

# Using Python Libraries

Dividing a bigger program into smaller manageable units is a good strategy of programming. These smaller units are called Libraries in Python. A Library is a collection of various similar modules. The module cater to specific type of need or application.

For example Numpy module used for scientific computing needs in python. We can also create user defined module in Python.

## Python Library

The library is collection of modules / packages that together cater to a specific type of application or requirement.

### 1. Python Standard Library

Python Standard Library contains some commonly used modules in Python as given below:

- ◆ **math module:** Provides a collection of mathematic functions to support different calculations.
- ◆ **cmath module:** Provides mathematical functions for complex numbers.
- ◆ **random module:** Provides functions for generating pseudo-random numbers.
- ◆ **statistics module:** Provides statistics mathematical functions

### 2. Python Numpy Library

This library provides some advance functionalities to create and manipulate numeric array.

3. **SciPy Library:** Provides algorithmic and mathematical tools for scientific calculations.
4. **tkinter Library:** It provides traditional Python user interface toolkit and used to create user friendly GUI applications.
5. **matplotlib Library:** This library provides various tools to produce variety of plots, charts, graphs etc.

# Import Module in Program

Python Module is a collection of various function and to use these functions in program an **import statement** take place before start the program.

**There are two forms to import Python Library.**

1. **To import entire module** (import with all functions of module)

Syntax: `import <module_name>` OR

`Import <module_name> as alias_name`

Example: `import numpy as np`

`import math`

2. **To import selected function.**

Syntax: `from <module_name> import <list_of_objects>`

Example: `from math import pi, pow`

`from math import sqrt as sq`

**`from math import *` (it is similar to `import math`)**

**[\* mean , all objects of module should import]**

**Program: write function to find the square root of given number.**

```
from math import sqrt
```

```
def square_root(n):
```

```
    if(n>=0):
```

```
        return sqrt(n)
```

```
    else:
```

```
        print("-ve Number not allowed")
```

```
        return ValueError
```

```
n=5
```

```
print(square_root(n))
```

# Built-in mathematical functions

Python provides many built-in mathematical functions (import math) which directly used as per requirement. Some of them are as given below-

| Function       | Description                           | Example                                       |
|----------------|---------------------------------------|---|
| <b>len()</b>   | Return length of sequence             | <code>len("abc xyz")</code> gives 7           |
| <b>pow()</b>   | Return $a^b$ where a and b are given  | <code>pow(2,3)</code> gives 8                 |
| <b>str()</b>   | Convert a number into string          | <code>str(123)</code> gives "123"             |
| <b>int()</b>   | Convert string into integer value     | <code>int("125")</code> gives 125             |
| <b>float()</b> | Convert string into float value       | <code>float("12.5")</code> gives 12.5         |
| <b>range()</b> | Returns an immutable sequence type    | <code>range(3)</code> gives 0,1,2             |
| <b>type()</b>  | Returns the data type of parameter    | <code>type(12.5)</code> gives <class 'float'> |
| <b>id()</b>    | Return the memory address of variable | <code>A=10 id(A)</code> gives like 1510705760 |
| <b>round()</b> | Return the round off value            | <code>round(12.21561,2)</code> gives 12.22    |

# Built-in string functions`

Python provides many built-in string functions which directly used as per requirement. Some of them are as given below-

| Function         | Description  | Example   |
|------------------|--|---|
| <b>join()</b>    | Join a string or character after each member of string iterator.<br><code>"".join("KVs")</code><br><b>Output:</b> K*V*S<br><code>"".join("KVs")</code><br><b>Output:</b> K**V**S<br><code>"\$\$".join(["KVS","JJN"])</code><br><b>Output:</b> KVS\$\$JJN | <code>"".join("KVS")</code><br><b>Output:</b> K*V*S<br><br><code>"".join(123)</code><br><b>Output:</b><br>TypeError: can only join an iterable                                |
| <b>split()</b>   | Split a string based on given character (default space) and return list of strings.<br><code>"KV, JJN, RO, JPR".split()</code><br><b>Output:</b><br>['KV', ' JJN', ' RO', ' JPR']  | <code>"KV JJN RO JPR".split()</code><br><b>Output:</b><br>['KV', 'JJN', 'RO', 'JPR']<br><br><code>"KV JHUNJHUNU JAIPUR".split("J")</code><br>['KV ', 'HUN', 'HUNU ', 'AIPUR'] |
| <b>replace()</b> | Replace the word or part of word of string with given string. It is case sensitive.  | <code>"Jhunjhunu".replace("J","T")</code><br><b>Output:</b> Thunjhunu   |

