this book).

Install and configure motion

The last thing you need to do is build an IP camera or one of the projects I suggested to you in the introduction. You have to install some software: Motion
Motion will give you access to the camera in real time via a web interface, and have many features like movement detection and notifications.

Note: At the time of my last edit, this didn't work with Raspberry Pi OS Bookworm. If you want to play with this, you may need to install a previous version or check if the project has been updated since then.

To use Motion, follow this procedure:

- Install Motion and the required dependencies:

  ```
  sudo apt install motion libavcodec-dev libavformat-dev
  libavutil-dev libjpeg-dev libjpeg62-turbo-dev libpq-dev
  libswresample-dev
  ```

- Answer "Yes" to install them
- Enable the camera driver:

  ```
  sudo modprobe bcm2835-v4l2
  ```

- Use this command to **check that the camera is visible** and displays its settings:

```
v4l2-ctl -V
```

- You will get something like:

```
Format Video Capture:
Width/Height : 1024/768
Pixel Format : 'JPEG'
Field : None
Bytes per Line : 0
Size Image : 786432
Colorspace : JPEG
Transfer Function : Default
YCbCr/HSV Encoding: Default
Quantization : Default
Flags
```

- Then you need to **configure motion**.
- The best way I found is to download the configuration we need:

```
wget https://raspberrytips.com/dl/motionconf -O motion.tar.gz
```

- Extract the files:

```
tar -zxvf motion.tar.gz
```

- **Start motion**:

```
sudo motion -c motion-mmalcam-both.conf
```

- Feel free to adjust settings in this file to fit your camera configuration (image height and width for example)

To finish the Motion installation part, you just need to check the video stream before moving to the computer configuration.

The live stream is available in HTTP, by using the Raspberry Pi IP address and the port 8081.
Open your web browser and go to http://<RASPBERRY_IP>:8081

Check that you can see the live stream with sufficient quality.
If you just want a live camera, you don't have to do anything else.
But if you want to go further (security cam, baby monitor, etc.), you need to adjust the Motion configuration file to fit your needs.

I can't give you a full documentation of the motion configuration file here, but I'll explain to you how to enable email notifications:

- Firstly, your Raspberry Pi must be able to send emails:
  - Install a few packages with apt:

    ```
    sudo apt install ssmtp mailutils mpack
    ```

  - Edit the configuration file:

    ```
    sudo nano /etc/ssmtp/ssmtp.conf
    ```

  - And fill it in with your email provider settings (mailhub, AuthUser, AuthPass, etc.)
- Then, you need to edit the configuration file to add an action on each picture save.
  Play with the *on_picture_save* option to do this.
  You can for example use the mail command, but I recommend using the *mpack* package to add the saved image as an attachment, like this:

  ```
  mpack -s "Alert: Enter your subject here"  %f EMAIL_ADDRESS
  ```

- Restart motion to apply changes

You can find all the resources needed on their GitHub page:
https://raspberrytips.com/go/motionconfig

Project 3: A Hi-fi system, start the music wirelessly

If you like to listen to music at home, you can use your Raspberry Pi as your own personal stereo.
In this part, I'll show you how to play Spotify on Raspberry Pi.

You already know that you can do this on Kodi, as seen in the first project, but here I will talk about a software I like a lot: Volumio.

What is Volumio?

Volumio is a software dedicated to acting as a music player.
You can play any file type with good quality and use the web interface to manage it.

It's the perfect solution for a Raspberry Pi, as you can keep the Raspberry Pi in wireless mode, with a speaker on the jack port.

Then you can use it as your home stereo, by controlling Volumio from your smartphone.

Install Volumio on Raspberry Pi

The installation is straightforward.
In fact, you already know how to do this, as so many projects start the same way.

You need to have an available SD card, then follow this procedure:

- Open Raspberry Pi Imager.
- Click on “Choose OS”, go to “Media Player OS” and select “Volumio”.
- Then, it’s the same steps as usual.
- Click on “Choose Storage” to select your SD card, and click on “Next” to start the installation. You may be asked to use the “OS customisation” options, but I don’t think it works with Volumio anyway.

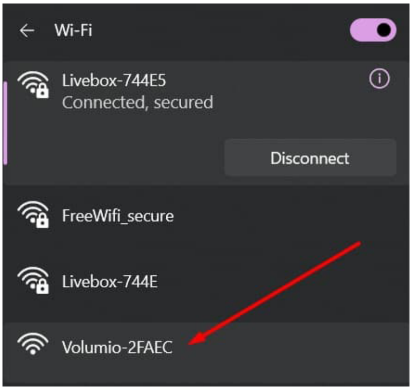
After a few minutes, your SD card is ready to use.
Insert it into your Raspberry Pi and turn it on.

Optional: Wi-Fi configuration

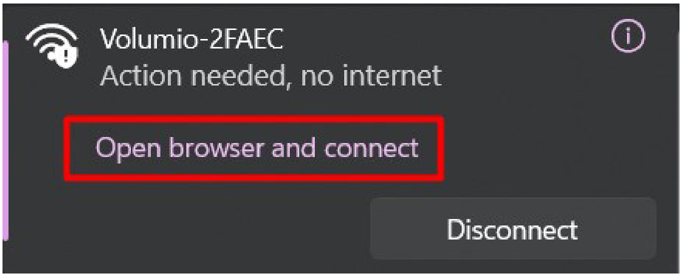
If you don’t have an Ethernet connection for your Raspberry Pi, you need to configure the Wi-Fi before anything else. Everything with Volumio will be done from the web interface, so you should do it on the first boot.

Let the Raspberry Pi boot, and follow these instructions:

- From your computer, connect to the Volumio Wi-Fi network.
- There is no password.



- A web page will open with a configuration wizard.
  If it doesn't open automatically, you may need to click on the link in your network list:

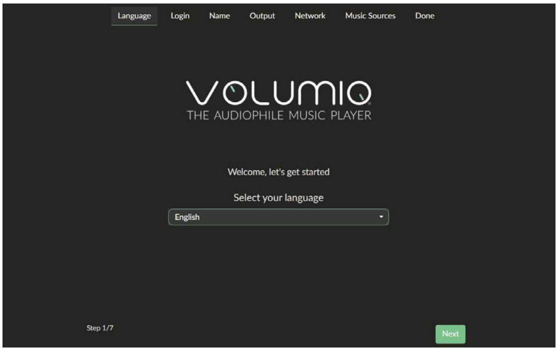


- Follow all the steps and enter your Wi-Fi settings.
  You can keep most of the default values. The most important one is to pick your SSID and type the password. All the other options can be changed later.
- Once finished, Volumio will apply the configuration and restart in normal mode.
  You'll switch back automatically to your usual Internet connection.

Access the web interface

Whatever network connection you use (Wi-Fi or Ethernet), you can now access the web interface to start the configuration:

- **Get the Raspberry Pi IP address:** If you have a monitor on the Raspberry Pi, the IP address is mentioned at the end of the boot, just before the login prompt. If you can't see it, you can use one of the methods explained previously (like the network scanner).
- **Open a web browser, and type the Raspberry Pi IP address.**
  You can use HTTP, so it would be something like http://192.168.1.10.
  Your browser may switch to HTTPS, make sure it's HTTP if it doesn't work.
- You should get something like this, where you will need to answer a few questions:



You don't need to enable anything here. No account is required to get started, I clicked "Next" on each page, keeping all the defaults. Accept the terms of service and you're ready to go.

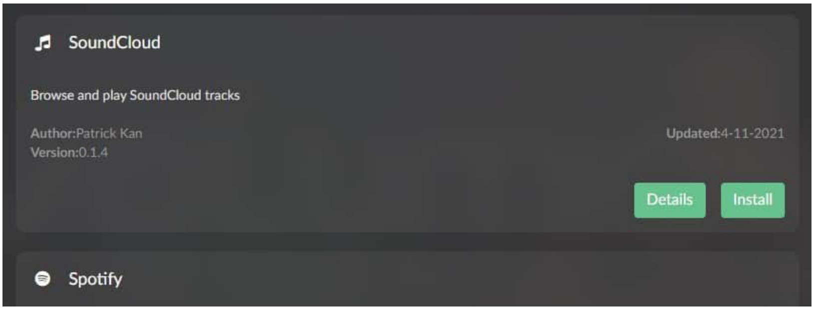
You should get access to the full interface now, with the media player taking most of the space, and a menu available on the left (icon in the top-left corner).

Install & Play Spotify

One nice feature about Volumio is that extensions are supported. Like with Kodi or some TVs, it's possible to add new features by installing plugins from their marketplace. There are not a ton of apps available, but it's still pretty useful.

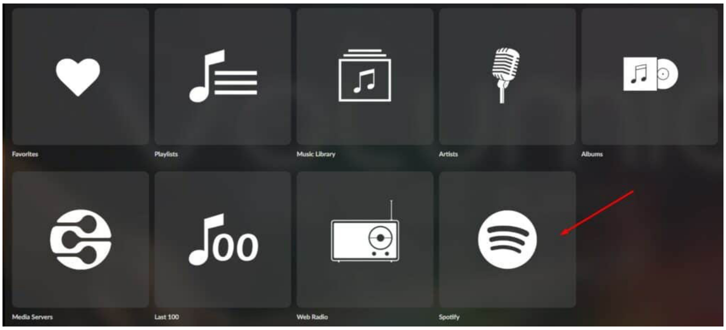
Warning: A free account is required to view and install extensions. If you are not yet connected (or don't have an account), click on "My Volumio" in the left menu, and sign up or log in from there.

You'll then have access to some plugins, like media services (Spotify or SoundCloud for example), that will then be accessible via the Volumio interface:



I tried Spotify, and it installed in one click. Then you need to fill in your Spotify username and password to access the library (Spotify Premium account required).

Once done, you'll get a new icon added in the "Browse" menu:



From there, any song or playlist you liked on your Spotify account is accessible.

Homework

You probably already know what to do :)
I gave you the same homework for a few of the previous chapters.

Try these three projects and play with some more features other than just the ones I have shown you.
The idea is to expand your vision of possibilities on Raspberry Pi.
By trying them, you'll remember them. By just reading, you may forget!

In the next chapter, you'll get some time to relax, but only if you work hard enough on this chapter! We'll start playing games! :)

Day 17:
Gaming on Raspberry Pi with Retropie

Introduction

A Raspberry Pi is often used as a gaming console.
There are many ways to do this, but the most common is by using Retropie.

Retropie is a piece of software that provides the entire retro-gaming solution, that's to say: interface, emulators, games and controllers management, etc.
Retropie is free. It's often installed on a Raspberry Pi like a true operating system, but it can also be installed on Raspberry Pi OS.

Today, I'll show you how to:

- Install and configure Retropie
- Find and install new games
- And also give you a few tips to go further

And I'll start now with some vocabulary and hardware requirements.

Retro Gaming vocabulary

Just before getting started, let's make sure that you understand two words I'll use a lot in the next few paragraphs.

What is an emulator?

An emulator is a piece of software that allows you to run games for another computer on your current computer.
A N64 emulator in Retropie allows you to run N64 games on a Raspberry Pi.

What is a ROM?

A ROM stands for Read-Only Memory. In gaming, this is an image from the original game.
For example, on N64, you played a game like GoldenEye by inserting the cartridge into the console.

On Raspberry Pi, you can't use this cartridge. You need to have a file with the cartridge's content.

What is a BIOS?

A BIOS is the only part of software that you have on a computer before installing an operating system.

On a PC, you may use it to set options like Boot devices order, date/time, hardware preferences etc.
On a game station it shows how the computer should work, so the emulator may need it for launching games.

Recommended hardware

To make the most of your new retro-gaming console, you'll need some hardware improvements.

SD Card

Even though any SD card could do the job, I recommend starting directly on a fast and large SD card.

I recommend fast because, as you'll see, the Raspberry Pi is a cheap console, but not the fastest for this kind of thing.
A fast SD card ensures that this part does not slow your games down.

You need a large one to store all the games you want.
Yes, you can start on a small one, and change later, but you must reinstall everything, backup your save games, etc.

So, the first thing you need to do is find a fast and large SD card.
If you don't have one, check my recommended products page for the best ones you can buy at the moment: **https://raspberrytips.com/go/basics/**

Raspberry Pi case

The second point is the Raspberry Pi case, or at least the cooling system.
Gaming on Raspberry Pi, more so with more recent games, can increase your hardware temperature, and it's not good for the lifespan of the Raspberry Pi.

So, ensure you have a well-ventilated case, heat sinks and maybe a fan too.
I wrote a post about how to install heat sinks you can check out if you need more information: **https://raspberrytips.com/go/heatsink**
But the best solution is the case I use for my Raspberry Pi 4, you can also find it here: **https://raspberrytips.com/go/basics/**

Game controllers

And finally, what is a good retro gaming console without retro controllers?
Even if you could play with your keyboard, it's not the same, the time machine requires the use of an old-fashioned controller :)

There are several good options for this; it's also in my recommended products page: **https://raspberrytips.com/resources/intermediate/**

Installation

Flash your SD card

Retropie is listed in Raspberry Pi Imager, under "Emulation and game OS".
You can use the usual process to install it on your SD card.

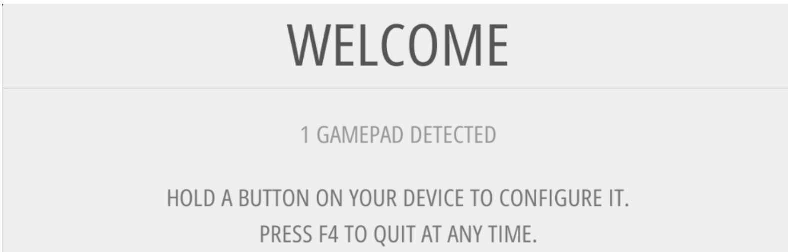
Several versions are listed, so pick the one corresponding to your Raspberry Pi model.

Note: At the time of my last edit, Retropie was not yet supported on Raspberry Pi 5. Before following this project, make sure it's been updated and is now supported. If you can find it in the filtered OS list for Raspberry Pi 5, it means it's not ready yet.

First boot

Insert the SD card and start the Raspberry Pi.
There is almost nothing else to do, so just wait for the boot sequence to finish.

The only thing you need to do is to configure the first controller:

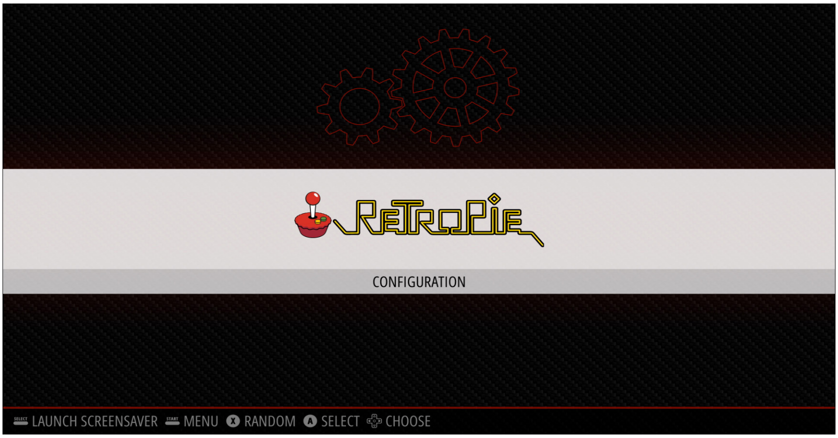


- When RetroPie asks you to plug in your controller, do it
- Press one key for a few seconds until RetroPie detects it
- Then follow the wizard to configure all of the controller keys
- If you have a basic controller, without all possible keys, leave one key pressed for a few seconds until the wizard moves to the next one

- When you have configured all of the keys, press “OK” and the main menu appears

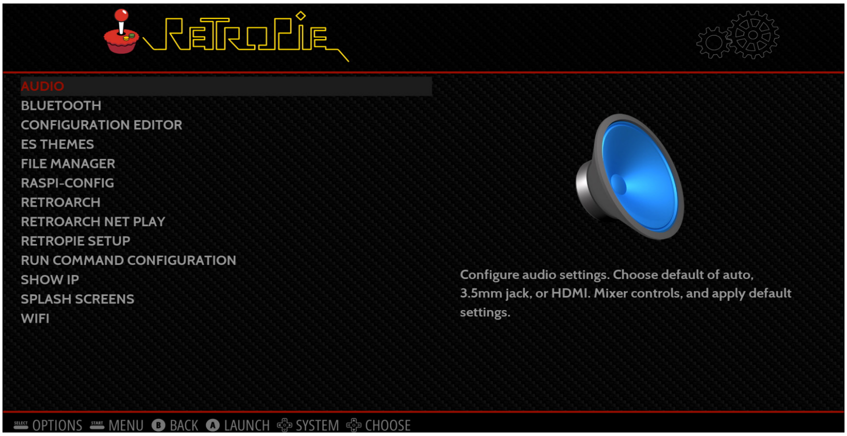
You can now use the main menu to follow the basic configuration procedure in the next paragraph.

Configuration



In the main menu, select “Configuration” and enter (press A).

You should now get a full configuration menu like this:



I'll give you all the things to configure in this menu before going further. Obviously, you can browse all of the submenus to set everything as you want it. But I will only give you the most important things to configure.

Audio

This menu is only used to set the audio output for Retropie.
Set the default sound output as you want it.
On a TV, HDMI should be enough.
On a basic computer screen, you may select jack 3.5 and plug in a speaker or headphones.

Raspi-config

This is the same menu as we have seen on Raspberry Pi OS Lite.

Note

On my installation, I had to switch back to the keyboard to use this menu. I'm not able to make a selection with my gaming controller.

Set what you need in this menu:

- Localization options
    - If needed, start by configuring locale, time zone, keyboard layout and Wi-Fi country
    - Setting the Wi-Fi country is mandatory before the connection (if you need it)

- Network options
    - Useful if you have a specific network configuration to make (Wi-Fi or static IP)
    - For RJ45 with DHCP you have nothing to do
- Update
    - Then select this one to update the system to the latest version

RetroPie setup

This menu will help you to manage RetroPie components.
Try updating the system:

- Updating Retropie
    - Choose “Update” in the first menu
    - Answer “Yes” to all of the following questions
    - This will update everything on your system
    - Then choose “Perform Reboot” to apply all changes
- Come back to this menu after the reboot
- Install Retropie manager
    - In the Retropie Setup menu, select “Manage Packages”
    - Then “Manage experimental packages”
    - Find “Retropie Manager” and install it from source
- Start Retropie manager
    - In the Retropie Setup Menu, choose “Configuration / tools”
    - Select “Retropie Manager”
    - Then select “Enable RetroPie-Manager on boot” and “Start Retropie-Manager now”

Show your IP

You must know your Raspberry PI IP address later in this guide.
So, enter this menu and note your current IP address.

Find ROMs

Retropie comes with no games so you need to download them before playing them.
It's not easy to find sources as a beginner and I also want you to avoid illegal downloads like torrent.
I'll give you a few tips now.

Disclaimer

The video game market is vast and very protected regarding copyrights.
So, playing a licensed video game if you haven't bought it is completely illegal and punishable by law.

We are talking here about ancient games, which have sometimes been abandoned by their publishers.
But that does not change the fact that they are protected, and it's illegal to use them anyway.

So, try to be cautious with the ROMs you find on the Internet.
Being available on a website doesn't mean that it is free of rights.
To avoid any problems, choose free games if possible.
We cannot, in any case, be held responsible for the actions you undertake following the reading of this book; this is your responsibility.

Websites

Free and legal ROMs

If you don't want to take any risks, there's a very famous website that lists free games, usable on RetroPie.
It's called MAMEDev, and the URL is: **http://www.mamedev.org/roms/**

You will find the classics in the history of video games (breakage, parachutes, pong etc.)
Just choose a game and click Download to download the corresponding ROM.

Other websites

Regarding ROM directory on the internet, there are many websites where you can find just about any game from your childhood.

I have gathered 4 websites that I recommend:

- **Rom Hustler**: http://romhustler.net/
- **Free ROMs**: https://www.freeroms.com/
- **Cool ROM**: http://coolrom.com/
- **Coleco Vision Addict**: http://cvaddict.com/list.php

These websites host thousands of ROMs each.
You will inevitably find your happiness!

However, remember what I told you earlier about copyright

Buying pre-loaded SD cards

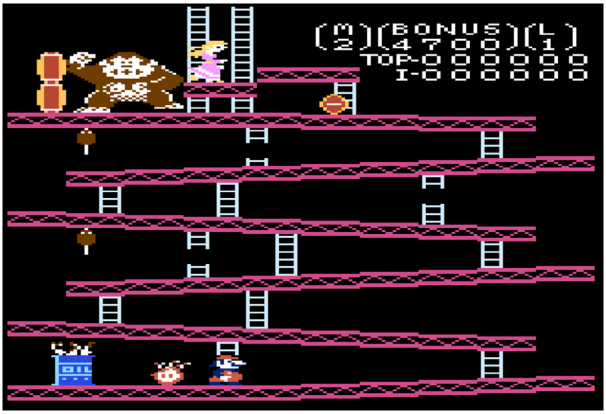
Perhaps the easiest way is to buy a pre-loaded SD card directly with everything installed and all the games you need on it.

As the links may vary, you can find the current best option here:
https://raspberrytips.com/resources/beginner/
A giant 100,000 games SD card with Retropie and Kodi is available for less than the price of a new PS4 game.

Game examples

Here are a few games I like that you can download on Retropie.

Donkey Kong



Let's start with a mythic game from the 80s: Donkey Kong.
For the youngest, you probably know more recent adventures of Donkey Kong, but we are talking here about one of the very first video games.
They released the first version in 1981 for an arcade and after that they applied the game concept to many systems like Atari computers, NES and Game Boy.

Even if we are in a Donkey Kong game, and not in Mario, the concept is similar.
You play as Mario (at the bottom of the screen) and you run to save Princess Peach (top of the screen).
Obviously, you must avoid the traps and enemies on your way.

I put this game in this post for the history, not really for the game play as it's quite basic.
You can play this game for a long time before completing it!

System: Atari 7800
Download Link: https://raspberrytips.com/dl/donkeykong (RomHustler)

Wave Race

Shigeru Miyamoto released this game only 10 years after Donkey Kong on Nintendo 64, and still you can note the big difference in the graphics technology.
A lot of colors (maybe too much ^^), a 3D render and many other details.
It's true that we are not using the same hardware, but it's big progress in the game evolution.

Wave Race 64 is a jet ski racing game.
There are several game modes like Championship, Time trial or Stunt mode.
You can play each mode solo or multiplayer.

I chose this game because it was one of my favorite games at the beginning of the Nintendo 64.
I played this game a lot and I still remember the jingle “Waaaave Race” :)
The best game of Nintendo 64 is obviously Super Mario 64, but it’s not available for download (because of ESA restrictions), so I didn’t include it in this post.

System: Nintendo 64
Download Link: https://raspberrytips.com/dl/waverace (RomHustler)

Street Fighter II Turbo

This one is a classic!
Street Fighter II Turbo was a bestseller in the 90s.
I think it’s the best version from the Street Fighter series.

I present here the SNES version, but you should know that they adapted it after that for many other platforms.
The game was available on PC, Saturn, PlayStation 2, and even on Xbox/Xbox360 and PlayStation 3.

If you really don’t know Street Fighter (I can’t even imagine this is possible), it’s a combat video game.
You compete in 1v1 fights by choosing a character and an opponent.
You can choose the difficulty level or use the multiplayer mode to play with friends.

Thanks to its 2D game environment, it has not aged too much and remains interesting to play today.

System: Super Nintendo
Download Link: https://raspberrytips.com/dl/streetfighter (RomHustler)

You can find more Retropie games ideas on the RaspberryTips website, here: **https://raspberrytips.com/go/games** and my 24 favorites games in one PDF in free download on the same page.

Install a ROM

Now that you've got your ROMs let's see how to transfer them onto Retropie.

Expand partition

If you are planning to download many ROMs, especially the new ones, keep in mind that you will need a lot of disk space.

The problem is that by default, old Retropie versions (when installed as a distribution) creates a partition with the size it needs, regardless of the size of your SD card.
It would be a shame to have only 2 GB available on a 64 GB card :)

Therefore, it will be necessary to extend the partition to the maximum capacity, by following these steps:

- Go to Retropie Configuration
- Choose Raspi Config
- Click on "Expand Filesystem" (first option)
- Confirm your choice
- Wait for the partition to expand
- Then click Finish and reboot your Raspberry Pi

After the reboot, the partition will use the whole size of your SD card.

Transfer files

There are two ways you can transfer games on your Raspberry Pi: USB or SFTP.

USB drive:

You can use a USB drive to copy ROM files to your Raspberry Pi:

- Format a USB drive to a FAT32 file system
- Create a folder named "retropie"
- Plug it in to the Raspberry Pi once and wait for 30 seconds

- Plug it in to your computer again and copy the ROM files in the “retropie/roms” folder
- Wait until USB end starts blinking
- The files were copied, restart Retropie to refresh the list

SFTP:

You can also use SCP to transfer files from your computer to RetroPie.

I already made a post about this, so if you need help with transferring files to your Raspberry Pi, I recommend checking this post:
https://raspberrytips.com/go/ssh

- To enable SSH on RetroPie, go to configuration, then raspi-config.
- Then choose Interface Options, then SSH, and yes.
- Validate and quit.
- Then transfer your files to the folder ~/RetroPie/roms/CONSOLE
- Replace CONSOLE by the name of the console (SNES etc).

Restart Retropie, and it’s done!

Play

After going through all these steps, you will now be able to play your favorite game

Most of the time, adding a new ROM is seen instantly by Retropie, so you have to go to the games list and find your new ROM to play.

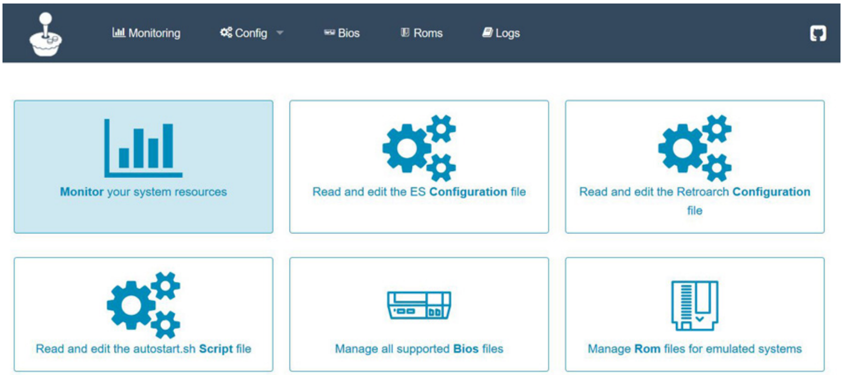
Advanced options

If you want to go further, here are a few useful tips.

Web interface

In a previous part, I asked you to install RetroPie Manager.

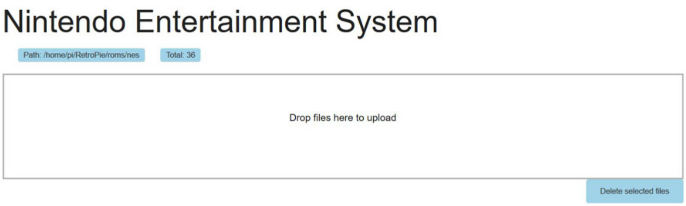
RetroPie Manager is a web interface used to manage your RetroPie system.
You can access it with your web browser at http://<RASPBERRYIP>:8080
If you didn't note your IP address, you can find it under the "Show your IP" menu in the RetroPie configuration screen.



In this interface you can find a lot of information about your system.
The monitoring screen will show you the current status of the Raspberry Pi (memory, file system and CPU).
But you will probably mainly use it to upload a new game on the RetroPie console, here is how to do this:

- Prepare the game files:
- Most of the time, you download a compressed file
- You need to extract it to another folder with your favorite software (for example WinRAR or 7zip on Windows)
- Upload the files:
- Click on "Roms" on the top menu
- Select the system corresponding to the game

- Drag and drop the extracted files into the intended frame:



Wait a few seconds, games are now visible in the list below

PS4 controller

The PS4 controller from Sony (Dualshock 4) is a masterpiece of technology and gaming comfort.
And if you already have one, I imagine you will want to use it on Retropie.
I'll show you how to install it and enjoy the pleasure of retro gaming with a high-tech controller.
The first way is just to plug it with the USB cable, it's plug-and-play and will work directly.
However, **the best way is to use Bluetooth to connect it wirelessly:**

- Start with the Raspberry Pi turned off (it's not mandatory, but it's easier for me to explain)
- You need to have a keyboard plugged in
- Then boot your Raspberry Pi

- When Retropie asks you for the controller configuration, use the keyboard. Choose a key for each action, and remember what you selected for the navigation arrows, the start button and the A button
  - Go to **Retropie configuration**
  - Then **Retropie Setup**
  - Choose the **Configuration / tools** menu
  - You should find a **Bluetooth** section in the next menu, click on it
  - Click on **"Register and Connect to Bluetooth Device"**
  - The wizard turns into "Searching" mode
  - Meanwhile, turn your PS4 controller into "Pairing" mode.
    Press and hold the "Share" and "PS" button.
    The PS4 controller light starts to blink fast
  - After a few seconds, the Bluetooth wizard shows you all available devices.

  Select the PS4 controller in the list (for me it was "Wireless controller")
  - Choose the first option for the security mode (**"DisplayYesNo"** for me)
  - You should get a success message like "Successfully registered and connected to <MACADDRESS>"
  - That's it. Exit the **Retropie Setup menu** and go back to the home menu

- In the home menu, **click on your Start button** to open the main menu
- Scroll down to select **"Configure Input"** and confirm you want to do this
- Then configure the PS4 controller by holding any button for 3 seconds.
  Follow the wizard to **configure each action and submit with "OK"**
  You can now use your PlayStation 4 controller with the wireless mode!

