

Extracting Media Links from a Web Page



Allen O'Neill

ENGINEER

@databytzai



Useful developer tools

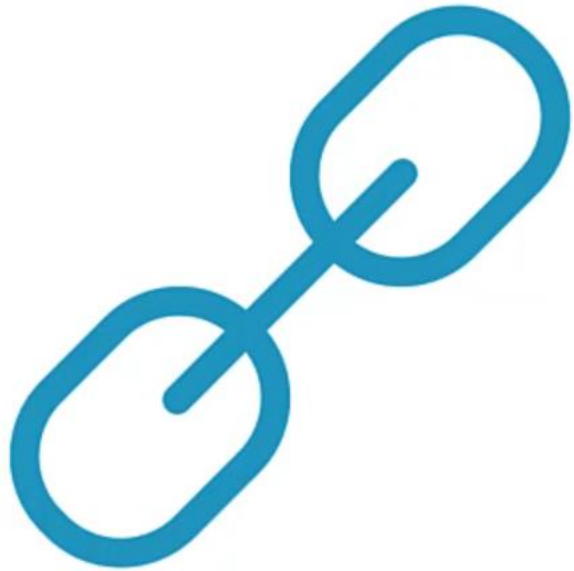
Categories of links

Code environment

Find media links

Extract media links

Media Links

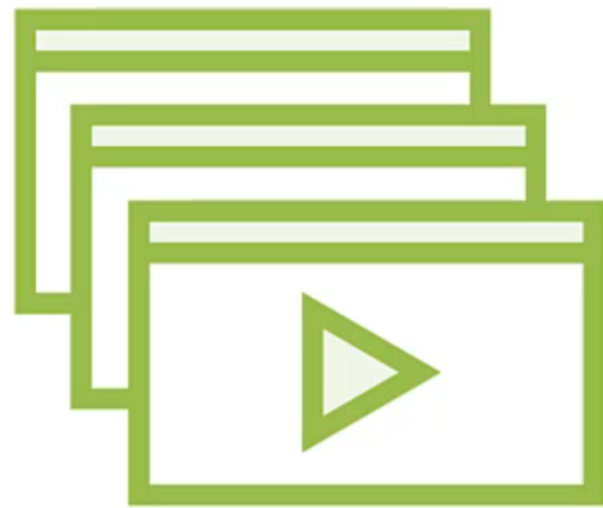


Identifying Media Links

Media



Web Media



Digital Media



```
<ul>
```

```
  <li> bullet point 1 </li>
```

```
  <li> bullet point 2 </li>
```

```
</ul>
```

```
<div>
```

```
  <span> page sub-title </span>
```

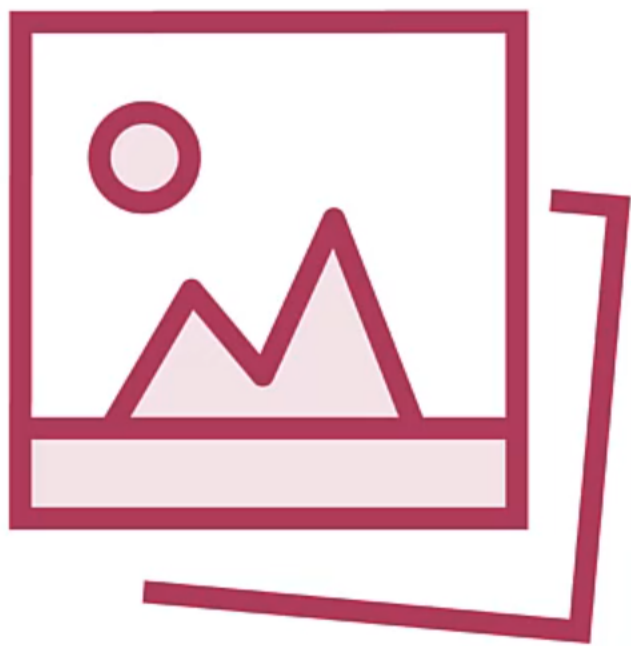
```
</div>
```

HTML Tags

Basic

Formatting

Forms



<CANVAS>

<SVG>

Attributes

```

```

Absolute vs Relative

`http://website.com/images/dog.jpg`

`/images/dog.jpg`

Canvas

The Canvas API provides a means for drawing graphics via JavaScript and the HTML `<canvas>` element.



Dynamic

Automated browser

Pluralsight courses

SVG

Scalable Vector Graphics (SVG): XML-based mark-up language that describes two-dimensional vector graphics.

