Python Modules :-

- **Python Module is a file that contains built-in functions, classes,its and variables. There are many Python modules, each with its specific work.**
- **In this article, we will cover all about Python modules, such as How to create our own simple module, Import Python modules, From statements in Python, we can use the alias to rename the module, etc.**

Example for Python Modules :-

 # importing module calc.py

import calc

print(calc.add(10, 2))

Output:

12

Python Package :-

- **We usually organize our files in different folders and subfolders based on some criteria, so that they can be managed easily and efficiently. For example, we keep all our games in a Games folder and we can even subcategorize according to the genre of the game or something like that. The same analogy is followed by the Python Packages.**

How to Create Package in Python?

Creating packages in Python allows you to organize your code into reusable and manageable modules. Here's a brief overview of how to create packages:

- **Create a Directory: Start by creating a directory (folder) for your package. This directory will serve as the root of your package structure.**
- **Add Modules: Within the package directory, you can add Python files (modules) containing your code. Each module should represent a distinct functionality or component of your package.**
- **Init File: Include an __init__.py file in the package directory. This file can be empty or can contain an initialization code for your package. It signals to Python that the directory should be treated as a package.**

- **Subpackages: You can create sub-packages within your package by adding additional directories containing modules, along with their own __init__.py files.**
- **Importing: To use modules from your package, import them into your Python scripts using dot notation. For example, if you have a module named module1.py inside a package named mypackage, you would import its function like this: from mypackage.module1 import greet.**
- **Distribution: If you want to distribute your package for others to use, you can create a setup.py file using Python's setuptools library. This file defines metadata about your package and specifies how it should be installed.**

Example of the Package in Python :-

```
# module1.py

def greet(name):

    print(f"Hello, {name}!")
```

Output :-

```
Hello, Alice!
The result of addition is: 8
```