Tips

If you have become a fan of Retropie, there are many other things you can do to enjoy it, like customizing the appearance, using an external hard drive, enabling achievements, etc.

As I try to keep each chapter accessible and not too long, I can't tell you everything here.
But you will find many tips on the RaspberryTips website, this post for example should be a good start: **https://raspberrytips.com/go/retrotips**

Conclusion

I will not give you homework today as you have now got a retro-gaming station on your Raspberry Pi :)
Just enjoy the power of Retropie, and realize what you can do on such a small device.
Download a few games, and if possible play with some friends or family :)

In the next chapter, I'll show you other gaming solutions available on Raspberry Pi.

Day 18: Gaming on Raspberry Pi: Other options

Introduction

We have already seen how to use Retropie on Raspberry Pi, which is probably the most used gaming solution for Pi.
In this chapter, I'll show you three alternatives if you want to try other things:

- RecalBox
- Lakka
- Steam

RecalBox and Lakka are similar to Retropie.
The general look is different and some of the features may vary, but the main system is the same, with the same game files.

Steam is a popular distribution platform on PC that allows you to buy, download and play games with everything automatically managed by the software (updates, add-ons, games library, etc.)
There is a similar way to play games on Raspberry Pi.
The difference with Retropie, RecalBox and Lakka is that this is not a retro-gaming solution, you can play more recent games with it.

RecalBox

RecalBox introduction

What is RecalBox?

RecalBox is a free operating system, dedicated to retro-gaming.
It allows you to play old games on your Raspberry Pi.

RecalBox features

RecalBox supports all the main gaming systems: NES/SNES, Game Boy, PSP, Megadrive, PlayStation, PC, Atari etc.
So, you can play games for all of these platforms in one place; on your Raspberry Pi.
You can even use RecalBox as a media center solution, as it also includes Kodi.

Other features:

- Multi players (5 max)
- Rewind
- Screenshots
- Save state to continue your game later
- Wireless Controllers

But instead of talking about the theory behind it all, let's see how it looks.

RecalBox installation guide

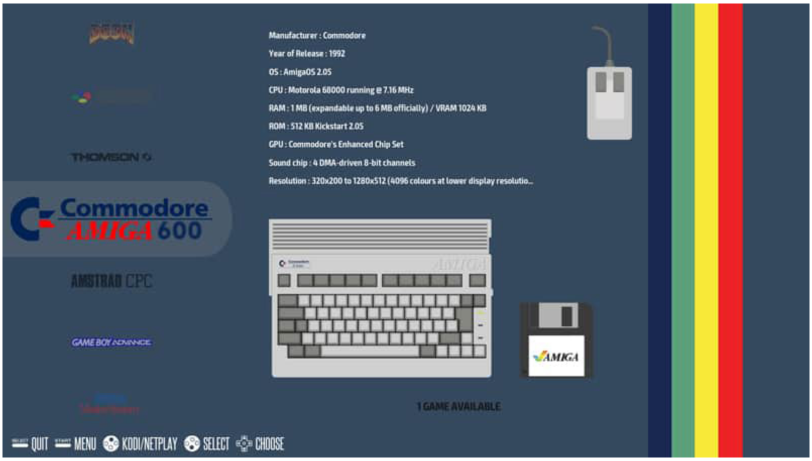
Flash a RecalBox SD card

As with Retropie, Recalbox is now listed under "Emulation and game OS" in Raspberry Pi Imager. So the easiest way to install it on a new SD card is to follow the same process as usual.
Several versions are listed, pick the one corresponding to your Raspberry Pi model.

Like most retro gaming systems, the Raspberry Pi 5 is not yet supported at the time of writing. If you can't find Recalbox, you'll have to be patient or try with an older model.

First boot

- Insert the SD card into the Raspberry Pi slot
- Start the Raspberry Pi
- You'll quickly get the main menu
- It looks like this:



You've finished the installation process, now you can move to the configuration part.

RecalBox configuration

Before playing your first game on RecalBox, there are a few things to configure to complete the system setup.

Controller configuration

You need a USB controller to use RecalBox (for the first controller).
As I wrote in the Retropie chapter, you'll find all of my current recommendations here: **https://raspberrytips.com/resources/intermediate/**

- **Plug your USB Controller** into the Raspberry Pi
- Press any key on it
- A configuration screen will appear, asking you to **hold a button to start the configuration**

- Then, a configuration assistant opens, asking you which key to use for each action
- **Fill in all the key configuration lines you can**
- With a basic controller you may not have everything available
- You can skip some with the down arrow key or holding one button
- Try to keep a button available for the hot key at the end

You can now move to the main menu and use your controller correctly.

Network configuration

The next step you need to take is to check the network configuration.
If you use an RJ45 cable with DHCP, there is probably nothing you need to do.

For wireless usage, you need to follow this procedure:

- Click on the **"Start"** button to open the configuration main menu
- Scroll down to **"Network settings"** and click on it
- A configuration menu like this will appear:

- **Enable the Wi-Fi** like on the screenshot
- Then **select your Wi-Fi SSID** from the list
- Finally, **enter your Wi-Fi password** and close this window to apply
- Come back to it to note your IP Address; you'll need it later

Audio settings

RecalBox is playing music in the menu, and if you want sound in your games, you may need to configure the audio device to use and the volume level.

To do this, open the start menu and choose "Sound Settings"
In the new window you can adjust the volume and force the output device to use.

Other settings

Basically, you'll find all the settings in this start menu.
You can check every submenu to see if you have something to configure.

For example:

- Language and keyboard in System Settings
- Autosave and shaders in Games Settings
- Add a wireless controller in Controllers Settings
- Overscan and over-clocking in Advanced Settings

Update the system

The last thing I recommend you do before playing is to update your system.

- Open the start menu with **"Start"**
- Then go to **"Updates"**
- Click on **"Start Update"**
- Confirm with **"Yes"**
- If a new version is available, it will download and install it now
- At the end of the update, the system will reboot automatically and complete the upgrade

Playing games

RecalBox already includes some free games after the installation.
So, you can play it right away by choosing a game in the main menu.

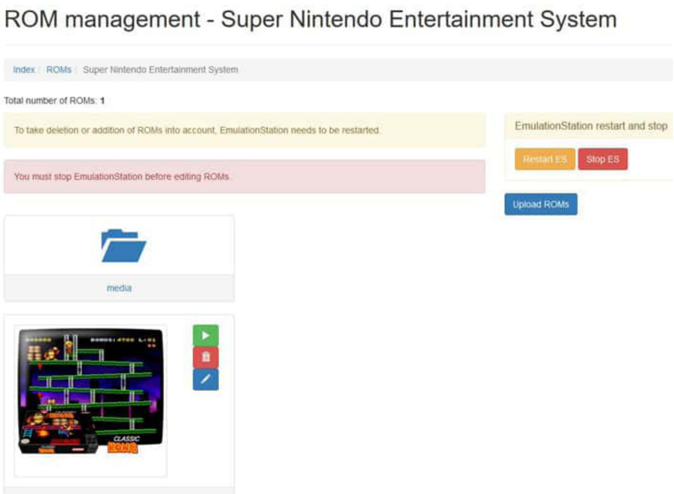
You can also use the same game you downloaded for Retropie, it's the same ROM file.
To install them the procedure is almost the same as with Retropie Manager.

Install games

- From your computer, open the web interface using: **http://IP_ADDRESS**
- If you didn't note it previously, the IP address can be found in the start menu > network settings
- In my case, I use: http://192.168.1.17
- A page with RecalBox Manager opens
- **Click on ROMs** in the left menu
- **Find your favorite system in the list**, for example "Super Nintendo

Entertainment System" and click on it

- You'll get a page like this:



- Click on **Upload ROMs** (blue button)
- Browse to your Downloads folder and upload the ROM you want to play
- The game will appear in the game list
- Finally, you need to Restart EmulationStation (yellow button) to apply changes

Note: you can also use the share at \\IP\share\roms to manage ROM files

Play games

As soon as your interface is back on the Raspberry Pi, you can play your new game:

- From the main menu, scroll to **Super Nintendo**
- **Select "Street Fighter II Turbo"** in the game list and click on it
- The game starts

Lakka

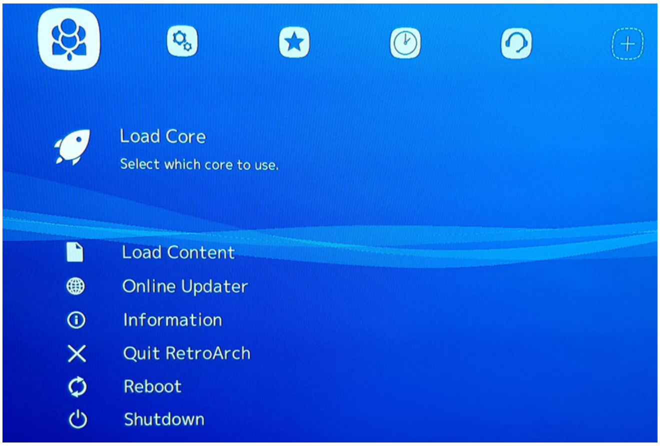
Lakka is the second gaming system I want to show you today.
Introduction

I'll introduce the Lakka system and explain what the differences with other retro gaming systems like Retropie are.

Presentation

Lakka is a Linux distribution created for games emulation.
It will turn your Raspberry Pi into a retro gaming console.

If you have played the PlayStation before, the interface should be familiar to you:



The Lakka system runs on LibreElec and uses RetroArch for the graphical interface.
It's available for many single computer boards (like Raspberry Pi, Orange Pi, Banana Pi or Tinkerboard), but you can also install it on a generic PC.

Lakka strengths

The first of Lakka's strengths is that it starts working directly on the first boot.
As I'll show you in the next sections, once the SD card is ready you can almost play straight away.
You get the main menu and you need no controller configuration; it's working right away!

Also, the configuration is very straightforward.
If you want to enable a share to upload ROMs, there is just one option to change (on/off) and it's ready.
You don't have to think about user configuration, service start/stop etc.

And the last thing I really like about it is that you can play PSP and PSX games immediately.

On Retropie for example, you need to find and install the BIOS to make it work (and it's not easy).
On Lakka, it's working right away, even without BIOS for some games.

Lakka vs Retropie

Lakka differs from Retropie firstly because they use different technologies.

Lakka is using RetroArch over LibreElec.
Retropie is using EmulationStation over Raspberry Pi OS.

Lakka highlights its user-friendly system (ready to play out of the box).
Lakka is also a lighter system, that can work smoothly on any platform, including all Raspberry Pi models.

Retropie has a bigger community, so you'll find help quickly on any issue you have.
Retropie also has a better front-end interface and runs Kodi to use it as a powerful media center too.

To sum it up, both are good emulation systems.
Perhaps Lakka is better for advanced users even if you can start with no configuration.
And maybe Retropie is better for kids that don't know too much about computing.
However, the best answer is to try both and make your choice :)

Installation guide

We can now move on to the installation process.
As I said in the introduction, this is straightforward.

Flash the SD card

Lakka is also listed under “Emulation and game OS” in Raspberry Pi Imager.
Pick the version corresponding to your Raspberry Pi model and flash it using the same process as with other systems.

First start

- Insert the SD card into your Raspberry Pi
- Start the Raspberry Pi
- The Lakka logo appears in the background while the installation process starts
- After a few seconds, the Raspberry Pi reboots and the Lakka main menu appears
- Plug a controller in or a keyboard if you haven’t already
- The system will enable and configure it automatically

You can now move to the menu to discover more of Lakka.
In the next section, I’ll help you with the first configuration steps.

Configuration tutorial

Now that our system is operational, we can customize the default configuration to get it exactly as what we want for our retro gaming station.

Audio, Video and Languages settings

These settings are basic, so I’ll not write a lot about it.
You just need to know they are in the Settings menu, but it’s the first thing to do if you have any issues with this.

In the video settings, you’ll find everything you need to adjust the display to your screen.
In my case everything was fine already, so I didn’t change anything in the menu.
I suppose it’s only useful if you have a small screen (or an old one maybe).

Audio settings are standard.
You can enable or disable audio in the game and in the menu.
Then, you can adjust the volume or change the output device.

Finally, you can change the language in the Settings > User menu.

Wireless configuration

If you need to use your Raspberry Pi on Wi-Fi, here is how to configure it:

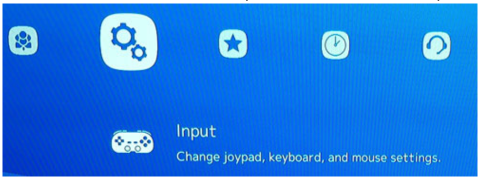
- From the main menu, select the **Settings** icon
- Scroll to the **Wi-Fi** item and click on it
- After a few seconds, the interface will display all of the available wireless networks
- Select the one you want to connect to
- Enter the passphrase
- After a few seconds, Lakka adds "Online" in front of your Wi-Fi network

Adjust the controller settings

Another one of Lakka's strengths is its ability to detect and configure almost any gaming controller automatically.
But you can always check your controller settings or set it differently if you want.

Here is how to do this:

- From the main menu, navigate to the Settings submenu (second icon)
- Select the Input item and click on it (Button B or Enter)



- Now you get a submenu with several options
  - **The Toggle Gamepad Combo**: to get back to the main menu from a game.
    Mine was L3+R3, while I do have an SNES controller without L3/R3
  - **The Swap OK & Cancel buttons**: if you want to revert ok and cancel buttons (A and B on an SNES controller)
  - **Hot keys settings**: there are many hot keys you can configure on your keyboard to do some actions. For example, F8 allows you to take a screenshot in a game

    And finally, the "**Input User X Binds**", where you can configure each key to a specific action.
    This is where you can configure your controller the way you want.

It looks like this:

Set your favorite settings in the User 1 section and move forward.

Enable Samba

A feature I really like in Lakka is the ability to access the file system via a Samba share in one click.
Samba is a file share service that allows you to get access to the Raspberry Pi file system from another computer on the network.

But before you can do this, you need to know the current IP address of your Raspberry Pi.

Get the current IP address

As with many things, finding the IP address on Lakka is easy.

- In the main menu, stay on the first item (with the Lakka berry icon)
- Scroll to the Information item and click on it
- Click on **"Network information"**
- You should have one line for each network interface, along with the IP address.
  I have three lines for example, but you may only have two (RJ45 or Wi-Fi, not both):

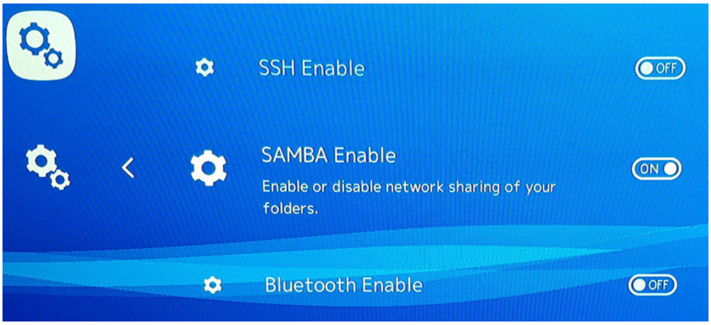
- lo: local interface, we don't need it
- eth0: if you use a network cable, use this IP
- wlan0: if you are using a Wi-Fi connection, use this one

- An IP address has this format: A.B.C.D
  For example, my IP address on the eth0 line is: 192.168.1.15
- You'll need it to access the file share later

Enable Samba

Enabling Samba is easy:

- From the main menu, go to the **Settings** menu
- Scroll down to the **Services** item, and click on it
- In the list, move to the **"SAMBA enable"** line
- Click once to turn it **"ON"** like this:



We'll use it later to upload files to Lakka.
For now, you just need to check that this is working, then move on to the next configuration option.

Customize the menu appearance

By default, the Lakka menu is correct, similar to the PS4 menu and you may enjoy it like this.
But I'll show you how you can configure it however you want in the settings menu:

- From the main menu, go to the **Settings** section (second icon)
- Select **User interface**
- You now have two interesting submenus:
- **Views**: in this part you can select which elements you want to display or hide in the main menu.
  For example, if you are not using the Video feature, you can hide it in the menu

- **Appearance** - here you can configure anything in the main menu:
  - Background
  - Menu layout (for small screens)
  - Menu icon theme (change the icons)
  - Menu shader pipeline (change the background animation)
    Be careful, some of them are very slow on Raspberry Pi; keep the ribbon
  - Menu color theme (choose the main color for your menu)
  - And some other small options to configure shadows and thumbnails

Make all the changes and tests you want, and we can now move on to the most important part: How to play a retro game on Lakka! :)

Playing games with Lakka

Once the basic configuration is done, we are ready to play our first game.
In this section, I'll teach you where to download ROMs and how to install them on Lakka.

Download ROMs

The ROM files you can use on Lakka are the same as on Retropie or RecalBox. I won't repeat how to do this here, so refer to the previous chapter if you need to.

Download at least one game, and follow to the end of this part to play your first game on Lakka!
For myself, I chose "Sonic The Hedgehog 2" on "Sega Master System"

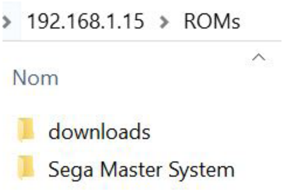
Upload new games to Lakka

In the last part, we enabled the Samba service on Lakka.
We will now use it to upload games to the Raspberry Pi.

That's why I like that the Samba features are enabled in one click: it's simple!

- Go back to your file explorer, as you did for the Samba test (\\<IP> on Windows for example)
- In the folder list go to the ROMs folder
- There is already a "downloads" folder inside
- I recommend creating a new folder for each console, so that you know which emulator you need to use for each game

- If you also chose the Sonic game, create a “Sega Master System” folder, like this:



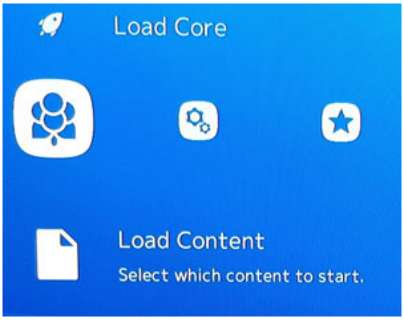
- Then copy and paste your ROM file in the new “Sega Master System” folder

That’s it, you don’t have to do anything else, as the game will now work on Lakka. I’ll show you how to start this game in the next paragraph.

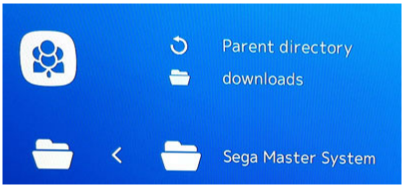
Start and play a game

The game file is now in the shared folder, so it’s on the SD card.
To play your first Lakka game, follow this procedure:

- From the main menu, scroll to the Load content item and click on it:



- You now need to select your ROM location:
- Choose the “Start directory”
- And then select “Sega Master System” in the list:



- Finally, select the rom file:
  It should be something like "Sonic_the_Hedgehog_2_(UE)_[!].sms"
- When you click on it, Lakka will ask you which core system to use.
- There are two available for Sega Master System
- Use the first one.

- Sometimes you'll have the choice between many consoles, so make sure to select the one corresponding to your ROM system
- And eventually, our game will start:

You now know how to download, upload and start any game on Lakka!
You can follow this procedure for each game you want to play.
Have fun and enjoy the ease of use of the Lakka system!

Shaders

Another interesting feature on Lakka is the shaders option.
It allows you to improve the image quality (or at least change it).

To try this, use this short procedure:

- Start your game
- While in the game, use your Toggle Combo to display the Lakka menu
- Scroll down to Shaders (last item)
- Click on Load Shader Preset

- Then choose a GLS shader in the list, for example “bilinear.glslp” near the end
- Return to your game, you should see a change in the game display
- You can try different shaders and download them to find the one you prefer
- This way, you can have a “cartoon aspect” instead of an old pixelated game

Not everybody will like this feature, but I wanted to tell you it’s possible anyway.
Meanwhile, you can note that there are many options available in this “pause menu”
You can:

- Take a screenshot
- Save your game even if the original game doesn’t offer this feature
- Record or stream your game
- Change game and controller options
- Use the rewind feature or change audio/video settings

This is a big improvement compared to other retro gaming systems.
If you want to play the game like it was years ago you can, but you can also use these features to play it differently.

I hope you enjoyed trying Lakka, a lesser known gaming solution, but probably my favorite!
And I have one last option to show you: Steam

Steam

A few years ago, Steam launched a new console: The Steam Link.
In fact, it wasn’t really a games console like the ones we are used to, but a repeater for a computer.
And, for a few months, it’s been possible to install this console software on your Raspberry Pi!

Steam Link History

Let’s start with some background about Steam.

Steam introduction

Steam is a well-known marketplace for games.

Like an app store on mobile, you can browse their games catalog, and then buy and download tons of games.
Now, Steam has over 150 million players and 15,000 games in the store.

Over the years, Steam has brought a lot of technological progress to game distribution.
Created in 2003, Steam arrived in a tough context, where the games were mainly sold in physical boxes.
With the development of the Internet, this concept was adopted by more and more players.

Steam also worked towards multi-platforms adoption in gaming.
In 2010, they worked on a Mac OS version. And in 2013, a Linux version became available.
This was a big step forward for these platforms.
Before that, they were way behind in the world of video games.

Steam consoles

Steam also tried to bring his product to games consoles, with less success for the moment.
They started by adding their software to the PlayStation 3 (for Portal2 for example).

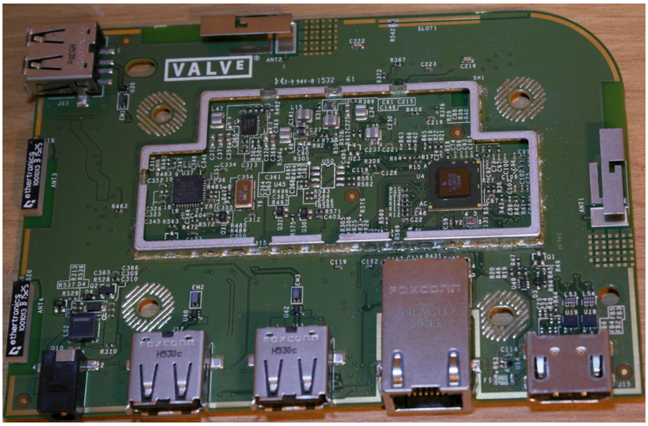
In 2013, things sped up with the release of Steam OS and the beginning of the Steam Machine concept.
Steam OS is a Linux distribution based on Debian, which allows the playing of Steam games on another computer.
And this computer is the Steam Machine, which officially arrived in 2015, and runs Steam OS.

Steam Link announcement

And finally, in 2015, Steam announced a new hardware in its catalog: The Steam Link.
The Steam Link is a small box, with USB, HDMI and RJ45 ports.
It allows us to play our favorite games on the TV, like a console.

But this little box is not really running the games, it's only a screen sharing system. You see the game on the TV or your computer screen, and you can play with other controllers.

There are many commonalities between the Steam Link hardware and the Raspberry Pi.
Here is what the Steam Link looks like inside the box:



Yes, a small, single board running Debian, it reminds us of something...

Recently, the Steam Link software was released in the Raspberry Pi OS repository.
So, we can install it easily and for free on a Raspberry Pi.

That's the goal of this part, so let's move on to the installation process.

How to turn your Raspberry Pi into a Steam Link

Prerequisites

As you probably have a Raspberry Pi already and a gaming controller you don't need anything else on the Raspberry Pi side.
But you will need a Windows computer with Steam installed to try the Steam Link.
Steam is available to download for free here: **https://store.steampowered.com/**

Note: At the time of my last test, Wayland was not supported by Steam Link. This means that it will not work natively on the latest version of Raspberry Pi OS. You may have to use an older version or switch to X11 instead of Wayland to try this project.

Raspberry Pi OS Installation

The first thing you need to do, if you haven't already, is to install the latest version of Raspberry Pi OS on your Raspberry Pi.
Choose the Desktop version and try to use a recent Pi Model (3B+ mini).
Even if we are just in screen share, it's a good idea to start with a more powerful Raspberry Pi.

Refer to "Day 4" if you don't remember how to install Raspberry Pi OS Desktop.

Steam Link installation

Then follow the installation steps below:

- Start a terminal, or connect via SSH
- Update your system

```
sudo apt update
sudo apt upgrade
```

- Install the Steam Link package

```
sudo apt install steamlink
```

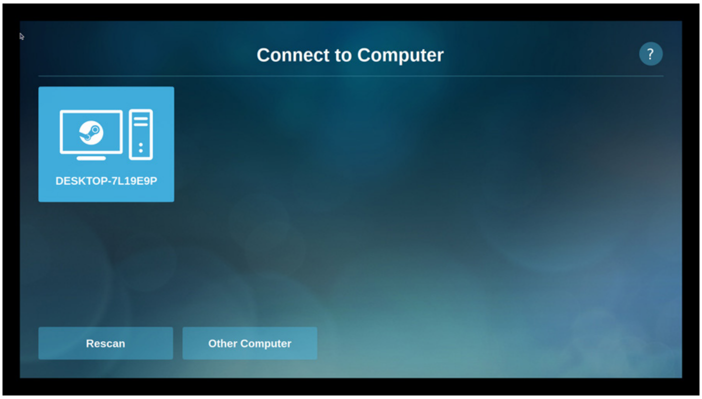
It's now finished in terms of installation.

First start

Then start the application:

- **Open the applications menu**
- Go to Games and click on **"Steam Link"**
- The app opens, click on **"Get started"**
- If asked, **pair your controller**.
  It's mainly if you use the Steam controller.
  For USB controllers, just plug them in, and Steam will detect them automatically.

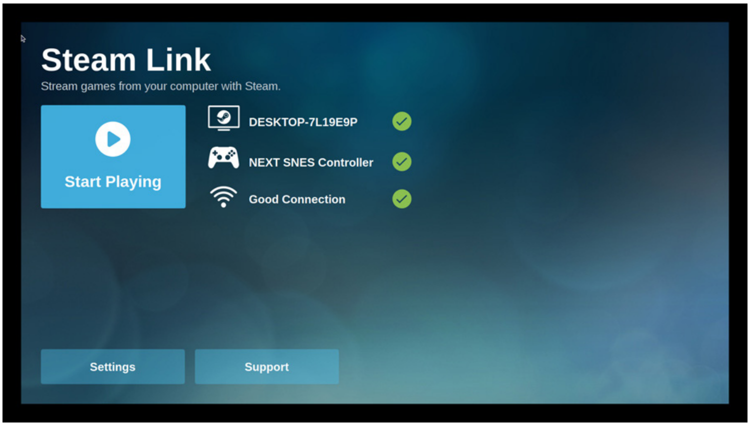
- Then, **select the computer** you want to use from the list:



- Finally, **enter the pin on your computer's screen** to pair the two devices. Steam starts some network tests, and after they are complete you are ready to play!

First game

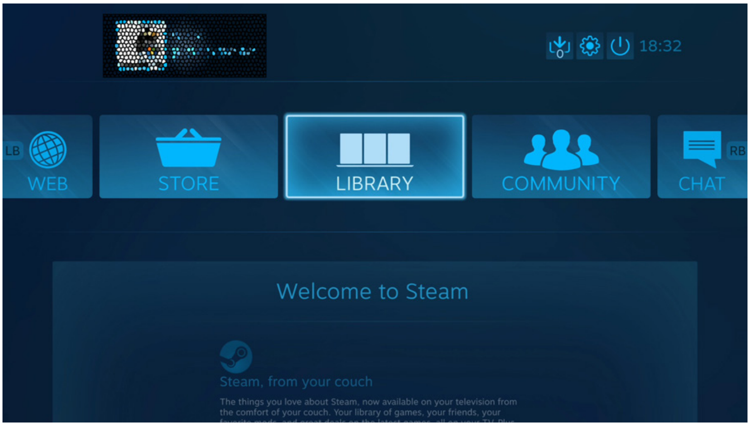
Here is the screen you should get:



Now click on **"Start playing"** and follow these steps:

- The first time you click on "Start Playing", Steam can ask you to install additional drivers on the main computer.
  In my case it was only an audio driver (to send the computer sound to the Raspberry Pi, I think).
  Click on **"Install"** on the computer and it's done

- Then the Steam main screen appears.
- It’s almost the same menu as on your PC:



- If you already have some games, **go into “Library”** and select one game
- On the Game page, you have two main parts:
- **Your game:**
  - This is where you can start the game.
    You can also get other information like your statistics, friends playing this game and recent news
- **Manage game:**
  - Here you have all the configurations and preferences for the game.
    I recommend you at least check the controller configuration, as the game controls might not work with your retro controller
    Or perhaps they will need a keyboard key which is not convenient for gaming on Raspberry Pi
- Then, in “Your game” click on “Launch game”
  The first launch can take a few minutes to configure the Steam Link correctly. Be patient!

- Finally, you are ready to play your favorite game:

Extra tip

I don't know if it will be of interest to you, but I found something funny.
If you force quit Steam on the computer (ALT+F4), the screen mirroring stays active.

You gain access to your computer on the TV.
This allows you to watch movies, pictures, documents or websites on your TV.
If you don't have a smart TV yet, this could be useful to bring the power of Windows apps on the Raspberry Pi.

Conclusion

As for the last chapter, I won't give you any homework today.
You have 3 gaming systems you can try and play with, so you'll probably not be very "productive" right now :)

Play as many games as you want, choose your favorite system, and keep the SD card or the game files safe if you want to play later.

Tomorrow is another day; playtime will be over. Do you like snakes? :)

Day 19: Programming on Raspberry Pi: Python basics

Introduction

This chapter is a difficult one to write for me.
As I told you I'm a developer, so I can learn any language quickly, that is not the problem.
The issue is that I have no idea about your level of skill or experience.

