NUS Fintech Society

Machine Learning Department

Training Wing

Session 5: Web Scraping (10/10/2020)

Content

- HTML Basics
- Web Scraping

HTML Basics

HTML

```
<!DOCTYPE html>
<html>
     <head>
          <title>Page Title</title>
     </head>
     <body>
          <h1>This is a Heading</h1>
          <p>This is a paragraph.</p>
     </body>
</html>
```

- The `<!DOCTYPE html>` declaration defines that this document is an HTML5 document

- The `<html>` element is the root element of an HTML page

- The `<head>` element contains meta information about the HTML page

HTML

```
<!DOCTYPE html>
<html>
      <head>
            <title>Page Title</title>
      </head>
      <body>
            <h1>This is a Heading</h1>
            <p>This is a paragraph.</p>
      </body>
</html>
```

- The `<title>` element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab)

- The `<body>` element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.

- The `<h1>` element defines a large heading

- The `<p>` element defines a paragraph

Commonly Used Tags

<h1>	Defines HTML headings
<p>	Defines a paragraph
<ul> &<ol>: <li>	Unordered list, ordered list, list item
<table>:<tr>:<td>	Defines a table, row in table, cell in table
<div>	Defines a section in a document
<span>	<div>This a large main division, with <span>a small bit</span> of spanned text!</div>

Difference between class and id

<div id = “first”></div>

ID’s are unique

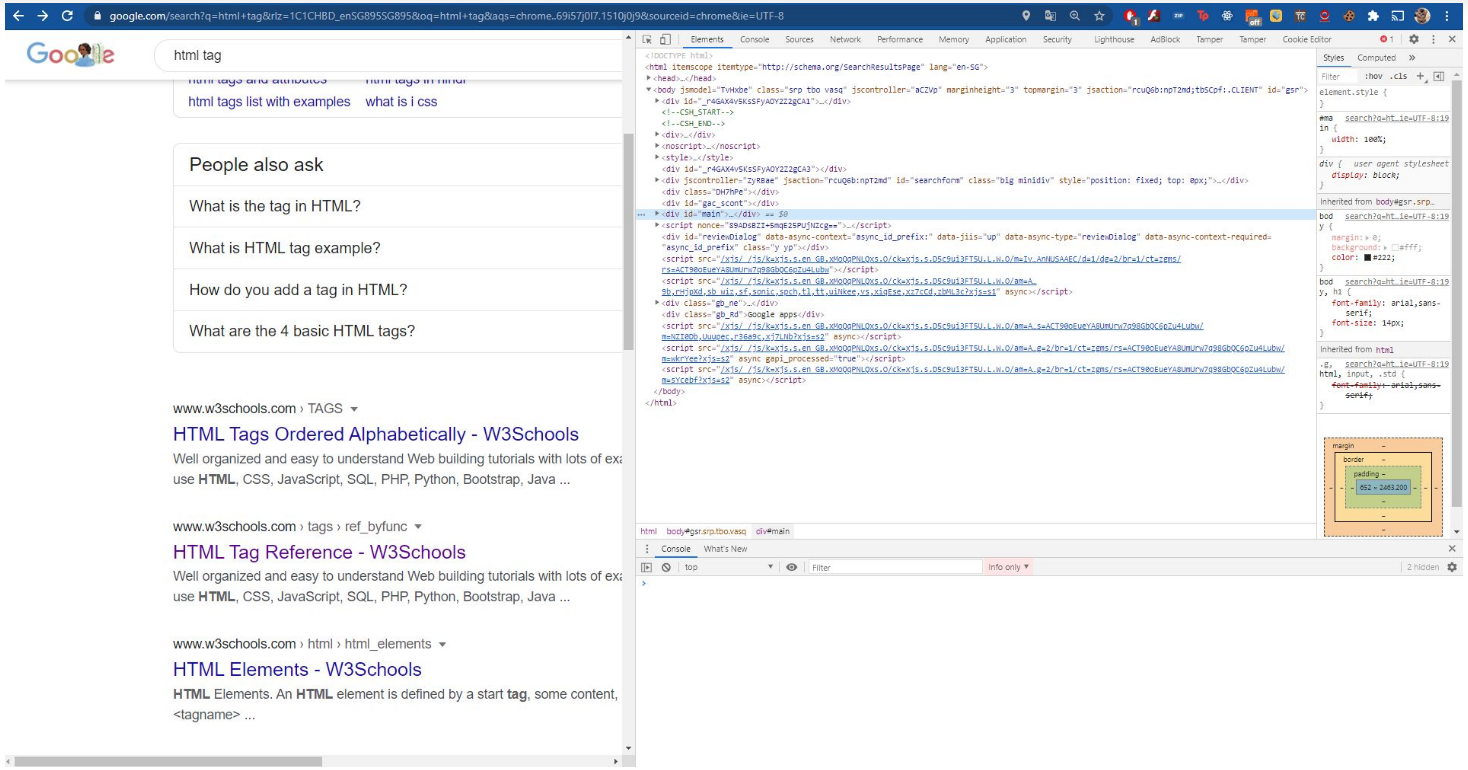
- Each element can have only one ID
- Each page can have only one element with that ID

