

# PHP Objet L'HÉRITAGE

Juin 2018



# 1.

## Principe

Récupérer le fonctionnement d'une autre classe

## Principe

### DVD

- » prix
- » dimensions
- » poids
- » genre
- » acteurs
- » zone

### Livre

- » prix
- » dimensions
- » poids
- » genre
- » auteur
- » ISBN

## Principe

### DVD

- » acteurs
- » zone

### Produit

- » prix
- » dimensions
- » poids
- » genre

### Livre

- » auteur
- » ISBN



## Le mot clé “protected”

Les propriétés et méthodes avec la visibilité **protected** peuvent être utilisées que par la classe elle-même et les classes qui en héritent

## Démonstration

```
class Produit
{
    protected $prix;
    protected $dimensions;
    protected $poids;
    protected $genre;

    public function prix()
    {
        return $this->prix;
    }
}
```

```
class DVD extends Produit
{
    private $acteurs;
    private $zone;

    public function getActeurs()
    {
        return $this->acteurs;
    }
}
```

# 2.

## Surcharge

Modifier les méthodes de la classe parente

## Surcharge

La surcharge est le fait de redéfinir une méthode de la classe parente

```
class Produit
{
    public function afficher() {
        echo $this->title;
    }
}
```

```
class Livre extends Produit
{
    public function afficher() {
        echo $this->title
        . ' '
        . $this->genre;
    }
}
```



## Surcharge

Il est possible d'appeler une méthode surchargée avec l'opérateur ::

```
class Livre extends Produit
{
    public function __construct()
    {
        parent::__construct();
    }
}
```

# 3.

## **Classes finales**

Elles ne peuvent pas être héritées

## Classes finales

Le mot clé final indique que la classe ne peut pas avoir d'enfant

```
final class Livre  
{  
}
```