

Activité POO: Helpers Class

Objectif

Comme vous avez pu le voir, nous pouvons intégrer du PHP dans du HTML. Cependant, ça devient très vite difficile à lire avec les ouvertures et fermetures successifs des balises PHP et l'alternance entre PHP et HTML.

Nous allons minimiser autant que possible cette alternance en créant des classes PHP qui vont générer de l'HTML directement.

Vous allez comprendre par exemple. Imaginez un formulaire d'une page de modification.

```
<form action="<?php echo $action; ?>" method="POST">
    <input name="nom" value="<?php echo $nom ?>" >
    <input name="prenom" value="<?php echo $prenom ?>" >
    <input type="submit" value="Modifier" >
</form>
```

Les variables `$action`, `$nom` et `$prenom` possèdent des valeurs directement récupérées de la base de données.

Comme vous pouvez le voir, l'HTML et le PHP sont mêlés.

Pour minimiser cela, nous pouvons créer une classe `Form` qui gère la génération du formulaire. On aura un code peu plus lisible.

```
echo $form = new Form();
echo $form->create($action); // créer le début du formulaire
echo $form->text('nom',$nom); // créer un input de type texte avec comme valeur
par défaut $nom
echo $form->text('prenom',$prenom); // créer un input de type texte avec comme
valeur par défaut $prenom
echo $form->submit('Modifier'); //Créer un bouton pour soumettre le formulaire
se nommant Modifier
echo $form->end(); //fermer le formulaire
```

Nous jonglons moins avec l'ouverture et les fermetures de balises php `<?php ?>`.

Il y a bien d'autres avantages d'utiliser du PHP pour générer de l'HTML, notamment bénéficier de la puissance de PHP pour personnaliser le code généré selon un simple paramètre :)

Activité

Créer une classe `Form` qui s'occupera de générer les différents éléments du formulaires. Cette classe devra avoir des méthodes et des propriétés afin de pouvoir générer :

- le début d'un formulaire `<form ...>`
- Un input text `<input type="text"...>`
- Un select `<select ...> ...</select>`
- Un bouton submit `<button type="submit"></button>`
- Un textarea `<textarea ...> ...</textarea>`
- Un radio button `<input type="radio"...>`
- Une checkbox `<input type="checkbox"...>`

Créer une class `Html` qui va générer différents éléments html :

- Lier des Fichiers CSS `<link rel="stylesheet" ...>`
- Créer des Balises meta `<meta ...>`
- Lier des Images ``
- Créer des Liens ``
- Lier des Fichiers Javascript

Vous devez créer ces classes en pensant qu'elles devront être utilisées par un autre développeur. Utilisez des noms de propriétés et de méthodes qui ont du sens, restez simple et commentez votre code.

Pour la classe `Form` Vous n'êtes pas obligé de vous baser sur l'exemple plus haut.

Pour aller plus loin

Créer une classe `Validator` qui va servir à valider les données envoyées par le formulaire. Cette classe contiendra des méthodes qui pourront valider une chaîne de caractère, un entier, un nombre à virgule etc.

Ressources

[Introduction à la POO](#) et à l'[Autoloading](#) de classes php