XPath Navigation

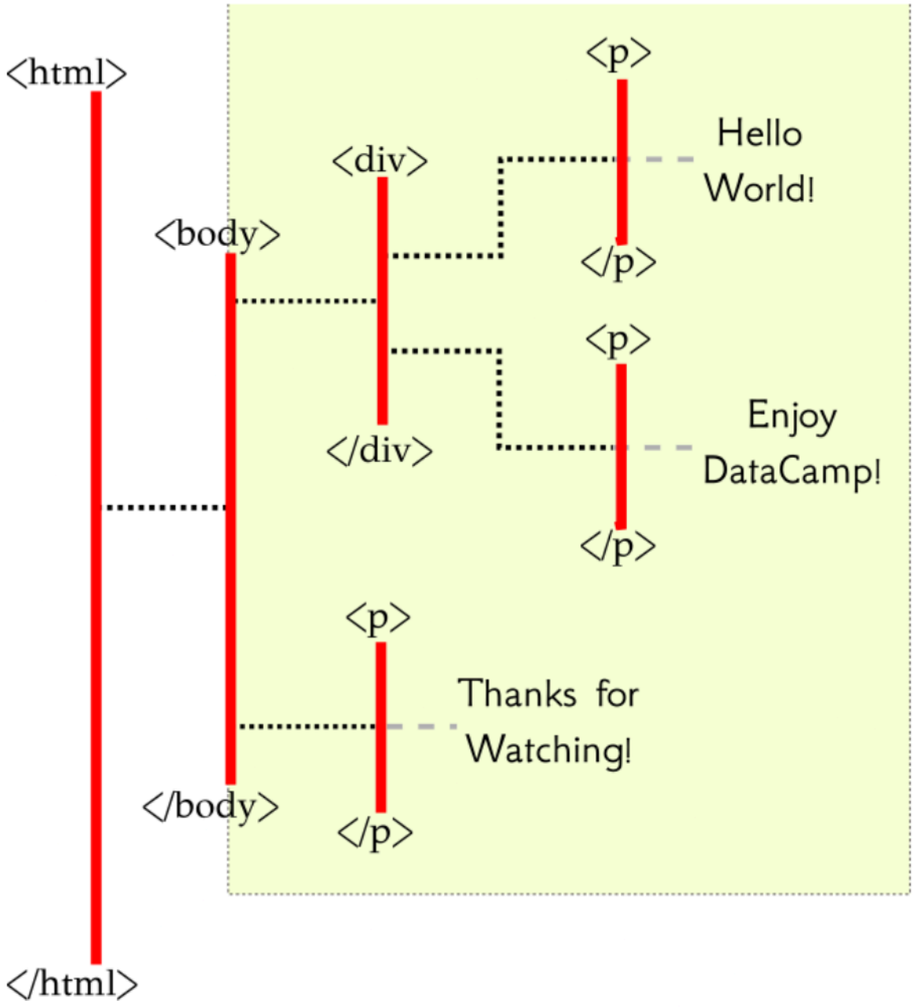
WEB SCRAPING IN PYTHON

Thomas Laetsch
Data Scientist, NYU

Slashes and Brackets

- Single forward slash / looks forward **one** generation
- Double forward slash // looks forward **all** future generations
- Square brackets [] help narrow in on specific elements

To Bracket or not to Bracket



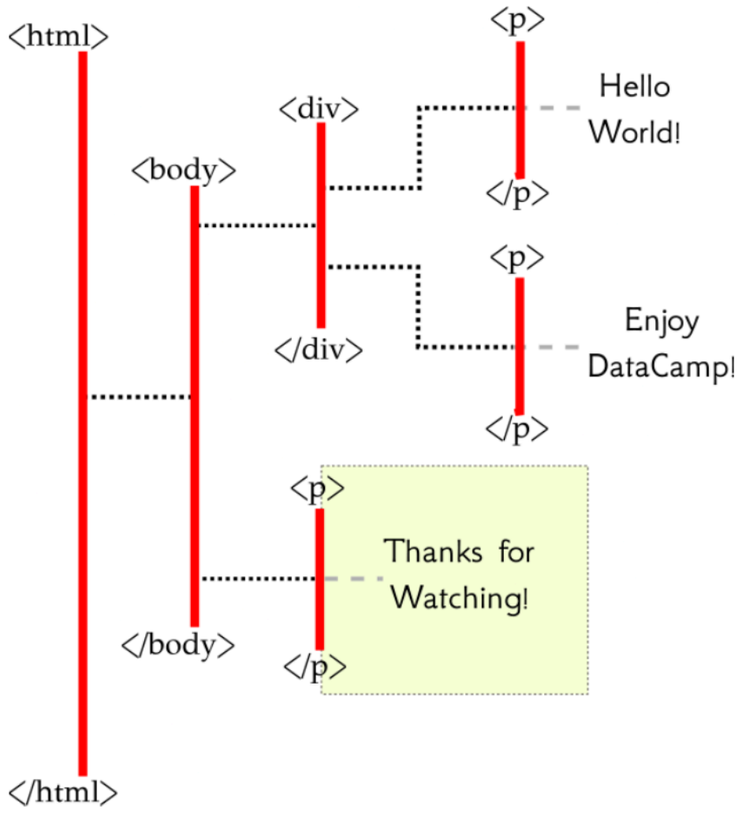

```
xpath = '/html/body'
```

```
xpath = '/html[1]/body[1]'
```

- Give the same selection

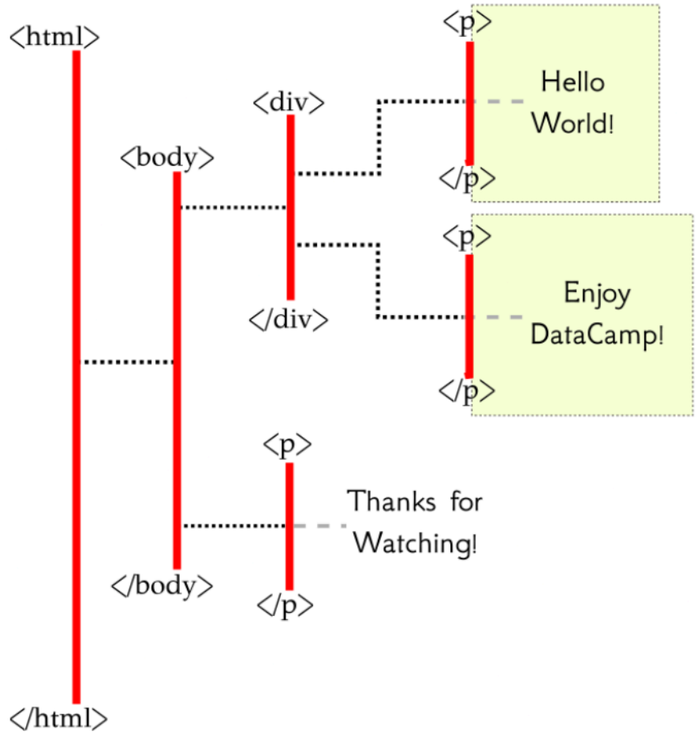
A Body of P

```
xpath = '/html/body/p'
```

