



# Programmation orientée objet en PHP

[Accueil](#) ► [Mes cours](#) ► [Développement logiciel](#) ► [POO PHP](#) ► La programmation orientée objet : premiers pas ► [Classes et objets](#)

## Classes et objets

Pour aborder un des concepts fondamentaux de la programmation orientée objet (POO), celui de classe, nous allons partir d'un exemple concret : celui de l'être humain.

Comment pourrions-nous décrire un être humain ? Nous pourrions par exemple dire qu'un humain possède nécessairement un âge, une taille, un poids... N'importe quel être humain pourra donc être décrit selon un ensemble choisi de caractéristiques : ici l'âge, la taille, le poids. Chaque être humain aura donc une valeur précise pour chaque caractéristique, ces valeurs ne seront pas forcément les mêmes d'un humain à l'autre. Par exemple un humain pourra peser 100 kg alors qu'un autre n'en pèsera que 50.

Nous pourrions aussi décrire un être humain en énumérant les compétences communes à tous les humains et la façon dont elles se réalisent ; c'est-à-dire ce que tous les humains sont capables de faire et comment ils le font. Par exemple, tous les humains savent marcher en mettant un pied devant l'autre, courir en passant d'un pied sur l'autre, parler en émettant des sons, etc... Toutes ces compétences sont communes au genre humain, tout comme les caractéristiques énumérées précédemment.

Nous pouvons représenter notre description de l'humain dans un tableau avec une case pour le nom de la classe, une pour les caractéristiques ou attributs, une autre pour les compétences ou méthodes :

Humain
age taille poids
marcher en mettant un pied devant l'autre courir en passant d'un pied sur l'autre parler en émettant des sons

La classe Humain est donc en quelque sorte un modèle, une trame ou encore une description qui énumère les attributs et méthodes que possèdent n'importe quel être humain.

Maintenant que nous avons notre "modèle" d'humain, c'est-à-dire notre classe Humain, nous allons l'utiliser pour décrire des humains concrets, par exemple Jack Sparrow et Bruce Wayne. Entre ces deux humains les compétences seront les mêmes, mais les valeurs de leurs attributs seront différentes :

sparrow : Humain
age = 42 taille = 1,84 poids = 75
marcher en mettant un pied devant l'autre courir en passant d'un pied sur l'autre parler en émettant des sons
wayne : Humain
age = 41 taille = 1,95 poids = 100

marcher en mettant un pied devant l'autre courir en passant d'un pied sur l'autre parler en émettant des sons
---

Nos deux humains sparrow et wayne sont des objets de type Humain, ou encore des instances de la classe Humain.

La classe représente le modèle, l'instance de classe (ou l'objet) représente la réalisation concrète de la classe.

Lorsqu'on crée un objet à partir d'une classe, on dit que l'on instancie la classe. L'action de créer un objet à partir d'une classe s'appelle l'instanciation.

Un objet est donc une instance de classe, ses propriétés sont des attributs d'instance et ses compétences des méthodes d'instance.

Fin

## NAVIGATION



### Accueil

#### ■ Ma page

Pages du site

Mon profil

Cours actuel

#### POO PHP

Participants

Généralités

La programmation orientée objet : premiers pas

[Introduction](#)

[Classes et objets](#)

[T.P. première classe](#)

[Les classes en PHP](#)

[T.P. Personne](#)

[Le constructeur et this](#)

[T.P. Personne v2](#)

[Accesseurs et mutateurs](#)

[Les méthodes magiques](#)

[T.P. Personne v3](#)

[Les constantes de classe](#)

[Le modificateur static](#)

[T.P. Personne v4](#)

[Une Personne et une Adresse : la relation has-a](#)

[Histoire de références](#)

[T.P. formation](#)

[T.P. formation v2](#)

[Une Personne et plusieurs Adresse\(s\)](#)

[T.P. formation v3](#)

[Bricolage et dépendance](#)

L'héritage

Les interfaces

Le typage

Les namespaces

Les exceptions

Les bases de données avec PDO

Les tests avec PHPUnit

Petite application version 2

Petite application version 3

[Mes cours](#)

## ADMINISTRATION



[Administration du cours](#)

---

Réglages de mon profil

---

Connecté sous le nom « [Arnaud Lemais](#) » ([Déconnexion](#))  
[POO PHP](#)