11/04/24

Today’s work

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

Opening and reading files and folder

Shutil and OS Modules.

We already know how to open an individual file with Python, but we still don’t know how to do a few things:

What if we have to open every file in a directory?

What if we want to actually moves files around on our computer?

Python’s os module and shutil allow us to easily navigate files and directories on the computer and then perform actions on them, such as moving them or deleting them.

OS MODULE in Python provides a way to interact with the operating system. It offers a wide range of functions for working with files and directories, managing processes, and more.

import os

os.getcwd() # current working directory

output: c:/Users/Tanujg/OneDrive/Desktop/OS module/main.py"

f = open("Practice.txt","w+") # here we open a textfile

f.write("This is a test string") # text writing in the files.

f.close() # here we close the file.

print(os.listdir()) # to know files in my cd

output: c:/Users/Tanujg/OneDrive/Desktop/OS module/main.py"

['main.py', 'Practice.txt']

Shutill module provides functions for copying, moving, and removing files and directories.

import shutil

shutil.move("Practice.txt","C:\\Users\\Tanujg\\OneDrive\\Desktop\\shutil") # first file name which we want to move then the location where we want to move file with \\ backslahes

Python datetime module in Python provides classes for manipulating dates and times. It allows you to create, manipulate, format, and convert dates and times, as well as perform arithmetic operations on them.

import datetime

mytime = datetime.time(2,20,56,3)#first hour then minutes after that seconds then microseconds

print(mytime.minute)

print(mytime.hour)

print(mytime)

print(mytime.second)

print(mytime.microsecond)

output: 20

2

02:20:56.000003

56

3

#date

today = datetime.date.today()

print(today)

print(today.year)

print(today.month)

print(today.day)

print(today.ctime())

output:

2024-04-12

2024

4

12

Fri Apr 12 00:00:00 2024

from datetime import date

date1= date(2023,3,11)

date2= date(2024,11,3)

result = date1 - date2

print(result) # here you got the days between date1 and date2.

output: -603 days

from datetime import datetime

date1= datetime(2023,3,11,23,56)

date2= datetime(2024,11,3,12,55)

result = date1 - date2

print(result) # here you got the days between date1 and date2 and time aswell.

Output: -603 days, 11:01:00

Python math and random modules:

Math module in Python provides mathematical functions and constants.

import math

value = 4 # it ia  value

print(math.floor(4.34)) # Here it is converted in integers

print(math.ceil(4.35)) # Here it is converted in rounds off

#numpy

print(math.sqrt(16))

print(math.pow(2, 3))

print(math.sin(math.pi/2))

print(math.cos(math.pi))

print(math.log(10))

print(math.exp(2))

print(math.pi)

print(math.e)

output: 4

5

4.0

8.0

1.0

-1.0

2.302585092994046

7.38905609893065

3.141592653589793

2.718281828459045

Random modules provides functions for generating random numbers.

import random

# random modules

print(random.randint(1, 100))  # Random integer between 1 and 100.

random.seed(12)

print(random.randint(1, 100)) # always got the same output due to seed funvtion

items = [1, 2, 3, 4, 5]

random.shuffle(items) # Random shuffle of a sequence

print(items)

print(random.choice([1, 2, 3, 4, 5])) # Random choice from a sequence

print(random.uniform(1, 10)) # Random float from a uniform distribution

output:

48

61

[4, 2, 1, 5, 3]

1

9.96960006932013

Around 1:30 I complete these three module.

Around 2:30 I started Python debugger

When we trying to figure out what errors there are in your code, we mostly used print to trackdown the error. but python have a built-in debugger tool that allows you to interactively explore variables within mid-operation of your Python code!

#x = [1,2,3]

#y = 2

#z = 3

#result = y + z

#result2 = x + y # TypeError: can only concatenate list (not "int") to list

Output: Traceback (most recent call last):

File "c:\Users\Tanujg\OneDrive\Desktop\OS module\pythondebug.py", line 14, in <module>

result2 = x + y

~~^~~

TypeError: can only concatenate list (not "int") to list

import pdb # Python debugger

x = [1,2,3]

y = 2

z = 3

result = y + z

pdb.set\_trace()

result2 = x + y

output: > c:\users\tanujg\onedrive\desktop\os module\pythondebug.py(15)<module>()

-> result2 = x + y

(Pdb) result

5

**(Pdb) q # q is used to terminate the debugger.**

Now python regular expressions part One

As we all know we can check substrings within a larger string with “in” operator:

Eg print("dog"in " my dog is great")

Output: True

There have some limitation

In regular Expression (regex) allow us to search for general patterns in text data

For eg a simple email format can be:

[user@gmail.com](mailto:user@gmail.com)

We know in this case we are looking for a pattern “text” + “a” + “text” + “.com”

The re lib allows us to create specialized pattern strings and then search for matches with text.

Phone Number

(666)-666-6666

Regex Pattern

R”(\d\d\d)-\d\d\d-\d\d\d\d”

text= " my dog is great"

import re

pattern = "dog"

print(re.search(pattern,text))

output: re.Match object; span=(4, 7), match='dog'>

if the pattern doesn’t match in the text

pattern ="Tanuj"

print(re.search(pattern,text))

output: None

If we have multiple match in the text but python only considered the first one match only and that index number aswell.

text2 = "My name is tanuj gupta and My age is 30"

pattern = 'My'

print(re.search(pattern,text2))

output: re.Match object; span=(0, 2), match='My'>

Around 6 I uploaded all today’s files in github and windup all.

Total sitting throughout the day =2

Total study time through the day = approx. 8 hr