## Day 5 - Comments, Escape Sequences & Print in Python

### 🗨️ Python Comments

Comments are ignored by the Python interpreter. Used for **explaining code** or **disabling lines temporarily**.

#### 🔹 Single-Line Comment

# This is a single-line comment

print("This is a print statement.")

✅ **Output:**

This is a print statement.

#### 🔹 Inline Comment

print("Hello World !!!") # Printing Hello World

✅ **Output:**

Hello World !!!

#### 🔹 Disabling a Line

print("Python Program")

# print("This won't run")

✅ **Output:**

Python Program

### 📝 Multi-Line Comments

You can use multiple # lines **or** triple quotes """ """.

#### Using #:

# This block explains a condition

# If true, it prints one line

# Else, it prints another

p = 7

if p > 5:

print("p is greater than 5.")

else:

print("p is not greater than 5.")

✅ **Output:**

p is greater than 5.

#### Using Triple Quotes:

""" This is a multi-line comment.

Explaining an if-else block below. """

p = 7

if p > 5:

print("p is greater than 5.")

else:

print("p is not greater than 5.")

✅ **Output:**

p is greater than 5.

### 🔸 Escape Sequence Characters

Used to include special characters inside strings.

#### Common Escape Sequences:

* \" → Inserts double quote
* \' → Inserts single quote
* \n → New line
* \t → Tab space

#### Example:

# Error:

# print("This doesn't "execute"")

# Correct:

print("This will \"execute\"")

✅ **Output:**

This will "execute"

### 📤 More on print() Statement

**Syntax:**

print(object(s), sep=' ', end='\n', file=sys.stdout, flush=False)

#### 🔹 Parameters:

| **Parameter** | **Description** |
| --- | --- |
| object(s) | Values to print |
| sep | Separator between objects (default = space " ") |
| end | What to print at the end (default = newline \n) |
| file | Output destination (default = screen) |
| flush | Flush the buffer (default = False) |

#### 🔹 Examples:

print("Hello", "World", sep="-", end=" END\n")

✅ **Output:**

Hello-World END