

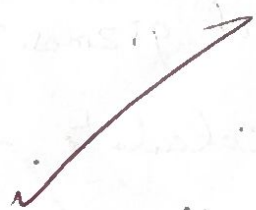
Importing Python modules and packages in Python Programming

3.1 Math module:

AIM: Write a program to perform mathematical calculation using math module

Algorithm:

- Step 1: open python script file.
- Step 2: import math module
- Step 3: `math.log` is used to find log base 10
- Step 4: `**` used to calculate power
- Step 5: `math.floor()`, `math.ceil()` used to calculate floor and ceil value
- Step 6: `abs()` method used to display absolute value
- Step 7: `factorial()` method to display the factorial value.



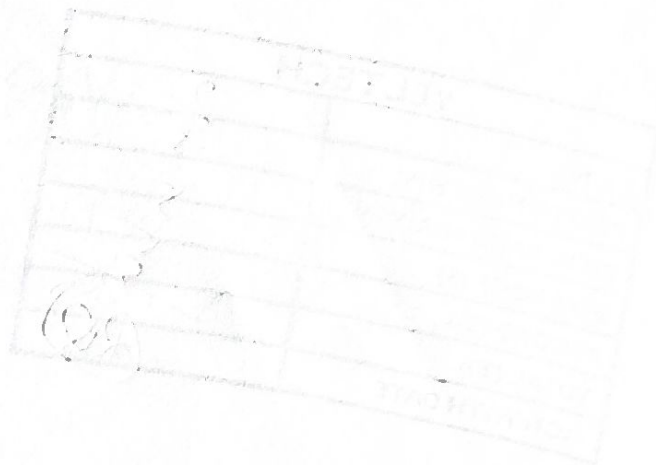
Result: Thus, the program to perform mathematical calculation using math module has been verified successfully

Source Code:

```
import math
Print ("Logarithm", math.log10(13))
Print ("Power", math.pow(10, 2))
Print ("Floor", math.floor(10.25201))
Print ("Ceil", math.ceil(10.25201))
Print ("absolute", abs(-10.024))
Print ("Factorial", math.factorial(5))
```

Output:

Logarithm 1.1139433523068367
Power 100.0
Floor 10
Ceil 11
absolute 10.024
Factorial 120



3.2

AIM: An online retailer sells two products: widgets and gizmos. Each widget weighs 75 grams. Each gizmo weighs 112 grams. write a program that reads the number of widgets and the number of gizmos from the user. Then your program should compute and display the total weight of the parts using the concept of packages and modules.

Algorithm:

1. start
2. create a package folder named shop with:
 - A module file `weight_calc.py` containing a function to calculate total weight.
 - `__init__.py` file to make it a package
3. In `weight_calc.py`.
 - Define constants: `WIDGET_WEIGHT = 75` grams and `GIZMO_WEIGHT = 112` grams.
 - create a function `calculate_total_weights(widgets, gizmos)` that:
 - multiplies the no. of widgets by `WIDGET_WEIGHT`
4. In the main program (`main.py`):
 - Import the `weight_calc` module from the shop package
 - prompt the user to enter the no. of widgets
5. End.

Program:

Here's the structure

shop /

-init-.py

weight-calc.py

main.py

shop/weight-calc.py

#weight-calc.py

def calculate_total_weight(widgets, gizmos):

WIDGET_WEIGHT = 75 #grams

GIZMO_WEIGHT = 112 #grams

total_weight = (widgets * WIDGET_WEIGHT) + (gizmos * GIZMO_WEIGHT)

return total_weight

2. shop/init.py

#init.py

#this makes 'shop' a package

3. main.py

#main.py

from shop import weight_calc

def main():

widgets = int(input("Enter the no. of widgets:"))

gizmos = int(input("Enter the no. of gizmos:"))

total_weight = weight_calc.calculate_total_weight(widgets, gizmos)

print(f"The total weight of the parts is {total_weight} grams.")

if __name__ == "__main__":

main()

AIM: write a program to display os and sys details using os and sys module

Algorithm:

- Step 1: open python script file
- Step 2: import os and sys module
- Step 3: os.name to display the name of the os.
- Step 4: os.getcwd() to display current working directory
- Step 5: os.listdir() to display the directory list details
- Step 6: sys.platform used to display linux platform
- Step 7: Platform.system(), platform.release() method to display the platform details and platform release details

Output:

os name nt

```
E 'io.py', 'code9.py', 'DLLS', 'doc', 'include', 'le3.2.py', 'le4.2.py', 'le.py',
'le.py', 'le6.py', 'Lib', 'libs', 'license.txt', 'News.txt', 'pyt code15.py',
'python.exe', 'python3.dll', 'python312.dll', 'pythonw.exe', 'scripts', 'tcl',
'vcruntime140.dll', 'vcruntime140_1.dll']
```

le6a

platform name win32


Platform name * system() windows

Platform, release() 11

Result: Thus, the program to display os and sys details using os and sys module.