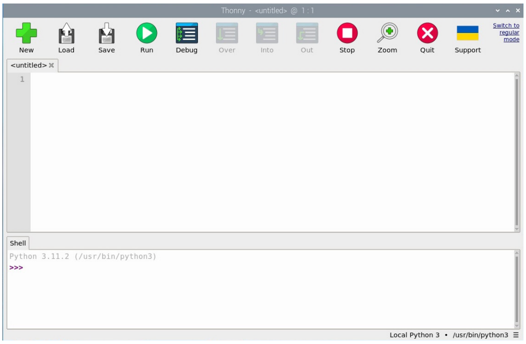
Before starting our first programming language, we often spend hours learning the code structure and the logic.
But it's not a book about programming, so I have to go faster.
The good news if you are a beginner is that Python is one of the easiest languages to understand, as it's actually quite close to regular English.

So, in this chapter I will introduce programming with Python on a Raspberry Pi.
I will give you a little background about Python, show you the basic instructions you can use for your first scripts, and end with an example.

Why Python?

In addition to being an easy language to learn, Python is the main Language on Raspberry Pi (hence the device name).
It's preinstalled on any Raspberry Pi OS system, and you even have software available by default on Raspberry Pi OS Desktop.

For example, you can use Thonny, or "Thonny Python IDE" for its full name.
It's available in the Main Menu > Programming and it looks like this:



In the top bar you'll find all the useful shortcuts you may need (save, run, stop, etc.)
Just below, you have the main part, where you'll type the Python source code.
And finally there is the Python shell, where you'll see what happens when you run your script.

Take a few minutes to install Raspberry Pi OS Desktop if needed (refer to chapter 4).
Then, open Thonny, we'll use it later for the Python basics on Raspberry Pi.
For now, I need to give you more information about Python in general.

Python introduction

What is Python?

Python is a high-level language to program all kinds of software (like C, C++ or other languages).
It's similar to a bash script because you can run it directly.
Most of the Python source code has a .py filename extension and you can run it using "python file.py"

Pythons users in the world

The conception of the Python language started in the 80s and it's still an active language, with a big community of users and developers .
Many major companies are using Python in their apps, like Google, Facebook, Netflix and Spotify.
So, it's definitely not an outdated language, learning it now could help you with your career or future projects.

Philosophy

Tim Peters wrote the Python philosophy in twenty aphorisms like these:

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.

These give you an idea of what to expect when using Python.
One of the main goals is to keep it easy to read.

Code structure

Even if you can build complex projects in Python, the basic structure is simple. Most of the instructions are easy to read, and even a beginner in Python can understand what the code is doing with a minimum of programming skills.

I will not give you more details now, but you'll see in the next part a few examples to illustrate this.

Basic instructions

The goal here is just to give you an introduction to Python.
There will be other chapters later in this book that will give you more advanced instructions and examples.

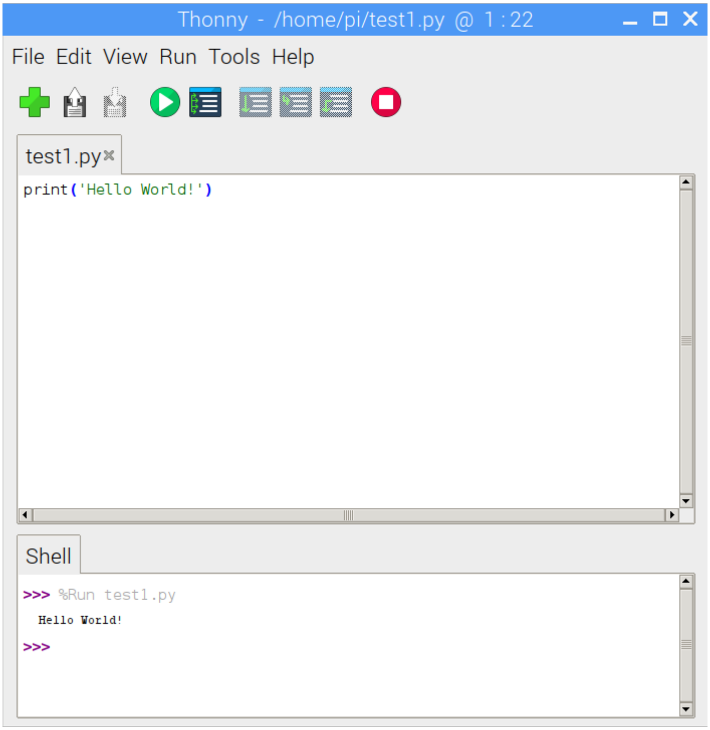
Hello World

It's a tradition when you study a new programming language, so let's start with it.
How to display the famous "Hello World!" in Python!

It's really that simple:

```
print('Hello World!')
```

In the Thonny IDE, you should get something like this:



Variable notion

In any language, variables are used to store values locally and temporarily.

For example, you could ask for a username at the beginning of your script and display it later in your code.
To do this, you need to define a variable and use it to store the username in it.

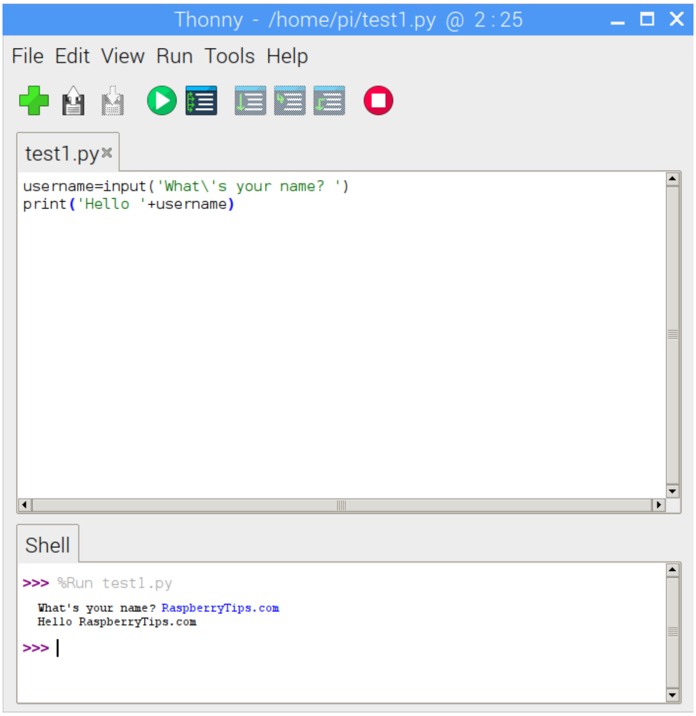
In Python here's how this example looks:

```
username=’Raspberry’
print(‘Hello ‘+username)
```

On the first line, we set the username variable with the 'Raspberry' string.
In most languages, you first need to create a variable, setting its type (string, integer, array), before using it in the code.
In Python, there is no command to declare a variable. You can use it directly.

On the second line, we use the same ***print*** function as seen before.
I introduce a new syntax element with the '+', used to concatenate strings.
When we ask Python to display 'Hello '+username, it will display 'Hello Raspberry' instead.
The variable value is used as another string to display, attached to the first part.

If you are following me, you should have noticed that this is not exactly the example I gave :)
Here is how to ask for a username and display it dynamically depending on the user's response:



The input function asks the user for an input value and sets it as the username variable.
Then you can display the variable as in the previous example.

We use the backslash to tell Python that the following apostrophe is not to close the string.

You could also use double quotes to avoid this:

```
input("What's your name?")
```

Conditions

Theory

The next step is to learn what conditions are and how to use them in Python.
When you create a new program, you'll often need to do something different depending on a value:

- If the input string is empty, display an error
- If the button is pressed, turn on the LED
- If the password is correct: do the action, else: ask again

In most languages there are three conditions you can use, in this order:

- *If*
- *Elseif*
- *Else*

For *if* or *elseif*, you need to define the condition associated and the code to execute if it's true.
This condition is called a Boolean statement. It's always true or false depending on your variables.
Else applies to any other cases, not described in *if* and *elseif*.

Python

In Python, you'll find something similar, except that *elseif* is called *elif*:

```
value=input("Value?")
if value == 'a':
   print("a")
elif value == 'b':
   print("b")
else:
   print("Anything else")
```

For each condition, the syntax is to use the main word (*if*, *elif* or *else*).
For *if* and *elif* you'll find the Boolean statement just after (*value == "a"*). It's true if the value contains 'a' and false otherwise.

The *else* doesn't have any condition, it will execute the code below if the value is anything other than a or b.
The ":" at the end of the line is here to indicate the end of the condition.

When you run this code, you can try to enter a, b or any other value when asked and see what happens.
Unlike the example in the previous paragraph, the display will be different depending on the input value you type.

Go further

The "*elif*" and "*else*" statements are optional.
If you have only one value to test, you could remove one part, like this:

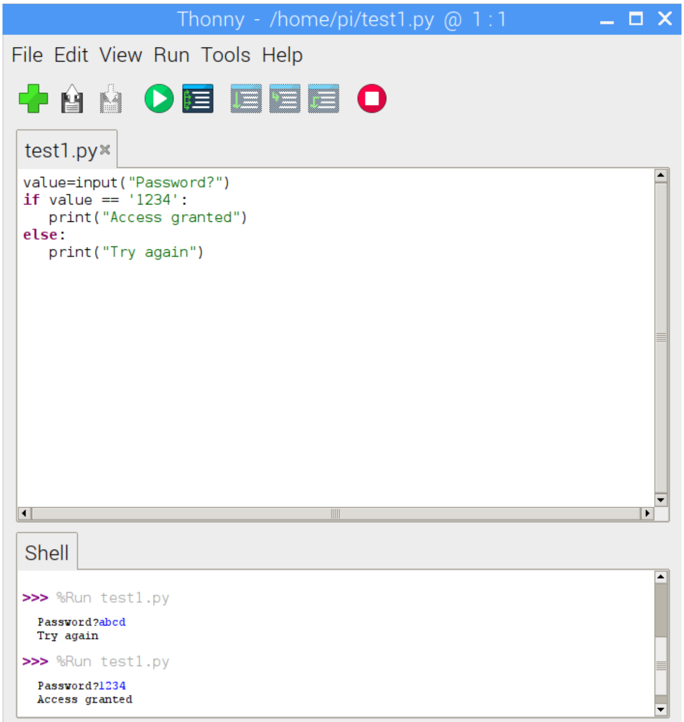
```
value=input("Password?")
if value == '1234':
   print("Access granted")
else:
   print("Try again")
```

If you have more than two values to test you can add as many *elif* as necessary

```
value=input("Value?")
if value == 'a':
   print("a")
elif value == 'b':
   print("b")
elif value == 'c':
   print("c")
elif value == 'd':
   print("d")
else:
   print("Error")
```

When the code below the matched condition ends, the script will continue after the end of the conditional group.
It will not try the other conditions.

In Thonny you should get something like this:



If you start learning to code today, take time to digest this because the following is even more complicated :)
It's a type of logic to learn, used for more than just one language.
Don't hesitate to make some tests on your side to be sure you understand this part.

Homework

I hope you liked this introduction chapter about Python on Raspberry Pi.
We'll see many more things in the following chapters, but it's important to learn this step by step.

Coding requires practice!
If you just started learning how to program today, you'll not be able to make something perfect the first time.
You have to try things, learn others and, most importantly, do not stop there!
If you do nothing in Python in the two next weeks, you will probably forget everything :/

Here is a quick exercise you can try at home to check that you have understood everything:

- Define a static number at the beginning of your script
- Make a script asking the user to find a number between 1 and 10
- The script can give 3 possible answers:
  - You win!
  - It's higher!
  - It's lower!

Let's see if you can find a dynamic way to do this in Python.
I didn't tell you but you can use < and > to compare variables, good luck :)

Day 20:
GPIO Introduction

Introduction

After a big part of this book about Linux basics, operating system, projects and programming, it's time to try the electronics aspect of the Raspberry Pi.

GPIO Pins on Raspberry Pi are an important main feature you need to know.
As I'm a Linux administrator, I mainly used it for testing systems and software in the first few months.
But to go further and create your own electronic systems and programs you need to learn how to use them.

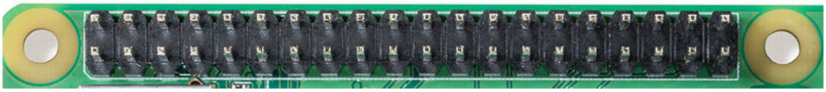
It's not mandatory to master the electronic circuits with Raspberry Pi, but you should at least know that they exist.

Presentation

Let's start with the basics about GPIO.

What are GPIO pins?

GPIO stands for General Purpose Input/Output.
It consists of the 40 pins you can see on the Raspberry Pi, near the edge.



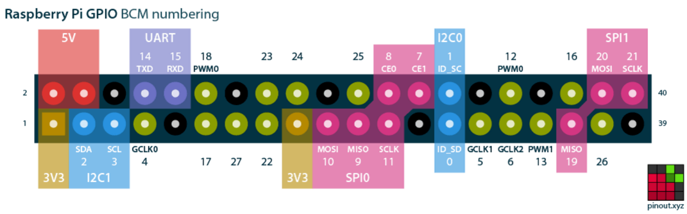
The goal of the GPIO Pins is to add some extensions to your Raspberry Pi.
For example, most of the Raspberry Pi HATs that we will see in the next chapter use these pins to connect to the Raspberry Pi.
You can also create your own electronic circuit by using these GPIO pins with cables, LEDs and other accessories. We'll see more of that later.

GPIO pinout

As you may have guessed, each pin has a specific purpose and you can only use it for that purpose.
Some of them are input/output, power (3.3V or 5V) or ground.

There are even more complex things we'll see later.
So, it's important to know what is what.

Here is an illustration:



The website **pinout.xyz** is useful for this as it gives you the exact layout and role of each pin.
I recommend printing this image to use it later.

Raspberry Pi configuration

Before starting with the GPIO pins practice, we need to do a few things on the Raspberry Pi to make sure that everything is ready.

I'm using Raspberry Pi OS Lite to write this tutorial; refer to chapter 9 and 10 if you want to install it too.
It works on Raspberry Pi OS Desktop if you prefer (just start a terminal to enter the following commands):

- Start by updating your system:

```
sudo apt update
sudo apt upgrade
```

- Install the rpi.gpio package:

```
sudo apt install python3-rpi.gpio
```

- Enable I2C and SPI in raspi-config:
  - You don't need them in this tutorial so you can skip this step.

- But if you go further after this one it may save you time, as nobody else explains this

  ```
  sudo raspi-config
  ```

  Go into “Interface Options”
- And enable I2C and SPI in each submenu

I2C and SPI pins are specific GPIO pins.
You may need to enable them with some hardware modules that use them (a screen for example).

Required hardware

Here is the recommended hardware you need to have to follow the end of this tutorial:

- A Raspberry Pi; any model will be ok
- A breadboard kit will be perfect, with everything you need inside
- Or, at least:
  - A breadboard
  - Some male/female jumper wires
  - Also, male/male jumper wires can be useful
  - LEDs and resistors (you can take this pack if you already have breadboard and wires)
  - I don’t have one yet, but a GPIO Expansion board is an excellent option to know which pin to use

You can find everything on my recommended products page if you are not sure what to choose: **https://raspberrytips.com/go/gpio**
These components will be useful for many projects later. It’s a long list of things you need to get started, but you’ll use them every time after that

That’s it! Once you have everything ready, you can move to the next part.

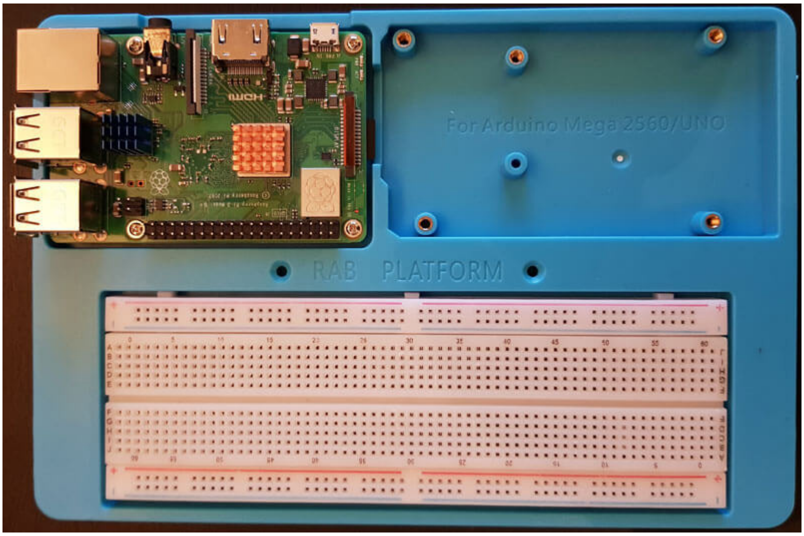
Theory: breadboard basics

Let’s start with the breadboard.
If it’s your first time with it, you may have a hard time understanding how it works

Breadboard installation

If you took the breadboard kit I recommended before, the first step is to install it in the blue plastic case.
Generally, you have to stick it in the large space and screw the Raspberry Pi to the corresponding location.

At this point your setup should look like this:



Don't plug in the Raspberry Pi power cable for the moment.

Breadboard schema

Before going further, you need to understand how a breadboard works.
Each hole is a pin you can use to plug something in.

Power input

On the edges of the board, there are two lines:

- The red line is for the power input
- The blue line is for the ground

Each port is connected with all the pins from the same line.

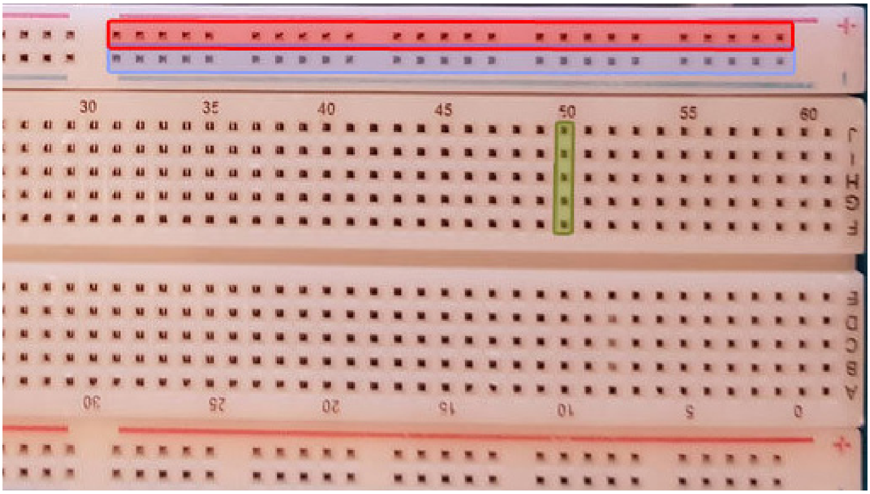
Note: With some breadboards (like mine), there is a separation in the middle, you can see a gap in the red line for example.
If you plug in near port 0, it will not work near port 50.

Other ports

The other ports can be used for everything else (LEDs, resistors, other modules).
A wire connects them in columns.
If you take one number on the board, it connects each port from the same column with the others.

Schema

It will be clearer with this picture:



I have squared each connected port.

The first line corresponds to a power input line.
There are four lines of this type on the board.
If you input power in one of the squared ports, you can use it from any other highlighted ports.

For the ground ports it's the same thing (second line).
And for the other ports you can see in the vertical square how they are connected.
It's the same for each column, for both sides of the middle line.

Exercise: blink the led

Ok, that's the end of the theory part, and the beginning of your first circuit tutorial.
Let's practice :)

Breadboard configuration

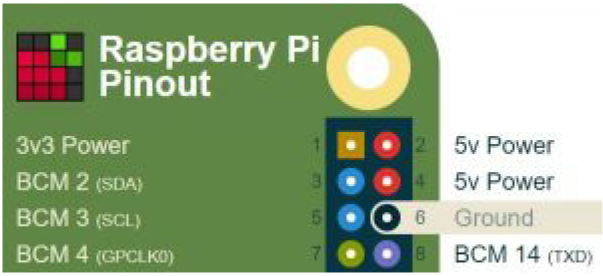
As always, start to plug the pins in without a power source plugged in.

To get started you need:

- 1x LED
- 2x male/female jumper wires
- 1x resistor

Take the two jumper wires and plug them like this:

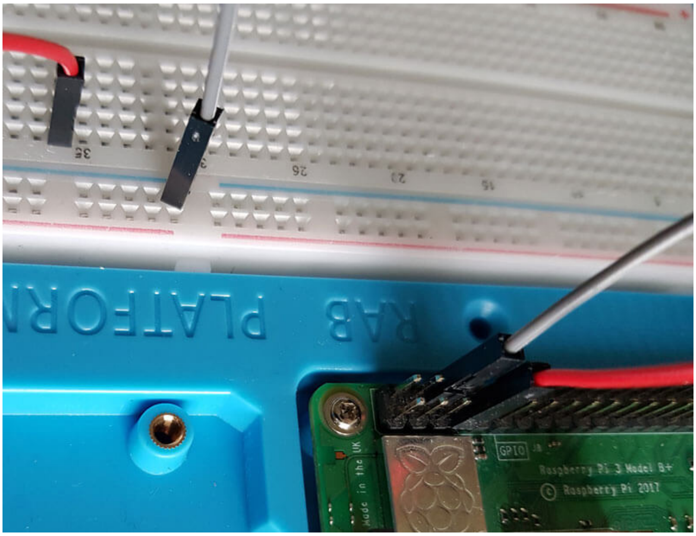
- One from the ground line to a ground pin of the Raspberry Pi (for example the third one from the second row, port 6)
- The other from an input/output port (for example the fourth one of the first row, port 7)



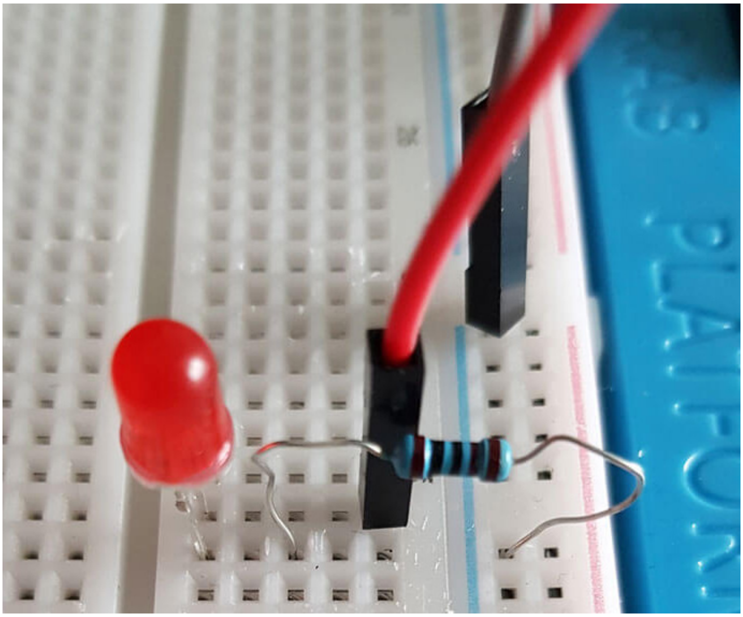
You now have two cables, connected only on one side.
On the other side, you need to connect them to the breadboard.

Plug the ground jumper wire to the ground line of the breadboard (any port).
Plug the other cable to one column of the breadboard (anywhere).

It should look like this:



Finally, put the resistor between the ground line and the column near the other cable.
Put a light with one foot on each column, the shorter one in the resistor/ground column.



You can now boot the Raspberry Pi and jump into SSH (or GUI if you prefer) to create the Python script.
As soon as the Raspberry Pi is on, avoid touching the circuit.

Python script

The first basic script we can code is to turn on the LED to check that everything works.
To do this, Raspberry Pi OS already includes all of the libraries you need:

- In a terminal, or with your favorite code editor, create a new python file:

```
nano led_on.py
```

- Write these lines inside:

```
#import libraries
import RPi.GPIO as GPIO
import time

#GPIO Basic initialization
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)

#Use a variable for the Pin to use
#If you followed my pictures, it’s port 7 => BCM 4
led = 4

#Initialize your pin
GPIO.setup(led,GPIO.OUT)

#Turn on the LED
print “LED on”
GPIO.output(led,1)

#Wait 5s
time.sleep(5)

#Turn off the LED
print “LED off”
GPIO.output(led,0)
```

I commented everything out so it should be clear.

The only trap is the pin number to use in the led variable.
You may have a different numeration in your breadboard expansion board or sticker, but you have to use the BCM number from pinout.xyz:
https://pinout.xyz/pinout/ground

- Save and exit (CTRL+X)
- Run the script with:

```
python led_on.py
```

The LED will turn on for 5 seconds and then turn off.
If it doesn't, double check every previous paragraph to see what you have missed!
If your circuit seems good, check the LED orientation, as there is a + and – side.

Homework

That's it, you have now learned the basics about GPIO pins and breadboard.
You're able to build basic circuits and you're ready for the next level :)

As homework you can for example try to mix this led circuit with the python basics we saw previously:
For example, ask the user something, and only turn the LED on if the answer is correct.

We won't see a lot about GPIO pins in this book.
If you are interested, don't hesitate to check other books and online tutorials to learn more about this.

Day 21:
Discover Raspberry Pi HATs

Introduction

In the last chapter, we saw what GPIO pins are and how you can build an electronic circuit with them, using the Raspberry Pi as the brain of the circuit.

This chapter is the logical follow up from the previous one.
As GPIO enables the possibility to extend the list of Raspberry Pi features, many manufactures work to offer new expansion cards, just like on a PC.
Each extension can add many features, depending on what you want to do.

So, in this chapter, I will start by explaining what a “HAT” is, then I will give you a few examples and finally we’ll try a short demonstration to show you exactly how it works.

What is a “HAT”?

We often use the term HAT to define a Raspberry Pi extension.
It stands for “Hardware Attached on Top.”

In short, it’s an additional card that you plug in to the top of your Raspberry Pi to give it some new features.
Generally, it uses the GPIO ports to connect the two cards together.

Don’t panic; the assembly is easy.
It requires no soldering, so you can plug and unplug it whenever you want to.
And, even better, the installation is automatic most of the time.
The Raspberry Pi will recognize the HAT thanks to an EEPROM module on the board that identifies the HAT model.

So, don’t be afraid to use HAT with your Raspberry Pi. It’s effortless and similar to an additional PCI card on your computer.
Just plug it in, maybe install one thing, and it’s ready to use.
You don’t need to be an experienced handyman to play with these toys :)

My 4 favorites Hats

It's time to browse my selection of the best HATs for Raspberry Pi.
There is no specific order; all of them are excellent and for different purposes, so it's not really possible to rank them in any way.

Sense Hat

The Sense HAT is the one you absolutely must try.
It's a bestseller, with many reviews, tests and publications.

The sense HAT provides many new sensors to the Raspberry Pi:

- **Accelerometer** (get the movement speed of the Pi)
- **Gyroscope** (capture the rotational movement of the Raspberry Pi)
- **Magnetometer** (magnetic field measurement)
- **Air pressure sensor**
- **Temperature and humidity sensors**

And there is also an LED display matrix and a joystick on the top of it.

The Raspberry Pi Foundation created this HAT originally for the Astro Pi project, to send Raspberry Pi to space and measure everything they possibly could.
But even on earth it's possible to make many projects with this extension (educationally at least).

Note: If you are interested in any of these HATs, I will give you a link to easily find them all at the end of this part.

Adafruit Servo

The Adafruit Servo board allows you to connect up to 16 Servo motors to your Raspberry Pi.
I call this HAT the "Arduino adapter" because it makes the difference between a Raspberry Pi and an Arduino.

The main strength of the Arduino is that it is good in a complex electronic circuit to manage physical devices like buttons and motors.
But the Raspberry Pi is better at being the brain, not the hand.
This extension, however, allows the Raspberry Pi to take on both roles.

If you are interested in this topic, check my post titled, "Why is the Raspberry Pi better than the Arduino?" :)
https://raspberrytips.com/go/arduino

Meanwhile, if you do want to build projects with motors in the future, I highly recommend trying this extension.

PoE Hat

The Raspberry Pi PoE HAT is a recent product created by the Raspberry Pi Foundation.
The goal is to power your device with your Ethernet cable.

PoE stands for "Power over Ethernet" and allows you to power devices like phones without an additional power source.
With this HAT on your Raspberry Pi and a PoE switch, you can remove the power supply.
It's awesome because you only need an Ethernet socket somewhere to plug your Raspberry Pi into.

There is no additional RJ45 port on this HAT; it'll use the default one and collect the electricity from it to run your Raspberry Pi.
This HAT is available on Amazon or on the official Raspberry Pi website.

Professional DAC

The HifiBerry DAC+ adds two RCA connectors to your Raspberry Pi.
For audiophile and sound projects, it may be useful to get connectors and better sound quality on your Raspberry Pi.
Or even just to get a microphone, as the Raspberry Pi doesn't provide any sound port input.

DAC stands for "Digital-to-Analog Converter" and this one provides 92kHz/24bit high-quality audio.
It's directly used by the Raspberry Pi operating system and you can control the volume with the ALSA mixer.

Where to find them?

As promised, you can find these 4 HATs on the RaspberryTips website.
You will also find many more HATs options, and which features you can get with them, I have a full guide on RaspberryTips.

You can find the top 13 Raspberry Pi HATs here:
https://raspberrytips.com/go/hats
This will allow you to get a better overview of the possibilities.

The Sense Hat in practice

Theory is useful, but if you buy one of these HATs, you probably need some information on how to actually use it.
In this part, I will take the example of the Sense Hat, and show you step-by-step how to control it from your Raspberry Pi.

