Dicts

◎ Objectives

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

- **describe** the concept of dictionnary
- **manipulate** dicts to represent data
- **construct** dicts from real-life situations

1. Context

Imagine you're a teacher and you are grading students. You want to be able to store the information about the grades in a (or more) variables, but you also want to be able to **keep** the information relating to the student available.

</> Code

```
students    = ["Alice", "Bob", "Charlie"]
first_test  = [12, 8, 15]
second_test = [17, 15, 20]
```

As you can see, this is not very practical, as the lists keep on being created and quickly get out of control.

2. Dictionnaries

Idea

Python introduces another data type, used to represent **structured** data: these are called **`dict`**

</> Code

```
example = {"first": 12, "condition":False}
```

As you can see in the example above, you can create a list just like any other variable. The notable difference is that the elements are enclosed by **square brackets** **`[`** and separated by **commas** **`,`**. The elements are each key-value pairs.

2.1 Accessing an element

You can access an element in a dict by using its key.

</> Code

```
>>> print(student["name"])
Alice
>>> print(student["first_grade"])
12
```

2.2 Adding an element

You can add elements by adding a key with its value.

</> Code

```
student["third_grade"] = 7
```

3. Useful functions

Here are a couple of functions that can be useful while using dicts.

</> Code

```
# Get all the keys in a dict
d.keys()
# Get only the values
d.values()
# Get the number of keys
len(d)
# Find out if a key is in a dict
key in d
```