

Day 2



Comments, Escape Sequences & Print Statement:

What are Comments?

Comments are notes written inside code for humans to understand. Python ignores comments while running the program.

Why are Comments used?

- To explain code
- To make code easy to read
- To remember why we wrote something

Types of Comments

1. Single-line comment

Use #

Example:

A screenshot of a terminal window titled "main.py". The code contains two lines: "#Single line commnet" and "print("Hello, World!")". Below the code, there are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected, showing the command "PS E:\Python\Day1-10\Day2> python .\main.py" followed by the output "Hello, World!".

2. Multi-line comment

Python does not have a special syntax, but we use """ """ or multiple #.

Example:

```
4     """This is a multi-line comment."""
5     print("This is a multi-line comment example.")
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
● PS E:\Python\Day1-10\Day2> python .\main.py
This is a multi-line comment example.
```

What is the print statement?

print() is used to show output on the screen.

Example:

```
7     print("Aditya")
8     print(7)
9     print('Utsav')
10    print(7+8)
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
● PS E:\Python\Day1-10\Day2> python .\main.py
Aditya
7
Utsav
15
```

What are Escape Sequences?

Escape sequences are special characters used inside strings to add:

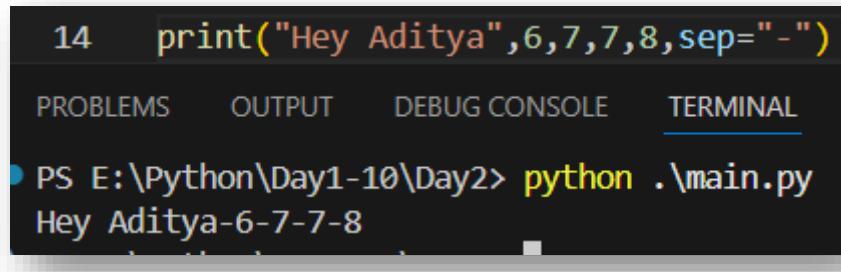
- new line
- tab space
- quotes
- special formatting

They start with a backslash (\).

Example:

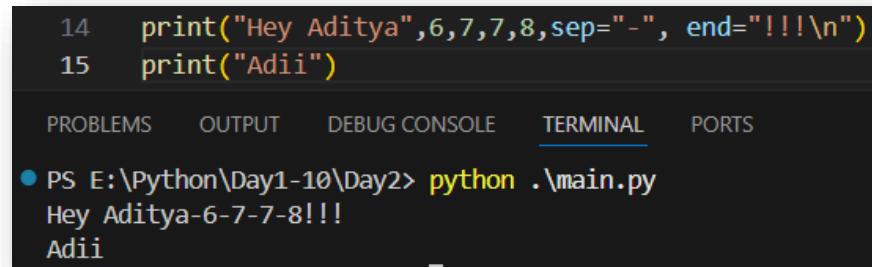
```
12    print("Name:\tAditya\nCourse:\tPython")
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
● PS E:\Python\Day1-10\Day2> python .\main.py
Name: Aditya
Course: Python
```

Example: using “sep” with the print.



```
14     print("Hey Aditya",6,7,7,8,sep="-")  
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL  
● PS E:\Python\Day1-10\Day2> python .\main.py  
Hey Aditya-6-7-7-8
```

Example: using “sep” and “end” with the print.



```
14     print("Hey Aditya",6,7,7,8,sep="-", end="!!!\n")  
15     print("Adii")  
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS  
● PS E:\Python\Day1-10\Day2> python .\main.py  
Hey Aditya-6-7-7-8!!!  
Adii
```

Summary:

- Comments are notes in code and are ignored by Python
- Use # for single-line comments
- print() is used to display output on the screen
- Escape sequences add special formatting in text
- Common escape sequences:
 - \n → new line
 - \t → tab
 - \" → double quote
 - \\ → backslash

Variables and Data Types:

What is a Variable?

A variable is a name that stores a value in memory. You can think of a variable as a container that holds data.

Why do we need Variables?

- To store data
- To reuse values
- To make programs dynamic

How to create a Variable?

In Python, you do not need to declare a type.

Example:

```
x = 10
```

```
name = "Aditya"
```

Rules for Naming Variables:

- Must start with a letter or _
- Can contain letters, numbers, _
- Cannot start with a number
- Cannot use spaces

Example:

Valid:

```
age = 21
```

```
_user = "Admin"
```

Invalid:

```
1age = 21
```

```
my name = "A"
```

Example: using variable and printing that too.

```
17  x = 9
18  name = "Aditya"
19  print(x)
20  print(name)

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
● PS E:\Python\Day1-10\Day2> python .\main.py
9
Aditya
```

Example: variables can change value.

```
22     x = 9
23     x = 8
24     print(x)

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL
● PS E:\Python\Day1-10\Day2> python .\main.py
8
```

What is a Data Type?

Data type tells Python what kind of data a variable is storing.

Common Data Types:

- int → whole numbers
- float → decimal numbers
- str → text
- bool → True/False

Example:

```
27     name = "Aditya" #String
28     a = 7           #Integer
29     b = 8.5         #Float
30     is_student = True #Boolean
```

Example: to get the type of the variable.

```
27     print(type(7))
28     print(type("Aditya"))

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PO
● PS E:\Python\Day1-10\Day2> python .\main.py
<class 'int'>
<class 'str'>
```

Example: Multiple Variables Assignment

The screenshot shows a code editor interface with the following content:

```
31     a, b, c = 1, 2, 3
32     print(a, b, c)
```

Below the code, there are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL. The TERMINAL tab is selected, showing the command:

```
● PS E:\Python\Day1-10\Day2> python .\main.py
```

The output in the terminal is:

```
1 2 3
```

Summary:

- Variable is a container to store data
- Python variables do not need type declaration
- Variables can change values
- Data Types tell the type of data stored
- Common data types:
 - int → whole numbers
 - float → decimal numbers
 - str → text
 - bool → True/False
- `type()` is used to check data type

--The End--