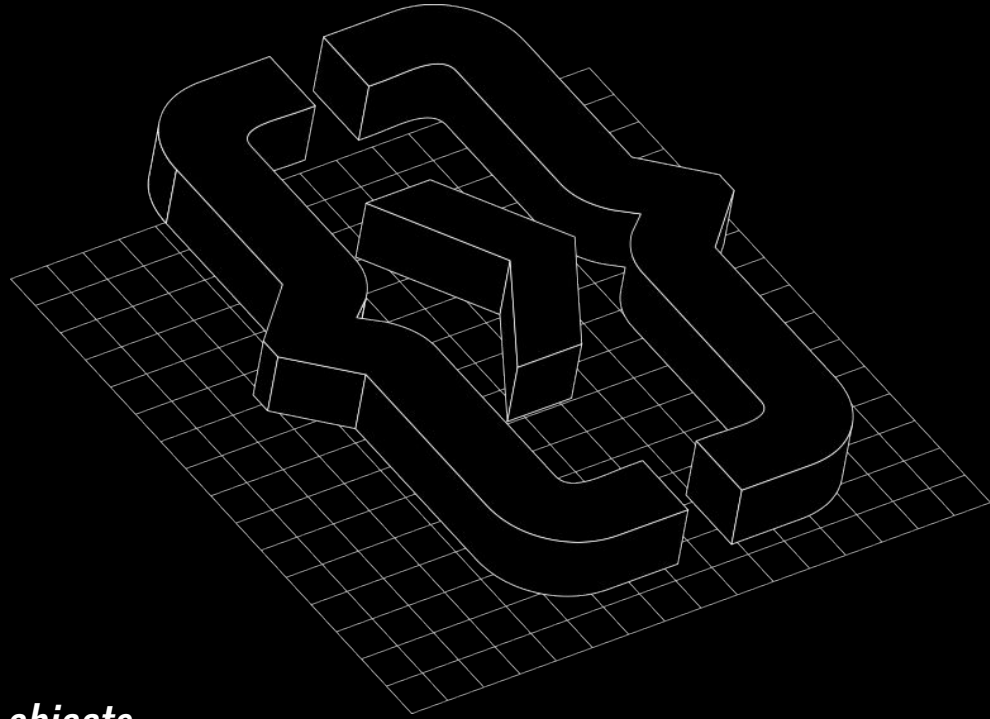


Welcome to

# GOMYC{>}DE



*Workshop 5: Classes & objects*  
*Instructor : Nihel Hacine*



## • Plan

- Programmation orientée objet
- Les classes
- Les objets
- Les méthodes
- L'héritage



## Programmation orientée objet

La POO est une méthode de structuration d'un programme en regroupant les propriétés et les comportements associés dans des objets individuels. Son objectif est de faciliter l'écriture et la maintenance du code.

Python est un langage de programmation multi-paradigmes, ce qui signifie qu'il prend en charge différentes approches de programmation tel que la POO



## Les classes

Une classe est comme un constructeur d'objets ou un « plan » pour créer des objets.

```
class MyClass:  
    x = 10
```



## Les objets

Un objet est l'unité de base de la programmation orientée objet. Un exemple simple d'objet serait une personne. Logiquement, on s'attendrait à ce qu'une personne ait un nom, prénom (propriétés)...

En POO, l'objet est considéré comme une instantiation d'une classe. Lorsque la classe est définie, seule la description de l'objet est définie.

Par conséquent, aucune mémoire ou stockage n'est alloué.



## Les objets

```
#créer la classe
```

```
class MyClass:
```

```
    x = 10;
```

```
#instancier un objet
```

```
myobject=MyClass();
```

```
print(myobject.x);
```



## La fonction `__init__()`

```
class Person:
    def __init__(self, name,
age):
    self.name = name
    self.age = age
```

```
p1 = Person("John", 36)

print(p1.name)

print(p1.age)
```



# Les méthodes

```
class Person:

    def __init__(self, name, age):

        self.name = name

        self.age = age

    #ceci est une méthode

    def affiche(self):

        print("Hello my name is " + self.name)
```

```
p1 = Person("John", 36)

p1.affiche()
```





## L'héritage

L'héritage est un moyen de créer une nouvelle classe en utilisant les détails d'une classe existante sans la modifier. La classe nouvellement formée est une classe dérivée (ou classe enfant). De même, la classe existante est une classe de base (ou classe parent).



Merci !