Modules in Python

A group of functions, variables and classes saved to a file, which is nothing but module.
Every Python file (.py) acts as a module.

Eg: durgamath.py
```
x=888
def add(a,b):
    print("The Sum:",a+b)


 def product(a,b):
    print("The Product:",a*b)
```

durgamath module contains **one variable and 2 functions.**
If we want to use members of module in our program then we should import that module.
import modulename
We can access members by using module name.
modulename.variable
modulename.function()
test.py:
```
import durgamath
print(durgamath.x)
durgamath.add(10,20)
durgamath.product(10,20)
```

```
Output
888
The Sum: 30
The Product: 200
```

Renaming a module at the time of import (module aliasing):
Eg:
import durgamath as m
here durgamath is original module name and m is alias name.
We can access members by using alias name m

test.py:
```
import durgamath as m
print(m.x)
m.add(10,20)
m.product(10,20)
```

from ... import:
We can import particular members of module by using from ... import .
The main advantage of this is we can access members directly without
using module
name.
Eg:
```
from durgamath import x,add
print(x)
add(10,20)
product(10,20)==> NameError: name 'product' is not defined
```
We can import all members of a module as follows
```
from durgamath import *
```
test.py:
```
from durgamath import *
print(x)
add(10,20)
product(10,20)
```

member aliasing:
```
from durgamath import x as y,add as sum
print(y)
sum(10,20)
```
Once we defined as alias name,we should use alias name only and we
should not use
original name
Eg:
```
from durgamath import x as y
print(x)==>NameError: name 'x' is not defined
```

Working with random module:
This module defines several functions to generate random numbers.
We can use these functions while developing games,in cryptography and to
generate

random numbers on fly for authentication.

1. random() function:

This function always generate some **float value** between 0 and 1 (not inclusive)

$0<x<1$

Eg:

```
from random import *
for i in range(10):
    print(random())
```

Output

0.4572685609302056
0.6584325233197768
0.15444034016553587
0.18351427005232201
0.1330257265904884
0.9291139798071045
0.6586741197891783
0.8901649834019002
0.25540891083913053
0.7290504335962871

2. randint() function:

To generate random integer beween two given numbers(inclusive)

Eg:

```
from random import *
for i in range(10):
print(randint(1,100)) # generate random int value between 1 and 100
```

Output

51
44
39
70
49
74
52

10
40
8

3. uniform():
 It returns random float values between 2 given numbers
Eg:
 from random import *
 for i in range(10):
     print(uniform(1,10))

 Output
 9.787695398230332
 6.81102218793548
 8.068672144377329
 8.567976357239834
 6.363511674803802
 2.176137584071641
 4.822867939432386
 6.0801725149678445
 7.508457735544763
 1.9982221862917555
random() ===>in between 0 and 1 (float Value)
randint(x,y) ==>in between x and y (integer value)
uniform(x,y) ==> in between x and y(float Value)

4. randrange([start],[stop],[step])
 returns a random number from range
 start<= x < stop
 start argument is optional and default value is 0
 step argument is optional and default value is 1
randrange(10)-->generates a number from 0 to 9
randrange(1,11)-->generates a number from 1 to 10
randrange(1,11,2)-->generates a number from 1,3,5,7,9
Eg 1:
 from random import *
 for i in range(10):
     print(randrange(10))

 Output

9
4
0
2
9
4
8
9
5
9

Eg 2:
```
from random import *
for i in range(10):
print(randrange(1,11))
```

Output
2
2
8
10
3
5
9
1
6
3

Eg 3:
```
from random import *
for i in range(10):
print(randrange(1,11,2))
```

Output
1
3
9

5
7
1
1
1
7
3

5. choice() function:

It did not return random number.

It will return a random object from the given list or tuple.

Eg:

```
from random import *
list=["Sunny","Bunny","Chinny","Vinny","pinny"]
for i in range(10):
print(choice(list))
```

Output

Bunny

pinny

Bunny

Sunny

Bunny

pinny

pinny

Vinny

Bunny

Sunny

Math Module: import math

 Use to handle mathematical operations with less lines of code:
Contains inbuilt mathematical functions as follows:

a)math.pi-->Return Value of Pi

b)math.factorial()--->math.factorial(4)---24

c)math.ceil()--->math.ceil(4.3)--->5(return greater value)

d)math.floor()--->math.floor(4.9)--->4(return smaller value)

e)math.trunc():Truncate Numbers (When you get a number with a decimal point, you might want to keep only the integer part and eliminate the decimal part. The math module has a function called trunc() which lets you do just that.)

math.trunc(12.44)---->12

f)math.pow()--->math.pow(x,y)--->x^y Returns Float Value

g)math.sqrt()--->returns square root of a number   Returns Float Value