Source code:

```
import os
import sys
import platform
import sysconfig
print("os name", os.name)
print(os.getcwd())
print(os.getlogin())
print("platform name", sys.platform)
print("platform.system()", platform.system())
print("platform.release()", platform.release())
```



3.4

Random WISH

AIM: write a program to display unique number wishes

Algorithm :

- Step 1: open python script file.
- Step 2: import random module
- Step 3: list the wishes in fortunes.
- Step 4: random.choice() using to display the random wishes

Output :

— A wise man once said, Everything in its own time and place.

RESULT : Thus, the program to display unique number wishes verified successfully.

Source code:

```
import random
```

```
fortunes = ["Good things come to those who wait.",
```

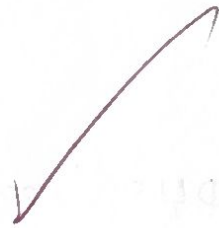
```
            "Patience is a virtue.",
```

```
            "The early bird gets the worm.",
```

```
            "A wise man once said, everything in its own  
            and place.",
```

```
            "Fortune Cookies nearly share fortunes."]
```

```
Print (random.choice(fortunes))
```



3.5

Mathematical and String operations

AIM:- To write a python program using packages and modules that performs mathematical operations (addition, multiplication, factorial) and string operations (reverse a string, count vowels) by organizing code into reusable components.

ALGORITHM:

Step 1: 1. Create the package structure

2. Create a folder named - my-package.

3. inside my-package, create the files:

- init.py
- math_utils.py
- string_utils.py

Step 2 - (implement) math_utils.py module

1. Define a function add(a,b) to return the sum of two numbers.

2. Define a function multiply(a,b) to return the product of two numbers

3. Define a function factorial(n) that:

- Returns 1 if n is 0 or 1.
- otherwise, returns $n * \text{factorial}(n-1)$ (recursive call)

Step 3 - Implement String_utils.py module

1. Define a function reverse_string(s) that returns the reversed string using slicing ($s[::-1]$).

2. Define a function Count_vowels(s) that:

- stores vowels in a string "aeiouAEIOU".
- Counts characters from the input and string that are present in vowels

Program

1. Folder structure:

my-package /

-init-.py

math-utils.py

string-utils.py

main.py

my-package/math-utils.py

math-utils.py

def add(a,b):

return a+b

def multiply(a,b):

return a*b

def factorial(n):

if n==0 or n==1:

return 1

return n*factorial(n-1)

my-package/string-utils.py

string-utils.py

def count_vowels(s):

vowels = 'aeiouAEIOU'

return sum(1 for char in s if char in vowels)

my-package/-init-.py

#-init-.py

from math-utils import add, multiply, factorial

from string-utils import reverse_string, count_vowels

(This makes it easy to access all functions directly from my-package.)

2. Main program(main.py)

```
#main.py
```

```
from my-package import add, multiply, factorial, reverse_string,  
Count_vowels
```

```
#math functions
```

```
print("Addition:", add(10,5))
```

```
print("Factorial of 5:", factorial(5))
```

```
print("Multiplication:", multiply(3,4))
```

```
#string functions
```

```
print("Reversed string:", reverse_string("Python"))
```

```
print("Number of vowels:", count_vowels("Hello world"))
```


Step 4 - initialize package with `_init_.py`

1. Import functions from `math_utils` and `string_utils` into `_init_.py` for Easy access.

Step 5 - Create `main.py` program

1. import functions from `my-package`.
2. call and display results of
 - Addition, multiplication, factorial from `math` module
 - Reverse string and vowel counting from `string` module.

Step 6: Execute the program

1. Run `python main.py` in the terminal.
2. Verify that outputs match expected results.

Sample output:

Addition : 15

multiplication : 12

Factorial of 5 : 120

Reversed string : nohtyp

Number of vowels : 3

VEL TECH	
EX No.	3
PERFORMANCE (5)	5
RESULT AND ANALYSIS	5
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	20
SIGN WITH DATE	

RESULT: Thus, the mathematical and string operations has been Executed successfully.