Web Scrapping.

YouTube (https://www.youtube.com/watch?v=ng2o98k983k) - Corey Schafer (https://github.com/CoreyMSchafer)

Prepared by: Sagun Shakya (https://github.com/sagsshakya)

- GITAM Institute of Science.

Setting up the working directory.

In [58]:
```
import os
os.chdir(r'C:\Users\acer\Desktop\PythonProgramming\nlp-in-python-tutorial-master\nlp-in-python-tutorial-mas
```

Web Scrapping part.

In [3]:
```
from bs4 import BeautifulSoup
import requests
```

Getting the data from teh web page.

In [4]:
```
source = requests.get('https://coreyms.com/').text
```

In [5]:
```
# Basic Parsers.
import lxml
import html5lib
```

In [7]:
```
soup = BeautifulSoup(source, 'lxml')
```

In [8]:
```
print(soup.prettify())    # Pretiffy is used to indent the tags at proper places.
```

```
 <title>
  CoreyMS - Development, Design, DIY, and more
 </title>
 <!-- This site is optimized with the Yoast SEO plugin v13.4.1 - https://yoast.com/wordpress/plugins/seo/ (https://yoast.com/wordpress/plugins/seo/) --
>
 <meta content="Development, Design, DIY, and more" name="description"/>
 <meta content="max-snippet:-1, max-image-preview:large, max-video-preview:-1" name="robots"/>
 <link href="https://coreyms.com/" rel="canonical"/>
 <link href="http://coreyms.com/page/2" rel="next"/>
 <meta content="en_US" property="og:locale"/>
 <meta content="website" property="og:type"/>
 <meta content="CoreyMS - Development, Design, DIY, and more" property="og:title"/>
 <meta content="Development, Design, DIY, and more" property="og:description"/>
 <meta content="https://coreyms.com/" property="og:url"/>
 <meta content="CoreyMS" property="og:site_name"/>
 <meta content="https://coreyms.com/wp-content/uploads/2014/11/SocialIcon.png" property="og:image"/>
 <meta content="https://coreyms.com/wp-content/uploads/2014/11/SocialIcon.png" property="og:image:secure_url"/>
 <meta content="800" property="og:image:width"/>
 <meta content="800" property="og:image:height"/>
 <meta content="summary_large_image" name="twitter:card"/>
 <meta content="Development, Design, DIY, and more" name="twitter:description"/>
 <meta content="CoreyMS - Development, Design, DIY, and more" name="twitter:title"/>
 <meta content="@CoreyMSchafer" name="twitter:site"/>
 <meta content="http://coreyms.com/wp-content/uploads/2014/11/SocialIcon.png" name="twitter:image"/>
 <meta content="6DD015B65209B2918B7865FB2180F677" name="msvalidate.01"/>
```

In [23]:
```
myarticle = soup.find('article')
#print(myarticle.prettify())
headline = myarticle.header.h2.a.text    # Look at INspect Element in the original site.
print(headline)
```

```
Python Tutorial: Zip Files – Creating and Extracting Zip Archives
```

Getting the paragraph text.

In [52]:
```
summary = myarticle.find('div', class_ = 'entry-content').text
```

In [53]:
```
print(summary)
```

```
In this Python Programming Tutorial, we will be clarifying the issues with mutable default arguments. We discussed this in my last video t
itled “5 Common Python Mistakes and How to Fix Them”, but I received many comments from people who were still confused. So we will be doin
g a deeper dive to explain exactly what is going on here. Let’s get started…
```

Getting the video ID from an embedded YouTube video file.

In [29]:
```
vid_src = myarticle.find('iframe', class_ = 'youtube-player')
print(vid_src)
```

```
<iframe allowfullscreen="true" class="youtube-player" height="360" src="https://www.youtube.com/embed/z0gguhEmWiY?version=3&amp;rel=1&amp;
fs=1&amp;autohide=2&amp;showsearch=0&amp;showinfo=1&amp;iv_load_policy=1&amp;wmode=transparent" style="border:0;" type="text/html" width
="640"></iframe>
```

We don't need the text. We only need the sourceID which is the set of numbers before the ?version i.e z0gguhEmWiY.

