Python

SOCOMEC

OCTOBRE 2025

Retour sur vos questions du deuxième jour

Propriété de B.S. Reproduction complète ou partielle sans autorisation préalable interdite.

Les MixIns

- Un mixin est une classe qui fournit des méthodes utilitaires ou un comportement spécifique à être utilisé avec d'autres classes.
- Elle n'est pas destinée à être instanciée seule, seulement à être héritée pour ajouter des fonctionnalités.
- Principe : composition par héritage léger, « a little inheritance, just for behavior ».

Propriété de B.S. Reproduction complète ou partielle sans autorisation préalable interdite.

Les MixIns

- Pas de constructeur complexe (__init__)
- Méthodes spécifiques et réutilisables
- Peut être combiné avec d’autres classes via l’héritage multiple
- Ne définit pas de hiérarchie métier principale, juste des fonctionnalités additionnelles

Propriété de B.S. Reproduction complète ou partielle sans autorisation préalable interdite.

Les MixIns

Exemple simple

```
class JsonMixin:
    import json
    def to_json(self):
        return self.json.dumps(self.__dict__)

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

# On combine la classe principale et le mixin
class PersonJson(Person, JsonMixin):
    pass

p = PersonJson("Alice", 30)
print(p.to_json())  # {"name": "Alice", "age": 30}
```


Propriété de B.S. Reproduction complète ou partielle sans autorisation préalable interdite.

Les MixIns

Exemple plus complexe

```
class ReprMixin:
    def __repr__(self):
        return f"<{self.__class__.__name__}: {self.__dict__}>"

class Person(Person, JsonMixin, ReprMixin):
    pass

p = Person("Bob", 25)
print(p)          # <Person: {'name': 'Bob', 'age': 25}>
print(p.to_json()) # {"name": "Bob", "age": 25}
```


Propriété de B.S. Reproduction complète ou partielle sans autorisation préalable interdite.

Les MixIns

Avantages

- Réutilisation facile du code
- Composition flexible sans créer une hiérarchie complexe
- Découplage des fonctionnalités (chaque mixin a une seule responsabilité)
- Favorise les bonnes pratiques OOP : une classe = une responsabilité, les fonctionnalités communes = mixins

Propriété de B.S. Reproduction complète ou partielle sans autorisation préalable interdite.

Les MixIns

Bonnes pratiques

- N’utiliser les mixins que pour des comportements transversaux
- Éviter de mettre trop de hiérarchie ou de mixins dépendants entre eux, cela peut devenir confus
- Documenter clairement que la classe est un mixin et ne doit pas être instanciée seule

Propriété de B.S. Reproduction complète ou partielle sans autorisation préalable interdite.