

20/8/23
Task 3: Importing and creating python modules and package in python program

Aim: To implement and demonstrate the process of importing built-in, thereby promoting code reusability, modularity, and maintainability.

3.1

- 1) Perform common math and random operations.
- 2) Work with the operating system (Create/ change directories, list contents) and read the python version.
- 3) Compute basic statistics (mean, median, mode, standard deviation).

Algorithm

1) Import required modules: math, random, os, sys, statistics, path lib.

2) Math & Random:

Compute sqrt(5), radians(30), a random float in [0.0, 1.0], a random integer in [2, 6] (inclusive), ceil(0.3), floor(2.3), factorial(5), gcd(5, 15), log(10), pow(3, 5), log base 3 of 2, log10(a) for a = 100, and check NaN/Infinity.

3) OS & sys:

- Create C:\Python\lab if not present and print the current working directory
- Create C:\Python\Lab\SL4 if not present and change the current working directory to it.
- List all files/directories in the new current directory
- Print python interpreter version

4) Statistics:

- On list [8, 6, 8, 10] and [2, 5, 3, 2, 3, 3, 4, 4, 2, 5, 6] compute mean, median, mode std dev,
- Print neatly formatted results.

```
import math
import random
import os
import sys
import statistics as stats
from pathlib import Path

print("In -- MATH & RANDOM--")
print("sqrt(5) = " + str(math.sqrt(5)))
print("radians(30) = " + str(math.radians(30)))
print("random() in [0,1] = " + str(random.random()))
print("randint(2,6) = " + str(random.randint(2,6)) + "# inclusive")
print("pi = " + str(math.pi))
print("ceil(2.3) = " + str(math.ceil(2.3)))
print("floor(2.3) = " + str(math.floor(2.3)))
print("factorial(5) = " + str(math.factorial(5)))
print("gcd(5,15) = " + str(math.gcd(5,15)))
print("abs(-10) = " + str(abs(-10)))
print("pow(3,5) = " + str(pow(3,5)))
print("log base 3 of 2 ≈ " + str(math.log(2,3)))    a_val = 10
print("log10(a_val) = " + str(math.log10(a_val)))
inf_val = float('inf')
nan_val = float('nan')

print(f"Isinf(0) = {math.isinf(0)}, isnan(NaN) = {math.isnan(NaN)}")
```

print("In -- os & sys--")

path_pythonlab = Path(r"C:\PythonLab")
path_pythonlab.mkdir(parents=True, exist_ok=True)

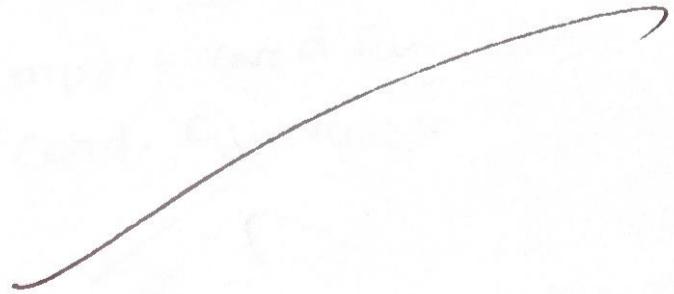
print(f"Created/ensured: {path_pythonlab}")

print("Current working directory: " + os.getcwd())

target_dir = Path(r"\\PythonLab\S2L4").
target_dir.mkdir(parents=True, exist_ok=True)

os.chdir(target_dir)

```
print ("f" "changed into: %target-dir%")  
print ("Directory contents", os.listdir())  
print ("Python version", sys.version)  
print ("... statistics...")  
data1 = [5, 6, 8, 10]  
data2 = [2, 5, 3, 2, 8, 3, 9, 4, 2, 5, 10]  
print ("mean (%data1%) =", stats.mean(data1))  
print ("median (%data1%) =", stats.median(data1))  
print ("mode (%data2%) =", stats.mode(data2))  
print ("stddev (%data2%) =", stats.stdev(data2))
```



2. Establish a family transmission line to exchange global signals required to implement the system.

Sample output:

$$\text{sqrt}(5) = 2 \cdot 23606797749978921088 \approx 2.236067977$$

$$\text{radians (B0)} = 0.5235987755982988$$

random() in [0,1] = 0.374448877564831 + 1.70158241923891e-10

$$\text{radiant}(2(6)) = 6$$

$\pi = 3.141592653589793$

$$\text{Cap H}(2,3) = 3$$

$$\log_2(2-3) = 2$$

$$\text{factorial}(5) = 120$$

g "Cd, (51.15) = 5

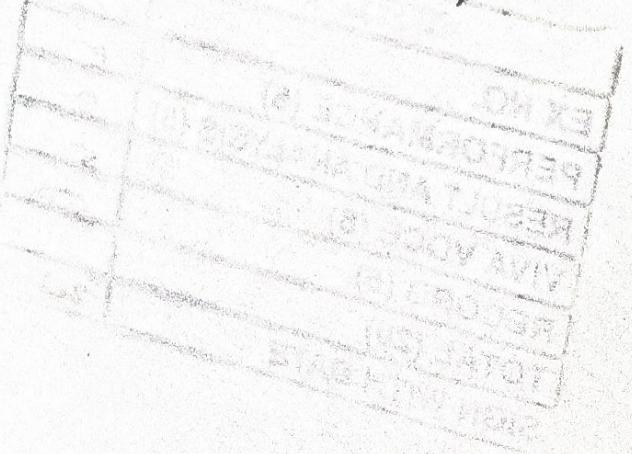
$$ab \leq (-\infty) = 0$$

$$\text{pow}(3, 5) = 243$$

$$\log_{\text{base } e} 30 \approx 3.40182941574$$

$$\log_{10}(100) = 2.0$$

`isinf(0.0) = true, isnan(Nan) = true`



Pluto was once a planet.

