

# Constructeur et destructeur

## Le constructeur

**Le constructeur d'une classe est une méthode spéciale qui est appelée à chaque création d'un objet.**

Les constructeurs sont donc utilisés pour toutes les initialisations d'objets.

Prenons un premier exemple :

```
<?php
class Personne {

    public string $prenom;
    public string $nom;

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

}

$jean = new Personne('Jean', 'Dupont');
$marie = new Personne('Marie', 'Joseph');

echo '<pre>';
var_dump($marie);
echo '</pre>';
/* object(Personne)#2 (2) {
    ["prenom"]=>
    string(5) "Marie"
    ["nom"]=>
    string(6) "Joseph"
} */
```

Vous pouvez maintenant commencer à comprendre l'utilité des classes : pouvoir simplement instancier des objets ayant des propriétés communes.

La syntaxe moderne, depuis [PHP 8](#), qui est un raccourci exactement équivalent à ce que nous avons vu, est :

```
<?php
class Personne {

    public function direBonjour() {
        echo "Bonjour ! Je m'appelle $this->prenom $this->nom <br>";
    }

    public function __construct(public string $prenom, public string $nom)
    {}

}

$jean = new Personne('Jean', 'Dupont');
$marie = new Personne('Marie', 'Joseph');

$jean->direBonjour(); // Bonjour ! Je m'appelle Jean Dupont
$marie->direBonjour(); // Bonjour ! Je m'appelle Marie Joseph
```

Il est également possible de passer des **paramètres nommés** lors de l'instanciation :

```
<?php
class Personne {

    public function direBonjour() {
        echo "Bonjour ! Je m'appelle $this->prenom $this->nom <br>";
    }

    public function __construct(public string $prenom, public string $nom)
    {}

}

$jean = new Personne(prenom: 'Jean', nom: 'Dupont');
$marie = new Personne(nom: 'Joseph', prenom: 'Marie');
```

```
$jean->direBonjour(); // Bonjour ! Je m'appelle Jean Dupont  
$marie->direBonjour(); // Bonjour ! Je m'appelle Marie Joseph
```

# Le déconstructeur

**Le déconstructeur d'une classe est une méthode spéciale qui est appelée à chaque fois qu'il n'y a plus aucune référence à un objet.**

Autrement dit, cette méthode permet de faire des opérations de nettoyage lorsqu'un objet n'est plus utilisé.

Voici l'exemple de la vidéo :

```
<?php  
class Reader {  
  
    public function __construct(public $filename) {  
        if (file_exists($filename)) {  
            $this->handle = fopen($filename, 'r');  
        } else {  
            echo "Le fichier n'existe pas";  
        }  
    }  
  
    public function read() {  
        echo fread($this->handle, 10);  
    }  
  
    function __destruct() {  
        fclose($this->handle);  
    }  
  
}  
  
$reader = new Reader(__DIR__ . '/test.txt');  
$reader->read();
```