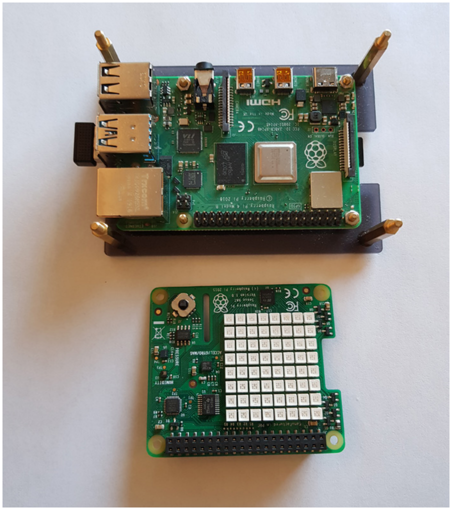
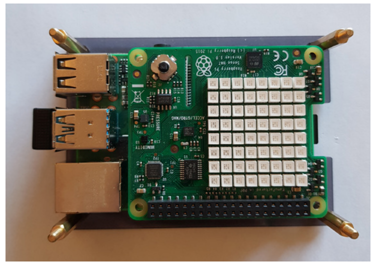
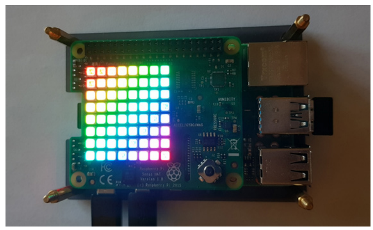
If you don't have a Sense Hat yet, I still recommend reading this practical information, as it gives you guidance on how a HAT works on a Raspberry Pi.
You can also follow this part with the "Sense Hat Emulator" on Raspberry Pi OS Desktop (pre-installed on the Full version only I think).

Plugging it

I'm using Raspberry Pi OS Desktop in this exercise to show you screenshots of what I'm doing
But you can use the Raspberry Pi OS Lite version, as seen in the previous chapters.

The first thing to do is to plug it in to the Raspberry Pi.
It's not difficult as the Sense HAT has GPIO connectors that slot into all the pins.

Start with the Raspberry Pi turned off.
Then, just put the Sense HAT on the GPIO pins and push it in gently to connect it up.
And finally, start your Raspberry Pi.
All the LEDs on the top will light up to draw a rainbow if it's properly plugged in.

You can use the screws provided to attach the HAT to the main board, but it's not mandatory.

Raspberry Pi OS Configuration

You now need to configure Raspberry Pi OS a little to make it work with your Sense HAT.

So, make sure that the Sense HAT package is installed on your system:

```
sudo apt install sense-hat
```

That's everything you need to do!
It will install the required dependencies automatically (like Python and the various libraries).

A basic Python script

The goal here is to show you a few key features of the Sense HAT.
I will go through them step by step, giving you the useful functions and then using them in a basic script in Python.
You'll learn more about the Sense HAT and also work on the Python basics we have seen previously.

What's the goal of the script?

The goal is to read the temperature, humidity and pressure from the Sense Hat, and then display it on the screen.

As a reminder, the Sense HAT can:
- Get the current temperature, humidity and pressure
- Set each LED to a specific color
- Display letters or text on the screen, and scroll it along

This list is not exhaustive, it's only what we will use in this exercise.

Useful Python functions

The Sense HAT package provides many functions within the Python library.
You can find an entire list here: **https://raspberrytips.com/go/sensehat-api**
But now I will just give you the ones that are needed for the exercise.

Use the Sense HAT library

To use the following functions in your script, you need to first include the Python library for the Sense HAT at the beginning of your script.

A library is a source code that provides everything you need for a specific usage. This avoids having to code it all by yourself.

Here are the two lines to add at the beginning of each Python code:

```
from sense_hat import SenseHat
sense = SenseHat()
```

The first line is used to import the library, the second to initialize the "sense" variable that you will use in the entire script.

Get data from the Sense Hat

Here are the 3 functions to get the data we need:

- `sense.get_temperature()`
- `sense.get_humidity()`
- `sense.get_pressure()`

The names are explicit.
They don't need any parameters, and they return the value directly as a float.

Temperature is in degrees Celsius; humidity is a percentage and pressure is in Millibars.

We get them with many decimals places, so we will round them to only one with:

```
round(value)
```

Display text on the screen

The second thing we want to do is to display something on the screen.
As the screen is really small, it would just be for the exercise; not very useful in real life :)

Here is the corresponding function:

- `sense.show_message(text_string, scroll_speed, text_colour, bg_colour)`

And here is a short explanation for the parameters:

- Text_string: the string you want to display on the screen
- Scroll_speed: the speed of the text scrolling. By default, it's 0.1 and you can play with it to make the text scroll faster or slower
- Text_colour: the color of the LEDs that will display the text string (in RGB)
- Bg_colour: the color of the screen background (also in RGB)

Aside from the text_string parameter, everything is optional (default is black and white if not set).

RGB stands for red, green, blue.
You can use a tool like this to find a specific color in RGB:
https://raspberrytips.com/go/rgbcolor

In Python, you need to set this value in a list (a set of values), like this:

```
color = (255, 0, 0)
```

Put everything together

You now have everything you need to create the script by yourself.
I think it's good practice to try it before seeing the solution.
Take your time, check the errors and fix them step by step.

And obviously, here is my solution:

```
from sense_hat import SenseHat
sense = SenseHat()

# Define the colors in RGB
red = (196, 25, 73)
white = (255, 255, 255)

# Get the values from the Sense HAT
t = sense.get_temperature()
p = sense.get_pressure()
h = sense.get_humidity()

# Round the values
t = round(t)
p = round(p)
h = round(h)

# Define the text
text = "Temperature: " + str(t) + " Pressure: " + str(p) + "
Humidity: " + str(h)

# Display the text
sense.show_message(text, text_colour=white, back_colour=red,
                    scroll_speed=0.2)
```

Write this source code in your favorite editor and save it with a ".py" extension, like myscript.py

Some notes about this code:

- My "red" is the one from RaspberryTips. You can use any color you want
- We store the data in variables (as seen in Chapter 20)
- The "+" character is used to concatenate several strings
- The str() function turns a float value into a text string, so you can concatenate it with other text
- And then we call the `show_message()` function.
  As you can see the parameter order does not matter
- To run the code, use Thonny on Raspberry Pi OS Desktop
  Or use the following command in a terminal:
  `python myscript.py`

If you want to go further and discover the other features, I have an entire tutorial about this:
https://raspberrytips.com/go/sensehat-tuto

Homework

You have seen here the power of the combination of the Raspberry Pi GPIO pins and Python. It's really an important part of the knowledge you need to get from this book, that's why the next chapter is again about Python.

Make sure you understood everything here by reading this chapter again if needed.
You can also try to play with other Sense HAT functions.

As homework, I want you to try this:

- Take my source code for the Sense HAT exercise
- Define 3 or 4 background colors (like a gradient or traffic light colors)
- Then change the background color of the screen depending on the current temperature (fix your values)

Good luck!

Day 22:
Advanced Python Notions

Introduction

In chapter 19, we looked at the basics of Python (presentation, hello world, variables and conditions).
Now it's time to see some other useful instructions like arithmetic operations and loops.

As usual, I will start with the theory, and give you a few examples at the end.

Operations

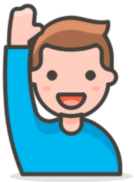
This part will be short, as I will just give you the four main arithmetic operations and how to write them in Python.

The four basic operators you can use are the following:
- Addition: +
- Subtraction: -
- Multiplication: *
- Division: /

Easy, right?

Here is a short example:

```
a=12
b=3
print(a+b)
print(a-b)
print(a*b)
print(a/b)
```



Hardware notice

How do you know if the + is an addition or a concatenation?
In a previous chapter, I showed you how to concatenate strings with the +. It's the same for the addition. In fact, it only depends on the variable type (string or number).
Number: ***var=3***
String: ***var="3"***

Loops

Loops are always a difficult concept when it comes to your first programming language.
It is essential to master this to create interesting scripts later.
You cannot do without it, but do not worry, I'll explain little by little.

Theory

Loops are a way to execute the same portion of code several times.

Let's look at some examples:

- As long as the password entered is not the right one, ask for it again and again
- For each file in a folder, do something
- Infinite loop: Do the same thing continuously, or every X minutes

This concept will allow us to have dynamic code depending on something else.
It also avoids having several identical pieces of code, or having to execute the script too often.

In Python

In Python, you only have two ways to create loops:

- **For**: Execute the code for each item in a sequence (list, range, string, ...)
- **While**: As long as your condition is true, the code will be executed

For those who know it from another language, there is no "foreach" instruction in Python. You have to use "*for*" instead.

While

I will start with "*while*" because it's very close to the conditions we have seen before.
As I said it will run your code until the Boolean statement becomes false.

Here is the same password example, with a loop:

```
password=input("What's your password?")
while password!="1234":
  print("Error")
  password=input("What's your password?")
print("Password correct")
```

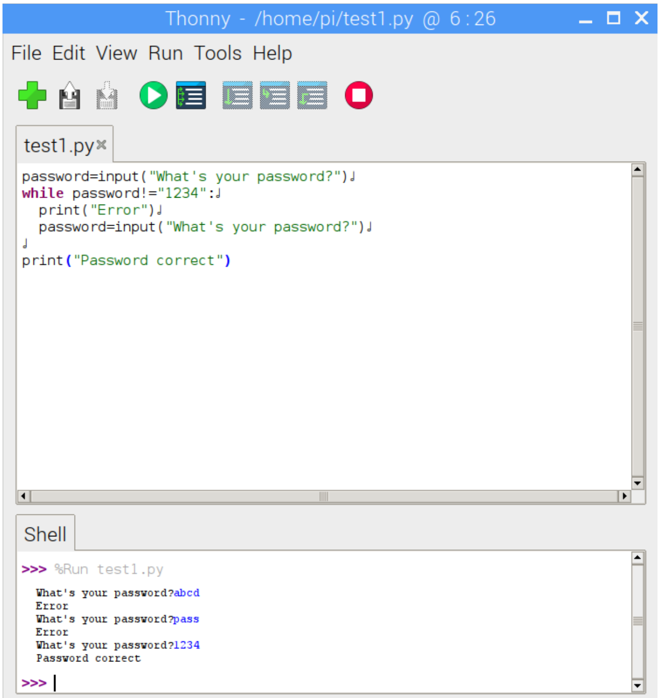
We already know almost everything in this piece of code.
We start by asking for the password.

Then there are two cases:

- **The password is correct:**
  - The script will never enter the *while* loop because the condition is false
  - It's the same thing as an *if* statement in this sense
- **The password is incorrect:**
  - The script will enter the *while* loop
  - It displays an error and asks for the password again
  - After that, the script comes back to the "*while*" word, and tries for the same Boolean condition again
  - This loop will not stop until the password is correct

Don't forget to change the variables in the *while* condition inside the loop.
Otherwise, you will create an infinite loop!

The execution of this script in Thonny with two password errors look like this:



For

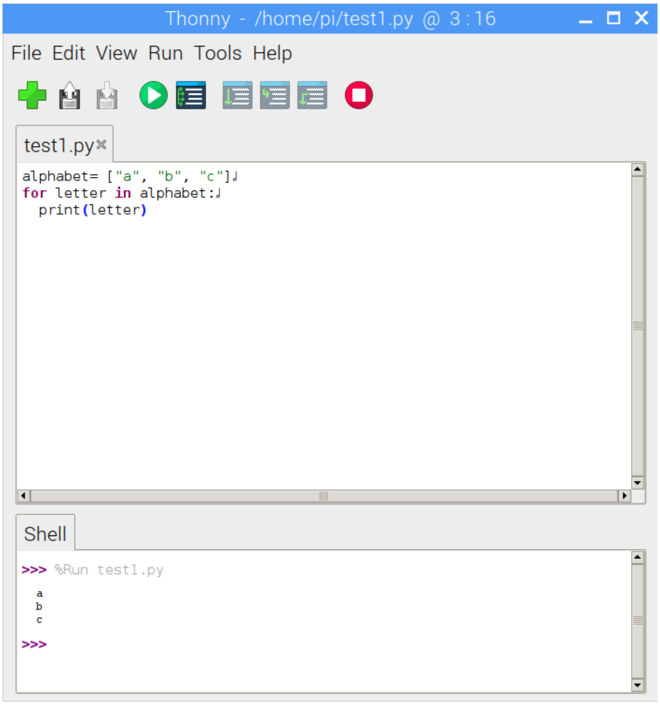
For is something a little different
You'll use it for an already defined sequence of things, not depending on the user entry or something that evolves.

Here is one example:

```
alphabet= ["a", "b", "c"]
for letter in alphabet:
  print(letter)
```

On the first line, I introduce a new type of variable. It's a list of elements (like an array in other languages).
The *for* syntax creates a new variable with the current item of your list (I call it "letter").
Then you can use this variable in the loop code.

In Thonny you will get this:



As you can see, the code is executed three times, once for each letter.
I only use one line, but there is no limit; there could be 300 lines.

Examples

Ok, you now have the basic knowledge in Python (with chapter 19 and this one) to start making complete programs.
We can focus on your main goal: finding Raspberry Pi projects you can do with some basic Python skills.

Here are two real examples to illustrate how to use loops in your projects.

Control your camera

If you have a camera on the camera port of your Raspberry Pi, you can try to control it with Python.

There is a library called "picamera2" that allows you to use your camera in a Python script.

Here is a basic sample including the main functions:

```
import time
from picamera2 import Picamera2, Preview

picam = Picamera2()

config = picam.create_preview_configuration()
picam.configure(config)

picam.start_preview(Preview.QTGL)

picam.start()
time.sleep(2)
picam.capture_file("test-python.jpg")

picam.close()
```

This script will take one shot each time you run it and save the picture in the specified folder.
Don't forget to import the picamera2 library at the beginning of the script.
Then you can use ***start_preview*** to start the camera, capture to shoot the picture and ***stop_preview*** to stop the camera.

Note: Like for the commands to use the camera, this is working differently on Raspberry Pi OS Legacy. You can find more details here:
https://raspberrytips.com/install-camera-raspberry-pi/

GPIO Pins

Another great feature of the Raspberry is the GPIO pins you'll find on the board. These pins allow you to create an electronic circuit with components like LEDs, buttons or whatever else you want.
And you can control it with a Python script!

I'm not an expert in electronic circuits, but I will give you a short example using the GPIO Zero library:

```
from gpiozero import LED
from time import sleep
led = LED(17)
while True:
    led.on()
    sleep(1)
    led.off()
    sleep(1)
```

At the beginning of the script, we need to import two libraries (*gpiozero* for LED management and *time* for the *sleep* function).
Then we set the GPIO pin to use for the next stage of the script.
And finally, we create an infinite loop which makes the LED blink every second.

There are other libraries similar to *gpiozero* you could use, like *RPI.GPIO*, *pigpio* or *wiringPi*.
If you are just starting out, *gpiozero* may be simpler to understand how it all works.

Homework

Yes, I do have a cool piece of homework for you this time :)
Once you have read and understood this chapter, here is what you can try:

- Run the code from the previous chapter (Sense Hat), and make sure that it's still working
- Now try to add a loop inside it.
  The script we made doesn't actualize the values from sensors.
  The goal here is to display the message, then actualize it, display the message again, etc.
- If you don't have a Sense Hat yet, you can replace the *show_message()* function by the *print()* function to check if your changes are working correctly

Good luck!
If you are interested in Python and want to learn more, try various projects, I have another book dedicated to this topic. You can find all the details here:
https://school.raspberrytips.com/master-python-on-raspberry-pi

Day 23: Raspberry Pi Zero Introduction

Introduction

Today I want to introduce the Raspberry Pi Zero.
It is a different Raspberry Pi model, with limitations but also benefits.

In this chapter, I will give you a general presentation, then the hardware specifications, and finally a step-by-step procedure on how to install it without a screen and keyboard (which is often the case for this model).

Presentation

In chapter 1, you already got a glimpse of the Raspberry Pi Zero.
However, I won't blame you if you have forgotten as it was right at the beginning.
I will try to give you more details now that you have a stronger knowledge about Raspberry Pi.

So, the Raspberry Pi Zero was launched in 2015 by the Raspberry Pi Foundation.
Its two strengths were the price ($5) and the compact format.
But this format implies several limitations in the hardware they used (1 micro USB port only, no Wi-Fi, etc.)
Things have evolved since that time, but there are still many constraints. We will see this in more detail in the next chapter.

Its compact format allows you to use it in specific environments, like embedded systems or robots.
It is also often used for hacking and pen-testing (to test a company's security level).

If you want more examples, there is a list of 16 projects ideas with a Raspberry Pi Zero on RaspberryTips: **https://raspberrytips.com/go/zero-projects**
These include a Game Boy with Retropie, a drone, etc.

Hardware presentation

The hardware part is mostly a reminder as we already saw this topic on Day 1.

Specifications

The Raspberry Pi Zero is the smallest and cheapest Raspberry Pi available.
So, you don't have to expect the best performance or hardware components on it.
But for $15 you can get decent results.

The Raspberry Pi models' names are almost the same, so make sure to order the latest version if you want to buy it now.
The latest version name at the time of writing is Raspberry Pi Zero 2 W.
Refer to chapter 1 for more details about this.

So, here are the specifications for the model W / WH:

- **Processor**: 1Ghz, single core (version 1) or quad core (version 2)
- **Memory**: 512M
- **USB ports**: 1 (micro USB)
- **Video**: Mini HDMI
- **GPIO**: 40 pins, compatible with HATs (or generally pHAT, with a smaller size)
- **Camera port**: available
- **Network**: Wi-Fi, Bluetooth

As you can see, there is no possible comparison with the Raspberry Pi 4, with a quad-core CPU and 8 GB RAM.
It is also not a good choice if you need to plug in many USB devices.
Even a mouse and keyboard will be complicated (hence the next part).
A screen is possible, but you need an adapter, etc.

The main strengths are the compact size and low-power consumption, not the embedded hardware.

But I love mine, I mainly use it as a VPN server when I am outside, and it works perfectly.
It runs quietly and discreetly, laid on its charger on a power socket I never use :)

Headless installation

If you read the previous parts carefully, you will understand that the main problem with the Raspberry Pi Zero could be the installation. You don't necessarily have a compatible cable for the display and no USB ports to connect your keyboard.

So the easiest way is to do what we call a "headless installation", which means that
your SD card should be pre-configured before the first boot.

Please read this part even if you don't have a Raspberry Pi Zero.
It can work with any model, and I use it all the time to speed up the installation process.

Prerequisites

Don't worry, there isn't anything complicated. Thanks to the release of Raspberry Pi Imager, the process is way simpler now than it was a few years ago when you had to do everything manually.

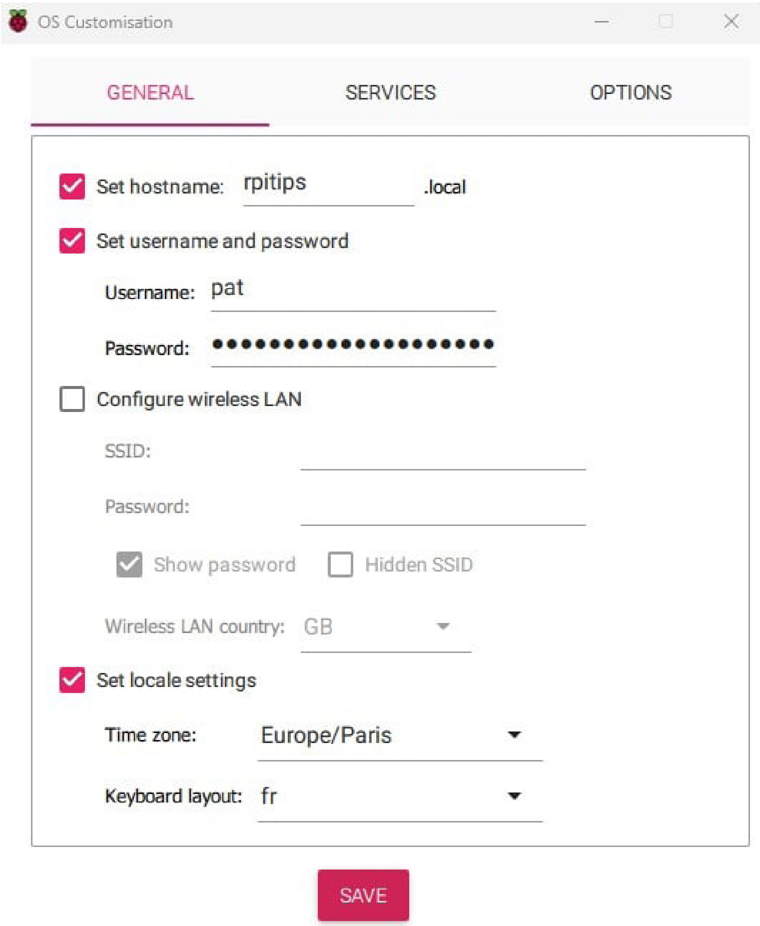
The only thing you need is a computer with Raspberry Pi Imager to flash the SD card (it can even be done on Raspberry Pi directly if you have a USB to SD card adapter, or if your main storage is on USB).

SD Card preparation

Basically, you'll follow the typicalsame steps as usual to flash Raspberry Pi OS to your SD card, using Raspberry Pi Imager.

This works perfectly with all Raspberry Pi OS versions (Lite included), and may work with other Debian-based distributions.

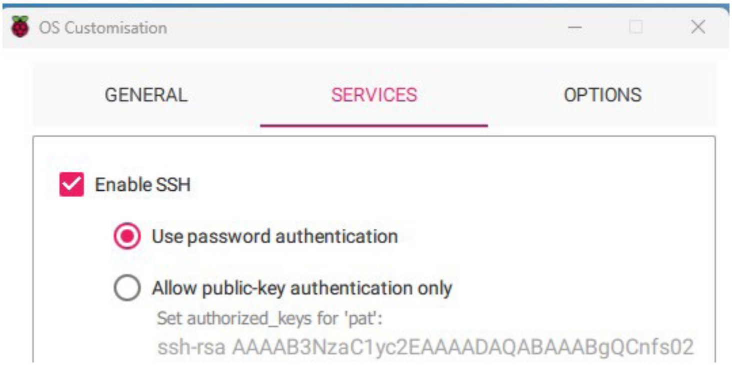
Anyway, what's important when you use Raspberry Pi Imager is to fill the "OS Customisation" tab completely, with everything required for the Raspberry Pi to connect to your network automatically, with remote access allowed.



The purpose of a headless setup is to skip the Welcome Wizard and be able to access the Raspberry Pi directly from a remote computer after the first boot. To do this, the username, password and network configuration must be completed through the "OS customisation" tab.

By default, Raspberry Pi OS won't boot completely, and will show the welcome wizard on the first boot, everything is set here. If you don't use Ethernet and want to access the Pi directly after the boot, you also need to configure the Wi-Fi network in this form.

Then, click on "Services" and make sure to enable SSH:



SSH is the remote access protocol that will give you access to the Raspberry Pi directly after the first boot. It's not enabled by default, unless you check this box, and have a username and password set.

Once all the configuration is done under "OS Customisation", click on "Save" and proceed with the SD card writing. Raspberry Pi Imager will apply the settings directly to the SD card or USB drive.

First boot

We can now boot the Raspberry Pi.

Insert the SD card

Eject the SD card from your computer (preferably by using the safely remove tool).
Insert it into your Raspberry Pi and start it.

The Raspberry Pi Zero starts as soon as you plug it into a power source, there is no start button.

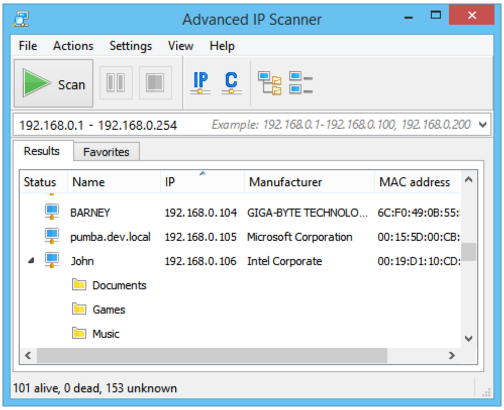
Find the IP address

Once the Raspberry Pi has started, you can connect to it with SSH.
But before you do this, you need to know the Raspberry Pi IP address.
I have an entire tutorial about that on the RaspberryTips website, but I will give you a short answer for Windows and Linux/Mac here.

Windows users

I recommend a free tool, which is called Advanced IP Scanner that can scan an entire network or just a portion of it.
The interface is very intuitive, and it might be useful for other projects.
Here is the link to the official website to download it:
https://www.advanced-ip-scanner.com/

I will give you a screenshot here so that you can get an idea of it:



Just press "Scan" without changing anything else, and the software will list all the IPs on your network.
By default the Raspberry Pi's name is "raspberrypi", so you'll find it easily if you only have one on the network.

Linux/Mac users

On Linux and probably Mac too, you don't need any software.
You are now used to the command line and I have the perfect one to do this.

A quick way to find the IP address is to display the arp cache of your machine.
The arp cache is globally a list of associations IP <=> mac address for your network.

On Linux or Mac you can display it by doing:

arp -a | grep raspberry
raspberrypi.home (192.168.1.15) at b8:27:eb:1a:40:c0 [ether] on wlxc8d7193bf721

Even you need another solution or more details, check the entire post here:
https://raspberrytips.com/go/ipscan

Connect to the Raspberry Pi

You are now confident that the Wi-Fi configuration worked, and you have the Raspberry Pi IP Address.
So you can connect to it like we have already done in previous chapter (Day 10 for example)

On Windows, start Putty, type in the IP address and connect
On Linux, `ssh pi@IP_ADDRESS` in a terminal.

Homework

That's the end of this chapter about the Raspberry Pi Zero and you have also seen some interesting tips on how to pre-configure a Raspberry Pi before the first boot.
You are on your way to becoming a true "master on Raspberry Pi."

This chapter may seem useless if you don't have a Raspberry Pi Zero, but it's in these details that the difference between an amateur and a master is present.
You have to know how to optimize the Raspberry Pi installation, and in many projects you may build in the future, the Raspberry Pi Zero is the best choice.
Do not limit yourself with your knowledge, learn the details.

Now, as homework, I want you to try the headless tutorial with any Raspberry Pi model you have.
Make sure you follow the procedure for the SD card preparation.
Find the right tool for you to find the IP address, and check that everything works.

See you on the next day :)

Day 24:
Raspberry Pi Zero Accessories

Introduction

As we saw in the previous chapter, the Raspberry Pi Zero is a unique model, mainly because it's smaller.
This model is a good choice for cheap projects, or when it is advantageous for the Pi to be small.

But you can also improve the Raspberry Pi Zero with many accessories to make it the ultimate tool for your project.
This part is a mix between basic accessories you need with any Raspberry Pi Zero installation, and original add-ons that makes the Pi Zero an excellent choice for many projects.

Pi Zero HATs

In a previous chapter, we have seen the Raspberry Pi HATs that are available.
But here I'll show you specific HATs you can find for a Raspberry Pi Zero.
As the size is smaller, you'll generally not use the same accessories on a Raspberry Pi Zero as for the other models.

SIM / 3G / 4G HAT

The first feature you can add to your Raspberry Pi Zero is support for phone calls, SMS or data transfer (3G / 4G).
Thanks to the small size of the Raspberry Pi Zero, you can use it in many projects where external connectivity can be useful, for example:

- Drones
- Phone or text messenger
- Pen-testing in an unknown environment

On the following link, I will give you an example of a good HAT you can find.

There are several options on the link depending on what exactly it is that you need, and you can also find other brands with similar features.

Link: **https://raspberrytips.com/go/pizero**

LCD Display HAT

You may also need a screen and a way to control the Raspberry Pi Zero in your project.
An external screen or a big display HAT will not fit exactly your needs, but don't worry, you can find cheap solution to add a screen on a Raspberry Pi Zero.

The one I suggest even has a joystick and 3 keys on it, so you can program them for the main features of your project.
For example, it could be a dynamic monitor screen, or a menu that uses the joystick and keys to make a selection.

You can also find the link to this product on
https://raspberrytips.com/go/pizero

Sound Card

A Raspberry Pi Zero may also be a good choice for home projects like a Hi-fi system (you have seen Volumio in Day 16 for example).
The main issue is that the Raspberry Pi Zero doesn't have an audio output (jack) like other models.
But you can find many HATs to add this feature and enjoy better quality music.

For example, this pHat by Pimoroni can be plugged directly into the Raspberry Pi Zero GPIO pins, and adds a Stereo jack output.

If you are interested in an input microphone, I recommend choosing the ReSpeaker Pi Hat.
It's a perfect solution for an Alexa or Google home assistant project.

Both are available here: **https://raspberrytips.com/go/pizero**

Sensors

Another HAT you may be looking to use with the Raspberry Pi Zero is one with sensors, compacted in a small sized package.
The Pimoroni Enviro pHAT is the perfect tool for this.

It adds the following features to your Pi Zero:

- Temperature sensor
- Pressure sensor
- Light and color sensor
- Accelerometer and magnetometer sensor

A Python library is included to create your first scripts quickly with these new sensors.

USB Hub

The last one in this list is not really a magic extension, but rather a very convenient accessory.

It's a 4 port USB Hub that you can plug in to your Raspberry Pi Zero to use standard accessories on it.
Basic, but mandatory to get your projects working faster on your Raspberry Pi Zero.

The case is also very useful, so I recommend getting both simultaneously.

It allows you to hold everything together, and the upper case has two add-ons for GPIO users:

- The top cover allows you to solder GPIO pins
- The GPIO pins names are printed on the transparent case to help you when plugging your cables in

As always, an updated link is available here:
https://raspberrytips.com/go/pizero
And I will also update this page with new products over time

Pi Zero Cases

In this part, I'll show you a few cases you can get.
The official one, even though you probably have it, and the also original case that brings new advantages to the Raspberry Pi Zero.

Official case

I will be short with this one, as there is nothing new here.
But if you don't have a case yet, it's the cheapest option you can find, perfectly compatible with a Raspberry Pi Zero W.

This case is provided with 3 lids, so you can change it depending if you are using GPIO pins, camera or nothing at all.
More information on the website.

USB dongle expansion module

The second one allows you to turn your Raspberry Pi Zero into a USB dongle.
The USB is useful to provide power to the board, and also a data connection with your computer so that you can easily access your files from your computer (without SSH or other protocol).

