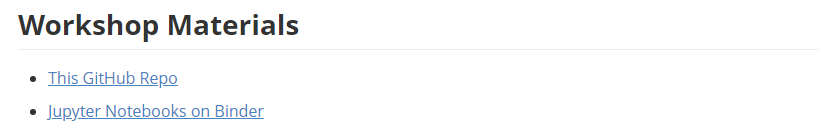
**Introduction to Web Scraping with BeautifulSoup in Python**

This workshop introduces how to extract information from a static HTML website with BeautifulSoup in Python. Please note that it is designed for participants with no programming experience.

**Getting Ready for the Workshop**

* Open the GitHub repo for the workshop <https://github.com/nuslds/intro-beautifulsoup>
* Under the “Workshop Materials” section, open the **Jupyter Notebooks on Binder**



**Introduction**

### **What is Web Scraping?**

Web scraping is a process of retrieving information from web services in an automated way.

### **Why Web Scraping?**

Web scraping saves you from the headaches of repeatedly copying or downloading data from different websites.

* It simplifies and automates the process of extracting data online.
* It transforms the scrapped data into structured formats.
* Through these, it creates datasets for data-driven projects.

[Source: How Web Scraping is Transforming the World with its Applications](https://towardsdatascience.com/https-medium-com-hiren787-patel-web-scraping-applications-a6f370d316f4)

### **What to Consider before Web Scraping**

* Terms and Conditions of the hosting sites
* The bandwidth of the hosting sites

[Source: Web Scraping, Columbia University Mailman School of Public Health](https://www.mailman.columbia.edu/research/population-health-methods/web-scraping)

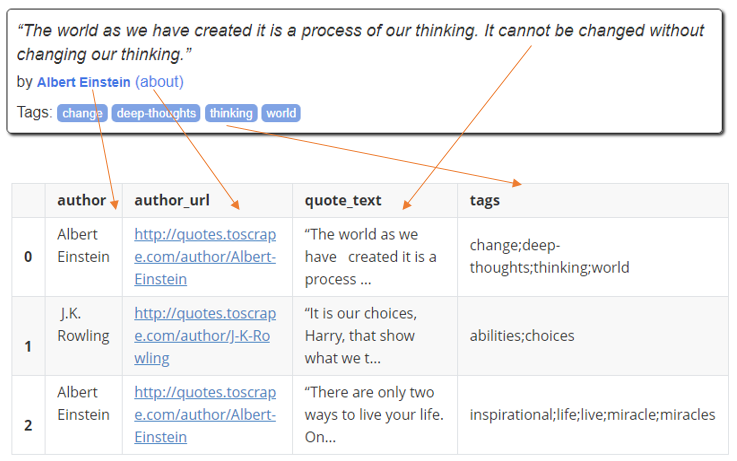
**Overview**

* Part I. Jupyter Notebook & Python Warmup
* Part II. Inspecting page source & HTML elements
* Part III. Extracting Data from HTML with BeautifulSoup
* Part IV. Scraping web pages with Requests and BeautifulSoup

**Task Today**

To demonstrate, we will extract quotes from [Quotes to Scrape](http://quotes.toscrape.com/). This is a project created by Scrapinghub ([Github repo](https://github.com/scrapinghub/spidyquotes)). We will create a reusable function that scrapes quotes from the website by page numbers. The outputs will be converted into tabular format and export into CSV.

# scrape quotes from page 1 to page 10  
outputs = scrape\_quotes(1, 10)



**Part I. Jupyter Notebook & Python Warmup**

### Just to get you familiarised with Jupyter Notebook and Python!

**Introducing Jupyter Notebook**

In this workshop, we will be using Jupyter Notebook to write your Python Codes. It is a powerful tool that allows you to immediately view the results and easily take notes while coding.