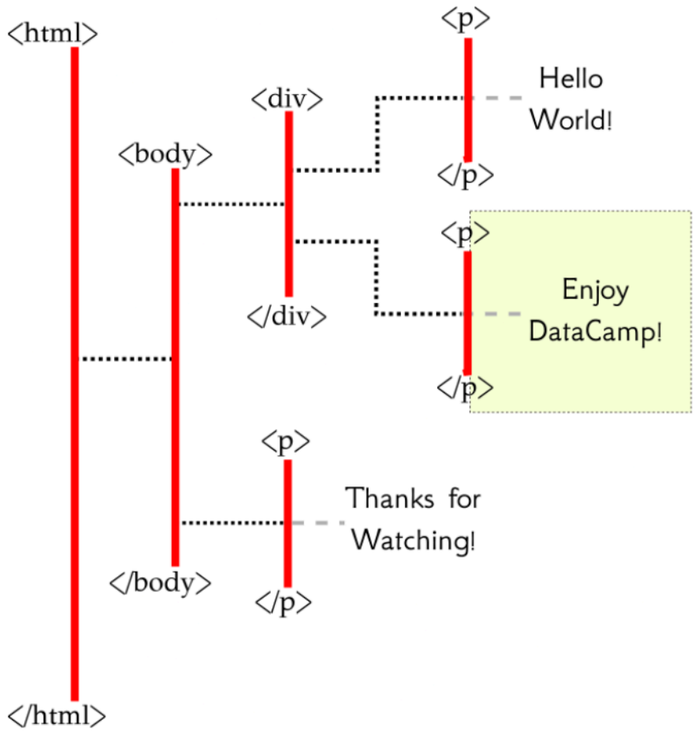


The Birds and the Ps

```
xpath = '/html/body/div/p'
```

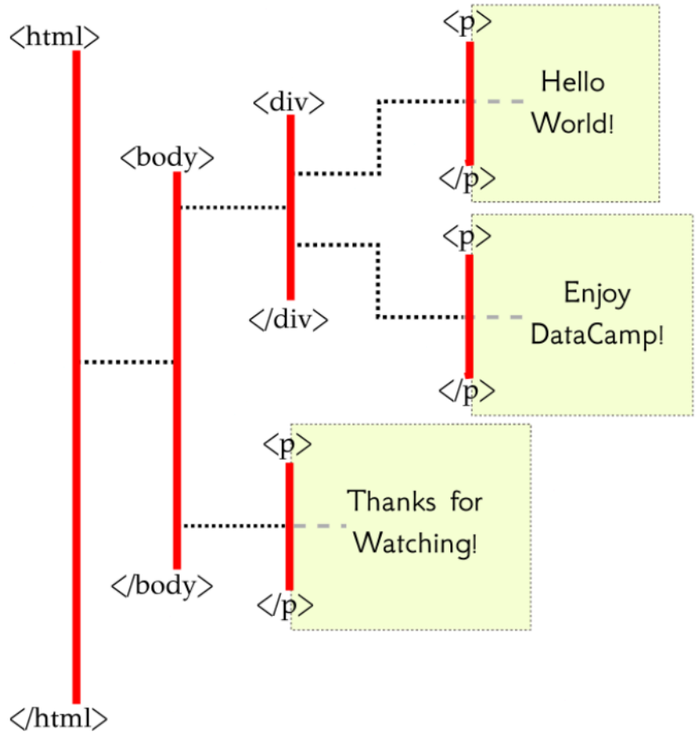



```
xpath = '/html/body/div/p[2]'
```

