

Output and Input Functions in Python

In Python, **output** means **print** displaying information on the screen while **input** means taking data from the user from keyboard.

The two most commonly used functions are:

print()	used to display output.
input()	used to take input from the user.

1) print() Function

The print() function is used to display or print information on the screen.

Basic syntax

```
print( literal , variable, ..)
```

Example:

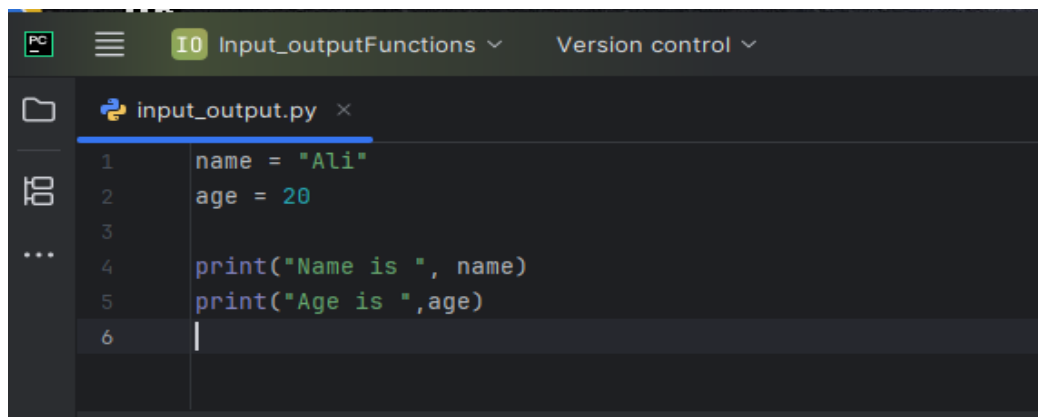
```
print("Hello World")
```

Output:

Hello World

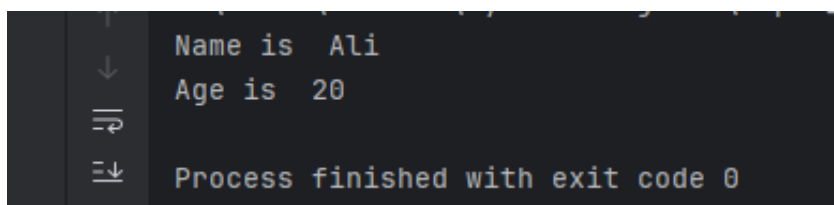
Printing Text and Variables Together

You can use print() to display the value stored in a variable.

A screenshot of a code editor window titled 'input_outputFunctions'. The editor shows a file named 'input_output.py' with the following code:

```
1 name = "Ali"
2 age = 20
3
4 print("Name is ", name)
5 print("Age is ",age)
6
```

Output

A screenshot of the program's output in a terminal window. It shows the text 'Name is Ali' and 'Age is 20' on two separate lines. At the bottom, it says 'Process finished with exit code 0'.

©Copy Right

Using f-strings

A more modern and convenient way to combine text and variables is an **f-string**.

```
name = "Ali"  
age = 20  
  
print(f"My name is {name} and I am {age} years old.")
```

Output:

```
My name is Ali and I am 20 years old.
```

The **f** before the string allows variables to be placed inside {}.

This is one of the most convenient ways to format output in modern Python.

sep Parameter

The **print()** function has a **sep** parameter that controls what is placed between multiple values.

Normally:

```
print("Ali", "Ahmed", "Khan")
```

Output

```
Ali Ahmed Khan
```

You can change the separator:

```
print("Ali", "Ahmed", "Khan", sep="-")
```

Output

```
Ali-Ahmed-Khan
```

Ali-Ahmed-Khan

Another example:

```
print(2026, 10, 6, sep="/")
```

Output:

```
2026/10/6
```

end Parameter

Normally, print() moves to a new line after displaying something.

You can change this behavior using end.

```
print("Hello", end=" ")  
print("World")
```

Another e

Output:

```
HelloWorld
```

2. Input() Function

The input() function allows a Python program to receive information from the user through the keyboard.

Syntax

```
Variable_Name = input("message")
```

For example:

For example:

```
name = input("Enter your name: ")  
print(name)
```

Output:

```
Enter your name: Ali  
Ali
```

Here:

- "Enter your name: " is the message displayed to the user.
- The user enters Ali.
- input() returns the entered value.
- The value is stored in the variable name.

