Functions defined in modules

Pre-defined functions that are in particular modules and can only be used when corresponding module is imported.

e.g.
import math
sin(), cos(),abs(),floor() etc

Modules

Modules

•is a part of a program

•used for dividing up the program into smaller, more easily understood, reusable parts.

A module is a file containing

1.Python definitions and statements.

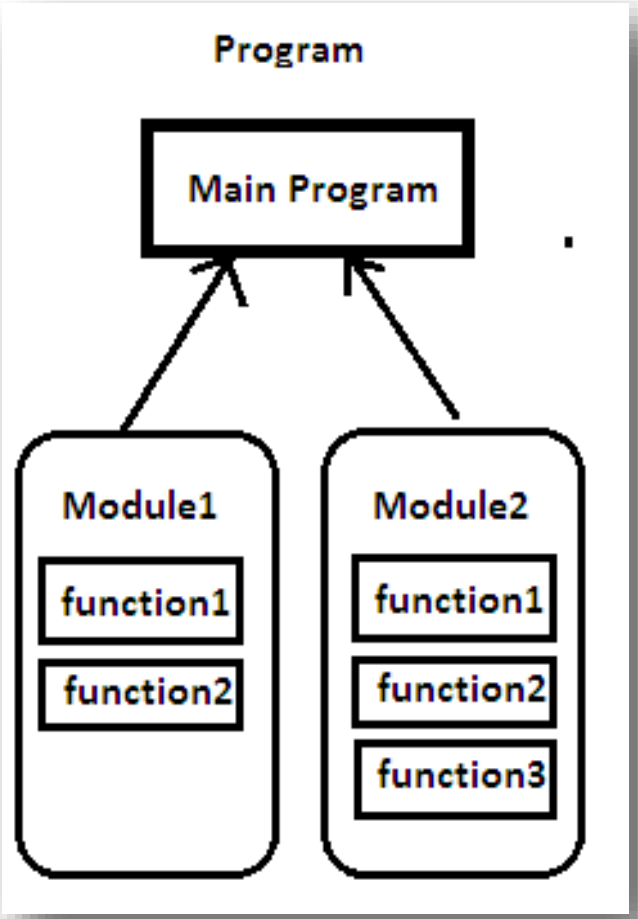
2.Module can define functions, classes and variables.

3.A module can also include runnable code.

4.Grouping related code into a module makes the code easier to understand and use.

Python Module

Python provide inbuilt standard modules, like math, random etc.



Python Module

math module

The math module is a standard module in Python and is always available. To use mathematical functions under this module, we have to import the module using import math statement.

How to use math function

```
import math
math.sqrt(4)
```

Functions using libraries(Functions defined in Modules)

Functions available in Python Math Module

Function	Description	Example
ceil(n)	It returns the smallest integer greater than or equal to n.	math.ceil(4.2) returns 5
factorial(n)	It returns the factorial of value n	math.factorial(4) returns 24
floor(n)	It returns the largest integer less than or equal to n	math.floor(4.2) returns 4
fmod(x, y)	It returns the remainder when n is divided by y	math.fmod(10.5,2) returns 0.5
exp(n)	It returns e**n	math.exp(1) return 2.718281828459045
log2(n)	It returns the base-2 logarithm of n	math.log2(4) return 2.0
log10(n)	It returns the base-10 logarithm of n	math.log10(4) returns 0.6020599913279624
pow(n, y)	It returns n raised to the power y	math.pow(2,3) returns 8.0
sqrt(n)	It returns the square root of n	math.sqrt(100) returns 10.0
cos(n)	It returns the cosine of n	math.cos(100) returns 0.8623188722876839
sin(n)	It returns the sine of n	math.sin(100) returns -0.5063656411097588
tan(n)	It returns the tangent of n	math.tan(100) returns -0.5872139151569291
pi	It is pi value (3.14159...)	It is (3.14159...)
e	It is mathematical constant e (2.71828...)	It is (2.71828...)