SVG Example



```
<svg height="210" width="600">  
  <polygon points="350,5 250,250 125,210 5,223"  
    style="fill:gray;stroke:gray;stroke-width:1" />  
</svg>
```

Dynamic Websites





Angular

ReactJS

Backbone



Challenging

Non-obvious

Investigate

Environment Requirements



Development tools

- All similar
- Established tools
- Libraries available

Assumption:

- Python V3+
- PIP3
- Python knowledge

Python Libraries

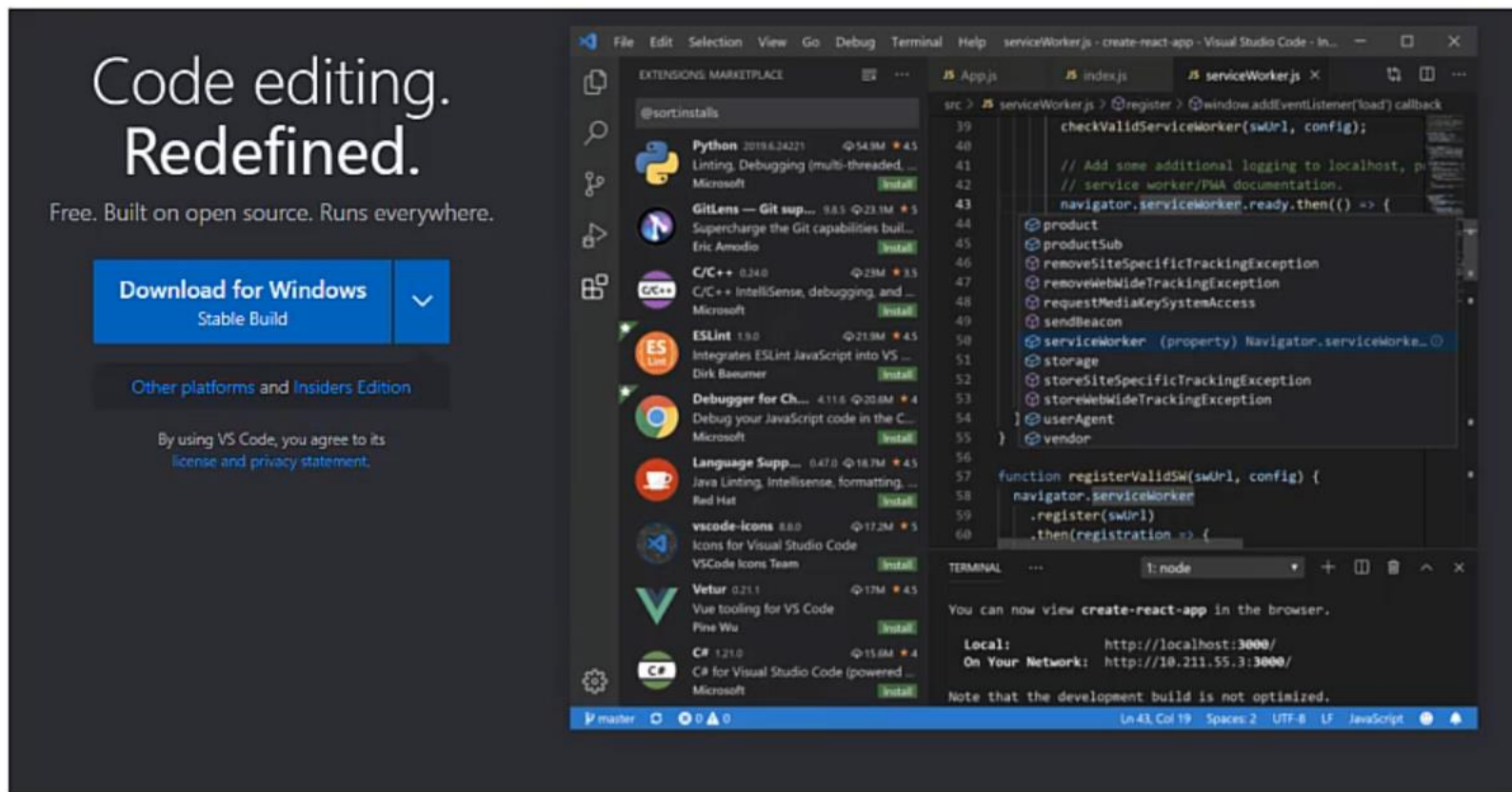
Requests 2.21

```
pip3 install requests
```

