

Lecture 27: Built-in Functions

Exploring Python's standard built-in functions.
Functions that are always available without importing.

Agenda

What is the Standard Library?

Type Conversions

Sequence Operations

Object Introspection

Map and Filter

Zip and Enumerate

Common Pitfalls

The Built-in Namespace

Python comes with many functions ready to use.
These reside in the 'Built-in' scope of the LEGB rule.
No 'import' statement is required.
Examples: 'print()', 'len()', 'type()', 'id()'.

Type Conversion Functions

Functions to cast between different data types.

'int(x)', 'float(x)', 'str(x)', 'bool(x)'.

Sequence & Collection Functions

Functions that operate on iterables.

'len(s)': Returns the number of items.

'sorted(iterable)': Returns a new sorted list.

'reversed(seq)': Returns a reverse iterator.

Introspection Functions

Functions to inspect objects at runtime.

'type(obj)': Returns the type of the object.

'id(obj)': Returns the memory address.

'isinstance(obj, class)': Checks inheritance.

'dir(obj)': Lists attributes and methods.

The 'enumerate()' Function

Adds a counter to an iterable and returns it as an enumerate object.
Highly useful in 'for' loops.

The 'zip()' Function

Takes iterables, aggregates them in a tuple, and returns it.

Higher-Order Functions: `map()`

Applies a function to all the items in an input iterable.

Higher-Order Functions: filter()

Constructs an iterator from elements for which a function returns True.

Summary

Built-in functions provide powerful, optimized operations.
Using them makes your code more Pythonic and efficient.
Avoid reinventing the wheel!

Next Lecture

We will focus deeply on input/output functions: `'input()'` and `'print()'`.