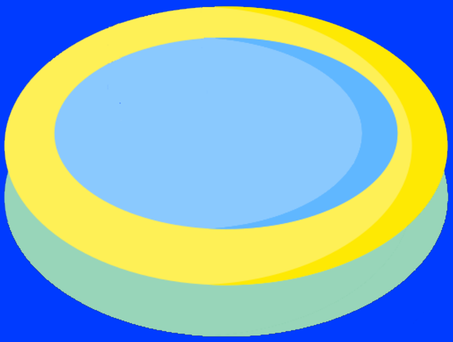
{EPITECH}

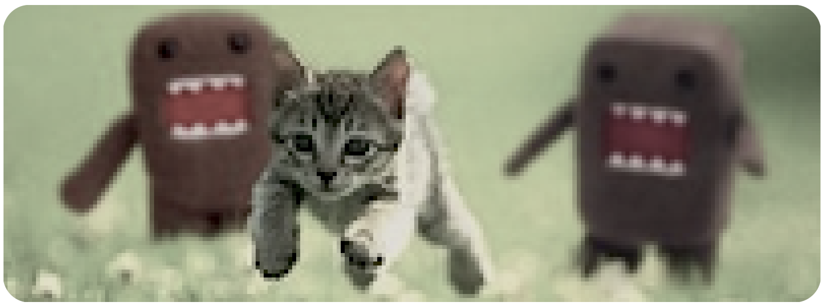
WADING POOL_

< 06 - DICTIONARIES + BOOLEANS />



WADING POOL

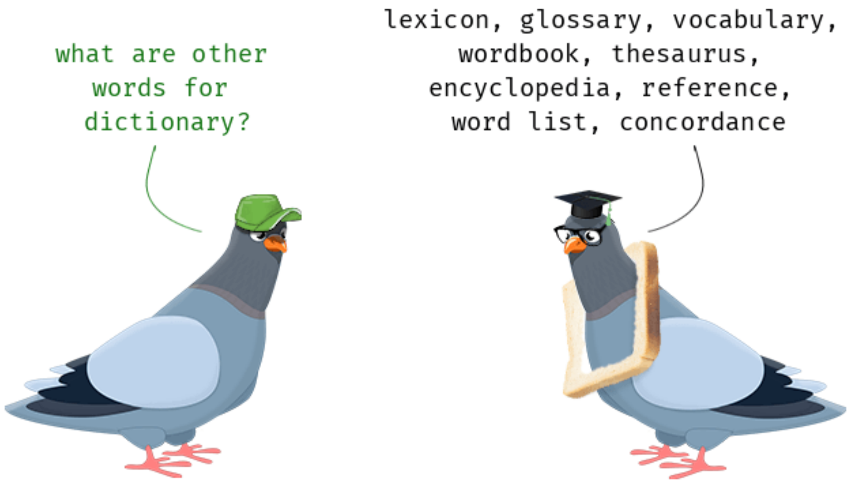
If you need a little break in your day, we suggest you to play the bandit wargame.
We also encourage you to resume your game later.



Dictionaries

Task 1.1

Create a **dictionary** stored in a `pokemons` variable. A dictionary must contain some key/value pair(s).
The keys are strings representing the names of some pokemons: `Pikachu`, `Bulbausaur` and `Charmander`.
The values are strings representing their respective type: `Electric`, `Grass` and `Fire`.
Eventually, print this dictionary.


what are other words for dictionary?
lexicon, glossary, vocabulary, wordbook, thesaurus, encyclopedia, reference, word list, concordance

Task 1.2

Inside `pokemons`, add the key `Blaziken` with the value `Fire`. Then, print this dictionary.

Task 1.3

In your dictionary `pokemons`, add the key `Pikachu` with the value `["Pichu", "Raichu"]`.
Look at what is happening. What do you observe? What do you make of it?

Task 1.4

Let's do it again, but differently. Start by creating the dictionary `types`.
The keys are strings representing Pokémon types (e.g., 'Electric', 'Grass', 'Fire').
For each key, add an **empty** list as value (for now). Finally, print the dictionary.

Task 1.5

In the values/lists of your dictionary `types`, add relevantly each Pokemon from this list [”Pikachu”, ”Bulbausaur”, ”Charmander”, ”Leafeaon”, ”Scovillain”] .

Find out by yourself the type(s) of each Pokemon.

Task 1.6

Print all the keys (Pokemon types) contained in your previous dictionary.

Task 1.7

Using your previous dictionary, retrieve the type of ”Pikachu”.

Try to do this in two different ways.

Task 1.8

Store this dictionary into the variable `superheroes`.
Then, print the value of `Superman`'s `city`.

```
{
  "Batman" : {
    "id": 1,
    "aliases": ["Bruce Wayne", "Dark knight"],
    "location": {
      "number" : 1007,
      "street": "Mountain Drive",
      "city": "Gotham"
      }
  },
  "Superman" : {
    "id": 2,
    "aliases": ["Kal-El", "Clark Kent", "The Man of Steel"],
    "location": {
      "number" : 344,
      "street": "Clinton Street",
      "apartment": "3D",
      "city": "Metropolis"
      }
  },
}
```

Task 1.9

Inside the dictionary `superheroes`, add:

✓ `Caped Crusader` inside `Batman`'s `aliases` ;
✓ a new superhero `Wolverine`, with 3 as `id`, and no aliases nor location.

