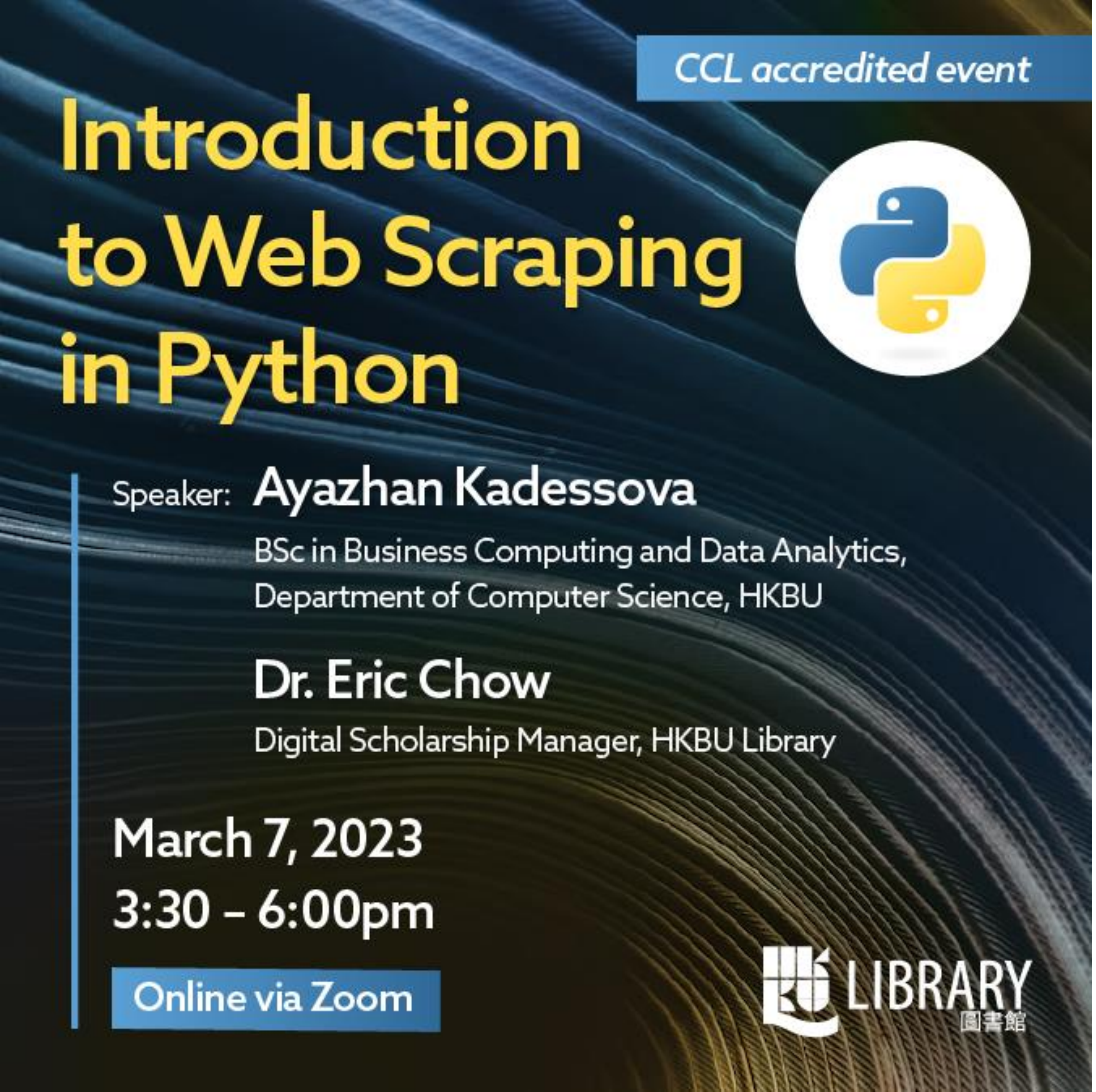


Sign-up / Sign-in for Google Account

Google colab

https://colab.research.google.com/

Expectations

- This is a hands-on workshop; follow along with our coding demo!
- Sign-up / sign-in to a Google account - we will be using **Google CoLab**
- Assumes you already have basic Python programming knowledge (LIB03 - Introduction to Python Programming)

