



```
render() {  
  return (  
    <React.Fragment>  
      <div className="py-5">  
        <div className="container">  
          <Title name="our" title="product">  
            <div className="row">  
              <ProductConsumer>  
                {(value) => {  
                  console.log(value)  
                }}  
              </ProductConsumer>  
            </div>  
          </div>  
        </div>  
      </React.Fragment>  
    )  
  )  
}
```

Programmation Orientée Objet

Laffet takwa

Qu'est-ce qu'une classe ?



une classe est comme un plan ou un modèle.

➔ Exemple :

Si tu veux créer plusieurs voitures dans ton programme,

tu vas d'abord créer une classe "Voiture" (le modèle).

Ensuite, tu pourras créer plusieurs objets (ex : `ma_voiture`, `ta_voiture`, etc.).

```
class Voiture:  
    pass
```

Ici, `pass` veut dire : "ne rien faire pour le moment".

Créer un objet à partir d'une classe

- Un objet est une chose concrète créée à partir de la classe.
- Le code crée un objet nommé `ma_voiture` à partir du plan `Voiture`.

```
class Voiture:  
    pass  
  
# Création d'un objet  
ma_voiture = Voiture()  
  
print(ma_voiture)
```

Le constructeur `__init__`

Quand on crée un objet, on veut souvent lui donner des informations dès le départ.

Pour cela, on utilise une fonction spéciale dans la classe : `__init__`.

```
class Voiture:
    def __init__(self, marque, couleur):
        self.marque = marque
        self.couleur = couleur
```

- `__init__` : c'est le constructeur, il s'exécute automatiquement quand on crée un objet.
- `self` : c'est l'objet lui-même (on y reviendra souvent).
- `self.marque` et `self.couleur` sont des attributs (informations propres à la voiture).

Les attributs

Les attributs sont les données ou informations d'un objet.

```
class Etudiant:
    def __init__(self, nom, age):
        self.nom = nom
        self.age = age

e1 = Etudiant("Ali", 20)
e2 = Etudiant("Sara", 19)

print(e1.nom, e1.age)  # Ali 20
print(e2.nom, e2.age)  # Sara 19
```

Chaque objet a ses propres valeurs.
Ali et Sara sont deux étudiants différents.

Les méthodes

```
class Etudiant:  
    def __init__(self, nom, age):  
        self.nom = nom  
        self.age = age  
  
    def se_presenter(self):  
        print(f"Bonjour, je m'appelle {self.nom} et j'ai {self.age} ans.")
```

Les méthodes sont des fonctions à l'intérieur d'une classe.

Elles permettent de faire des actions.

```
e1 = Etudiant("Ali", 20)  
e1.se_presenter()  # Bonjour, je m'appelle Ali et j'ai 20 ans.
```

- `def se_presenter(self):` → méthode
- `self.nom` → va chercher le nom de cet étudiant
- `e1.se_presenter()` → exécute la méthode pour `e1`

Exercices

Crée une classe CompteBancaire avec :

- attributs : nom et solde
- méthodes :
 - afficher_solde()
 - deposer(montant)
 - retirer(montant)



Thank You