



WEB SCRAPING IN PYTHON

XPath Navigation

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_body_sel.png

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

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

- Give the same selection



A Body of P

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



xpath_p_sel.png



The Birds and the Ps

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



xpath_div_p_sel.png

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



xpath_div_p2_sel.png



Double Slashing the Brackets

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



xpath_p_sel.png

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



xpath_body_ssp1_sel.png



The Wildcard

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



xpath_body_ast_sel.png

- The asterisks * is the "wildcard"



WEB SCRAPING IN PYTHON

Xposé



WEB SCRAPING IN PYTHON

Off the Beaten XPath

Thomas Laetsch
Data Scientist, NYU



(At)tribute

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



Brackets and Attributes



xpathattr.png



Brackets and Attributes



xpathattr_div_p1.png

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



Brackets and Attributes



xpathattr_div.png

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



Brackets and Attributes



xpathattr_div_astc2.png

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



Content with Contains

Xpath Contains Notation:

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



Contain This

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



ClassSelection-Xpath-contains.png



Contain This

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



ClassSelection-Xpath-eq.png



Get Classy



xpathattr_div_astc2.png

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



Get Classy



xpathattr_div_p2-class.png

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



WEB SCRAPING IN PYTHON

End of the Path



WEB SCRAPING IN PYTHON

Introduction to the scrapy Selector

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 `Selectors` 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>'
```



WEB SCRAPING IN PYTHON

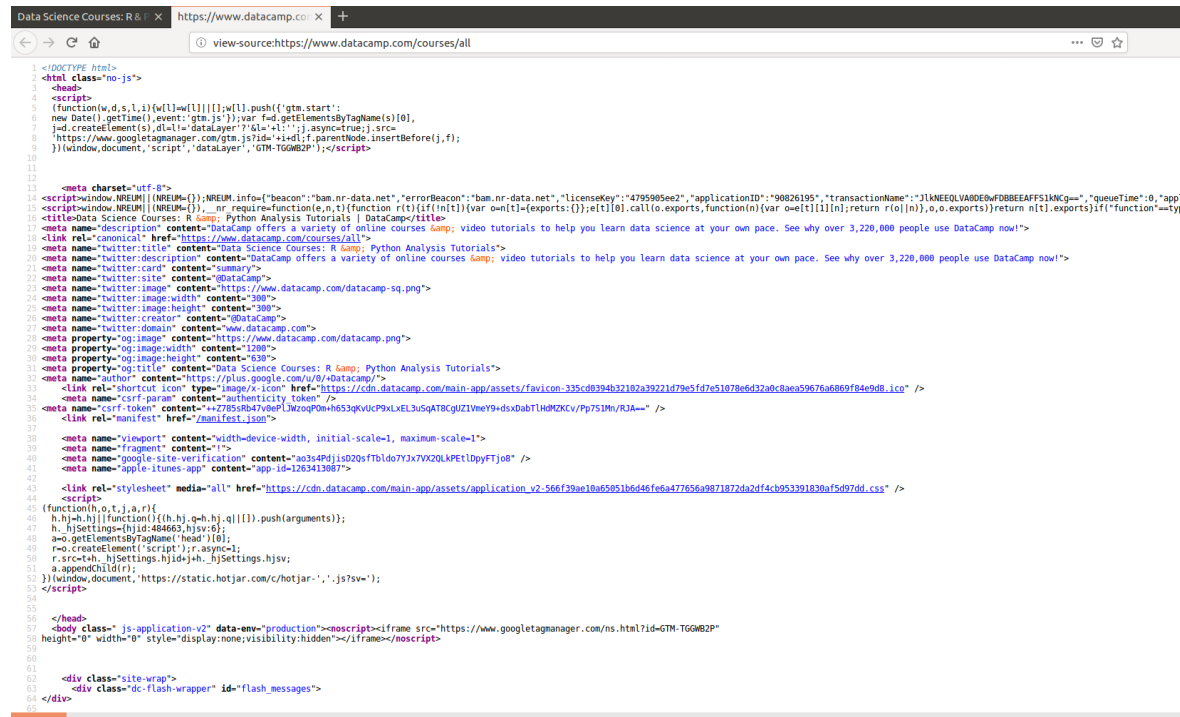
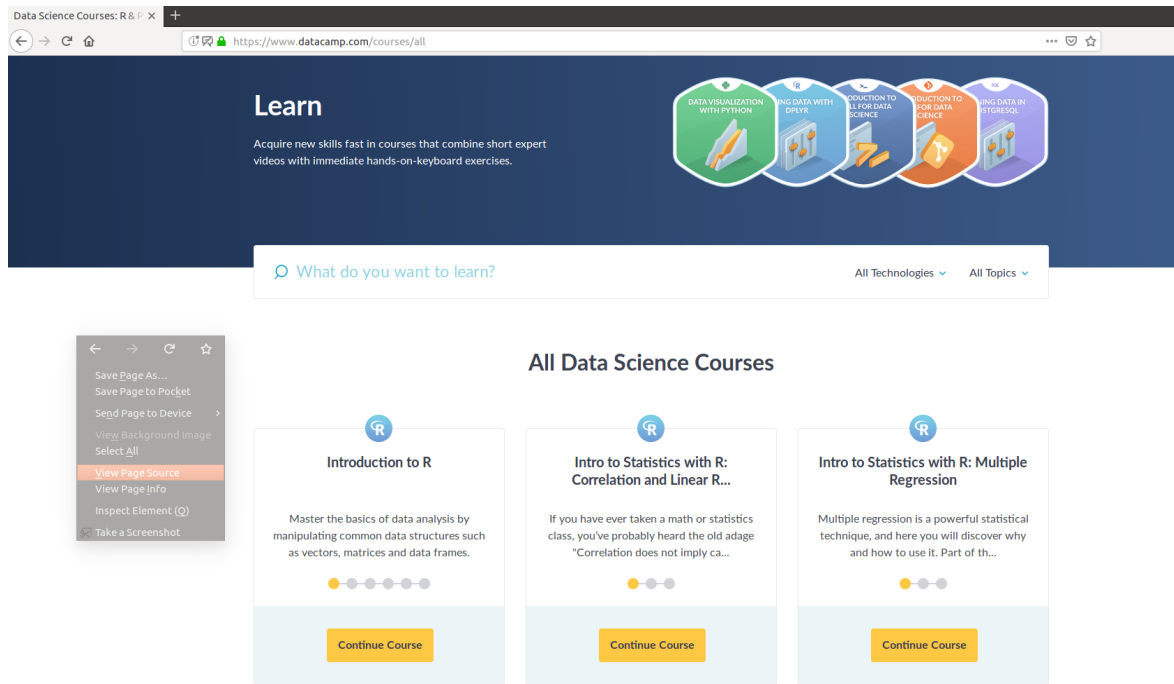
Select This Course!



WEB SCRAPING IN PYTHON

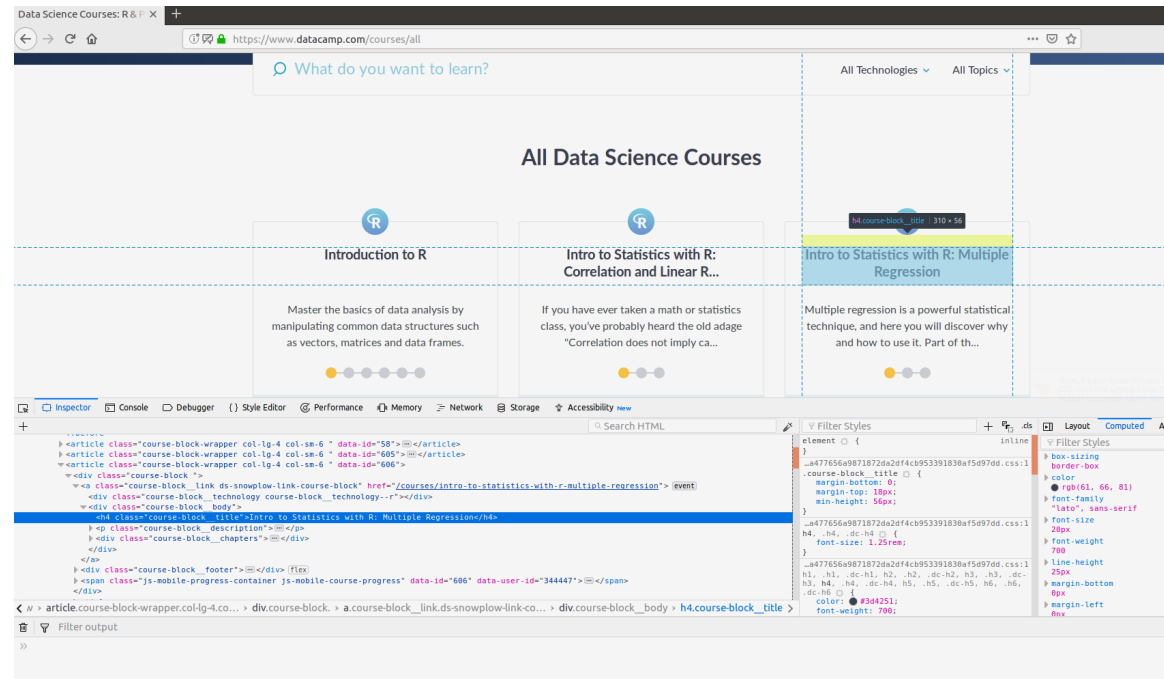
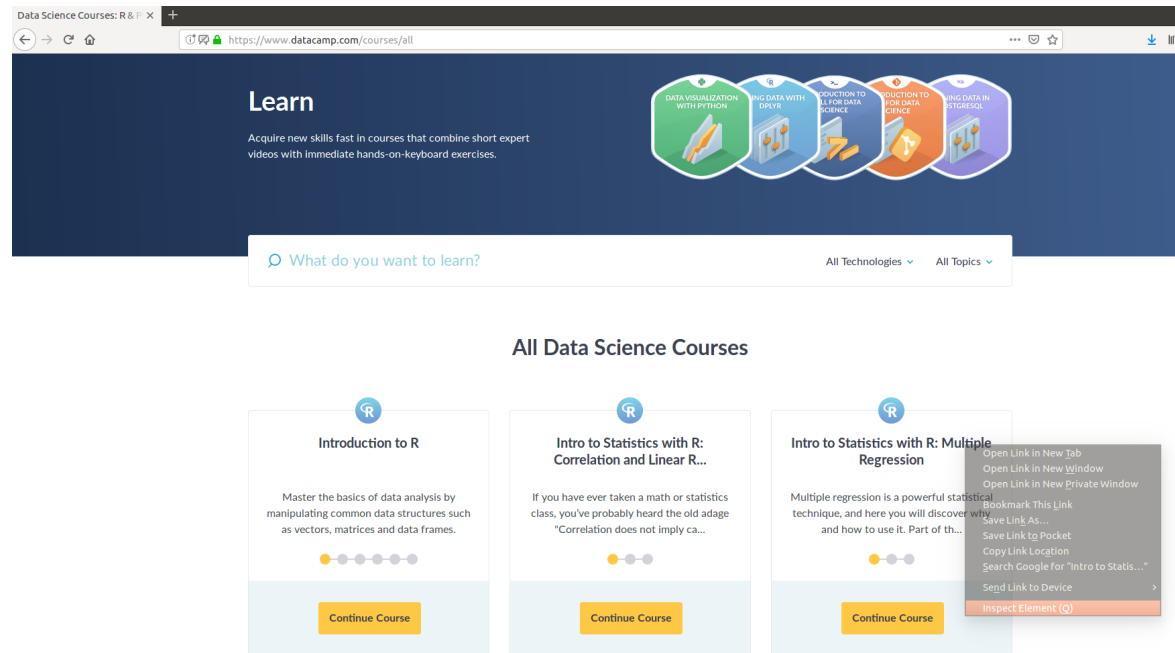
"Inspecting the HTML"

Thomas Laetsch, PhD
Data Scientist, NYU





Inspecting Elements





HTML text to Selector

```
from scrapy import Selector
```

```
import requests
```

```
url = 'https://www.datacamp.com/courses/all'
```

```
html = requests.get( url ).content
```

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



WEB SCRAPING IN PYTHON

You Know Our Secrets