Web Scraping using Python

**1. Learn this from Youtube** <https://www.google.com/search?rlz=1C5CHFA_enCA911CA911&tbm=vid&sxsrf=AJOqlzXowfz8T0GccjEbPEGtXRTM-P3JhQ:1676958988456&q=python+web+scrape+using+beautifulsoup+codecademy&spell=1&sa=X&ved=2ahUKEwiW8LyG96X9AhXwGjQIHZhCDUEQBSgAegQIERAB&biw=1514&bih=707&dpr=1#fpstate=ive&vld=cid:19c6a581,vid:GjKQ6V_ViQE>

**2. Use this HTML page to practise**: <https://keithgalli.github.io/web-scraping/example.html>

**3. Use BeautlfulSoup to scrap pages.**

* Load the required Library: requests and BeautlfulSoup (<https://www.crummy.com/software/BeautifulSoup/bs4/doc/> )

Graphical user interface, text, application, email

Description automatically generated

* In general, there are following simple ways to navigate the data structure.

Graphical user interface, text, application, email

Description automatically generated

* Normally we only need to deal with four kinds of objects: Tag, NavigableString, BeautifulSoup, and Comment.
  + A Tag object corresponds to an XML or HTML tag in the original document:

Graphical user interface, text, application, email

Description automatically generated

* + The BeautifulSoup object represents the parsed document as a whole. For most purposes, you can treat it as a Tag object.

Graphical user interface, text, application, email

Description automatically generated

* + The difference between Tag and BeautifulSoup object is subtle, but can be very big, as follows:
    - you can use find(“li”) to find li tag when it is BeautifulSoup object, but it wont work when it is already a li tag, but you still use find(“li”). You can only find nested tag inside this li tag now.

Graphical user interface, text, application, email

Description automatically generated

* find() and find\_all(): find the tag(s).
  + find() only returns the first occurrence, while find\_all() returns all the occurrence.
  + You can also pass a list of tags to look for, but again, find() only returns the first occurrence from any tag in the tag list, while find\_all() returns all the occurrence.

Graphical user interface, text, application, email

Description automatically generated

* + pass attributes to the find() or find\_all() functions.

A picture containing text

Description automatically generated

* + Nest find() and find\_all().

Graphical user interface, text, application, email

Description automatically generated

* You can also search for specific strings in find/find\_all calls
  + You use regular expression inside find/find\_all.
  + For example, you use (|) to do multiple selections.

Graphical user interface, text

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* The keyword argument in find/find\_all calls.
  + For most keyword, it is simply using its html keyword like id, href, attrs…

Text, application

Description automatically generated

* Use 2 keywords at the same time.

Text

Description automatically generated

* + class is a little special, because in Python, class is a keyword. You will need to use class\_ as the keyword argument.

Text

Description automatically generated

* Selector.
  + BeautifulSoup has a .select() method which uses the SoupSieve package to run a CSS selector against a parsed document and return all the matching elements. Tag has a similar method which runs a CSS selector against the contents of a single tag.
    - NOTE, it returns a list with that specific tag
  + Example:
    - #For example, selecting all the paragraph tag in the page. This basically is the same thing as: finding all of "p".

Text

Description automatically generated

* + - #But then I only want to grab p tag inside div, or # I only want to grab p tag that is strictly following h2 tag

Graphical user interface, text, application, email

Description automatically generated

* + - One thing note is that you can also nest calls.

Graphical user interface, text, application, email

Description automatically generated

**4. Get different properties of the HTML**

* How to get only the string from the page
  + If the returned tag does not contain child, you can use .string, otherwise, you use .get\_text()

Graphical user interface, text, application, email

Description automatically generated

* How to get the link.
  + Get a specific property from an element using [].
  + A tag may have any number of attributes. The tag <b id="boldest"> has an attribute “id” whose value is “boldest”. You can access a tag’s attributes by treating the tag like a dictionary:

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

* Code Navigation.
  + Path Syntax: in this case, it only returns the first occurrence.
  + Know the terms: parents, child, sibling

Graphical user interface, text, application, email

Description automatically generated

**5. Exercise.**

* Grab all of the social links from the webpage.
  + HTML

Graphical user interface, text, application

Description automatically generated

* + Code

Text

Description automatically generated

* Grab HTML table into a pandas DataFrame
  + HTML

Graphical user interface, text, application

Description automatically generated

* + Code

Table

Description automatically generated

* Grab all the images and show / download them locally

Text

Description automatically generated