Web Scraping

PIERIAN DATA

Complete Python Bootcamp

- Web scraping is a general term for techniques involving automating the gathering of data from a website.
- In this section we will learn how to use Python to conduct web scraping tasks, such as downloading images or information off a website.

- In order to web scrape with Python we need to understand the basic concepts of how a website works.
- When a browser loads a website, the user gets to see what is known as the “front-end” of the website.

Complete Python Bootcamp

WIKIPEDIA
The Free Encyclopedia

Complete Python Bootcamp

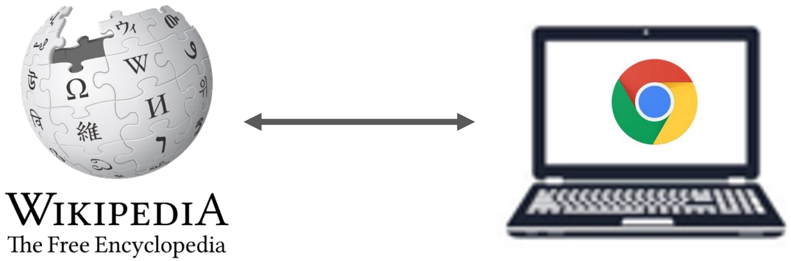
WIKIPEDIA
The Free Encyclopedia

Complete Python Bootcamp

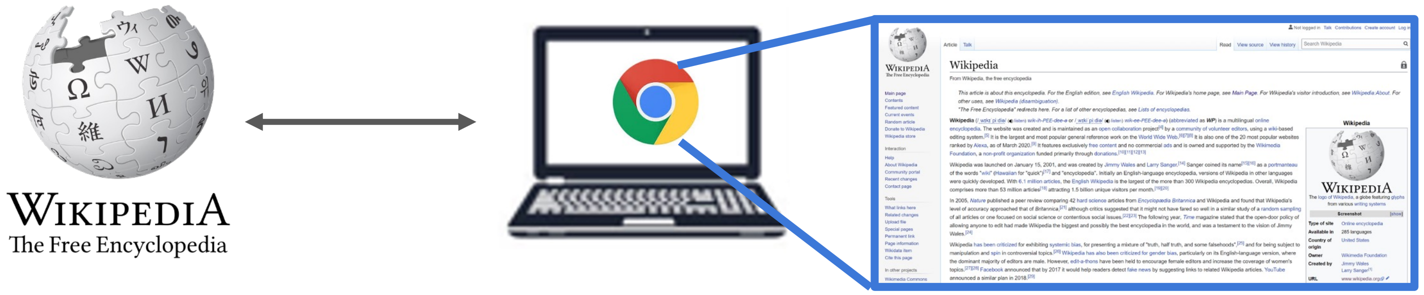
WIKIPEDIA
The Free Encyclopedia

Complete Python Bootcamp

Complete Python Bootcamp



Complete Python Bootcamp



Not logged in Talk Contributions Create account Log in

Article Talk Read View source View history Search Wikipedia

WIKIPEDIA The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction

Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools

What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

In other projects

Wikimedia Commons

Wikipedia

From Wikipedia, the free encyclopedia

This article is about this encyclopedia. For the English edition, see English Wikipedia. For Wikipedia's home page, see Main Page. For Wikipedia's visitor introduction, see Wikipedia:About. For other uses, see Wikipedia (disambiguation).

"The Free Encyclopedia" redirects here. For a list of other encyclopedias, see Lists of encyclopedias.

Wikipedia (/ˌwɪkɪˈpiːdiə/ (listen) *wik-ih-PEE-dee-ə* or /ˌwɪkiˈpiːdiə/ (listen) *wik-ee-PEE-dee-ə*) (abbreviated as ***WP***) is a multilingual online encyclopedia. The website was created and is maintained as an open collaboration project[4] by a community of volunteer editors, using a wiki-based editing system.[5] It is the largest and most popular general reference work on the World Wide Web.[6][7][8] It is also one of the 20 most popular websites ranked by Alexa, as of March 2020.[9] It features exclusively free content and no commercial ads and is owned and supported by the Wikimedia Foundation, a non-profit organization funded primarily through donations.[10][11][12][13]