Run-Down

- Warm Up, Google CoLab, review of basic Python programming (20 min)
- Use of Python + BeautifulSoup to scrape *BookDepository.com* (120 min)
- Wrap up, ethics of web scraping, Q&A (10 min)

Unveiling the Basics of HTML, CSS, and Javascript for Web Scraping with Python

Ayazhan Kadessova - Year 2 Business Computing and Data Analytics Student

March, 7

HTML, CSS, Javascript Overview

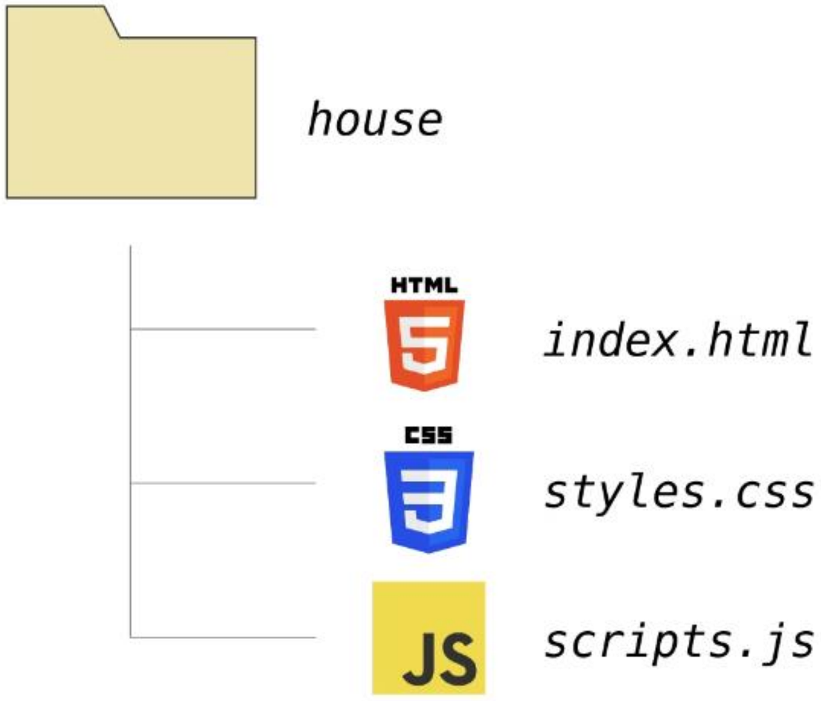
How to Obtain HTML using Requests

Parsing Data with BeautifulSoup

HTML, CSS, Javascript

HTML elements have attributes such as class or id which are used for styling with CSS and adding interactivity with Javascript.

- HTML - text file with a syntax that will tell the browser what content to paint, what text to show, and what resources to download.
- CSS - will format and style the content (i.e., colors, fonts, and many more.
- Javascript - adds functionality and behavior, such as opening a pop-up window.



