

# What are Functions?

This section provides an in-depth look into the most powerful feature of Python programming: functions.

## We'll cover the following



- Definition
  - Why Use Functions?
- Types of Functions in Python

## Definition #

A function is a reusable set of operations.

That sounds like a pretty straightforward definition. But what does it exactly mean?

Let's take a step back. Remember the `print()` and `len()` statement? Both always perform predefined tasks. Well, it turns out they were functions all along!

## Why Use Functions? #

Think of a function as a box which performs a task. We give it an input and it returns an output.

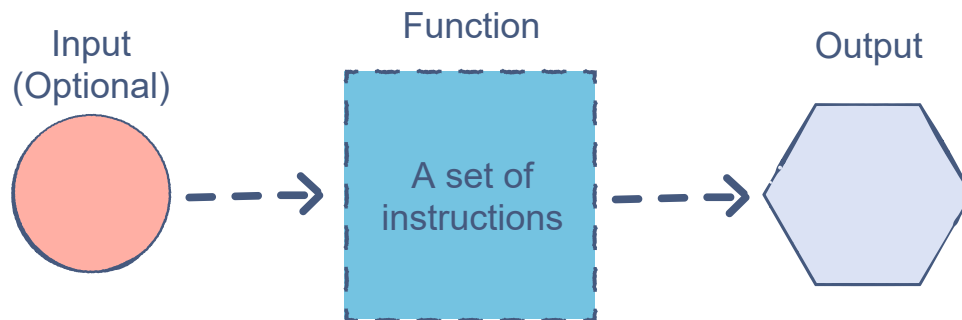
We don't need to write the set of instructions again for a different input, we could just call the function again.

Functions are useful because they make the code concise and simple. The primary benefits of using functions are:

- **Reusability:** A function can be used over and over again. You do not have to write redundant code. For example, a `sum()` function could compute the sum of all the integers we provide it. We won't have to write the summing operation ourselves each time.

- **Simplicity:** Functions are easy to use and make the code readable. We only need to know the inputs and the purpose of the function without focusing on the inner workings. This abstraction allows us to focus more on gaining the output instead of figuring out how it was computed.

An input isn't even necessary. A function could perform its own computations to complete a task.



Suppose we want to find the smaller value between two integers:

```
num1 = 10
num2 = 40
if num1 < num2:
    minimum = num1
else:
    minimum = num2
print(minimum)

num1 = 250
num2 = 120
if num1 < num2:
    minimum = num1
else:
    minimum = num2
print(minimum)

num1 = 100
num2 = 100
if num1 < num2:
    minimum = num1
else:
    minimum = num2
print(minimum)
```



For every new pair of integers, we need to write the `if-else` statement again.

All this could become much simpler if we had a function to perform the steps necessary for calculating the minimum.

The good news is that Python already has the `min()` function:

```
minimum = min(10, 40)
print(minimum)

minimum = min(10, 100, 1, 1000) # It even works with multiple arguments
print(minimum)

minimum = min("Superman", "Batman") # And with different data types
print(minimum)
```



Now the code looks pretty awesome. Plus, it's easier to write. And that is the beauty of functions!

## Types of Functions in Python #

Functions are perhaps the most commonly used feature of Python. There are two basic types of functions in Python:

1. Built-in functions
2. User-defined functions

We've already seen some instances of built-in function such as `len()`, `min()`, and `print()`. We'll see more of these in the future.

The coolest feature, however, is that the language allows us to create our own functions that perform the tasks we require.

In the coming lessons, we'll explore functions in detail, after which you'll be able to create and manipulate functions with ease!

---

Let's get started by creating our first function.