Task 1.10

For each superhero contained in `superheroes`, enumerate all the aliases she/he has.
Your result should look like this:

```
Batman:
Bruce Wayne
Dark knight
Caped Crusader

Superman:
Kal-El
Clark Kent
The Man

Wolverine:
No aliases found.
```

Task 1.11

Inside this dictionary, get the key with the maximum value:

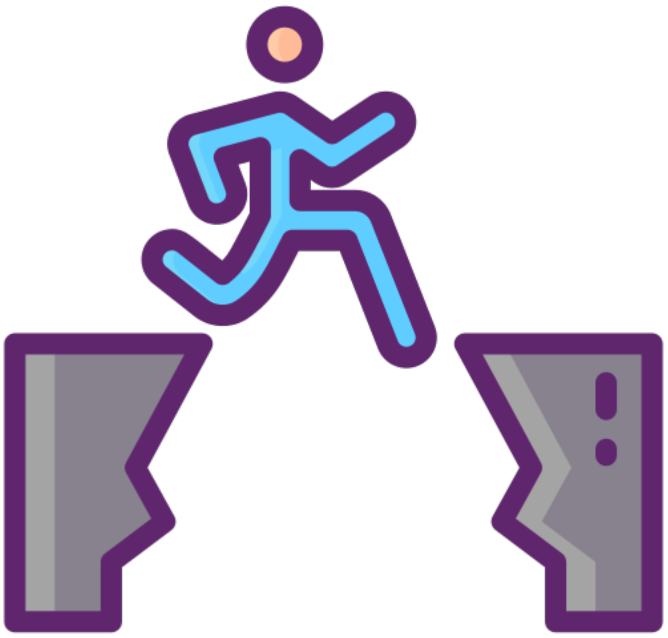
```
{
  "dalmatians": 101,
  "pi": 3.14,
  "beast": 666,
  "life": 42,
  "googol": 10^100,
  "jordan": 23,
  "life, the universe and everything": 42,
  "emergency": 911,
  "euler": 2.71828
}
```

CHALLENGE

Let's play Scrabble.
Without any bonus, the score for a word is calculated based on the letter values as follows:

- ✓ A, E, I, O, U, L, N, S, T, R = 1 point each ;
- ✓ D, G = 2 points each ;
- ✓ B, C, M, P = 3 points each ;
- ✓ F, H, V, W, Y = 4 points each ;
- ✓ K = 5 points ;
- ✓ J, X = 8 points each ;
- ✓ Q, Z = 10 points each.

Write a Python function that takes a string as input and returns the base score for that word by summing the scores of its letters.



If you do not feel comfortable now, have a look at the next exercises, then come back here.
Using Python built-in functions will make your solution as clean and efficient as possible.
And it should take very few lines of code.

Booleans

Task 2.1

Create two **boolean** variables: `is_sunny` and `is_raining`.
Set `is_sunny` value to `True` and `is_raining` to `False`.
Print the values of both variables.

Task 2.2

Think about what this code could do. Then, run it to check if your guess is correct.

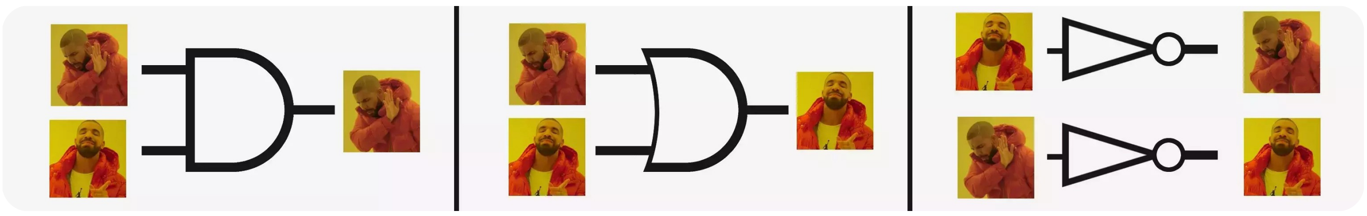
```
is_sunny = True
is_snowing = not is_sunny
```

Task 2.3

Below are four lists of two booleans.

```
l1 = [True, True]
l2 = [False, False]
l3 = [True, False]
l4 = [False, True]
```

Write a first snippet of code to check if both values are True, in each list.
Write a second snippet of code to check if one of the value is True, in each list.



Task 2.4

Write some code to check if either one value or the other is True, but not both values, in each list.

Task 2.5

Create three boolean variables: `is_raining`, `is_umbrella` and `at_home`.
Set their respective values to `True`, `False` and `True`.

Use some **boolean operators** to check if it is either raining and you have an umbrella, or you are at home.
Print the result of this condition.

v 2.2