WEB SCRAPING IN PYTHON

# Web Scraping With Python

Thomas Laetsch
Data Scientist, NYU

# Business Savvy

**What are businesses looking for?**

- Comparing prices
- Satisfaction of customers
- Generating potential leads
- ...and much more!

# It's Personal

**What could you do?**

- Search for your favorite memes on your favorite sites.
- Automatically look through classified ads for your favorite gadgets.
- Scrape social site content looking for hot topics.
- Scrape cooking blogs looking for particular recipes, or recipe reviews.
- ...and much more!

# About My Work

# Pipe Dream

# Pipe Dream: Setup


**Setup**

- Understand what we want to do.
- Find sources to help us do it.

# Pipe Dream: Acquisition


**Acquisition**

- Read in the raw data from online.
- Format these data to be usable.

# Pipe Dream: Processing


**Processing**

- Many options!

# How do you do?

**Our Focus**

- Acquisition!
- (Using `scrapy` via `python`)

WEB SCRAPING IN PYTHON

# Are you in?

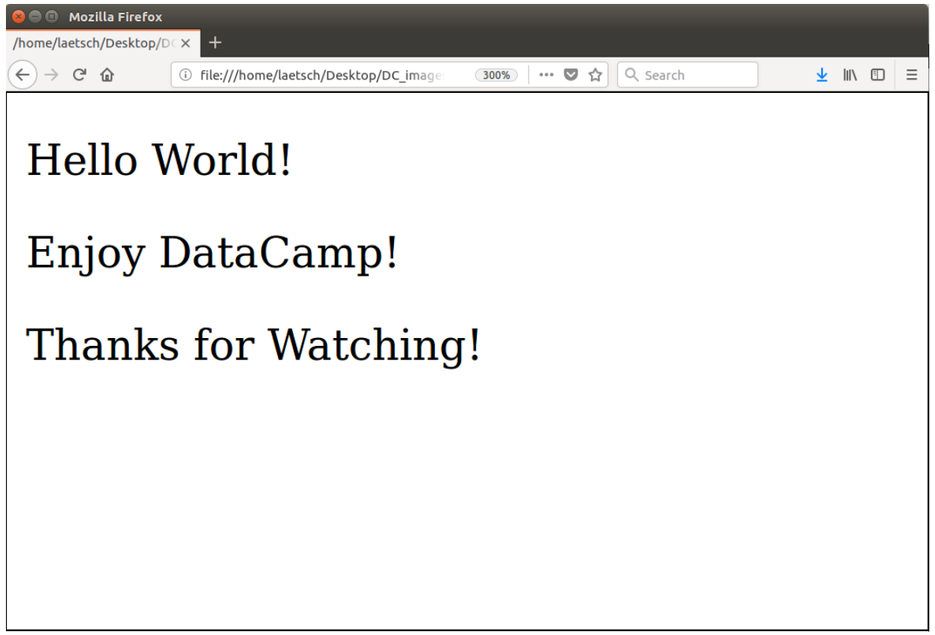
WEB SCRAPING IN PYTHON

# HyperText Markup Language

Thomas Laetsch
Data Scientist, NYU

# The main example

```
<html>
    <body>
        <div>
            <p>Hello World!</p>
            <p>Enjoy DataCamp!</p>
        </div>
        <p>Thanks for Watching!</p>
    </body>
</html>
```



# HTML tags

```
<html>
    <body>
        <div>
            <p>Hello World!</p>
            <p>Enjoy DataCamp!</p>
        </div>
        <p>Thanks for Watching!</p>
    </body>
</html>
```

- `<html> ... </html>`
- `<body> ... </body>`
- `<div> ... </div>`
- `<p> ... </p>`

# The HTML tree

```
<html>
    <body>
        <div>
            <p>Hello World!</p>
            <p>Enjoy DataCamp!</p>
        </div>
        <p>Thanks for Watching!</p>
    </body>
</html>
```

# The HTML tree: Example 1

```
<html>
    <body>
        <div>
            <p>Hello World!</p>
            <p>Enjoy DataCamp!</p>
        </div>
        <p>Thanks for Watching!</p>
    </body>
</html>
```

# The HTML tree: Example 2

```
<html>
    <body>
        <div>
            <p>Hello World!</p>
            <p>Enjoy DataCamp!</p>
        </div>
        <p>Thanks for Watching!</p>
    </body>
</html>
```

WEB SCRAPING IN PYTHON

# Introduction to HTML Outro

WEB SCRAPING IN PYTHON

# HTML Tags and Attributes

Thomas Laetsch
Data Scientist, NYU

# Do we have to?

- Information within HTML tags can be valuable
- Extract link URLs
- Easier way to select elements

# Tag, you're it!


- We've seen **tag names** such as **html**, **div**, and **p**.
- The **attribute name** is followed by **=** followed by information assigned to that attribute, usually quoted text.

# Let's "div"vy up the tag


- **id** attribute should be unique
- **class** attribute doesn't need to be unique

# "a" be linkin'


- **a** tags are for **hyperlinks**
- **href** attribute tells what link to go to

# Tag Traction

WEB SCRAPING IN PYTHON

# Et Tu, Attributes?

WEB SCRAPING IN PYTHON

# Crash Course X

Thomas Laetsch
Data Scientist, NYU

# Another Slasher Video?

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

**Simple XPath**:

- Single forward-slash `/` used to move forward one generation.
- tag-names between slashes give direction to which element(s).
- Brackets [] after a tag name tell us which of the selected siblings to choose.

# Another Slasher Video?

highlight_div.png

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

# Slasher Double Feature?

- Direct to all `table` elements within the entire HTML code:

```
xpath = '//table'
```

- Direct to all `table` elements which are descendants of the 2nd `div` child of the `body` element:

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

WEB SCRAPING IN PYTHON

# Ex(path)celent