In [ ]:
```
#What is Modules and  Package
#Modules:  collection of functions variables and class. Each and every file with .py
#extension is a module
#Package:  Collection of Modules
```

In [ ]:
```
Modula -3 based on this language python is also known as modular language
```

In [ ]:
```
#Types of modules
1.Inbuilt Modules or Builtin  Modules --> That modules that we can directly use we need not to define
that modules Internally PVM knows how many functions , classes and variable are present in that module.
2.User Defined Module --> that are prepared by Developer
```

In [ ]:
```
#How to import modules
Math --> sqrt(), ceil(), floor(),fact ......
random --> randint(),random()...........
```

In [28]:
```
import math
#print(help(math))
print(math.sqrt(4))
print(math.ceil(4.8))
print(math.floor(4.5))
print(math.pi)
print(math.e)
```
```
2.0
5
4
3.141592653589793
2.718281828459045
```

In [30]:
```
import random
#print(help(random))
for i in range(10):
    print(random.random())
```
```
0.941040106181066
0.7059438439049942
0.7081495442222925
0.5876607554408589
0.3391788573250727
0.27766087741761614
0.7871818833183066
0.1553360879825153
0.34173736263960575
0.46130697131578824
```

In [34]:
```
import random
for i in range(10):
    print(random.randint(1,200))
```
```
111
5
186
16
138
144
30
122
115
179
```

In [38]:
```
import random
for i in range(10):
    print(random.uniform(1,200))
#Uniform --> It is returning decimal Number between the given range
```
```
103.75603736512501
189.61535950201076
16.17884070539316
98.16799232292414
107.03385525011225
86.41324655683202
127.18032528958
188.5927736433548
171.40069519266916
117.72049875805477
```

In [41]:
```
#choice --> return a random object
x=["Pr","Python","Twilight"]
for i in range(10):
    print(random.choice(x))
```
```
Twilight
Python
Pr
Twilight
Pr
Python
Python
Python
Python
Pr
```

In [45]:
```
#About User Defined Module
import Functions
```
```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Input In [45], in <cell line: 2>()
      1 #About User Defined Module
----> 2 import Functions

ModuleNotFoundError: No module named 'Functions'
```

In [46]:
```
import addf
print(addf.add(10,20))
print(addf.x)
```
```
30
None
88
```

In [59]:
```
import Calculator
import importlib
importlib.reload(Calculator)

print(dir(Calculator))
print(help(Calculator))
print(Calculator.add(10,20,30))
print(Calculator.sub(10,20))
print(Calculator.mul(10,20))
print(Calculator.div(10,20))
print(Calculator.Strong(145))
```
```
['Strong', '__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', 'add', 'div', 'fact', 'mul',
'pi', 'sub']
Help on module Calculator:

NAME
    Calculator - Created on Wed Aug 24 19:22:28 2022

DESCRIPTION
    @author: praty

FUNCTIONS
    Strong(n)

    add(x, y, z)
        return the Addition

    div(x, y)

    fact(num)

    mul(x, y)
        return The multiplication

    sub(x, y)
        return the Subtratction

DATA
    pi = 3.14

FILE
    c:\users\praty\calculator.py


None
31
-10
200
0.5
Strong Number
```

In [64]:
```
from Calculator import *
import importlib
importlib.reload(Calculator)

print(dir(Calculator))
print(help(Calculator))
print(add(10,20,30))
print(sub(10,20))
print(mul(10,20))
print(div(10,20))
print(Strong(145))
```
```
['Strong', '__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', 'add', 'div', 'fact', 'mul',
'pi', 'sub']
Help on module Calculator:

NAME
    Calculator - Created on Wed Aug 24 19:22:28 2022

DESCRIPTION
    @author: praty

FUNCTIONS
    Strong(n)

    add(x, y, z)
        return the Addition

    div(x, y)

    fact(num)

    mul(x, y)
        return The multiplication

    sub(x, y)
        return the Subtratction

DATA
    pi = 3.14

FILE
    c:\users\praty\calculator.py


None
31
-10
200
0.5
Strong Number
```