The case is cheap and protects your Raspberry Pi Zero, while also providing new features.

Game Boy case

And the last one is my favorite.
Thanks to the small size of the Raspberry Pi Zero, you can hide it in a Game Boy case!
Many users have built it themselves, and it's now available in a complete kit and is easy to assemble.

Everything is included in this kit:

- 2.8" IPS screen
- The Game Boy case
- The bag and the carrying bag
- Heatsink
- Manual

Just put your Raspberry Pi Zero in it, and find the perfect project for this case :)

All the cases presented here are available on the same page as the previous section
Here is the link: **https://raspberrytips.com/go/pizero**

Pi Zero Cables

For this last part, I will just give you all the cables you will probably need to enjoy your Raspberry Pi Zero.

This model uses specific ports to keep it small, but compatible cables are not provided.
You can find my recommendations for the best accessories to buy here:

- **Mini HDMI to HDMI Cable**: If you want to use a screen on your Raspberry Pi Zero, you will probably need this kind of adapter
- **Micro USB to USB**: The Pi Zero only has Micro USB ports, so you can't plug a keyboard or mouse in without an adapter
- **Power Supply**: Yes, you can use a USB cable or a smartphone charger, but it's better to use the one I recommend your for stability
- **Micro USB to Ethernet**: You can use your Pi Zero with an Ethernet cable by using this kind of adapter
- **Micro USB to USB Hub**: And finally, a USB Hub is a good option if you need to keep several USB devices plugged to it (like an external hard drive, a mouse and a keyboard)

If you are interested, you can find all the links on this page:
https://raspberrytips.com/go/pizero

Conclusion

That's the end of this part about the Raspberry Pi Zero.
If you have just started on Raspberry Pi and already have another model, you probably don't need another one for now.
But as soon as you want to try other things, I really recommend coming back to these two chapters and prepare a Raspberry Pi Zero for a specific project.

It's like trying another single board computer, but already knowing everything about it.
Once you have the right accessories, you can start to enjoy its new possibilities.

Day 25:
Raspberry Pi & Security tools

Introduction

In this chapter, I want to introduce a new facet of the Raspberry Pi potential.
As it is a computer in pocket-size, it allows us to use it as a perfect tool for security audit and pen-testing.

General introduction

I already said this earlier, but the Raspberry Pi, and especially the Raspberry Pi Zero, is one of the most used tools by security specialists and pen-testers to break into a target network.
With its small size, it's easy to plug it in discreetly on a network socket, and work from there to look for a security breach.

With the accessories we have seen in the last chapter (SIM card reader for example), it's easy to get connected to the internet, so as to have access to the device from outside while targeting the local network.

Even though it's possible with any computer on Linux, the Raspberry Pi is the perfect tool to do this, with reliable performances and a tiny size compared to classic computers.

About this chapter

In this chapter, I will show you a specific Linux distribution: Kali Linux.
Its main purpose is to gather a lot of security software in the same system, so you can do everything from it.
I will teach you how to install it, and which tools you can use to test it at home.

Disclaimer:
Most of the tools I will introduce are at a border in terms of legislation.
Hacking another network is illegal and, in many countries, you can go to jail for this. This tutorial is here for educational purposes only.
Try this on your network if you want to learn how hackers work and then protect yourself. But don't use these techniques on other networks without the owner's permission. In no case can I be held responsible for your actions.
Now that this is clear, let's get down to the practice!

Kali Linux

Presentation

Kali Linux is a Debian-based Linux distribution, which includes security and penetration testing tools.
Formerly known as Backtrack, it is used by many security companies and also by hackers
Associated with Raspberry Pi, as it turns it into a perfect hacking kit.

Kali Linux is available for the ARM architecture, so its installation is relatively simple. We will now see how to install it on your Raspberry Pi.

Installation

Create a new SD Card

Here are the steps to install Kali Linux on the SD card:

- Start Raspberry Pi Imager.
- Click on "Choose Device" to filter the OS list.
- Click on "Choose OS" to open the OS list.
- Go to "Other specific-purpose OS" and click on "Kali Linux".
  You may get a list of the different versions available for your model (or just one in most cases).
  Click on the version you want to install (64-bit is recommended if supported).
- Then click on the next button to pick your SD card (you can also use a USB drive if you prefer).
- And finally, click on "Next" to start the installation (with or without custom settings).

After a few minutes, Kali Linux should be ready to use, you can eject the storage device and plug it into the Raspberry Pi.

Install Kali Linux

There is nothing else you have to do :-)

Just insert the SD card into your Raspberry Pi and start it up.
Kali Linux will go directly to the login screen.
No questions or anything; just insert, start, wait.

First Login

Once Kali has started, you need to log in.

The default identifiers are:
– **login**: kali
– **pass**: kali

It is strongly recommended changing them quickly.
You can change it by opening the terminal and typing the command:
passwd

If you do not have a US keyboard, you can change the layout in the Settings, via the Keyboard options.
But be careful, on the login screen you will keep the US layout for the moment, so choose your password knowing this if you want to use it with desktop.

First steps

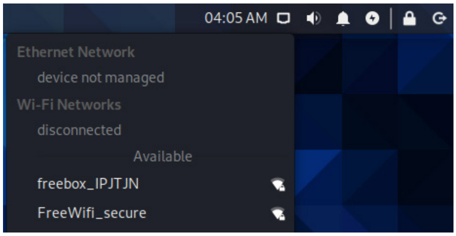
As Kali Linux is based on Debian, you will not be lost; many things are similar.

Connect to your network

We are only talking here about networks in DHCP. If you have to define a fixed IP address, look at the following step.

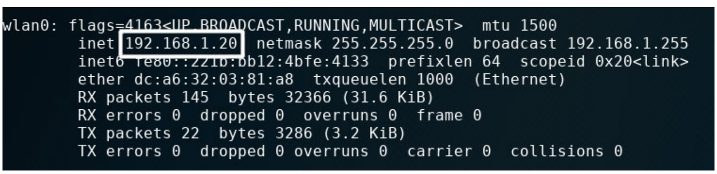
Ethernet:
Just connect the RJ45 cable to your Raspberry Pi and wait a few seconds for an IP address to be assigned to it, as there is nothing else to do.

Wi-Fi:
On the Kali desktop, click on the network icon at the top right, and choose the SSID of your Wi-Fi network.
Type the password of your access point and wait a few moments.



Get your current IP address:
Whatever your connection mode is, you can retrieve the IP address obtained with the ifconfig command.

The addresses are indicated on the second line of each interface, after the keyword “inet”



eth0 = Ethernet, wlan0 = Wi-Fi

I advise you not to activate both simultaneously, even if it works, I had problems response time problems, probably a problem of routing (I didn’t look any longer, but after disabling the Wi-Fi I had no problem).

Set a static IP address

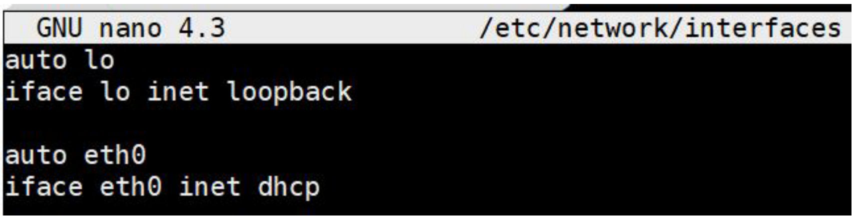
A static IP address will allow you to choose the IP address associated with your Raspberry PI, and therefore find it more easily later.

To set a static IP open the /etc/network/interfaces file:

```
sudo nano /etc/network/interfaces
```

You will see something like this:

```
auto eth0
allow-hotplug eth0
iface eth0 inet dhcp
```



Replace it with these lines:

```
auto eth0
allow-hotplug eth0
iface eth0 inet static
address 192.168.1.200
netmask 255.255.255.0
gateway 192.168.1.1
nameserver 8.8.8.8
```

Replace the IPs indicated by what fits your network.

Reboot your Raspberry Pi or unplug/plug in the network cable to update your IP.

You can do the same thing for your Wi-Fi connection by replacing eth0 with wlan0.

Update the system

As always, it's a good idea to start by updating your new system:

```
sudo apt update && sudo apt upgrade
```

Enable SSH and VNC

Now that we have a fixed IP address, it's time to make our Raspberry Pi accessible from another network computer.

Enable SSH

Usually SSH is installed by default.
If you don't have access, it's probably because you need to start the service:

```
sudo service ssh start
```

Install VNC

VNC will allow you to have access to a remote desktop on your Raspberry Pi.
To install it, follow these steps:

- Update your repository

  ```
  sudo apt update
  ```

- Install Tight VNC Server if needed (included on the latest versions)

  ```
  sudo apt install tightvncserver
  ```

- Start the service and set the password

  ```
  vncserver
  ```

You can now connect to your Raspberry Pi on Kali Linux with any VNC Viewer.
For example, on Ubuntu:

- *sudo apt-get install xtightvncviewer*
- *xtightvncviewer 192.168.1.200:1*

Remember that VNC is not a secure protocol, and if you use it at home it's ok, but in a more extensive network it is better to use it through an SSH tunnel for example.

Available tools

MAC Changer

Overview
A MAC address is a unique identifier for each network adapter. It depends on each manufacturer and it's often used to give access to a specific part of the network to restricted computers. A DHCP server can also assign the same IP to a MAC Address.
For example, you can configure your Wi-Fi network to whitelist your MAC address, and prevent anyone else from connecting to it.

MacChanger is a tool which allows you to do MAC address spoofing, i.e. to pretend to be someone else.

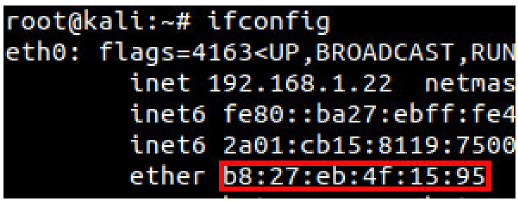
Usage

Install it if needed:

```
sudo apt install macchanger
```

See your current MAC Address:

```
ifconfig eth0
```



Disable your network card:

```
sudo ifdown eth0
```

Get a random MAC address:

```
sudo macchanger -r eth0
```

Set a specific MAC address:

```
sudo macchanger -m XX:XX:XX:XX:XX:XX eth0
```

Reboot to reset and get the standard MAC Address.

Wi-Fi hacking

Overview

I will come back to this in the last part, but you can use Kali Linux to hack a Wi-Fi password.

AirCrack-NG is one of the most known tools in Kali Linux.
It's a complete suite of software to test the wireless security of a network.
It provides tools for monitoring, attacking, testing and cracking Wi-Fi networks.

Usage

You must disconnect from Wi-Fi on your Raspberry Pi before starting.
Then, check that your network card is compatible (it is):

```
sudo airmon-ng
```

Start monitoring:

```
sudo airmon-ng start wlan0
```

Show wireless network available:

```
sudo airodump-ng mon0
```

And you are ready to go!
You will find more info in the last part of this chapter.

Brute force

Overview

Brute force is a password cracking method that involves trying passwords from a dictionary or other source continuously through all of the possibilities until it works.

Hydra is a tool that makes brute force very fast, and comes from a piece of Kali Linux software and supports a lot of protocols.

Usage

First, you will need to make a list of passwords and put this list in a file, like /root/passwords.txt (one per line).
Then you can try it. For example, I have decided to brute force SSH on my computer from the Raspberry Pi:

```
hydra -l root -P /root/passwords.txt -t 6 ssh://192.168.1.51
```

If I check in my /var/log/auth.log, I can see tries from the Raspberry:

```
Aug 12 15:55:37 PingusPC sshd[2481]: Failed password for root from
192.168.1.22 port 37226 ssh2
Aug 12 15:55:37 PingusPC sshd[2487]: Failed password for root from
192.168.1.22 port 37234 ssh2
Aug 12 15:55:39 PingusPC sshd[2482]: Failed password for root from
192.168.1.22 port 37228 ssh2
```

Packet analyzer

Overview

A packet analyzer (or sniffer) is a tool that can intercept traffic from the network and capture it to analyze it.

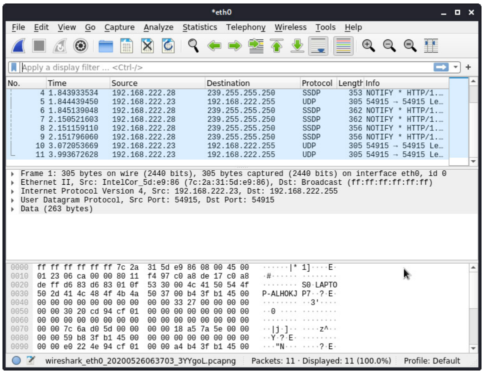
On Kali Linux, you can use Wireshark, which is the most used tool to analyze network traffic.
It's a graphical tool, but you can capture packets with tcpdump or something else, and then open it with Wireshark.

Usage

You can find the app in the Applications menu, under Sniffing and spoofing.

Start it and then go to Capture > Start
You will now see all packets from the network.
Click Stop when you want.

Then there are many features that you can use to filter or analyze what you have captured:



SQL injection

Overview

SQL injection is a technique for attacking insecure applications, including injecting code into user fields that are not protected.

This technique is mainly used to attack websites.
For example, if you replace a parameter of the URL, say ?user=yourname by something like ?user=yourname ' OR 1
If the field is poorly protected, the SQL query will be modified and will return all the data, not just those of your user.

On Kali Linux, the sqlmap tool allows the testing of SQL injection vulnerabilities.

Usage

Sqlmap is a straightforward tool to use.
You only need to indicate the URL of the page to be tested, something like this:

```
sqlmap -u https://www.domain.com/?p=123
```

Once you have found a security hole, it is possible to dig deeper with this tool to see what you can get. But the best thing to do is to fix it :-)

Vulnerabilities exploit

Overview
Metasploit is a tool that will allow you to validate vulnerabilities and use them. Metasploit enables you to automate the process of discovery and exploitation and provides you with the tools required to perform the manual testing phase of a penetration test.

Usage
Install Metasploit with *apt*:

```
sudo apt install metasploit-framework
```

Then you can start it in Applications > Exploitation Tools > Metasploit framework
This tool will initialize and start a terminal that will allow you to use this software.

For example, you can use *nmap* in the framework:

```
db_nmap -v -sV 192.168.1.51
```

You can also retrieve information about a known vulnerability, and try to use it.
Link here: **https://www.rapid7.com/db/vulnerabilities**

```
db_rebuild_cache
search CVE-2018-9864
use exploit/folder/folder/name
```

Replace the search parameter with your vulnerability ID and use the exploit path displayed in the search results.

Example: Hack a Wi-Fi network

Aircrack introduction

I already introduced this in the last part, so I will be quick here.

AirCrack-NG is a suite of tools used to hack Wi-Fi networks, or at least to test their security.
AirCrack-NG offers tools to test, monitor, attack and crack Wi-Fi networks.

In this part, we'll see how to use it step-by-step to:

- Turn your wireless card into monitor mode
- Scan all Wi-Fi networks nearby
- Listen to a specific target to get needed packets (handshakes)
- Brute force handshakes data to find the password

If you are using Kali Linux, everything is already installed on the first boot.
On other systems you have to install it manually.

Set your wireless card in monitor mode

The first step is to turn your wireless card into monitor mode.
This mode allows you to see all networks around you and listen for handshakes:

- Use the ***airmon-ng*** command a first time to display your wireless card(s)

  sudo airmon-ng

```
root@kali:~# airmon-ng

PHY     Interface       Driver          Chipset

phy0    wlan0           brcmfmac        Broadcom 43430
```

  Here, I have only one card named wlan0
- So, we can start ***airmon-ng*** with the interface we just found:

  sudo airmon-ng start wlan0

- On your first try, you'll get errors about processes interfering with the monitor mode.
  You have to kill them before moving forward.

  airmon-ng offers a command to kill them all easily:

  sudo airmon-ng check kill

- Then start again:

  sudo airmong-ng start wlan0

- Enter the ***airmon-ng*** command again to see the new interface:
  sudo airmon-ng

```
root@kali:~# airmon-ng

PHY     Interface       Driver          Chipset

phy0    wlan0           brcmfmac        Broadcom 43430
phy0    wlan0mon        brcmfmac        Broadcom 43430
```

Scan for Wi-Fi networks

Once your wireless card is ready, we can move to the next tool: ***airodump-ng***
Airodump-ng allows you to scan Wi-Fi networks to find your target.

Use the following command to start the scan:

sudo airodump-ng wlan0mon

You'll get a screen like this:

```
CH  6 ][ Elapsed: 42 s ][ 2019-03-08 05:31

BSSID              PWR  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH E

46:8A:AE:33:74:4E  -46      105        0    0   4  54   OPN              o
34:8A:AE:33:74:4E  -46      105        2    0   4  54   WPA2 CCMP   PSK  L
00:24:D4:70:0A:7A  -59       21        0    0  12  54   OPN              F
00:24:D4:70:0A:79  -60       16        3    0  12  54   WPA2 CCMP   PSK  F
00:24:D4:70:0A:7B  -60       27        0    0  12  54   WPA2 CCMP   MGT  F
E4:5D:51:EC:6D:BE  -67       37        0    0  11  54   WPA2 CCMP   PSK  S
6A:5D:51:EC:6D:BD  -68       31        0    0  11  54   WPA2 CCMP   MGT  S
FA:86:F2:E1:5A:23  -69       44        0    0  11  54   WPA  CCMP   MGT  F
FA:86:F2:E1:5A:22  -70       42        0    0  11  54   OPN              F
FA:86:F2:E1:5A:21  -68       34        0    0  11  54   WPA2 CCMP   PSK  <
6A:5D:51:EC:6D:BF  -64       34        0    0  11  54   OPN              S
46:8A:AE:35:34:9E  -72       52        0    0   1  54   OPN              o
34:8A:AE:35:34:9E  -71       41        0    0   1  54   WPA2 CCMP   PSK  L
FA:86:F2:E1:5A:20  -73       11        0    0  11  54   WPA  CCMP   PSK  f
92:F5:F3:54:39:68  -75       34        0    0  11  54   WPA  CCMP   PSK  f
92:F5:F3:54:39:69  -75       32        0    0  11  54   OPN              F
36:89:E4:F1:5F:73  -77        6        0    0   9  54   WPA  CCMP   MGT  F
36:89:E4:F1:5F:71  -78       21        0    0   9  54   WPA2 CCMP   PSK  <
7C:26:64:CE:84:E0  -79       33        1    0   1  54   WPA2 CCMP   PSK  B
36:89:E4:F1:5F:72  -78       16        0    0   9  54   OPN              F
72:5D:51:9D:D0:85  -79       58        0    0   6  54   WPA2 CCMP   MGT  S
E4:5D:51:9D:D0:86  -78       35        4    0   6  54   WPA2 CCMP   PSK  S
72:5D:51:9D:D0:87  -82       36        0    0   6  54   OPN              S
14:0C:76:F4:8D:1D  -82       23        0    0  11  54   OPN              F
14:0C:76:F4:8D:1C  -81       24        0    0  11  54   WPA  CCMP   PSK  F
62:5D:51:FA:66:DD  -81        3        0    0  11  54   WPA2 CCMP   MGT  S
14:0C:76:F4:8D:1E  -81       13        0    0  11  54   WPA2 CCMP   MGT  F
68:A3:78:85:AF:32  -81       23        0    0   1  54   WPA2 CCMP   PSK  F
36:89:E4:F1:5F:70  -80       25        0    0   9  54   WPA  CCMP   PSK  f
68:A3:78:85:AF:34  -81       21        0    0   1  54   WPA2 CCMP   MGT  F
68:A3:78:85:AF:33  -82       21        0    0   1  54   OPN              F
7C:26:64:E5:98:F0  -82       12        0    0   6  54   WPA2 CCMP   PSK  B
28:9E:FC:F8:55:20  -83        5        0    0   1  54   WPA2 CCMP   PSK  B
E4:5D:51:FA:66:DE  -83        4        0    0  11  54   WPA2 CCMP   PSK  S
68:A3:78:5F:A6:2A  -84        2        0    0   6  54   OPN              F
E4:9E:12:BB:57:DC  -83        2        0    0  10  54   WPA2 CCMP   PSK  F

BSSID              STATION            PWR   Rate    Lost    Frames  Probe
```

Each line is a Wi-Fi network around you.
You're close to the first networks in the list and you can see their channel in the CH column.

Just below the Wi-Fi networks list, you can see the stations detected and to which network they are connected.
To collect data about a target, we need some active stations.

Choose one target

The first thing to do is to choose one target.

A target is a Wi-Fi network (one line in the list), preferably with a few active devices on it.

In this lesson, you will choose your network, anything else is forbidden at the moment.
To filter the list to display only one Wi-Fi network, follow these steps:

- Stop the scan networks command with CTRL+C
- Then use this command to scan only one network and write data in a file:

  ```
  sudo airodump-ng wlan0mon --bssid XX:XX:XX:XX:XX --channel X
      --write airodump
  ```

  Replace XX by the BSSID mac address and X by the channel number.
  airodump is the filename where you'll collect data; we'll use it later
- You'll get a filtered list like this:

  ```
  CH  4 ][ Elapsed: 36 s ][ 2019-03-08 05:42

  BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH E

  34:8A:AE:33:74:4E  -48   0      338       78    1   4  54  WPA2 CCMP   PSK  L

  BSSID              STATION            PWR   Rate    Lost    Frames  Probe

  34:8A:AE:33:74:4E  50:C7:BF:DD:43:95  -28    0 - 0e     0        1
  34:8A:AE:33:74:4E  B8:27:EB:CC:9E:FE  -44    0e- 0e     0       14
  34:8A:AE:33:74:4E  88:E9:FE:42:3A:6A  -63    0 -24     38       34
  ```

- Here I have one wireless network and 3 devices connected, it's perfect

Now you need to wait until one device reconnects to the Wi-Fi network.
As it's your network, you can disconnect and reconnect your smartphone and see what happens.

In real life, hackers are sending packets to force a device to reconnect.
You can do this with *aireplay-ng* like this:

```
sudo aireplay-ng wlan0mon --deauth 10 -a XX:XX:XX:XX:XX
```

Replace the XX by the BSSID of the network you are targeting.
Run this command in another terminal (or another SSH session).
You can't stop the airodump command or you won't get the result.

If everything is going well, you'll see a "WPA Handshake" message at the top of your scan window:

```
CH  4 ][ Elapsed: 7 mins ][ 2019-03-08 05:49 ][ WPA handshake: 34:8A:AE:33:74:4E

BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH ESSID
```

Your attack was successful!
You now get one handshake in the file.
You can crack it to get the password.

A handshake sample is like an encrypted password.
You can't decrypt a password hash, but you can encrypt words to see if the result is the same encrypted hash.
In the next steps we'll try to find a password by doing this.

Get passwords dictionaries

With new security on Wi-Fi networks, it's no longer possible to find the password directly from the handshake data.
You need to use dictionaries to try many words and finally find the corresponding password.

So, the first thing is to get those dictionaries.
Here is a list of links you can use to get them:

- **AirCrack NG Website**: https://raspberrytips.com/dl/pass1
- **Crack Station**: https://raspberrytips.com/dl/pass2
- **Other links on Reddit**: https://raspberrytips.com/dl/pass3

Beware, some are big. On Raspberry Pi you don't have unlimited disk space.
It all depends on the size of your SD card.

To download them, you can use *wget*, for example:

```
wget https://mirrors.edge.kernel.org/openwall/wordlists/passwords/password.gz
```

Then you need to extract them, depending on the extension, with gunzip, unzip or unrar.

Sometimes, you can also generate your dictionary (I think that John The Ripper offers this feature: https://www.openwall.com/john/).

Depending on the SSID name, you can probably guess what type of password is set by default (each ISP/router has a different default format).
Most of the time, people don't change the default password.

Hack the password

The last step is to try cracking the password with aircrack-ng and your dictionaries:

- You now have to use this command to start cracking the Wi-Fi password:

  `sudo aircrack-ng airodump-01.cap -w password`

  airodump-01.cap is the file previously generated by airodump with the WPA Handshake.
  password is the name of my dictionary file
- Then `aircrack-ng` will try all the passwords from the file
- If you have a strong password, `aircrack-ng` won't find it.
- You can add it in to the dictionary file to see what happens when it's found

```
                              Aircrack-ng 1.5.2

[00:00:01] 647/646 keys tested (528.09 k/s)

Time left: 0 seconds                                       100.15%

                KEY FOUND! [ 55D514757C5DED739162EA56C5 ]


Master Key     : F4 85 40 EA 6D C3 5E A5 39 E0 F4 27 A8 E2 D6 67
                 CC 7D 2C AB F7 DE 8A 64 BB 20 14 A0 0F F3 DB 56

Transient Key  : B0 E6 1D D1 50 13 F6 41 26 CE E5 CB C3 FF FE 4C
                 7C 2C 26 85 46 06 70 09 C9 99 1A 33 DC 09 90 22
                 5A 7B 67 12 9F 09 37 FF 86 80 FB 29 23 A8 7D A7
                 30 1F 1D 7F 08 5E C0 07 70 BD E2 65 B0 33 85 80

EAPOL HMAC     : EB 7D FD 39 32 E1 5B 91 6D 76 FD 62 5D 44 3C 22
```

Conclusion

You now have a good overview on how you can use a Raspberry Pi for security projects.
I also showed you a practice case with a tutorial on how hackers go about hacking Wi-Fi passwords.

Even if it seems difficult or long, you should know that hackers will generally not use a Raspberry Pi for the last step (hacking the password).
Depending on the target, they can transfer the cap file to a computer or even a super-calculator to try giant password lists, and finally find it much faster than you think.

I will not give you any homework today, but I don't doubt that if you are interested even slightly by the computer security aspect, you will play with Kali Linux and try what inspires you the most :)

Day 26:
Other Raspberry Pi Operating systems

Introduction

In this book, I mainly write about what you can do on Raspberry Pi OS.
I sometimes introduce other systems, but I come back to Raspberry Pi OS each time.

Why?
Because Raspberry Pi OS is the official operating system, probably the most compatible with your Raspberry Pi.
And also because Raspberry Pi OS is a Debian-like system. Debian is the most used operating system in the Linux world (including Ubuntu and other variants) and can bring your skills to other devices than just a Raspberry Pi.

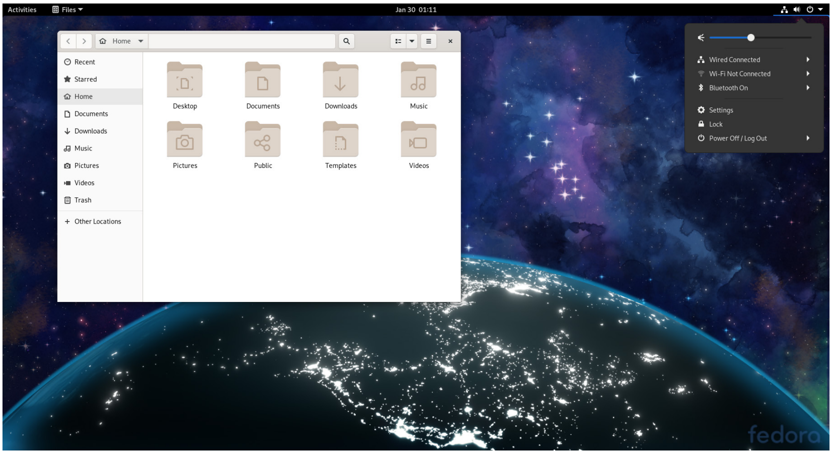
But if your goal is to become a Raspberry Pi master (and you are really close to that), you need to know that there are many other operating systems for Raspberry Pi out there.
This chapter is here to give you an overview of my favorite ones at the moment.
They are up to date, easy to use and work very well on Raspberry Pi.

I can't give you a complete tutorial for each one, the goal is just to introduce each, give you useful links and a few tips when needed.
For most of these systems, there is a free tutorial available on RaspberryTips that you can read to go further and install it, I will give you the link each time.

There is no ranking in this list; I sorted it by category to make it easier to read.

Desktop

Fedora



Presentation

Fedora is a popular Linux distribution based on RedHat, with around 1.5 million users at the time of writing.
It has been around since 2003, and is the open-source alternative to RedHat.
As it's not based on Debian, you may find some differences in the commands (for example dnf is the equivalent to apt).

Fedora exists in three main versions:

- **Fedora Workstation**: for desktop users, with a graphical interface and preinstalled software like LibreOffice or Firefox
- **Fedora Server**: release dedicated to servers, with a short life cycle and tools for sysadmins
- **Fedora IoT**: release intended especially for Internet of Things

We have seen a development cycle of one to two releases each year since the beginning.
They use a simple version number: 1, 2, 3 ... 29, 30, 31 and there is also a less public code name.

Links

For each system, I'll give you a few links to help you get started, and find help if needed:

- Official website: **https://raspberrytips.com/go/fedora**
- Download page: **https://raspberrytips.com/dl/fedora**
- Official wiki: **https://raspberrytips.com/wiki/fedora**

Installation

SD card preparation

You can download the Fedora version you prefer, and flash it with Raspberry Pi Imager or any other tool. In Raspberry Pi Imager, choose "Use custom" in the OS list, and select the Fedora image file on your computer.