<div class = “footer”></div>

Classes are not unique

- You can use the same class on multiple elements.
- You can use multiple classes on the same element.

Inspect element



html tags list with examples
what is i css

People also ask

What is the tag in HTML?

What is HTML tag example?

How do you add a tag in HTML?

What are the 4 basic HTML tags?

www.w3schools.com › TAGS

HTML Tags Ordered Alphabetically - W3Schools

Well organized and easy to understand Web building tutorials with lots of exa
use **HTML**, CSS, JavaScript, SQL, PHP, Python, Bootstrap, Java ...

www.w3schools.com › tags › ref_byfunc

HTML Tag Reference - W3Schools

Well organized and easy to understand Web building tutorials with lots of exa
use **HTML**, CSS, JavaScript, SQL, PHP, Python, Bootstrap, Java ...

www.w3schools.com › html › html_elements

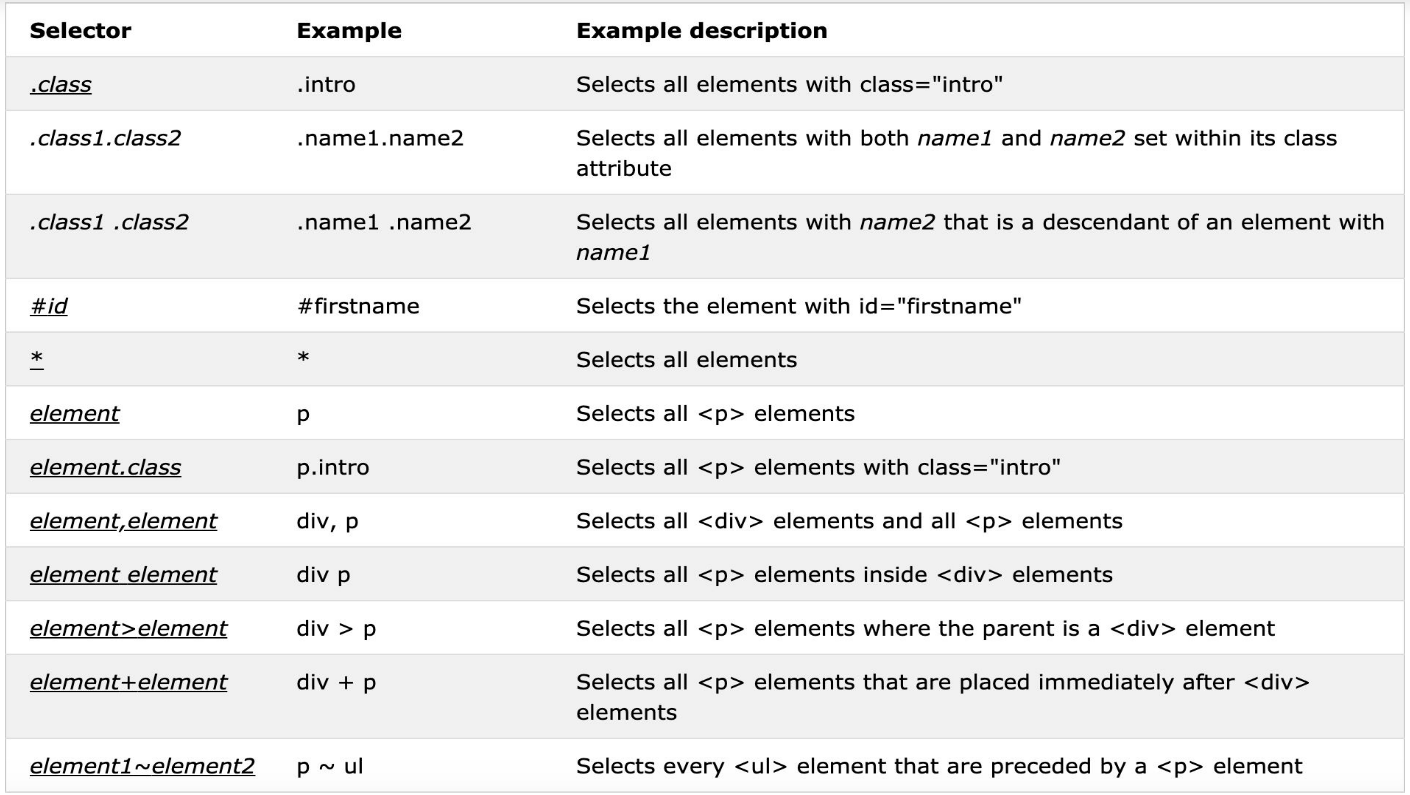
HTML Elements - W3Schools

HTML Elements. An **HTML** element is defined by a start **tag**, some content,
<tagname> ...

```
<!DOCTYPE html>
<html itemscope itemtype="http://schema.org/SearchResultsPage" lang="en-SG">
 ▶<head>…</head>
 ▼<body jsmodel="TvHxbe" class="srp tbo vasq" jscontroller="aCZVp" marginheight="3" topmargin="3" jsaction="rcuQ6b:npT2md;tbSCpf:.CLIENT" id="gsr">
   ▶<div id="_r4GAX4v5KsSFyAOY2Z2gCA1">…</div>
    <!--CSH_START-->
    <!--CSH_END-->
   ▶<div>…</div>
   ▶<noscript>…</noscript>
   ▶<style>…</style>
    <div id="_r4GAX4v5KsSFyAOY2Z2gCA3"></div>
   ▶<div jscontroller="ZyRBae" jsaction="rcuQ6b:npT2md" id="searchform" class="big minidiv" style="position: fixed; top: 0px;">…</div>
    <div class="DH7hPe"></div>
    <div id="gac_scont"></div>
…  ▶<div id="main">…</div> == $0
