

Programmation Web

11 septembre 2017

- Elles sont écrites en majuscules avec un underscore au début
- Ce sont des tableaux associatifs : des ensembles de clé/valeur
- Elles sont créées à chaque chargement de page
- `$_GET` contient les données envoyées par l'URL
- `$_POST` contient les informations transmises par les formulaires avec la méthode POST

Les fonctions de PHP

PHP propose des multitudes de fonctions "toutes prêtes" qui permettent entre autres :

- de manipuler des chaînes de caractères,
- de déplacer / envoyer des fichiers,
- de manipuler des images,
- d'envoyer des e-mails,
- de crypter des mots de passe,
- de manipuler des dates,
- ...

Remarque

Le site web de PHP référence toutes les fonctions par catégorie :
<http://fr.php.net/manual/fr/funcref.php>

```
function nomDeLaFonction($param1 , $param2=5)
{
    // Bloc d'instructions
    return $variableContenantValeurDeRetour;
}
```

- Une fonction peut retourner
 - rien (procédure)
 - une seule valeur de type simple
 - un type structuré comme un tableau ou un objet

- Les objets = les instances
 - Création avec mot clé **new**
 - Appel des méthodes et accès aux propriétés avec ->

- PHP peut s'utiliser avec différents SGBD : **MySQL**, Oracle, PostgreSQL, ...
- Interfaces web d'administration des bases de données écrites en PHP : **PHPMyAdmin** (MySQL), phpPgAdmin (PostgreSQL), ...
 - <http://iutdoua-web.univ-lyon1.fr/phpMyAdmin>
 - [http://localhost:\[port_apache\]/phpMyAdmin/](http://localhost:[port_apache]/phpMyAdmin/)
- Plusieurs façons d'accéder à la base de données à partir d'un script PHP
 - des extensions spécifiques des SGBD comme MySQLi
 - des interfaces d'accès aux bases de données comme **PDO** (PHP Data Objects) qui permettent de faire abstraction du SGBD utilisé.

- A partir de PHP5
- Il faut que l'extension PDO soit activée
- Il faut le pilote MySQL pour PDO

Remarque

`phpinfo()` permet de savoir quel fichier de configuration `php.ini` est utilisé par le serveur, ainsi que les extensions et pilotes activés.

Fonction de connexion à une base de données MySQL

```
<?php
define('BD_HOST','localhost');
define('BD_DBNAME','pXXXX');
define('BD_USER','pXXXX');
define('BD_PWD','XXXX');
try
{
    $bdd=new PDO('mysql:host='. BD_HOST .'; dbname='.
        BD_DBNAME .'; charset=utf8', BD_USER , BD_PWD );
    $bdd->setAttribute(PDO::ATTR_ERRMODE, PDO::
        ERRMODE_EXCEPTION);
}
catch ( Exception $e)
{
    die ('Erreur : '.$e->getMessage());
}
```


PDO : méthodes pour faire des requêtes

- requêtes avec SELECT

```
$req= $bdd->query('SELECT Titre, Année, Score,  
    Votes from Movie');
```

Retourne un jeu de résultats sous forme d'objet PDOStatement.

- requêtes avec modification des données dans la BD (UPDATE, INSERT,...)

```
$count = $bdd->exec("DELETE FROM fruit WHERE  
    couleur = 'rouge'");
```

Retourne le nombres de lignes modifiées/effacées.

A noter

Les requêtes SELECT effectuées avec un objet PDO retournent un objet PDOStatement.

PDOStatement : quelques méthodes utiles

- compter le nombres de lignes sélectionnées.

```
$count = $req->rowCount();
```

- Récupération de toutes les lignes d'un jeu de résultats dans un tableau associatif

```
$result = $req->fetchAll();  
print_r($result);
```

- Récupération de la ligne suivante d'un jeu de résultats

```
$result = $req->fetch(PDO::FETCH_ASSOC); /*  
    résultat sera un tableau associatif */
```

Retourne FALSE quand il n'y a plus de ligne dans le jeu.

PDO : des requêtes avec des variables - prepare/execute

- avec des marqueurs

```
$req1= $bdd->prepare('SELECT * FROM Casting  
WHERE MovieID = ? ORDER BY Ordinal');  
$req1->execute(array($_GET['movieid']));
```

- avec des paramètres nommés

```
$req1= $bdd->prepare('SELECT * FROM Casting  
WHERE MovieID = :movieid ORDER BY Ordinal');  
$req1->execute(array(':movieid' => $_GET['  
movieid']));
```

A noter

Ces méthodes protègent des injections de code SQL.

