

WordPress : thème enfant

Principe de fonctionnement d'un thème enfant

Utilité d'un thème enfant

Si vous souhaitez avoir une personnalisation poussée qui va au-delà de ce que permet votre thème (modification du code source des modèles de pages HTML / PHP, ajout de style CSS...), la création d'un thème enfant est indispensable !

Remarque : si vous comptez uniquement utiliser les options du thème, de votre constructeur de page... sans personnalisation poussée, l'usage d'un thème enfant est inutile.

La problématique de mise à jour

La problématique : si vous modifiez directement le code source dans le thème principal (thème parent), en cas de mise à jour, celui-ci sera écrasé (les fichiers sont remplacés par les nouveaux).

Lorsqu'un thème est mis à jour, l'intégralité de ses fichiers sont rechargés.

Par conséquent, vos modifications sur mesures réalisées sur le style CSS, les templates PHP... seront écrasées par les fichiers récupérés sur le dépôt officiel de l'éditeur.

C'est là qu'intervient le thème enfant ! Il vous permet de conserver vos modifications au cours des différentes mises à jour du thème que vous aurez sélectionné.

Si vous utilisez exclusivement l'éditeur CSS présent depuis **Apparence > Personnaliser > CSS additionnel** pour personnaliser votre thème, vous n'avez pas besoin de générer un thème enfant.

Dans ce cas, le code CSS est enregistré en **base de données** et n'est donc pas écrasé en cas de mise à jour du thème.

Override

Le **thème enfant** permet d'effectuer des retouches sans impacter le thème initial (parent).

Le **thème enfant** hérite du design du thème parent, mais peut s'en distinguer via ses propres directives de **styles (CSS)** et **fonctionnalités (code personnalisé)**.

Le thème enfant peut se distinguer du thème parent via la surcharge (override) des fichiers et dossiers du thème parent :

- style CSS personnalisé
- fonctionnalités sur mesures (**functions.php**)
- modèle de pages sur mesures (templates)

Le principe : il vous suffit de copier le fichier de modèle du thème parent (**template PHP**) et de le coller dans le thème enfant, en conservant la même structure de dossiers et sous-dossiers, le cas

échéant.

Le fichier ainsi dupliqué pourra alors être modifié à votre guise.

Les fichiers modifiés dans le thème enfant seront prioritaires par rapport à ceux du thème parent.

Certains constructeurs comme **Divi** ou **Elementor (Pro)** vous permettent de traiter la personnalisation complète des modèles de page (templates), sans avoir besoin d'avoir recours au mécanisme de surcharge.

Alternative à l'utilisation d'un thème enfant : utiliser l'extension [Code Snippets](#), qui facilite la gestion de vos ajouts de code PHP (sous forme de **snippets**) directement depuis l'interface de WordPress avec activation ou désactivation rapide d'un "snippet" selon les besoins.

Créer un thème enfant manuellement

La création d'un thème enfant WordPress est relativement simple et nécessite quelques étapes de base.

Générer manuellement un thème enfant

[Theme-enfantTélécharger](#)

Voici un guide étape par étape pour créer un thème enfant :

1. **Créez un nouveau répertoire pour votre thème enfant :** dans le répertoire des thèmes de votre installation WordPress, créez un nouveau répertoire pour votre thème enfant. Par exemple, si votre thème parent s'appelle "**MyTheme**", vous pouvez nommer votre répertoire de thème enfant "**MyTheme-child**".
2. **Créez un fichier style.css :** à l'intérieur du répertoire du thème enfant, créez un fichier **style.css**. Ce fichier est utilisé pour décrire les informations du thème enfant et ajouter votre propre style CSS. Ouvrez le fichier style.css et ajoutez le code ci-après.
3. **Créez un fichier functions.php (optionnel) :** si vous souhaitez ajouter des fonctionnalités personnalisées à votre thème enfant, vous pouvez créer un fichier **functions.php** dans le répertoire du thème enfant. Vous pouvez y ajouter votre code PHP personnalisé pour étendre les fonctionnalités du thème parent.
4. **Créez un fichier screenshot.png (optionnel) :** une image d'illustration de votre thème

Fichier style.css

```
/*  
Theme Name: Nom du thème enfant  
Template: nom-du-theme-parent  
*/
```

Remplacez "**Nom du thème enfant**" par le nom de votre thème enfant et "**nom-du-theme-parent**" par le répertoire du thème parent.

Par exemple, si le thème parent est **MyTheme**, remplacez "**nom-du-theme-parent**" par **MyTheme**.

Attention ! Ne mettez pas d'espace avant les deux points, sinon le commentaire ne pourra être interprété.

Par défaut, le style CSS du thème parent n'est pas chargé, il vous faut l'appeler manuellement depuis le fichier **functions.php**.

```
<?php
function mytheme_enqueue_style(){
    wp_enqueue_style( 'my-theme-parent-style', get_template_directory_uri() .
'/style.css' );
}

add_action( 'wp_enqueue_scripts', 'mytheme_enqueue_style' );
```

Activer le thème enfant

Une fois que vous avez créé votre thème enfant avec les fichiers nécessaires, vous pouvez l'activer dans le tableau de bord de WordPress.

Accédez à “**Apparence**” → “**Thèmes**” dans votre tableau de bord WordPress, et vous devriez voir votre thème enfant répertorié.

Activez-le pour appliquer les modifications.

À partir de maintenant, WordPress chargera la feuille de style (fichier **style.css**) du thème enfant en plus de celle du thème parent, il lira le fichier **functions.php** en plus de celui du parent, et vous pourrez remplacer les modèles de templates en les copiant dans le répertoire du thème enfant.

Générateur de thème enfant

URL de votre site web ou nom du dossier du thème parent pour générer le thème enfant.

Pensez à installer / activer votre thème enfant depuis le menu **Apparence > Thèmes**.

Ne supprimez surtout pas le thème parent !

Une alternative : utiliser l'extension **WPS Child Theme Generator**.

Certains éditeurs de thèmes (ex. **Ocean WP**, ...) fournissent directement un thème enfant.

Surcharge (override)

Dans certains cas où votre surcharge CSS ne fonctionnerait pas, vous pourriez être contraint d'utiliser l'instruction CSS **!important** pour forcer la prise en compte d'un style récalcitrant.

Exemple d'utilisation de l'instruction CSS **!important**

```
.my-element {  
    color: red !important;  
}
```

Dans cet exemple, la classe **.my-element** aura une couleur de texte rouge, même si d'autres règles CSS sont appliquées à cet élément.

L'ajout de **!important après la valeur de **color** donne la priorité à cette règle et force la couleur rouge à être appliquée, même si d'autres règles moins spécifiques ou sans **!important** sont présentes.**

Il est important de noter que l'utilisation excessive de **!important** peut entraîner une mauvaise gestion des styles et rendre le code CSS difficile à maintenir.

Il est recommandé d'utiliser **!important** avec parcimonie et d'explorer d'abord des solutions alternatives, telles que l'amélioration de la spécificité du sélecteur CSS ou la réorganisation des règles CSS existantes, avant de recourir à **!important**.

Mise à jour et compatibilité

Selon la personnalisation que vous aurez apportée à votre thème enfant, il peut arriver que les mises à jour successives entraînent des fonctionnements altérés.

Ce sera notamment le cas si votre thème enfant utilise des fonctions qui sont maintenant obsolètes.

Il vous sera alors peut-être nécessaire de mettre à jour le code de votre thème enfant.