   ▶<script nonce="89ADsBZI+5mqE25PUjNZcg==">…</script>
    <div id="reviewDialog" data-async-context="async_id_prefix:" data-jiis="up" data-async-type="reviewDialog" data-async-context-required=
    "async_id_prefix" class="y yp"></div>
    <script src="/xjs/_/js/k=xjs.s.en_GB.xMoQqPNLQxs.O/ck=xjs.s.D5c9ui3FT5U.L.W.O/m=Iv…AnNUSAAEC/d=1/dg=2/br=1/ct=zgms/
    rs=ACT90oEueYA8UmUrw7q98GbQC6pZu4Lubw"></script>
    <script src="/xjs/_/js/k=xjs.s.en_GB.xMoQqPNLQxs.O/ck=xjs.s.D5c9ui3FT5U.L.W.O/am=A…
    9b,rHjpXd,sb_wiz,sf,sonic,spch,tl,tt,uiNkee,vs,xiqEse,xz7cCd,zbML3c?xjs=s1" async></script>
   ▶<div class="gb_ne">…</div>
    <div class="gb_Rd">Google apps</div>
    <script src="/xjs/_/js/k=xjs.s.en_GB.xMoQqPNLQxs.O/ck=xjs.s.D5c9ui3FT5U.L.W.O/am=A…s=ACT90oEueYA8UmUrw7q98GbQC6pZu4Lubw/
    m=NZI0Db,Uuupec,r36a9c,xj7LNb?xjs=s2" async></script>
    <script src="/xjs/_/js/k=xjs.s.en_GB.xMoQqPNLQxs.O/ck=xjs.s.D5c9ui3FT5U.L.W.O/am=A…g=2/br=1/ct=zgms/rs=ACT90oEueYA8UmUrw7q98GbQC6pZu4Lubw/
    m=wkrYee?xjs=s2" async gapi_processed="true"></script>
    <script src="/xjs/_/js/k=xjs.s.en_GB.xMoQqPNLQxs.O/ck=xjs.s.D5c9ui3FT5U.L.W.O/am=A…g=2/br=1/ct=zgms/rs=ACT90oEueYA8UmUrw7q98GbQC6pZu4Lubw/
    m=sYcebf?xjs=s2" async></script>
  </body>
</html>
```

```
element.style {
}
#ma   search?q=ht…ie=UTF-8:19
in {
  width: 100%;
}
div {   user agent stylesheet
  display: block;
}
Inherited from body#gsr.srp…
bod   search?q=ht…ie=UTF-8:19
y {
  margin: ▶ 0;
  background: ▶ #fff;
  color: #222;
}
bod   search?q=ht…ie=UTF-8:19
y, h1 {
  font-family: arial,sans-
     serif;
  font-size: 14px;
}
Inherited from html
.g,   search?q=ht…ie=UTF-8:19
html, input, .std {
  ~~font-family: arial,sans-~~
     ~~serif;~~
}
```