## Built-in random module functions

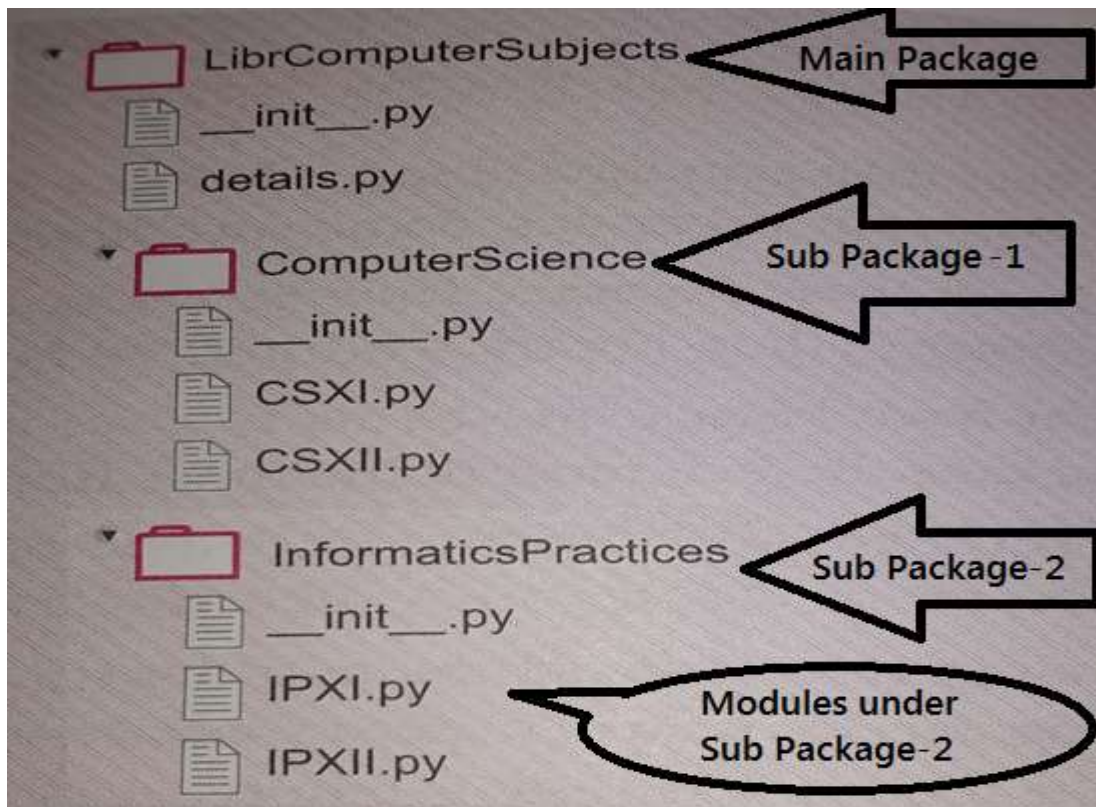
Python provides some built-in random number generator (import random) functions which directly used as per requirement. Some of them are as given below-

| Function   | Description  | Example   |
|--|--|---|
| <b>random()</b>  | It returns a random floating point number N in the range of $0.0 \leq N < 1.0$<br>(It can returns 0.0 but never 1.0) | <code>random.random()</code><br>0.021543  |
| <b>randint(a,b)</b>  | It returns a random integer number N in the range of $a \leq N \leq b$   | <code>random.randint(2,7)</code><br>Generate an integer random number between 2 and 7(both are inclusive)   |
| <b>randrange(start,stop,step)</b><br>start and step- optional<br>default start=0, step=1 | Returns integer random selected number from range.<br>(Upper Limit is exclusive)                                     | <code>random.randrange(5,2,-1)</code><br>output: may be 5,4,3<br><br><code>random.randrange(5)</code><br>output: may be any of number : 0,1,2,3,4 |

## Structure of Package

Python Package / Library is a collection of modules under common namespace. This namespace is created as a directory that contains all related modules. In order to create a package importable in the Python programs, we have to create a `__init__.py` file in the namespace (directory). A Package can contain other sub packages inside it. Every sub package should have their individual `__init__.py` file to make each sub package importable. `__init__.py` file may be an empty file.

## Example of Package



## Procedure for Creating Package

**Step-1:** Open the **Site-Package folder** (directory) of your Python installation.

Path of Site-Package directory can find by using following commands in python interactive mode.

A. `import sys`

B. `print(sys.path)`

**Step-2:** In site-package directory, **create own directory** (main package).

Remember there should not any space or special character in package name.

Example: **LibrComputerSubjects**

**Step-3:** Save the **`__init__.py`** file in main package by using python interpreter.

**Step-4:** **Save different modules** in package by using python interpreter. These modules should contains various user defined function for various purposes.

Example: **`details.py`** # Module name

Write code in details.py file

# module under LibrComputerSubjects Main Package

**def subjects():**

```
print("CBSE runs 2 Computer subjects")
print("1. Computer science")
print("2. Informatics practices")
```

**def classes():**

```
print("CBSE runs Computer subjects in two classes")
print("1. XI- computer science and informatics practices")
print("2. XII-computer science and informatics practices")
```

Step-5 Repeat Step-2, Step-3 and Step-4 for creation of sub package under main package

**Key Points to remember:**

1. **Package name and module name** should be in mind.
2. **User defined function name** created in specific module should be in mind.
3. If function required any parameter then the **type and number of parameter** should be remembered.

Step-6 Open a new Python file and write the code given below.

```
import LibrComputerSubjects.details as cs
```

```
cs.subjects()
```

```
cs.classes()
```

Run the program and check the output:

Output:

CBSE runs 2 Computer subjects

1. Computer science
2. Informatics practices

CBSE runs Computer subjects in two classes

1. XI- computer science and informatics practices
2. XII-computer science and informatics practices

**Assignment: Create above package and sub packages structure given in image and create their modules and implement in programming.**