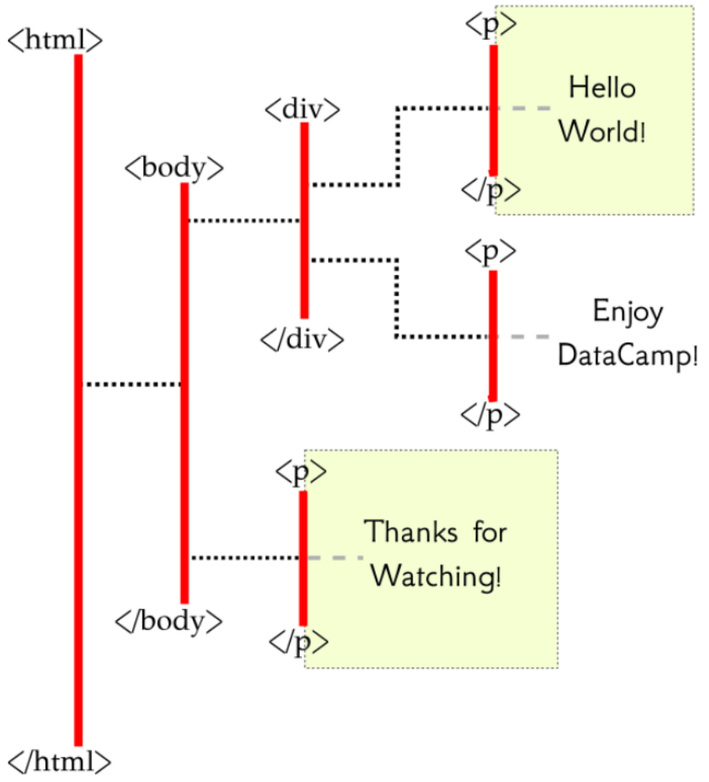


Double Slashing the Brackets

```
xpath = '//p'
```



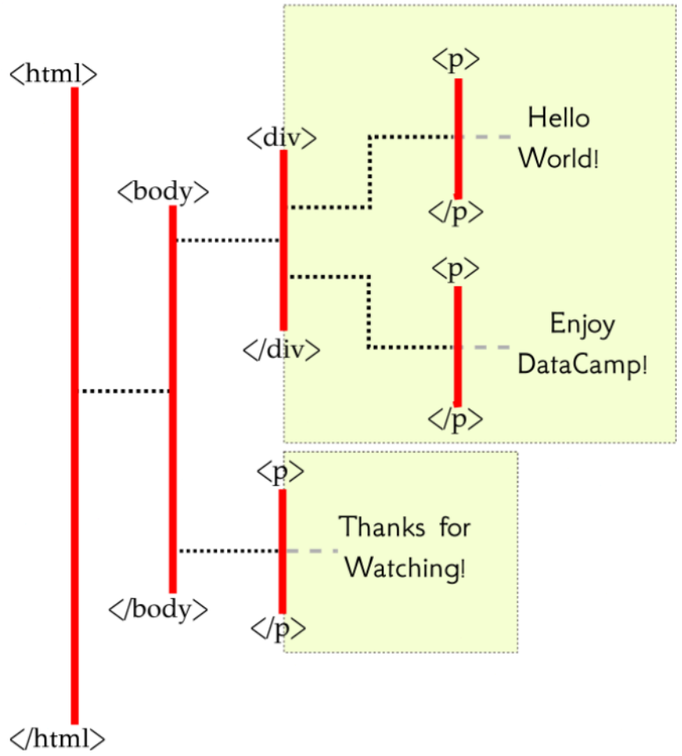

```
xpath = '//p[1]'
```



The Wildcard

```
xpath = '/html/body/*'
```

- The asterisks * is the "wildcard"



