

▼ NAME : ANUB DANIEL J

ROLL NO : 205229101

▼ Getting User Data

Install PyGithub requests and get information about your GitHub Profile

```
!pip install pygithub requests
```

```
Requirement already satisfied: pygithub in /usr/local/lib/python3.7/dist-packages (1.55)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (2.23)
Requirement already satisfied: deprecated in /usr/local/lib/python3.7/dist-packages (from requests) (1.2.12)
Requirement already satisfied: pynacl>=1.4.0 in /usr/local/lib/python3.7/dist-packages (from pygithub) (1.5.0)
Requirement already satisfied: pyjwt>=2.0 in /usr/local/lib/python3.7/dist-packages (from pygithub) (2.4.0)
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests) (2.10)
Requirement already satisfied: urllib3!=1.25.0,!<1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests) (1.26.5)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-packages (from requests) (2021.10.8)
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-packages (from requests) (3.0.4)
Requirement already satisfied: cffi>=1.4.1 in /usr/local/lib/python3.7/dist-packages (from pynacl->pygithub) (1.14.6)
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from pynacl->pygithub) (1.16.0)
Requirement already satisfied: pycparser in /usr/local/lib/python3.7/dist-packages (from cffi->pynacl->pygithub) (2.21)
Requirement already satisfied: wrapt<2,>=1.10 in /usr/local/lib/python3.7/dist-packages (from deprecated->requests) (1.12.1)
```

```
import requests
from pprint import pprint
username = "ptwobrussell"
url = f"https://api.github.com/users/{username}"
user_data = requests.get(url).json()
pprint(user_data)
```

```
{'avatar_url': 'https://avatars.githubusercontent.com/u/98668?v=4',
 'bio': None,
 'blog': 'http://ptwobrussell.me',
 'company': None,
 'created_at': '2009-06-25T00:56:57Z',
 'email': None,
 'events_url': 'https://api.github.com/users/ptwobrussell/events{/privacy}',
 'followers': 899,
 'followers_url': 'https://api.github.com/users/ptwobrussell/followers',
 'following': 0,
 'following_url': 'https://api.github.com/users/ptwobrussell/following{/other_user}',
 'gists_url': 'https://api.github.com/users/ptwobrussell/gists{/gist_id}',
 'gravatar_id': '',
 'hireable': None,
 'html_url': 'https://github.com/ptwobrussell',
 'id': 98668,
 'location': 'Franklin, TN',
```

```

'login': 'ptwobrussell',
'name': 'Matthew A. Russell',
'node_id': 'MDQ6VXNlcjk4NjY4',
'organizations_url': 'https://api.github.com/users/ptwobrussell/orgs',
'public_gists': 12,
'public_repos': 25,
'received_events_url': 'https://api.github.com/users/ptwobrussell/received\_events',
'repos_url': 'https://api.github.com/users/ptwobrussell/repos',
'site_admin': False,
'starred_url': 'https://api.github.com/users/ptwobrussell/starred{/owner}/{/repo}',
'subscriptions_url': 'https://api.github.com/users/ptwobrussell/subscriptions',
'twitter_username': None,
'type': 'User',
'updated_at': '2021-08-27T01:27:35Z',
'url': 'https://api.github.com/users/ptwobrussell'}

```

## ▼ Getting Public repositories of a user

```

import base64
from github import Github
from pprint import pprint

username = "ptwobrussell"
g = Github()
user = g.get_user(username)
for repo in user.get_repos():
    print(repo)

Repository(full_name="ptwobrussell/amplify-ios")
Repository(full_name="ptwobrussell/aws-sdk-ios")
Repository(full_name="ptwobrussell/bootstrap-combobox")
Repository(full_name="ptwobrussell/brat")
Repository(full_name="ptwobrussell/django-cacheback")
Repository(full_name="ptwobrussell/etc")
Repository(full_name="ptwobrussell/FuXi")
Repository(full_name="ptwobrussell/go")
Repository(full_name="ptwobrussell/gspread")
Repository(full_name="ptwobrussell/heroku-django-template")
Repository(full_name="ptwobrussell/ipython")
Repository(full_name="ptwobrussell/jpype")
Repository(full_name="ptwobrussell/jquery-scrolldepth")
Repository(full_name="ptwobrussell/Legal-Forms")
Repository(full_name="ptwobrussell/Mining-the-Social-Web")
Repository(full_name="ptwobrussell/Mining-the-Social-Web-2nd-Edition")
Repository(full_name="ptwobrussell/PayPal-APIs-Up-and-Running")
Repository(full_name="ptwobrussell/pinax-starter-projects")
Repository(full_name="ptwobrussell/pinax-theme-bootstrap")
Repository(full_name="ptwobrussell/python-boilerpipe")
Repository(full_name="ptwobrussell/python-linkedin")
Repository(full_name="ptwobrussell/Recipes-for-Mining-Twitter")
Repository(full_name="ptwobrussell/SwiftyCam")
Repository(full_name="ptwobrussell/Tweet-Relevance")
Repository(full_name="ptwobrussell/twitter")

