

# Searching for Elements in the Parse Tree

---



**Janani Ravi**

CO-FOUNDER, LOONYCORN

[www.loonycorn.com](http://www.loonycorn.com)

# Overview

**Searching the tree using advanced functionality**

**Searching parents, descendants and siblings**

**Understanding differences and similarities between XML and HTML**

**Finding page elements using XPath and CSS selectors**

# Selector

Specification of what HTML elements ought to be selected for processing.

# CSS Selectors

CSS selectors are supported using the Soup Sieve Python library which is installed along with BeautifulSoup

# XML and XPath

---

# XML (eXtensible Markup Language)

Markup language with some similarities to HTML, but commonly used for information transfer (like CSV or JSON) rather than for display

# Beautiful Soup and XML

The lxml parser allows BeautifulSoup to parse and work with XML documents

# XML vs. HTML

## XML

**eXtensible Markup Language**

**Used as a data serialization  
and transfer protocol**

## HTML

**Hypertext Markup Language**

**Used almost exclusively for web  
pages**



# XML vs. HTML

## XML

**Must be well-formed - no room for missing tags and errors**

**User-defined tags, meaningful based on content**

**Quite human-readable, especially when compared to CSV or JSON**

## HTML

**Browsers are quite lax about ill-formed markup such as missing tags**

**Tags are understood and interpreted by web browsers**

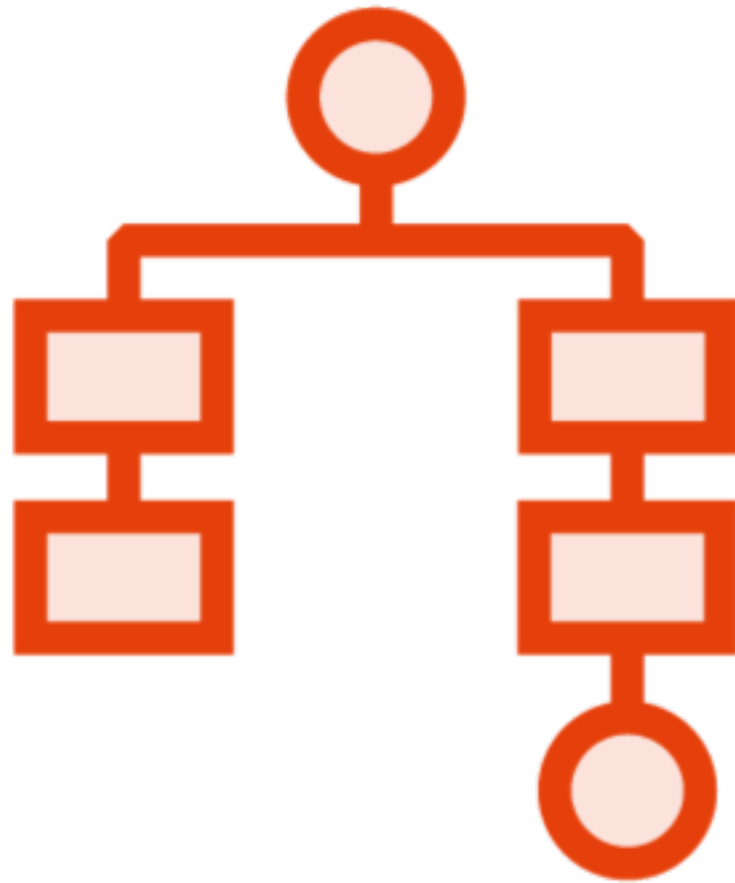
**Production-grade web content is almost never human readable**

Despite these differences,  
XML and HTML are similar  
enough in structure that  
XPath works with both

# XPath

A query language that can be used to select nodes from an XML (or HTML) document

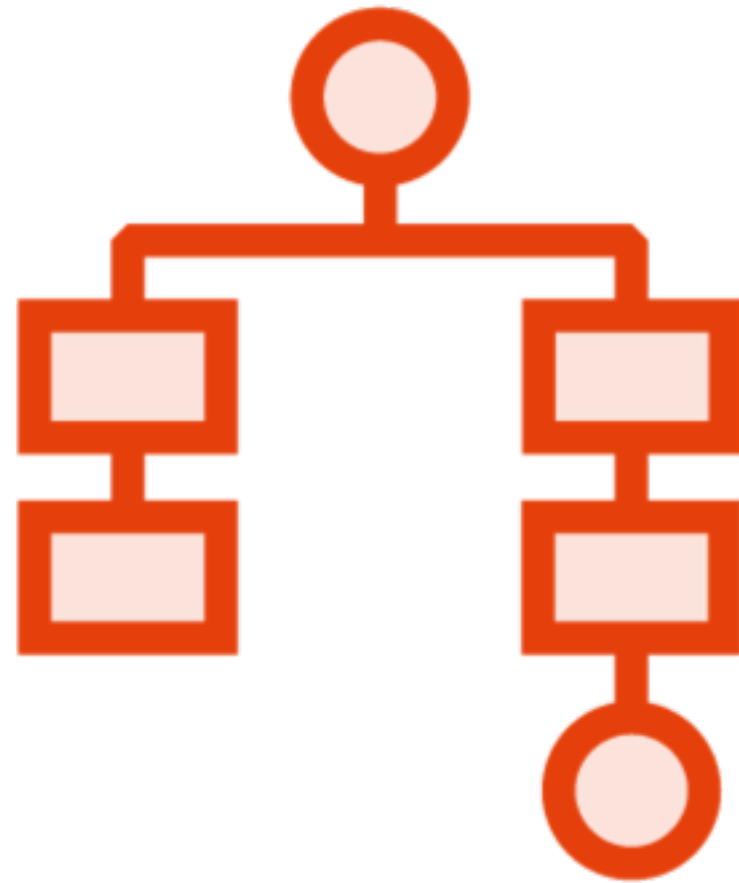
# XPath



**Given a parse tree, XPath is a way to:**

- Select specific nodes based on their location relative to the root
- Traverse up and down the tree
- Select all nodes that meet a predicate

# XPath



**XPath syntax is relatively intuitive**

**Use notation broadly similar to Unix file paths**

**Much additional functionality for predicates and other uses**

# Beautiful Soup and XML

The lxml parser allows BeautifulSoup to parse and work with XML documents

Demo

**Using advanced search functionality in  
Beautiful Soup**

Demo

**Applying CSS selectors using Soup  
Sieve**



Demo

**Applying XPath selectors using lxml**

# Summary

**Searching the tree using advanced functionality**

**Searching parents, descendants and siblings**

**Understanding differences and similarities between XML and HTML**

**Finding page elements using XPath and CSS selectors**