

## Chapitre 4

# Les thèmes basés sur des blocs

### 1. Introduction

Comme nous l'avons vu au chapitre Introduction à WordPress section Apparence - Thèmes, WordPress dispose de différents thèmes. Dans ce chapitre, nous allons voir comment fonctionnent les thèmes basés sur des blocs, dits *block themes* en anglais.

Depuis l'intégration de Gutenberg, WordPress met à disposition de nouveaux outils et une nouvelle façon de concevoir un site, en intégrant le Full Site Editing (édition complète du site) qui permet, tel un page builder, de pouvoir concevoir un site grâce à un système de blocs, et ainsi modifier et personnaliser toutes les parties de votre site.

Ainsi, vous êtes en mesure d'utiliser des blocs dans les modèles fournis par un thème ou une extension, par exemple une page d'archive ou une page 404. Enfin, vous avez la possibilité de créer vos propres modèles.

C'est bien une nouvelle façon de penser qui change complètement de ce qui se faisait jusqu'alors sur WordPress : vous avez un nouveau fichier `theme.json` et une interaction avec les fichiers qui est différente.

Pour savoir comment les fichiers sont structurés dans les thèmes basés sur des blocs, allez au chapitre WordPress et PHP - Les fichiers et dossiers des thèmes basés sur des blocs.

Mais WordPress vous laisse le choix avec les anciennes méthodes de conception de sites qui représentent encore une belle part de marché. En effet, tout ce système est encore nouveau et des fonctionnalités sont à chaque fois améliorées. De nouveaux thèmes et de nouvelles extensions sont ainsi créés pour ce système de blocs.

WordPress se voulait modulable, il l'est, d'autant plus avec cette nouvelle façon de créer un site !

## 2. Qu'est-ce qu'un thème basé sur des blocs ?

Un thème WordPress basé sur les blocs fonctionne avec des modèles entièrement composés de blocs avec Gutenberg de telle sorte que l'éditeur peut être utilisé pour modifier non seulement le contenu des publications (pages, articles...), mais aussi toutes les zones du site : en-tête, pied de page, barres latérales... C'est ce que l'on appelle le Full Site Editing.

La structure des fichiers est différente, comme vous l'avez vu au chapitre WordPress et PHP - Les fichiers et dossiers des thèmes basés sur des blocs, et la différence principale avec les thèmes classiques réside dans le fait que les modèles et éléments de modèles relatifs à la hiérarchie classique de WordPress sont au format HTML au lieu d'être au format PHP et qu'ils sont intégralement composés de blocs.

Comme vous avez pu le voir, l'outil de personnalisation n'est pas accessible avec un thème basé sur des blocs, à moins que vous utilisiez une extension ou un thème qui nécessite qu'il soit activé. En effet, les blocs proposés dans l'éditeur de modèles ou l'éditeur de sites, vous permettent d'effectuer les mêmes modifications. Par exemple, il existe désormais les blocs **Titre du site**, **Description du site** et **Logo du site**. Vous pouvez également sélectionner des couleurs pour votre site dans le panneau de gestion des styles, accessible depuis le Tableau de bord via le menu **Apparence - Éditeur (bêta)**, en cliquant sur le bouton **Styles**.

Actuellement, l'option permettant d'ajouter une favicône ne fait pas partie des fonctionnalités disponibles, mais elle sera certainement incluse dans une version ultérieure. Pour l'instant, la favicône est automatiquement définie comme étant l'image que vous utilisez dans le bloc **Logo du site**.

### 3. Qu'est-ce qu'un modèle de bloc ?

Un modèle de bloc est composé d'une liste de blocs. Tout bloc WordPress peut être utilisé dans un modèle. Les modèles peuvent également partager des parties de leur contenu en utilisant les « éléments de modèles ». Par exemple, tous les modèles de blocs peuvent utiliser le même en-tête en incluant un élément de modèle distinct **header.html** ou **footer.html**.

Le commentaire suivant appelle le fichier **header.html** contenu dans le dossier **Parts** :

```
<!-- wp:template-part {"slug":"header","tagName":"header"} /-->
```

Ces différents commentaires ressemblent à des commentaires HTML, et ils ont une fonction spécifique : délimiter des blocs. Les **délimiteurs de blocs** indiquent à WordPress quel bloc doit être rendu à l'écran. Ils fournissent également des valeurs pour les propriétés du bloc dans un objet JSON. Ces propriétés déterminent la manière dont le bloc doit être rendu à l'écran.