Wikipedia was launched on January 15, 2001, and was created by Jimmy Wales and Larry Sanger.[14] Sanger coined its name[15][16] as a portmanteau of the words "wiki" (Hawaiian for "quick")[17] and "encyclopedia". Initially an English-language encyclopedia, versions of Wikipedia in other languages were quickly developed. With 6.1 million articles, the English Wikipedia is the largest of the more than 300 Wikipedia encyclopedias. Overall, Wikipedia comprises more than 53 million articles[18] attracting 1.5 billion unique visitors per month.[19][20]

In 2005, *Nature* published a peer review comparing 42 hard science articles from *Encyclopædia Britannica* and Wikipedia and found that Wikipedia's level of accuracy approached that of *Britannica*,[21] although critics suggested that it might not have fared so well in a similar study of a random sampling of all articles or one focused on social science or contentious social issues.[22][23] The following year, *Time* magazine stated that the open-door policy of allowing anyone to edit had made Wikipedia the biggest and possibly the best encyclopedia in the world, and was a testament to the vision of Jimmy Wales.[24]

Wikipedia has been criticized for exhibiting systemic bias, for presenting a mixture of "truth, half truth, and some falsehoods",[25] and for being subject to manipulation and spin in controversial topics.[26] Wikipedia has also been criticized for gender bias, particularly on its English-language version, where the dominant majority of editors are male. However, edit-a-thons have been held to encourage female editors and increase the coverage of women's topics.[27][28] Facebook announced that by 2017 it would help readers detect fake news by suggesting links to related Wikipedia articles. YouTube announced a similar plan in 2018.[29]

Wikipedia

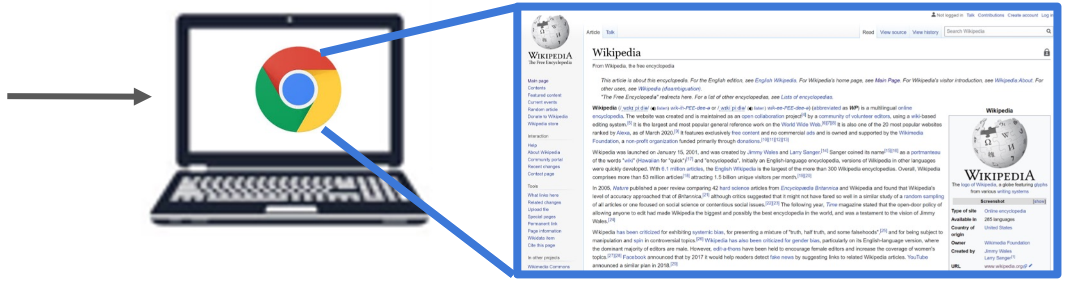
WIKIPEDIA

The logo of Wikipedia, a globe featuring glyphs from various writing systems

Screenshot	[show]
Type of site	Online encyclopedia
Available in	285 languages
Country of origin	United States
Owner	Wikimedia Foundation
Created by	Jimmy Wales Larry Sanger[1]
URL	www.wikipedia.org

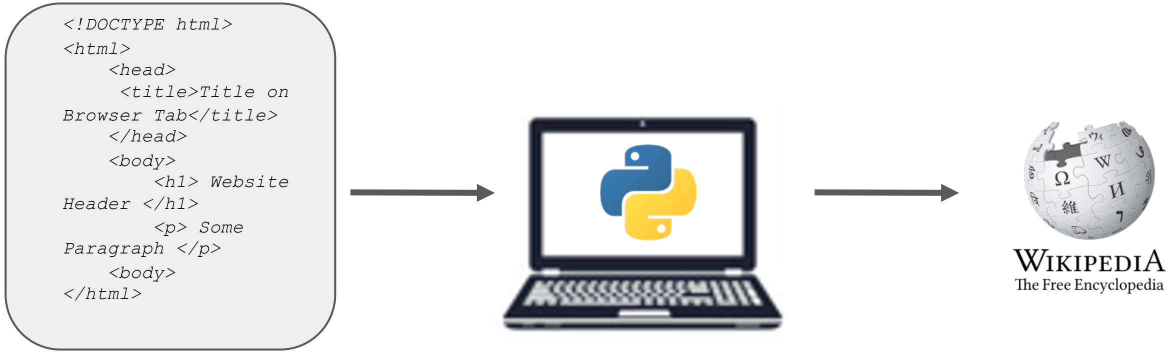
Complete Python Bootcamp

```
<!DOCTYPE html>
<html>
    <head>
     <title>Title on
Browser Tab</title>
    </head>
    <body>
        <h1> Website
Header </h1>
        <p> Some
Paragraph </p>
    <body>
</html>
```



