WordPress : les templates parts

Templates parts : définition

Sous WordPress, un **template part** est **une fonctionnalité qui permet de réutiliser du code HTML ou PHP dans plusieurs endroits de votre thème WordPress**.

Cela facilite la gestion et la mise à jour de votre site, car vous pouvez définir une partie du code une fois et l'inclure dans plusieurs fichiers de modèle.

Un template part est généralement **utilisé pour des éléments récurrents ou communs** sur votre site, tels que l'**en-tête**, le **pied de page**, la **barre latérale** ou des sections spécifiques répétitives.

En les regroupant dans des templates parts, vous pouvez **éviter la duplication de code** et maintenir une structure claire et modulaire pour votre thème.

Principe du **Don't Repeat Yourself (DRY)**.

Utilisation des templates parts

Pour utiliser un template part, vous devez suivre les étapes suivantes :

1. Créez un fichier PHP contenant le code HTML ou PHP que vous souhaitez réutiliser. Par exemple, vous pouvez créer un fichier nommé « **newsletter.php** » pour le formulaire d'inscription à la newsletter de votre site.
2. Dans les fichiers de modèle où vous souhaitez inclure cette partie, utilisez la fonction **`get_template_part()`**. Par exemple, pour inclure le formulaire d'inscription à la newsletter de votre site, vous pouvez ajouter la ligne suivante à votre fichier de modèle : **`get_template_part('newsletter');`**L'argument passé à la fonction **`get_template_part()`** est le **nom du fichier (sans l'extension .php)** que vous avez créé pour votre template part.
3. Répétez cette étape pour inclure d'autres template parts dans d'autres fichiers de modèle selon vos besoins.

Lorsque WordPress rencontre la fonction **`get_template_part()`**, il recherche le fichier correspondant dans le thème actif.

Si le fichier est trouvé, son contenu sera inclus à cet endroit. Sinon, il ne se passera rien.

Par rapport aux fonctions PHP **`include()`** ou **`require()`** la fonction **`get_template_part()`** ne retourne pas d'erreur si elle ne trouve pas le fichier de template devant être chargé.

Il est également courant d'utiliser des paramètres dans les template parts pour personnaliser leur contenu en fonction du contexte dans lequel ils sont inclus.

Vous pouvez passer des arguments à `get_template_part()` et les récupérer dans le fichier du template part à l'aide de variables.

En utilisant des template parts, vous pouvez améliorer la modularité de votre thème WordPress et rendre la gestion de votre site plus efficace en évitant la duplication de code et en réutilisant du contenu à plusieurs endroits.

Exemple : un formulaire d'inscription à la newsletter (fichier newsletter.php)

```
<form class="newsletter-form">
    <p>Profitez d'offres exclusives, restez informé des dernières actualités WordPress et accédez en avant-première à du contenu premium.</p>
    <input type="email" name="email" placeholder="Adresse e-mail">
    <input type="submit" value="S'abonner">
</form>
```

Le fichier pourra ensuite être inclus via get_template_part() dans vos modèles (ex. single.php) :

```
<?php get_template_part( 'newsletter' ); ?>
```

Attention ! Ne mettez pas l'extension **.php** du template à charger via l'appel à la fonction **`get_template_part()`**.

Si besoin, vous pouvez organiser vos templates parts en dossiers et les charger en spécifiant la structure d'accès au fichier de template.

```
<?php get_template_part('parts/newsletter'); ?>
```

Le cas des formats de publications (posts formats)

Les **posts formats** sont des formats spécifiques pour les **articles**, tels que la **vidéo**, l'**image**, la **galerie**, la **citation**, etc.

Voici comment vous pouvez les traiter avec `get_template_part()` :

1. Identifiez les post formats que vous souhaitez traiter différemment dans votre thème. Par exemple, vous pourriez décider de personnaliser l'affichage des post formats « **vidéo** » et « **image**« .
2. Créez des fichiers de template correspondants pour chaque post format que vous souhaitez personnaliser. Par exemple, vous pouvez créer les fichiers « **content-video.php** » et « **content-image.php**« .
3. Dans votre boucle d'affichage des articles (généralement dans le fichier « **single.php** » ou « **index.php**«), utilisez **`get_template_part()`** en spécifiant le nom du fichier correspondant au post format en cours. Par exemple, pour afficher le contenu d'un post format vidéo, vous pouvez utiliser **`get_template_part('content', 'video');`**. Assurez-vous d'utiliser le **slug** du post format (par exemple, '**video**' ou '**image**') comme partie du nom de fichier, précédé de « content-« .
4. Créez les fichiers de template correspondants avec le code HTML et PHP approprié pour chaque post format. Par exemple, dans « **content-video.php**« , vous pouvez personnaliser l'affichage du contenu vidéo spécifique.

Vous pouvez également traiter les différents cas de figure en utilisant la fonction

get_post_format() de WordPress.

```
<?php
        // Ex. si le fomat de publication sélectionné est "audio", WordPress
ira chercher le modèle d'affichage content-audio.php
        get_template_part('content', get_post_format());
?>
```

Utiliser les templates parts pour factoriser les modèles de pages

Si vous utilisez le même rendu pour la page d’accueil, des archives, des résultats de recherches… vous pouvez très bien fortement réduire la longueur du code source de ces différents templates en utilisant la fonction **get_template_part()**.

Exemple dans le fichier search.php :

```
<?php get_template_part('archive'); ?>
```