# Introduction à la Programmation Orientée Objet en PHP

La programmation orientée objet (POO) est un paradigme de programmation basé sur l'organisation du code en objets. En PHP, la POO permet de structurer et de réutiliser le code de manière efficace et modulaire.

## Fondements de la Programmation Orientée Objet

La POO repose sur trois concepts fondamentaux : les classes, les objets et les méthodes.

**Classe** : Une classe est un modèle ou un plan qui définit les propriétés et les comportements d'un type d'objet. Elle est définie par le mot-clé class.

**Objet** : Un objet est une instance d'une classe. Il possède des propriétés (variables) et des méthodes (fonctions) qui définissent son comportement.

**Méthodes** : Ce sont des fonctions définies à l'intérieur d'une classe et qui opèrent sur ses propriétés.

La POO est particulièrement utile pour les projets complexes comme notre gestion de réservation et de menus, car elle favorise la réutilisabilité et la maintenabilité du code.

## Encapsulation et Modificateurs d'Accès

L'encapsulation est un principe de la POO qui consiste à restreindre l'accès direct aux données de l'objet. Ce concept est mis en œuvre à l'aide des modificateurs d'accès en PHP :

**Public** : Permet l'accès à une propriété ou une méthode depuis n'importe où.

**Private** : Limite l'accès à une propriété ou une méthode à l'intérieur de la classe où elle est définie.

**Protected** : Limite l'accès à une propriété ou une méthode à la classe où elle est définie et à ses sous-classes.

Ces modificateurs permettent de protéger les données internes d'une classe et d'assurer une interaction contrôlée avec elles, ce qui est crucial pour la sécurité et la cohérence des données dans notre projet.

## Héritage et Polymorphisme

### Héritage

L'héritage est un mécanisme qui permet à une classe (dite classe enfant) de dériver d'une autre classe (dite classe parent). La classe enfant hérite des propriétés et des méthodes de la classe parent, ce qui favorise la réutilisabilité du code.

L'héritage est utile pour structurer les rôles dans notre projet, comme distinguer les fonctionnalités entre les clients et l'admin tout en partageant un comportement commun.

### Polymorphisme

Le polymorphisme permet aux objets de différentes classes liées par héritage de répondre différemment à la même méthode. Cela simplifie le traitement des objets en permettant des comportements spécifiques tout en utilisant une interface commune.

Le polymorphisme est essentiel pour gérer les différences de comportement entre les rôles (par exemple, afficher des menus pour les clients et des options de gestion pour l'admin) tout en utilisant des appels uniformes.

Avec ces concepts, la POO en PHP fournit une base solide pour développer des applications robustes et bien organisées, comme notre système de gestion de réservations et de menus.