Xposé

WEB SCRAPING IN PYTHON

Off the Beaten XPath

WEB SCRAPING IN PYTHON

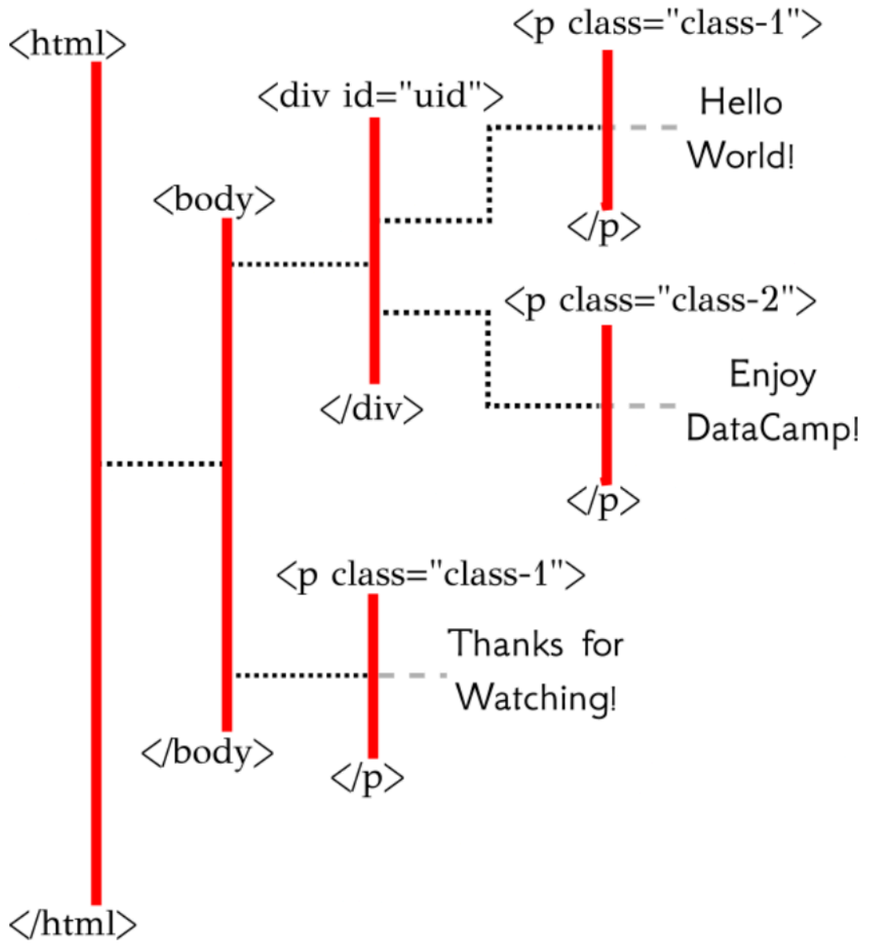
Thomas Laetsch
Data Scientist, NYU

datacamp

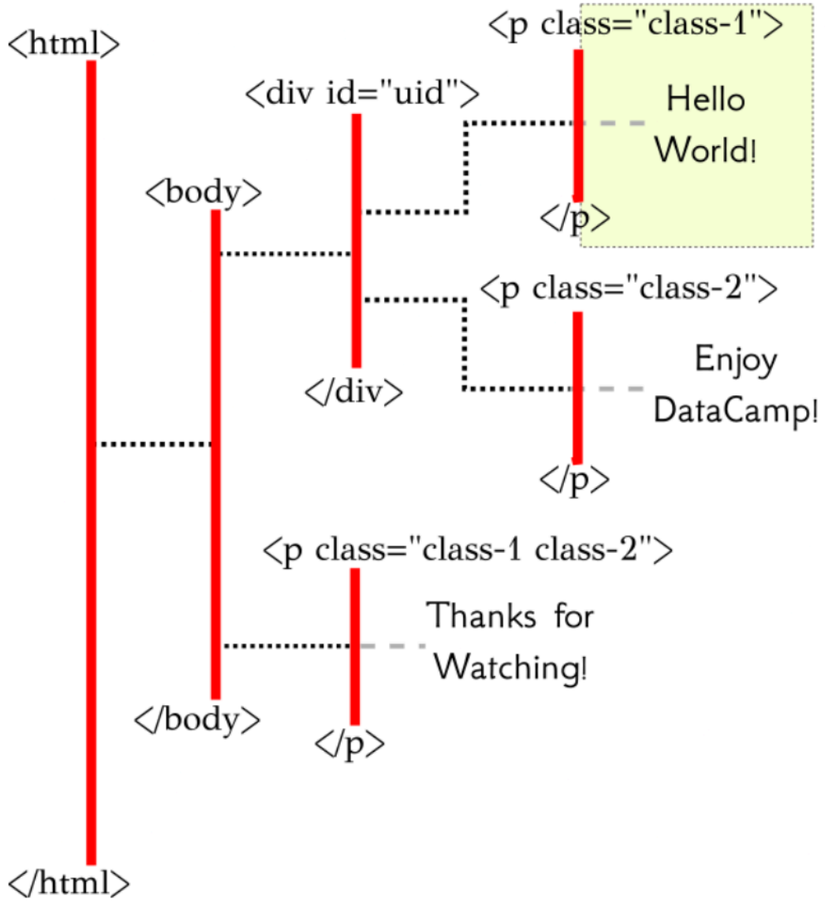
(At)tribute

- `@` represents "attribute"
  - `@class`
  - `@id`
  - `@href`

Brackets and Attributes

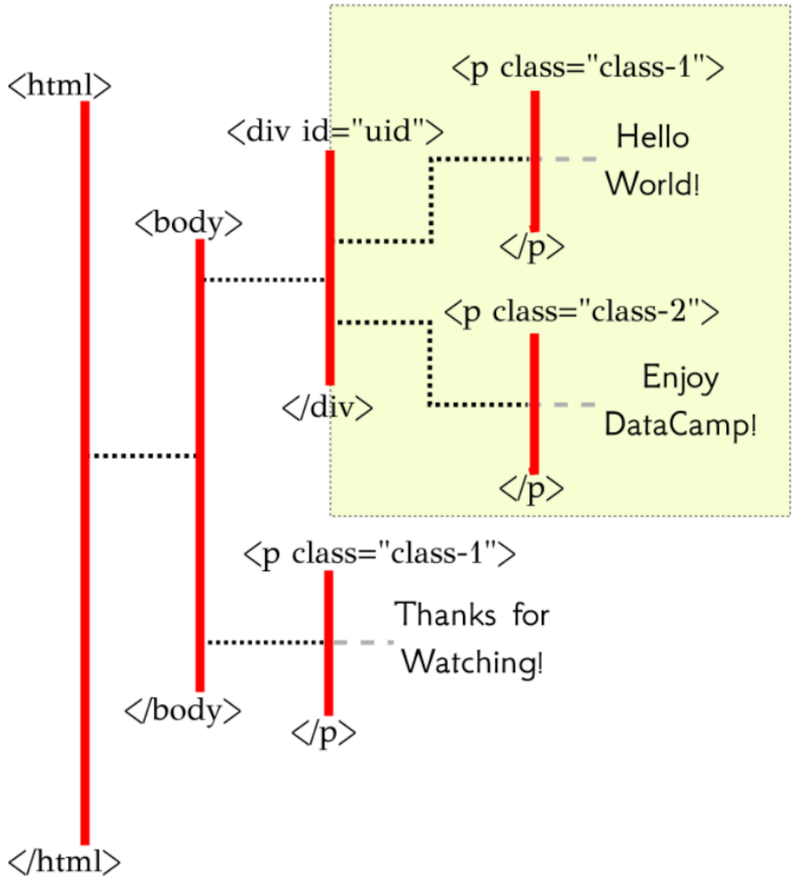


Brackets and Attributes



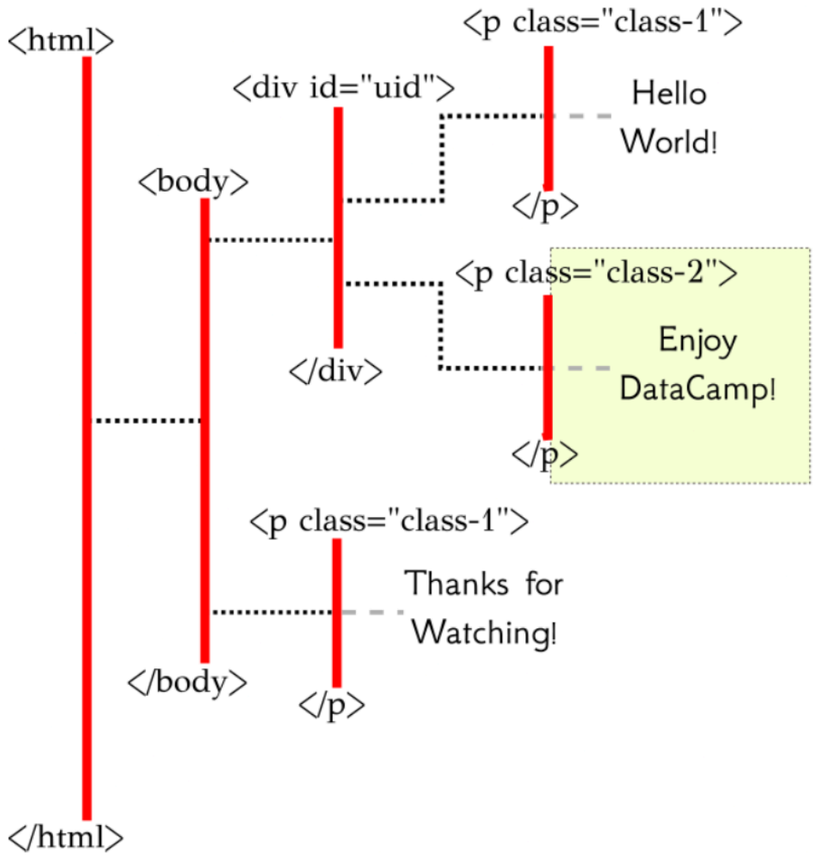

```
xpath = '//p[@class="class-1"]'
```

