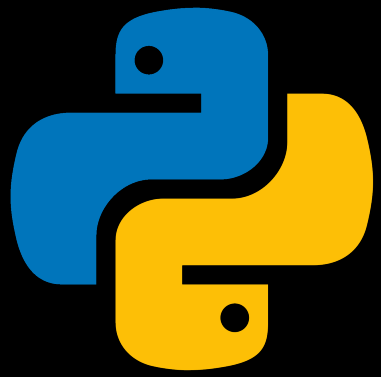

CLASSES IN PYTHON

Realised by **Azddine Elmoumny**
Supervised by ***Pr.*** **Rajae Tamri**

OUTLINE

1. Reminder 🤦‍♂️🤦‍♀️

- `list` →

```
fruits=["🍋","🍍","🍉","🍓"]
```

- `tuple` →

```
fruits=("🍋","🍍","🍉","🍓")
```

- `dict` →

```
item={"name":"🍓","type":"fruit","available": True}
```

2. Challenge 💪💪

Create a program to manage a library's inventory . The program should allow users to add books remove books, search for books by title or author, and display the available books in the inventory. Additionally, the program should support multiple libraries, each with its own inventory.

3.  Classes

In object-oriented programming (OOP), a class is a structure that combines related data (attributes) and functionality (methods). It serves as a blueprint to create specific objects. Objects created from a class are called instances. Classes enable encapsulation, abstraction, and modularity of code. They facilitate code reuse, maintenance, and management of entities within a program.

3.1. Syntax :

```python
class Car: # Substitue Car with whatever name you want
    def __init__(self, make, model, year):
        self.make = make
        self.model = model
        self.year = year

    def start_engine(self):
        print(f"The {self.make} {self.model}'s engine has started.")

    def stop_engine(self):
        print(f"The {self.make} {self.model}'s engine has stopped.")
```

3.2. Boook Class 📚📚 :

```
class Book:
    def __init__(self, title, author):
        self.title = title
        self.author = author

    def __str__(self):
        return f"{self.title} by {self.author}"
```