Python Module

math.sqrt()

The math.sqrt() method returns the square root of a given number.

```
>>>math.sqrt(100)
10.0
>>>math.sqrt(3)
1.7320508075688772
```

Python Module

math.ceil() and math.floor()

The ceil() function approximates the given number to the smallest integer, greater than or equal to the given floating point number.

The floor() function returns the largest integer less than or equal to the given number.

```
>>>math.ceil(4.5867)
5
>>>math.floor(4.5687)
4
```

Python Module

math.pow()

The math.pow() method receives two float arguments, raises the first to the second and returns the result.

In other words, pow(2,3) is equivalent to 23.**

```
>>>math.pow(2,4)
16.0
```

Python Module

math.fabs()

Returns the absolute value of x

```
>>> import math
>>> math.fabs(-5.5)
5.5
```

The math module contains functions for calculating various trigonometric ratios for a given angle.

The functions (sin, cos, tan, etc.) need the angle in radians as an argument.

```
>>> math.sin(270)
-0.1760459464712114
```

Write the corresponding Python expressions for the following mathematical expressions:

- (i) $\frac{\sqrt{a^2+b^2+c^2}}{2a}$ — math.sqrt(a*a+b*b+c*c)/2*a
- (ii) $2-ye^{2y}+4y$ — 2-y*math.exp(2*y)+4*y
- (iii) $p + \frac{q}{(r+s)4}$ — p+q/math.pow((r+s),4)
- (iv) (cos x /tan x)+x — (math.cos(x) /math.tan(x))+x
- (v) $|e^2-x|$ — math.fabs(math.exp(2)-x)

Python Module

Random Module

The random module provides access to functions that support many operations.Perhaps the most important thing is that it allows us to generate random numbers.

random.randint()

Randint accepts two parameters: a lowest and a highest number.

```
import random
print (random.randint(0, 5))
```

This will output either 1, 2, 3, 4 or 5.

random.random()

Genereate random number from 0 to 1.If we want a larger number, we can multiply it.

```
import random
print(random.random() * 100)
```

Python
Module

randrange()

generate random numbers from a specified range and also allowing rooms for steps to be included.

Syntax : random.randrange(start(opt),stop,step(opt))

```
import random
# Using randrange() to generate numbers from 0-100
print ("Random number from 0-100 is : ",end="")
print (random.randrange(100))
```

OUTPUT

Random number from 0-100 is : 27

```
# Using randrange() to generate numbers from 50-100
print ("Random number from 50-100 is : ",end="")
print (random.randrange(50,100))
```

OUTPUT

Random number from 50-100 is : 48

```
# Using randrange() to generate numbers from 50-100 # skipping 5
print ("Random number from 50-100 skip 5 is : ",end="")
print (random.randrange(50,100,5))
```

OUTPUT

Random number from 50-100 skip 5 is : 80

Python Module

statistics module

This module provides functions for calculating mathematical statistics of numeric (Real- valued) data.

statistics.mean(data)

Return the sample arithmetic mean of data which can be a sequence or iterator.The arithmetic mean is the sum of the data divided by the number of data points(AVERAGE).

```
import statistics
print(statistics.mean([5,3,2]))
```

OUTPUT

3.3333333333333335

statistics.median(data)

Return the median (middle value) of numeric data, using the common "mean of middle two" method. If data is empty, StatisticsError is raised.

```
import statistics
print(statistics.median([5,5,4,4,3,3,2,2]))
```

OUTPUT

3.5

Python Module

- statistics.mode(data)
- Return the most common data point from discrete or nominal data. The mode (when it exists) is the most typical value, and is a robust measure of central location.If data is empty, or if there is not exactly one most common value, StatisticsError is raised.

- import statistics
- print(statistics.mode([1, 1, 2, 3, 3, 3, 3, 4]))

- OUTPUT
- 3