Parameters

Week 5

Function parameter and variables

- A **parameter** is a variable listed inside the parentheses in the function definition.
- A function must be called with the *correct* number of values: if your function has 2 parameters, you have to call the function with 2 values, not more, and not less.
- Parameter values come in by **position** within the parentheses (..) in the function call.
- The parameters must be given in this order, otherwise it does not make any sense.

The scope of the variables

- The variables and parameters in each function are ***independent***, isolated from those in other functions.
- By default, an `N` in one function is independent of an `N` in some other function.
- The same holds with variables that are passes as parameters. If their values are changed inside the function, that change does not change their values outside the functions.
- The reason is that values are **copied** to the parameters.

File: `add_one.py`

Parameters

- All values that a function needs from "outside" **must** be passed to it as parameters.
- It ensures that a function operates independently, relying solely on the information provided to it.
- By passing all necessary values as parameters, a function is independent and can be reused in different contexts.
- Functions that rely on parameters are easier to test because their behavior can be controlled and isolated.

Default parameters

- Default parameters in Python allow you to specify a default value for a function parameter.
- This means that if no parameter is provided for that parameter when the function is called, the default value will be used.

```python
def greet(name, greeting = "Hello"):
    return greeting + ", " + name + "!"

# Calling the function with both parameters
print(greet("Alice", "Hi"))  # Output: Hi, Alice!

# Calling the function with only the required parameter
print(greet("Bob"))  # Output: Hello, Bob!
```