- Avant de pouvoir lancer une autre requête

```
$req1->closeCursor();
```

Intégrer des fichiers externes

- PHP a été pensé pour la conception d'applications Web
- permet de définir des "briques de base" qui peuvent être réutilisées plusieurs fois
- permet de répercuter facilement un changement sur l'ensemble de l'application
- Les fonctions d'intégration :
 - `include ("header.php");` inclut les lignes du fichier `header.php` avec un warning si il n'existe pas.
 - `require ("header.php");` inclut les lignes du fichier `header.php` avec un erreur fatale si il n'existe pas (sortie du script).
 - `include_once` et `require_once` sont les équivalents avec vérification avant intégration que c'est la première fois. Dans le cas contraire, le fichier n'est pas intégré.

- Se fait à l'aide de la fonction `header('Location : chemin')` qui envoie une redirection au navigateur.

```
<?php
    header('Location: index.php?status=right');
    exit(0); // ou exit(); ou exit;
?>
```

- Pas d'espace avant les deux points et un espace après
- Cette redirection doit être effectuée avant l'envoi de toute sortie au navigateur donc :
 - aucun appel de la fonction `echo` avant notamment dans le PHP
 - pas d'espace ou de retour-charriot dans le HTML

Le pattern MVC : Model, View, Controller

- C'est une architecture applicative donc une façon d'organiser le code source d'une application.
- Il consiste à découper le code de cette application en séparant clairement :
 - Une partie modèle
 - Une partie vue
 - Une partie contrôleur
- C'est l'architecture utilisée par le framework PHP Symfony 2 par exemple.

Framework

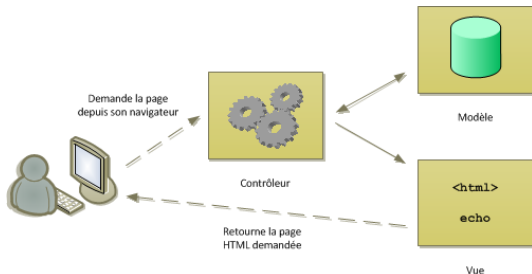
C'est un ensemble de composants qui sert à créer les fondations, l'architecture et les grandes lignes d'un logiciel.

- C'est la partie affichage
- Elle contient le code HTML
- Elle contient un peu de code PHP (boucles, conditions) pour l'affichage du contenu des variables
- C'est la problématique présentation
- C'est le designer du projet qui travaille sur les vues

- Elle contient la logique métier (business logic)
- C'est la problématique traitement / calculs
- Elle contient aussi tout ce qui concerne l'accès aux données
- C'est là qu'on trouvera les requêtes SQL

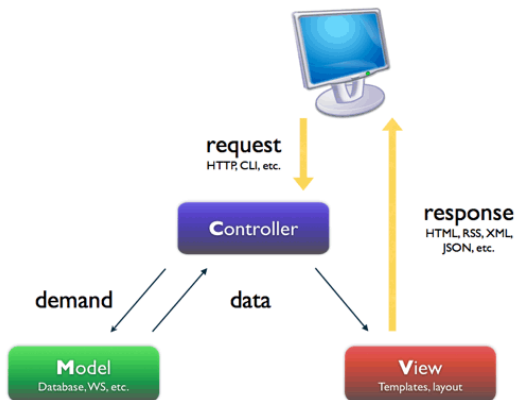
- Fait le lien entre l'utilisateur et le reste de l'application
- Il est la couche qui se charge d'analyser et de traiter la requête de l'utilisateur.
- Il demande au modèle les données, les traite et appelle la vue qui utilisera ces données pour afficher la page.

Représentation Schématique (1/2) :



Source image : OpenClassrooms

Représentation Schématique (2/2) :



Source image : Documentation Symfony

Avantages du pattern MVC

- Code clair, bien organisé, gage de réutilisabilité
- Découpage standard qui permet aux autres développeurs de rapidement rentrer dans votre code
- Découpage des responsabilités qui permet à des développeurs avec différents profils de travailler sur un projet

- Lire le chapitre du cours "Gestion de bases de données"