

Héritage

Héritage

Constructeur

final

final static

Accès à une propriété statique

Opérateur ::

Comme en Java, il n'y a **pas d'héritage multiple** en php.

```
class chien extends Animal{
    function affiche(){
        parent::affiche(); //référence vers la classe mère (statique ou non)
    }
}
```

Constructeur

Contrairement au java, où il faut faire un appel explicite au constructeur parent, le php hérite de la méthode constructeur comme n'importe quelle autre :

```
function __construct($nom,$prenom,$age,$login){
    parent::__construct($nom, $prenom, $age);
    this->login = $login;
}
```

final

Une **fonction** déclarée dans le parent comme final ne pourra **pas être redéfinie** par les classes filles. Dans le même ordre d'idée une **classe** définie comme finale ne pourra **pas** avoir **de filles**.

final static

(ou constantes propres)

```
const MAX_AGE = 99;
```

Accès à une propriété statique

```
class Etudiant{
    static $nom;
    function hello(){
        echo "hello" . self::$nom;
    }
}
```

On utilisera à l'extérieur/intérieure de la classe :

```
Etudiant::$nom;
```

mais on peut aussi utiliser à l'intérieur de la classe :

```
self::$nom;
```

this->nom ne fonctionne pas et les méthodes statiques vont générer un warning

Opérateur ::

il est utilisé dans les exemples ci-dessus mais il peut aussi être utilisé comme suit :

```
$nomDeClasse = 'Etudiant';
$nomDeClasse::$AgeMaximum;
```