- every programming language will depends on packages
- modules
- libraries

In [ ]:
```
# select a random number between 1 to 10
```

In [ ]:
```
#python program

#that very helpful ====

#anaconda organisation

#if any ppl download your anaconda======


#his code now avaialbel in package

#what is the name of package

#name: random
```

In [ ]:
```
#now random package is there in your laptop
#bcz installed anaconda

#now you want to use random package

```

Random Package

In [1]:
```
import random   # import <package_name>
```

In [2]:

```
# i want to know use case of random
dir(random)   # dir(<package_name>)
```

```
Out[2]: ['BPF',
         'LOG4',
         'NV_MAGICCONST',
         'RECIP_BPF',
         'Random',
         'SG_MAGICCONST',
         'SystemRandom',
         'TWOPI',
         '_ONE',
         '_Sequence',
         '_Set',
         '__all__',
         '__builtins__',
         '__cached__',
         '__doc__',
         '__file__',
         '__loader__',
         '__name__',
         '__package__',
         '__spec__',
         '_accumulate',
         '_acos',
         '_bisect',
         '_ceil',
         '_cos',
         '_e',
         '_exp',
         '_floor',
         '_index',
         '_inst',
         '_isfinite',
         '_log',
         '_os',
         '_pi',
         '_random',
         '_repeat',
         '_sha512',
         '_sin',
         '_sqrt',
         '_test',
         '_test_generator',
         '_urandom',
         '_warn',
         'betavariate',
         'choice',
         'choices',
         'expovariate',
         'gammavariate',
         'gauss',
         'getrandbits',
         'getstate',
         'lognormvariate',
         'normalvariate',
         'paretovariate',
         'randbytes',
         'randint',
         'random',
         'randrange',
         'sample',
         'seed',
         'setstate',
         'shuffle',
         'triangular',
         'uniform',
         'vonmisesvariate',
         'weibullvariate']
```

In [ ]:

```
# randint

random is a package

in that random, randint is method
```

In [3]:

```
help(random.randint)
```

```
Help on method randint in module random:

randint(a, b) method of random.Random instance
    Return random integer in range [a, b], including both end points.
```

In [10]:

```
random.randint(2,3)
```

Out[10]:

```
3
```

Package name: Math

In [13]:

```
# package: math

# step-1: import <package_name>
import math

# step-2: what are the methods are avialbale in math package

dir(math)


# step-3:
help(math.sin)
```

```
Help on built-in function sin in module math:

sin(x, /)
    Return the sine of x (measured in radians).
```

In [14]:

```
math.sin(90)
```

Out[14]:

```
0.8939966636005579
```

In [ ]:

```
# step-1:

import  <package_name>

#Step-2

dir(<package_name>)


#Step-3

help(<package_name>.<method_name>)
```

In [15]:

```
import random
```

In [16]:

```
random
```

Out[16]:

```
<module 'random' from 'C:\\Users\\omkar\\anaconda3\\Lib\\random.py'>
```

In [ ]:

```

```

package name: keyword

Keywords in Python are reserved words that can not be used as a variable name, function name, or any other identifier.

In [ ]:

```
#import <package_name>
```

In [17]:

```
import keyword
```

In [18]:

```
dir(keyword)
```

Out[18]:

```
['__all__',
 '__builtins__',
 '__cached__',
 '__doc__',
 '__file__',
 '__loader__',
 '__name__',
 '__package__',
 '__spec__',
 'iskeyword',
 'issoftkeyword',
 'kwlist',
 'softkwlist']
```

In [19]:
```python
help(keyword.kwlist)
```

```
Help on list object:

class list(object)
 |  list(iterable=(), /)
 |  
 |  Built-in mutable sequence.
 |  
 |  If no argument is given, the constructor creates a new empty list.
 |  The argument must be an iterable if specified.
 |  
 |  Methods defined here:
 |  
 |  __add__(self, value, /)
 |      Return self+value.
 |  
 |  __contains__(self, key, /)
 |      Return key in self.
 |  
 |  __delitem__(self, key, /)
 |      Delete self[key].
```

In [20]:

```
keyword.kwlist

# keyword
```

Out[20]:

```
['False',
 'None',
 'True',
 'and',
 'as',
 'assert',
 'async',
 'await',
 'break',
 'class',
 'continue',
 'def',
 'del',
 'elif',
 'else',
 'except',
 'finally',
 'for',
 'from',
 'global',
 'if',
 'import',
 'in',
 'is',
 'lambda',
 'nonlocal',
 'not',
 'or',
 'pass',
 'raise',
 'return',
 'try',
 'while',
 'with',
 'yield']
```

In [21]:

```
import keyword
keyword.kwlist
```

Out[21]:

```
['False',
 'None',
 'True',
 'and',
 'as',
 'assert',
 'async',
 'await',
 'break',
 'class',
 'continue',
 'def',
 'del',
 'elif',
 'else',
 'except',
 'finally',
 'for',
 'from',
 'global',
 'if',
 'import',
 'in',
 'is',
 'lambda',
 'nonlocal',
 'not',
 'or',
 'pass',
 'raise',
 'return',
 'try',
 'while',
 'with',
 'yield']
```

In [22]:

```
# show me the output pi value
import math
dir(math)
math.pi
```

Out[22]:

```
3.141592653589793
```

round

In [25]:

```
round(math.pi,3)
```

Out[25]:

```
3.142
```

In [26]:

```
val=math.pi
```

```
In [27]: val
```

```
Out[27]: 3.141592653589793
```

```
In [28]: round(val,2)
```

```
Out[28]: 3.14
```

Importnat packages in Data science

numpy ------ numerical python

pandas ------ data frame read write

matplotlib ----- plot the images/graphs visulaization

seaborn ------ graphs visulaization

skleran ------ sikit -learn (heart of Machine learning)

tensorflow ---- Deep learning

keras ----- Deep learning

pytorch ----- Deep learning package

nltk ------ natural language tool kit (NLP)

scipy ------- NLP

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```