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

1. Write a python script called lab05.03-githubbymodule.py
2. Test that your pyGithub works

from github import Github

from config import config as cfg

apikey = cfg["githubkey"]

# use your own key

g = Github(apikey)

for repo in g.get\_user().get\_repos():

    print(repo.name)

1. 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(apikey)

repo = g.get\_repo("yourccount/yourrepo")

print(repo.clone\_url)

1. 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)

1. I would comment out the print statements once you are happy the program is working
2. 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)

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

newContents = contentOfFile + " more stuff \n"

print (newContents)

1. 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)

1. Look at the file on github and confirm that the text was added

### Just make sure you do not push the key to github

1. put the key in a file called config.py

config = {

    "githubkey" : "github\_pat\_XXXXXXXXXXXXXXXXXXXXXXXD"

}

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

# my configuration files

config.py

1. Your program does not have your key in it.

from config import config as cfg

apikey = cfg["githubkey"]