

La POO en PHP

Introduction :

La programmation orientée objet (POO) est un paradigme de programmation qui repose sur la création et l'interaction d'objets. En PHP, la POO permet de structurer le code de manière modulaire et réutilisable, ce qui facilite la maintenance et l'évolution des applications.

Les Classes:

Une classe est un modèle ou un plan qui définit les propriétés (variables) et les comportements (méthodes) d'un objet.

```
1 class Voiture
2 {
3     public $marque;
4     public $couleur;
5
6     public function rouler()
7     {
8         echo "La voiture roule.";
9     }
10 }
```

Les Objet :

Un objet est une instance d'une classe.

Il est créé en utilisant le mot-clé **new**.

```
1 $maVoiture = new Voiture();
2 $maVoiture->marque = "Toyota";
3 $maVoiture->couleur = "Rouge";
4 echo $maVoiture->marque;
5 // Affiche "Toyota"
6 $maVoiture->rouler();
7 // Affiche "La voiture roule."
```

Caractéristiques des Objets en PHP :

- **Propriétés :**
Les variables définies dans une classe qui décrivent l'état ou les attributs de l'objet. Par exemple, **\$marque** et **\$couleur** dans la classe Voiture.
- **Méthodes :**
Les fonctions définies dans une classe qui décrivent le comportement ou les actions que l'objet peut effectuer. Par exemple, la méthode **rouler()**.
- **Utilisation du Mot-Clé \$this :**
Le mot-clé **\$this** fait référence à l'instance actuelle de la classe

Encapsulation et Modificateurs d'Accès :

L'encapsulation, c'est l'idée de ranger ensemble les données (propriétés) et les actions (méthodes) qui les utilisent dans une même boîte appelée "classe". Cela permet de protéger les données en contrôlant qui peut les voir ou les modifier, et de cacher les détails inutiles pour rendre le code plus sûr et facile à utiliser.

Les Modificateurs d'Accès en PHP :



```
1 protected $propriete;
2 protected function methode() { }
```

```
1 private $propriete;
2 private function methode() { }
```

```
1 public $propriete;
2 public function methode() { }
```

- **protected :**
Permet l'accès à la propriété ou méthode dans la classe où elle est définie et dans les classes qui en héritent.
- **private :**
Limite l'accès à la propriété ou méthode à l'intérieur de la classe où elle est définie.
- **public :**
Permet d'accéder à la propriété ou méthode depuis n'importe où.

Avantages de l'Encapsulation :

- **Protection des Données :**
Les données sensibles peuvent être protégées contre les modifications non autorisées.
- **Contrôle de l'Accès :**
Les développeurs peuvent définir comment les données sont lues ou modifiées via des méthodes spécifiques (getters et setters).
- **Facilité de Maintenance :**
En isolant les données, toute modification du fonctionnement interne n'a pas d'impact sur le code externe qui utilise la classe.

Héritage en PHP :

L'héritage est un concept de la POO qui permet à une classe (appelée classe fille ou classe dérivée) de réutiliser et d'étendre les fonctionnalités d'une autre classe (appelée classe parent).

- Pour déclarer une classe fille en PHP, on utilise le mot-clé **extends**.
- La classe fille hérite des propriétés et méthodes publiques et protégées de la classe parent.

```
1 class ParentClass
2 {
3     public function methodeParent()
4     {
5         echo "Méthode de la classe parent";
6     }
7 }
8
9 class EnfantClass extends ParentClass
10 {
11     public function methodeEnfant()
12     {
13         echo "Méthode de la classe enfant";
14     }
15 }
```

Polymorphisme en PHP :

Le polymorphisme est la capacité d'une méthode d'avoir des comportements différents en fonction de la classe qui l'implémente. En PHP, cela se fait souvent par la redéfinition des méthodes (méthodes override).

- La classe fille peut redéfinir une méthode de la classe parent avec le même nom.
- L'utilisation des interfaces ou des classes abstraites permet également d'exploiter le polymorphisme.

```
1 class ParentClass {
2     public function afficherMessage() {
3         echo "Message depuis la classe parent";
4     }
5 }
6
7 class EnfantClass extends ParentClass {
8     public function afficherMessage() {
9         echo "Message depuis la classe enfant";
10    }
11 }
12
13 $objet = new EnfantClass();
14 $objet->afficherMessage(); // Affiche "Message depuis la classe enfant"
```

Résumer :

1. Fondements de la POO : Classes et Objets

- Une classe est une structure qui regroupe des données (propriétés) et des actions (méthodes).
- Un objet est une instance d'une classe créée avec le mot-clé **new**.

2. Encapsulation et Modificateurs d'Accès

- **Encapsulation** : Protège et contrôle l'accès aux propriétés et méthodes d'une classe.
- **Modificateurs d'accès** :
 - **public** : Accessible partout.
 - **private** : Accessible uniquement dans la classe où il est défini.
 - **protected** : Accessible dans la classe et ses classes dérivées.

3. Héritage et Polymorphisme

- **Héritage** : Permet à une classe fille de réutiliser et étendre les fonctionnalités d'une classe parent avec `extends`.
- **Polymorphisme** : Les méthodes peuvent être redéfinies dans une classe fille pour changer leur comportement tout en conservant la même signature.