PYTHON TUTORIAL FOR BEGINNERS

Source: www.youtube.com/@RishabhMishraOfficial

Chapter - 21

Modules, Packages & Libraries in Python

- What is Module
- Create & Use a Module
- What is Package
- What is Library
- Python pip
- Most used Libraries

Modules in Python

A module is a single Python file (.py) containing Python code. It can include functions, classes, and variables that you can reuse in other programs.

Why use modules?

- To organize code into smaller, manageable chunks.
- To reuse code across multiple programs.

 # Create a module:

- Save the following as `mymodule.py`

```
def say_hello(name):
    return print(f"Hello, {name}!")
```

 # Use the module:

```
import mymodule
greetings.say_hello("Madhav")
# Output: Hello, Madhav!
```

Packages in Python

A package is a **collection** of modules organized in **directories** (folders) with an __init__.py file. It allows you to structure your Python projects logically.

Why use packages?

- To group related modules together.
- To create larger applications or libraries.

 # Structure Example:

```
my_package/
    __init__.py
    math_utils.py
    string_utils.py
```

 # Use the package:

```
Syntax: from my_package import <package_name>
Example: from my_package import math_utils, string_utils
```

Libraries in Python

A library is a collection of modules and packages that provide pre-written functionality for your program. Libraries are typically larger and more feature-rich than packages or modules.

Why use libraries?

To avoid writing common functionality from scratch.

To leverage powerful tools developed by the community.

Example: Python has many popular libraries, such as:

- Pandas: For data manipulation.
- Matplotlib: For plotting and visualization.

 # Using a library (Pandas):

```
import pandas as pd
```

Python PIP

pip stands for "Pip Installs Packages". It is the package manager for Python that allows you to install, update, and manage Python libraries (packages) from the Python Package Index (PyPI).

Think of pip as an app store for Python libraries. You use it to search, install, and manage Python tools, just like downloading apps on your phone.

When you use pip install <package_name>, it:

- Connects to PyPI (Python Package Index) online.
- Downloads the specified library or package.
- Installs it into your Python environment.

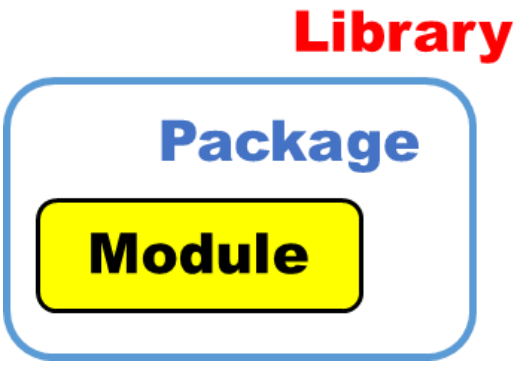
To install packages, we use: `pip install <library_name>`

Example: installing pandas to work on dataframe:

```
pip install pandas
```

Summary: Module, Package and Library

- **Module:** A single page.
- **Package:** A book containing multiple pages.
- **Library:** A book store with many books.



Concept	Key Feature	Example
Module	A single Python file with reusable code.	import math or custom file
Package	A directory of modules with an __init__.py.	from my_package import <*name*>
Library	A collection of modules/packages for functionality.	import pandas

Most Used Python Libraries

Data Analytics, data visualization and ML

Application	Library	Description	Install Command
Data Analytics	**Pandas**	Data manipulation and analysis.	pip install pandas
	NumPy	Numerical computing with array support.	pip install numpy
	SciPy	Scientific computing and technical computing.	pip install scipy
	Statsmodels	Statistical modeling and testing.	pip install statsmodels
	Dask	Parallel computing for large datasets.	pip install dask
Data Visualization	**Matplotlib**	Basic plotting and visualization.	pip install matplotlib
	Seaborn	Statistical data visualization.	pip install seaborn
	Plotly	Interactive graphs and dashboards.	pip install plotly
Machine Learning & Deep Learning	**Scikit-learn**	Classic machine learning algorithms.	pip install scikit-learn
	TensorFlow	Deep learning and ML models.	pip install tensorflow
	PyTorch	Deep learning with dynamic computation.	pip install torch torchvision
	Keras	High-level deep learning API.	pip install keras
	XGBoost	Gradient boosting for structured data.	pip install xgboost

Web Scraping, web development and game development

Application	Library	Description	Install Command
Web Scraping	**BeautifulSoup**	Parsing HTML and XML for data extraction.	pip install beautifulsoup4
	Scrapy	Advanced web scraping framework.	pip install scrapy
	Selenium	Browser automation for scraping dynamic sites.	pip install selenium
	Requests	HTTP library for fetching web pages.	pip install requests
	Lxml	Fast XML and HTML parsing.	pip install lxml
Web Development	**Django**	Full-stack web framework.	pip install django
	Flask	Lightweight web framework.	pip install flask
	FastAPI	High-performance API framework.	pip install fastapi
Game Development	**Pygame**	Game development library.	pip install pygame
	Arcade	Advanced 2D game development library.	pip install arcade
	Panda3D	Real-time 3D rendering and game creation.	pip install panda3d

Python Tutorial Playlist: **Click Here**

https://www.youtube.com/playlist?list=PLdOKnrf8EcP384Ilxra4UlK9BDJGwawg9