Voici quelques exemples pour comprendre le code de l'exemple suivant :

```
<!-- wp:group --> correspond à un bloc groupe.
```

```
<!-- wp:query --> correspond à un bloc boucle de requête.
```

```
<!-- wp:post-title /--> correspond à un bloc titre.
```

```
<!-- wp:post-excerpt /--> correspond à un bloc résumé.
```

```
<!-- wp:post-date /--> correspond à un bloc date.
```

```
<!-- wp:spacer --> correspond à un bloc saut de ligne.
```

```
<!-- wp:query-pagination --> correspond à un bloc pagination.
```

Vous remarquerez que certaines balises ont besoin d'une balise fermante, tandis que d'autres sont inline, tout comme en HTML.

Voici un exemple de modèle de bloc avec le fichier **index.html** dans le dossier **Templates** du thème Twenty Twenty-Three :

```
<!-- wp:template-part {"slug":"header","tagName":"header"} /-->

<!-- wp:group {"tagName":"main","layout":{"type":"constrained"}} -->
<main class="wp-block-group">
    <!-- wp:query {"query":{"perPage":10,"pages":0,
"offset":0,"postType":"post","order":"desc","orderBy":"date","author":
"","search":"","exclude":[],"sticky":"","inherit":true,"taxQuery":
null,"parents":[],"displayLayout":{"type":"flex","columns":3},
"align":"wide","layout":{"type":"default"}}} -->
    <div class="wp-block-query alignwide">
        <!-- wp:post-template {"align":"wide"} -->
            <!-- wp:post-featured-image
{"isLink":true,"width":"100%","height":
"max(15vw, 30vh)","align":"wide"} /-->
                <!-- wp:post-title {"isLink":true,"align":"wide"} /-->
                <!-- wp:post-excerpt /-->
                <!-- wp:post-date {"isLink":true} /-->

                <!-- wp:spacer {"height":"var(--wp--preset--spacing
--70)"} -->
                    <div style="height:var(--wp--preset--spacing--70)"
aria-hidden="true" class="wp-block-spacer"></div>
                    <!-- /wp:spacer -->
                <!-- /wp:post-template -->

                <!-- wp:query-pagination {"paginationArrow":
"arrow","align":"wide","layout":{"type":"flex","justifyContent":
"space-between"}} -->
                    <!-- wp:query-pagination-previous /-->
                    <!-- wp:query-pagination-next /-->
                <!-- /wp:query-pagination -->
            </div>
        <!-- /wp:query -->
    </main>
<!-- /wp:group -->

<!-- wp:template-part {"slug":"footer","tagName":"footer"} /-->
```

### 4. Écrire et modifier des modèles

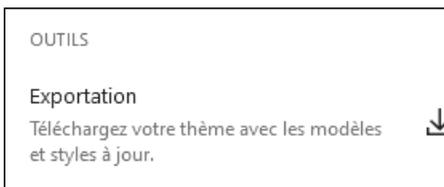
Pour écrire, modifier ou exporter des modèles, vous devez le faire via l'administration du site. Pour cela, vous devez accéder à la modification des modèles via le menu **Apparence - Éditeur (bêta)**. Vous pouvez ajouter les blocs que vous souhaitez et créer la page qui vous convient. Pour le détail de tous les blocs, voir le chapitre Introduction à WordPress - L'administration et les menus - Articles - Gutenberg et le système de blocs.

Vous avez la possibilité de passer en mode éditeur de code pour récupérer le code HTML du modèle. Vous pouvez également cliquer sur l'outil **Exporter** accessible via le bouton **Options** (trois points verticaux), situé à la droite de la barre supérieure de l'éditeur de site.

Vous pourrez ainsi télécharger un fichier ZIP contenant tous les modèles et éléments de modèles que vous avez créés sous l'éditeur et que vous pourrez placer dans le dossier de votre thème.

Vous pouvez ainsi créer des thèmes avec des modèles personnalisés et créer vos propres modèles pour le dossier **Templates** et le dossier **Parts**, si votre thème en dispose.

Veuillez noter que lorsque vous faites cet export, le balisage du bloc **Élément de modèle** inclut un identifiant de publication (`postID`) qui peut être supprimé en toute sécurité si vous redistribuez votre thème.

