# 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 <u>P</u>rogrammation <u>O</u>rienté <u>O</u>bjet ou <u>POO</u>.

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

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





# CRÉER UNE CLASSE

Par convention, lorsque l'on créé une classe on créé 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 »:

```
class Marque
       Nom de la classe
             Attributs
                                                          public function __construct(string $nom)
Constructeur de la classe
                                                          public function getNom(): string {
                                                              return $this->nom;
     Méthodes (getter)
```

# CRÉER UNE CLASSE

**Attributs** 

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

```
lass Marque
     Nom de la classe
                                                  public function __construct(private string $nom) { }
Constructeur de la classe
                                                  public function getNom(): string {
                                                       return $this->nom;
     Méthodes (getter)
```



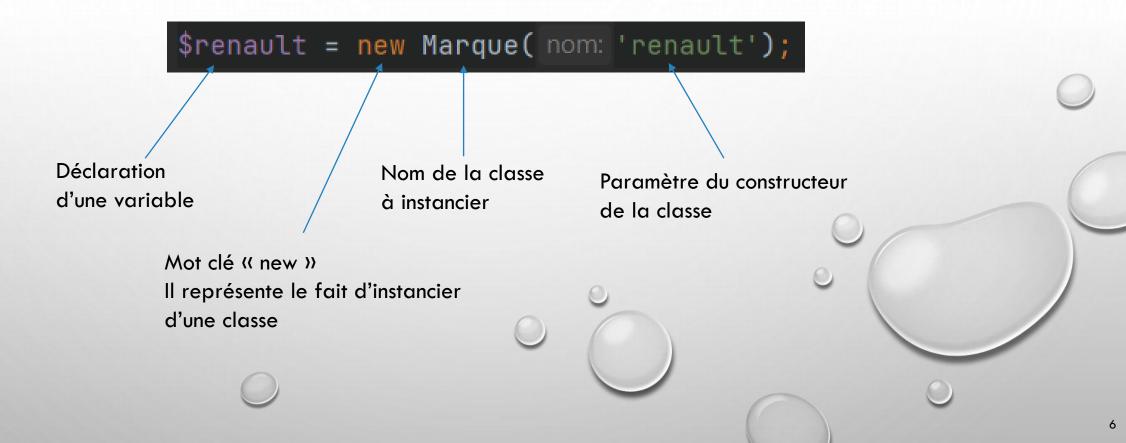
### CLASSE: LE \$THIS

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

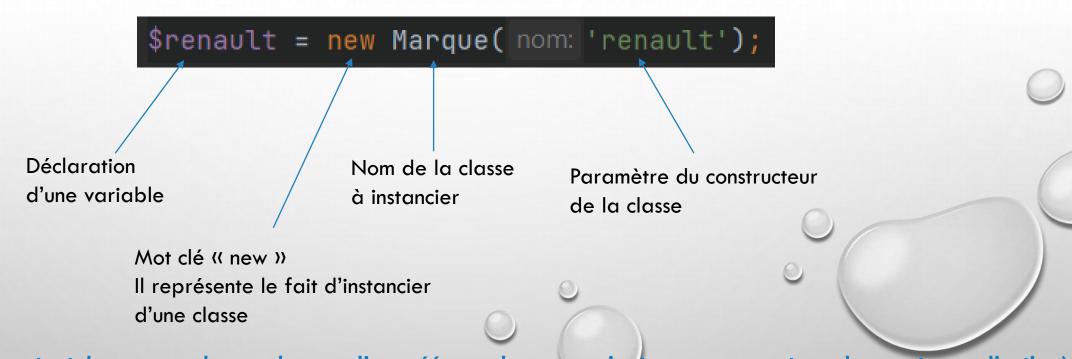
Ainsi le contenu de la méthode « **getNom()** », renvoie la valeur de l'attribut « **nom** » de la classe courante.

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

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



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

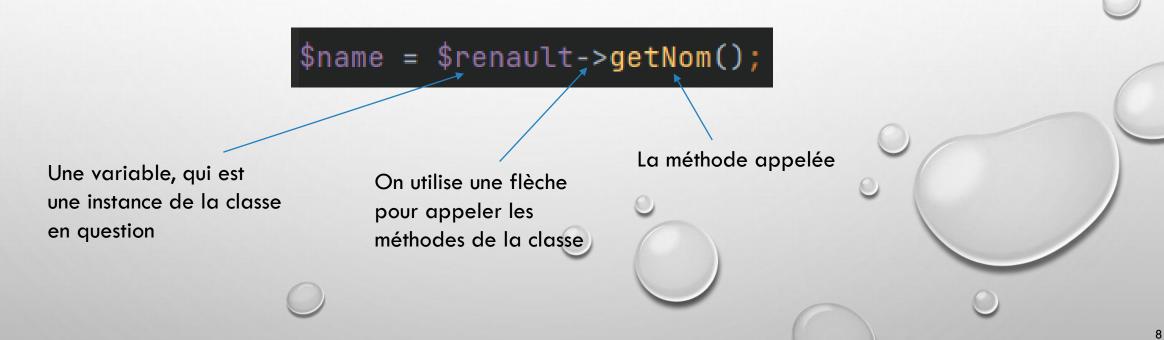


(PS: il est très important de comprendre que lorsque l'on créé une classe, on ajoute un nouveau type dans notre application)



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 » :





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;
```