Lxml 2.92

```
pip3 install lxml
```

Visual Studio Code



IntelliSense



Run and Debug



Built-in Git



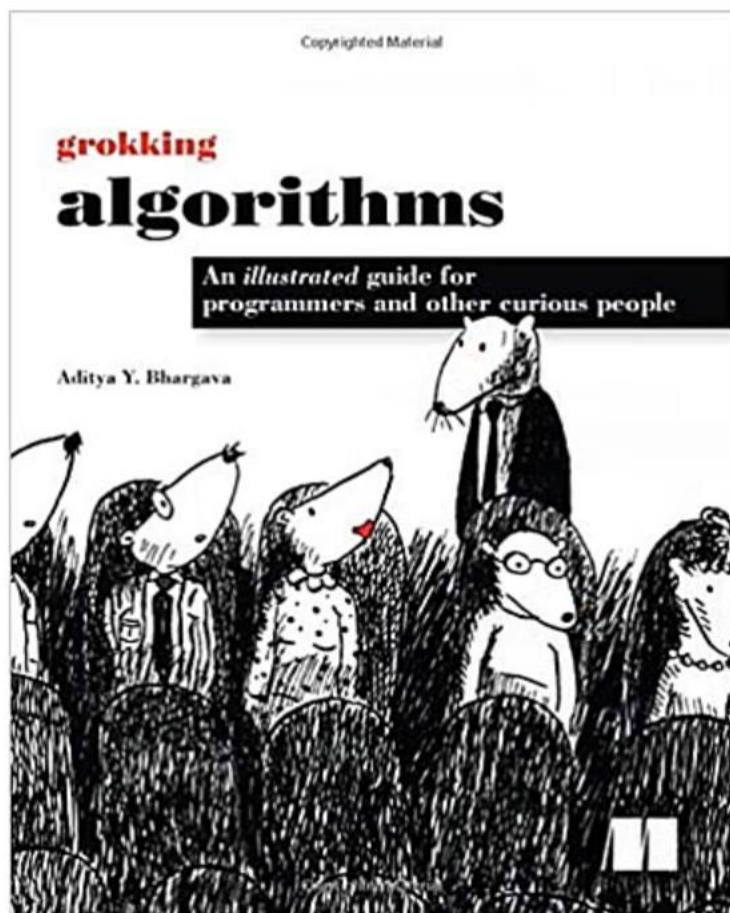
Extensions

<https://code.visualstudio.com/>

Finding Media Links

Grokking Algorithms: An illustrated guide for programmers and other curious people

A book by Aditya Bhargava



Algorithms are nothing more than step-by-step procedures for solving problems and most algorithms used by programmers have already been discovered, tested, and proven. Those wanting to take a hard pass on Knuth's brilliant but impenetrable theories, and the dense multi-page proofs found in most textbooks, will want to read Grokking Algorithms. This fully-illustrated and engaging guide makes it easy to learn how to use algorithms effectively.

Grokking Algorithms is a disarming take on a core computer science topic and shows readers how to apply common algorithms to practical problems faced in the day-to-day life of a programmer. It starts with problems like sorting and searching and builds up skills in thinking algorithmically. Then it tackles more complex concerns such as data compression or artificial intelligence. Whether writing business software, video games, mobile apps, or system utilities, readers will learn algorithmic techniques for solving problems that they thought were out of reach. By the end of this book, they will know some of the most widely applicable algorithms, as well as how and when to use them.

Product details Paperback: 300 pages

Publisher: [Manning Publications](https://www.manning.com/); 1 edition (31 Dec. 2015)

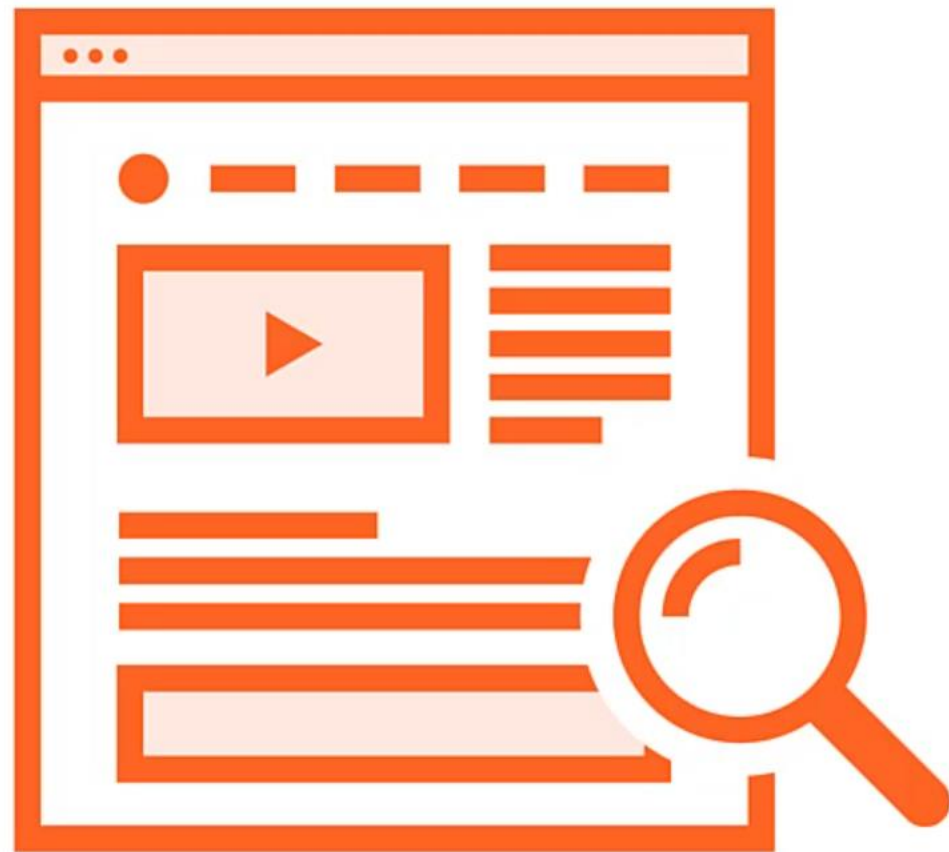
Language: English

ISBN-10: 1617292230

Product Dimensions: 19 x 1.3 x 23.5 cm

This book and the image on this page are copyright manning publications. The book can be purchased from Manning here: <https://www.manning.com/books/grokking-algorithms>

