

Input and Output Statements in Python

Input and output (I/O) are essential for any programming language. Python provides several ways to perform input and output operations.

Input Statements

1. `input()` Function

The `input()` function allows the user to take input from the keyboard as a string.

```
# Example of input function
name = input("Enter your name: ")
print(f"Hello, {name}!")
```

2. Reading Numeric Input

To read numeric input, you need to convert the input string to an appropriate numeric type.

```
# Example of reading numeric input
age = int(input("Enter your age: "))
print(f"You are {age} years old.")
```

Output Statements

1. `print()` Function

The `print()` function is used to output data to the standard output (screen).

```
# Example of print function
print("Hello, World!")
```

2. Formatted String Literals (f-strings)

Python 3.6+ supports formatted string literals, also known as f-strings, to include expressions inside string literals.

```
# Example of formatted string literals
name = "Alice"
age = 25
print(f"My name is {name} and I am {age} years old.")
```

3. String Formatting with `str.format()`

The `str.format()` method allows formatting strings using curly braces `{}` as placeholders.

```
# Example of string formatting with str.format()
name = "Bob"
age = 30
print("My name is {} and I am {} years old.".format(name, age))
```

4. Old-style String Formatting with % Operator

The % operator can be used for string formatting in a way similar to the C programming language.

```
# Example of old-style string formatting with % operator
name = "Charlie"
age = 35
print("My name is %s and I am %d years old." % (name, age))
```

File Input and Output

1. Opening a File

The `open()` function is used to open a file in Python. It returns a file object and takes two arguments: the filename and the mode (`r`, `w`, `a`, `b`, etc.).

```
# Example of opening a file
file = open("example.txt", "r")
```

2. Reading from a File

You can read the content of a file using methods like `read()`, `readline()`, and `readlines()`.

```
# Example of reading from a file
with open("example.txt", "r") as file:
    content = file.read()
    print(content)
```

3. Writing to a File

You can write data to a file using the `write()` method.

```
# Example of writing to a file
with open("example.txt", "w") as file:
    file.write("Hello, World!")
```

4. Appending to a File

You can append data to an existing file using the `a` mode.

```
# Example of appending to a file
with open("example.txt", "a") as file:
    file.write("Append this line.")
```

5. Closing a File

It is a good practice to close a file after you are done with it using the `close()` method or the `with` statement.

```
# Example of closing a file
file = open("example.txt", "r")
content = file.read()
file.close()
```

Stay Updated

Be sure to [star](#) this repository to stay updated with new examples and enhancements!

License

This project is protected under the MIT License.

Contact

Panagiotis Moschos - pan.moschos86@gmail.com

Note: This is a Python script and requires a Python interpreter to run.

Happy Coding

Made with [love](#) by Panagiotis Moschos (<https://github.com/pmoschos>)