Chapitre 6 : Les Objets en php

Constructeurs

Les constructeurs sont des fonctions qui vont se charger de créer des objets. Il existe 2 syntaxes, une ancienne et une nouvelle (utilisant les fonctions magiques).

Il est important de noter que la syntaxe flèche vu ci-dessous s'effectue sans $ devant le nom de l'attribut

Ancienne syntaxe

```php
class Etudiant{
    function Etudiant($nom,$age){
        this->nom = $nom;
        this->age = $age;
    }
}
```

Nouvelle syntaxe (fonction magique __construct)

```php
class Etudiant{
    public $nom,$age;
    function __construct($nom,$age){ // automatiquement appelé quand crée un nouvel objet
        this->nom = $nom;
        this->age = $age;
    }
    function __deconstruct(){
        // fonction appelée automatiquement avant la destruction de l'objet
    }
}
```

> attention, il y a **2** Under score devant une fonction magique

Instancier l'objet

```php
$antoine = new Etudiant("antoine",18);
```

modifier / accéder à une propriété

```php
$antoine->age = 22;
```

Afficher le contenu d'un objet

```php
printr($obj);
```

Les valeurs par défaut

```php
class Etudiant{
    public $nom,$age = 69;
}
```

This

Attention à toujours bien utiliser this lorsque l'on souhaite accéder à une propriété d'un objet.

Comparaison d'objets

```php
$O1 == $O2; // comparaison par propriétés
$O1 === $O2; // comparaison par référence
```

Si on compare 2 objets par propriétés, cette égalité sera vraie si les 2 objets ont :

- tous les 2 la même classe
- et ont les valeurs de leurs propriétés identiques

Tester le type d'un objet

```php
($e instanceof Etudiant)
```

Parcourir un Objet

On peut parcourir les propriétés d'un objet comme si l'on parcourais un tableau (cfr forEach chapitres boucles)

Supprimer un objet

Pour supprimer un objet, il faut supprimer toutes les références vers celui-ci.

Il est aussi possible d'utiliser unset($obj);

> avant d'utiliser unset, veiller à bien retirer toutes les réf. sinon l'appel au destructeur ne se fera pas

Static

On peut créer une variable statique.

attention : pour accéder à cette variable il faut utiliser self :

```
self::$varStatique
Etudiant::$varStatiqueDuneClasse // acces depuis une classe extérieure
```

Constantes d'une classe

Même foctionnement que une static

Attention nom de variable en majuscule

Auto-Load de classes

Il peut être intéressant de séparer les classes dans des fichiers qui leurs sont propres. (zqddle fichier : maClasse.class.php pour maClasse). Et pour éviter à chaque page de charger toutes les classes, on peut faire appel à un autoloader. Une fonction qui va automatiquement charger les classes inconnues qu'il rencontre.

```
function __autoload($classe){
    require_once "classes/$classe.class.php";
}
```

Cette fonction est extrêmement pratique pour garder le code clean mais **Attention** car les classes doivent **impérativement** se trouver dans un fichier classes et se nommer de la façon suivante *NOMDECLASSE.class.php*

Propriétés

Visibilité

Les propriétés ont 3 modes de visibilité (comme en Java) :

- Public (défaut)
- Protected : accès uniquement par la classe et ses sous classes

- Private : accès uniquement par la classe

Destruction d'un objet

Un objet est détruit si on supprime toute référence vers lui.

> supprimer la référence d'un objet via unset($monObjet);

Fonction magique __destruct

La fonction magique __destruct() est automatiquement appelée juste avant la destruction d'un objet.