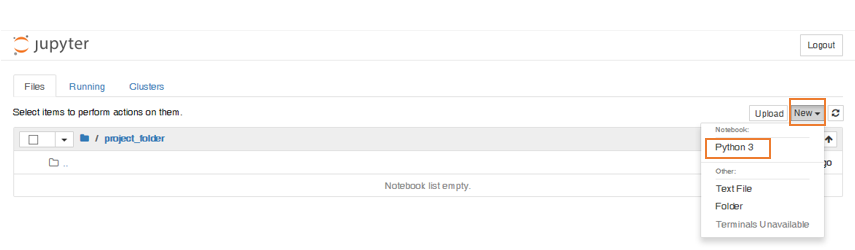
### **1. Accessing Jupyter Notebook on Your PC**

### Please refer to the corresponding markdown file on the GitHub Repo for more information.

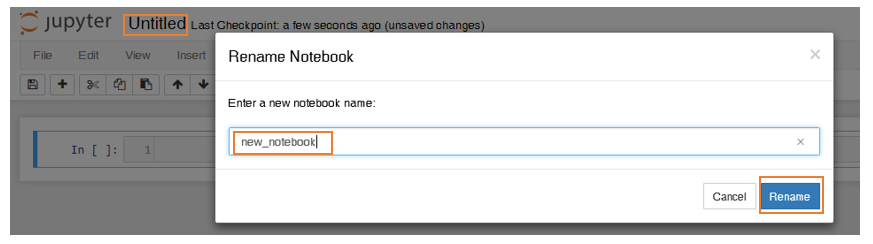
### **2. Trying out Jupyter Notebook**

Please go to the [Jupyter Notebooks on Binder](https://mybinder.org/v2/gh/nuslds/intro-beautifulsoup/master/) - Open the folder warm\_up.

