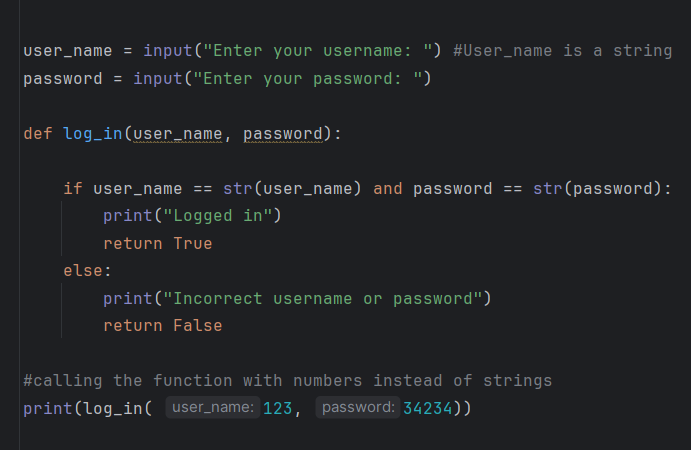
**Debugging functions using section 6.9**

In Section 6.9 of Think Python by Allen Downey, the author explains that when a function isn’t working as expected, we should check three main things:

1. **There is something wrong with the arguments the function is receiving** – This means the values passed into the function (inputs) don’t meet the expectations or requirements of the function. These are often called **preconditions**, which are conditions that must be true before a function is executed for it to work correctly.
2. **There is something wrong with the function itself** – This implies a logical or syntactical error exists within the body of the function. Even if the correct input is provided, if the code within the function is incorrect, the output will also be incorrect.
3. **There is something wrong with the return value or the way it is used** - This means the function might be returning the wrong value or the caller (code using the function) might be misinterpreting or misusing the return value. A **post-condition** is a condition that should be true after the function finishes executing if it has done its job properly.

Code example and explanation



1. Precondition Problem

The function expects strings (like “Merhawit” and “12345”) for username and password. But in the call log\_in(123, 34243), we give it numbers instead.

This is a precondition error because the input types are not what the function was built to handle.

1. **Function Logic Problem**

The logic inside the function is not really checking for correct login info.

if user\_name == str(user\_name) and password == str(password):

This line just checks if the inputs are equal to their string versions - which doesn’t help us verify if they’re valid.

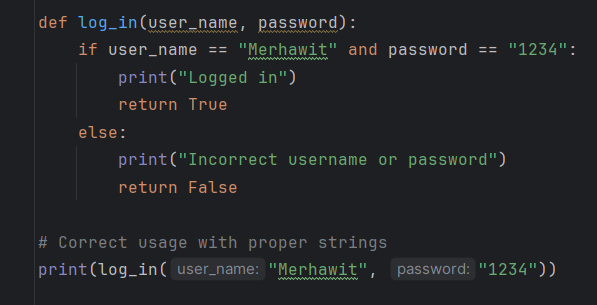
**Yes, this is a logic error**. The function should check if the input matches actual correct values, like:

if user\_name == "Merhawit" and password == "1234":

**3. Post-condition Problem (Wrong Use of Output)**

We print the result of the function using print(log\_in(123, 34234)).  
This is fine technically, but because the logic and input are already incorrect, the output is not helpful.

**Minor post-condition issue**, but the main problem is the logic and inputs.



Discussion question:

How can using type hints or input validation in Python help prevent precondition problems before they happen?