9) Modules

- **Introduction to Python modules and importing modules.**

What Is a Python Module?

A module is simply a .py file that contains Python definitions—functions, classes, variables, and executable statements. You can think of it as a reusable code library. Grouping related functionality into modules makes your code cleaner, more maintainable, and easier to share.

- Python has many built-in modules like math, random, datetime, etc.
- You can also create your own custom modules by saving code in a .py file.
- To use a module, you import it using the import keyword.

Importing Modules

Python provides several ways to import modules:

1. Basic Import
   import math

2. Import All Components (Not Recommended)
   from math import *

- **Standard library modules: math, random.**

Python comes with a large collection of built-in modules called the Standard Library. These modules provide pre-written functions to handle common tasks like math operations, random number generation, file handling, date/time, etc.

1. **math module - for mathematical operations**
2. **random module - for generating random numbers**

1. **math module**

The math module provides access to mathematical functions like square roots, powers, trigonometry, constants (like π), etc.

```
import math
```

```
import math
print(math.sqrt(16))      # Square root → 4.0
print(math.pow(2, 3))     # Power → 8.0
print(math.factorial(5))  # Factorial → 120
```

2. random module

The random module is used to generate pseudo-random numbers and perform random actions like shuffling.

```
import random
```

```
import random


print(random.randint(1, 10))       # Random int between 1 and 10
print(random.random())             # Random float between 0 and 1
print(random.choice(['a', 'b', 'c']))  # Randomly chooses one
print(random.shuffle([1, 2, 3, 4]))    # Shuffles the list (done in place)
```

- **Creating custom modules.**

A module in Python is just a .py file containing functions, variables, or classes that can be imported and used in other Python files.

Creating a custom module means writing your own Python code in a separate file and then importing it into other scripts as needed. This makes your code more reusable and organized.

Step 1: Create a Python file (your module)

Create a file called mymodule.py:

```
# mymodule.py

def greet(name):
    print("Hello,", name)

def add(a, b):
    return a + b
```

Step 2: Use the module in another Python file

Create another file, e.g., main.py:

```
# main.py


import mymodule  # importing your custom module


mymodule.greet("Ayaan")

result = mymodule.add(10, 5)

print("Sum is:", result)
```