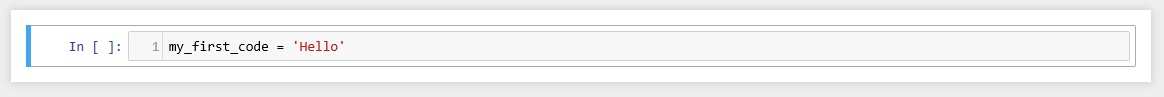
* Create a new notebook



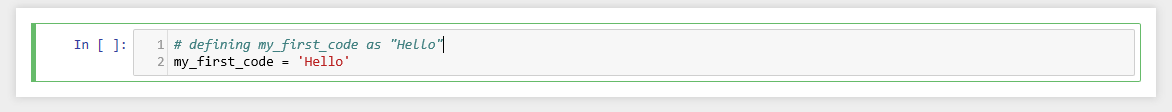
* Rename the notebook



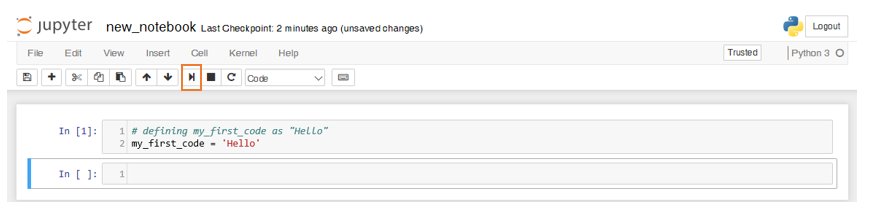
* Write your first code



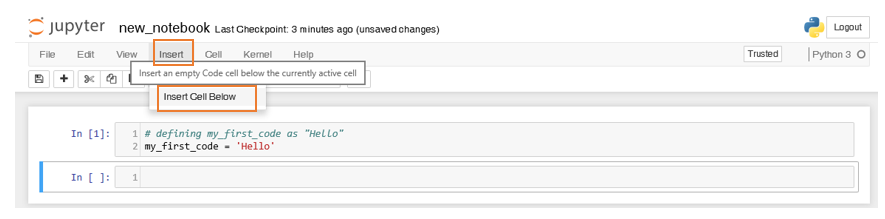
* Write comments in Python



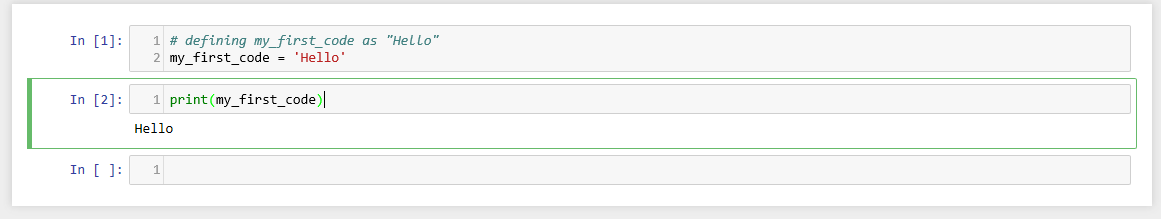
* Run the cell



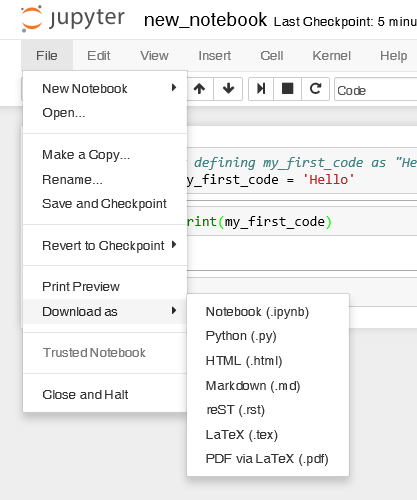
* Insert a new cell below



* Display the value of a variable



* Download the notebook

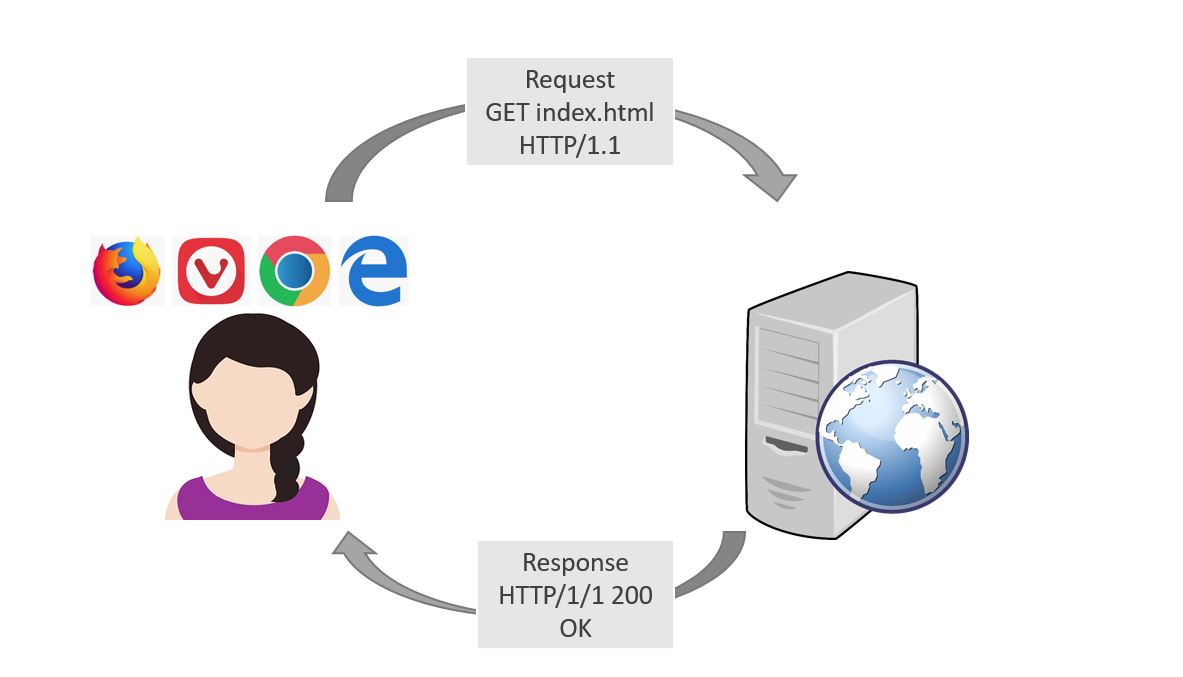


## **Python Basics**

Please go to the [Jupyter Notebooks on Binder](https://mybinder.org/v2/gh/nuslds/intro-beautifulsoup/master/) - Open the notebook Part1\_Warmup.ipynb.

