User Input

A comprehensive tutorial

Table of contents

01

User Input

Different ways of taking user input.

Basic User Input

The **input()** function is used to capture user input in Python. It **reads** a line from the standard **input** (usually the keyboard), **converts it into a string**, and **returns it**.

Python

```
name = input("Enter your name: ")
print("Hello, " + name + "!")
```

Prompt: The string passed to **input()** (e.g., **"Enter your name: "**) is displayed to the user as a prompt.

Return Type: The value returned by **input()** is always of type **str (string)**, even if the user enters numbers.

Type Conversion - Integer Conversion

Since **input()** returns a string, you often need to convert it into another data type, like **int**, **float**, or **bool**, depending on your needs.

Convert the input to an integer

Python

```
age = int(input("Enter your age: "))
print("You are", age, "years old.")
```

Prompt: The string passed to **input()** (e.g., **"Enter your age: "**) is displayed to the user as a prompt.

Return Type: The value returned by **input()** is converted to type **int (integer)**.

Type Conversion - Float Conversion

Since **input()** returns a string, you often need to convert it into another data type, like **int**, **float**, or **bool**, depending on your needs.

Convert the input to an Float

Python

```
price = float(input("Enter the price: "))
print("The price is", price)
```

Prompt: The string passed to **input()** (e.g., **"Enter the price: "**) is displayed to the user as a prompt.

Return Type: The value returned by **input()** is converted to type float **(float)**, even if the user enters numbers.

Type Conversion - Boolean Conversion

Since **input()** returns a string, you often need to convert it into another data type, like **int**, **float**, or **bool**, depending on your needs.

Convert the input to an boolean

Python

```
is_sunny = input("Is it sunny today?
(yes/no): ") == 'yes'
print("Is it sunny? ", is_sunny)
```

Prompt: The string passed to **input()** (e.g., **“Is it suny today? (yes/no): "**) is displayed to the user as a prompt.

Return Type: The value returned by **input()** is converted to type **bool (boolean)**.

02

Handling Multiple Inputs

Accepting and handling multiple inputs

Handling Multiple Inputs

You can handle multiple inputs from the user in a single line by using **split()**.

Python

```
data = input("Enter your name, age, and city (comma-separated): ").split(',')
name, age, city = data[0], int(data[1]), data[2]
print(f"Name: {name}, Age: {age}, City: {city}")
```

Output

```
Enter your name, age, and city (comma-separated): amar,12,noida
Name: amar, Age: 12, City: noida
```

`split(',')` splits the input string into a list of strings based on the comma delimiter.

03

Input Validation

Scenarios where user input is needed.

Input Validation

There are basically **two** methods to validate user inputs.

Using try-except:

This technique allows you to **attempt to execute** a block of code (such as converting input to a number) and **handle any errors gracefully** if something goes wrong..

Using in-built methods:

Python provides several **built-in string methods** that can be used to validate user input. These methods allow you to **check** if the **input meets certain criteria**, such as being numeric, alphabetic, lowercase, or uppercase.

Why Input Validation is Important

Prevents Errors

Ensures that your program receives the expected type of data, avoiding unexpected crashes.

Improves UX

Provides clear feedback when the input is invalid, helping users understand what's expected.

Enhances Security

Helps prevent malicious inputs that could exploit your program.

04

Best Practices

Best Practices for User Input Handling

Best Practices for User Input Handling

Always Validate: Never trust user input. Always validate it to prevent errors and security vulnerabilities.

Use Clear Prompts: Ensure your prompts are clear so users know exactly what is expected.

Handle Errors Gracefully: Use try-except blocks to handle errors and provide feedback to the user.

Consider Edge Cases: Think about unusual or unexpected inputs and how your program should handle them.

Knowledge Reinforcement

1

Question

What function is used to capture user input in Python?

Answer

A) input()

B) print()

C) read()

D) capture()

1

Question

What function is used to capture user input in Python?

Answer

A) input()

B) print()

C) read()

D) capture()

2

Question

When using the **input()** function, what data type is returned by default?

Answer

A) Integer

B) Float

C) String

D) Boolean

2

Question

When using the **input()** function, what data type is returned by default?

Answer

A) Integer

B) Float

c) String

D) Boolean

3

Question

Which method can be used to **split** user input into multiple parts in Python?

Answer

A) partition()

B) split()

C) divide()

D) break()

3

Question

Which method can be used to **split** user input into multiple parts in Python?

