

Méthode

- Accès à travers une instance de la classe (un objet).
- La méthode a accès à toutes les propriétés de l'instance via `self`.
- La méthode peut accéder aux propriétés de classe via `self.__class__`.

Exemple :

```
class MaClasse:
    def une_methode(self):
        self.propriete = 42

objet = MaClasse()
objet.une_methode()
print(objet.propriete)
```

Méthode de classe (décorateur `@classmethod`)

- Accès à travers la classe elle-même, pas besoin d'instancier la classe.
- La méthode a accès aux propriétés de classe mais pas aux propriétés d'instance.
- Peut être utilisée pour créer des "usines" d'objets ou des méthodes liées à la classe plutôt qu'à une instance particulière.

Exemple :

```
class MaClasse:
    propriete_de_classe = 42

    @classmethod
    def une_methode_de_classe(cls):
        cls.propriete_de_classe = 43

MaClasse.une_methode_de_classe()
print(MaClasse.propriete_de_classe)
```

Méthode statique (décorateur `@staticmethod`)

- Accès à travers la classe elle-même, pas besoin d'instancier la classe.
- La méthode n'a pas accès aux propriétés de classe ni aux propriétés d'instance.

- Utilisée pour des fonctions liées à la classe mais qui n'ont pas besoin d'accéder à des données spécifiques à l'instance ou à la classe.

Exemple :

```
class MaClasse:
    @staticmethod
    def une_methode_statique():
        print("Ceci est une méthode statique")

MaClasse.une_methode_statique()
```

Accès

Actions / Types de Méthodes	Méthode	Méthode de Classe	Méthode Statique
une instance	objet.methode()	-	-
propriétés de l'instance	self.propriete	-	-
propriétés de classe	cls.propriete	Classe.propriete	-
"usines" d'objets	-	Classe.factory()	-
propriétés de la classe	self.propriete_classe	cls.propriete_classe	-
propriétés d'instance	self.propriete_instance	-	-
fonctions utilitaires	-	-	Fonction.utilitaire()