3.2 Create a Python package name Card pack containing a module card fun that imports the random module. Assign a range of cards, call a function from the module, and display a random sample of cards.

Algorithm:

Step1: Start

Step2: To create a package Card pack

Step3: To create a module Card fun and import random function.

Step4: Assign a cards range

Step5: Call a module function

Step6: Display the random sample cards

Step7: Stop

Program:

Card fun

```
import random
```

```
def fun():
```

```
    cards = []
```

```
    for i in range(1, 52):
```

```
        cards.append(i)
```

```
shuffled_cards = random.sample(cards, k=52)
```

```
print("".join(shuffled_cards), "\n")
```

My mod.py

```
import card fun
```

```
card.fun.fun()
```

(006828) 20 (Whitman's estimated) 10
(006828) 20 (Whitman's estimated) 10
100% cover 20% + 100% cover 100% 69%
Cover 20% Estimate 2 - 100%

[01 : 8.00 : 2]

($\partial_1 \mathcal{E}, \mathcal{E}_{1,0}, P, \mathcal{E}_{1,2}, \mathcal{E}_{1,3}, \dots$)

Output:

(E5129) 13, 22, 20, 44, 188, 151, 16, 2, 34, 49, 14, 150, 32, 80
15, 35, 17, 18, 33, 39, 36, 42, 102, 61, 16, 19, 48,
29, 2, 27, 11, 131, 46, 28, 21, 32, 8, 23, 39, 23, 26,
(10, 143, 47, 3, 44, 52, 1, 16, 5, 9)

8.3 you are tasked with developing a modular calculator application in python. The calculator should support basic arithmetic operations: addition, subtraction, multiplications, and division. Each operation should be implemented in a separate module. Additionally, you should create a main program to handle user input, call the appropriate module, and display the results.

Algorithm:

- 1) Define functions for addition, subtraction, multiplications, and division.
- 2) Handle division by zero by raising an error if the divisor is zero.
- 3) Import the module (mymath) containing these functions
- 4) Initialize two numbers (a=10, b=5)
- 5) Call each function using mymath.function name(a,b)
- 6) Print the results of all operations.

Program: (my) math

```
def add(a,b):  
    return a+b  
  
def subtract(a,b):  
    return a-b  
  
def multiply(a,b):  
    return a*b  
  
def divide(a,b):  
    if b==0:  
        raise ValueError("cannot divide by zero")  
    return a/b  
  
import mymath  
a=10, b=5  
  
print("Addition", mymath.add(a,b))  
print("Subtraction", mymath.subtract(a,b))  
print("Multiplication", mymath.multiply(a,b))  
print("Division", mymath.divide(a,b))
```

8.4 You are working on a python project that requires you to perform various mathematical operations and geometric area calculations. To organize your code better, you decide to create a package named mypackage which includes sub packages pack1 and pack2 with two modules: math functions and area functions demonstrate the use of the functions by performing a few calculation and printing the results.

Algorithm:

- 1) Create mathfunctions.py module.
- 2) Create areafunctions.py module.
- 3) Create main.py.
- 4) Print the output as expected.

Program:

- 1) Create the mathfunctions.py module

```
def add(a,b):  
    return a+b  
  
def subtract(a,b):  
    return a-b  
  
def multiply(a,b):  
    return a*b  
  
def divide(a,b):  
    if b==0:  
        return "Error: division by zero!"  
    return a/b
```

- 2) Create the areafunctions.py module

```
import math  
  
def circle_area(radius):  
    return math.pi * radius * radius  
  
def rectangle_area(length, width)  
    return length * width  
  
def triangle_area(base, height)  
    return 0.5 * base * height
```

Program has advantages of portability,
reusability and scope of application
which is simple to understand and
confirms to the basic concepts and
principles of mathematics. It also
enables us to practice and learn
mathematics at the beginning.

output:

Addition : 15

Subtraction : 5

Multiplication : 50

Division : 2.0

Circle Area (radius=7) : 153. 9380400258825

Rectangle area(5x10) : 50

Triangle area (base=6, height=8) : 24.0

```

import math functions
import area functions
#using math functions

Print("Addition:", math functions.add(10,5))
Print("Subtraction:", math functions.subtract(10,5))
Print("Multiplication:", math functions.multiply(10,5))
Print("Division:", math functions.divide(10,5))

#using area functions

Print("Circle Area (radius=7):", area functions.circle_area(7))
Print("Rectangle Area (5x10):", area functions.rectangle_area(5,10))
Print("Triangle Area (base=6, height=8):", area functions.triangle_
area (6,8))

```

| VEL TECH | |
|-------------------------|------------|
| EX NO. | 3 |
| PERFORMANCE (5) | 5 |
| RESULT AND ANALYSIS (5) | 5 |
| VIVA VOCE (5) | 5 |
| RECORD (5) | 5 |
| TOTAL (20) | 20 |
| SIGNATURE DATE | 11/10/2022 |



Result: Thus the program for importing python modules and packages was successfully executed and the output was verified.