

Chapter 5 : Basic I/O Operations in Python

Basic I/O Operations in Python

Welcome to Day 5 of our 45-day Python course! Today, we'll be exploring Basic Input/Output (I/O) operations in Python. I/O operations are fundamental to any programming language as they allow your program to interact with the user and handle data. By the end of this lesson, you'll be able to gather input from users and display output in your Python programs.

What Are I/O Operations?

- **Input Operations:** This refers to taking data from the user or other sources (like files or databases) and making it available for the program to use.
- **Output Operations:** This refers to sending data from the program to the user or other destinations (like files, databases, or external systems).

[Online Quiz Using Python Django: Comprehensive Guide](#)

Input in Python

In Python, you can capture user input using the `input()` function. This function reads a line of text from the user and returns it as a string.

Basic Input Example

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

- **Explanation:**

- The `input()` function prompts the user with the string provided ("Enter your name: "), waits for the user to enter a value, and then stores that value in the variable `name`.
- The `print()` function is used to display a message to the user, in this case, greeting them by name.

Input with Data Types

By default, `input()` captures everything as a string. To handle other data types, you'll need to convert the input using functions like `int()`, `float()`, or `bool()`.

```
python
Copy code
age = input("Enter your age: ") # This captures input as a string
age = int(age) # Converts the input to an integer
print("You are " + str(age) + " years old.")
```

Advanced Input Handling

You can prompt the user for multiple inputs on a single line using the `split()` function.

```
python
Copy code
x, y = input("Enter two numbers separated by a space: ").split()
x = int(x)
y = int(y)
print("Sum:", x + y)
```

- **Explanation:** The `split()` function splits the input string by spaces and returns a list of substrings. These substrings are then assigned to the variables `x` and `y`.

Complete Python Course :- [Click Here](#)

Output in Python

Output in Python is managed using the `print()` function. This function is versatile and allows you to display text, variables, and even formatted data.

Basic Output Example

```
python
Copy code
print("Welcome to Python programming!")
```

Output with Variables

You can include variables in your output by concatenating strings or using f-strings (formatted string literals).

```
python
Copy code
name = "Alice"
age = 30
print("Name:", name)
print(f"Name: {name}, Age: {age}")
```

- **Explanation:**
 - In the first `print()` statement, the variable `name` is separated by commas, which automatically adds a space between items.
 - The second `print()` statement uses an f-string (`f"..."`), which allows you to embed expressions inside string literals, enclosed by curly braces `{}`.

Formatting Output

Python provides several ways to format output, such as specifying the number of decimal places for floating-point numbers.

```
python
Copy code
pi = 3.14159
print(f"Pi rounded to 2 decimal places: {pi:.2f}")
```

Combining Input and Output

Often, you'll want to combine input and output operations in a single program to create interactive scripts.

Example: Simple Calculator

```
python
Copy code
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))

print(f"The sum is: {num1 + num2}")
print(f"The difference is: {num1 - num2}")
print(f"The product is: {num1 * num2}")
print(f"The quotient is: {num1 / num2}")
```

- **Explanation:**

- The program takes two numbers as input from the user, performs basic arithmetic operations, and prints the results.

Practice Exercises

1. User Greeting Program:

- Write a program that asks for the user's name, age, and favorite color, then displays a message using this information.

2. Basic Arithmetic Calculator:

- Extend the calculator example by adding options for the user to choose an operation (e.g., addition, subtraction, multiplication, division) before entering the numbers.

3. Temperature Converter:

- Write a program that converts temperatures from Fahrenheit to Celsius. The program should take the temperature in Fahrenheit as input and display the result in Celsius.

- **Complete Python Course :** [Click here](#)
- **Free Notes :-** [Click here](#)
- **New Project :-** <https://www.youtube.com/@Decodeit2>