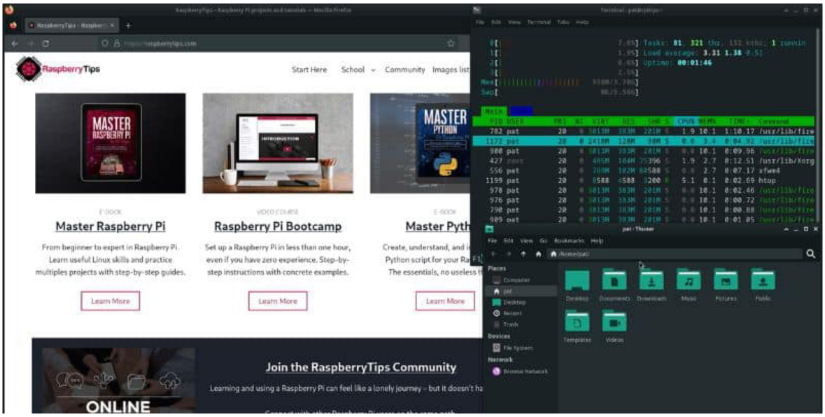
Extra tips

The configuration is easy, no matter which version you choose:

- In the desktop version, you can configure everything from the interface, so I don't have anything to add
- And in the minimal version, Fedora will ask you to configure all the settings on first boot

If you need more help, on how to install Fedora (minimal mainly), I have a detailed tutorial here: **https://raspberrytips.com/tuto/fedora**

Manjaro



Presentation

Manjaro is introduced like a fast and user-friendly Linux distribution.
It's based on Arch Linux, a system we don't see too much recently on Raspberry Pi due to the complex installation process (compared to other distributions).

So Manjaro brings fresh air on Raspberry Pi and will delight fans of Arch Linux.

Manjaro released its first version in 2011 and was in the beta stage until 2013.
So, it's pretty new in the Linux history, but old enough to offer us a stable operating system.
The official goals of Manjaro have always been to bring usability and accessibility and to make it operational directly after the installation.
This operating system is available in many versions, with a lot of compatible hardware. But on the Raspberry Pi we have the choice between three versions: XFCE, KDE or Minimal.

Links

- Official website: https://raspberrytips.com/go/manjaro
- Download page: https://raspberrytips.com/dl/manjaro
  Browse to Editions > ARM > Raspberry Pi to get the ARM images
- Wiki: https://raspberrytips.com/wiki/manjaro
- Forum: https://raspberrytips.com/forum/manjaro

Installation

SD card preparation

The Manjaro image is ready to use, you can flash it with Raspberry Pi Imager (select "Use custom") and boot your Raspberry Pi right away.

Extra tips

On the first boot, Manjaro shows up a quick wizard to configure the basic settings.
Answer the questions and continue.
You'll create a user and password, there are no default values.

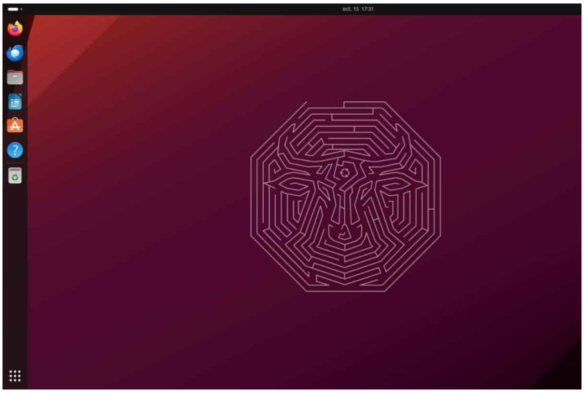
The only issue I had with Manjaro was the wireless connection not working on Raspberry Pi 4.
But it may be a fault with my network, or it may have been fixed since that time.

Finally, if you choose the minimal version, note that the package manager is "pacman":

- Search a package name: `pacman -Ss <search>`
- Install it: `pacman -S <package>`
- Update the system: `pacman -Syu`

If you need more help, visit the RaspberryTips website, here:
https://raspberrytips.com/tuto/manjaro

Ubuntu



Presentation

Ubuntu Desktop is probably the best alternative to Raspberry Pi OS for desktop use on a Raspberry Pi. It's not as optimized as the official distribution, but it works really well and has a nicer interface.

Links

- Official website: **https://ubuntu.com/**
- Download page: **https://ubuntu.com/download/raspberry-pi**
- Wiki: **https://wiki.ubuntu.com/ARM/RaspberryPi**
- Forum: **https://ubuntuforums.org/**

Installation

SD card preparation

The easiest way to install Ubuntu with Desktop on a Raspberry Pi is to use Raspberry Pi Imager and find the latest image available in the OS list. It can then be flashed on any storage media (SD card or USB).

You'll find it under "Other general-purpose OS" in the OS list.

Extra tips

- Unlike other distributions, Ubuntu is not fully installed and ready to use once it's flashed on the SD card. There is an extra step with a system configuration wizard on the first boot. And only after that step is the installation completed.
- Ubuntu will automatically check for new updates regularly, so you'll occasionally get a popup (like with Raspberry Pi OS). Try to do the updates regularly.

Once you get used to the interface and the default applications, you can really start to enjoy your new system. One last thing, you may need is some guidance to install new applications on Ubuntu, it's a bit different from Raspberry Pi OS.

You can find step-by-step tutorials about Ubuntu on the website:
Desktop version:
https://raspberrytips.com/install-ubuntu-desktop-raspberry-pi/
Server version: **https://raspberrytips.com/ubuntu-server-raspberry-pi/**

Server / Minimal

Many systems offer several versions, including a minimal version (Raspberry Pi OS, Manjaro etc).
But DietPi is the only one I like now as an alternative to Raspberry Pi OS Lite.

DietPi

Presentation

As its name suggests, DietPi is a lightweight distribution for your Pi.
It's working on a Debian OS you can configure to start with a minimal operating system (less than 200M once installed).

As it's based on Debian, DietPi is similar to Raspberry Pi OS Lite.
The two main advantages are the regular updates and the better performance (it includes far fewer packages than Raspberry Pi OS Lite).
It's perfect for a project with basic needs, for use with an old Raspberry Pi model.

Links

- Official website: **https://dietpi.com/**
- Community: **https://dietpi.com/forum/**

Installation

The installation and configuration processes are similar to what you already know with regards to Raspberry Pi OS Lite, so I don't have many things to add.

The default login is "root" and the password "dietpi"

Once logged in, a wizard shows up to help you with the configuration process. Here you can do the basic configuration and install additional packages (you can even install a desktop environment!).

And here are a few configuration commands you need to know:

- To go back to the installation wizard that I showed you at any time, you can use:

  ```
  dietpi-software
  ```

- To only see and edit the configuration part (raspi-config equivalent), just use:

  ```
  dietpi-config
  ```

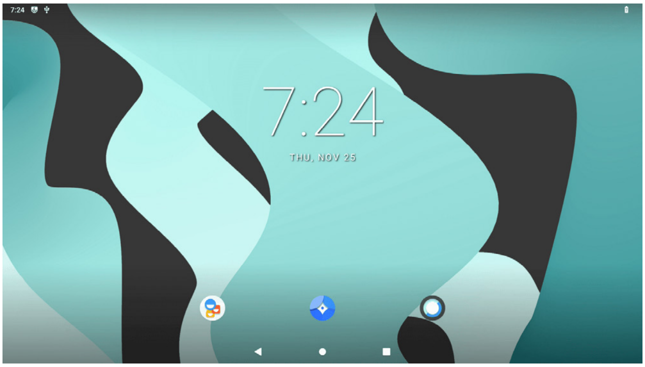
  Here you'll find all the basic things you saw during the installation wizard: network, display, audio, etc.
- To check if a new DietPi version is available for your Raspberry Pi, use:

  ```
  dietpi-update
  ```

If you need more detail about the installation and configuration of DietPi on a Raspberry Pi, check my complete guide here: https://raspberrytips.com/tuto/dietpi

Others

Android



Presentation

Installing Android on Raspberry Pi has always been a challenge. But thanks to the work of several teams of developers, the task has become easier over the years.

Android is a Google product designed for smartphones and tablets. Only partners (manufacturers) can get the full version and install it on their products after being approved by Google. Raspberry Pi support is clearly not a priority for Google.

The second problem is that while the core of the Android system is open source, key applications like the Play Store and other Google apps are proprietary. This means that you won't find a ready-to-use Android image with everything included.

Hopefully, this isn't just a problem for Raspberry Pi owners, but for many other devices as well. That's why projects like Emteria and LineageOS have been created. Some people are also interested in an open-source version of Android and are working hard on it.

For example, LineageOS is a free and open-source version of Android designed for tablets and smartphones. It comes with a few basic applications (browser, calendar, calculator, etc.), but it doesn't come with any Google applications or useless pre-installed software.

Half of the problem was solved, and that was the starting point for another developer, who created a custom build of LineageOS, working on Raspberry Pi. We also have other developers working on a redistributable version of Google applications, that can be installed on top of LineageOS.

In short, all of this is very complicated, but there are now solutions to get Android running on a Raspberry Pi, often with a very recent version, and the ability to install all the applications we need.

Installation

Due to the number of development teams involved and the traditional Android updates from Google, the installation process to get Android working on a Raspberry Pi changes regularly (I think I update my tutorial at least once a year with major changes).

So I won't take the risk of including the step-by-step installation process in this book. I highly recommend that you check out the tutorial on the website for a better chance of getting the latest version:
https://raspberrytips.com/android-raspberry-pi-4/.

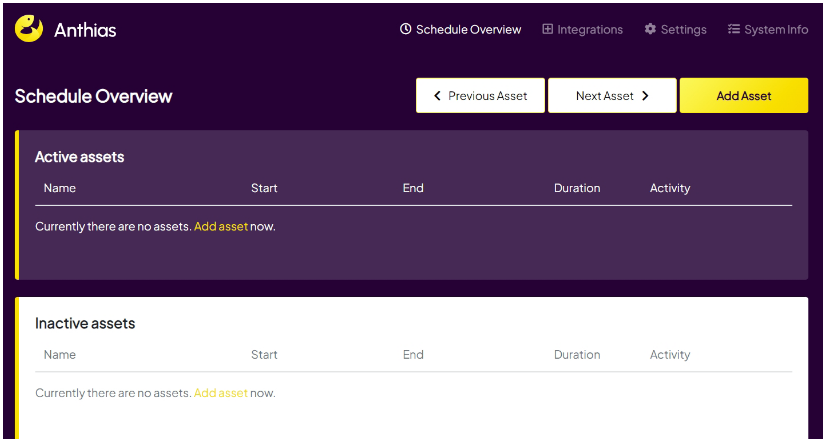
To give you an idea, here are the main steps of this installation:

- Download the latest custom image from KonstaKang.
  It's the custom build of LineageOS for the Raspberry Pi I was referring to.
- Install it on a new SD card with Raspberry Pi Imager or Balena Etcher.
- Download the Google Apps package from another site (NikGapps for example).
- Once the system is installed, reboot in recovery mode to add the Google Apps to the system.

After a final restart, you should have access to something very close to what you're used to on smartphones and tablets. With access to the Google Play Store, any application should work as usual.

By the way, it's also possible to follow a similar process to get Android TV on a Raspberry Pi, which can be a good alternative to Kodi.
More details here: **https://raspberrytips.com/android-tv-on-raspberry-pi/**

Anthias by Screenly



Presentation

This one is a bit particular, and not the most known :)
The goal of this system is to provide a digital signage solution on a Raspberry Pi.

Screenly is another company.
They started their activities in 2011, a few months before the Raspberry Pi official release.
At first, the main goal was to build a cost killer solution in the digital signage market.

They started to build the open-source solution (probably to test it and improve it quickly), and they quickly identified a need in the market for a cloud solution at a low cost.
That's why they expanded their offers to add hosting and support plans.

Today, Screenly allows thousands of stores to use digital signage at low cost.
The software part is fairly easy to use.
Everything is in your browser, you can select a screen, add a new "asset" (image/video/...), and schedule this for the day or for an event.

If you want to try Screenly for free, there is an open-source solution called "Anthias".
It's free forever, even if you're a business, so it's a good way to try it out.

For larger companies with many stores, you might want to look at the "Screenly Pro" solution to simplify remote screen provisioning and management. But for a first try, the open-source system is fine.

Links

- Official website: https://www.screenly.io/
- Open-source website: https://anthias.screenly.io/
- Help: https://support.screenly.io/

Installation

SD card preparation

Anthias is now included in the OS list in Raspberry Pi Imager. You'll find it under "Other specific-purpose OS". There is a version for most Raspberry Pi models, so this is the easiest way to try it out.

Follow the same steps as usual to get it flashed to your SD card.

Extra tips

- If you need to use a wireless connection, a screen will give you the instructions to follow on the first boot.
  You'll get access to a temporary SSID and a web interface to configure it
- Then, everything can be managed from the web interface.
  You can remove the default assets, and add your own content for this screen (web page, image, etc)

For more details about Screenly, you can go to **https://raspberrytips.com/tuto/screenly** and read the tutorial.

Homework

The goal here was to get an overview of what is possible on Raspberry Pi.
But to master this, you have to try it yourself, and see how you can use another system to your advantage in your future projects.

If you don't have time now to test all the operating systems in this list, it's not a big deal:

- **Pick at least one system**, the one you prefer, or that inspires you.
  You may also think of a specific project you have, and see the potential in this system
- **Install it on your Raspberry Pi.**
  You can use the links provided to find help, or follow the corresponding tutorial on the RaspberryTips website
- **Try to go a bit deeper than just booting on it.**
  Spend a bit more time to find the pros and cons of this system

I hope you'll find a system that you really like today :)
If those presented here are not enough for you, you my up-to-date list of operating systems available on Raspberry Pi here:
https://raspberrytips.com/best-os-for-raspberry-pi/

Day 27: Shell scripting on Raspberry Pi

Introduction

On day 19 and 22 you learned how to code in Python, because Python is an essential part of the Raspberry Pi OS system.
But on Linux you have many options, and you'll often make your choice depending on your skills or what looks like the easier way to do the task in the moment.

Today, I want to add a second option for you (if you are a beginner in coding), it's what I call "Shell scripting."
Basically, it's using mainly Linux commands to achieve your goal.
As you already know many Linux commands, and because Shell scripting is a good skill to put in your resume, I think it's important to have a chapter about that in this book, even if it's not specific to Raspberry Pi.

Let's see what it looks like!
I will go a little faster than in chapter 19 as you already know how to code, so I hope that Python was OK for you :)

Shell scripting introduction

Differences with Python

As you already know the basics of Python, I will take that to start, and give you a comparison.

In Python, we have basic functions already available for your code, but we often need to use libraries to interact with the hardware and the system.

In Shell, you can use the system command directly, you don't need any libraries.
The shell you use provides a syntax and basic elements we will see later (variables, conditions, loops, etc), but everything else is within the Linux commands available on your own system.

If, for example, you install php on your Raspberry Pi, you can have a condition in your script that runs a PHP script to log an event in a database.

Scripting on Linux

I told you in the introduction that Shell scripting is a good skill to add to your resume.
It's good because you can use it on any Linux system, not just Raspberry Pi.
So if, further down the line, you get a job as a system administrator or something equivalent, you will be able to use this new skill on the company servers.

I think the major part of a Shell script is compatible with any systems under Linux, whatever the distribution.
So be attentive today, and you will learn something that you can use for a long time.

Shell or bash?

It's a question you may have if you are experienced and talk about bash rather than shell.
We are not here to debate about the good wording to use, but rather to learn new things.

But to help you with this idea here are a few useful bits of information.
Basically, shell is a program on Linux, managing interactions between the user and the system. You can use different shells on Linux, and "bash" is one of them, and we will use it here. So, if you are used to "bash scripting" you are right too, it's just more specific than "shell scripting."

Anyway, let's code, whatever the name is :)

Instructions you need to know

Let's start with the rules you have to follow in your scripts: the syntax.

Basic syntax

Here is the basic syntax to start a script:

```
#!/bin/bash
#My script purpose
```

I recommend starting any script like this, even if it's not mandatory:

- On the first line, you tell the system how to execute this script, or which shell to use.
  In this chapter, I will use bash, but you have other options like sh, zsh, csh, etc.
- The second line is a comment.
  You can add a comment on a separate line, or at the end of a line, always starting with a #
  It's good practice to note at the beginning what the goal is of the current script

To create your first script, you can use Nano, and write the lines above in it.
I will show you in the second part how to run it.
If you want, you can create a dummy script with all the syntax rules I give you here, just to keep it on your system as a quick reference point for later.

Display a message

To display a message on the screen, the command is easy, and you may already know it.

It's echo, a Linux command that you can even use in the terminal.

The traditional Hello World looks like this:

```
#!/bin/bash
#Hello world script

echo "Hello World!"
```

Quotes are not mandatory, but I recommend taking this habit up to keep it easy to read.

Also, a quick note, there is no marker at the end of a line in a shell script.
You do not need to add ";" or any other syntax delimiter, the line return is enough.

Define variables

The same way as in Python, you will often use variables to store data during the script execution.
The syntax is specific, but nothing complicated.

Here is an example:

```
#!/bin/bash
#Hello name

firstname='Patrick'
echo "Hello $firstname"
```

So just pick a name for your variable and define the content as in many other languages.

Execute a Linux command

Defining variables statically is cool, but generally you will have a dynamic input here, for example by running a Linux command and storing the result in a variable to use it later in your script.

Here is an example:

```
#!/bin/bash
#Find big files in /home/pat

#No matter the command I use here,
#just something returning a value
nbfiles=`find /home/pat -size +100M | wc -l`

echo "There are $nbfiles files over 100M in /home/pat"
```

Note the orientation of the quotes, it's important to use these ones when you call an external command with many arguments.

If you use an incorrect syntax, your script may crash or execute it (often with incomplete arguments), but never store the result in your variable.
On a French keyboard, the shortcut is "Alt Gr + 7", and on a qwerty keyboard, it should be the first key, on the left of the "1"

Conditions

Once you know how to define a variable, you can start to play with conditions.
As with python, basic conditions like if or else are available.
You can use them to execute a portion of code only in a specific case.

As you already know the theory about this, I will just give you the syntax (and explain it later).
Here is the first example, it will not work, but just to give you the syntax :

```
if  [ condition1 ];
then
   echo "Condition 1 is true"
elif [ condition2 ];
then
   echo "Condition 2 is true"
else
   echo "None of the above"
fi
```

So, the keywords to remember are: "*if*", "*elif*" and "*else*"
The condition is in square brackets, ending with a ";"
And finally, you end the condition group with "fi"

It's not obvious and you may need to give these a quick look after coding for months in Python or any other languages, but it's like that :)

I keep putting emphasis on the conditions, because the syntax is specific, and not intuitive.
In other languages, we are used to using operators like ">" or ">=", but in shell it's different.

Here is a list of the main conditions that you can use in your script:

Arithmetic conditions	
-eq	True if the two numbers are equal
-gt	True if the first number is greater than the second Ex: [3 -gt 1] is true
-ge	True if the first number is greater than or equal to the second Ex: [3 -ge 3] is true
--lt	True if the first number is less than the second Ex: [1 -lt 3] is true
-le	True if the first number is less than or equal to the second Ex: [1 -le 1] is true

File condition	
-f	True if the argument is a file on the system Ex: [-f /home/pi/test.txt]
-d	True if the argument is a directory on the system Ex: [-d /home/pi] is probably true on Raspbian
String conditions	
-z	True if the string is empty Ex: [-z “”] is true
==	True if the two strings are the same

For your information, you can invert all these conditions by adding a “!” before the condition
Ex: `[ ! 1 -gt 3 ]` is true
So, I didn’t put the negative forms of these conditions.

That’s it for the basics that you will probably use often, I don’t want to give you too much information here.
If you need more information on this topic, you can find the complete bash reference here: **https://raspberrytips.com/ref/bash**

Loops

The last thing we need to look at is how to build loops in your scripts.
I will show you here the two basic loops: for and while, with 2 options for each.
It should be enough for anything you may have to code, but you should know that other options are possible.

While

Infinite loop

The first one is the easiest one, but you may have to use it sometimes.
This kind of script will run until you stop it with CTRL+C

Here is what it looks like:

```
#!/bin/bash
#Infinity

while true
do
        echo "My loop is infinite!"
        sleep 1
done
```

Note the structure of the while loop: *while*, *do*, *done*.
And between do and done you have your code that will be repeated each time.

Condition

You can obviously put a condition at the beginning, and execute the loop while the condition is true.
Here is an example

```
while [ $load_average -gt 4 ];
do
        echo "Too busy, I need a break :/"
        sleep 5
done
```

For

Enumeration

I never use this option, but it's an important solution to have in mind.
You can run a bit of code several times, for a predetermined number of executions:

```
for i in {1..5}
do
        echo "Iteration $i"
done
```

The syntax is the same as with "while" : *for* / *do* / *done*.
You have many other options after the keyword "in"
I will show you one that is very useful now.

Command results

As you will mainly use scripts to run system commands, you will often want to analyze the results of a command.
Here is the syntax:

```
for line in $(YOUR-COMMAND)
do
        #do your job here
        echo $line
done
```

The syntax after the "*in*" keyword is a bit specific, but all the rest of it stays the same.

Shell scripting in practice

Now that you have all the basic syntax in mind, I will give you a few examples on how to use it in practice.
But first, I need to tell you how to run your first script.

Run your first script

Let's say you want to try the "Hello World!" script on your Raspberry Pi.
Here is what you need to do:

- Open your favorite text editor, let's say Nano (but you can use a graphic editor if you prefer):

  ```
  nano myscript.sh
  ```

- Write your code in it, or paste these lines:

  ```
  #!/bin/bash
  #Hello world script

  echo "Hello World!"
  ```

- Save and exit
- Then you need to add the execution permission to your script:

  ```
  chmod +x myscript.sh
  ```

- And finally run it!

```
./myscript.sh
```

- You can replace ./ by the script location if you are not in the same folder, for example:

```
/home/$USER/myscript.sh
```

Example 1: Disk usage alert

Here is my first example of what you can do with shell scripting in real life.
I'm using this script in all of my servers.
The goal is to send me an email if the disk usage becomes too high.

I will give you the code, and I will comment on it later.
The goal is not to understand everything, but to recognize the code syntax I gave you previously.

```
ADMIN="you@gmail.com"
ALERT=80 # disk usage threshold (80%)

df -HL | grep -vE '^Filesystem|tmpfs|svc|cdrom|udev|shm|Sys.' | awk '{
print $5 "; " $1 }' | while read output;
do
    usep=$(echo $output | awk '{ print $1}' | cut -d'%' -f1  )
    partition=$(echo $output | awk '{ print $2 }' )
    echo $partition
    if [ $usep -ge $ALERT ];
    then
        echo "Running out of space $partition ($usep%) on $(hostname)
         as on $(date)" | mail -s "$(hostname) DISK ALERT" $ADMIN
    fi
done
```

It's a script I'm copying without reading for many years, I have no idea from where it comes, I probably didn't write it at the beginning (I hope there is no copyright on this ^^).

So, you may recognize a few structures that you should already know:

- At the beginning, we define variables with the recipient email and the maximum disk usage threshold

- Then there is a more complex command.
  Basically we use *`df -H`* to get the disk usage, grep to filter the results and awk to select only the columns of interest
- This giant command goes into a while loop (not the way I show you, but it's not important)
- In this loop we create two new variables.
  One with the partition name (*`$partition`*), and the other with the disk usage for this partition (*`$usep`*)
- Finally, there is a condition using *`-ge`* (greater than or equal)
  If the disk usage is greater than or equal to the threshold, we send an email to the administrator

My goal here is not to tell you to use this script, as there are many methods to do the same thing (and probably a few better ones).
It's just an example of a script I use that can help you to understand how to use everything we have seen in the same script.

Example 2: Dump all databases (MySQL)

The second one is probably a cleaner one that I also use quite often.
The goal of this script is to create a backup of all the existing databases on your system.
If you already have some knowledge of how MySQL works, there is a command to export one database, but not all of them. You need to write one line for each database.
On a big server, it's just not possible, and it's not dynamic (if you create a new database and don't add it to the backup script, it doesn't work).

Here is what the script looks like:

```
#!/bin/bash

# Destiny folder where backups are stored
DEST=/home/pat/backup
CURRDATE=$(date +"%F")

# Hostname where MySQL is running
HOSTNAME="raspberrypi"
# User name to make backup
USER="root"
# File where has the mysql user password
PASS="yourpassword"

DATABASES=$(mysql -h $HOSTNAME -u $USER -p$PASS -e "SHOW DATABASES;" | tr
-d "| " | grep -v Database)

[ ! -d $DEST ] && mkdir -p $DEST

for db in $DATABASES; do
  FILE="${DEST}/$db.sql.gz"
  FILEDATE=

  # Be sure to make one backup per day
  [ -f $FILE ] && FILEDATE=$(date -r $FILE +"%F")
  [ "$FILEDATE" == "$CURRDATE" ] && continue

  [ -f $FILE ] && mv "$FILE" "${FILE}.old"
  mysqldump --single-transaction --routines --quick -h $HOSTNAME -u $USER
   -p$PASS -B $db | gzip > "$FILE"
  chown bacula:disk "$FILE"
  rm -f "${FILE}.old"
done
```

Take a few minutes to read and analyze the script.
It's easy to read, the syntax is clean (the condition syntax is a bit different, but not a big deal to understand) and really close to what we have seen at the beginning of this chapter.
Don't be put off by the mysql command. It could be any other command as it doesn't change anything.

If you need some quick help with this script, here is what it does in English:

- Define a few variables (backup folder, date, MySQL information)
- Run a command to list all the databases on the system and store the result in a variable: (`$DATABASES`)
- Make sure that the destination folder exists, or create it if needed
- Use a for loop to do the same thing for each database
- Make a few tests before creating the new backup file.
  Archive the old one if the file already exists for the same day
- Create the new backup file (`mysqldump`)
- Remove the old one (.old)

I found the exact same script here: **https://raspberrytips.com/script/mysql**
So it's probably coming from here originally!

Homework

Wow, if you were new to shell scripting, like for Python it could be a bit difficult to understand everything at first.
Take your time, read it again and again.
Even if I try to keep this book accessible to everyone, here it all depends on your level in scripting.

As I told you in the introduction, Shell scripting is useful on Raspberry Pi, but mandatory on Linux servers.
So don't rush this chapter and make sure to understand the concept.
You may not remember the exact syntax in one week, but you need to memorize the logic behind all of this.

As homework, I want to give you a short script to write:

- The main goal of this script is to display the multiplication tables
- Start by displaying one table, 9 for example.
  Display each line like this: 9 x 1 = 9
- Once you have done this, display all the tables
- Work on the display to improve the output look
- Make sure to add useful comments in your code

Good luck!

Day 28: Minecraft Pi Coding

Introduction

You probably need a break after the last chapter, and I will grant you that.
We will now play Minecraft, but not in the usual way :)

Did you already notice that Minecraft is included on Raspberry Pi OS Desktop?
It's not the full Minecraft version, but a specific version for the Raspberry Pi: Minecraft Pi.
The limited features make it difficult to play but bring about other interesting concepts and features.

As you know, Raspberry Pi was built to offer schools a way to teach coding to their students.
The main language is Python, and what we'll see today is that there is a way to play Minecraft (or at least interact with the game), from a Python script.
And that's the main focus of this chapter :)

Prerequisites

Raspberry Pi OS Desktop

Minecraft Pi is only available on Raspberry Pi OS Desktop.
So the first thing is to have that on your Raspberry Pi.

If you need to install it, you can check the "Day 4" chapter as a reminder.
Here are the steps you need to follow:

- Install Raspberry Pi OS Desktop
- Configure the network
- Update your system
- Enable SSH.
  It's not mandatory, but probably the easiest way to code in Python (=from your computer)

Once this is done, you can read the following.

Minecraft introduction

Just in case you have been living in a cave for the last ten years, I'll give a short introduction about Minecraft :-)

Minecraft is a sandbox game where the entire world is generated with blocks of the same size.

The player spawns in this environment and must survive while doing anything he or she wants.
The game includes a mix of exploration, building, crafting and combat.
Yes, you can also fight with passive or hostile mobs (like zombies or cows).

Here is what it looks like when you start the game on a Raspberry Pi:

Minecraft is a paid game.
On Raspberry Pi you get a free edition, but it's an alpha version.
If you are used to playing the PC version, you'll see many differences.

There is no configuration menu, no mobs and there are fewer blocks.
You can't even join a server or add a texture pack.
It's good for coding purposes, but that's all. If you want more functionality, you will need to buy the standard version (it's available for Linux, so it should work on Raspberry Pi).

Comfortable installation

Do you have a good setup? Like classic keyboard and mouse, desk and screen?
If not, I recommend starting with this.
In my case, I only have a mini Bluetooth keyboard with a touch pad.
It's not the easiest way to write code.

I've tried remote access to make this tutorial from my computer, but Minecraft's screen only displays on the Raspberry Pi; on any remote access software I only have the console.
It can help for typing the code, but it's not convenient for seeing what is happening in the game.
So, if possible, put your Raspberry Pi on a desk with a keyboard, mouse and screen.
This will make all the things below much easier.

My favorite setup is to have the Minecraft game on a screen (with mouse and keyboard), and to code in Python via SSH.
But you can do this in other ways if you prefer.

Reminders about Minecraft

Let's start with the basics, like how to use Minecraft the normal way.

Installation

On recent Raspberry Pi OS versions, Minecraft Pi is no longer installed by default.
If you want to follow this guide, you need to install it manually first.
The easiest way I found is to download and use "Minecraft Pi Reborn".

- Download the latest client package for your architecture (**direct link**).
- Add the execution permissions to anybody (right-click > Properties > Permissions).
- Double-click on it to start the game.

You can then follow the next steps as with any Raspberry Pi OS versions.
You may also need to install the Python library manually:
sudo apt install python3-minecraftpi

Create a new world

The first thing to do is to start the game and create a new world:

- Start the game (**Start Menu > Games > Minecraft Pi**)
- The window is a little buggy. You see the console behind, it's normal.
- Try to put the game on one side of your screen, you'll need space for the Python editor later
- To move the window, click on the blue console bar with the small cursor (yes, you have two…)
- Then click on "**Start Game**"
- You're now in the "**Select world**" menu. Click on "**Create new**" to create your world
- Wait a few seconds for the world to generate

You can now control your player with the mouse to see the world around you.
I'll give you all the controls keys later.