Brackets and Attributes




```
xpath = '//*[@id="uid"]'
```

Brackets and Attributes




```
xpath = '//div[@id="uid"]/p[2]'
```

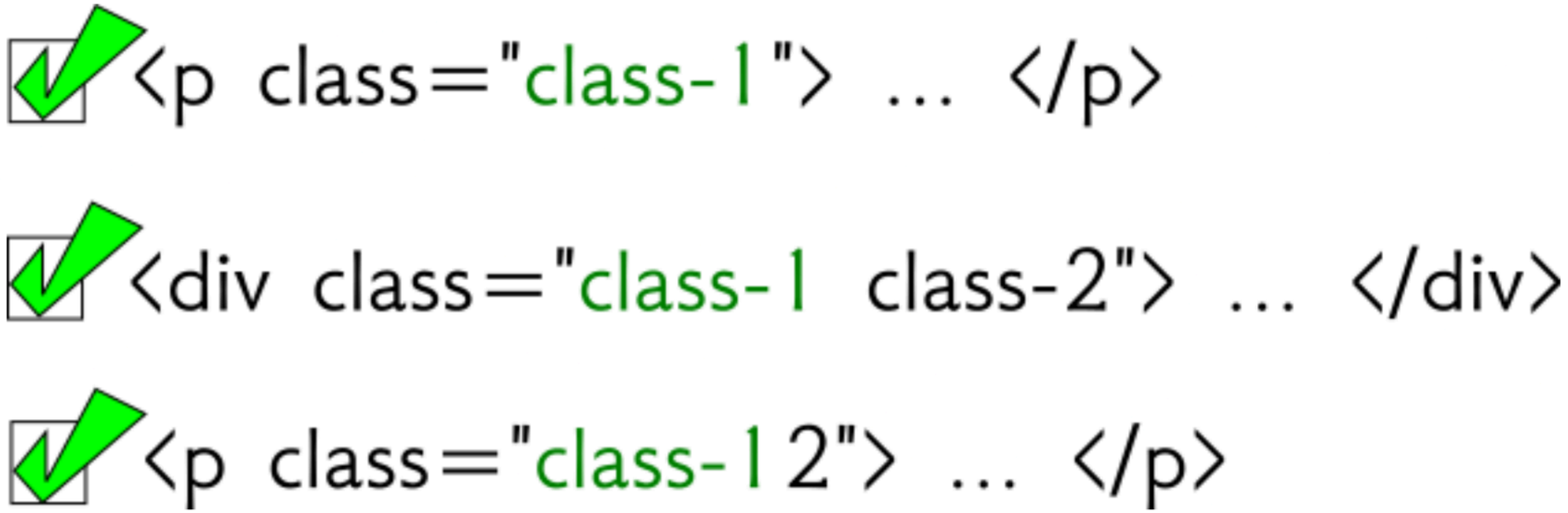
Content with Contains

Xpath Contains Notation:

contains(@attri-name, "string-expr")

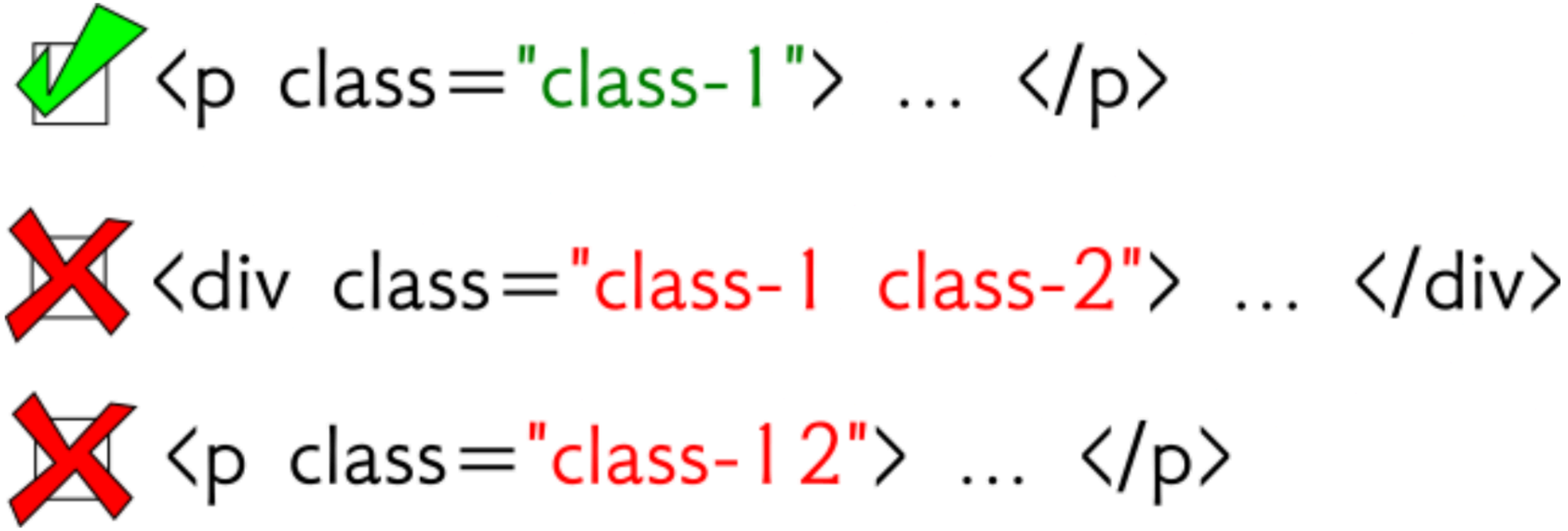
Contain This

```
xpath = '//*[contains(@class,"class-1")]'
```