PIERIAN DATA

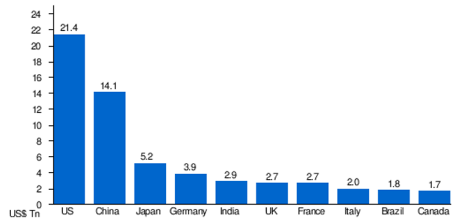
Complete Python Bootcamp



```
<!DOCTYPE html>
<html>
    <head>
     <title>Title on
Browser Tab</title>
    </head>
    <body>
        <h1> Website
Header </h1>
        <p> Some
Paragraph </p>
    <body>
</html>
```

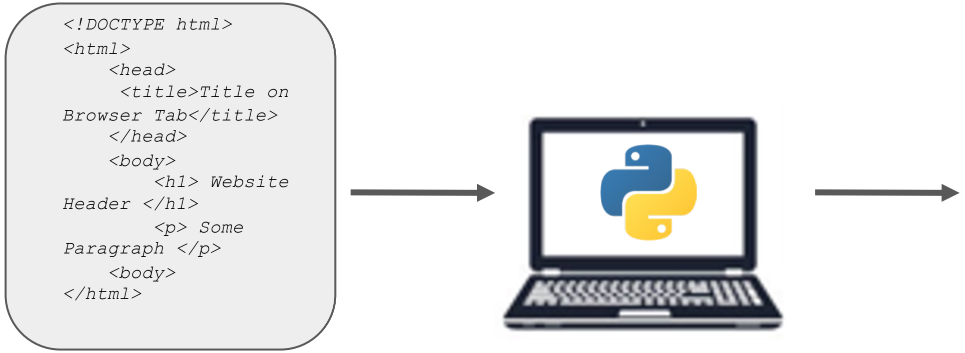
Complete Python Bootcamp

```
<!DOCTYPE html>
<html>
    <head>
     <title>Title on
Browser Tab</title>
    </head>
    <body>
        <h1> Website
Header </h1>
        <p> Some
Paragraph </p>
    <body>
</html>
```



PIERIAN DATA

Complete Python Bootcamp



```
<!DOCTYPE html>
<html>
    <head>
     <title>Title on
Browser Tab</title>
    </head>
    <body>
        <h1> Website
Header </h1>
        <p> Some
Paragraph </p>
    <body>
</html>
```

[“Germany”, “France”, “Spain”]

Complete Python Bootcamp

- Main things we need to understand
    - Rules of Web Scraping
    - Limitations of Web Scraping
    - Basic HTML and CSS