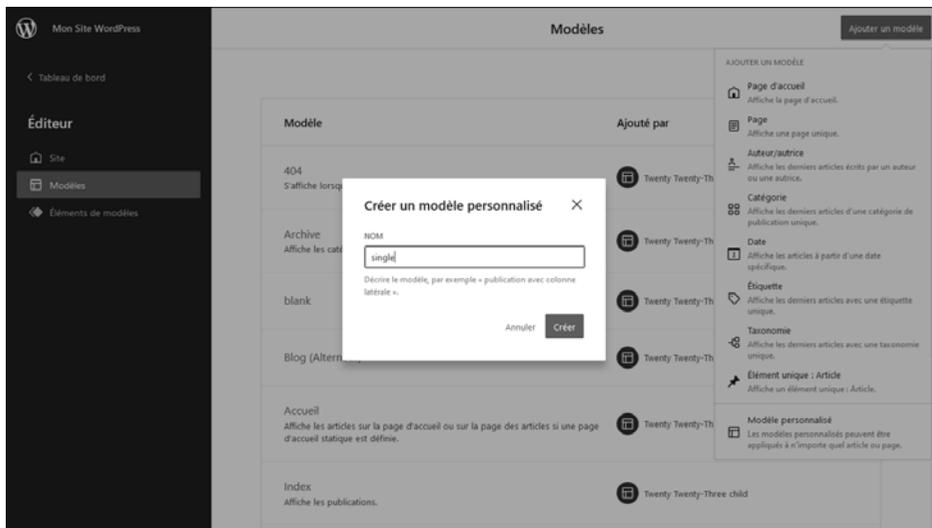


*Menu **Options** section **Outils** onglet **Exportation***

En enregistrant les modèles à partir du menu **Editeur - Modèles** en cliquant sur le logo WordPress, puis sur le bouton en haut à droite **Ajouter un modèle - Modèle personnalisé**, vous pourrez créer un modèle et ainsi le prioriser par rapport aux modèles présents dans le dossier de votre thème.

Par exemple, en utilisant le mot « single » comme titre de votre modèle et en l'enregistrant, ce modèle aura la priorité sur le fichier **single.html** présent dans le dossier de votre thème.

Notez cependant qu'il ne surchargera aucun des modèles ayant une spécificité plus élevée dans la hiérarchie des modèles. La résolution va du plus spécifique au moins spécifique, en recherchant d'abord un modèle de publication personnalisé, puis un modèle de thème, et ce à chaque niveau.



*Pop-up permettant d'ajouter un modèle personnalisé*

Voici un site vous permettant d'avoir toutes les références des blocs :  
<https://fullsiteediting.com/block-reference>

### 5. Le fichier theme.json

Depuis WordPress 5.8 et les nouveaux thèmes basés sur les blocs, un nouveau fichier a vu le jour : `theme.json`. Celui-ci permet de gérer des styles prédéfinis et de configurer son site. Depuis la version 5.9, une v2 de ce fichier est apparue.

Grâce à ce fichier, vous pourrez notamment gérer les couleurs par défaut, la taille de police, définir la mise en page par défaut de l'éditeur comme les largeurs, les alignements, préréglager les blocs pour des blocs spécifiques, activer ou supprimer des fonctionnalités et définir les modèles de pages auxquels le thème peut faire appel.

Voir la documentation complète sur l'éditeur de blocs :  
<https://developer.wordpress.org/block-editor>

Voir la documentation sur le fichier **theme.json** :  
<https://developer.wordpress.org/block-editor/how-to-guides/themes/theme-json>

### 6. La structure de base

Le fichier `theme.json` a une structure de base :

```
{
  "version": 2,
  "settings": {},
  "styles": {},
  "customTemplates": {},
  "templateParts": {}
}
```

`version` permet de spécifier la version de la structure du fichier. La version est actuellement la 2, introduite lors de la version 5.9 de WordPress. Elle permet de spécifier le format de la structure du fichier **theme.json**. Si vous avez déjà utilisé la v1, vous n'avez pas besoin de mettre à jour la version du fichier v1 vers la v2, car elle sera mise à jour automatiquement. Il est important d'inclure l'attribut `version`, sinon les données seront analysées en tant que "version 0" et les données risquent d'être mal interprétées.