Sentinel

Python Workshop: Dictionaries

DEFENDING OUR DIGITAL WAY OF LIFE

Introduction

Let’s say that we have a list of students, and we want to save the grade in Python of each student.

How would you do that?

Is there a better way?

Introduction

Dictionary is a data type, like list.

It maps variables to other variables.

Example:

```
students_dict = {"Mark": 80, "David": 90, "Laura": 65,
"Johanna": 100}
```

The thing we look up (the “word”) is called a *key*.

What we find by it (the “definition”) is called a *value*.

Adding keys to the dictionary

Adding keys to the dictionary looks like this:

```
>>> students_grades_dict = {}
>>> students_grades_dict[“Mark”] = 80
>>> students_grades_dict[“Johanna”] = 95
>>> print(students_grades_dict)
{“Mark”: 80, “Johanna”: 95}
```

Looking up values

Looking up values looks like this:

```
>>> print(students_grades_dict[“Mark”])
80
>>> print(students_grades_dict[“Johnna”])
95
```

Replacing and deleting values

You can update an existing key with a new value:

```
>>> print(students_grades_dict[“Mark”])
80
>>> students_grades_dict[“Mark”] = 95
>>> print(students_grades_dict[“Mark”])
95
```

You can delete existing keys:

```
del students_grades_dict[“Mark”]
```

Looking up values - get

You can also use the “get” function to look up values:

```
>>> print(students_grades_dict.get(“Mark”))
80
>>> print(students_grades_dict.get(“Johnna”))
95
```

When using “get”, there is no error when a key is not found.

What did we learn?

Dictionaries, and why we need them

Adding, updating and removing values

Practice time!

Do the first exercises in Dictionaries.

Solution

- Let’s solve together the first Dictionaries exercise.
- There are 2 solutions - one using “has_key”, one using “get”.
- If you need more dictionary practice, make sure you solve the “Contacts” exercise.