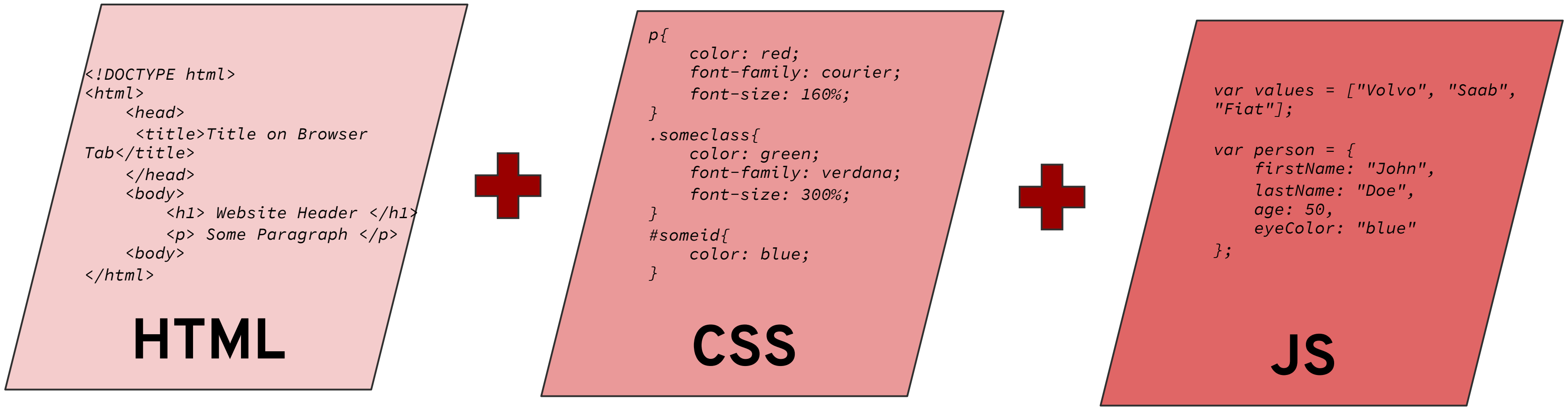
Complete Python Bootcamp

- Rules of Web Scraping
    - Always try to get permission before scraping!
    - If you make too many scraping attempts or requests your IP Address could get blocked!
    - Some sites automatically block scraping software.

Complete Python Bootcamp

- Limitations of Web Scraping
    - In general every website is unique, which means every web scraping script is unique.
    - A slight change or update to a website may completely break your web scraping script.

Main front end components of a website



```
<!DOCTYPE html>
<html>
    <head>
     <title>Title on Browser
Tab</title>
    </head>
    <body>
        <h1> Website Header </h1>
        <p> Some Paragraph </p>
    <body>
</html>
```

```
p{
    color: red;
    font-family: courier;
    font-size: 160%;
}
.someclass{
    color: green;
    font-family: verdana;
    font-size: 300%;
}
#someid{
    color: blue;
}
```

```
var values = ["Volvo", "Saab",
"Fiat"];

var person = {
    firstName: "John",
    lastName: "Doe",
    age: 50,
    eyeColor: "blue"
};
```

Complete Python Bootcamp

- When viewing a website, the browser doesn’t show you all the source code behind the website, instead it shows you the HTML and some CSS and JS that the website sends to your browser.

Complete Python Bootcamp

- HTML is used to create the basic structure and content of a webpage
- CSS is used for the design and style of a web page, where elements are placed and how it looks
- JavaScript is used to define the interactive elements of a webpage

- For effective basic web scraping we only need to have a basic understanding of HTML and CSS.
- Python can view these HTML and CSS elements programmatically, and then extract information from the website.
- Let’s explore HTML and CSS in more detail.

- HTML is Hypertext Markup Language and is present on every website on the internet.
- You can right-click on a website and select “View Page Source” to get an example.
- Let’s see a small example of HTML code.

Complete Python Bootcamp

```
<!DOCTYPE html>
<html>
    <head>
     <title>Title on Browser Tab</title>
    </head>
    <body>
        <h1> Website Header </h1>
        <p> Some Paragraph </p>
    <body>
</html>
```

Complete Python Bootcamp

```
<!DOCTYPE html>
<html>
    <head>
     <title>Title on Browser Tab</title>
    </head>
    <body>
        <h1> Website Header </h1>
        <p> Some Paragraph </p>
    <body>
</html>
```

Complete Python Bootcamp

```
<!DOCTYPE html>
<html>
    <head>
     <title>Title on Browser Tab</title>
    </head>
    <body>
        <h1> Website Header </h1>
        <p> Some Paragraph </p>
    <body>
</html>
```

Complete Python Bootcamp

```
<!DOCTYPE html>
<html>
    <head>
     <title>Title on Browser Tab</title>
    </head>
    <body>
        <h1> Website Header </h1>
        <p> Some Paragraph </p>
    <body>
</html>
```

- CSS stands for Cascading Style Sheets.
- CSS gives “style” to a website, such as changing colors and fonts.
- CSS uses tags to define what html elements will be styled.