HTML

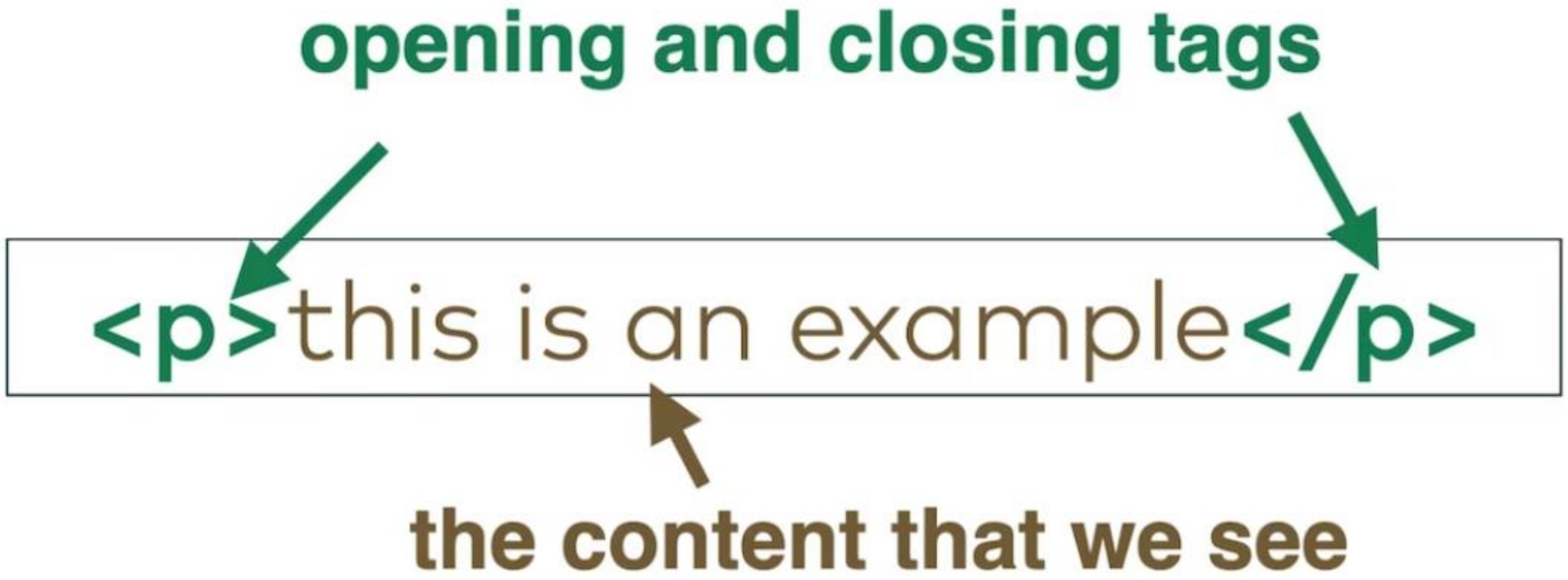
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <link rel="stylesheet" href="./styles.css">
  <title>Document</title>
</head>
<body>
  <h1>This is a first level heading in HTML. With CSS, I will turn this into red co
  <h2>This is a second level heading in HTML. With CSS, I will turn this into blue
  <h3>This is a third level heading in HTML. With CSS, I will turn this into green
  <p>This is a <em>paragragh</em> As you can see, I placed an empahisis on the word
    the background color of the word "paragraph" to black, and its text color  to g
  <p>The main essence of this tutorial is to:</p>
    <ul>
       <li>Show you how to format a web document with HTML</li>
       <li>Show you how to design a web page with CSS</li>
       <li>Show you how to program a web document with JavaScript</li>
    </ul>

  <p>Next, I am going to add the following two numbers and display the result, all
    <p>First number:<span id= "firstNum">2</span> <br></p>
    <p>Second number: <span id= "secondNum">7</span> </p>
    <p>Therefore, the sum of the two of those numbers is: <span id= "answer">(placel
    <input type="button" id="sumButton" value="Click to add!">
</body>
</html>
```

HTML Elements

Elements (links, paragraphs, headings, blocks) are wrapped between tags.

Content is between opening and closing tags. Example: sentence in a paragraph (<p>) element.



HTML Elements and Attributes

HTML elements may also have attributes that contain additional information about the element.

Attributes are defined in the opening tags with the following syntax: attribute name=”attribute value”.

CSS

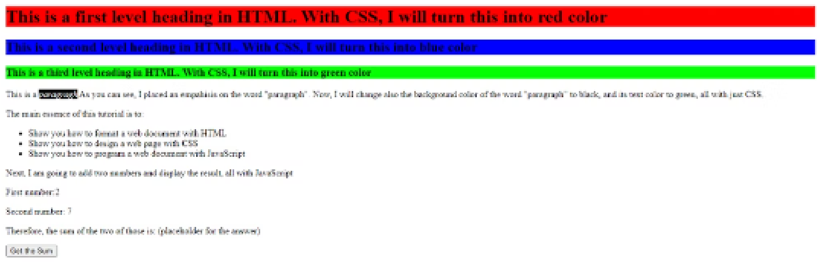
With CSS you can set the colour and background of your elements, as well as the typeface, margins, spacing, padding and so much more.

