

## Scripting with Python

[https://en.wikipedia.org/wiki/Python\\_Imaging\\_Library](https://en.wikipedia.org/wiki/Python_Imaging_Library)

<https://pillow.readthedocs.io/en/stable/>

```
In [1]: pip install Pillow

Requirement already satisfied: Pillow in c:\users\krishnaaleti\anaconda3\lib\site-packages (8.0.1)
Note: you may need to restart the kernel to use updated packages.

https://www.geeksforgeeks.org/os-module-python-examples/

https://www.tutorialspoint.com/python/os\_mkdir.htm
https://www.ibm.com/docs/en/aix/7.2?topic=modes-numeric-representation-access

"D:\Edu\Python_ZTM\Scripting_Images"

"D:\Edu\Python_ZTM\Scripting_PDF"

"D:\Edu\Python_ZTM\Scripting_emails"

"D:\Edu\Python_ZTM\Scripting_Passwordchecker"
```

### Hash Generators

<https://www.miraclesalad.com/webtools/md5.php>

<https://passwordsgenerator.net/sha1-hash-generator/>

Some key aspects of hash functions.

1. It's one way.In the sense that if I give somebody Hash of password, they have no idea what the input was and

it's practically impossible for me to have any clue as to what the input is. 2. Hash is always same for the same input, it won't change until we change the input. If I give it the same input, it's going to give me the same result.But I have no idea how to convert this back into hello. - Idempotent(saying that a function, given an input, always outputs the same output.)

Technically, a hash function that we use for hash tables is going to take input, generate some sort of gibberish and then convert it to an index base or an address base that it has based on this hash.

To review, we have a key that is grapes, we send it through a hash function that is going to hash something

really, really fast and then map whatever the hash came out to be into a memory address where we want

to store our Data which is graps=10000.

\*\*\*we usually assume a time, complexity or bigo notation of O1, because this happens really fast.

## Twitter bot

<https://developer.twitter.com/en/docs/basics/developer-portal/faq>

<https://developer.twitter.com/en/docs/twitter-api>

<https://developer.twitter.com/en/apply-for-access>

Tweepy: <https://www.tweepy.org/>

```
In [ ]: #You will need to PIP INSTALL tweepy for this to work and also create a twitter API. Run this on your own machine, not in this Repl.
import tweepy
import time

consumer_key = ''
consumer_secret = ''
access_token = ''
access_token_secret = ''
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth)

user = api.me()
print (user.name) #prints your name.
print (user.screen_name)
print (user.followers_count)

search = "zerotomastery"
numberOfTweets = 2

def limit_handle(cursor):
    while True:
        try:
            yield cursor.next()
        except tweepy.RateLimitError:
            time.sleep(1000)

#Be nice to your followers. Follow everyone!
for follower in limit_handle(tweepy.Cursor(api.followers).items()):
    if follower.name == 'Usernamehere':
        print(follower.name)
        follower.follow()

# Be a narcissist and Love your own tweets. or retweet anything with a keyword!
for tweet in tweepy.Cursor(api.search, search).items(numberOfTweets):
    try:
        tweet.favorite()
        print('Retweeted the tweet')
    except tweepy.TweepError as e:
        print(e.reason)
    except StopIteration:
        break
```

SMS with Python: <https://www.twilio.com/docs>

## Scraping Data with Python

Webscrapping usecases: <https://www.excellerate.com/blogs/web-scraping-introduction-applications-and-best-practices/#:~:text=Web%20scraping%20typically%20extracts%20large,show%20data%20from%20a%20website.>

url: <https://www.airbnb.co.in/>

use robots.txt at the end of url to check what is allowed and not allowed for scraping of a particular website: <https://www.airbnb.co.in/robots.txt> (whatever not present here can be scraped)

actually we can scrap everything and no one can prevent us but we should be ethical and should only scrap allowed items.

The only thing that they ask is that we have a crawl delay of 30, that is if we're going to crawl their Web site, we should have a pause so that we don't constantly use programs to hit the Web site because, well, this is going to overload their servers and maybe slow down their website.

chrome plugin of webscraper:- <https://chrome.google.com/webstore/detail/web-scraper-free-web-scrapi/jnhgnonknehpejjnehehlklipmbmhnhl=en>

some APIs which gives us the infromation for free: <https://swapi.dev/>

<https://swapi.dev/api/people>

<https://jsonplaceholder.typicode.com/>

The easiest way to get data from a website is usually using an API.

**Scraping**

Scraping Hacker rank and getting news having more than 100 votes

<https://www.crummy.com/software/BeautifulSoup/>

<https://www.crummy.com/software/BeautifulSoup/bs4/doc/>

[https://developer.mozilla.org/en-US/docs/Learn/CSS/Building\\_blocks/Selectors](https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Selectors)

<https://docs.python.org/3/library/pprint.html>

<https://scrapy.org/>

In [ ]: