TechSaksham | SAP | edunet foundation | Microsoft

Chapter: 3

Advanced Python Programming

About the Course

“

The course on "Modules and Packages in Python" teaches organizing and extending Python code through modules and third-party packages, emphasizing modular programming benefits and practical application integration from PyPI. Meanwhile, "Reading from and Writing to Files" covers essential file handling operations, including opening, reading, and writing in various modes, alongside effective exception handling with try-except blocks. Both modules emphasize practical application, ensuring students gain hands-on experience in enhancing Python projects while maintaining best practices in code organization and error management.

”

Learning Objectives

- Understanding modules and packages for better code organization and maintainability.
- Creating and importing modules into Python scripts effectively.
- Using pip to install third-party packages for extending functionality.
- Learning essential file handling operations including reading, writing, and appending files.
- Exploring different file modes (read, write, append) for managing data.
- Mastering exception handling techniques to manage errors during program execution.
- Utilizing pip for installing and managing third-party packages within the Python environment.

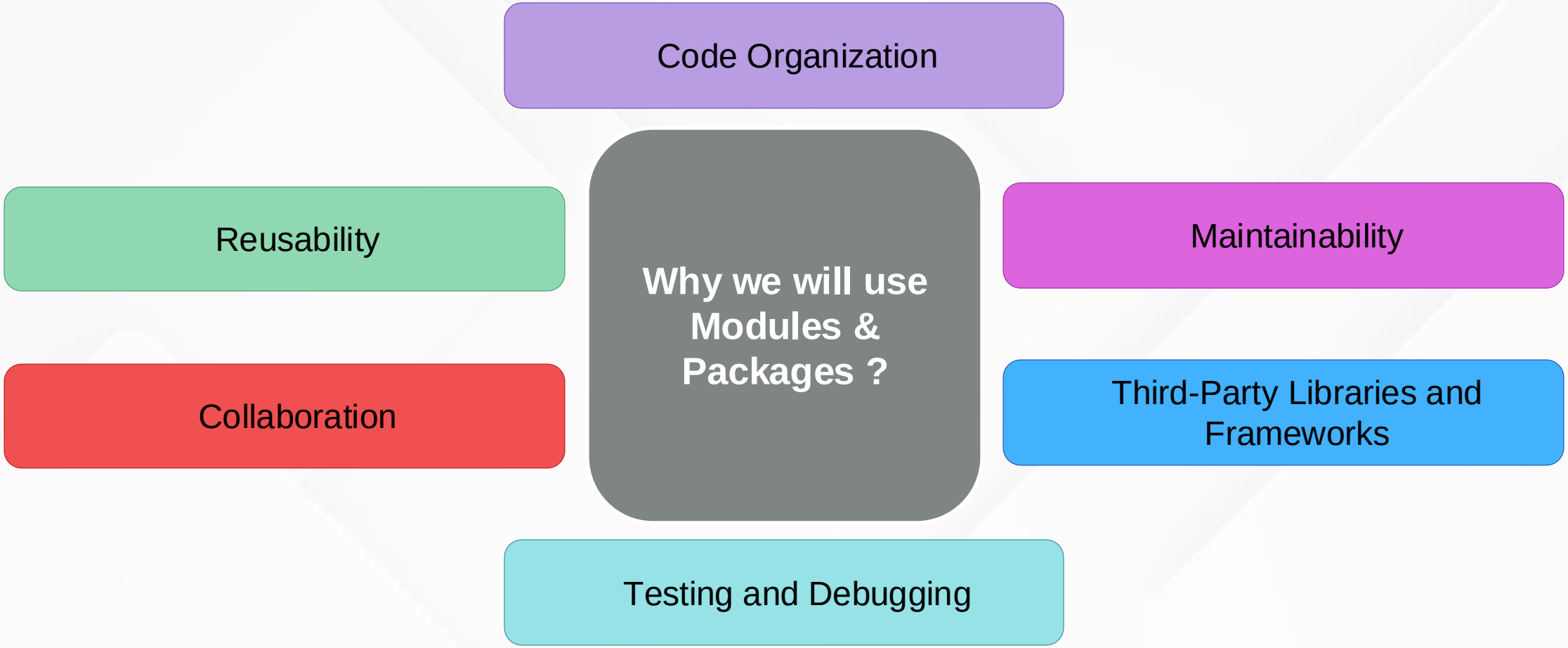
Packages & Modules Introduction

- **Modules:** A module is a file containing Python definitions and statements. It can define functions, classes, and variables, and can also include runnable code.
- Import keyword is used to import any module in Python script file.

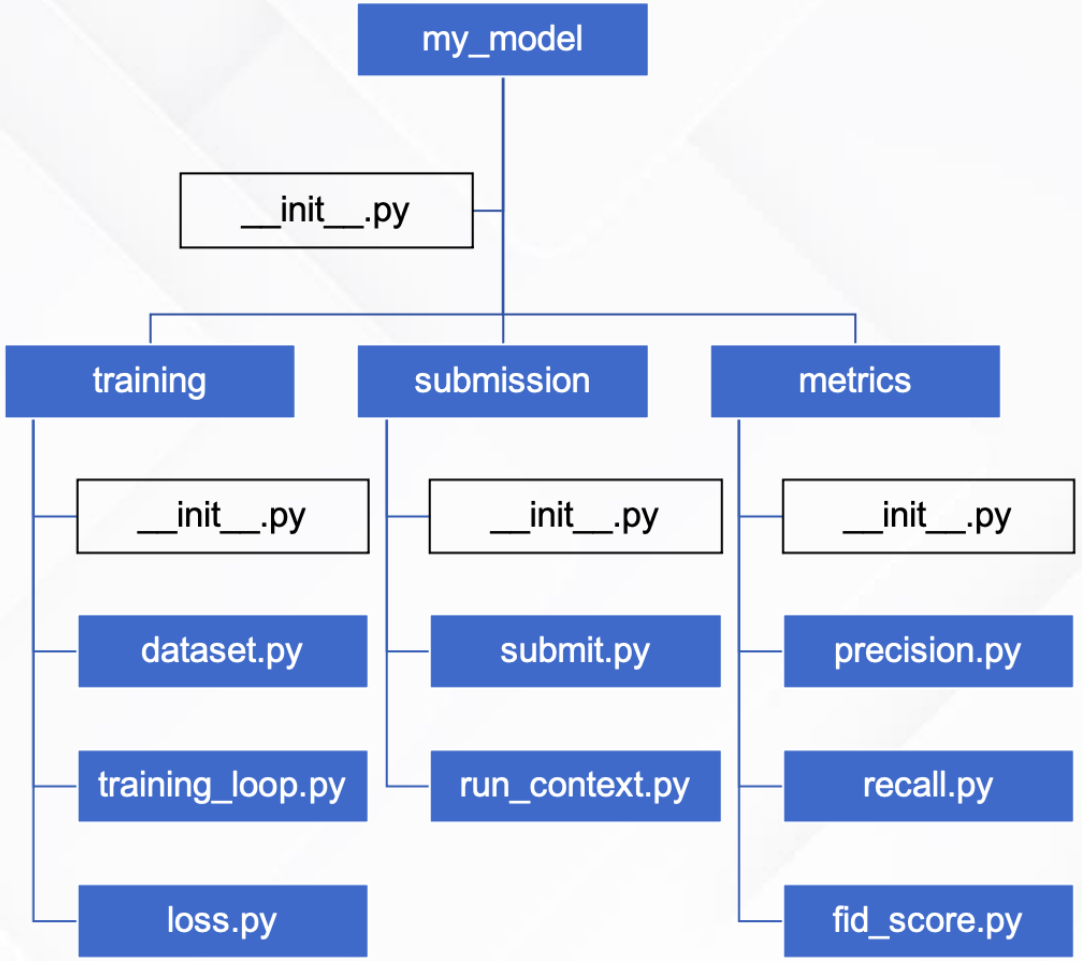
Syntax : import pandas as pd

- **Packages:** A package is a way of organizing related modules into a directory hierarchy. It is a directory that contains a special __init__.py file, which can be empty or execute initialization code for the package.

Syntax : from system import os


Code Organization
Reusability
Why we will use Modules & Packages ?
Maintainability
Collaboration
Third-Party Libraries and Frameworks
Testing and Debugging

Example - Structure of Package



Source : https://learnpython.com/blog/python-modules-packages-libraries-frameworks/1.png

Installing and using Third-party Packages

PyPI – Python Package Index

- Third-party packages provide additional functionality that can save time and effort.
- They are often hosted on the Python Package Index (PyPI).
- PyPI is the official third-party package repository for the Python programming language.

Source : https://en.wikipedia.org/wiki/Python_Package_Index

PIP - Python Package Installer

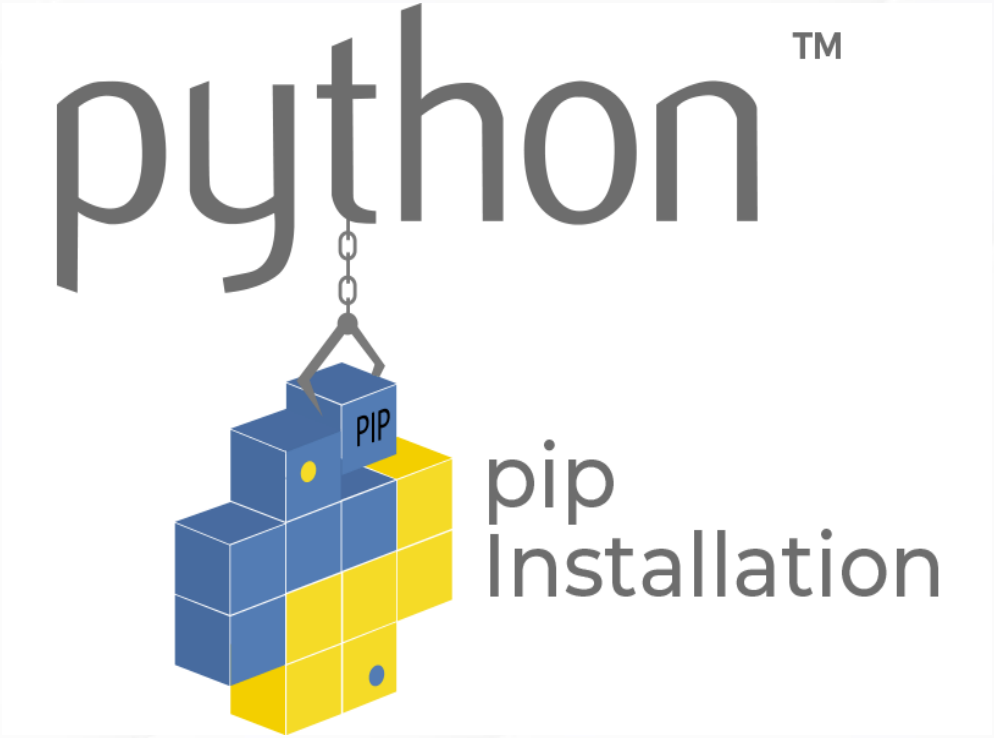
- pip is a package-management system written in Python and is used to install and manage software packages.

Syntax : pip install *package-name*
pip uninstall *package-name*

Example :

pip install seaborn
pip install numpy

pip uninstall seaborn



Source : https://en.wikipedia.org/wiki/Python_Package_Index

Class Work

Lab 10 - Creating & Importing Packages & Modules

Solution: GitHub Link

Self Practice

Lab - You are tasked with creating a small Python project that includes multiple functionalities. The project should include a module for basic mathematical operations and another module for greeting users. Additionally, organize these modules into a package for better structure.

Solution: GitHub Link

File I/O & File Handling

- File I/O and file handling are foundational skills in Python for interacting with files on your computer.
- Understanding file handling is crucial for tasks like reading, writing, and managing files efficiently.
- This module covers opening, reading, and writing to text files, and explains file modes such as read, write, and append.
- Learn effective techniques for handling exceptions and errors to maintain program stability.
- Mastering these concepts allows for confident management of file operations in Python projects.
- Integrate file handling seamlessly to enhance functionality and data management in your applications.

Reading from & Writing to Files

- Python provides various functions and methods to read from and write to files. Understanding these basics is crucial for working with data stored in files.

There are various functions as listed below,

1. Open - Used to open a file
2. Read - Use to content from file
3. Write - Used to write content to file

Opening Files

- To open a file in Python, you use the open() function. This function requires at least one argument: the path to the file. It can also take a second argument, the mode, which specifies the purpose for which you are opening the file.

Syntax :

```
file = open(“example.txt” , ”r”)
```

Reading Files

- To read a file in Python, you use the read() function.
- You can also use other methods like readline(), readlines() to read content line by line.

Syntax :

```
with open('example.txt', 'r') as file:
    content = file.read()
    print(content)
```

* here with keyword is used here to automatically close the file after the operation is done.

Writing to Files

- To write to a file, you can use the write() or writelines() method.

Syntax:

```
with open('example.txt', 'w') as file:
    file.write("Hello, World!\n")
```

Different File Modes

File modes are used to specify the purpose of opening a file.

