

Lecture 24: Intro to Functions

Introduction to user-defined functions.
Understanding DRY (Don't Repeat Yourself).

Agenda

- What is a Function?
- Defining Functions in Python
- Function Syntax
- Calling a Function
- Return Values
- Docstrings
- Function Best Practices

What is a Function?

A function is a block of organized, reusable code that is used to perform a single, related action.

Functions provide better modularity for your application and a high degree of code reusing.

Python gives you many built-in functions like `print()`, etc., but you can also create your own functions (user-defined functions).

Why Use Functions?

DRY Principle: Avoid code duplication.

Abstraction: Hide complex implementation details.

Modularity: Break down large problems into smaller, manageable chunks.

Maintainability: Easier to fix bugs when code is organized.

Defining Functions

Use the 'def' keyword followed by the function name and parentheses '()'. Any input parameters should be placed within these parentheses. The code block within every function starts with a colon ':' and is indented.

Calling a Function

To execute the function, simply type its name followed by parentheses. Function calls can be placed anywhere in the code after the function is defined.

Return Statement

The 'return' statement exits a function, optionally passing back an expression to the caller.

A return statement with no arguments is the same as returning 'None'.

Returning Multiple Values

Python allows a function to return multiple values as a tuple.

Docstrings

Docstrings (documentation strings) describe what your function does. Enclosed in triple quotes `"""` immediately after the function header.

Function Stubs (pass)

Sometimes you want to define a function signature but implement it later. Use the 'pass' keyword to avoid syntax errors.

Summary

Functions modularize code and prevent repetition.
Defined with 'def', executed with '()'.
Can return values using 'return'.

Next Lecture

In the next lecture, we will dive deeper into Arguments and Parameters, exploring positional, keyword, and default arguments.