Step 1&2 Installing beautifulsoup4 and lxml

In [1]:
```
!pip install beautifulsoup4
```

```
Requirement already satisfied: beautifulsoup4 in /Users/shyam/opt/anaconda3/lib/python3.9/site-packages (4.1
1.1)
Requirement already satisfied: soupsieve>1.2 in /Users/shyam/opt/anaconda3/lib/python3.9/site-packages (from
beautifulsoup4) (2.3.1)
```

In [2]:
```
!pip install lxml
```

```
Requirement already satisfied: lxml in /Users/shyam/opt/anaconda3/lib/python3.9/site-packages (4.8.0)
```

3. Connect to the website using urllib.request

4. Create an object called ‘soup’ using the source

5. Print page title, get attributes, values, and beginning navigation. Get specific values.

6. Find all the paragraph instead of just one in the previous query

7. You can iterate through them using ‘str(paragraph.text)’

8. Grab all the links through ‘url.get(‘here)

9. Just grab the text using ‘get_text’

In [12]:
```
import urllib.request
from bs4 import BeautifulSoup

#3: Connect to the website
url = "https://pythonprogramming.net/parsememcparseface/"
response = urllib.request.urlopen(url)

#4: Create a BeautifulSoup object
soup = BeautifulSoup(response, 'html.parser')
```

```
#5: Print page title
print("\n Page Title: \n", soup.title.string)

# Get attributes and values
link = soup.find('a')
print("\n Link tag attributes: \n", link.attrs)

#6: Find all paragraphs
paragraphs = soup.find_all('p')

# Step 7: Iterate through paragraphs and print their text
print("\nParagraphs : ")
for paragraph in paragraphs:
    print(str(paragraph.text))

#8: Find all links
links = soup.find_all('a')


#9: Iterating through links and print their text
print("\nLinks \n")

for link in links:
  print("Link :", link.get("href"), "--> Text:", link.string)
```

```
 Page Title:
 Python Programming Tutorials

 Link tag attributes:
 {'href': '/', 'class': ['brand-logo']}

Paragraphs :
Oh, hello! This is a wonderful page meant to let you practice web scraping. This page was originally created
to help people work with the Beautiful Soup 4 library.
The following table gives some general information for the following programming languages:
I think it's clear that, on a scale of 1-10, python is:
Javascript (dynamic data) test:
y u bad tho?
Whɑt hɑppéns now¿
sitemap
Contact: Harrison@pythonprogramming.net.
Programming is a superpower.

Links

Link : / --> Text: None
Link : # --> Text: None
Link : / --> Text: Home
Link : /+=1/ --> Text: +=1
Link : /support/ --> Text: Support the Content
Link : https://goo.gl/7zgAVQ --> Text: None
Link : /login/ --> Text: Log in
Link : /register/ --> Text: Sign up
Link : / --> Text: Home
Link : /+=1/ --> Text: +=1
Link : /support/ --> Text: Support the Content
Link : https://goo.gl/7zgAVQ --> Text: None
Link : /login/ --> Text: Log in
Link : /register/ --> Text: Sign up
Link : https://www.crummy.com/software/BeautifulSoup/bs4/doc/ --> Text: Beautiful Soup 4
Link : /sitemap.xml --> Text: sitemap
Link : /support-donate/ --> Text: Support this Website!
Link : /consulting/ --> Text: Consulting and Contracting
Link : https://www.facebook.com/pythonprogramming.net/ --> Text: Facebook
Link : https://twitter.com/sentdex --> Text: Twitter
Link : https://instagram.com/sentdex --> Text: Instagram
Link : /about/tos/ --> Text: Terms and Conditions
Link : /about/privacy-policy/ --> Text: Privacy Policy
Link : https://xkcd.com/353/ --> Text: Programming is a superpower.
```