In Python, modules refer to the Python file, which contains Python code like Python statements, classes, functions, variables, etc. A file with Python code is defined with extension.py

For example: In Test.py, where the test is the module name.

In Python, large code is divided into small modules. The benefit of modules is, it provides a way to share reusable functions

In Python, there are two types of modules.

1. Built-in Modules
2. User-defined Modules

To import modules in Python, we use the Python import keyword.

In [1]:

```
import math

# use math module functions
print(math.sqrt(5))
```

```
2.23606797749979
```

In [3]:

```
 #Import two modules
import math, random

print(math.factorial(5))
print(random.randint(10, 20))
```

```
120
12
```

In [5]:

```
# import only factorial function from math module
from math import factorial

print(factorial(5))
```

```
120
```

In [6]:

```
#Import a module by renaming it

import random as rand

print(rand.randrange(10, 20, 2))
```

```
12
```

In [7]:

```
#Import all names
from math import *
print(pow(4,2))
print(factorial(5))

print(pi*3)
print(sqrt(100))
```

```
16.0
120
9.42477796076938
10.0
```

dir() is a powerful inbuilt function in Python3, which returns list of the attributes and methods of any object (say functions , modules, strings, lists, dictionaries etc.)

In [15]:

```
print(dir())


# Now let's import two modules
import random
import math

# return the module names added to
# the local namespace including all
# the existing ones as before
print(dir())
```

```
['In', 'Out', '_', '__', '___', '__builtin__', '__builtins__', '__doc__', '_
_loader__', '__name__', '__package__', '__spec__', '_dh', '_exit_code', '_
i', '_i1', '_i10', '_i11', '_i12', '_i13', '_i14', '_i15', '_i2', '_i3', '_i
4', '_i5', '_i6', '_i7', '_i8', '_i9', '_ih', '_ii', '_iii', '_oh', '_sh',
'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'cities_
list', 'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exit', 'ex
p', 'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma',
'gcd', 'get_ipython', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf', 'isna
n', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'math', 'modf', 'na
n', 'pi', 'pow', 'quit', 'radians', 'rand', 'random', 'remainder', 'sin', 's
inh', 'sqrt', 'tan', 'tanh', 'tau', 'test_module', 'trunc']
['In', 'Out', '_', '__', '___', '__builtin__', '__builtins__', '__doc__', '_
_loader__', '__name__', '__package__', '__spec__', '_dh', '_exit_code', '_
i', '_i1', '_i10', '_i11', '_i12', '_i13', '_i14', '_i15', '_i2', '_i3', '_i
4', '_i5', '_i6', '_i7', '_i8', '_i9', '_ih', '_ii', '_iii', '_oh', '_sh',
'acos', 'acosh', 'asin', 'asinh', 'atan', 'atan2', 'atanh', 'ceil', 'cities_
list', 'copysign', 'cos', 'cosh', 'degrees', 'e', 'erf', 'erfc', 'exit', 'ex
p', 'expm1', 'fabs', 'factorial', 'floor', 'fmod', 'frexp', 'fsum', 'gamma',
'gcd', 'get_ipython', 'hypot', 'inf', 'isclose', 'isfinite', 'isinf', 'isna
n', 'ldexp', 'lgamma', 'log', 'log10', 'log1p', 'log2', 'math', 'modf', 'na
n', 'pi', 'pow', 'quit', 'radians', 'rand', 'random', 'remainder', 'sin', 's
inh', 'sqrt', 'tan', 'tanh', 'tau', 'test_module', 'trunc']
```

###Sys Module

The sys module in Python provides various functions and variables that are used to manipulate different parts of the Python runtime environment. It allows operating on the interpreter as it provides access to the variables and functions that interact strongly with the interpreter

In [ ]:

```
```