Finding Links





UI \neq HTML

Check, don't assume

Investigate framework

Developer Tools

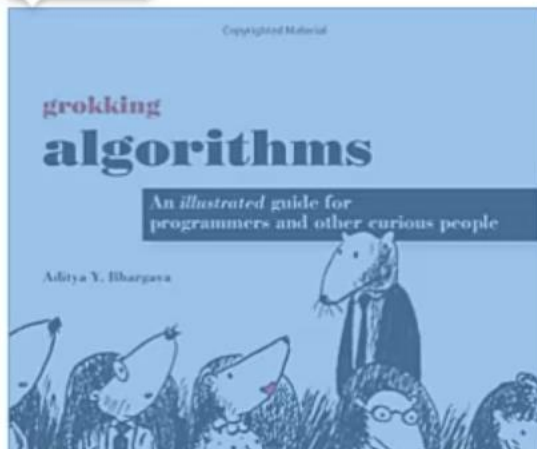


MEDIA SCRAPE EXAMPLE PAGE 1

Grokking Algorithms: An illustrated guide for programmers and other curious people

A book by Aditya Bhargava

img 399 x 499



Algorithms are nothing more than step-by-step procedures for solving problems and most algorithms used by programmers have already been discovered, tested, and proven. Those wanting to take a hard pass on Knuth's brilliant but impenetrable theories, and the dense multi-page proofs found in most textbooks, will want to read Grokking Algorithms. This fully-illustrated and engaging guide makes it easy to learn how to use algorithms effectively.

Grokking Algorithms is a disarming take on a core computer science topic and shows readers how to apply common algorithms to practical problems faced in the day-to-day life of a programmer. It starts with problems like sorting and searching and builds up skills in thinking algorithmically. Then it tackles more complex concerns such as data compression or artificial intelligence. Whether writing business software, video games, mobile apps, or system utilities, readers will learn algorithmic techniques for solving problems that they thought were out of reach. By the end of this book, they will know some of the most widely applicable algorithms, as well as how and when to use them.

Product details Paperback: 300 pages
 Publisher: [Manning Publications](#); 1 edition (31 Dec. 2015)
 Language: English
 ISBN-10: 1617292230

Elements Console Sources Network Performance Memory Application Security Lighthouse Media

```
<html>
  <head>...</head>
  <body>
    <br>
    <center>
      <h3>MEDIA SCRAPE EXAMPLE PAGE 1</h3>
      <h1>...</h1>
      <div>...</div>
      <br>
      <br>
      <table width="60%" cellpadding="20">
        <tbody>
          <tr>
            <td valign="top">
              
            <td valign="top">...</td>
          </tr>
        </tbody>
      </table>
    </center>
  </body>
</html>
```

html body center table tbody tr td img

Console What's New x

Highlights from the Chrome 87 update

55 | Settings |

Styles Computed Layout Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls +

element.style { }

Inherited from table

table { user agent stylesheet

border-collapse: separate;

text-indent: initial;

white-space: normal;

line-height: normal;

font-weight: normal;

font-size: medium;

font-style: normal;

color: -internal-quirk-inherit;

text-align: start;

border-spacing: 2px;

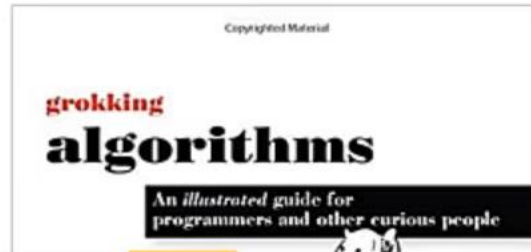
font-variant: normal;

4

MEDIA SCRAPE EXAMPLE PAGE 1

Grokking Algorithms: An illustrated guide for programmers and other curious people

A book by Aditya Bhargava



Algorithms are nothing more than step-by-step procedures for solving problems and most algorithms used by programmers have already been discovered, tested, and proven. Those wanting to take a hard pass on Knuth's brilliant but impenetrable theories, and the dense multi-page proofs found in most textbooks, will want to read Grokking Algorithms. This fully-illustrated and engaging guide makes it easy to learn how to use algorithms effectively.

