

la Programmation Orientée Objet en PHP

1. Introduction à la Programmation Orientée Objet (POO) en PHP

La programmation orientée objet (POO) est une manière d'écrire du code en utilisant des **objets**.

Un objet représente une chose réelle ou logique, comme un utilisateur, une voiture ou un compte.

1.1 Classe

Une **classe** est un modèle (un plan) qui décrit les caractéristiques et les comportements d'un objet.

Par exemple:

```
class Voiture {  
  
    public $couleur;  
  
    public $marque;  
  
}
```

1.2 Objet

Un objet est une instance créée à partir d'une classe.

Par exemple:

```
$maVoiture = new Voiture();  
$maVoiture->couleur = "Rouge";  
$maVoiture->marque = "BMW";
```

2. Encapsulation et Modificateurs d'Accès

2.1 Encapsulation

L'encapsulation permet de **protéger les données** d'un objet.

Elle évite l'accès direct à certaines informations sensibles.

2.2 Modificateurs d'accès en PHP

- **public** : accessible partout
- **private** : accessible uniquement dans la classe
- **protected** : accessible dans la classe et ses classes filles

3. Héritage et Polymorphisme

3.1 Héritage

L'héritage permet à une classe de **reprendre les propriétés et méthodes** d'une autre classe.

Par exemple :

```
class Animal {  
  
    public function parler() {  
  
        echo "L'animal fait un bruit";  
  
    }  
  
}  
  
class Chien extends Animal {  
  
    public function parler() {  
  
        echo "Le chien aboie";  
  
    }  
  
}
```

3.2 Polymorphisme

Le polymorphisme permet d'utiliser une même méthode avec des comportements différents selon l'objet.

Par exemple (se basons sur l'exemple d'héritage) :

```
$animal1 = new Animal();
```

```
$animal2 = new Chien();
```

```
$animal1->parler();
```

```
$animal2->parler();
```