```css
h1 {
  background-color: #ff0000;
}

h2 {
  background-color: #0000FF;
}

h3 {
  background-color: #00FF00;
}

em {
  background-color: #000000;
  color: #ffffff;
}
```

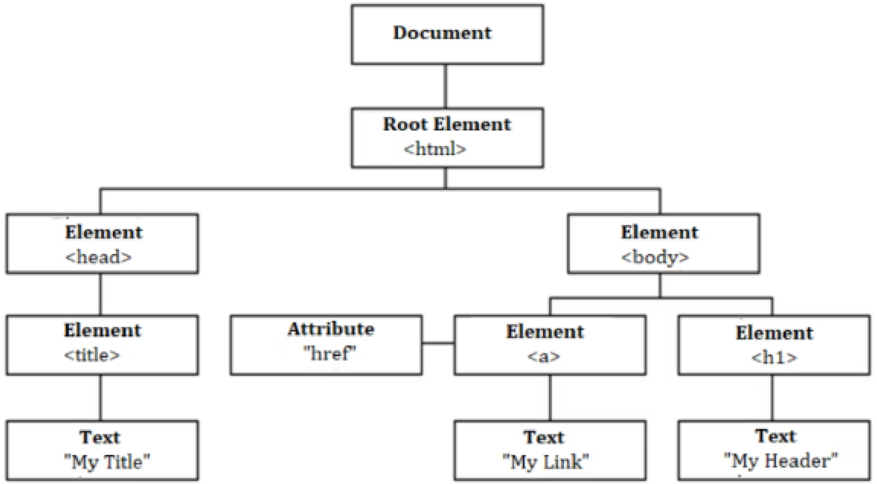


Javascript

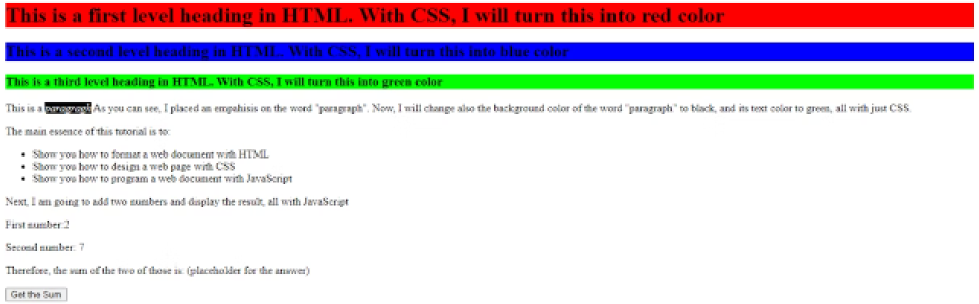
The DOM is a tree-like representation of the web page that gets loaded into the browser.

Thanks to the DOM, we can use methods like `getElementById()` to access elements from our web page.

JavaScript allows you to make your webpage **“think and act”**, which is what programming is all about.



Each element on the web page is represented on the DOM



Clicking the "Get the sum" button will display the sum of 2 and 7

How to Obtain HTML using Requests

Requests makes it easy to make HTTP requests and access the response data.

To obtain HTML from a website, you need to make a GET request using the requests library. This will return the HTML of the website in the response.