Grokking Algorithms is a disarming take on a core computer science topic and shows readers how to apply common algorithms to practical problems faced in the day-to-day life of a programmer. It starts with problems like sorting and searching and builds up skills in thinking algorithmically. Then it tackles more complex concerns such as data compression or artificial intelligence. Whether writing business software, video games, mobile apps, or system utilities, readers will learn

Application Security Lighthouse Media

Application

- Manifest
- Service Workers
- Clear storage

Storage

- Local Storage
 - http://www.howtowebscrape.com
- Session Storage
- IndexedDB
- Web SQL
- Cookies

Cache

- Cache Storage
- Application Cache

Background Services

- Background Fetch
- Background Sync
- Notifications
- Payment Handler
- Periodic Background Sync

Console What's New x

Highlights from the Chrome 87 update

Local Storage

[Learn more](#)

Web scraping media with Python

We are Pluralsight - YouTube


youtube.com/watch?v=-HTcEYIzA6c

Incognito

YouTube

Search

SIGN IN



Elements | Console | Sources | Network | Performance | Memory | Application | Security | Lighthouse | Media

Players

We are Pluralsight - YouTube

Properties | Events | Messages | Timeline

Resolution

854x356

Playback frame URL

https://www.youtube.com/watch?v=-HTcEYIzA6c

Playback frame title

We are Pluralsight - YouTube

Video Decoder

Track #1

Decoder name

MojoVideoDecoder

Hardware decoder

true

Decrypting demuxer

false

Audio Decoder

Track #1

Decoder name

FFmpegAudioDecoder

Hardware decoder

false

Decrypting demuxer

false

No text tracks

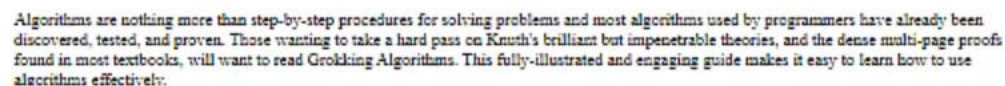
Console | What's New

Highlights from the Chrome 87 update

Source & Element Tabs

Grokking Algorithms: An illustrated guide for programmers and other curious people

A book by Aditya Bhargava



Grokking Algorithms is a disarming take on a core computer science topic and shows readers how to apply common algorithms to practical problems faced in the day-to-day life of a programmer. It starts with problems like sorting and searching and builds up skills in thinking algorithmically. Then it tackles more complex concerns such as data compression or artificial intelligence. Whether writing business software, video games, mobile apps, or system utilities, readers will learn algorithmic techniques for solving problems that they thought were out of reach. By the end of this book, they will know some of the most widely applicable algorithms, as well as how and when to use them.

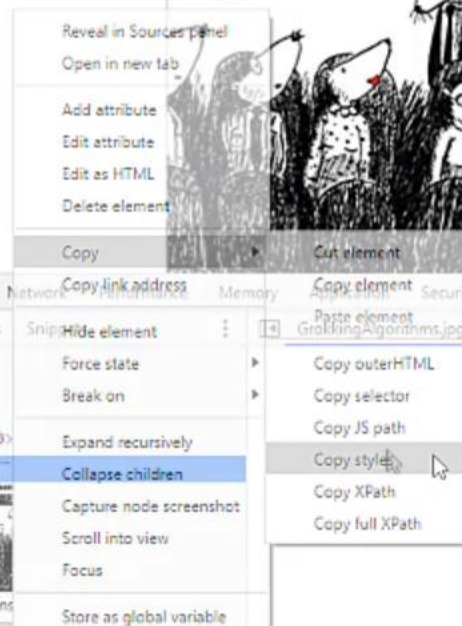
Product details Paperback: 300 pages
Publisher: [Manning Publications](#); 1 edition (31 Dec. 2015)
Language: English
ISBN-10: 1617292230
Product Dimensions: 19 x 1.3 x 23.5 cm

This book and the image on this page are copyright Manning publications. The book can be purchased from Manning here: <https://www.manning.com/books/grokking-algorithms>

The screenshot shows the Chrome DevTools interface. In the 'Elements' panel, the 'center' element is selected, showing its HTML structure: `<html><head></head><body><center></center></body></html>`. The 'Styles' panel on the right shows the 'center' rule with the following styles: `display: block;` and `text-align: -webkit-center;`. A box model diagram is visible at the bottom right of the Styles panel, showing the element's dimensions as 2362.770 x 727.754.

Grokking Algorithms: An illustrated guide for programmers and other curious people

A book by Aditya Bhargava



Grokking Algorithms is a disarming take on a core computer science topic and shows readers how to apply common algorithms to practical problems faced in the day-to-day life of a programmer. It starts with problems like sorting and searching and builds up skills in thinking algorithmically. Then it tackles more complex concerns such as data compression or artificial intelligence. Whether writing business software, video games, mobile apps, or system utilities, readers will learn algorithmic techniques for solving problems that they thought were out of reach. By the end of this book, they will know some of the most widely applicable algorithms, as well as how and when to use them.

