Functions

◎ Objectives

At the end of this lesson, you should be able to:

- **define** the concept of function
- **recognise** functions in code
- **implement** functions to extend your code

1. Introduction

In the code you have been writing so far, you probably noticed that writing the same pieces of code at different places is **tedious** and sub-optimal. We have seen that **loops** are one way to avoid repeating code, but we need to find another way to do so, so the same instructions can be reused in different contexts.

Functions are used to do just that. They allow to set aside groups of instructions to be used at different parts of the program.

Idea

The main idea is: define **once**, use **everywhere**

2. Function definition

To **define** a function, Python gives us the **`def`** keyword[1]. In Python, and most programming languages, a function is a special object composed of:

- a name
- a sequence of instructions

Optionnally, functions can also have:

- one (or more) parameters
- a return value

</> Code

```
def say_hello():
  print("Hello!")
```

>_ Output

```
>>> say_hello()
Hello!
```

[1] awfully adequate, don't you think?

With parameters, the function definition looks like this:

</> Code

```
def say_hello(name):
  print(f"Hello, {name}")
```

>_ Output

```
>>> say_hello("Bob")
Hello, Bob
```

3. Transforming values

Functions can also be used to **create** new values or **transform** existing ones. To do this, we can use the **`return`** keyword.

Information

Careful! The **`return`** keyword stops the execution of the function

A couple of examples are presented below.

</> Code

```
def create_player(name, player_class):
  return {"name": name, "class": player_class, "xp": 0}
```

>_ Output

```
>>> create_player("Ned Stark", "Warden")
{"name": "Ned Stark", "class": "Warden", "xp": 0}
```

</> Code

```
def add_two(number):
  number = number + 2
  return number
```

>_ Output

```
>>> add_two(14)
16
```

Idea

Most of the time, when you create or use a function that returns a value, you want to **store** it in a variable so it can be used.