



# Formation PHP Symfony - 0

## OOP

*Résumé: Dans ce projet de la piscine PHP Symfony, vous allez (ré)apprendre les bases de la programmation orientée objet et des fonctionnalités avancées de PHP.*

*Version: 1.1*

# Table des matières

<b>I</b>	<b>Préambule</b>	<b>2</b>
<b>II</b>	<b>Règles communes</b>	<b>3</b>
<b>III</b>	<b>Consignes spécifiques à cette journée</b>	<b>4</b>
<b>IV</b>	<b>Exercice 00</b>	<b>5</b>
<b>V</b>	<b>Exercice 01</b>	<b>6</b>
<b>VI</b>	<b>Exercice 02</b>	<b>7</b>
<b>VII</b>	<b>Exercice 03</b>	<b>8</b>
<b>VIII</b>	<b>Exercice 04</b>	<b>10</b>
<b>IX</b>	<b>Exercice 05</b>	<b>11</b>
<b>X</b>	<b>Rendu et peer-évaluation</b>	<b>12</b>

# Chapitre I

## Préambule

Vous allez également apprendre à écrire des programmes modulaires et scalables en PHP.

# Chapitre II

## Règles communes

- Votre projet doit être réalisé dans une machine virtuelle.
- Votre machine virtuelle doit avoir tout les logiciels necessaire pour réaliser votre projet. Ces logiciels doivent être configurés et installés.
- Vous êtes libre sur le choix du systmème d'exploitation à utiliser pour votre machine virtuelle.
- Vous devez pouvoir utiliser votre machine virtuelle depuis un ordinateur en cluster.
- Vous devez utiliser un dossier partagé entre votre machine virtuelle et votre machine hôte.
- Lors de vos évaluations vous allez utiliser ce dossier partager avec votre dépôt de rendu.
- Vos fonctions de doivent pas s'arrêter de manière inattendue (segmentation fault, bus error, double free, etc) mis à part dans le cas d'un comportement indéfini. Si cela arrive, votre projet sera considéré non fonctionnel et vous aurez 0 au projet.
- Nous vous recommandons de créer des programmes de test pour votre projet, bien que ce travail **ne sera pas rendu ni noté**. Cela vous donnera une chance de tester facilement votre travail ainsi que celui de vos pairs.
- Vous devez rendre votre travail sur le git qui vous est assigné. Seul le travail déposé sur git sera évalué. Si Deepthought doit corriger votre travail, cela sera fait à la fin des peer-evaluations. Si une erreur se produit pendant l'évaluation Deepthought, celle-ci s'arrête.


# Chapitre III

## Consignes spécifiques à cette journée

- Si aucune information contraire n'est explicitement présente, vous devez assumer les versions de langages suivantes :
  - PHP - Symfony version LTS
  - HTML 5
  - CSS 3
  - Vous devez suivre scrupuleusement les demandes de chaque exercice et respecter les conventions de nommage des fichiers.

# Chapitre IV

## Exercice 00


	Exercice : 00
Exercice 00 : Les paramètres HTML	
Dossier de rendu : <i>ex00/</i>	
Fichiers à rendre : <code>TemplateEngine.php</code> , <code>index.php</code>	
Fonctions Autorisées :	

A l'aide du fichier template **book\_description.html**, vous devez créer un fichier **TemplateEngine.php** avec une classe **TemplateEngine** contenant une méthode **createFile(\$fileName, \$templateName, \$parameters)**. Cette méthode devrait générer un nouveau fichier HTML avec le nom **\$fileName** en remplaçant les paramètres entre **{}** dans le fichier template (ici **book\_description.html**) avec les valeurs des paramètres en provenance du tableau **\$parameters**.

Vous allez également créer un fichier **index.php** qui devra instancier la classe et appeler la méthode décrite avec un ensemble d'arguments que vous allez spécifier.

# Chapitre V

## Exercice 01

	Exercice : 01
Exercice 01 : Affichage de l'object	
Dossier de rendu : <i>ex01/</i>	
Fichiers à rendre : <code>TemplateEngine.php</code> , <code>index.php</code> , <code>Text.php</code>	
Fonctions Autorisées :	

Vous devez creer une classe **Text** qui devra contenir au minimum :


- un constructeur avec pour paramètre un tableau de chaines de caractères.
- une méthode **append** qu doit permettre d'ajouter de nouvelles chaînes de caractères à cette classe.
- une méthode **readData** qui doit retourner toutes les chaînes de caractères en HTML. Chacune chaine doit être dans une balise `<p>`.

Vous devez ensuite modifier la méthode **createFile(\$fileName, \$templateName, \$parameters)** de la classe **TemplateEngine**. Elle doit prendre un objet Text comme dernier paramètre : **createFile(\$fileName, \$text)**.

La nouvelle méthode fraîchement modifié **createFile** doit créer un fichier HTML statique similaire à celui généré dans l'Exercice 00. Le corps doit inclure le contenu rendu par la classe Text.

# Chapitre VI

## Exercice 02

	Exercice : 02
Exercice 02 : Affichage de l'objet 2	
Dossier de rendu : <i>ex02/</i>	
Fichiers à rendre : <code>TemplateEngine.php</code> , <code>index.php</code> , <code>HotBeverage.php</code> , <code>Coffee.php</code> , <code>Tea.php</code>	
Fonctions Autorisées :	

Vous devez créer une classe **HotBeverage** avec les attributs 'name', 'price' et 'resistance' et créer des "getter" (méthodes permettant de retourner les valeurs) pour toutes les propriétés.