CSS Selectors



Selector	Example	Example description
.class	.intro	Selects all elements with class="intro"
.class1.class2	.name1.name2	Selects all elements with both *name1* and *name2* set within its class attribute
.class1 .class2	.name1 .name2	Selects all elements with *name2* that is a descendant of an element with *name1*
#id	#firstname	Selects the element with id="firstname"
*	*	Selects all elements
element	p	Selects all <p> elements
element.class	p.intro	Selects all <p> elements with class="intro"
element,element	div, p	Selects all <div> elements and all <p> elements
element element	div p	Selects all <p> elements inside <div> elements
element>element	div > p	Selects all <p> elements where the parent is a <div> element
element+element	div + p	Selects all <p> elements that are placed immediately after <div> elements
element1~element2	p ~ ul	Selects every <ul> element that are preceded by a <p> element

Intro to Web Scraping

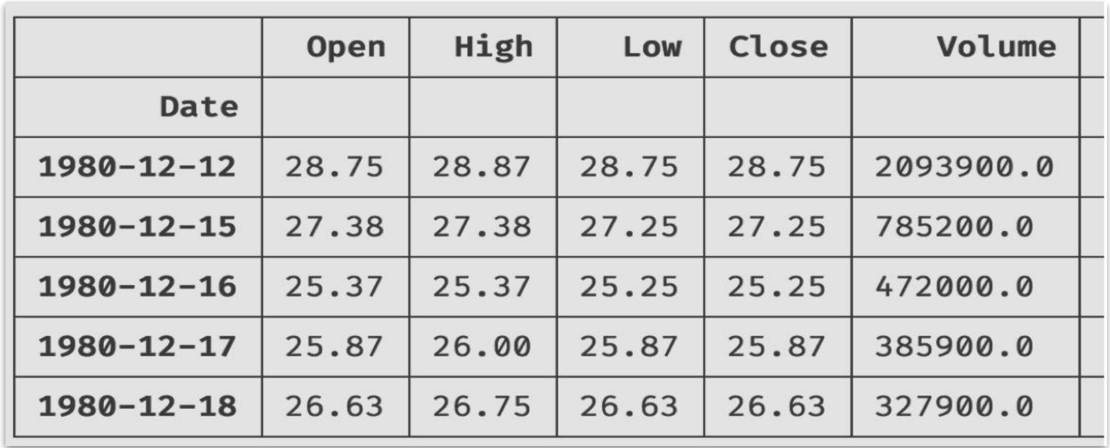
Fetching Data from the Web – API

The most commonly used, widely available source of data is the web. Many organisations (profit and non-profit alike) provide tons of data that come in a multitude of formats.

