Input and Output in Python

Python Programming

Bindeshwar Singh Kushwaha

PostNetwork Academy

What is Input and Output?

- Input refers to data provided by the user to the program.

What is Input and Output?

- Input refers to data provided by the user to the program.
- Output is the information displayed by the program to the user.

What is Input and Output?

- Input refers to data provided by the user to the program.
- Output is the information displayed by the program to the user.
- Python provides built-in functions for handling input and output.

Example: Basic Input and Output

Python Code

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

Taking Input in Python

- Python uses the input() function to take user input.

Taking Input in Python

- Python uses the input() function to take user input.
- Input is always read as a string.

Example: Taking Input

Python Code

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

Type Conversion for Input

Python Code

```
age = int(input("Enter your age: "))
print("Next year, you will be", age + 1)
```

Displaying Output in Python

- Python uses the print() function to display output.

Displaying Output in Python

- Python uses the print() function to display output.
- The print function supports multiple arguments.

Example: Displaying Output

Python Code

```python
print("Name:", name, "Age:", age)
```

String Formatting in Output

- Python provides multiple ways to format output.

Example: String Formatting

Python Code

```
name = "Alice"
age = 25
print(f"Name: {name}, Age: {age}")
print("Name: {}, Age: {}".format(name, age))
print("Name: %s, Age: %d" % (name, age))
```

Taking Multiple Inputs

- Python allows multiple inputs in one line using split().

Example: Taking Multiple Inputs

Python Code

```
name, age = input("Enter name and age: ").split()
print(f"Name: {name}, Age: {age}")
```

Example: Converting Input Types

Python Code

```
x, y = map(int, input("Enter two numbers: ").split())
print("Sum:", x + y)
```

End and Separator in print()

- The end parameter in print() controls line ending.

End and Separator in print()

- The end parameter in print() controls line ending.
- The sep parameter controls the separator between values.

Example: End and Separator

Python Code

```
print("Hello", end=" ")
print("World!")  # Output: Hello World!

print("Python", "Programming", sep=" - ")  # Output: Python - Programming
```

Best Practices for Input and Output

- Always convert input types when expecting numbers.

Best Practices for Input and Output

- Always convert input types when expecting numbers.
- Use f-strings for clean and readable output.

Best Practices for Input and Output

- Always convert input types when expecting numbers.
- Use f-strings for clean and readable output.
- Use split() for multiple inputs.

Example: Best Practices

Python Code

```
marks = int(input("Enter your marks: "))
print(f"Your marks: {marks}")
```

Summary

- Python uses input() for user input.

Summary

- Python uses input() for user input.
- The print() function displays output.

Summary

- Python uses input() for user input.
- The print() function displays output.
- String formatting improves output presentation.

Summary

- Python uses input() for user input.
- The print() function displays output.
- String formatting improves output presentation.
- Multiple inputs can be taken using split().

Summary

- Python uses input() for user input.
- The print() function displays output.
- String formatting improves output presentation.
- Multiple inputs can be taken using split().
- Use type conversion when dealing with numbers.

Thank You!