Answer

A) partition()

B) split()

C) divide()

D) break()

4

Question

If the user input is **"John,25,New York"** and you use **split(",")**, what is the output?

Answer

A) ['John', '25', 'New York']

B) ['John', 25, 'New York']

C) ['John25New York']

D) ['John', '25']

4

Question

If the user input is **"John,25,New York"** and you use **split(",")**, what is the output?

Answer

A) ['John', '25', 'New York']

B) ['John', 25, 'New York']

C) ['John25New York']

D) ['John', '25']

5

Question

Which method can be used to check if a string contains **only digits**?

Answer

A) isdigit()

B) isalpha()

C) islower()

D) isnumeric()

5

Question

Which method can be used to check if a string contains **only digits**?

Answer

A) isdigit()

B) isalpha()

C) islower()

D) isnumeric()

6

Question

In a **try-except** block, what happens if the **input is invalid** and **causes an error**?

Answer

A) The program crashes.

B) The try block runs again.

C) The code in the except block is executed.

D) The program exits automatically.

6

Question

In a **try-except** block, what happens if the **input is invalid** and **causes an error**?

Answer

A) The program crashes.

B) The try block runs again.

C) The code in the except block is executed.

D) The program exits automatically.

7

Question

What function would you use to convert user input to an **integer**?

Answer

A) str()

B) int()

C) float()

D) bool()

7

Question

What function would you use to convert user input to an **integer**?

Answer

A) str()

B) int()

C) float()

D) bool()

8

Question

Which type conversion is typically used when dealing with monetary values like prices?

Answer

A) Convert to int

B) Convert to str

C) Convert to float

D) Convert to bool

8

Question

Which type conversion is typically used when dealing with monetary values like prices?

Answer

A) Convert to int

B) Convert to str

C) Convert to float

D) Convert to bool

9

Question

If a user is asked to answer a yes/no question, which conversion method would best suit the input?

Answer

A) Convert to int

B) Convert to float

C) Convert to str

D) Convert to bool

9

Question

If a user is asked to answer a yes/no question, which conversion method would best suit the input?

Answer

A) Convert to int

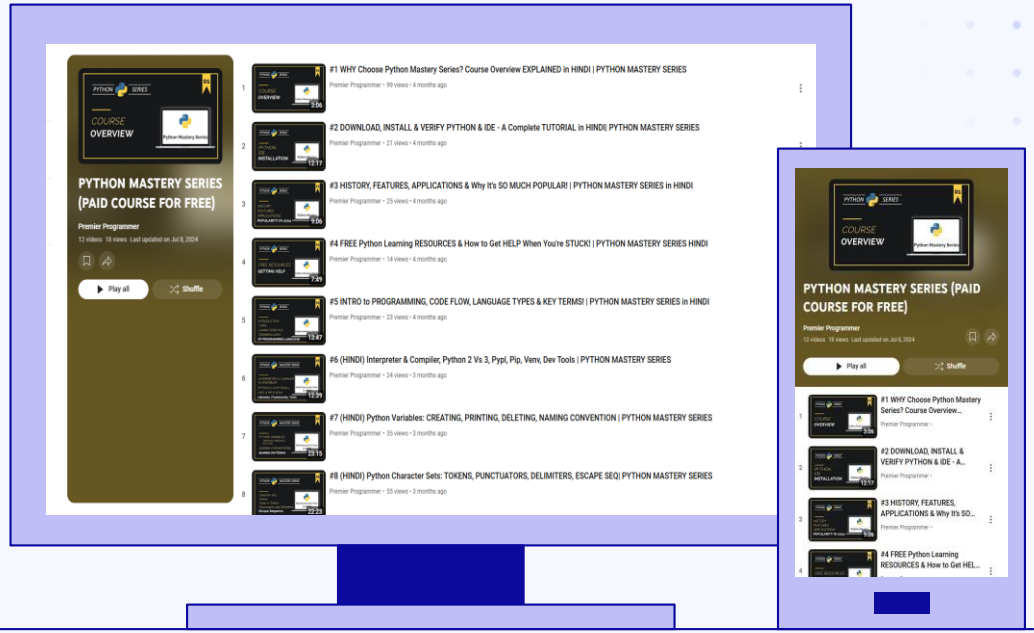
B) Convert to float

C) Convert to str

D) Convert to bool

WATCH

Level up your coding with each episode in this focused Python series.



Next Video!

Printing Output