

## Day 5

# Comments, Escape sequence & Print in Python

## Python Comments

A comment is a part of the coding file that the programmer does not want to execute, rather the programmer uses it to either explain a block of code or to avoid the execution of a specific part of code while testing.

### Single-Line Comments:

To write a comment just add a '#' at the start of the line.

#### Example 1

```
#This is a 'Single-Line Comment'  
print("This is a print statement.")
```

**Output:**

*This is a print statement.*

#### Example 2

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

**Output:**

*Hello World !!!*

#### Example 3:

```
print("Python Program")  
#print("Python Program")
```

**Output:**

*Python Program*

## Multi-Line Comments:

To write multi-line comments you can use '#' at each line or you can use the multiline string.

**Example 1:** The use of '#'.

```
#It will execute a block of code if a specified condition is true.  
#If the condition is false then it will execute another block of code.  
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.*

**Example 2:** The use of multiline string.

*"""This is an if-else statement.*

*It will execute a block of code if a specified condition is true.*

*If the condition is false then it will execute another block of code."""*

*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

To insert characters that cannot be directly used in a string, we use an escape sequence character.

An escape sequence character is a backslash \ followed by the character you want to insert.

An example of a character that cannot be directly used in a string is a double quote inside a string that is surrounded by double quotes:

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

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

## More on Print statement

The syntax of a print statement looks something like this:

```
print(object(s), sep=separator, end=end, file=file, flush=flush)
```

## Other Parameters of Print Statement

1. **object(s):** Any object, and as many as you like. Will be converted to string before printed
2. **sep='separator':** Specify how to separate the objects, if there is more than one. Default is ' '
3. **end='end':** Specify what to print at the end. Default is '\n' (line feed)
4. **file:** An object with a write method. Default is sys.stdout

Parameters 2 to 4 are optional