

Lecture 28: I/O Operations

Standard Input and Output in Python.
Mastering 'input()' and 'print()'.

Agenda

Standard I/O Streams

The 'print()' function in detail

Formatting Output

The 'input()' function

Type Casting Input

Handling Multiple Inputs

Summary

Standard Streams

Python programs typically interact with standard streams.

stdin (Standard Input): Usually the keyboard.

stdout (Standard Output): Usually the screen/console.

stderr (Standard Error): Used for error messages.

The print() Function

Used to send data to standard output.

Signature: `print(*objects, sep=' ', end='\n', file=sys.stdout, flush=False)`.

Can print multiple objects separated by commas.

Customizing print()

The 'sep' parameter changes the separator between objects.
The 'end' parameter changes what is printed at the very end.

Output Formatting (f-strings)

Introduced in Python 3.6, f-strings offer the most readable way to format output.

Prefix string with 'f' and use '{ }' for expressions.

The input() Function

Reads a line from input, converts it to a string (stripping a trailing newline), and returns it.

Optionally takes a prompt string.

Type Casting with input()

Because 'input()' ALWAYS returns a string, you must cast it for math operations.

Handling Multiple Inputs

You can read multiple values on a single line using `'split()'`.

Flushing the Output Buffer

Sometimes output is buffered and not displayed immediately.
Use `'flush=True'` in `'print()'` to force immediate output, crucial for loading bars.

Summary

'print()' is versatile with 'sep' and 'end'.
Always cast the result of 'input()' if numerical data is expected.
f-strings are the modern standard for formatting.

Next Lecture

Next, we delve into mathematical built-in functions like `'abs()'`, `'max()'`, and `'min()'`.