Publisher: **Manning Publications**; 1 edition (31 Dec. 2015)

ISBN-10: 1617292230

Product Dimensions: 19 x 1.3 x 23.5 cm

<https://www.manning.com/books/grokking-algorithms>

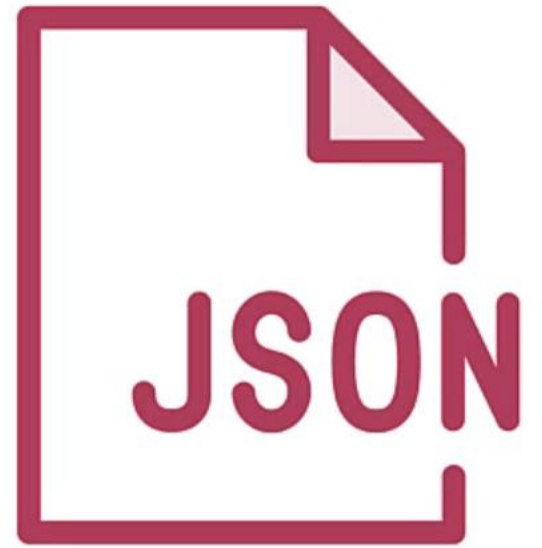
<https://www.manning.com/books/grokking-algorithms>



```
Elements Console Sources Network Performance Memory Application Security Lighthouse
<h1>Slideshow / Carousel of Images</h1>
<div style="max-width:1000px;min-width:250px;position:relative;margin:auto;">
  <div class="mySlides fade" style="display: none;">...</div>
  <div class="mySlides fade" style="display: none;">...</div>
  <div class="mySlides fade" style="display: none;">...</div>
  <div class="mySlides fade" style="display: block;">
    <div class="numbertext">4 / 4</div>
    
    <div class="text"></div>
  </div>
```

Extract with Strings & Regex

Data Formats



Extraction Methods

String manipulation

Regex

CSS

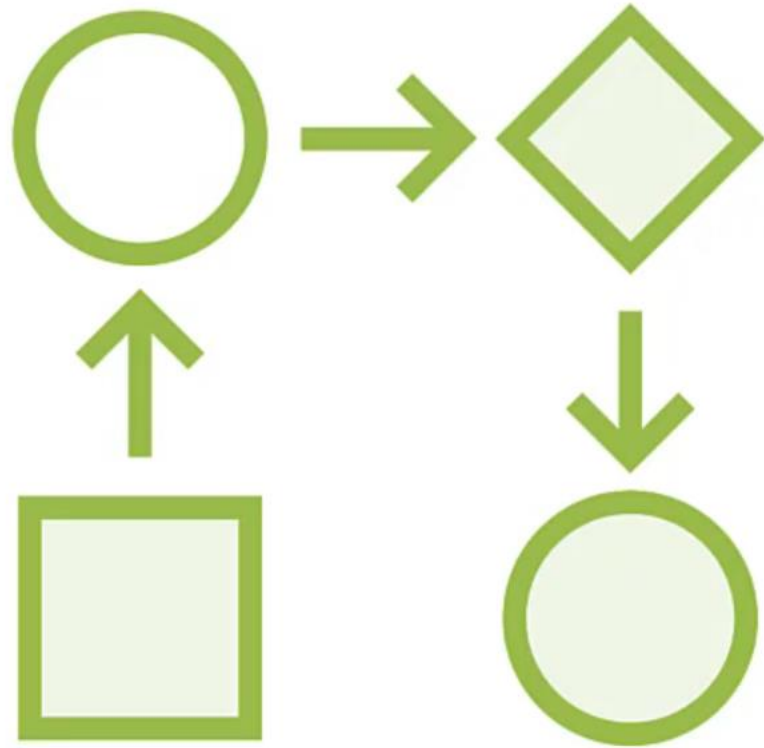
XPath

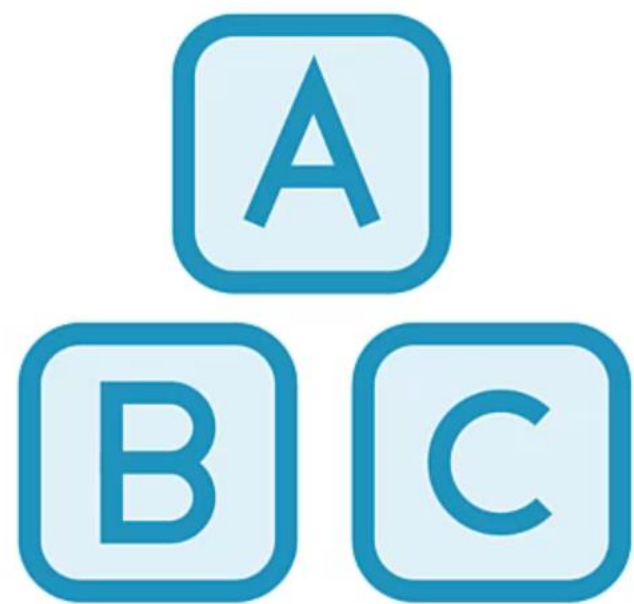
This video

Next video



Webscrape Workflow





find

pos

sub-string

upper

lower

```
HtmlString = "this is an example  
string \n <a  
href='/media/podcast.mp3'>Download  
podcast</a>"
```

◀ Declare the string

```
StringPositionStart =  
HtmlString.find("href")
```

◀ Locate starting point

```
StringPositionEnd =  
HtmlString.find("'>")
```

◀ Locate end point

```
DownloadLink =  
HtmlString[StringPositionStart:StringP  
ositionEnd]
```

◀ Extract sub-string

Regular Expression

Also called regex or regexp, it is a sequence of characters that define a search pattern.