Important Note: input() returns a string

Even if the user enters a number, Python normally treats it as a **string (str)**.

```
age = input("Enter your age: ")  
  
print(age)  
print(type(age))
```

If the user enters 20, the output will be:

```
20  
<class 'str'>
```

So, if you want to perform mathematical calculations, you need to convert the input. Use `int()` to convert the input into an integer.

Taking an integer

Use `int()`:

```
age = int(input("Enter your age: "))  
  
print("Your age is", age)
```

Example: Addition

```
a = int(input("Enter first number: "))  
b = int(input("Enter second number: "))  
  
sum = a + b  
  
print("Sum =", sum)
```

Taking a decimal number

Use `float()` when you need a decimal number.

```
price = float(input("Enter price: "))
print("Price =", price)
```

Type Conversion Function in Python

int()	This function is use to convert string data into Integer value/data
float()	This function is use to convert data into floating-point decimal
str()	This function is use to convert data from integer into string.
bool()	This function is use to convert string data into Boolean.

8. Taking Multiple Inputs

The screenshot shows the PyCharm IDE with a file named `InputOutputFunction.py`. The code in the editor is as follows:

```
1 name = input("Enter your name.....: ")
2 age = int(input("Enter your age.....: "))
3 per = float(input("Enter Your Percentage..: "))
4 present = bool(input(" Are You Present: [true/false]..: "))
5
6 print("\n\n\n outPut\n Your name is .....: " , name)
7 print(" Your age is .....: " , age," years Olds")
8 print(" Your Percentage is.....: " , per)
9 print(" Your Present in Class..: " , present)
```

Below the editor, the 'Run' console shows the execution of the script. The user inputs are: 'Umer Ahmed' for name, '19' for age, '89.345' for percentage, and 'true' for presence. The output matches the print statements in the code.

```
Run InputOutputFunction
C:\Users\Student\PycharmProjects\Input_Output_Function\.venv\Scripts\python.exe
Enter your name.....: Umer Ahmed
Enter your age.....: 19
Enter Your Percentage..: 89.345
 Are You Present: [true/false]..: true

outPut
Your name is .....: Umer Ahmed
Your age is .....: 19 years Olds
Your Percentage is.....: 89.345
Your Present in Class..: True

Process finished with exit code 0
```

Note: `\n` a special instruction for move cursor on to the new line.

Note: `bool(input())` usually does **not** work the way beginners expect, because any non-empty string is considered True. For yes/no input, it's better to explicitly check the text

Exercise

Theory Questions

1. How you understand to **input** and **output** in the python programming language.
2. Describe the input function of python with example.
3. How work f-string with print function?
4. Which Auto-Global for use uploaded files and writes their element names.
5. What difference between get and post method when form submitted.

Practical Questions

1. Write a simple program of following using post method:

Enter Student Name:

Enter Student G.R #:

Enter Class:

Enter Section:

// Output should be display on new tab of browser.

2. Input your Gr.No and Course Name and then print using get and post method.
3. Write make a simple form where input your Gr No. and Name with submit and reset button after submit form display Gr No. and name to other web page.
4. You send Roll number and Name by using hyperlinks from one page to other page.
5. Write syntax of the **action** and **method** attributes in the opening <form> tag.

Objective MCQ's

1. Which of the following auto-global can you use to access submitted form values?
 - a) \$_SERVER
 - b) \$_REQUEST
 - c) \$_POST
 - d) Nothing all
2. Which of the following separates the URL from the form data in a get request
 - a) ?
 - b) &
 - c) =
 - d) +
3. Which of the following separates multiple name/values pair from each other in a get request?
 - a) =
 - b) +
 - c) ?
 - d) &
4. What is the maximum length of a value in a get request?

- a) 800
 - b) 512
 - c) 1000
 - d) 2000
5. Which auto-global ,Returns the filename of the currently executing script
- a) `$_SERVER['PHP_SELF']`
 - b) `$_SERVER['SERVER_NAME']`
 - c) `$_SERVER['HTTP_HOST']`
 - d) `$_SERVER['SCRIPT_NAME']`
6. Which Auto-Global array get information about upload files.
- a) `$_GET`
 - b) `$_POST`
 - c) `$_FILES`
 - d) `$_SERVER`