Modules in Python

Math Module

```
import math

# Round a number upward to its nearest integer
print(math.ceil(1.4))
print(math.ceil(5.3))
print(math.ceil(-5.3))
```

```
2
6
-5
```

```
# Total number of possible combinations
n = 7
k = 5

print(math.comb(n, k))
```

```
21
```

```
# Return factorial of a number
print(math.factorial(9))
print(math.factorial(6))
print(math.factorial(12))
```

```
362880
720
479001600

#Round numbers down to the nearest integer
print(math.floor(1.4))
print(math.floor(5.3))
print(math.floor(-5.3))

1
5
-6

#Returns the sum of all items in any iterable (tuples, arrays, lists, etc.)
print(math.fsum([1, 2, 3, 4, 5]))
print(math.fsum([100, 400, 340, 500]))
print(math.fsum([1.7, 0.3, 1.5, 4.5]))

15.0
1340.0
8.0

#Return the value of x raised to the power of y
print(math.pow(9, 3))
print(math.pow(7, 4))
print(math.pow(2, 8))

729.0
2401.0
256.0

pie = math.pi #Return the value of pie
print(pie)

3.141592653589793
```

Date & Time Module

```
import datetime
import time

now = datetime.datetime.now()
print(now)
print(now.year)
print(now.month)
print(now.day)

2023-12-07 16:54:16.181697
2023
```

```
12
7

print(time.time())

1701948256.6933472
```

- the sleep() function inside time module is used to delay execution of current time thread by the given number of seconds.
- ctime() the current time in human readable format

```
time.sleep(10)
print(time.ctime())
```

Calendar:

We can even print a calendar for a particular month. This can be done using the calendar module.

```
import calendar
print(calendar.month(2023, 11))

months = []
for i in range(1,13):
    months.append(i)

for i in months:
    print(calendar.month(2023,i))
```

Random Module

Python has a built-in module that you can use to make random numbers.

```
import random

print(random.random()) #Return random number between 0.0 and 1.0:
print(random.randint(1, 10)) #Return a number between 3 and 9
print(random.choice(months)) #Return a random element from a list
```

File Handling in Python

- Python too supports file handling and allows users to handle files i.e., to read and write files, along with many other file handling options, to operate on files.
- Python has several functions for creating, reading, updating, and deleting files.

- Python also supports different modes for opening files, including:
  - "r": Read (default mode). Opens the file for reading.
  - "w": Write. Opens the file for writing, truncating the file if it already exists.
  - "a": Append. Opens the file for writing, but appends new content to the end of the file.
  - "b": Binary mode. For working with binary files, like images or videos.
  - "x": Exclusive creation. Opens the file for writing, but only if the file doesn't already exist.

Working in Read mode

```
file = open("myfile.txt", "r")
print (file.read())

file = open("myfile.txt", "r")
print (file.read(12))
```

Working in Write mode

```
file = open('myfile.txt','w')
file.write("Email - zahidshaikh10101@gmail.com")
file.close()

file = open("myfile.txt", "r")
print (file.read())
```

Working in Append mode

```
file = open('myfile.txt', 'a')
file.write("Name - Zahid Salim Shaikh")
file.write("Age - 23")
file.write("Qualification - BE IT")
file.close()

file = open("myfile.txt", "r")
print (file.read())
```

Working in Create mode

```
f = open("my_file3.txt", "x")
```

Learning Progress:

Modules:

- I have comprehensively explored the concept of modules in Python, understanding their role in organizing code and providing reusable functionalities.
- I have successfully mastered importing and using various built-in modules like math, datetime, calendar, time, and random.

- I can effectively utilize functions within these modules to perform mathematical calculations, manipulate dates and times, generate random numbers, and handle files efficiently.
- I have gained a solid understanding of module search paths and package structure.

File Handling:

- I have thoroughly grasped the concept of file handling in Python, including opening, closing, reading, writing, and manipulating files.
- I have mastered various file operations like open(), close(), read(), write(), seek(), and tell().
- I can effectively write programs that interact with files, managing data storage, reading information from files, and writing outputs to desired locations.

Next Steps:

Program Development:

I will build a small program to demonstrate my understanding of utilizing Python modules:

- Date and Time Manipulation: This program will current datetime and generate datetime for next 5 days. It will showcase my ability to integrate various modules for a specific task.
- Random Dice Roll Game: This program will leverage the random module to generate random numbers and create a simple game with interactive elements. It will demonstrate my understanding of using modules for generating random data and creating engaging programs.

```
from datetime import datetime, timedelta

current_time = datetime.now()
future_time = current_time + timedelta(days=5)


formatted_current_time = current_time.strftime("%Y-%m-%d %H:%M:%S")
formatted_future_time = future_time.strftime("%Y-%m-%d %H:%M:%S")

print(f"Current time: {formatted_current_time}")
print(f"Time in 5 days: {formatted_future_time}")

import random

num_sides = 6
roll = random.randint(1, num_sides)

print(f"You rolled a {roll}!")
```