

# Programmation Web

18 septembre 2017

# PHP objet : syntaxe de classe

- mot clé **class**,
- classe abstraite (non instanciable) avec le mot clé **abstract**,
- héritage de classe avec le mot clé **extends**.

```
<?php
abstract class A
{
}

class B extends A
{
}
?>
```

# PHP objet : propriétés et méthodes

- Portée (visibilité) :
  - public, protected, private

```
<?php
class A
{
    private $_param;
    public function hello()
    {
    }
}
```

# PHP objet : instances et accès

- Création avec mot clé **new**
- Appel des méthodes et accès aux propriétés avec **->**
- Accès à l'instance explicitement avec **\$this**

```
<?php
class A
{
    private $_text = "Hello World";
    public function parler()
    {
        echo $this->_text;
    }
}

$obj = new A();
$obj->parler();
```

# PHP objet : méthodes statiques

- Méthode de la classe indépendante de l'instance,
- Mot clé **static**,
- Peut être appelée avec l'opérateur de résolution de portée ::

```
<?php
class A
{
    public static function parler()
    {
        echo "Hello World";
    }
}

A::parler();
```

# PHP objet : propriétés et constantes statiques

- Création avec mot clé **static**,
- Accès depuis l'extérieur de la classe avec **::** (exemple PDO::**FETCH\_ASSOC**),
- Accès depuis une méthode statique avec **self**

```
<?php
class A
{
    private static $_compteur = 0;
    public static function parler()
    {
        echo self->_compteur;
    }
}

$objj = new A();
$objj->parler();
```

- Leur nom commence par deux underscore. Exemple : `__construct`
- Elles sont appelées lors d'événements particuliers. Exemple : création d'un objet avec `new`

# Design Pattern (masques de conception)

- répondent à des problématiques de conception récurrentes,
- correspondent à un ensemble de bonnes pratiques.



Objectif : S'assurer qu'une seule et même instance d'une classe soit utilisée

- Nous utilisons PDO pour accéder à la base de données
- Nous n'avons besoin que d'une unique connexion à la base de données

# Le pattern DAO : Data Access Object

Objectif : Permettre d'accéder aux données persistantes sans connaître comment elles sont stockées (BD, fichiers XML,...)

- DAO = Repository
- Nous utilisons une base de données pour stocker les films
- Nous avons besoin d'accéder à ces données

# Le pattern DTO : Data Transfer Object

Objectif : Simplifier les transferts de données (entre modèle et contrôleur, contrôleur et vue)

- DTO = Entity
- Des classes très simples pour les objets métiers (propriétés + accesseurs)