

Structure de données

Ensemble d'objets organisés qui sont liés

Informations structurées

Interface : définition théorique d'une structure de données

Implémentation : mise en œuvre concrète, pratique de l'interface dans un langage de programmation en particulier

Interface	Exemple d'implémentation
type abstrait liste	→ liste chaînée (python)
Classe en POO	→ Définition d'une classe (python)
Schéma d'une base de données relationnels	→ Table SQL
Le plan de votre maison de rêve	→ Le plan de l'architecte avec le choix des matériaux

Informations brutes	
↓	
Objet	Couleur
Radiateur	bleu
Livre	rouge
Tableau	blanc
Données structurées	

Programmation orientée objet (POO) :

- Basé sur les concepts de classe, attribut, méthode et objet
- Paradigme de programmation

Autres paradigmes : programmation impérative, programmation fonctionnelle

	Définition	Métaphore culinaire	Métaphore architecturale
Classe	Définition des caractéristiques d'un ensemble d'objets et des fonctions associées	Le moule et la recette	Plan d'un studio
Objet	Représente un élément du monde physique ou un concept. Il est caractérisé par une structure interne (attributs) et un comportement (méthodes).	Un gâteau	Réalisation d'un studio
Attribut	Paramètres d'un objet de la classe	Poid, temps cuisson...	Taille, région...
Méthode	Fonction associée à une classe	Découper, ajouter une bougie...	Louer, estimer la valeur...

POO avec python

Docstring (aide utilisateur)

Constructeur

Méthodes

Création d'un objet

Utilisation des méthodes

```
1 class Pays:
2     """
3     La classe Pays définit des objets caractérisés par les attributs nom (string),
4     drapeau (chemin d'un fichier image), continent (string) et roi (booléen) et
5     par les méthodes __init__, __repr__ et est_en_afrique.
6     """
7     def __init__(self, nom, drapeau, continent, roi):
8         self.nom = nom
9         self.drapeau = drapeau
10        self.continent = continent
11        self.roi = roi
12
13    def __repr__(self):
14        if self.roi:
15            return self.nom + " est un pays (" + self.continent
16            + ") qui est dirigé par un roi."
17        else:
18            return self.nom + " est un pays (" + self.continent
19            + ") qui n'est pas dirigé par un roi."
20
21    def est_en_afrique(self):
22        return self.continent == "Afrique"
23
24 tokelau = Pays("Tokelau", "Océanie", "drapeau_tokelau.png", True)
25 print(tokelau)
26 print(tokelau.est_en_afrique())
```

Attributs

objet.méthode()