RegEx:

- Useful tool
- Use with care
- Complex

```
import re
```

```
HtmlString = "this is an example  
string \n <a  
href='/media/podcast.mp3'>Download  
podcast</a>"
```

```
DownloadLink =  
re.findall('<a.+?\s*href\s*=\s*["\']?([^\s">]+)["\']?', HtmlString)
```

```
print(DownloadLink[0])
```

```
bash> /media/podcast.mp3
```

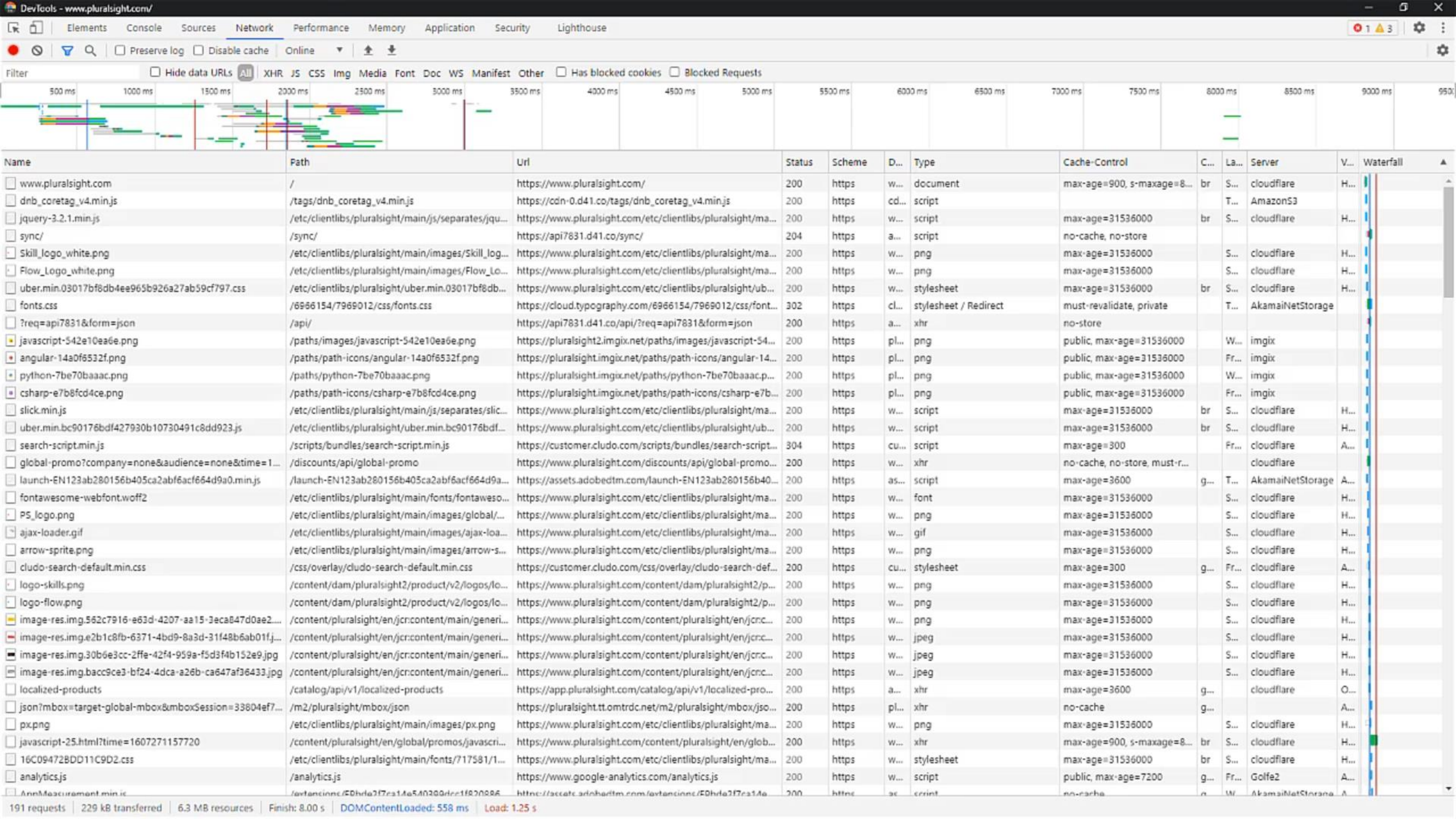
◀ Declare the library

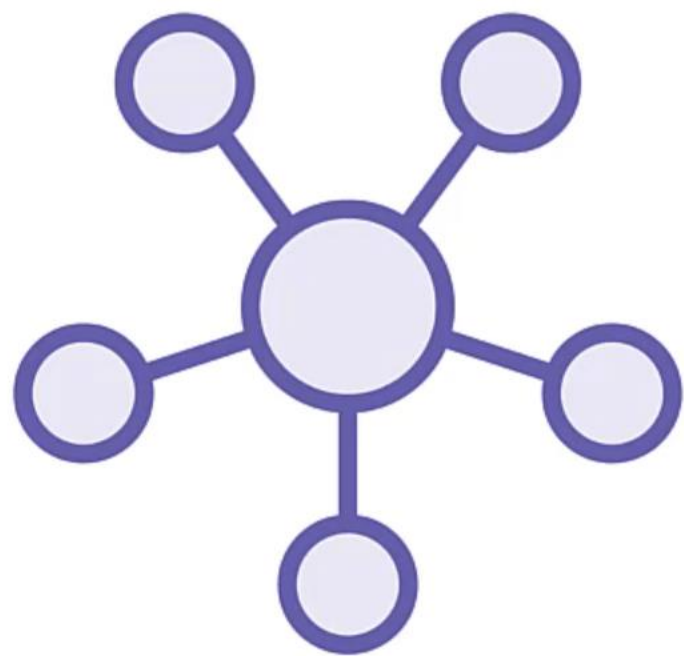
◀ Declare the string

◀ Define RegEx expression

◀ Extract result

Network Tab





Identify the element
Check page source

CSS Links

```
<body>
<br/>
  <center>

    <h3>MEDIA SCRAPE EXAMPLE PAGE 1</h3>

    <h1>Grokking Algorithms: An illustrated guide for programmers and other curious people </h1>

    <div> <font color="gray"> <h2><i> A book by <span id="author">Aditya Bhargava</span> </i></h2> </font> </div>


    <br/> <br/>

<div class="imgDiv">

</div>


  </center>

</body>
```

No IMG tag?

MEDIA SCRAPE EXAMPLE PAGE 1

Grokking Algorithms: An illustrated guide for programmers and other curious people

A book by Aditya Bhargava

The screenshot shows a web browser's developer tools interface. The 'Network' tab is active, displaying a list of requests. The first request, 'GrokkingAlgorithms.jpg', is selected. The 'Response' tab is open, showing the raw response data. The response is a CSS rule for '.imgDiv'.

Network tab details:

- Filter: ☐ Hide data URLs ☒ All
- XHR JS CSS Img Media Font Doc WS Manifest Other ☐ Has blocked cookies ☐ Blocked Requests
- Timeline: 10 ms, 20 ms, 30 ms, 40 ms, 50 ms, 60 ms, 70 ms, 80 ms, 90 ms, 100 ms

Selected request: GrokkingAlgorithms.jpg

Response tab details:

```
1 .imgDiv {  
2   background-image: url('GrokkingAlgorithms.jpg');  
3   background-repeat: no-repeat;  
