wbs
WARWICK BUSINESS SCHOOL
THE UNIVERSITY OF WARWICK

For the Change Makers

Programming for Data Analytics

Week 2: Data Collection
Information Systems and Management
Warwick Business School

Web Scraping

Legal issues

- The legality of web scraping is really hard to determine.
- Location.
  - USA (CFAA, DMCA, etc)
  - EU (GDPR, etc)
- Scraping.
- Data.

Some basis of HTML

- Webpages are created using **HTML (Hypertext Markup Language)**, with CSS (Cascading Style Sheets) and JavaScript.
- HTML is a standard markup language (others like XML, LaTex) that uses tags to annotate the webpage content so they can be displayed as desired.

```latex
\documentclass{article} % Starts an article
\usepackage{amsmath} % Imports amsmath
\title{\LaTeX} % Title

\begin{document} % Begins a document
  \maketitle
  \LaTeX{} is a document preparation system for
  the \TeX{} typesetting program. It offers
  programmable desktop publishing features and
  extensive facilities for automating most
  aspects of typesetting and desktop publishing,
  including numbering and  cross-referencing,
  tables and figures, page layout,
  bibliographies, and much more. \LaTeX{} was
  originally written in 1984 by Leslie Lamport
  and has become the  dominant method for using
  \TeX; few people write in plain \TeX{} anymore.
  The current version is \LaTeXe.

  % This is a comment, not shown in final output.
  % The following shows typesetting  power of LaTeX:
  \begin{align}
    E_0 &= mc^2 \\
    E &= \frac{mc^2}{\sqrt{1-\frac{v^2}{c^2}}}
  \end{align}
\end{document}
```

LaTeX

LaTeX is a document preparation system for the TeX typesetting program. It offers programmable desktop publishing features and extensive facilities for automating most aspects of typesetting and desktop publishing, including numbering and cross-referencing, tables and figures, page layout, bibliographies, and much more. LaTeX was originally written in 1984 by Leslie Lamport and has become the dominant method for using TeX; few people write in plain TeX anymore. The current version is LaTeX 2_ε.

$$E_0 = mc^2 \tag{1}$$

$$E = \frac{mc^2}{\sqrt{1-\frac{v^2}{c^2}}} \tag{2}$$

A simple html example

```
<!DOCTYPE html>
<html>
<head>
<title>A simple html example</title>
</head>
<body>
<p id="first1"> Zhewei </p>
<p id="last1"> Zhang </p>
</body>
</html>
```

A simple html example

```
<!DOCTYPE html>
<html>
<head>
<title>A simple html example</title>
</head>
<body>
<p class = "name" id="first1"> Zhewei </p>
<p class = "name" id="last1"> Zhang </p>
</body>
</html>
```

- Most tags will be used in pairs with opening tag and closing tag, i.e. <head> and </head>.
- Content between a pair of opening and closing tags will be displayed accordingly, which is also called element in a html file.

A simple html example

```
<!DOCTYPE html>
<html>
<head>
<title>A simple html example</title>
</head>
<body>
<p class = "name" id="first1"> Zhewei </p>
<p class = "name" id="last1"> Zhang </p>
</body>
</html>
```

- All HTML elements can have attributes that provide additional information about that element.
- Attributes are always specified in the opening tag.
- For web scraping, we normally deal with two kinds of attributes: id and class.
  - id can uniquely identify individual elements.
  - class can identify a set (class) of similar elements.

Web scraping in two steps

1. Access the webpage and download the html content.

- Requests
- https://2.python-requests.org/en/master/user/quickstart/

2. Extract elements (data) from a webpage, which can be located by the tags and attributes.

- BeautifulSoup
- https://www.crummy.com/software/BeautifulSoup/

Fetch the webpage with Requests

- Requests is a HTTP library for Python that help you to make http request to the web server and fetch the webpage.
- You send a request to the web server asking for certain webpages.
- You can make **GET** and **POST** request.

```
import requests
url = "http://www.wbs.ac.uk"
result = requests.get(url)
```

Requests results

- The get() function will send a http request to retrieve the webpage specified. The only required argument for get() is the url address to the webpage.
- Returns a requests object that contains various information about the webpage retrieved:
  - status_code: 200 is good and 4xx (403,404,408,etc.) is bad.
  - encoding: how characters are coded digitally, utf-8 is generally the best.
  - text: the plain content of the webpage, including all markings.

Retrieve links with parameters

- Some webpages, such as search results, may require you to send necessary parameters.

  http://yourmainlink.com/search?name=python&format=ebook&lang=en

- You can pass those parameters by using a dictionary as the argument param.

```
payload = {'key1' : 'value1', 'key2' : 'value2'}
payload = {'name' : 'python', 'format' : 'ebook', 'lang' : 'en'}
result = requests.get(url,params=payload)
```

Deal with limits

- Regardless legal issue, websites generally do not like scraping as it can take up the bandwidth. Therefore, many websites limit the number of requests you can make within certain time period. Your requests can be blocked after reaching the limit.
- So, be considerate, and pause a while if needed.
- proxy and headers can be provided as additional arguments to the get().

Exercise

- Go to eBay, try search something with different criteria.
- Write the urls that you may use to download the html content of the search results below:
  - Used iphone 12 in the price range 500 – 850
  - New listing first
  - First three pages of results

Parsing your result with BeautifulSoup

- You need to parse the text retrieved by Requests: convert the plain text into structured data.
- BeautifulSoup is a parser library to create such tree-structured data by parsing HTML or XML files.
- You would need to give at least two arguments: the string to parse and the parser to use

```
from bs4 import BeautifulSoup
url_html = result.text
url_content = BeautifulSoup(url_html, 'html.parser')
```

Navigate the parsed data

- Parsed result will be return as a BeautifulSoup object that has a range of methods to help you navigate and search the data.
  1. Using tag names directly.

```
url_content.head # return the first matching element.
```

  2. Using find() and find_all()

```
url_content.find_all('p') # find_all() returns a list
```

Tags and attributes can be passed as arguments.

  3. Using select()

```
url_content.select() # use CSS selectors instead of HTML tags.
```

Exercise

- Write a script to extract FTSE indices from London Stock Exchange and save to a csv file.
- https://www.londonstockexchange.com/exchange/prices-and-markets/stocks/indices/ftse-indices.html