```

## ▼ Querying for stargazers of a particular repository

Try this for different GitHub users

```
from github import Github # pip install pygithub
ACCESS_TOKEN = "ghp_o1WmL71lXeZRuhNi7SpqLhT1Lw3L5z0ewfyd"
USER = 'ptwobrussell'
REPO = "Mining-the-Social-Web"
#REPO = '&#39;Mining-the-Social-Web-2nd-Edition&#39;'
client = Github(ACCESS_TOKEN, per_page=100)
user = client.get_user(USER)
repo = user.get_repo(REPO)
stargazers = [ s for s in repo.get_stargazers() ]
print("Number of stargazers", len(stargazers))
```

Number of stargazers 1203

## ▼ Create a network graph using the networkx package

Install the networkx package

```
!pip install networkx
```

Requirement already satisfied: networkx in /usr/local/lib/python3.7/dist-packages (2.6.1)

```
import networkx as nx # pip install networkx
# Create a directed graph
g = nx.DiGraph()
# Add an edge to the directed graph from X to Y
g.add_edge('X', 'Y')
# Print some statistics about the graph
print(nx.info(g))
```

DiGraph with 2 nodes and 1 edges

```
# Get the nodes and edges from the graph
print("Nodes:", g.nodes())
print("Edges:", g.edges())
print()
# Get node properties
print("X props:", g.nodes['X'])
print("Y props:", g.nodes['Y'])
print()
# Get edge properties
print("X-Y props:", g.edges['X-Y'])
```

```
print(X=>Y props, g[X][Y])
print()
```

```
Nodes: ['X', 'Y']
Edges: [('X', 'Y')]
```

```
X props: {}
Y props: {}
```

```
X=>Y props: {}
```

```
# Update a node property
g.nodes['X'].update({'prop1' : 'value1'})
print("X props:", g.nodes['X'])
print()
# Update an edge property
g['X']['Y'].update({'label' : 'label1'})
print("X=>Y props:", g['X']['Y'])
```

```
X props: {'prop1': 'value1'}
```

```
X=>Y props: {'label': 'label1'}
```

## ▼ Constructing an ego graph of a repository and its stargazers

```
g = nx.DiGraph()
g.add_node(repo.name + '(repo)', type='repo', lang=repo.language, owner=user.login)

for sg in stargazers:
    g.add_node(sg.login + '(user)', type='user')
    g.add_edge(sg.login + '(user)', repo.name + '(repo)', type='gazes')
```

## ▼ Perform handy graph operations

```
print(nx.info(g))
print(g.nodes['Mining-the-Social-Web(repo)'])
print(g.nodes['ptwobrussell(user)'])
print(g['ptwobrussell(user)']['Mining-the-Social-Web(repo)'])
print(g['ptwobrussell(user)'])
print(g['Mining-the-Social-Web(repo)'])
print(g.in_edges(['ptwobrussell(user)']))
print(g.out_edges(['ptwobrussell(user)']))
print(g.in_edges(['Mining-the-Social-Web(repo)']))
print(g.out_edges(['Mining-the-Social-Web(repo)']))
```

```
DiGraph with 1204 nodes and 1203 edges
```

```
{'type': 'repo', 'lang': 'JavaScript', 'owner': 'ptwobrussell'}  
{'type': 'user'}  
{'type': 'gazes'}  
{'Mining-the-Social-Web(repo)': {'type': 'gazes'}}  
{}  
[]  
[('ptwobrussell(user)', 'Mining-the-Social-Web(repo)')]  
[('rdempsey(user)', 'Mining-the-Social-Web(repo)'), ('prb(user)', 'Mining-the-Social-Web(repo)')]  
[]
```

✓ 0s completed at 12:37 PM