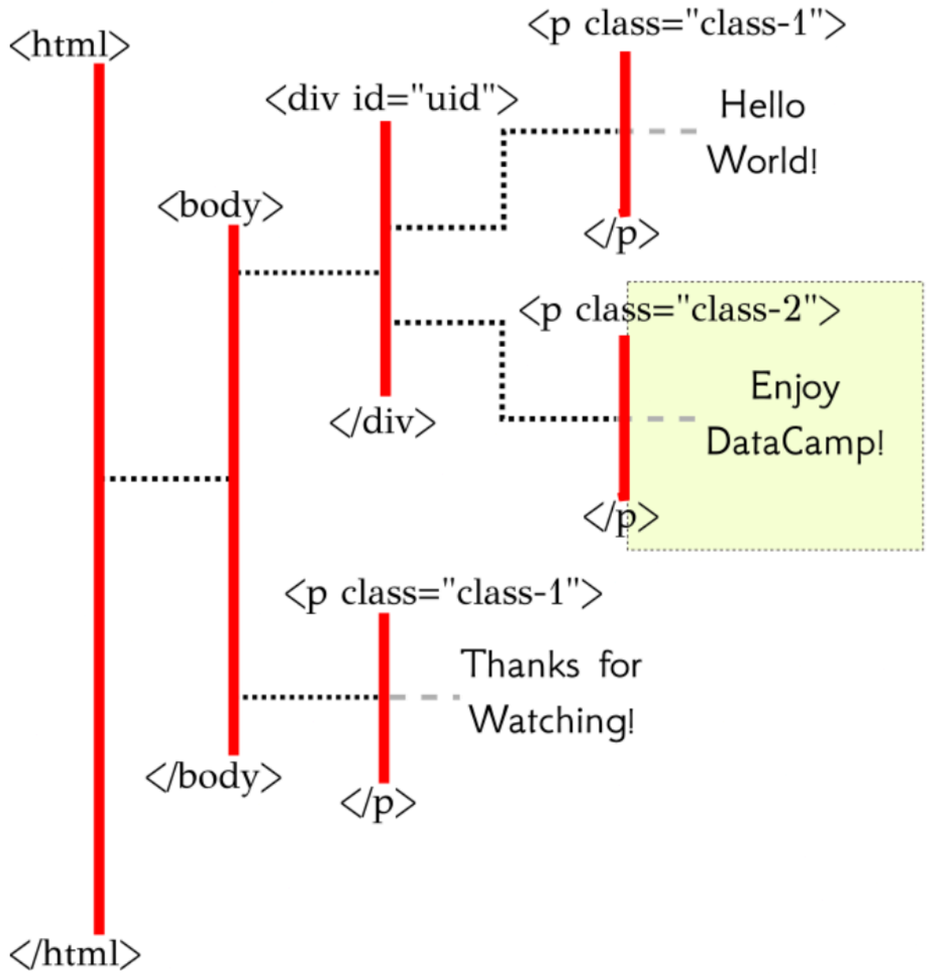


Contain This

```
xpath = '//*[@class="class-1"]'
```

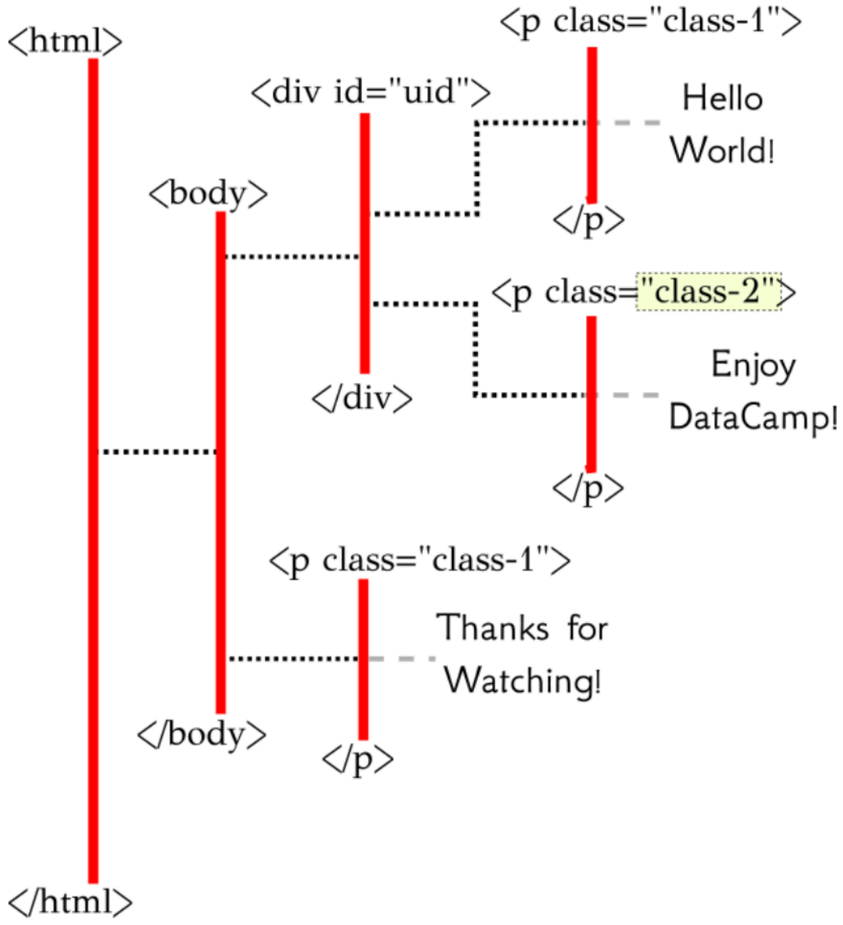


Get Classy




```
xpath = '/html/body/div/p[2]'
```

Get Classy




```
xpath = '/html/body/div/p[2]/@class'
```

End of the Path

WEB SCRAPING IN PYTHON

Introduction to the scrapy Selector

WEB SCRAPING IN PYTHON

Thomas Laetsch
Data Scientist, NYU

Setting up a Selector

```
from scrapy import Selector
```

```
html = '''
<html>
  <body>
    <div class="hello datacamp">
      <p>Hello World!</p>
    </div>
    <p>Enjoy DataCamp!</p>
  </body>
</html>
'''
```

```
sel = Selector( text = html )
```

- Created a scrapy Selector object using a string with the html code
- The selector `sel` has selected the **entire** html document

Selecting Selectors

- We can use the `xpath` call within a `Selector` to create new `Selector` s of specific pieces of the html code
- The return is a `SelectorList` of `Selector` objects

```
sel.xpath("//p")
# outputs the SelectorList:
[<Selector xpath='//p' data='<p>Hello World!</p>'>,
 <Selector xpath='//p' data='<p>Enjoy DataCamp!</p>'>]
```

Extracting Data from a SelectorList

- Use the `extract()` method

```
>>> sel.xpath("//p")
out: [<Selector xpath='//p' data='<p>Hello World!</p>'>,
      <Selector xpath='//p' data='<p>Enjoy DataCamp!</p>'>]
```

```
>>> sel.xpath("//p").extract()
out: [ '<p>Hello World!</p>',
       '<p>Enjoy DataCamp!</p>' ]
```

- We can use `extract_first()` to get the first element of the list

```
>>> sel.xpath("//p").extract_first()
out: '<p>Hello World!</p>'
```

Extracting Data from a Selector

```
ps = sel.xpath('//p')
second_p = ps[1]
```

```
second_p.extract()
out: '<p>Enjoy DataCamp!</p>'
```

Select This Course!

