

28/10/25

PACKAGE - Collection of modules

MODULE - Collection of fuctions

FUCTIONS - Collection of statements

```
In [ ]: import math as m
```

```
In [7]: print(m.sqrt(100))
```

10.0

```
In [9]: print(m.sqrt(10))
```

3.1622776601683795

```
In [15]: round(12.4567890)
```

Out[15]: 12

```
In [17]: round(23.45678,3)
```

Out[17]: 23.457

```
In [19]: print(m.floor(23.45654))
```

23

```
In [21]: print(m.ceil(23.45654))
```

24

```
In [23]: print(m.floor(2.0))
```

2

```
In [25]: print(m.ceil(2.0))
```

2

```
In [27]: print(m.ceil(2.1))
```

3

```
In [29]: round(3)
```

Out[29]: 3

```
In [31]: round(12.4573)
```

Out[31]: 12

```
In [33]: round(12.4573,1)
```

```
Out[33]: 12.5
```

```
In [35]: round(12.4573,2)
```

```
Out[35]: 12.46
```

```
In [37]: round(12.4573,3)
```

```
Out[37]: 12.457
```

```
In [39]: round(12.4573,4)
```

```
Out[39]: 12.4573
```

```
In [41]: round(12.4573,5)
```

```
Out[41]: 12.4573
```

```
In [45]: print(m.pow(2,8))
```

```
256.0
```

```
In [47]: print(m.pow(2,20))
```

```
1048576.0
```

```
In [49]: print(m.pow(8,8))
```

```
16777216.0
```

```
In [51]: print(m.pi) #constant
```

```
3.141592653589793
```

```
In [53]: print(m.e) #constant
```

```
2.718281828459045
```

```
In [55]: from math import sqrt,ceil,floor,pow
```

```
In [59]: sqrt(25)
```

```
Out[59]: 5.0
```

```
In [61]: floor(12.09876)
```

```
Out[61]: 12
```

```
In [63]: ceil(12.09876)
```

```
Out[63]: 13
```

```
In [65]: pow(12,4)
```

```
Out[65]: 20736.0
```

```
In [67]: tax=12.5/100
         price=100.50
         print(tax*price)
```

```
12.5625
```

```
In [1]: a=10
         b=20
         a,b=b,a
         print(a)
         print(b)
```

```
20
```

```
10
```

```
In [3]: dic={"n":1,"a":2,"d":5}
         dic[4]="t"
         print(dic)
```

```
{'n': 1, 'a': 2, 'd': 5, 4: 't'}
```

```
In [15]: var=2
         print(var==2.0)
```

```
True
```

```
In [17]: a=4+0j
         print(type(a))
```

```
<class 'complex'>
```

```
In [19]: print(int(3.9))
```

```
3
```

```
In [21]: print('python'+".py")
```

```
python.py
```

```
In [23]: x = 50
         def fun1():
             x = 25
             print(x)

         fun1()
         print(x)
```

```
25
```

```
50
```

```
In [31]: str1 = ""Ault'Kelly""
         str1
```

```
Out[31]: "Ault'Kelly"
```

```
In [37]: str2 = 'Ault\''Kelly'  
str2
```

```
Out[37]: "Ault'Kelly"
```

```
In [39]: bool(-2)
```

```
Out[39]: True
```

```
In [41]: print(type(range(5)))
```

```
<class 'range'>
```

29/10/25 INPUT FUCTIONS

```
In [2]: x=input()  
x
```

```
Out[2]: '20'
```

```
In [4]: x=input()  
y=input()  
z=x+y  
print(z)
```

```
1020
```

```
In [6]: print(type(x))
```

```
<class 'str'>
```

```
In [8]: a=int(input())  
b=int(input())  
c=a+b  
print(c)
```

```
30
```

```
In [10]: print(type(a))
```

```
<class 'int'>
```

```
In [38]: x1=input("enter the text1:")  
y1=input("enter the text2:")  
z1=x1+y1  
print(z1)
```

```
anjaliarun
```

```
In [14]: x1=input("enter the number1:")  
y1=input("enter the number2:")  
z1=x1+y1  
print(z1)
```

```
2030
```

```
In [16]: a1=int(input("enter the number1:"))
        b1=int(input("enter the number2:"))
        c1=a1+b1
        print(c1)
```

20

```
In [18]: s=input("enter a string:")
        s
```

Out[18]: 'anjali'

```
In [20]: s[0]
```

Out[20]: 'a'

```
In [22]: s[:]
```

Out[22]: 'anjali'

```
In [24]: s[::-1]
```

Out[24]: 'ilajna'

```
In [26]: s[0:3]
```

Out[26]: 'anj'

```
In [28]: s1=input("enter a string:")[1] #forward indexing in input fuction
        s1
```

Out[28]: 'e'

```
In [30]: s1=input("enter a string:")[-1] #backward indexing in input fuction
        s1
```

Out[30]: 'o'

```
In [32]: s1=input("enter a string:")[1:4] #slicing in input fuction
        s1
```

Out[32]: 'ell'

```
In [34]: result=int(input("enter an expr:"))
        print(result)
```

```
-----
ValueError                                Traceback (most recent call last)
Cell In[34], line 1
----> 1 result=int(input("enter an expr:"))
      2 print(result)

ValueError: invalid literal for int() with base 10: '5+8-3'
```

```
In [36]: result=eval(input("enter an expr:")) #eval is used for expressions  
print(result)
```

10

```
In [ ]:
```