# Python Tools & Concepts – Study Notes

## Pip – Installing Packages

What: Package manager for Python.  
Usage: Install libraries (e.g., pip install numpy).  
Where: Any Python project needing third-party libraries.  
Advantage: Easy dependency management, access to thousands of packages (PyPI).

## Virtual Environments (venv/conda)

What: Isolated Python environments for different projects.  
Usage: venv or conda to separate dependencies.  
Advantage: Avoids version conflicts, keeps projects independent.

## NumPy

What: Library for numerical operations with arrays/matrices.  
Usage: Scientific computing, data analysis, ML preprocessing.  
Example:  
import numpy as np  
arr = np.array([1,2,3])  
print(arr.mean())  
Advantage: Fast computations, supports broadcasting.

## Pandas

What: Data analysis/manipulation library using DataFrames.  
Usage: Data cleaning, financial data, ML preprocessing.  
Example:  
import pandas as pd  
df = pd.DataFrame({'Name':['A','B'], 'Marks':[90,85]})  
print(df.head())  
Advantage: Easy tabular handling, integrates with NumPy/Matplotlib.

## Matplotlib

What: Data visualization library.  
Usage: Graphs, charts, plots for analysis/reporting.  
Example:  
import matplotlib.pyplot as plt  
plt.plot([1,2,3],[2,4,6])  
plt.show()  
Advantage: Customizable, many plot types.

## Data Preprocessing

What: Cleaning and transforming raw data before analysis.  
Usage: Handle missing values, scaling, encoding.  
Example:  
df.fillna(0, inplace=True)  
Advantage: Ensures data quality → better ML models.

## HTTP Basics

What: Communication protocol of the web.  
GET → Retrieve data  
POST → Send data  
Status Codes: 200 (OK), 404 (Not Found), 500 (Server Error)  
Usage: APIs, websites, client-server communication.  
Advantage: Standardized, universal in web systems.

## FastAPI Fundamentals

What: Modern Python framework for APIs.  
Usage: Build REST APIs, microservices.  
Example:  
from fastapi import FastAPI  
app = FastAPI()  
  
@app.get('/')  
def read\_root():  
 return {'message':'Hello World'}  
Advantage: Fast, async support, auto docs (Swagger UI).

## Pydantic

What: Data validation/parsing library.  
Usage: Validate inputs, serialize/deserialize data.  
Example:  
from pydantic import BaseModel  
  
class Student(BaseModel):  
 name: str  
 age: int  
Advantage: Type safety, integrates with FastAPI.

## Async Programming

What: Concurrent code using async/await.  
Usage: Web servers, API calls, I/O tasks.  
Example:  
import asyncio  
  
async def greet():  
 print('Hello')  
 await asyncio.sleep(1)  
 print('World')  
  
asyncio.run(greet())  
Advantage: Efficient handling of requests, scalable.

## Conclusion

These tools form the backbone of Python for Data Science, Web Development, and Modern Applications. They make projects faster, cleaner, and more reliable.