WEB SCRAPING IN PYTHON

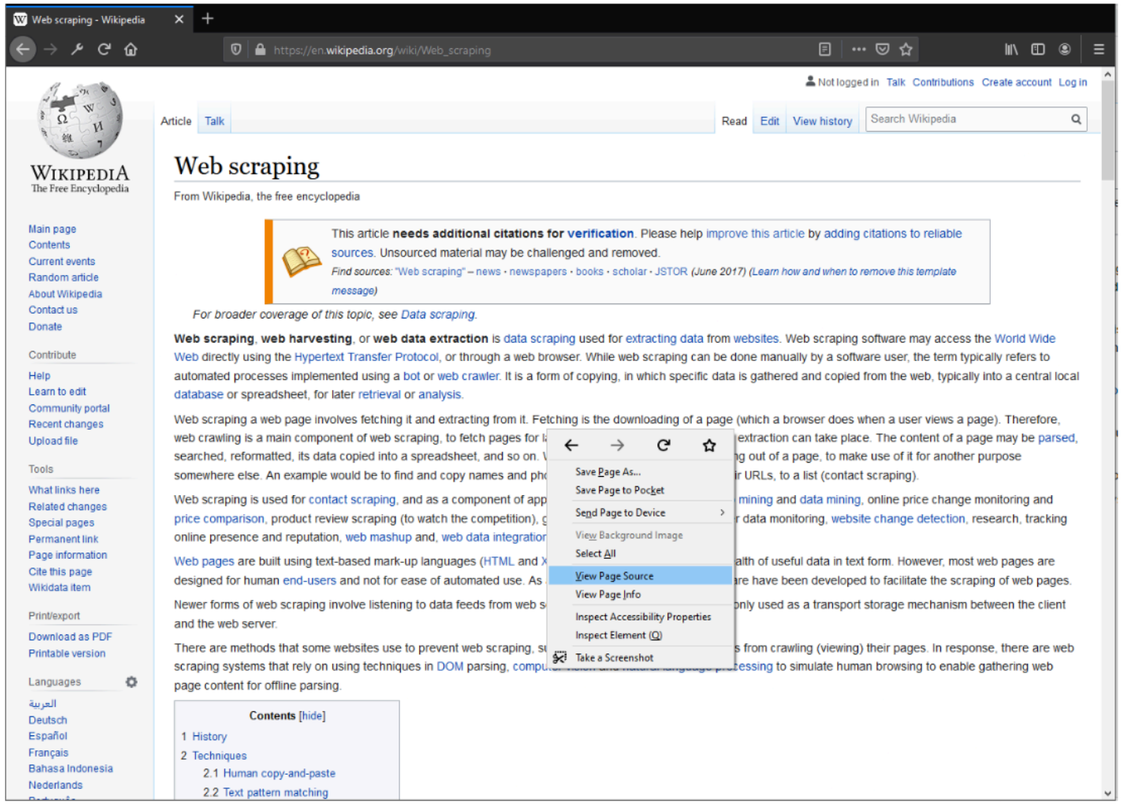
"Inspecting the HTML"

