

Overview

In this lesson, you will learn how a function can send a value back to the rest of a program using `return`. You will also learn how `return` can be used to end a function early, even when no value is being sent back.

Important Information

So far, the functions you have written only *do* something, such as printing text. Sometimes, instead of printing a result, we want a function to **give a value back** so it can be used elsewhere in the program.

This is done using the `return` keyword.

When a function uses `return`:

- The function immediately stops running
- A value can be sent back to where the function was called
- The returned value can be stored in a variable

Returning a Value

Here is a simple example of a function that returns a value:

```
def get_number():
    return 7
```

When this function is called, it does not print anything. Instead, it sends the value `7` back to the line where it was called.

To use the returned value, we usually store it in a variable:

```
result = get_number()
print(result)
```

The variable `result` now holds the value that was returned by the function.

Return vs Print

It is important to understand the difference between `print` and `return`: * `print` displays something to the terminal * `return` sends a value back to the program

A function can return a value without printing anything, and a function can print something without returning a value.

Ending a Function Early

The `return` keyword can also be used **without a value**.

When `return` is used by itself the function stops running immediately, and no value is returned. This is useful when you want to exit a function early and skip the rest of the code.

Example:

```
def check_status():
    print("Checking status...")
    return
    print("This line will never run")
```

Once `return` is reached, the function ends, so any code after it is ignored.

Set Up

Create a new python file called `returnvalues.py`.

Copy, Change, Challenge - Return a Value

Copy

```
def get_score():
    return 100

score = get_score()
print(score)
```

Run the program and observe where the number comes from.

Change

Change the value being returned so that a different number is printed.

Challenge

Create a new function that returns a string instead of a number. Store the returned value in a variable and print it.

Copy, Change, Challenge - Return Early in a Function

Copy

```
def early_exit():
    print("Start of function")
    return
    print("End of function")

early_exit()
```

Change

Move the `return` statement so that both print statements run.

Challenge

Create a function that prints a message, then uses `return` to stop early based on your own logic. Add at least one line of code after the `return` that never runs.