09/04/24

Today’s work

Today at 09.00 we started practicing python for strong basic knowledge.

As lalit bhaiya’s feedback I had to start git and github for including in summary by today onwards.

Git and github

What is git?

It is a distributed version control system which exist locally on your system.

(track the history of files)

What is github?

Github is a company that lets you host your file using git.

First I opened gitbash and type ls command to show all the file in tanujg

Ls -a to show all hidden files.

$ git config --global user.name

Tanuj Gupta

$ git config --global user.email

tanujg722@gmail.com

Now let’s understand how to track a file

Made a folder name GITDEMO in vs code.then open gitbash and run command git init to track GITDEMO.

Tanujg@DESKTOP-3LSQRCO MINGW64 ~/OneDrive/Desktop/Gitdemo (main)

$ git init

Reinitialized existing Git repository in C:/Users/Tanujg/OneDrive/Desktop/Gitdemo/.git/

How Git works ?

Gitdemo represent working directory.

.git represent repository. Repository have 2 areas 1 Stagging area and 2 commit history .

First we made a file named hello.txt and save it.(hello everyone !!) msg in the file.

And check git status Tanujg@DESKTOP-3LSQRCO MINGW64 ~/OneDrive/Desktop/Gitdemo (main)

$ git status

We push are hello.txt file in staging area by using command git add hello.txt

Tanujg@DESKTOP-3LSQRCO MINGW64 ~/OneDrive/Desktop/Gitdemo (main)

$ git add helllo.txt

After that we commit it. by using command git commit -m “first commit”

Tanujg@DESKTOP-3LSQRCO MINGW64 ~/OneDrive/Desktop/Gitdemo (main)

$ git commit -m "first commit"

Right now I confused in all of my repository So that’s why I delete all repository form my github account.

And made a new repository named 09-04-2024

Clone- cloning a repository on our local machine.

$ git clone <https://github.com/tanujg722/09-04-2024.git>

Change directory $ cd 09-04-2024

To check all file ls

README.md

To check all file and hidden file aswell.

$ ls -a

./ ../ .git/ README.md

Now I start python

Errors and exeption exercise

Handle the exception thrown by the code below by using try and except blocks.

try:

    for i in ['a','b','c']:

        print(i\*\*2)

except:

    print("Type error! Watch out!")

output

Type error! Watch out!

Problem2

try:

    x = 5

    y = 0

    z=x/y

except:

    print("Error!!")

finally:

    print("All done")

output: Error!!

All done

Now check git status after making two problem solving file.

$ git status

Untracked files:

(use "git add <file>..." to include in what will be committed)

error and exception.py

problem2.py

after we add both files $ git add .

Now time commit file.

$ git commit -m "commiting e&e and problem2"

After all we need to use push command for uploading local repo content to remote repo

$ git push origin main

And my file error and exception.py and problem2.py successfully uploaded on my git hub account.

It takes time but done it successfully around 1:30.

Around 2:30 started

Unit testing lib.

Made 2 file named cap.py and test\_cap.py and saved it.

In cap.py

def cap\_text(text):

    '''

    Input a string

    Output the capitalized string

    '''

    return text.capitalize()

in test\_cap.py

import unittest

import cap

class TestCap(unittest.TestCase):

    def test\_one\_word(self):

        text = "python"

        result = cap.cap\_text(text)

        self.assertEqual(result,"Python")

    def test\_multiple\_words(self):

        text = 'monty python'

        result = cap.cap\_text(text)

        self.assertEqual(result,'Monty Python')

if \_\_name\_\_== '\_\_main\_\_' :

    unittest.main()

after that change the directory cd 09-04-2024

then add both file git add .

commit it git commit -m "commit cap.py and test\_cap.py"

and the final step push local repository content to remote repository.

git push origin main

after that cap.py and test\_cap.py was successfully uploaded on my git hub account.

Now decorators

Let’s discuss a more advanced Python topic: Decorator, it allow us to decorate a function.

Lets suppose we want to addd some new capabilities to the function:  
for this you have only two option

1 as we all know add extra code to your old code

2 create a new code and old code and new code.

But what if you then want to delete or remove that extra “functionality”.

For all this python has decorators that allow you to tack on extra functionality to an already existing function.

# 09-04-2024

#def simple\_function():

    # Want to do more stuff!

    # Do simple stuff return something

# @some\_decorator

#def simple\_func():

    # Do simple stuff

    #return somthing

def greet(fx):

    def mfx():

        print("Good Morning")

        fx()

        print("Thanks for using this function")

    return mfx

@greet

def hello():

    print("Hello World")

hello()

output: Good Morning

Hello World

Thanks for using this function

After this decorator code I started it to upload on git hub account

PS C:\Users\Tanujg\OneDrive\Desktop\Gitdemo> cd 09-04-2024

PS C:\Users\Tanujg\OneDrive\Desktop\Gitdemo\09-04-2024> git add .

PS C:\Users\Tanujg\OneDrive\Desktop\Gitdemo\09-04-2024> git commit -m "deco"

PS C:\Users\Tanujg\OneDrive\Desktop\Gitdemo\09-04-2024> git push origin main

Around 6 in the evening Finally I successfully upload my file in 4 steps

Total sitting throughout the day -2

Total study time throughout the day = approx. 8hr