Common modes include:

'r': Read (default mode)

'w': Write (creates a new file or truncates an existing file)

'a': Append (creates a new file if it does not exist)

'rb', 'wb', 'ab': Same as above but for binary files

Class Work

Lab 11 - File Handling

Solution: GitHub Link

Exception Handling

Exception handling in programming is crucial for managing unforeseen errors that can disrupt program flow. In Python, exceptions are managed using try-except blocks, where risky code is enclosed in try and potential errors are caught and managed in except. The else block executes if no exceptions occur, while the finally block ensures essential cleanup actions are performed regardless of errors. Effective exception handling not only ensures program stability but also improves debugging and user experience by providing meaningful error messages.

Understanding Exceptions and Their Types

- Exceptions are errors detected during execution and can be handled.

Some common exceptions include:

FileNotFoundError:

Raised when a file operation (such as open) is requested but the file does not exist.

ValueError:

Raised when a built-in operation or function receives an argument with the right type but an inappropriate value.

ZeroDivisionError:

Raised when the second argument of a division or modulo operation is zero.

Syntax

```
try:
file = open(‘example.txt', 'r’)
except FileNotFoundError as e:
    print(f"Error: {e}")
```

You can catch multiple exceptions by specifying them in a tuple.

```
try:
    file = open('example.txt', 'r')
    content = file.read()
    number = int(content)
except (FileNotFoundError, ValueError) as e:
    print(f"Error: {e}")
```

Class Work

Lab 12 – Exception Handling

Solution: GitHub Link

Self Practice

Create a function called read_file_contents that takes a file path as input and returns the contents of the file.

Solution: GitHub Link

Creating a Mini Project on Personal Fitness Tracker

The objective is to create a personal fitness tracker that helps users track their daily workouts, view progress summaries, and save/load data by applying Python programming concepts such as data structures, file handling, functions, and object-oriented programming (OOP).

Lab: Mini Project on Personal Fitness Tracker

Solution: **GitHub Link**

Summary

Modules in Python encapsulate functions, classes, and variables for better code organization, improving usability and readability. Packages group modules akin to folders on a computer. PyPI serves as Python's official third-party package repository. PIP, Python's package manager, facilitates easy installation of third-party modules. File handling involves methods like opening, reading, and writing files, with modes such as read, write, and append. Exceptions are errors encountered during program execution, managed via try-except blocks for graceful error handling.

Quiz

1. Python Package is a collection of ……..?

a) Multiple Modules
b) Multiple python script files
c) Multiple Functions /Classes
d) None of the above

Answer: A
Correct Ans

Quiz

2. Python Package is a collection of ……..?

a) Multiple Modules
b) Multiple python script files
c) Multiple Functions /Classes
d) None of the above

Answer: C
Multiple Functions / Classes

Quiz

3. Python Package is a collection of ……..?

a) Multiple Modules
b) Multiple python script files
c) Multiple Functions /Classes
d) None of the above

Answer: C
Multiple Functions / Classes

Quiz

4. To install any package or module which of below tool is used ?

a) PyPI
b) PIP
c) Python
d) None of the above

Answer: B
PIP

Quiz

5. Syntax for installing any package or module using PIP ?

a) pip uninstall *package_name*
b) pypi install *package_name*
c) pip install *package_name*
d) None of the above

Answer: C
pip install package_name

Quiz

6. To read a file which function is used among below……..?

a) Write()
b) Open()
c) Read()
d) None of the above

Answer: C
Read()

Quiz

7. To write a file which function is used among below……..?

a) None of the above
b) Read()
c) Write()
d) Open()

Answer: C
Write()

Quiz

8. Which is not a mode used in file handling from below?

a) “a”
b) “r”
c) “w”
d) “o”

Answer: d
“o”

Quiz

9. Append mode “a” will do?

a) Write text to new file without truncating old text.
b) Write text to new file by truncating old text.
c) Append new text to old text at last.
d) Append new text to old text.

Answer: C
Append new text to old text at last.

Quiz

10. Write mode “w” will do?

a) Write text to new file without truncating old text.
b) Write text to new file by truncating old text.
c) Append new text to old text at last.
d) Append new text to old text.

Answer: B
Write text to new file by truncating old text.

Quiz

11. Exceptions are errors that raise during program execution & can be handled?

a) True
b) False

Answer: A
True

Quiz

12. Which of following block is used to handle exceptions in Python?

a) Try & Catch
b) Try & Except
c) Try & Solve
d) Cannot say

Answer: B
Try & Except

Refrences

- https://docs.python.org/3/tutorial/modules.html
- Python PIP (w3schools.com)
- Getting Started - pip documentation v24.1 (pypa.io)
- https://docs.python.org/3/tutorial/errors.html
- https://www.tutorialspoint.com/python/python_exceptions.html

Thank you!