Python Development – Basics Recap

# 1. Understanding Syntax

Syntax = the set of rules that define how code should be written.

Every programming language has its own syntax (e.g., Python, Java, C, JavaScript).

Parentheses () and curly braces {} are used differently in different languages.

Python is case-sensitive (e.g., Print ≠ print).

# 2. Compiler vs Interpreter

Compiler: Translates the entire program into machine code before execution.

- Faster execution.

- Used in languages like C, C++.

Interpreter: Reads and executes code line by line.

- Slower than compiler.

- Python, JavaScript, Ruby use interpreters.

# 3. Print Function

print() is a built-in function in Python.

It displays values on the screen.

Works with numbers, characters, or full sentences.

Example:

print("Hello, World!")

print(123)

# 4. Quotations in Python

Single ' ' and Double " " quotations are both valid.

Anything inside quotes is treated as string data.

Example:

print('Hello')

print("Python is fun")

# 5. Variables

A variable is like a container/box that stores data.

You assign a value to a variable using the = operator.

Example:

name = "Faizan"

age = 25

Rules:

- Variable names must be meaningful.

- Cannot start with a number.

- Case-sensitive (Age ≠ age).

# 6. Errors in Python

Errors show up when code is written incorrectly.

In most IDEs, errors are shown in red color.

Example: writing Print("Hello") instead of print("Hello") will throw an error.

# 7. Extensions & File Naming

Python files are saved with the .py extension.

Example: hello.py

Create project folders (e.g., Coding Exercises) to keep your work organized.

# 8. Coding Practice Examples

Star Patterns (using print() and loops):

print("\*")

print("\*\*")

print("\*\*\*")

print("\*\*\*\*")

print("\*\*\*\*\*")

Output:

\*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

Other patterns can be built with loops for practice.