Depuis PHP8, les variables "attributs" d'une classe tel qu'on les entend en Orienté Objet peuvent être nommées de deux façons en PHP. A vous de trouver ces deux types de variables et de voir ce qui correspond le mieux à notre cas !

Vous devez créer deux classes qui étendent la classe **HotBeverage**, **Coffee** et **Tea**, les deux attribuant une valeur aux attributs 'name', 'price' et 'resistance' et ayant en plus les attributs privés 'description' et 'comment' qui toutes les deux devraient avoir une valeur et un getter.


Vous devez modifier la méthode **createFile(HotBeverage \$text)** de la classe **TemplateEngine** afin qu'elle prenne en paramètre un objet **HotBeverage** et, en utilisant le fichier template **template.html**, créer un fichier HTML statique ayant le nom de la classe de l'objet **HotBeverage** en remplaçant les paramètres dans le template avec la valeur des attributs de l'objet. La classe PHP **ReflectionClass** devrait être utilisée afin de retrouver les attributs de la classe et ensuite récupérer leurs valeurs en utilisant des appels aux getters respectifs.

Le fichier **index.php** devrait créer un objet **Coffee** et un objet **Tea** et appeler la fonction **createFile** pour les deux objets.



# Chapitre VII

## Exercice 03

	Exercice : 03
Exercice 03 : Générer HTML	
Dossier de rendu : <i>ex03/</i>	
Fichiers à rendre : <code>TemplateEngine.php</code> , <code>index.php</code> , <code>Elem.php</code>	
Fonctions Autorisées :	

Vous devez créer une classe **Elem** représentant une balise HTML. La classe devrait avoir un constructeur avec les paramètres suivants :

- element - le nom de la balise HTML
- content - le contenu de la balise (optionnel)

La classe devrait supporter les balises HTML suivantes : meta, img, hr, br, html, head, body, title, h1, h2, h3, h4, h5, h6, p, span, div.

La classe devrait également contenir la méthode **pushElement** permettant d'ajouter un nouvel élément au contenu de la balise. Une méthode **getHTML** devrait retourner le code HTML de la balise. Si un élément/balise contient d'autres éléments, le code HTML des sous-éléments devrait aussi être inclus.

Exemple :

```
$elem = new Elem('html');
$body = new Elem('body');
$body->pushElement(new Elem('p', 'Lorem ipsum'));
$elem->pushElement($body);

echo $elem->getHTML();
```

Devrait générer le code HTML suivant :


```
<html>
  <body>
    <p>Lorem ipsum</p>
  </body>
</html>
```

Vous devez modifier la classe **TemplateEngine** en y ajoutant un constructeur qui prend en paramètre un objet de la classe **Elem**. La méthode **createFile(\$fileName)** devra retourner le code HTML généré par l'objet dans un fichier.

Le fichier **index.php** devrait créer plusieurs objets **Elem** et sauvegarder le contenu généré dans un fichier.

# Chapitre VIII

## Exercice 04

	Exercice : 04
Exercice 04 : Générer HTML 2	
Dossier de rendu : <i>ex04/</i>	
Fichiers à rendre : <code>TemplateEngine.php, index.php, Elem.php, MyException.php</code>	
Fonctions Autorisées :	

Vous devez modifier la classe **Elem** de l'exercice précédent afin de lever l'exception **MyException** si une balise non autorisée est fournie en paramètre dans le constructeur.

Vous devez ajouter la prise en charge des balises suivantes : table, tr, th, td, ul, ol, li.

Modifiez également le constructeur afin de prendre un nouveau paramètre optionnel, **attributes**, qui contiendra un tableau avec les attributs à ajouter dans le rendu HTML de la balise.

Exemple :

```
$elem = new Elem('html');
$body = new Elem('body');
$body->pushElement(new Elem('p', 'Lorem ipsum', ['class' => 'text-muted']));
$elem->pushElement($body);

echo $elem->getHTML();


$elem = new Elem('undefined'); // Leve une exception de type MyException$
```

Devrait générer le code HTML suivant :

```
<html>
  <body>
    <p class="text-muted">Lorem ipsum</p>
  </body>
</html>
```

# Chapitre IX

## Exercice 05

	Exercice : 05
Exercice 05 : Valider HTML	
Dossier de rendu : <i>ex05/</i>	
Fichiers à rendre : <code>TemplateEngine.php</code> , <code>index.php</code> , <code>Elem.php</code> , <code>MyException.php</code>	
Fonctions Autorisées :	

Vous devez ajouter une nouvelle méthode à la classe **Elem**, appelée **validPage()**, qui retournera soit *true* soit *false* selon que le code HTML d'un élément donné est une page HTML valide.

Cette méthode devrait traverser l'ensemble des éléments et vérifier que les conditions suivantes sont vraies :

- l'élément/balise parent est **html** et contient exactement un seul élément **head** suivi par un élément **body**
- l'élément **head** devrait contenir un seul élément **title** et un seul élément **meta charset**
- la balise **p** ne devrait contenir aucune autre balise, seulement du texte
- la balise **table** ne devrait contenir que des balises **tr**, ces dernières contenant que des balises **th** ou **td**
- les éléments **ul** et **ol** ne devrait contenir que des éléments **li**

Vous devez ajouter des tests afin de valider si toutes les conditions ci-dessus sont respectées.

# Chapitre X

## Rendu et peer-évaluation

Rendez votre travail dans votre dépôt Git comme d'habitude. Seul le travail présent dans votre dépôt sera évalué en soutenance. Vérifiez bien les noms de vos dossiers et de vos fichiers afin que ces derniers soient conformes aux demandes du sujet.



L'évaluation se déroulera sur l'ordinateur du groupe évalué.