**Working with Modules and Handling Exceptions**

**Standard libraries**

standard Libraries(modules) are built-in modules that come with the Python Standard Library. These modules provide a wide range of functionalities like math operations, file handling, date/time, regular expressions, system interaction, and more — without needing to install anything extra.

**Mathematical Modules**

* math – Mathematical functions like sqrt(), pow(), sin(), etc.
* random – Functions to generate random numbers.
* statistics – Functions for statistical operations like mean(), median().

**Date and Time Modules**

* datetime – Date and time manipulation.
* time – Working with time-related functions like sleep(), time().

**File and Directory Management**

* os – Interact with the operating system (e.g., list files, paths).
* shutil – File operations like copy, move, delete.

**System and Environment Interaction**

* sys – Access to system-specific parameters and functions.
* platform – Information about the platform (OS, processor, etc.).
* getpass – Secure password input.

**Data Handling**

* json – Reading and writing JSON data.
* csv – Working with CSV files.
* pickle – Serializing and deserializing Python objects.

**Text and Pattern Matching**

* re – Regular expression operations.
* string – String constants and functions.

**Internet and Web Services**

* urllib – Handling URLs (open, parse).
* http – Modules for handling HTTP.
* socket – Network connections using TCP/IP.

**Error and Exception Handling**

* logging – Logging messages for tracking.
* warnings – Issue warning messages.
* traceback – Print or retrieve a stack traceback.

**Utilities**

* itertools – Functions for efficient looping.
* functools – Higher-order functions (e.g., lru\_cache, reduce()).
* collections – Specialized container datatypes (deque, Counter).

**Example**

import math

print(math.sqrt(25)) # Output: 5.0

import datetime

print(datetime.datetime.now()) # Current date and time

import random

print(random.randint(1, 10)) # Random number between 1 and 10

**Packages and import statements**

A **package** is a directory that contains multiple modules (Python .py files) and a special \_\_init\_\_.py file (can be empty, used to mark the directory as a package).

Example Package Structure:

my\_package/

│

├── \_\_init\_\_.py

├── module1.py

└── module2.py

You can use this package by importing the modules inside it.

**2. Import Statements in Python**

Python provides several ways to import code from modules and packages.

a. import module\_name

import math

print(math.sqrt(16)) # Output: 4.0

b. import module\_name as alias

import numpy as np

print(np.array([1, 2, 3]))

c. from module\_name import function\_name

from math import sqrt

print(sqrt(25)) # Output: 5.0

d. from module\_name import \* *(Not recommended)*

from math import \*

print(factorial(5)) # Output: 120