# Part II. Inspecting page source & HTML elements

## **What happens when you type an URL in the browser and press enter?**



## **What you see vs. What your crawler sees –**

## **How to view the page source on Chrome**

* Go the website for demonstration - [Quotes to Scrape](http://quotes.toscrape.com/)
* Right click - Select View page source

## **How to inspect HTML element on Chrome**

* Go the website for demonstration - [Quotes to Scrape](http://quotes.toscrape.com/)
* Right click - Select Inspect
* Click on Select Element tool
* Hover over the element you want to inspect

## **Activity 1**

Inspect the quotes on the page and complete Activity 1 in the Activity Sheet.

Part III. Extracting Data from HTML with BeautifulSoup

In Part III, we will create a reusable function to extract information below from the HTML strings of a single quote on [Quotes to Scrape](http://quotes.toscrape.com/).

| **Variable Name** | **Description** |
| --- | --- |
| quote\_text | Text of the quote |
| author | Name of the author |
| author\_url | URL of the author page, e.g. '/author/Albert-Einstein' |
| tags | Tags that assigned to the quote |

**The Steps**

**Step 1. Inspecting HTML Elements**

It is important to inspect the HTML elements before you write you codes to extract information from the HTML strings.

**HTML Element**

An HTML element usually starts with a **starting tag** and ends with a **closing tag** in which the element name is prefixed with a slash.

Example: <small>Albert Einstein</small>

| **Start Tag** | **Content** | **End Tag** |
| --- | --- | --- |
| <small> | Albert Einstein | </small> |

**Attributes**

Attributes, placing inside the starting tags, define the characteristics of HTML elements. An attribute is always a **name-value pair**. Example: <small class="author">Albert Einstein</small>

| **Tag Name** | **Attribute** | **Attribute Name** | **Attribute Value** |
| --- | --- | --- | --- |
| small | class="author" | class | author |

**Activity 2**

Now let's complete **Activity 2** on the **Activity Sheet**. You will use the information you fill in the table when you code.

**Step 2. Enjoy Coding**

Please go to the [Jupyter Notebooks on Binder](https://mybinder.org/v2/gh/nuslds/intro-beautifulsoup/master/) - Open the notebook Part3\_Extracting\_Data\_from\_HTML\_with\_BeautifulSoup.ipynb.

**The main steps**



Part IV. Scraping web pages with Requests and BeautifulSoup

In part III, we will create a reusable function that scrapes quotes from the website by stating the range of the pages and save the outputs as CSV.

# extract quotes from page 1 to 10  
outputs = scrape\_quotes(1, 10)  
outputs.head()

| **author** | **author\_url** | **quote\_text** | **tags** |
| --- | --- | --- | --- |
| Albert Einstein | <http://quotes.toscrape.com/author/Albert-Einstein> | “The world as we have created it is a process ... | change;deep-thoughts;thinking;world |
| J.K. Rowling | <http://quotes.toscrape.com/author/J-K-Rowling> | “It is our choices, Harry, that show what we t... | abilities;choices |
| Albert Einstein | <http://quotes.toscrape.com/author/Albert-Einstein> | “There are only two ways to live your life. On... | inspirational;life;live;miracle;miracles |
| Jane Austen | <http://quotes.toscrape.com/author/Jane-Austen> | “The person, be it gentleman or lady, who has ... | aliteracy;books;classic;humor |

**The Steps**

**Step 1. Inspecting the patterns**

Try to be sensitive in observing different patterns before you write you codes.

In **Activity 3**, we will be figuring out the patterns of the URL of each page.

**Step 2. Enjoy Coding**

Please go to the [Jupyter Notebooks on Binder](https://mybinder.org/v2/gh/nuslds/intro-beautifulsoup/master/) - Open the notebook Part4\_Scraping\_Web\_Pages\_with\_Requests\_and\_BeautifulSoup.

**The main steps**