In [71]:
```
#Way of Importing
#1. wITH THE HELP OF IMPORT STATEMENT --> While accessing the
#functionality of the module you need to write module name

import Calculator as Cal
import importlib
importlib.reload(Calculator)

print(dir(Calculator))
print(help(Calculator))
print(Cal.add(10,20,30))
print(Cal.sub(10,20))
print(Cal.mul(10,20))
print(Cal.div(10,20))
print(Cal.Strong(145))
```
```
['Strong', '__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', 'add', 'div', 'fact', 'mul',
'pi', 'sub']
Help on module Calculator:

NAME
    Calculator - Created on Wed Aug 24 19:22:28 2022

DESCRIPTION
    @author: praty

FUNCTIONS
    Strong(n)

    add(x, y, z)
        return the Addition

    div(x, y)

    fact(num)

    mul(x, y)
        return The multiplication

    sub(x, y)
        return the Subtratction

DATA
    pi = 3.14

FILE
    c:\users\praty\calculator.py


None
31
-10
200
0.5
Strong Number
```

In [70]:
```
#2. We can import any module like this. In this we need not to write the module name again and again.
#Synatax is --> from module_name import *

from Calculator import *
from test import *
import importlib
importlib.reload(Calculator)

print(dir(Calculator))
print(help(Calculator))
print(add(10,20,30))
print(sub(10,20))
print(mul(10,20))
print(div(10,20))
print(Strong(145))
f1()
```
```
['Strong', '__builtins__', '__cached__', '__doc__', '__file__', '__loader__', '__name__', '__package__', '__spec__', 'add', 'div', 'fact', 'mul',
'pi', 'sub']
Help on module Calculator:

NAME
    Calculator - Created on Wed Aug 24 19:22:28 2022

DESCRIPTION
    @author: praty

FUNCTIONS
    Strong(n)

    add(x, y, z)
        return the Addition

    div(x, y)

    fact(num)

    mul(x, y)
        return The multiplication

    sub(x, y)
        return the Subtratction

DATA
    pi = 3.14

FILE
    c:\users\praty\calculator.py


None
31
-10
200
0.5
Strong Number
The code is executed outside from the module
```

In [ ]:
```
#2. We can import any module like this. In this we need not to write the module name again and again.
#and we can aslso specifiy the functionality
#Synatax is --> from module_name import  Function_name
from Calculator import add
import importlib
importlib.reload(Calculator)

print(dir(Calculator))
print(help(Calculator))
print(add(10,20,30))
```

In [65]:
```
if __name__=="__main__":
    print("Hello")
else:
    print("Bye")
```
```
Hello
```

In [ ]:
```
#The special Variable  --> __name__
for every python program a special variable named as __name__ is added internally.
This variable gives you the information regarding the program is executed as
an individual program or  as a module.
```

In [66]:
```
import test
print(f1)
```
```
The code is executed outside from the module
<function f1 at 0x0000016B607B4CA0>
```

In [ ]:
```
numpy , pandas scipy
```

In [68]:
```
import numpy
import pandas
```

In [ ]:
```
from numpy import array
```

In [74]:
```
import importlib
importlib.reload(test)
#from test import wish

from wish import wish
wish()
```
```
The code is executed outside from the module
```
Out[74]:
```
'Hello Guys how are you'
```

In [80]:
```
import importlib
importlib.reload(test)
import importlib
importlib.reload(wish)
import test
import wish
print(test.level())
print(wish.level())
```
```
The code is executed outside from the module
None
Hello Guys how are you
```

In [83]:
```
from test import level
from wish import *
print(level())
```
```
Hello Guys how are you
```

In [ ]:
```
#if we are having common functions in two modules then python will always consider the last
import statement
```