Complete Python Bootcamp

```
<!DOCTYPE html>
   <html>
            <head>
     <link rel="stylesheet" href="styles.css">
     <title>Some Title</title>
     </head>
     <body>
          <p id='para2'> Some Text </p>
     <body>
</html>
```

Complete Python Bootcamp

```
<!DOCTYPE html>
   <html>
            <head>
     <link rel="stylesheet" href="styles.css">
     <title>Some Title</title>
     </head>
     <body>
          <p id=‘para2’> Some Text </p>
     <body>
</html>
```

Complete Python Bootcamp

```
<!DOCTYPE html>
   <html>
            <head>
     <link rel="stylesheet" href="styles.css">
     <title>Some Title</title>
     </head>
     <body>
          <p id='para2'> Some Text </p>
     <body>
</html>
```

Example of the style.css file:

```
#para2 {
    color: red;
}
```

Complete Python Bootcamp

```
<!DOCTYPE html>
<html>
          <head>
    <link rel="stylesheet" href="styles.css">
    <title>Some Title</title>
    </head>
    <body>
        <p class=‘cool’> Some Text </p>
    <body>
</html>
```

Example of the style.css file:

```
.cool {
    color: red;
      font-family: verdana;
}
```

Complete Python Bootcamp

```
p{
    color: red;
    font-family: courier;
    font-size: 160%;
}
.someclass{
    color: green;
    font-family: verdana;
    font-size: 300%;
}
#someid{
    color: blue;
}
```

Complete Python Bootcamp

- Don’t worry about memorizing this! We’ll see lots of examples, main ideas to note:
    - HTML contains the information
    - CSS contains the styling
    - We can use HTML and CSS tags to locate specific information on a page

Complete Python Bootcamp

- To web scrape with Python we can use the BeautifulSoup and requests libraries.
- These are external libraries outside of Python so you need to install them with either conda or pip at your command line.

Complete Python Bootcamp

- Directly at your command line use:
  - pip install requests
  - pip install lxml
  - pip install bs4
- Or for Anaconda distributions, use conda install instead of pip install.

Complete Python Bootcamp

- Let’s work through some examples of web scraping with Python!

PIERIAN DATA

Setting Up
For Web Scraping

Complete Python Bootcamp

- Install the necessary libraries
- Explore how to inspect elements and view source of a webpage
- Note: We will suggest you use Chrome so you can follow along exactly as we do, but these tools are available in all major browsers.

Grabbing a Page Title

Grabbing All Elements of a Class

- We previously mentioned a big part of web scraping with the BeautifulSoup library is figuring out what string syntax to pass into the soup.select() method.
- Let’s go through a table with some common examples (these make a lot of sense if you know CSS syntax)

Syntax	Match Results
soup.select(‘div’)	All elements with ‘div’ tag
soup.select(‘#some_id’)	Elements containing id=’some_id’
soup.select(‘.some_class’)	Elements containing class = ‘some_class’
soup.select(‘div span’)	Any elements named span within a div element.
soup.select(‘div > span’)	Any elements named span **directly** within a div element, with nothing in between.

PIERIAN DATA

Grabbing an Image

PIERIAN DATA

- Now that we understand how to grab text information based on tags and element names, let’s explore how to grab images from a website.
- Images on a website typically have their own URL link (ending in .jpg or .png)

- Beautiful Soup can scan a page, locate the <img> tags and grab these URLs.
- Then we can download the URLs as images and write them to the computer.
- Note: You should always check copyright permission before downloading and using an image from a website.

Working with Multiple Pages and Items

Complete Python Bootcamp

- We’ve seen how to grab elements one at a time, but realistically, we want to be able to grab multiple elements, most likely across multiple pages.
- This is where we can combine our prior python knowledge with the web scraping libraries to create powerful scripts!

PIERIAN DATA

Complete Python Bootcamp

- We will use a site specifically designed to practice web scraping: **www.toscrape.com**
- We will practice grabbing elements across multiple pages.
- Let’s get started!

Working with Multiple Pages

Web Scraping Exercises Overview

PIERIAN DATA

Web Scraping Exercises Solutions

PIERIAN DATA

Web Scraping Exercises Solutions - Part Two

PIERIAN DATA