```python
html = response.content
print(html)
```

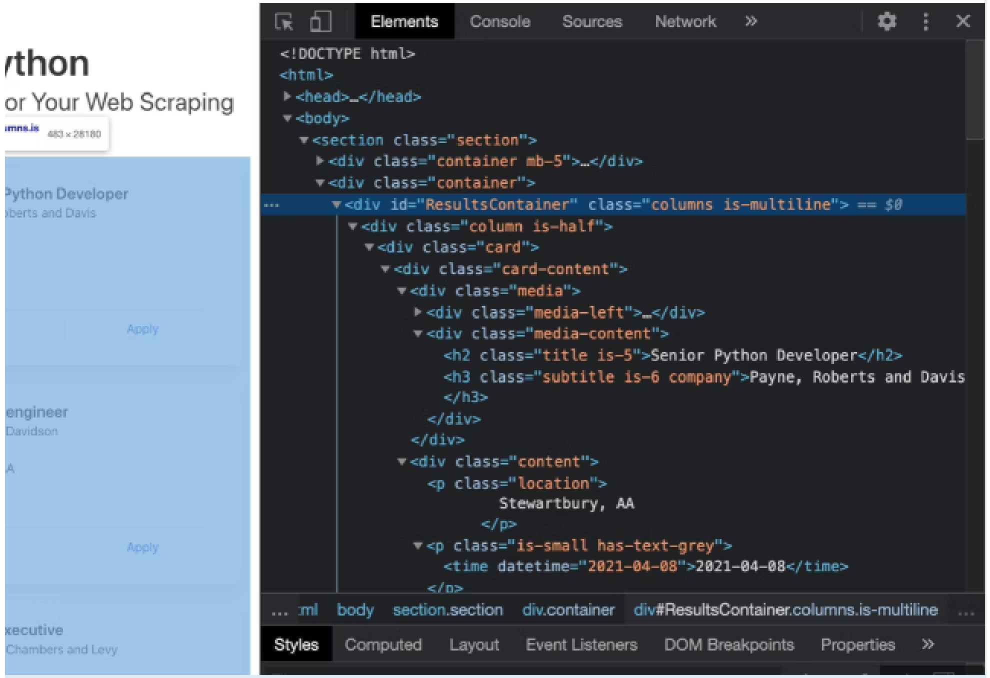
```
b'<!DOCTYPE html>\n<html lang="en">\n<head>\n\n    <link rel="preconnect" href="https://d3ogvdx946i4sr.cloudfront.n
et"/>\n<link rel="dns-prefetch" href="https://d3ogvdx946i4sr.cloudfront.net"/>\n<script type="text/javascript">\n
function csmWidgetStart(widgetName) {\n            if (typeof uet == \'function\') {\n                uet(\'bb\', w
idgetName, {wb: 1});\n            }\n        }\n        function csmWidgetEnd(widgetName) {\n            if (typeof
uet == \'function\') {\n                uex(\'ld\', widgetName, {wb: 1});\n            }\n        }\n    </script>\
n    <noscript>\n        <style>\n            .hide-when-no-js {\n                display: none !important;\n
}\n\n            .show-when-no-js {\n                display: block !important;\n            }\n        </style>\n
</noscript>\n\n    <meta http-equiv="content-type" content="text/html; charset=UTF-8" />\n<meta name="copyright" co
ntent="&copy;2020 Book Depository Ltd." />\n<meta name="author" content="Book Depository" />\n<meta name="viewport"
content="width=device-width, initial-scale=1, maximum-scale=2, user-scalable=1">\n<meta name="google-site-verificat
ion" content="ogOme2fCVYq3eJxsHzQiMuXsUIRVwYrtI2Xn-U1nNpc" />\n<meta name="msvalidate.01" content="D45E907CC9A963F7
8BD3129AAAAFE4F0" />\n<meta http-equiv="X-UA-Compatible" content="IE=edge; charset=UTF-8"> <meta name="description"
content="See our top 1000 bestselling books, charts and future bestsellers. Free delivery worldwide on over 20 mill
ion books at Book Depository." />\n    <meta name="keywords" content="top 1000,bestseller,book,featured,chart,futur
e bestsellers,free delivery worldwide,books" />\n\t<meta name="Revisit-After" content="30 days" />\n\t<link rel="ca
nonical" href="https://www.bookdepository.com/bestsellers" />\n<link rel="alternate" hreflang="x-default" href="htt
ps://www.bookdepository.com/bestsellers" />\n\t<link rel="alternate" hreflang="en" href="https://www.bookdepositor
y.com/bestsellers" />\n\t<link rel="alternate" hreflang="es" href="https://www.bookdepository.com/es/bestsellers" /
>\n\t<link rel="next" href="/bestsellers?page=2" />\n\t<title>\n\tBestselling books online with free delivery at Bo
```

Parsing Data with BeautifulSoup

BeautifulSoup allows to navigate and search the HTML.

It can be used to find specific elements in the HTML, such as tags, attributes, and text.

Once the data is extracted, it can be used for further analysis or manipulation.



Conclusion

Web scraping is a process of extracting data from websites.

Popular libraries: Requests and BeautifulSoup.

Requests is used to obtain HTML from a website.

BeautifulSoup is used to parse the HTML and extract the data.

HTML elements have attributes such as class or id which are used for styling with CSS and adding interactivity with Javascript.

Ethics of Web Scraping

- Be mindful of copyright
- If data is clearly private, then **do not** scrape
- **Do not** overload other people's server
- Use API when available

Examples of API

- https://developers.google.com/books/docs/overview
- https://wiki.harvard.edu/confluence/display/LibraryStaffDoc/LibraryCloud+APIs
- https://github.com/HKBULIB/TVWeek_API