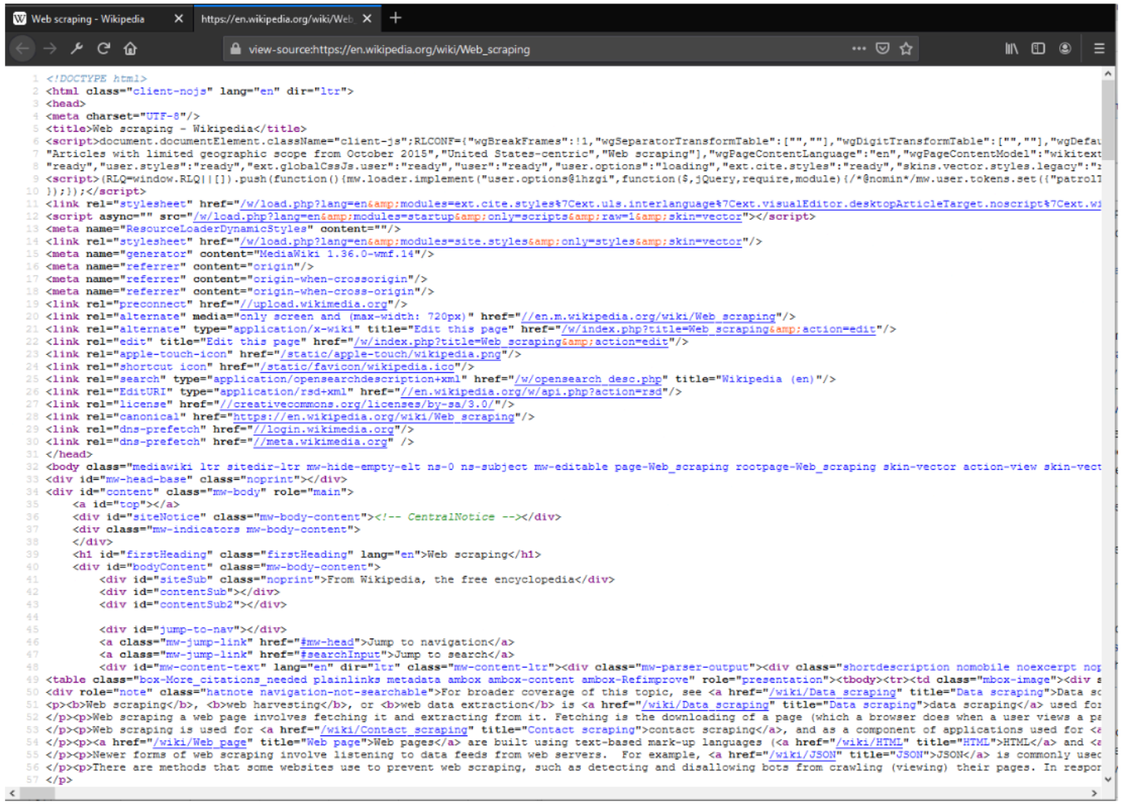
WEB SCRAPING IN PYTHON

Thomas Laetsch, PhD
Data Scientist, NYU

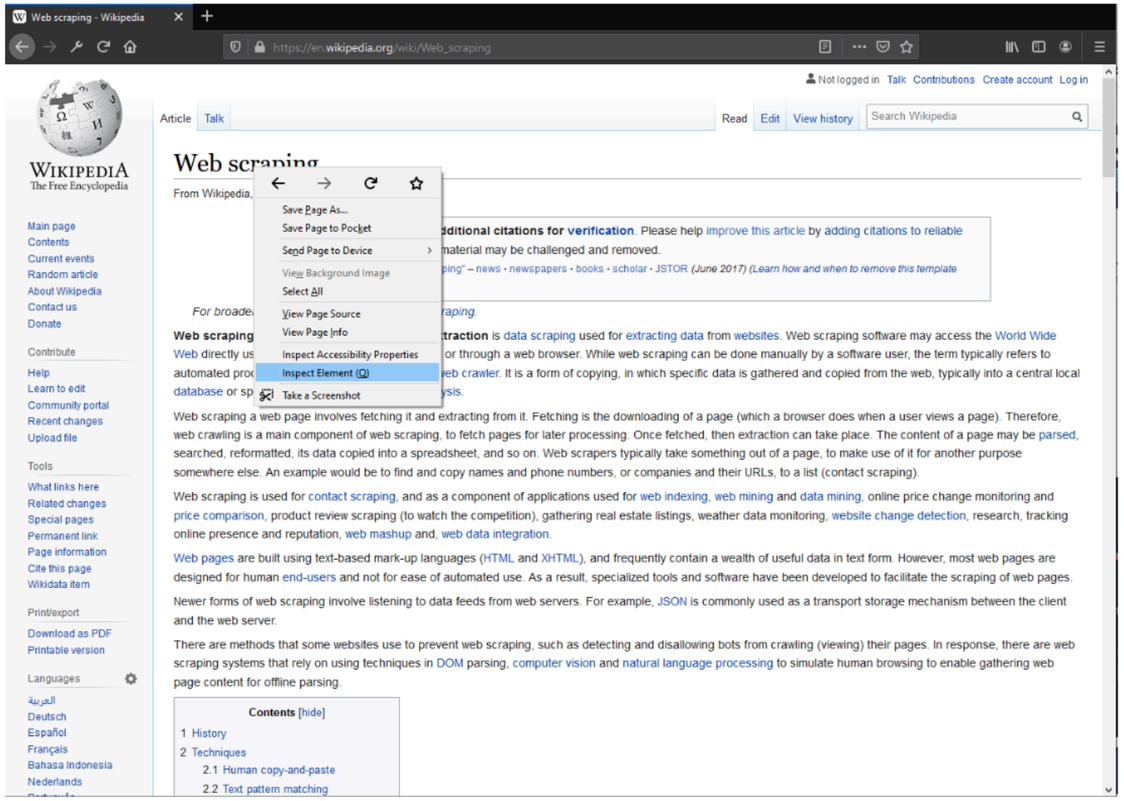
datacamp

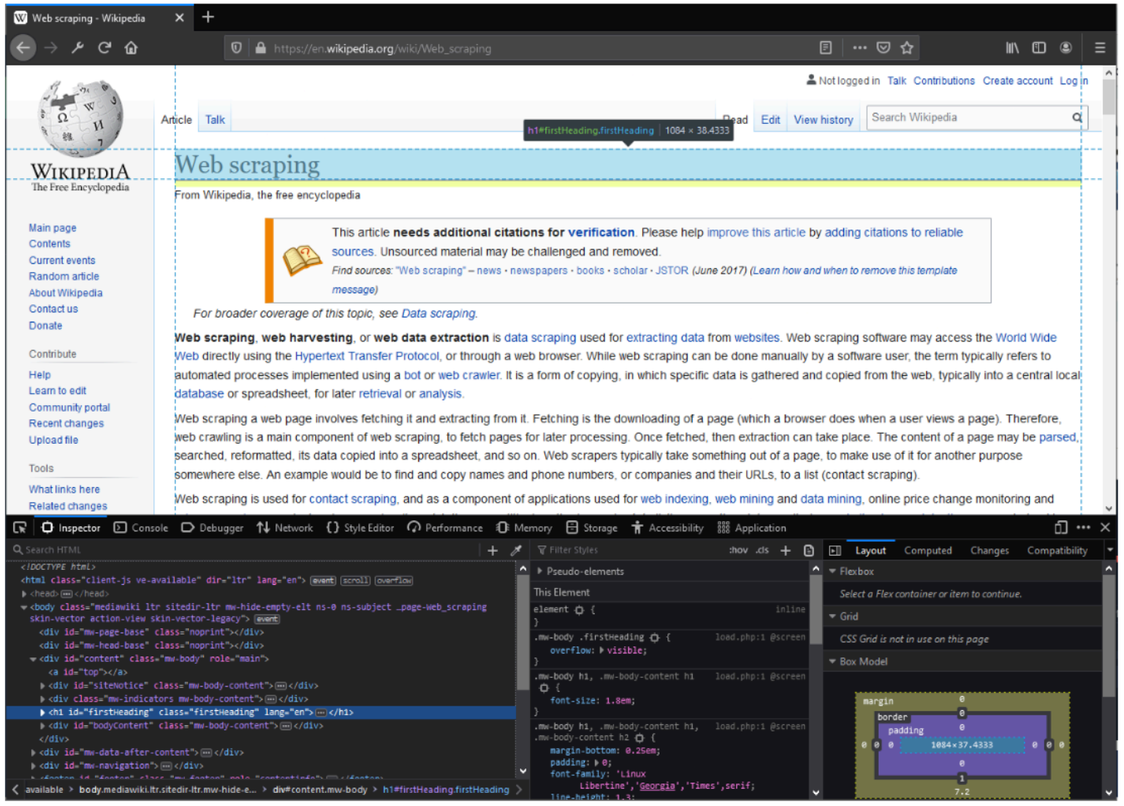
"Source" = HTML Code





Inspecting Elements





HTML text to Selector

```
from scrapy import Selector
```

```
import requests
url = 'https://en.wikipedia.org/wiki/Web_scraping'
html = requests.get( url ).content
```

```
sel = Selector( text = html )
```

You Know Our Secrets

WEB SCRAPING IN PYTHON