Section-3: Module

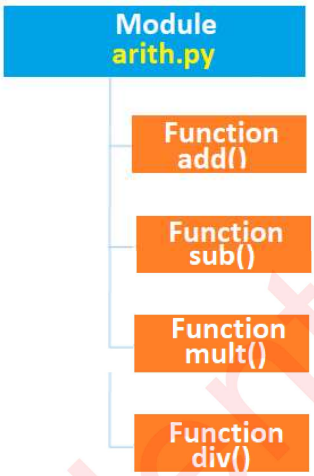
What are Modules and Packages in Python?

Modules in Python:

👉A Python **Module** is a Python File (**. py**) that contains **collection of Functions** and Global variables.

A module is a simple Python file with several functions that can be used to provide different functionalities in a program. The Python modules serve as a ready-made library available to programmers and users.

In the following example, we created a **module "arith.py"**, imported that module into a **program "arith_calc.py"**.



File name: arith.py

```
# Arithetic module
def add(a,b):
    print(a+b)
def sub(a,b):
    print(a-b)
def mult(a,b):
    print(a*b)
def div(a,b):
    print(a/b)
```

File name: arith_calc.py

```
# Importing our own Module arith.py
import arith
arith.add(10,5)
arith.sub(10,5)
arith.mult(10,5)
arith.div(10,5)
```

Output:

```
15
5
50
2.0
```

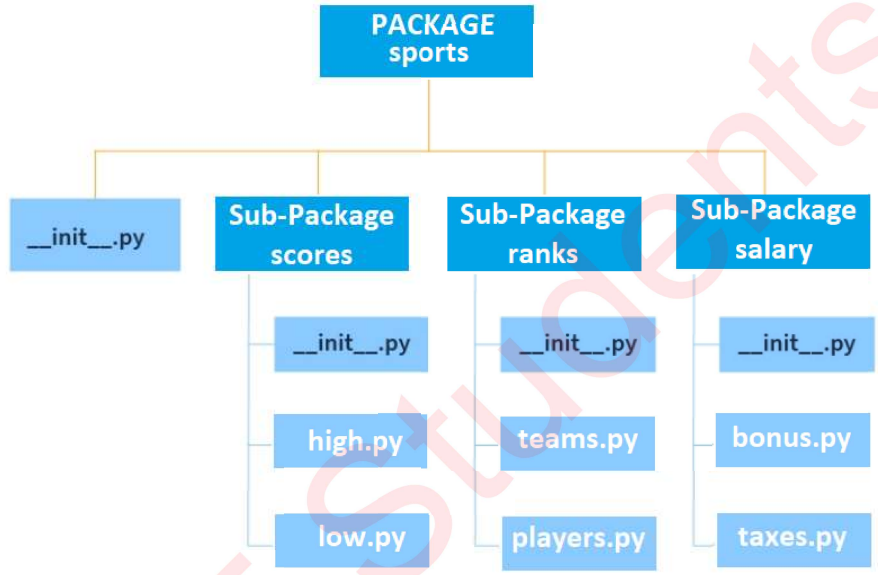
Packages in Python:

👉A Python **Package** contains,

1. **Collection of Sub-Packages or Modules** (related modules are put in a same package) and
2. **__init__.py** file through which Python interpretes it as a package. This file stores contents of the package. **Note:** The **__init__.py** Python file works as a **Constructor** for the Python Package.

👉When a module from an external package is required in a program, that package can be imported and its modules can be used.

👉Python packages ensure modularity by dividing the packages into sub-packages, making the Python project easier to maintain.




```
# math package
import math
# Square root
sqrt_val = math.sqrt(16)
print(sqrt_val)  # Output: 4.0
```

a. How to import specific attributes from a module into the current Namespace. Illustrate it with an appropriate code.

1. **Importing Module From a Package:**

Packages help in reusability of code. We use "**import**" statement in Python program to use a module. We can import only a module from a package using a dot operator (.).

Syntax:

```
import module1[, module2,... moduleN]

import package.sub-package.module

from package import module
```

Examples:

```
Import arith

import sports.ranks

from sports import ranks
```

2. **Importing specific attribute from a Module:**

To access a function called **top_rank()** of the "**rank**" module, you use the following code:
Writing.Book.edit.plagiarism_check()

```
import sports.ranks

sports.ranks.top_rank()
```

b. Briefly describe and illustrate any three Python packages with an example program.