4   background-position: center center;  
5   background-size: contain;  
6   height: 399px; width: 499px;  
7 }
```

Raw response data

3 requests | 328 B transferred | 51.2 kB resources | Finish: 43 ms | DOM | Line 2, Column 1



Basic tools:

- Page source
- Element inspector
- Application view
- Network traffic

Extract with CSS & XPath

Webpage Construction





CSS Selectors:

- Extract data
- Multiple languages



P	Paragraph
H1	Heading
#	Identifier
.	Class

Selectors



Syntax varies

Always research!

XPath

XPath (XML Path Language) is a query language for selecting nodes from an XML document.

Selectors



Can be slower

Use when needed

CSS Versus XPath

p

//p

CSS Versus XPath

#foo

//*[@id=foo]

Media Content Types

Types of Media



IMAGE



VIDEO



AUDIO



DOC

File Extensions



IMAGE

JPG

PNG



VIDEO

AVI

MP4



AUDIO

MP3

WAV



DOC

PDF

RTF

Request & Response

X Headers Preview Response Initiator Timing

▼ General

Request URL: `http://www.howtowscape.com/examples/media/images/GrokkingAlgorithms.jpg`

Request Method: GET

Status Code: 200 OK (from memory cache)

Remote Address:

Referrer Policy: `strict-origin-when-cross-origin`

▼ Response Headers

Accept-Ranges: bytes

Connection: Keep-Alive

Content-Length: 50424

Content-Type: image/jpeg

ETag: "c4f8-58850c8f7d680"

Keep-Alive: timeout=5, max=99

Last-Modified:

Server: Apache

X-SERVER: 3239

▼ Request Headers

Provisional headers are shown. Disable cache to see full headers.

DNT: 1

Referer: `http://www.howtowscape.com/examples/media1.html`

User-Agent: `Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/88.0.4324.27 Safari/537.36`

Content
Types

Application

Audio

Image

Multi-part

Text

Video

Vnd

Extracting Links - Demo
