

## Programmation coté serveur avec PHP

### Introduction

- Une archive contenant les fichiers mentionnés dans cette fiche est à récupérer sur la liste de diffusion.
- Une référence du langage PHP est disponible à cette adresse : <http://php.net/manual/fr/>
- Le manuel de référence de MySQL est disponible à cette adresse : <http://dev.mysql.com/doc/refman/5.6/en/index.html>
- Le manuel de référence de PDO est disponible à cette adresse : <http://php.net/manual/fr/book.pdo.php>

### Exercice 1 – Hello CGI

Récupérez le fichier `form.php` depuis l'archive et ouvrez le avec un éditeur de texte.

**Question 1.1 :** Dans un terminal, exécutez la commande `php form.php`. Que constatez-vous ?

**Question 1.2 :** Que devient le code `<?php echo $_SERVER['PHP_SELF']; ?>` ? À quoi sert-il ? Placez ce fichier sur votre serveur web et testez à nouveau en examinant le code envoyé à votre navigateur.

**Question 1.3 :** Ajoutez une fonction qui affiche le contenu d'un tableau associatif et appliquez-la aux tableaux `$_GET`, `$_POST`, `$_REQUEST`, `$_SERVER` et `$_FILES`. Que contiennent-ils ?

### Exercice 2 – Les sessions en PHP

**Question 2.1 :** Rappelez ce que sont les sessions et à quoi elles servent.

**Question 2.2 :** Documentez le rôle des fonctions suivantes :

- `session_start()`,
- `session_destroy()`,
- `session_register("var")`,
- `session_unregister("var")`.

**Question 2.3 :** Expliquez pourquoi `session_start()` doit être appelé avant tout envoi d'information au client.

**Question 2.4 :** Modifiez le programme de l'exercice précédent pour que après chaque requête soient affichées la nouvelle valeur du champs `nom` ainsi que son ancienne valeur.

**Question 2.5 :** Ajoutez un bouton pour détruire la session.

### Exercice 3 – Un premier MVC

Pour cet exercice, on veillera à placer les différents composants du MVC dans des fichiers séparés.

**Question 3.1 :** Écrivez une classe `Person` contenant un `identifiant`, un `nom`, un `prenom`, et un `titre` (civilité) dont les valeurs seront passées au constructeur. Fournissez des accesseurs pour chacun de ces attributs. Vous placerez le code dans le fichier `Person.php`.

**Question 3.2 :** Écrivez une classe `PersonModel` (`PersonModel.php` dans laquelle vous placerez une méthode statique `getPersons($start, $nb)` qui renverra un tableau de `$nb` personnes créées avec des valeurs aléatoires (des tableaux des constantes sont fournis dans le fichier) avec des identifiants séquentiels commençant à `$start`.

**Question 3.3 :** Écrivez une classe `PersonView` (`PersonView.php` dans laquelle vous placerez une méthode statique `displayPersonList($pl)` qui renverra un tableau HTML contenant les données de la liste de personnes `$pl` (une personne par ligne) avec les en-têtes `Id`, `Titre`, `Nom` et `Prénom`.

**Question 3.4 :** Écrivez une classe `AppView` (`AppView.php`) dans laquelle vous placerez une méthode statique `showApp($content)` qui produira un document complet en HTML5 avec un titre et affichera le contenu de `$content`.

**Question 3.5 :** Écrivez un contrôleur (`controler.php`) qui affiche dans le navigateur l'application listant 10 personnes aléatoirement générées en utilisant `AppView`, `PersonModel` et `PersonView`.

#### **Exercice 4 – PHP et les bases de données**

Depuis la version 5.1 de PHP, l'extension PDO (PHP Data Objects) fournit une couche d'abstraction pour accéder aux bases de données. Le code écrit devient portable et plus sûr. Voici un court exemple :

```
1  <?php
2  try {
3      $dbh = new PDO('mysql:host=localhost;dbname=test', $user, $pass);
4      foreach($dbh->query('SELECT * from FOO') as $row) {
5          print_r($row);
6      }
7      $dbh = null;
8  } catch (PDOException $e) {
9      print "Erreur !: " . $e->getMessage() . "<br/>";
10     die();
11 }
12
13 // Fermeture de la connexion :
14 $dbh = null;
15 ?>
```

**Question 4.1 :** Exécutez le script `person.sql` sur votre base de données MySQL pour créer la table `PERSONS` et y charger des données de test.

**Question 4.2 :** Modifier la classe `PersonModel` (que vous placerez dans `PersonDB.php`) pour que la méthode `getPersons()` récupère les informations depuis la base de données plutôt que de les générer aléatoirement.

#### **Exercice 5 – Bonus : génération et affichage de contenu sur plusieurs pages**

**Question 5.1 :** Modifiez votre contrôleur pour qu'il accepte une variable de formulaire indiquant le numéro de la page à afficher.

**Question 5.2 :** Modifier votre vue pour y ajouter des boutons/liens de navigation (page suivante, page précédente, première page, dernière page) et afficher le numéro de la page en cours.

**Question 5.3 :** Ajoutez en face de chaque ligne une icône permettant de supprimer l'entrée. Ajouter le contrôleur effectuant cette action.