The commonly used packages in Python are **NumPy, Pandas, SciPy, Matplotlib, Selenium, math, random or statistics,** etc.

Let's briefly describe and illustrate 3 packages:

1. **Math -** for mathematical operations,
2. **Random -** random number generation, **and**
3. **Statistics -** statistical analysis**.**

1. **Math:**

The math package is a built-in package that provides various mathematical functions and constants. It is a library of basic and advanced mathematical functions for arithmetic, trigonometry, logarithms, exponentials, etc.

- It helps us write prgrams with ease and efficiency.
- We need to import the math package before using its functions.

Here's an example:

```
# Module math
import math
# Square root
sqrt_val = math.sqrt(16)
print(sqrt_val)  # Output: 4.0

# Trigonometric functions
sin_val = math.sin(math.pi / 2)
print(sin_val)  # Output: 1.0

# Logarithm
log_val = math.log(10, 2)
print(log_val)  # Output: 3.3219280948873626

# Constants
print(math.pi)      # Output: 3.141592653589793
print(math.e)       # Output: 2.718281828459045
print(math.inf)     # Output: inf
print(math.nan)     # Output: nan
```

2. Random Package:

The random package is a built-in package in Python that is used to produce pseudo-random numbers. The numbers are not exactly random but are generated by computer algorithms.

- The Python Random Module can be used to generate a random integer, choose a random element from a list, rearrange items randomly, etc.
- We need to import random at the beginning of our code to use the Python Random Module.

Here's an example:

```
# Module random
import random
# Random integer between a range
rand_int = random.randint(1, 10)
```

```
print(rand_int)  # Output: 4

# Random float between 0 and 1
rand_float = random.random()
print(rand_float)  # Output: 0.6627291929320501

# Random selection from a list
my_list = [1, 2, 3, 4, 5]
rand_choice = random.choice(my_list)
print(rand_choice)  # Output: 3
```

3. Statistics Package:
The statistics package provides statistical functions to analyze and calculate mathematical statistics of numeric data. The statistics module was new in Python 3.4.

Here's an example:

```
# Module statistics
import statistics

# Mean
data = [1, 2, 3, 4, 5]
mean_val = statistics.mean(data)
print(mean_val)  # Output: 3

# Median
median_val = statistics.median(data)
print(median_val)  # Output: 3

# Standard deviation
std_dev = statistics.stdev(data)
print(std_dev)  # Output: 1.5811388300841898

# Variance
variance = statistics.variance(data)
print(variance)  # Output: 2.5
```

What is PIP? Explain how to install packages using PIP with at least 2 examples.

- Python packages are published to the **PyPI** (Python Package Index). PyPI hosts an extensive collection of packages, including development frameworks, tools, and libraries.
- **PIP (P**referred **I**nstaller **P**rogram**)** is the package manager that maintains packages from **PyPI** on our PC. Use **pip3** if you installed Python from the Python website or the Microsoft Store.

Requirement:
Before installing Python packages, make sure Python is installed on your machine.

➔ **To install a package using pip3:**
open a Terminal on Command Prompt on Windows
Syntax:

```
pip3 install {package_name}
```
{package_name} refers to a package you want to install.

➔ To install the **numpy** package, you would type:
pip3 install numpy

➔ To install the **scipy** package, you would type:
pip3 install scipy

➔ **To list all the installed packages:**
pip3 freeze
Or
pip3 list

```
Package            Version
------------------ ----------
numpy              1.24.1
scipy              1.10.1
```

Note: If the package has dependencies (i.e., it requires other packages for it to function), pip3 will automatically install them as well.

➔ **To use a package in Python program:**
Once the installation is complete, you can import the package into your Python code.

Ex: If you installed the **numpy** package, you could import it and use it as follows.

```
import numpy as np
arr = np.array(["I", "love", "Python", "package", "management"])
```

➔ **To update a package to the latest version:**
pip3 install --upgrade {package_name}

Ex:
To update the **numpy** package to the latest version, use the following command:
pip3 install --upgrade numpy

➔ **To uninstall a package:**
pip3 uninstall {package_name}

Ex:
pip3 uninstall numpy