# Overview

In this lesson, you will learn how to write functions that **input**, **process**, and **output** data. These three steps are often called **IPO**. Thinking about functions in terms of IPO helps make it clear what information goes into a function, what the function does with that information, and what comes back out.

# Important Information

Many problems in computer science can be broken down into three clear steps:

- **Input**: What information does the program need?
- **Process**: What work is done with that information?
- **Output**: What result is produced?

Functions are a natural way to organize code using IPO.

## IPO and Functions

When working with functions:

- **Input** is handled using function parameters
- **Process** is the code inside the function
- **Output** is handled using `return`

A function that follows IPO does not need to print anything. Instead, it returns a value so the rest of the program can decide what to do with the result.

## Example: IPO in a Function

```
def double_number(number):
    result = number * 2
    return result
```

Breaking this down:

- **Input**: `number`
- **Process**: `number * 2`
- **Output**: the returned value stored in `result`

When the function is called:

```
answer = double_number(5)
print(answer)
```

The function:

1. Receives `5` as input
2. Processes it by doubling it
3. Outputs `10` back to the program

## Why Return Is Important

Using `return` allows the output of the process to be stored as a variable to be used by other code. By returning a variable, the function can be tested to see if it produces the expected output. The `return` keyword on it's own doesn't provide any additional feedback to the user, it only provides data to the program itself.

When the user does require feedback, the program needs to `print()` the result. Printing is something the **main program** should usually handle, not the function itself.

## Multiple Steps of Processing

Functions can have more than one processing step before returning a value.

```
def calculate_score(points):
    bonus = 5
    total = points + bonus
    return total
```

IPO breakdown:

- **Input**: `points`
- **Process**: add a bonus
- **Output**: the final total

## Using Returned Values

Returned values can be stored in a variable, used again later, or printed.

```
score = calculate_score(10)
final_score = score * 2
print(final_score)
```

The function does the processing, and the rest of the program decides what to do with the output.

# Set Up

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

# Copy, Change, Challenge - Square a Number

## Copy

```
def square(number):
    result = number * number
    return result

answer = square(4)
print(answer)
```

Identify the input, process, and output in this function.

## Change

Change the function so it cubes the number instead of squaring it.

## Challenge

Create a function that takes a number as input, adds 10 to it, and returns the result. Store the returned value in a variable and print it.

# Copy, Change, Challenge - Converting Numbers

## Copy

```
def convert_minutes(minutes):
    seconds = minutes * 60
    return seconds

time_in_seconds = convert_minutes(3)
print(time_in_seconds)
```

## Change

Change the input value so the output is different.

## Challenge

Write a new function that:

- Takes one number as degrees celsius as input
- Converts the number to fahrenheit
- Returns the final result