Minecraft game presentation

Minecraft exists in several game modes:

- **Survival**: you need to gather blocks and resources, craft stuff and survive during the night
- **Creative**: you get all the blocks you want for free and can't die
- **Adventure**: for map creators, you can't break blocks, but you can use levers and buttons
- **Spectator**: no interaction, you are flying all the time and you can go through blocks

On Minecraft Pi, as it's mostly an educational game, you are in the creative mode. You already have a sword and some blocks in the quick bar at the bottom of the screen.
It's possible to get more blocks (see the next paragraph), it's always sunny and you can't die.

In creative mode it's possible to break blocks in one shot, while in survival mode it depends on the tool you use (wood tools are slower than diamonds tools for example).

Minecraft controls

Here are all the controls you need to know:

- **Camera**: Move the mouse
- **Break block**: Left mouse
- **Place block**: Right mouse
- **Moving**: W,S,A,D (Z,S,Q,D on an AZERTY keyboard)
- **Jumping**: Space
  The auto jump is enabled when moving
  Double space enables the flying mode and then use space again to gain altitude
- **Access inventory**: E
- **Pause/Quit**: ESC

Try to move a little in your world to get used to the movements.

Hello world

Ok, let's start with the code part.

Start your Python editor

That's it! Now start your favorite Python editor.
I will explain and show you the results with Thonny, as it's the easiest way to start, but it works the same with nano or any other tool:

- Keep Minecraft open on one side and start Thonny
- Open the App Menu > Programming > Thonny Python IDE
- Then keep the new window on the other side

If you don't remember Thonny, the code goes in the big part under the menu (like any file editor).
And under it, you have the Python Shell to see what happen when you run or debug the code.

First code

In each new language, the first thing to learn is how to code the famous "Hello World!"
So, let's do this!

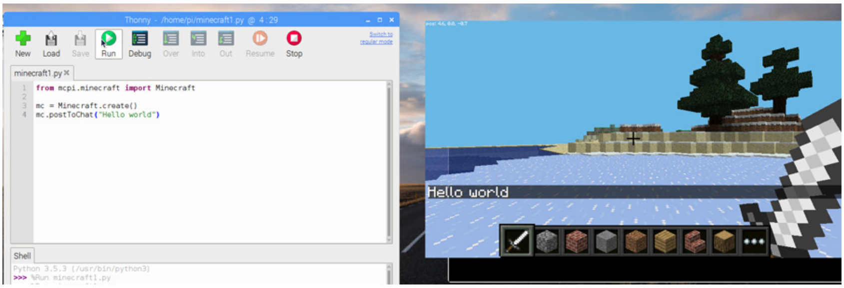
To start, type this code in Thonny:

```
from mcpi.minecraft import Minecraft

mc = Minecraft.create()
mc.postToChat("Hello world")
```

- Click on "**Save**" in the top menu
  Save it where you want (ex: /home/pi/minecraft1.py)
- Go back to Minecraft to see the game
  Then type "**TAB**" to free the mouse pointer
- In the Thonny top menu, click on "**Run**"

Check what happen in the Minecraft window:



Explanations

Let's see what we have done line by line:

- **First line:**

  ```
  from mcpi.minecraft import Minecraft
  ```

  In this line, we include the Minecraft library for Python.
  So, now Python knows how to speak to Minecraft.
  You must add this line in every code for Minecraft
- **Second line:**

  ```
  mc = Minecraft.create()
  ```

  In this one, we initialize the mc variable.
  This variable is the representation of the library in our code.
  We'll use this to do things in Minecraft

- **Third line:**

    `mc.postToChat("Hello world")`

    And the last one is to send a message to Minecraft.
    We use the mc variable defined previously and call the function postToChat from the Minecraft library.
    This function allows us to send the message between brackets within Minecraft itself

Good work, you know how to send a message in Minecraft from your Python code.

Change a block

Now, we can try something funny.
We can switch a block with another one, but first I need to give you more information about Minecraft.

Minecraft coordinates

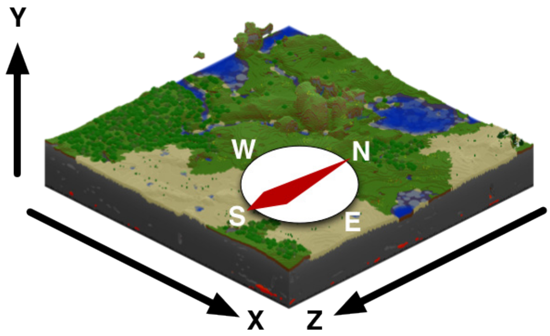
Minecraft uses coordinates to know the player position, and each block has a different position.
The player's position is visible in the top left of the Minecraft window:

Try to move and see how it changes
Each time you move one block away, one of these values is changing by 1 too.

So, my player position is defined by three values:

- **X:** 4.6 – It's my east/west position
- **Y:** 0.0 – It's my altitude
- **Z:** -0.7 – It's my north/south position

This picture should help you understand this more:



Try to move your player in the game again and see how the position changes in the indicator.
At any time, we know the player position, and we can thus use it to make funny things happen.

Minecraft blocks IDs

The second thing to know is how to define a block type in Minecraft.

When you open the inventory (E) you can see that Minecraft has many blocks available:

Basically, each block in the game has an ID to define it.
Stone is 1, Grass is 2, Dirt is 3, etc ...

So, if we want to set a specific block near the player, we only need to know its ID.

There is a website you can use to get the list of IDs:
https://raspberrytips.com/go/mcids

Choose a block from the website list, and note the ID; you'll need it later.

Change a block under the player

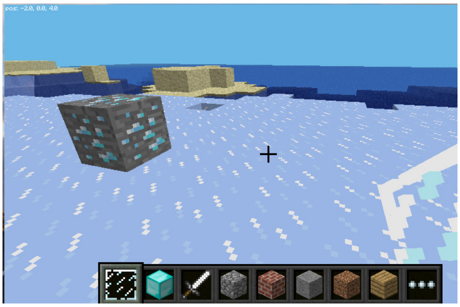
Now you'll type this code in Thonny to set a block at a specific position:

```
from mcpi.minecraft import Minecraft
mc = Minecraft.create()
mc.setBlock(0, 0, 0, 56)
```

It's almost the same code as before, the only difference is that we use the *setBlock* function instead of *postToChat*.
The parameters are: x, y, z (position) and the block id

I choose the diamond ore block (ID 56) and the 0/0/0 position (near my player position)
But you can choose what you want, try to set a position close to the player.

Here is the result in Minecraft:



I saw a diamond block appear close to me when I ran the code.

Yes nice, but the title says, "under the player", how do you do that?

Here is the entire code with a minor change:

```
from mcpi.minecraft import Minecraft

mc = Minecraft.create()
x, y, z = mc.player.getPos()
mc.setBlock(x, y-1, z, 56)
```

And that's it, I got a diamond block under my feet :-)
I used the `player.getPos()` function to get the player coordinates in real time.
And then I used them to set the block exactly where I wanted (with y-1 to set it just underneath).

I have taught you the basics, so now we can move some more complex things.

Other possibilities

In this part, I'll try to explain to you all of the possibilities there are when it comes to coding your Minecraft game.
I'll not teach you each function one by one, but I will give you a few functions you can use.

I didn't find an official API reference. I don't know if Mojang have this hidden somewhere, but I didn't find it.
The only page I found is the website linked to the "Adventures in Minecraft" book.
You can find it in my recommended books if you like this kind of tutorial:
https://raspberrytips.com/go/books

This website gives you all the functions and the parameters and explains what they do.
Click here to see this page: **https://raspberrytips.com/go/mcapi**

Teleport the player

The first thing I want you to try is to teleport the player to another place.
If you are near 0/0, you could go to 1000/1000 thanks to the Python code in just one second.

First try

Exercise:

If you feel ready, try to write this code yourself before checking the answer.

Which API function listed in the previous link will you use to do this?
What do you need?
Make a try without checking the answer part.

Answer:

This is easy, in fact we only need one function, this one:

.setPos(x,y,z)
"Moves the player to a position in the world by passing co-ordinates ([x,y,z])"

```
#set the players position as floats
mc.player.setPos(0.0,0.0,0.0)
```

This function does exactly what we need: teleport the player somewhere.

Here is my first code:

```
from mcpi.minecraft import Minecraft

mc = Minecraft.create()
mc.player.setPos(100,50,100)
```

If you are still unsure, try other numbers.
I put 50 for altitude because the mountains can be high, and I don't want to teleport myself under the ground.
If it spawns you 50 blocks up in the sky you might fall, but you're in creative mode, so you can't die.

Going further

Now try to use your Python knowledge to improve this script.
Let's say we want to send the player back to his initial position 10 seconds after the teleportation.

We already know how to get the player position and how to change it.
So, the only thing I should teach you is how to wait 10 seconds, right?
If you have a good memory, you already know how to do this, we have seen this in the Python chapters of this book.

Here is the entire code to do this:

```
from mcpi.minecraft import Minecraft
from time import sleep

mc = Minecraft.create()
x,y,z = mc.player.getPos()
mc.player.setPos(100,50,100)
sleep(10)
mc.player.setPos(x,y,z)
```

Click on "**Run**" and try it.

You'll see exactly what I said, teleport, then 10 seconds, then back to the first position.
In this code, we import the "*sleep*" function from the time library (classic library in Python).
We use all the functions already seen and *sleep(10)* to wait 10 seconds between the two teleportations.

Good job! Jump to the next exercise!

Build a house

Ok, in this part we want to build a house around the player by placing no blocks our self.
You already know how to get the player positions and how to place blocks near him, so it should be easy, right?

House foundations

To start, we'll just build the floor.
Let's say a 5×5 zone out of wood.

If you want to try it on your own, go ahead.
If not, here is my answer:

```
from mcpi.minecraft import Minecraft

mc = Minecraft.create()
x, y, z = mc.player.getPos()
mc.setBlocks(x-2, y-1, z-2, x+2, y-1, z+2, 5)
```

Find a flat place and run the code! Yes, I cheated :)

You can use ***setBlock*** 25 times if you want, but ***setBlocks*** (with an "s") allows us to define a zone to fill with one block type, like this:

		z+2		
		z+1		
x-2	x-1	x/y/z	x+1	x+2
		z-1		
		z-2		

So, we need to fill a zone from x-2/y-1/z-2 to x+2/y-1/z+2 to set the 25 blocks.

After running the previous code, you should get the floor of your house.
Try to build walls and roof alone, it's exactly the same thing.
If it's too easy, try to add some windows :)

Entire house

Here is the small house I built with my script:

Yes … I'm not bad with code but I'm not the best Minecraft builder :-)

The corresponding code is here:

```
from mcpi.minecraft import Minecraft

mc = Minecraft.create()
x,y,z = mc.player.getPos()

mc.setBlocks(x-2,y-1,z-2,x+2,y-1,z+2,5) #floor

mc.setBlocks(x-2,y,z-2,x-2,y+2,z+2,5) #wall1
mc.setBlocks(x-2,y,z-2,x+2,y+2,z-2,5) #wall2
mc.setBlocks(x+2,y,z-2,x+2,y+2,z+2,5) #wall3
mc.setBlocks(x-2,y,z+2,x+2,y+2,z+2,5) #wall4

mc.setBlocks(x-2,y+3,z-2,x+2,y+3,z+2,17) #roof

mc.setBlock(x-2,y+1,z,20) #window 1
mc.setBlock(x,y+1,z-2,20) #window 2
mc.setBlock(x+2,y+1,z,20) #window 3

mc.setBlock(x,y+1,z+2,0) #door 1
mc.setBlock(x,y,z+2,0) #door 2
```

Minecraft block IDs: 5 is wood, 17 is oak log, 20 is glass and 0 is air.
I use `setBlocks` to create the floor, walls and roof.
Then I use `setBlock` (without "s") to create the small windows and the door.

Once you understand the logic behind this script, you should be able to build what you want in Minecraft using Python.

Block interaction

The last things I want to show you are the block interactions.
There is a way to detect a player interaction with a block and trigger actions after this.

This code is difficult to guess for a beginner, so I'll give it to you directly and explain each line afterwards.

Here is the code I created for this topic:

```
from mcpi.minecraft import Minecraft
from time import sleep

mc = Minecraft.create()

try:
    while True:
        blockEvents = mc.events.pollBlockHits()
        if blockEvents:
            for blockEvent in blockEvents:
                mc.postToChat("Hit detected")
                x,y,z = blockEvent.pos
                mc.setBlock(x,y,z,56)
        sleep(1)
except KeyboardInterrupt:
    print("Stopped")
```

Here are some explanations for this code:

- This code turns each block you touch in the game (right-click) into diamond ore
- `pollBlockHits` is the main function we use in this code.
  You already know the other Minecraft functions
- As we don't know when the player will touch a block, we need to create an infinite loop: `while True`
  The try/except thing is mandatory when you create an infinite loop, to allow an exit (here it's CTRL+C)
- We add a timer in this loop so that we don't overload the Raspberry Pi: `sleep(1)`
- At the beginning of the loop, we call `pollBlockHits` to get all the blocks hit by the player
- If there is a result, we create another loop (for) to browse all results (`blockEvent`)
- Finally, for each event, we post a message in Minecraft (`postToChat`)
- And then we change the block (`setBlock`)

Copy this code in Thonny, run it and try it.
Try to change it as you want to be sure you understand my code.

If you're not familiar with programming, it can be difficult.

There are concepts independent of the Minecraft API here, and you'll not learn them all from one chapter.
We have already seen most of them in this book, but it may not be enough for you.

It's normal, as with many things, for programming to take a lot of practice to become natural.

Conclusion

There you go! You should already know a good part of the Minecraft API if you followed this chapter entirely.
And the most important thing would be that you understand the API reference.
That way you'll be able to create anything you want.

I will not give you homework today, as you have already practiced it at the end.
But you can take some time to experiment with the API functions that you are interested in.
For example, you could try to create a quiz in the Minecraft chat :)

Day 29:
Manage your Raspberry Pi from a web interface: Webmin

Introduction

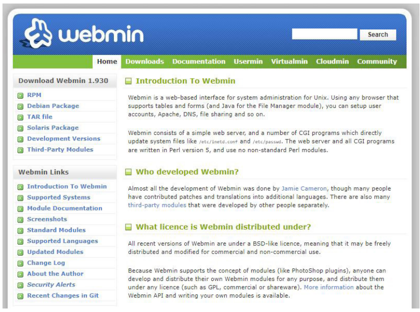
We are close to the end of this book, but I still have a few things to show you with Raspberry Pi.
If you are not comfortable with Linux in command line, you will probably love this chapter :)

In my job, I often train new staff with Linux commands, so I know how difficult it can be when you have never used them.
Even if it's mandatory as a Linux administrator, you may prefer to do this another way at home for your projects.
Today, I'll show you how to install Webmin to configure everything on the Raspberry Pi from a web interface.

Webmin presentation

Webmin is a web-interface that you can install on most of the Linux systems to help you with the configuration.
For each package and service on your system, you can get a more intuitive interface to edit the settings and to manage it.

Webmin is an old software, I have always seen it ever since I left school :)
According to Wikipedia, the first version dates from 1997; it's been around for a long time in the computing era.
It's still updated regularly, and the community is big enough to find help easily.
So even if the website and tool designs seems old, it's not a problem.



Webmin is written in Perl, and included in most distributions repositories.
The tool development was mainly done by only one developer (Jamie Cameron), but one strength of Webmin is that you can install third-party modules to meet your needs with specific software.

Webmin installation

As said in the introduction, there are two ways to install Webmin.
But first you need to have Raspberry Pi OS on your Raspberry Pi.

Install Raspberry Pi OS

I didn't try with other distributions, but the available package on the official website is a Debian package.
So Raspberry Pi OS is perfect, and it's probably the one you use the most.

Any version of Raspberry Pi OS is OK.
I write this chapter mainly for beginners on Raspberry Pi OS Desktop who prefer not to learn Linux commands.
But any experienced guy on Raspberry Pi OS Lite has the right to try it to make his life easier :-)

You already know how to install Raspberry Pi OS, so start with that.
Then I recommend updating the system and enabling SSH, so you can just copy and paste the commands I give you from your computer.

Option 1: Add a new repository

The first way to install Webmin is to add a new repository.
It may be a bit more complicated, but I think it's probably the best option.

This way you can manage updates as with any other software (graphically or with apt upgrade).

So, here is how to do this:

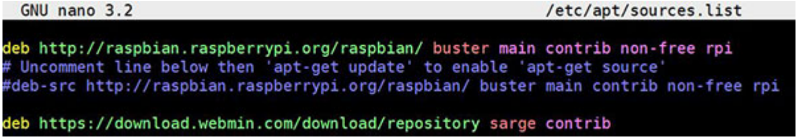
- Open the apt sources.list file:

  sudo nano /etc/apt/sources.list

- Add this line at the end:

  deb https://download.webmin.com/download/repository sarge contrib

- Yes, Sarge is an old Debian version, but this repository is updated regularly.
- It should look like this after the addition:



- Then you need to install the GPG key corresponding to this repository:

  wget http://www.webmin.com/jcameron-key.asc
  sudo apt-key add jcameron-key.asc

- Finally, install Webmin:

  sudo apt update
  sudo apt install webmin

That's it, it's not so difficult, and each time you update your system, you'll get updates for Webmin simultaneously.

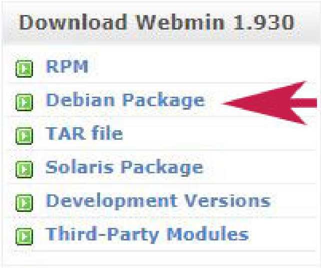
But it can also be an issue on production systems.
In my job, I prefer to do the updates myself for critical apps, this way I'm already on the server and can directly check if anything is broken.

As some of you may not like this option, I will give you the second option.

Option 2: Directly install the Webmin package

The second option is to download the latest version from the Official website and install it "manually"

- Go on the official Webmin site here: **http://webmin.com/**
- In the left menu, click on Debian package:



- The link lead to Sourceforge, and it will download it directly.
  If you are on your computer, you can use SFTP to transfer it to the Pi, or copy the link (in your browser downloads) and use wget
- For example, for the 1.930 version:
  wget http://prdownloads.sourceforge.net/webadmin/webmin_1.930_all.deb
- You can also take that link and change the version number, depending on the version displayed on the website:

- Then install the downloaded package:

```
sudo dpkg -i webmin_1.930_all.deb
```

- It may give you a warning because of missing prerequisites, it's normal, we'll fix this now
- Finish the installation with:

```
sudo apt -f install
```

- It will install Webmin and the missing dependencies

Your Webmin installation is done, you can move on to the next part.

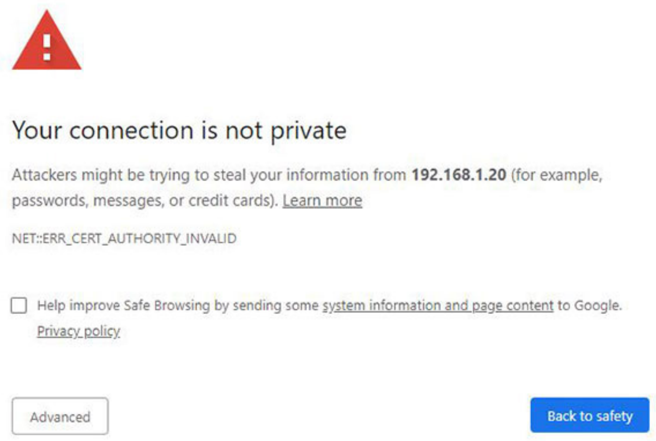
Webmin interface

Webmin works with a web interface.
It's pretty intuitive, but I will give you a few details here.

First access

Once installed, you can access it with the following URL:
https://IP_ADDRESS:10000
For example: https://192.168.1.20:10000

You'll get an SSL error in your browser as there is no certificate:



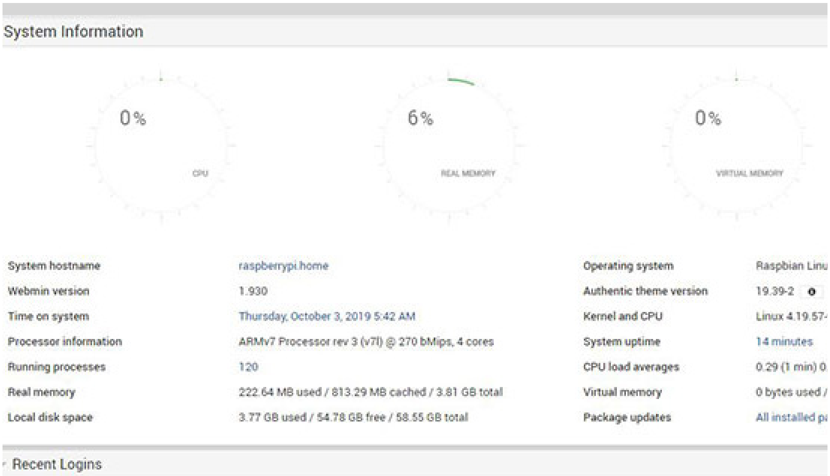
Just ignore it, by clicking on Advanced > Proceed to IP_ADDRESS (unsafe).

You can also disable the SSL encryption if you prefer. You can do this on Webmin directly (Webmin > Webmin configuration > SSL Encryption).
The corresponding file on the Pi is /etc/webmin/miniserv.conf (change ssl=1 to ss=0).

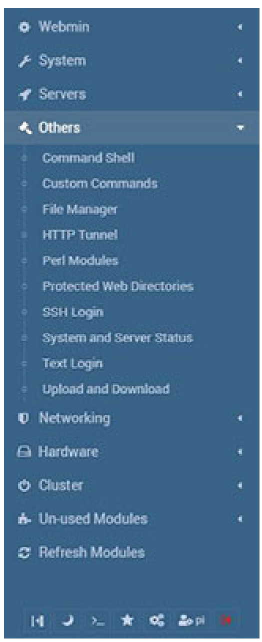
You'll get a login form with the Webmin logo.
The default credentials are those from the pi user.
If you kept the default password: pi / raspberry

Interface overview

Once logged in, you'll get the Dashboard page:

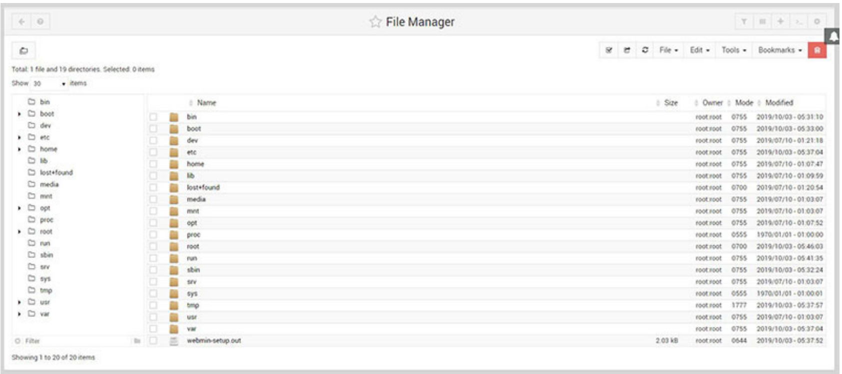


This first page allows you to have a quick overview of your server or Raspberry Pi. Then, you have a menu on the left, with all the default submenus:



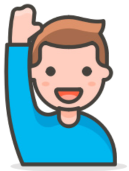
Click on one item to display the corresponding submenu below.

For example, you can click on **Others > File Manager**:



We now have access to a graphical file browser, even on Raspberry Pi OS Lite.

I will let you browse the different menus and submenus to give you an idea of everything you can do with this powerful tool.



Warning

As soon as you are connected to Webmin, you have administrator privileges (as with sudo).
So be careful if you change some values

Examples

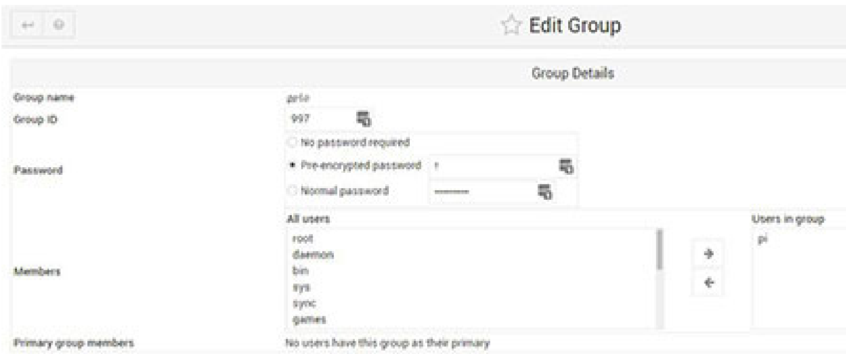
As you can see, there are many tools available by default.
Here are a few ones I like!

Users management

Creating and managing users and groups may be difficult for a beginner with just the command line.

Webmin offers a tool to do this intuitively:

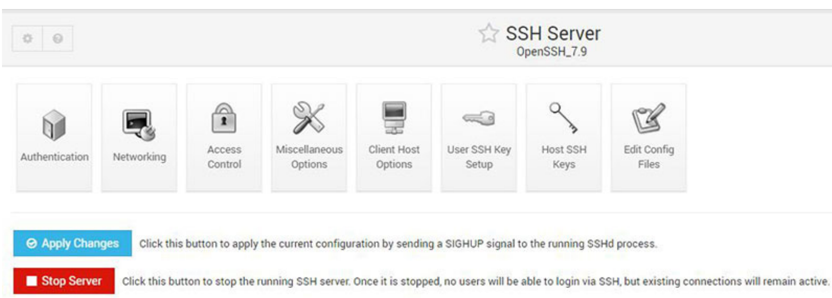
- Go to **System > Users and Groups** in the left menu
- Here you can see all the existing users, and also the groups by clicking on the other tab
- From here you can add new users and groups, and manage everything easily
- For example, when you click on a group, you can switch users in or out like this:



SSH server

Another example is the tool to configure your SSH options.
The tool is in **Servers > SSH Server**

From here you can manage the SSH server directly in the interface:



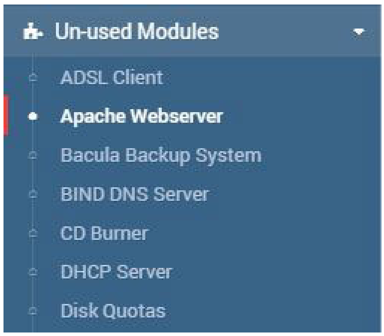
If you can never remember the options you need to change or the corresponding values, it's the perfect tool for you.
For each available option, there is a checkbox or a dropdown list so you can't make a mistake.

There are dozens more like these, so take your time to browse the menu and you'll probably find suitable tools for you

Install new modules

That's not all; you can also install new modules to Webmin.

You have probably seen the "Un-used modules" list in the left menu.
From here, you can see the default modules provided by Webmin that are not enabled because you don't use this software:

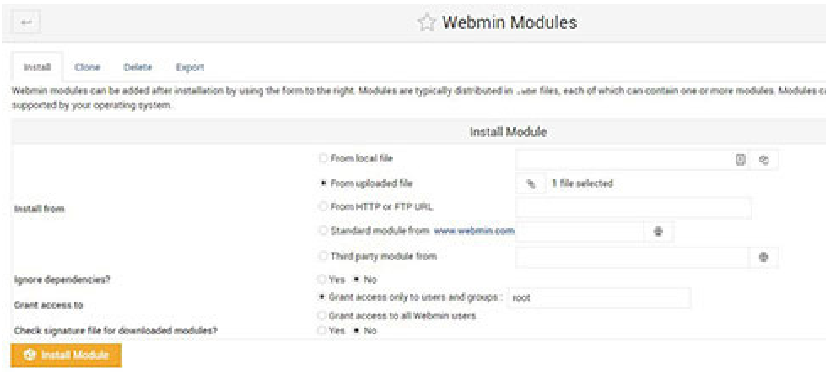


If you install one of them later on your system, Webmin will enable it.

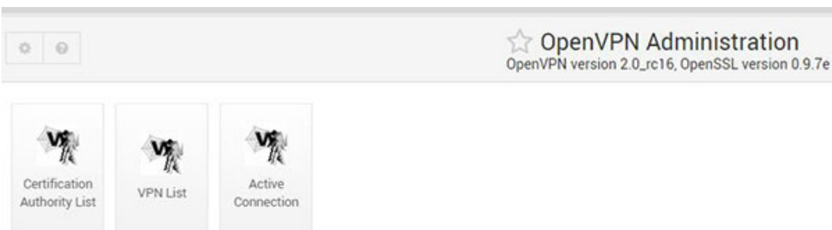
You can also find other modules here on the Webmin website.
By default, you only get the one developed by Webmin, but there are many modules created by other developers.

Click on the previous link, and a search engine will allow you to find a specific module.
For example, there is a module to manage the OpenVPN configuration:

- Click on Download to get it on your computer, or just copy the link
- Then go to Webmin, **Webmin > Webmin Configuration**
- Click on the Webmin Modules icon
- From here you can install a new module, from a local file or a URL:



- Your new module will appear directly in the menu after the installation:



Homework

That’s it, you know how to install Webmin on a Raspberry Pi and what you can do with it (almost everything).
I hope you liked this chapter, as I think it’s really a great tool for beginners, even if it’s a bit old-school.

To verify that you have understood this chapter well, you can try to install one module and see how it works.
Do you remember Fail2ban? We looked at this in chapter 13 if you need a quick reminder:)
Try to find a module for Webmin, and find out how to configure it the same way as in the configuration files.

Day 30:
30 Bonus Tips
For Raspberry Pi

Introduction

That's it, we have reached the last chapter of this book!
I think you have almost everything you need to take a good start on Raspberry Pi, and to become an expert in Linux if you work with this book enough to understand everything.
But before the end, I wanted to add a gift for you, with a few extra tips to help you with most of your projects.

When I say "a few", I actually mean "30 bonus tips" to close this 30-day challenge. They are sorted by category, and most of them come with a link to get more info (the idea is to make them short and useful).

