

PHP : OBJET

CLASSE EN PHP

Une classe est une représentation informatique d'un objet de la vie courante, comme une voiture, elle contient des attributs et des méthodes.

Mais quelle est l'intérêt ? On parle de Programmation Orienté Objet ou POO.

Une classe est peu une « boîte à outils » permettant de créer des objets, on dit alors qu'un objet est une instance de classe.

C'est-à-dire que l'on crée des objets qui vont contenir les attributs et méthodes de notre classe, et chaque instance a ses propres attributs et méthodes, tel que déclaré dans la classe.



CRÉER UNE CLASSE

Par convention, lorsque l'on crée une classe on crée un fichier « **MaClasse.php** » en « Pascal-case ».

Afin de créer une classe on utilise le mot-clé « class » avec le nom de celle-ci, il s'agit du nom du fichier sans le « .php » :

Nom de la classe

Attributs

Constructeur de la classe

Méthodes (getter)

```
class Marque
{
    private string $nom;

    public function __construct(string $nom)
    {
        $this->nom = $nom;
    }

    public function getNom(): string {
        return $this->nom;
    }
}
```

CRÉER UNE CLASSE

A partir de PHP 8.0, on peut écrire notre classe de cette manière :

Nom de la classe

Constructeur de la classe

Méthodes (getter)

Attributs

```
class Marque
{
    public function __construct(private string $nom) { }

    public function getNom(): string {
        return $this->nom;
    }
}
```

CLASSE : LE \$THIS

Le « **\$this** » représente « **cette instance** » de la classe, on ne peut s'en servir qu'à l'intérieur d'une classe.

Ainsi le contenu de la méthode « **getNom()** », renvoie la valeur de l'attribut « **nom** » de la classe courante.

```
public function getNom(): string {  
    return $this->nom;  
}
```


UTILISATION DES CLASSES

C'est à ce moment là que l'on va parler d'instanciation, pour cela on va utiliser le mot-clé « **new** » :

```
$renault = new Marque( nom: 'renault');
```

Déclaration
d'une variable

Nom de la classe
à instancier

Paramètre du constructeur
de la classe

Mot clé « new »
Il représente le fait d'instancier
d'une classe

UTILISATION DES CLASSES

C'est à ce moment là que l'on va parler d'instanciation, pour cela on va utiliser le mot-clé « **new** » :

```
$renault = new Marque( nom: 'renault');
```

Déclaration
d'une variable

Nom de la classe
à instancier

Paramètre du constructeur
de la classe

Mot clé « new »
Il représente le fait d'instancier
d'une classe

(PS : il est très important de comprendre que lorsque l'on crée une classe, on ajoute un nouveau type dans notre application)

UTILISATION DES CLASSES

Lorsque l'on instancie une classe, la variable devient un objet du type de la classe créée, autrement dit la variable « **\$renault** » est du type « **Marque** ».

On peut maintenant utiliser les méthodes déclarées dans la classe Marque depuis la variable « **\$renault** » :

```
$name = $renault->getNom();
```

Une variable, qui est une instance de la classe en question

On utilise une flèche pour appeler les méthodes de la classe

La méthode appelée

UTILISATION DES CLASSES

On a déclaré un « **getter** » pour le nom, afin de récupérer le nom de la marque, si l'on souhaite la modifier en dehors, on doit faire un « **setter** ».

Tous les setter fonctionnent de la même manière : par convention on met toujours « **set** » devant suivi du nom de la propriété.

Le contenu de celle-ci doit venir affecter la valeur du nom au sein de la classe.

```
class Marque
{
    public function __construct(private string $nom) { }

    public function getNom(): string {
        return $this->nom;
    }

    public function setNom(string $nom): void {
        $this->nom = $nom;
    }
}
```