Data Representation
Lab 5.03 using packages

Lecturer: Andrew Beatty

In this lab we are going to use the package PyGitHub to interact with GitHub

It is easier than making all our own requests;

Install using

Pip install PyGithub

Documentation on this package:

- https://pygithub.readthedocs.io/en/latest/introduction.html

For examples

- https://pygithub.readthedocs.io/en/latest/examples.html

Full reference

- https://pygithub.readthedocs.io/en/latest/reference.html

Use your own github account make sure you do not put the token in any code you push.

I suggest that you make a special repository for this lab to interact with, put a text file in the repository called **test.txt**

1. install pyGithub

```
pip install PyGithub
```

2. Write a python script called lab05.03-githubbymodule.py

3. Test that your pyGithub works

```
from github import Github
from conf import auth

token = auth['token']
# use your own key
g = Github(token)

for repo in g.get_user().get_repos():
    print(repo.name)
```

4. Modify the program to get the clone url of a repository on your account m(you could make a private one just for this if you wish). Put a file in the repository called test.txt

```
g = Github(token)


repo = g.get_repo("yourccount/yourrepo")
print(repo.clone_url)
```

5. Get the downloadurl of the file in this repository called test.txt (make sure that there is a file called test.txt in there

```
fileInfo = repo.get_contents("test.txt")
urlOfFile = fileInfo.download_url
print (urlOfFile)
```

6. I would comment out the print statements once you are happy the program is working

7. Use the download URL to make a http request to the file can output the contents of the file (TEXT contents).

```
response = requests.get(urlOfFile)
contentOfFile = response.text
print (contentOfFile)
```

8. Append the text more stuff (with a newline character) to the contents of the file.

```
newContents = contentOfFile + " more stuff \n"

print (newContents)
```

9. Update the contents of the file on git up by using the function

update_file(*path*, *message*, *content*, *sha*, *branch=NotSet*, *committer=NotSet*, *author=NotSet*)

```
gitHubResponse=repo.update_file(fileInfo.path,"updated by prog",
newContents,fileInfo.sha)

print (gitHubResponse)
```

10. Look at the file on github and confirm that the text was added
11. Take the key out of the main code and put it in a file called config.py

```
config = {
    "githubkey" : "github_pat_XXXXXXXXXXXXXXXXXXXXXXD"
}
```

12. Add the name config.py to your .gitignore file, so that it is not pushed up to your repository

```
# my configuration files
config.py
```

13. Update your program so that it does not have your key in it.

```
from config import config as cfg
apikey = cfg["githubkey"]
```