Fetching Data from the Web – API Example

```
import  quandl
Df = quandl.get("YAHOO/AAPL")

df.head(5)
```



```
             Open   High    Low  Close     Volume
Date
1980-12-12  28.75  28.87  28.75  28.75  2093900.0
1980-12-15  27.38  27.38  27.25  27.25   785200.0
1980-12-16  25.37  25.37  25.25  25.25   472000.0
1980-12-17  25.87  26.00  25.87  25.87   385900.0
1980-12-18  26.63  26.75  26.63  26.63   327900.0
```

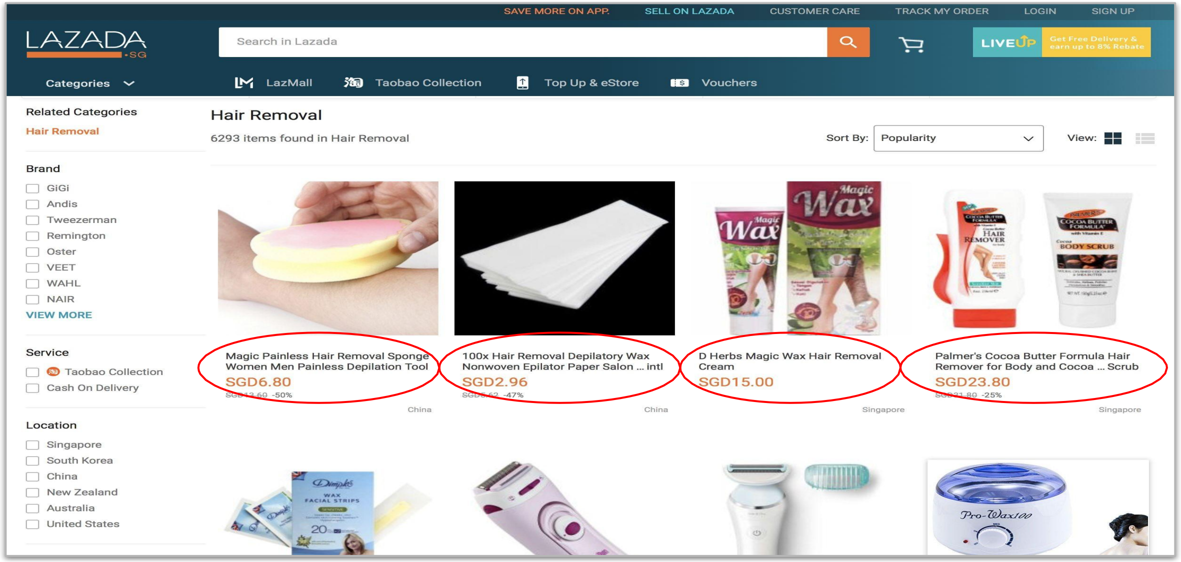
With open source APIs like this, we can easily obtain and play with datasets, even dynamically.

Data providers make our lives extremely easy when they set up clean, user-friendly APIs like this.

But not everybody is so kind!

Fetching Data from the Web – NoAPI?

What if we don't have a nice API / dataset for us to use? The data we want exists on some webpage, but there's no *data.csv* to download?



Hair Removal

6293 items found in Hair Removal

Sort By: Popularity

Magic Painless Hair Removal Sponge Women Men Painless Depilation Tool
SGD6.80
~~SGD13.60~~ -50%
China

100x Hair Removal Depilatory Wax Nonwoven Epilator Paper Salon ... intl
SGD2.96
~~SGD5.62~~ -47%
China

D Herbs Magic Wax Hair Removal Cream
SGD15.00
Singapore

Palmer's Cocoa Butter Formula Hair Remover for Body and Cocoa ... Scrub
SGD23.80
~~SGD31.80~~ -25%
Singapore

Beautiful Soup!!