Desktop

Let's start with a few tips on Raspberry Pi OS Desktop that I didn't mention in this book.

Create a shortcut on the Desktop

I don't exactly know why people are asking this so many times, but here is the answer.
In fact, there are two cases:

First case: create a shortcut for a Raspberry Pi OS software.

If the app appears in the main menu, it's straightforward:

- Open the main menu
- Browse to the app name in the menu
- Right-click on it, and select "**Add to Desktop**"
- The shortcut appears instantly

Second case: create a shortcut for a custom app.

If you have made a script or an app and want to create a shortcut on the Desktop, the process is slightly different.
Here is how to do this:

- Create a new file on the desktop:
  Right-Click on the desktop → Create New → Empty File
- Write the following lines in it:

```
[Desktop Entry]
Name=<AppName>
Comment=<AppDescription>
Icon=<YourIconPath>
Exec=<YouAppPath>
Type=Application
Encoding=UTF-8
Terminal=false
```

- Replace all the variables between <..> by your parameters
- Save the file, and name it like this: “AppName.desktop”

That’s it, you can double-click on it to run your script or custom application.

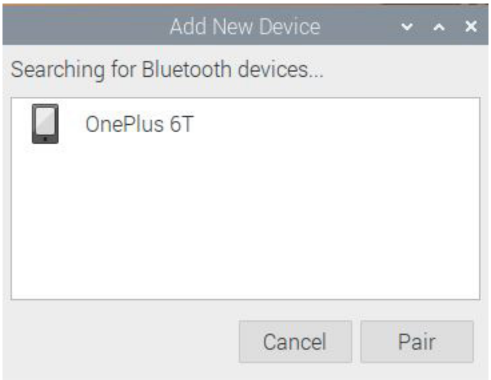
Connect a Bluetooth device

On recent Raspberry Pi models, you already know that Bluetooth features are integrated.
So on Raspberry Pi OS Desktop, you can pair your Bluetooth device with the Raspberry Pi.

Not all devices will work, but for basic things, it’s pretty well managed.

To do this, click on the Bluetooth icon near the clock (top-right of the screen).
Click on “**Add Device**”

A window like this shows up:



You can now pair your device by clicking on "**Pair**."
I'm often using my Bluetooth headset and it's working pretty well.

For some hardware, you can also do this the other way, by making the Raspberry Pi discoverable and connecting to it from another device.
Refer to your device manual for more information.

Free disk space on Raspberry Pi OS Desktop

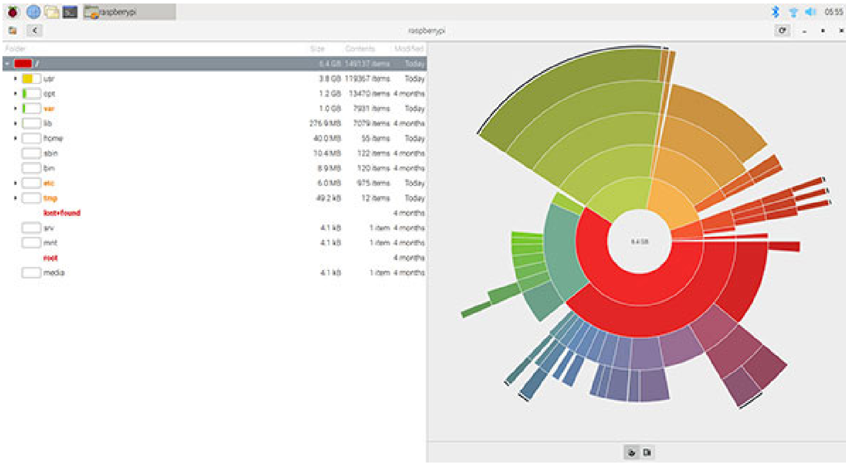
If your SD card is full, it may be complicated to continue using your Raspberry Pi correctly.
Before buying a bigger one, I recommend searching for the largest files on it remove what you can.

On Raspberry Pi OS Desktop, you can use a graphical tool to help you locate the largest files on the SD card (folders in fact).
This tool is named "Baobab".

You can find it in the Add/Remove Software tool and install it.
Then follow the quick procedure here to identify what is taking so much space on your SD card:

- Open the software from Application Menu > System tools > Disk Usage Analyzer
- Select a folder to scan (/ for example)

- You'll get something like this:



- You can directly see the biggest subfolders on the left and do the needed cleanup :)

Check this post for more information or to do the same thing on Raspberry Pi OS Lite: **https://raspberrytips.com/go/freediskspace**

System

Then here is a bigger part; a list of commands and things you could get stuck with.

Stopping the Raspberry Pi properly

Let's start with an easy one.
Most likely, you have already found that it's possible to turn off the Raspberry Pi by pressing the power button :-)

But it's better to do it properly with this command.

To stop the Raspberry Pi immediately use:

```
sudo shutdown -h now
```

If you use this to stop the Raspberry Pi, you'll need to press the power button twice for the next start.

Setting permissions

Linux permissions in command line are not the easiest thing to understand, and you may struggle to find out how to set them.
I will give you the basics here.

For any file or directory on your system, there are 3 levels of permissions, for each kind of person:

- **Owner**: generally the user that created the file (ex: pi)
- **Group**: it depends on the situation, it can be the system administrators, a specific group for the program, or any other group existing on the system (ex: root)
- **Others**: people that are not the owner and not in the previous group (ex: another user)

Then for each of these levels, there are 3 permissions that you can set:

- **Read**: allowed to read the file
- **Write**: allowed to edit, rename or delete the file
- **Execution**: allowed to run the file (for a script for example)

So basically, you can give all permissions to you, only read for a group of people and nothing to others.

To configure this, you need to use these 3 commands:

- **Chown**: to change the owner of a file:
  Ex: *`chown pi /home/pi/myfile.sh`*
- **Chgrp**: to change the group of a file:
  Ex: *`chgrp root /home/pi/myfile.sh`*

- **Chmod**: to change the permissions of a file:
- There are two ways of using this
  - The easiest way is to add or remove right by using letters like this:

    ```
    chmod g+x /home/pi/myfile.sh
    ```

  - This will add (+) the execution permission (x) to the group (g)

    ```
    chmod o-w /home/pi/myfile.sh
    ```

- This will remove (-) the write permission (w) to the others (o)

  chmod u+rwx /home/pi/myfile.sh

  This will add (+) all permissions (rwx) to the owner (u)
  Note: you need the write permission to change the file permissions, so this makes no sense, but I just used it as an example :)

To check the current file permissions, you can use the "ls" command like this:

ls -l /home/pi/myfile.sh

Most of the time, system administrators are using a numerical value as a shortcut to set the permissions.
For example you can use "*chmod 777 myfile.sh*" to add all permissions to the file, but this may be a bit complex for a beginner.
You can use this calculator if you want to play around a bit with this notation:
https://chmod-calculator.com/

Login as root

On Raspberry Pi OS, the pi user is the default one, and you can't use root directly (for security reasons).
But how do you use administrator commands with pi?

Well, most of the time, the "sudo" command is here to help you.
As you have seen many times in this book, you can add "sudo" at the beginning of an admin command to get the right to use it.
But you can also switch to a superuser account for a while, do what you need to do, and switch back to pi after that.

Here is the "magic" command to do this: *sudo su*
After typing this, your terminal prefix will start by "root@raspberrypi" and you can do anything you want without using "sudo" :)
Type "*quit*" to come back to the pi user terminal.

If you need more details about this, I have a complete tutorial here on RaspberryTips: **https://raspberrytips.com/go/root**

Analyze log files

Another thing you may want to understand is the logging system on Raspberry Pi.

It’s the same as on all Linux distributions, but if you are new to this, it can be a bit confusing.

All the system log files are located at the same place under /var/log
Here are the most important files in this folder:

- **/var/log/messages**: Generic events on the system, generally used for non-critical events
- **/var/log/auth.log**: Here you can read all the authentication events happening on your system (ssh attempts for example)
- **/var/log/syslog**: Probably the file to remember, as it contains any events on the system (except auth.log)

But you can also find other files depending on the services you are using.
For example, Apache (a web server) will create a new subfolder here, with all the log files.

They are standard text files, so you can use any command you like on it (more, cat, grep or even nano).
But here is my favorite command to see what happens in a specific file:

```
tail -f /var/log/syslog
```

This will display the new lines written to the syslog file in real-time.
So you can run this in a terminal, and start the service that won’t start in another, and you’ll see what happens in real time.

Set a static IP

Throughout this book, I have assumed that you are connected to a network with a DHCP server running. This means that IP addresses are automatically assigned to computers, usually by your internet router.

But it can be annoying to have the Raspberry Pi’s IP address change every time the router decides to do so. It’s possible to avoid this by setting up a static IP address for your Raspberry Pi.

If you have a DHCP server with a web interface or something fairly intuitive, the easiest way is usually to configure it directly on the server or router.

However, it's also possible to do this directly on the Raspberry Pi by following these steps:

- Open a terminal (or use SSH), and start the network manager configuration tool:
  sudo nmtui
- Choose "Edit connection" and find the interface you want to set to a static IP.
- Select it, and choose "Edit" in the right menu.
- Scroll down to "IPv4 Configuration" and switch from "Automatic" to "Manual".
- Fill the form with the IP address, gateway and DNS server you want to use.

```
═ IPv4 CONFIGURATION <Manual>                                  <Hide>
           Addresses 192.168.222.231__________ <Remove>
                     <Add...>
             Gateway 192.168.222.1____________
         DNS servers 192.168.222.1____________ <Remove>
                     <Add...>
      Search domains <Add...>
```

- Save and exit the tool.
  On the next reboot, the system will use these settings all the time.

This is the short version of a longer tutorial I have on the website, going into more details, here is the link:
https://raspberrytips.com/set-static-ip-address-raspberry-pi/.

Find the MAC Address

Sticking with the topic of network, you may also need to know the MAC address of your Raspberry Pi.
By the way, it may also be useful to fix the IP address on your router.
You already know the command, it's "ifconfig"
The MAC address is close to the IP address in the command result:

```
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.35  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 2a01:e34:ee90:5110:2322:cac7:7337:2aa9  prefixlen 64  scopeid 0x0<global>
        inet6 fe80::5d96:6eee:1e4f:dc12  prefixlen 64  scopeid 0x20<link>
        ether b8:27:eb:1a:40:c0  txqueuelen 1000  (Ethernet)
        RX packets 23254  bytes 14041799 (13.3 MiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 33627  bytes 38105709 (36.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

A MAC address is on the following format: AA:BB:CC:DD:EE:FF
You can find it after the keyword "ether" in the "ifconfig" output.

As explained previously, you can also use “ip a” which is the new way of doing this, as ifconfig will soon be removed.

The MAC address is different for the Ethernet and the Wi-Fi card, so make sure to note the correct one.

Find the Raspberry Pi OS version

Not sure on which Raspberry Pi OS version your Raspberry Pi is running?
You can find it easily with only one command:

```
cat /etc/os-release
```

The result looks like this:

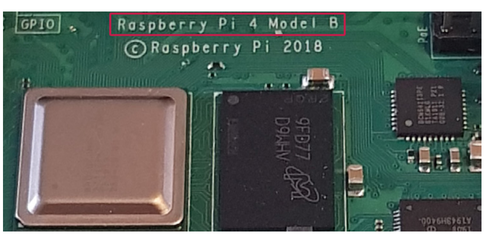
```
pi@raspberrypi:/var/log $ cat /etc/os-release
PRETTY_NAME="Raspbian GNU/Linux 10 (buster)"
NAME="Raspbian GNU/Linux"
VERSION_ID="10"
VERSION="10 (buster)"
VERSION_CODENAME=buster
ID=raspbian
ID_LIKE=debian
HOME_URL="http://www.raspbian.org/"
SUPPORT_URL="http://www.raspbian.org/RaspbianForums"
BUG_REPORT_URL="http://www.raspbian.org/RaspbianBugs"
```

In this case, my system is running on Raspberry Pi OS Buster.

Find the Raspberry Pi model

As far as I know, there is no magic command to identify the model version you have in hand. But it should not be complicated.

First thing, if you have the Raspberry Pi close to you, the model is written on the main board. Here is an example with the Raspberry Pi 4:



I have checked on the other devices I have, and it's written each time close to the GPIO pins.
So that is the easiest way to find this.
If you are connected to a remote device, it can become more complicated.

But here are the things you can note:

- The amount of memory:
  - The easiest way is probably to use the htop command, then note the memory total displayed at the top (same line as mem)
  - You can also use this command:
    cat /proc/meminfo

    And note the value on the first line (MemTotal)
- The processor information:
  - How many CPUs in htop?
  - You can also use this command:

    cat /proc/cpuinfo

    And note the number of CPUs, and the hardware version

Now, go back to "Day 1" in this book, and check the model tables to find the corresponding model.
It should be easy enough to pick the correct answer :)

Change the device name

By default, all Raspberry Pi are installed with the same device name: raspberrypi
You may want to know the command to display the current name:
hostname

And here is how to change it:

- Open the hostname configuration file:

  sudo nano /etc/hostname

- There is only one line with the host name.
  Change it to what you want.
  For example: raspberrytips01
- Reboot your system:

  sudo reboot

After the reboot, use “hostname” again to check that it has been updated correctly.

You can also see it in the terminal prefix:

`pi@raspberrytips01:~ $`

Schedule something

We have seen this quickly on Day 14, but maybe it was not enough to understand it correctly.

On any operating system, there is a way to schedule tasks.
That’s to say that you can choose to run a script or a program at a specific date and time, once or regularly.
On Day 14, we scheduled a backup every day at midnight, but you can do whatever you want.

As a reminder, here are the commands:

- **crontab -l**: display the list of scheduled tasks
- **crontab -e**: edit the task list

If you add “sudo” before the command, the task will start as the super user.

Then you need to respect a simple format:
<time> <script>
The format seen like this is simple, but in the details, you need to be careful what you type.

The script is the complete path to the file, here are two examples:

- /home/pi/myscript.sh
- /bin/php /var/www/myscript.php

But the complex part is the time scheduler :)
There are five columns to define:

- Minute
- Hour
- Day of month
- Month
- Day of week

For each value, you can set it to “*” to run it using whatever the current minute/hour/day/month is.
You can also use “,” to set several values, “-” for ranges and “/” for step values.

Here are a few examples to help you to understand this:

- * * * * *: Run the script each minute
- 0 * * * *: Run the script each hour at minute 0 (0:00, 1:00, 2:00, ...)
- 55 11,23 * * *: Run the script each day at 11:55 and 23:55
- 0 0 1 * *: Run the script on the first day of each month
- 0 3 * * 1-5: Run the script at 3:00 from Monday to Friday
- Etc.

Even advanced system administrators can make mistakes with this, so there are many websites that have a tool to help you.
For example you can check this one out if you are struggling:
https://webinpact.com/crontab-generator/

For more information, you can also read this tutorial:
https://raspberrytips.com/go/cron

Autostart a program

On Raspberry Pi, you will often want to start something on boot.
It can be a server, a display or anything else.

The first option is to use the crontab we have seen in the last tip.
I didn't tell you here, but there is a magic option that you can use to schedule a task: @reboot
For example:

```
@reboot /home/pi/myscript.sh
```

But there are several options that you can use depending on what you want to start.
You can read the step-by-step tutorial here:
https://raspberrytips.com/go/autostart

Add a repository

Another tip I want to give you is how to manage the apt repositories on Raspberry Pi OS.
When you use "`apt install`", the tool will check if there is a program with the same name available on the servers.
By default, you start with the main Raspberry Pi OS repository, but it's possible to change the configuration to use another server instead or add one (we'll use this in the following tip).

The configuration file is here: /etc/apt/sources.list
You can edit it with *nano* and change anything you want.

You can find a list of Raspberry Pi OS repositories here:
https://raspberrytips.com/go/mirrors
And some programs may ask you to install another one (the Oracle Java version for example).

After editing the file, you need to run "*apt update*" to update the local index.
Some repositories will ask you to install a key to use it, refer to the repository documentation to learn how to do that.

Upgrade Raspberry Pi OS version to the latest one

This is following from the previous tip.
The Raspberry Pi Foundation will regularly release a new Raspberry Pi OS version (generally every one or two years).
For example, in 2019 they released "Buster" to replace the old "Stretch" version.

If you have a complete system with everything in place, you may prefer to upgrade it rather than reinstalling all the packages and configurations.
There is a way to do this, but first I recommend doing a complete backup of your system (check Day 14 of this book).

Then you can edit the apt repositories:

- Open the configuration file:

```
sudo nano /etc/apt/sources.list
```

- Find all references to the old Raspberry Pi OS name and replace it with the new one.
  For example if you a have a Raspberry Pi running on Raspberry Pi OS Buster:

```
deb http://raspbian.raspberrypi.org/raspbian/ buster main contrib non-free rpi
=>
deb http://raspbian.raspberrypi.org/raspbian/ bullseye main contrib non-free rpi
```

- Save and exit (CTRL+X)
- Update your local cache:

```
sudo apt update
```

- Upgrade all the packages:

```
sudo apt upgrade
sudo apt dist-upgrade
```

- Reboot the system:

```
sudo reboot
```

That's it, your system is now using the latest version :)
I hope that everything works perfectly (If not, you can still use the backup :/).

Time syncing

In theory, your Raspberry Pi should already have time sync enabled by default. But on specific networks, it may be useful to change the configuration, or synchronize it with one of your servers.

As it's a complex topic, with many solutions depending on what you want to do, I will not go into more details here.
Just know that it's possible, you just need to know what fits your needs the best.

There is a complete tutorial here that you can read to learn how to do this:
https://raspberrytips.com/go/timesync

Dual boot

If you are just starting with a Raspberry Pi, you may want to try many systems (I know this feeling very well ^^)
Rather than flashing your SD card again and again with a new distribution, you can install several systems on the same SD card and switch from one to another quite easily.

Firstly, you need an SD card with enough space (probably 32GB or more).
Check my up-to-date recommendation here if needed:
https://raspberrytips.com/go/basic

Then you need to download BerryBoot from the official website, here:
https://raspberrytips.com/berryboot

And then create a new SD card:

- Extract the downloaded file to a new folder
- Create a new partition on the SD card (1GB should be enough)
- Copy all the extracted files in this partition
- Eject the SD card

Now you can start your Raspberry Pi, and the wizard will show up.

For a more detailed tutorial, check this link:
https://raspberrytips.com/go/dualboot

Format & mount usb drives

To finish this section, another tip that you may want to know is how to manage USB drives on Raspberry Pi.
On Raspberry Pi OS Desktop it's intuitive, but with a command line, it's not that easy for a beginner.

As with many of the other tips, I will quickly give you the steps to follow here, and you'll get a link to a complete tutorial at the end.

On Raspberry Pi OS, USB devices are assigned to a virtual name, something like /dev/sdX (with X as the first letter available).
So the first thing to do before manipulating the USB drive, is to find this name.
To do this, you can use the following command:

```
fdisk -l
```

And look for this in the command output:

```
Disk /dev/sda: 7.5 GiB, 8053063680 bytes, 15728640 sectors
Disk model: UDisk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x163287fe

Device     Boot Start      End  Sectors  Size Id Type
/dev/sda1  *     2048 15728639 15726592  7.5G  c W95 FAT32 (LBA)
```

Here is my 8G USB key, and the corresponding partition is /dev/sda1, formatted in FAT32.

If needed, you can now use fdisk to format it:

```
fdisk /dev/sda
```

Or you can mount it with the mount command, in a temporary folder, like this:

```
sudo mkdir /media/usb
sudo mount /dev/sda1 /media/usb -o umask=000
```

As I said at the start, here is the link to the step by step tutorial if you need more help: **https://raspberrytips.com/go/mount**

Projects

And finally, I want to add a few other ideas on many things you can install on your Raspberry Pi to start a new project.
They are just hints to get started or to give ideas, and a link will lead you to a more complete tutorial.

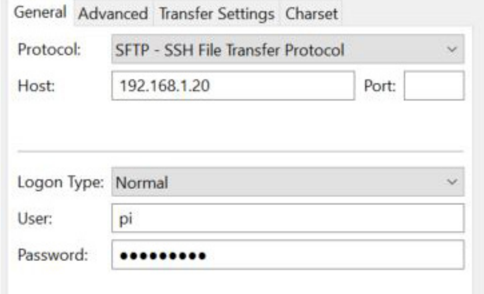
Transfer files to/from your computer

The first one is not really a projects idea, but you will use it often when installing or configuring new services for your projects.
How do you transfer files from the Raspberry Pi to your computer? Or the opposite?

You have several solutions to do this, but the easiest one I'm using all the time is Filezilla.
Once SSH is enabled on Raspberry Pi OS, you can use this protocol to connect to your Raspberry Pi, and also to transfer files!

- Download and install FileZilla.
  It's a free software, available here: **https://filezilla-project.org/**
- Open the site manager with CTRL+S

- **Create a new site**, name it “Raspberry Pi” for example, and fill in the general options like this:



- Click on “**Connect**”

The content of the /home/pi folder appears on the right.
On the left you can browse through your computer.
You can transfer files from one to the other by double-clicking on any file, or with right-click → Upload/Download

That’s it :)

Install a web server

Hosting a website can seem a difficult mission when you don’t know how it works.
But in fact, it just takes a few services to install and configure correctly.
Nothing complicated with your current skills!

The basic service you can use for a static website is Apache.
And the installation can be done with apt, like any other package:

```
sudo apt install apache2
```

After the installation you can go to **http://<IP>**
Replace <IP> with the Raspberry Pi IP address, and a web page will appear.
The website files are located under /var/www/html

If you have any experience with HTML you can try to create html files in this folder (manually or as the result of your scripts).

But if not, any other files that a browser can read or download is fine (text, images, etc).

For example, you are now able to create a script that runs daily and writes the result in a text log file that you can read from your browser :)

Add dynamic features to your web server

If you want to add dynamic capabilities to your web server, you need to install something like PHP.
The goal of the PHP language is to add dynamic parts in a HTML page (typically by reading the content of a database).
You can also install PHP with apt.
You need to install the PHP core package and a link package to enable PHP in Apache, like this:

```
sudo apt install php libapache2-mod-php
```

You can now create PHP files under /var/www/html
Here is an example, if you don't know how PHP works:

- Create a new file:

  ```
  sudo nano /var/www/html/test.php
  ```

- Paste these lines in it:

  ```
  <?php
  echo date("Y-m-d H:i:s");
  ?>
  ```

- Check if it's working.
  Go to http://<IP>/test.php
  The current date and time appears, and is updated each time you refresh the page :)

Install a MySQL server

To complete your web server, you can now install a database server on your Raspberry Pi, basically a MySQL server.
This time you need the MySQL server package, and a package to make the link between PHP and MySQL:

```
sudo apt install mariadb-server php-mysql
```

Before using it, you need to restart apache:

```
sudo service apache2 restart
```

Then you need to create your first database, insert data, add a new user and write a bit of code to read the database from a PHP page.
It's an entire project that I can't include directly here, to keep the size of this chapter reasonable.
But if you want to try it, you'll find all the information here:
https://raspberrytips.com/go/webserver

Install a mail server

The end of this chapter will mainly be a list of projects ideas you can start to bring new services to your network.
For each one, I will introduce it and give you a link to the complete tutorial.

Sending email is often a prerequisite for many projects, as it allows you to send notifications or reports to your email addresses.
To do this you'll need a domain name and a static IP. A dynamic DNS service like No-IP can also be used.

Then you have two choices:
- Install only the basics to be able to send email to anyone (SMTP server)
- Install a complete mail server to be able to receive emails on the Raspberry Pi (with POP and IMAP protocols)

Postfix with a basic configuration is enough to send emails, but being able to receive emails is a bigger project, as you will probably need to install and configure other services like Dovecot (POP/IMAP) and Roundcube, Apache and MySQL for the Webmail interface.

You will find everything you need in this complete tutorial:
https://raspberrytips.com/go/mailserver

Install a DNS server

This one is easier to set up.
DNS stands for Domain Name System. The role of a DNS server is to translate domain names into IP addresses.
Most of the time, you use the DNS server from your provider or a big company (like Google for example).

It can be a good idea to install one on your Raspberry Pi.

It can:

- Speed up your Internet browsing
- Keep your browsing safe
- Be more stable (with speed and availability)
- Manage custom records for your local network (ex: pi.mydomain.com)

For this project you will need to install a service on your Raspberry Pi, and then change the computer configuration to use this DNS server rather than the default one.

You can find all the steps to do this in this tutorial:
https://raspberrytips.com/go/dns

Install a file server

You may have a NAS at home or work, or maybe you are using a network hard drive on your Internet router.
The goal of this project is to build the same thing on your Raspberry Pi.
You can have a Raspberry Pi running somewhere in your house, and store files on it (like backups or utilities).

The major advantage to doing this is the price.
You can build your home NAS for something like $50 on a Raspberry Pi, but not less than $300 with an entry level NAS.
Yes, it's not the same thing, but for basic usage a Raspberry Pi may be enough for you.

Here are the main steps of this project installation:

- Choose a big SD card or a USB hard drive to add more spac2.
- Install Raspberry Pi OS (Lite) on your Raspberry Pi
- Install Samba and configure it to allow network access
- (Optional) Turn this server into a torrent server or add DLNA features
- (Optional) Try a NAS distribution and configure RAID and backups

As usual, you will find all the details of these steps in this complete guide here:
https://raspberrytips.com/go/fileserver

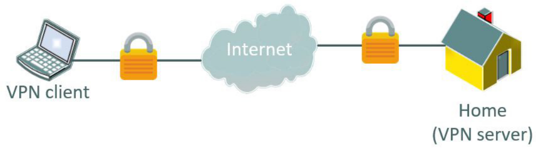
Install a VPN server

VPN stands for Virtual Private Network.
And that's exactly what it is. When connected to a VPN, it's as if you were on a

private network between you and the VPN server.

The main goal of a VPN is to encapsulate your data in a secure tunnel between you and the VPN server.

Let's take an example.
If you share a web server at home with port forwarding (public_ip:80 => local_ip:80), data could be accessible to hackers, as data flows in clear on the network (man in the middle attacks are possible).
If you use a VPN server on your Raspberry Pi, data flows in the secure tunnel, so nobody can decrypt them:



To do this you need to install the server on your Raspberry Pi and the client on your computer (it also works on smartphones).
Once configured, you can securely access your local network from anywhere in the world.

The configuration is a bit complex on both sides, so I will refer you to this tutorial:
https://raspberrytips.com/go/openvpn

Unifi: a Wi-Fi controller on Raspberry Pi

This is probably more of a project that you will build in a company than at home but let me explain the ideas behind it.

At home, you probably only have one wireless access point and it's ok to be like this.
In companies or public places, it will not be enough, and you need to install many access points to cover the entire area.
And it's not so easy, it can quickly become a nightmare depending on the choices you made for the hardware.

Ubiquiti Networks is a company I'm using at work, and that works pretty well compared to other solutions I have tested before.
And one of its strengths is to have a controller software available on Linux, including Rasbian :)

You need to use the access points from the same company to use it, but they work well and are not expensive.
If you are interested, you will find all the details about the hardware and the installation process here:
https://raspberrytips.com/go/unifi

Install Gitlab

If you don't know, Git is a code manager, used to share sources between all developers of the same project.
It allows concurrent editing of the same files, and even if you are alone, it's a good tool to back up your code and note the changes you make.
The main difference between SVN (Subversion) and Git is that Git doesn't need a main server to keep files, but instead it's using the developers computers to store this.
And GitLab is precisely the missing server in the default Git architecture (like GitHub).

GitLab allows you to create a private GitHub-like server at home on your Raspberry Pi.
Since the acquisition of GitHub by Microsoft you could be forgiven for worrying about data privacy
A lot of companies have moved to other platforms, and the cheapest solution for individuals is to use their own private Git server.

But as an expert in Raspberry Pi, you can even host it on your device rather than using the public GitLab servers :)
GitLab is not available in the Raspberry Pi OS repositories, so you have to install it manually

To do this, you can follow the step-by-step tutorial here:
https://raspberrytips.com/go/gitlab

Build a cluster

A cluster is a group of computers in a single entity.
The goal is to make them work together, to improve the global performance.
All the computers work on the same task, reducing the time needed to finish it.

As you know, the Raspberry Pi is not so powerful, but it's cheap.
So, it's the perfect device to build a cluster.

We can make it run tasks faster on 4 devices instead of only one, for a reasonable price.

To do this you need to link all your devices on the same network, and install the same thing on each Raspberry Pi.
In fact, you will install everything on the master once, and then duplicate the SD card as we have seen in “Day 14.”

The software part is managed by MPICH and MPI4PY.
And as usual, you can find the complete guide on RaspberryTips:
https://raspberrytips.com/go/cluster

Conclusion

That’s it!
I hope this chapter taught you some new things and brought you new project ideas.

No homework for this last day, but I’m sure that after reading this book (and this chapter in particular), you will try many new things to increase your skills on Raspberry Pi :)

Conclusion

We have come to the end of this adventure in which we discovered this small computer; the Raspberry Pi.
I have no doubt that your skills have greatly improved if you have been serious about reading this book.
On first sight, the Raspberry Pi is just a tiny computer with poor performance, but you now know that it is way more than that.

I hope this book has given you what you were looking for when you started it, and that you are happy with its content.
If so, I would appreciate if you left a review, it will just take a few seconds but will help me a lot in my goal of promoting the book to other people:
https://raspberrytips.com/book-review

Feel free to check out the tutorials on RaspberryTips for more details, and for other project ideas with your device.
The advantage of the site is that it is possible to exchange with others in the comment section if you have corrections, information to add or questions to ask.

I hope we will speak on the website, but in any case I am at your disposal by email: **contact@raspberrytips.com**
Feel free to contact me for any suggestions you have or any mistakes you have found in this book.
My role is to help you, give you the answer if I can, or point you to people who can help you better than I can myself.

Good luck in your future endeavours, master of the Raspberry Pi :)