In [31]:
```
# Accessing the URL like a value in a dictionary.
vid_src = myarticle.find('iframe', class_ = 'youtube-player')['src']
print(vid_src)
```

```
https://www.youtube.com/embed/z0gguhEmWiY?version=3&rel=1&fs=1&autohide=2&showsearch=0&showinfo=1&iv_load_policy=1&wmode=transparent (https://www.youtube.com/embed/z0gguhEmWiY?version=3&rel=1&fs=1&autohide=2&showsearch=0&showinfo=1&iv_load_policy=1&wmode=transparent)
```

In [37]:
```
# Splitting the items using '/'.
vid_id = vid_src.split('/')
print(vid_id)
```

```
['https:', '', 'www.youtube.com', 'embed', 'z0gguhEmWiY?version=3&rel=1&fs=1&autohide=2&showsearch=0&showinfo=1&iv_load_policy=1&wmode=transparent']
```

In [38]:
```
# Taking the fourth element.
vid_id = vid_id[4]
print(vid_id)
```

```
z0gguhEmWiY?version=3&rel=1&fs=1&autohide=2&showsearch=0&showinfo=1&iv_load_policy=1&wmode=transparent
```

In [39]:
```
# Splitting using '?'.
vid_id = vid_id.split('?')
print(vid_id)
```

```
['z0gguhEmWiY', 'version=3&rel=1&fs=1&autohide=2&showsearch=0&showinfo=1&iv_load_policy=1&wmode=transparent']
```

In [40]:
```
# Taking the oth element.
vid_id = vid_id[0]
print(vid_id)
```

```
z0gguhEmWiY
```

Generating our own YouTube link for the video.

In [41]:
```
yt_link = f'https://www.youtube.com/watch?v={vid_id}'
print(yt_link)
```

```
https://www.youtube.com/watch?v=z0gguhEmWiY (https://www.youtube.com/watch?v=z0gguhEmWiY)
```

Using find_all to generate the headlines, sumarries and youtube links from the webpage.

In [49]:
```
import csv
```

In [57]:

```
csv_file = open('cms_scrape_new.csv', 'w')
csv_writer = csv.writer(csv_file)
csv_writer.writerow(['Headline', 'Summary', 'Video Link'])

for myarticle in soup.find_all('article'):
    #print(myarticle.prettify())
    headline = myarticle.header.h2.a.text    # Look at INspect Element in the original site.
    print(headline)
    print()
    ### Getting the paragraph text.

    summary = myarticle.find('div', class_ = 'entry-content').p.text

    print(summary)
    print()
    vid_src = myarticle.find('iframe', class_ = 'youtube-player')
    #print(vid_src)

    '''We don't need the text. We only need the sourceID
     which is the set of numbers before the ?version i.e z0gguhEmWiY.'''

    try:
        # Accessing the URL like a value in a dictionary.
        vid_src = myarticle.find('iframe', class_ = 'youtube-player')['src']
        #print(vid_src)

        # Splitting the items using '/'.
        vid_id = vid_src.split('/')

        # Taking the fourth element.
```

```
            vid_id = vid_id[4]

            # Splitting using '?'.
            vid_id = vid_id.split('?')

            # Taking the oth element.
            vid_id = vid_id[0]

            #### Generating our own YouTube link for the video.
            yt_link = f'https://www.youtube.com/watch?v={vid_id}'

        except Exception as e:
            yt_link = None

        print(yt_link)
        print('\n\n\n')
        csv_writer.writerow([headline, summary, yt_link])
    csv_file.close()
```

```
Python Tutorial: Zip Files – Creating and Extracting Zip Archives

In this video, we will be learning how to create and extract zip archives. We will start by using the zipfile module, and then we will see how to do this using the shutil module. We will learn how to do this with single files and directories, as well as learning how to use gzip as well. Let’s get started…

https://www.youtube.com/watch?v=z0gguhEmWiY (https://www.youtube.com/watch?v=z0gguhEmWiY)



Python Data Science Tutorial: Analyzing the 2019 Stack Overflow Developer Survey

In this Python Programming video, we will be learning how to download and analyze real-world data from the 2019 Stack Overflow Developer Survey. This is terrific practice for anyone getting into the data science field. We will learn different ways to analyze this data and als
```

o some best practices. Let's get started…

https://www.youtube.com/watch?v=_P7X8tMplsw (https://www.youtube.com/watch?v=_P7X8tMplsw)

