**Les classes et les objets (ou instances)**

Le concept de base lié à la P.O.O est que dans un script **"tout est objet"** (ou considéré comme tel). Avant d'aborder la notion d'objet, voyons ce qu'est une classe.

"Une classe est un ensemble cohérent de code qui contient généralement à la fois des variables et des fonctions et qui va nous servir de plan pour créer des objets. Le but d’une classe va donc être de créer des objets que nous allons ensuite pouvoir manipuler."

**Une classe peut être donc considérée comme un moule à partir duquel on peut créer des objets**, chaque objet représentant un objet du monde réel à gérer par le programme. Ainsi, une classe Animal décrit les données et les actions que peut exécuter toute animal et chaque animal à gérer sera un exemplaire (une ‘instance’) de cette classe Animal.

**Une classe décrit la structure interne d'un objet** : les **données** qu'il regroupe, les **actions** qu'il est capable d'assurer sur ses données.

**Le développeur écrit donc le code des classes, mais à l’exécution, ce sont les objets, instances des classes, qui réalisent les opérations.**

**Définition d'une classe**

Il y a quelques règles à connaître et à respecter lorsque l'on définit une nouvelle classe:

* Conventions de nommage:
  + Votre classe ne peut avoir un nom qui est déjà un mot réservé en PHP.
  + Le nom de votre classe doit commencer soit par une lettre majuscule, soit par un underscore.
  + Le nom de votre classe ne peut contenir que des caractères alphanumérique ainsi que le \_.
* Chaque classe doit être définie dans un fichier séparé qui porte par convention le nom de la classe suivi de l'extension .class.php.

En PHP, on définit une nouvelle classe avec le mot-clé class. Nous allons donc nous en servir pour créer notre première classe que nous appellerons Animal.

class Animal{

}

Voilà ! Nous avons défini la classe Animal (une définition certes basique mais une définition tout de même).

**Instanciation d'une classe**

**L'instanciation d'une classe** est l'opération consistant à créer un **objet** à partir de la **classe** en question. on parle aussi de créer une instance.

Cette opération s'effectue par l'intermédiare du mot clé new.

new Animal()

Une instance correspond à la « copie » d’une classe. Le grand intérêt ici est qu’on va pouvoir effectuer des opérations sur chaque instance d’une classe sans affecter les autres instances.

Donc lorsqu'on instancie une classe, on crée un objet. Pour manipuler ce dernier et l'utiliser, il nous faut le stocker dans une variable. C'est cette variable que nous appellerons par la suite notre **objet**:

$monChien = new Animal();

Notre classe est donc le plan de création d'un animal. Nous allons pouvoir y définir les **caractéristiques** de chaque animal (son espèce, sa taille, son poids, son régime alimentaire, son nombre de pattes, etc.) ainsi que les **actions** qu'il peut effectuer (manger, dormir, se reproduire, se déplacer, etc.).

Lorsque nous créerons des animaux à partir de cette classe (ou **nous instancierons cette classe**), chaque animal créé aura ces caractéristiques et ces actions.

Mais les valeurs seront spécifiques à chaque animal. L'espèce sera différente d'un animal à l'autre, tous les animaux ne pèsent pas le même poids et n'ont pas le même régime alimentaire.

De même, les actions définies pour la classe seront les mêmes pour tous les animaux, mais ces actions auront un impact différents selon l'animal (l'objet) qui les executent. Par exemple un animal de grande taille parcourt une plus grande distance lorsqu'il se déplace qu'un aninmal plus petit.