Beautiful Soup

Hugely popular framework for web scraping. It provides very easy-to-use functionality through pythonic, readable, memorable syntax.

<Step 1a> Using Beautiful Soup

Import the 2 required libraries

Import requests

requests.**get**(url)
We need this function in the requests library to retrieve the HTML page into memory, then we can process it.

If you provide this get() function with a url, it fetches the HTML just like how your browser does.

<Step 1b> Using Beautiful Soup

Import the 2 required libraries

```
Import requests
from bs4 import BeautifulSoup
```

bs4.BeautifulSoup
A BeautifulSoup object -- code written by others -- provides very useful HTML processing functionality.

Processing HTML as one long string is a pain. BeautifulSoup makes the task tremendously easier. You will see why in a bit!

<Step 2> Using Beautiful Soup

Fetch the HTML page using urlopen

```
Import requests
from bs4 import BeautifulSoup


response = requests.get('        ')
```

requests.get(url)

*It takes in a URL (*string*) as an argument, and returns the HTTP response stored in a HTTP object (another special object with useful functionality), containing the HTML. To see this raw HTML,* print(response.text).

<Step 3> Using Beautiful Soup

Turn the response into a BeautifulSoup object

```
Import requests
from bs4 import BeautifulSoup
response  =  requests.get('        ')
soup = BeautifulSoup(response.text)
```

bs4.BeautifulSoup(x, ...)

*x can be a **HTML** string, or a **file-like object** such as `response` in this case.*

<Step 4> Using Beautiful Soup

Look for the **tags** you want from the html page, and ask your soup for it.

```
Import requests
from bs4 import BeautifulSoup
response = requests.get('        ')
soup = BeautifulSoup(response.text)
for div in soup.findAll('div'):
  print(div)
```

bs4.BeautifulSoup.findAll(name, ...)

name refers to the name of the HTML tag you're looking for. In this case, let's use the div tag as an example.

<Step 5> Using Beautiful Soup

Process! Extract the relevant data from the `div` variable in the for-loop.

```
Import requests
from bs4 import BeautifulSoup

response =  requests.get(' ')
soup = BeautifulSoup(response.text)
data = []
for div in soup.findAll('div'):
         data.append(div.text)
```

This gets me ALL the *div* tags' *texts*. But what if I want *divs*
with only a specific **class**? How do I filter the search?

<Step 5> Using Beautiful Soup

If I want to narrow down the search to just div tags with a class “a”,

```
Import requests
from bs4 import BeautifulSoup
response =  requests.get('  ')
soup = BeautifulSoup(response.text)
data = []
for div in soup.findAll('div', attrs={'class': 'a'}):
  data.append(div.text)
```

bs4.BeautifulSoup.findAll(name, **attrs**)

*The **attrs** parameter takes in a dictionary of {attribute: value}. In this case, we specify that we want div tags with the class attribute as ‘a’.*

<Step 6> Using Beautiful Soup

Etc etc etc...

```
import requests
from bs4 import BeautifulSoup
response =  requests.get('  ')
soup = BeautifulSoup(response.text)
data = []
for div in soup.findAll('div', attrs={'class': 'a'}):
  data.append(div.text)


#Process  `data`...
```

Activity Time!

Activity Time!

- You will now be split into breakout rooms of 4-5 people! This discussion session will last for about 20 minutes.
- These questions are based on https://www.testpapersfree.com/
- The files should be downloaded through web scraping (code) only.
- The code is only allowed to start at the link above, and not sublinks.
- Each group will send a representative to answer these questions:

1. Download *P2-Chinese-2014-SA2-Tao-Nan.pdf* (Difficulty: Easy, Hint: It is one of the pdfs available on the main website)
2. Download *P6_Maths_SA2_2018_Raffles_Exam_Paper.pdf* (Difficulty: Challenging)

THANKS!

And much thanks to these ppl :D

- Hackwagon
- https://www.w3schools.com/cssref/css_selectors.asp