Python Multiprocessing Tutorial: Run Code in Parallel Using the Multiprocessing Module

In this Python Programming video, we will be learning how to run code in parallel using the multiprocessing module. We will also look at h ow to process multiple high-resolution images at the same time using a ProcessPoolExecutor from the concurrent.futures module. Let's get s tarted…

https://www.youtube.com/watch?v=fKl2JW_qrso (https://www.youtube.com/watch?v=fKl2JW_qrso)

Python Threading Tutorial: Run Code Concurrently Using the Threading Module

In this Python Programming video, we will be learning how to run threads concurrently using the threading module. We will also look at how to download multiple high-resolution images online using a ThreadPoolExecutor from the concurrent.futures module. Let's get started…

https://www.youtube.com/watch?v=IEEhzQoKtQU (https://www.youtube.com/watch?v=IEEhzQoKtQU)

Update (2019-09-03)

Hey everyone. I wanted to give you an update on my videos. I will be releasing videos on threading and multiprocessing within the next wee k. Thanks so much for your patience. I currently have a temporary recording studio setup at my Airbnb that will allow me to record and edi t the threading/multiprocessing videos. I am going to be moving into my new house in 10 days and once I have my recording studio setup the n you can expect much faster video releases. I really appreciate how patient everyone has been while I go through this move, especially th ose of you who are contributing monthly through YouTube

None

Python Quick Tip: The Difference Between "==" and "is" (Equality vs Identity)

In this Python Programming Tutorial, we will be learning the difference between using “==” and the “is” keyword when doing comparisons. Th e difference between these is that “==” checks to see if values are equal, and the “is” keyword checks their identity, which means it’s go ing to check if the values are identical in terms of being the same object in memory. We’ll learn more in the video. Let’s get started…

https://www.youtube.com/watch?v=mO_dS3rXDIs (https://www.youtube.com/watch?v=mO_dS3rXDIs)

Python Tutorial: Calling External Commands Using the Subprocess Module

In this Python Programming Tutorial, we will be learning how to run external commands using the subprocess module from the standard librar y. We will learn how to run commands, capture the output, handle errors, and also how to pipe output into other commands. Let’s get starte d…

https://www.youtube.com/watch?v=2Fp1N6dof0Y (https://www.youtube.com/watch?v=2Fp1N6dof0Y)

Visual Studio Code (Windows) – Setting up a Python Development Environment and Complete Overview

In this Python Programming Tutorial, we will be learning how to set up a Python development environment in VSCode on Windows. VSCode is a very nice free editor for writing Python applications and many developers are now switching over to this editor. In this video, we will l earn how to install VSCode, get the Python extension installed, how to change Python interpreters, create virtual environments, format/lin t our code, how to use Git within VSCode, how to debug our programs, how unit testing works, and more. We have a lot to cover, so let’s go ahead and get started…

https://www.youtube.com/watch?v=-nh9rCzPJ20 (https://www.youtube.com/watch?v=-nh9rCzPJ20)

Visual Studio Code (Mac) – Setting up a Python Development Environment and Complete Overview

In this Python Programming Tutorial, we will be learning how to set up a Python development environment in VSCode on MacOS. VSCode is a ve ry nice free editor for writing Python applications and many developers are now switching over to this editor. In this video, we will lear n how to install VSCode, get the Python extension installed, how to change Python interpreters, create virtual environments, format/lint o ur code, how to use Git within VSCode, how to debug our programs, how unit testing works, and more. We have a lot to cover, so let’s go ah ead and get started…

https://www.youtube.com/watch?v=06I63_p-2A4 (https://www.youtube.com/watch?v=06I63_p-2A4)

```
Clarifying the Issues with Mutable Default Arguments

In this Python Programming Tutorial, we will be clarifying the issues with mutable default arguments. We discussed this in my last video t
itled “5 Common Python Mistakes and How to Fix Them”, but I received many comments from people who were still confused. So we will be doin
g a deeper dive to explain exactly what is going on here. Let’s get started…

https://www.youtube.com/watch?v=_JGmemuINww (https://www.youtube.com/watch?v=_JGmemuINww)
```

The End.