

PHP – Semaine 1/6 – Séance d'exercices 2/3

Exercice n°4 – Découvrir une architecture MVC

Téléchargez sur moodle.vinci.be le package **Semaine 1 - Ex. PHP 2 - Code Source.zip** et décompressez-le.

Faites fonctionner le code source présenté dans le répertoire **4.(m)vc** à l'aide de Wampserver et d'un navigateur Web de votre choix.

Examinez en détails le code de l'exemple **4.(m)vc** et répondez aux questions suivantes.

Quel est **LE script** auquel il faut faire appel en tant qu'URL dans le navigateur Web ?

Que **produit ce** script PHP ?

Quelle est l'**URL complète** que vous avez entrée dans le navigateur Web pour faire fonctionner l'exemple **4.(m)vc** ?

Quel est le **répertoire racine** du site Web ?

Que réalise la **fonction** `define('CHEMIN_VUES', 'views/');` ?

Que réalise en général la **fonction** `require_once` ?

Dans l'expression `CHEMIN_VUES.'header.php'` **que réalise l'opérateur . ?**

A quoi sert la classe `AccueilController` ?

Où est appelé le constructeur de la classe `AccueilController` ?

Où est exécuté le code de la fonction `run` du contrôleur `AccueilController` ?

Que contient le fichier `accueil.php` ?

Que réalise l'instruction `<?php echo $message ?>` ?

Y a-t-il une couche Modèle dans ce premier exemple **4.(m)vc** ?

Remarque : pour que les lettres accentuées s'affichent correctement dans les navigateurs Web, tous les fichiers de développement doivent être encodés selon le **format UTF-8 (sans BOM)**.

Quelques éléments de réponses à l'exercice n°4

index.php est LE script à appeler, c'est LE contrôleur principal du site Web.

Ce fichier index.php n'est au final qu'un seul script PHP qui produit du code source HTML.

`http://localhost/4.(m)vc/index.php` est l'URL complète. Si n'écrivez pas `index.php`, il est appelé par défaut.

`define('CHEMIN_VUES', 'views/');` définit une variable constante `CHEMIN_VUES` et l'initialise à la valeur `'views/'`.

L'instruction `require_once` inclut le code du fichier passé en paramètre. Le chemin du fichier doit être correct, sinon il se produit une erreur fatale et l'arrêt du script. Si le fichier a déjà été inclus précédemment, il n'est pas inclus une seconde fois.

L'opérateur `.` est l'opérateur de concaténation de chaînes de caractères. Dans l'exemple `CHEMIN_VUES.'header.php'`, il colle la valeur de la constante `CHEMIN_VUES` et `'header.php'` donc cela donne `'views/header.php'`.

La classe `AccueilController` contient un constructeur vide et une méthode `run()` qui initialise deux variables à l'aide de code PHP et, à la fin, écrit le code de la vue `'views/accueil.php'`.

Le constructeur de la classe `AccueilController` est appelé dans le contrôleur principal `index.php` grâce à la méthode `new AccueilController();`.

La method `run()` est appelée dans le contrôleur principal `index.php` en écrivant
`$controller = new AccueilController();`
`$controller->run();`

Le fichier `accueil.php` contient du code HTML parsemé de balises PHP pour afficher le contenu de variables initialisées dans le contrôleur correspondant.

L'expression `<?php echo $message ?>` passe en mode PHP et écrit ici le contenu de la variable `$message`.

La source HTML produite par le script `index.php` est (exécution du script le 12 janvier 2019) :

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta charset="utf-8">
    <title>Un premier script PHP selon un modèle (m)vc</title>
  </head>
  <body>
    <p>Bonjour tout le monde !</p>
    <p>Nous sommes le 12/01/2019.</p>
  </body>
</html>
```

Exercice n°5 – Afficher l'heure

Sur base de l'exemple précédent du répertoire **4.(m)vc**, écrivez une solution Web **respectant l'architecture MVC** qui affiche :

Nous sommes à la deuxième séance d'exercices de PHP !

Il est exactement 10:01:15.

C'est bien sûr l'heure actuelle qui est affichée. Les textes sont centrés. Veillez à correctement utiliser une feuille de style CSS.

Pour ce faire, utilisez la fonction `date` de PHP et lisez la documentation de la fonction à l'adresse <http://php.net/date>.

Il est possible de devoir utiliser la fonction `date_default_timezone_set` au préalable pour obtenir un affichage correct de l'heure selon le fuseau horaire (*timezone*) ou avoir pu modifier le fichier de configuration du moteur php intitulé `php.ini`.

Enfin, comme vu au cours d'HTML, validez votre code source (HTML 5) à cette adresse : <http://validator.w3.org/>.

Assurez-vous que les lettres accentuées s'affichent correctement. Vous aurez peut-être besoin de lire la page suivante : <http://www.alsacreations.com/astuce/lire/69-declarer-encodage-des-caracteres.html>.

Exercice n°6 – Utiliser des variables

Nous vous invitons à lire les informations suivantes :

<https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/911847-les-variables>

En respectant le modèle MVC acquis à l'exercice n°4, écrivez une solution Web qui affiche dans un navigateur les 25 premiers nombres de la suite de Fibonacci comme ceci :

Les 25 premiers nombres de la suite de Fibonacci sont : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
610 987 1597 2584 4181 6765 10946 17711 28657 46368

Pour cet algorithme, vous avez besoin de savoir écrire une boucle `for` en PHP, vous trouverez des explications pour réaliser cela chez <https://openclassrooms.com/fr/courses/918836-concevez-votre-site-web-avec-php-et-mysql/912133-les-boucles>.

Veillez à correctement utiliser une feuille de style CSS.

Exercice n°7 – Utiliser des constantes

Complétez l'exercice n°6 en affichant dans le *footer* comme suit votre prénom et votre nom, la version de PHP installé sur votre machine de travail, et le type de navigateur utilisé pour afficher la page :

Jean-Luc Collinet
Version du moteur PHP installé : 5.5.12
Mozilla/5.0 (Windows NT 6.1; WOW64; rv:34.0) Gecko/20100101 Firefox/34.0

Veillez à correctement utiliser une feuille de style CSS.

Exercice n°8 – Afficher un formulaire

En respectant l'architecture MVC élaborée jusqu'à présent, écrivez une solution Web en PHP qui affiche le formulaire suivant dans un navigateur Internet :

Convertisseur EUR-BTC

Valeur à convertir ☒ Euros vers Bitcoin ☐ Bitcoin vers Euros

Jean-Luc Collinet



Points à respecter :

- La valeur 0 (valeur par défaut) est affichée dans la zone de texte.
- Le *radio button* Euros vers Bitcoin est sélectionné par défaut.
- La valeur associée au 1^{er} *radio button* est "toBTC".
- La valeur associée au 2^{ème} *radio button* est "toEUR".
- Les valeurs du formulaire seront envoyées au serveur par la méthode **POST**, elles ne seront ainsi **pas** visibles dans l'URL.

Continuez l'exercice en rajoutant une feuille de style pour mettre en forme le formulaire à votre goût.

Points à respecter :

- La feuille de style `.css` est à placer dans un sous-répertoire `/views/css`
- Une image est à afficher, à placer dans un sous-répertoire `/views/images`
- Le code HTML 5 produit est à valider.

C'est la semaine prochaine que nous réaliserons l'action dans un contrôleur pour convertir la valeur entrée dans le formulaire et afficher le résultat de la conversion.