

WordPress : hiérarchie des templates (template hierarchy)

La **hiérarchie des templates WordPress** fait référence à l'ordre dans lequel WordPress recherche et utilise différents fichiers de modèle pour afficher les pages sur un site.

Lorsqu'une page est chargée, WordPress suit une structure de recherche spécifique pour déterminer quel fichier de modèle doit être utilisé pour afficher le contenu.

Important à savoir :

- Les pages standard sont des **pages**.
- La liste des articles d'un blog correspond aux **archives**.
- Lorsqu'on clique sur une catégorie, cela correspond également à une **archive**.
- L'affichage d'un article individuel est considéré comme un élément unique (**single**).

Le schéma template hierarchy (avec lecture de gauche à droite, en commençant par le type de page qui va être affiché)



Template Hierarchy WordPress

Choix du type de page à afficher :

- Archives par catégorie, mot clé, auteur, date... : archive.php

- **Page singulière** : affichage d'un contenu d'une **page** avec **page.php** ou d'un **article** avec **single.php**
- **Page d'accueil du site** : **front-page.php** si le site est configuré pour afficher une page statique
- **Page d'accueil du blog** : un type d'archive particulier répondant via **home.php**
- **Page d'erreur 404** : en cas de page non trouvée (lien brisé) via **404.php**
- **Page des résultats d'une recherche** : **search.php**

Pour rappel, les réglages de la page d'accueil s'effectuent depuis **Réglages > Lecture > La page d'accueil affiche**.

WordPress va vérifier s'il existe un fichier de modèle plus spécifique créé par le développeur du thème.

Si un tel fichier n'est pas trouvé, WordPress utilisera le modèle plus générique en tant que solution de repli.

Ex. si **front-page.php** n'est pas présent pour afficher la page d'accueil, ce sera **page.php** qui sera utilisé.

En dernier recours, ce sera le template **index.php** qui sera utilisé, d'où le fait que celui-ci soit obligatoire à la création d'un thème WordPress.

Voici un aperçu simplifié de la hiérarchie des templates WordPress, de la priorité la plus élevée à la plus basse :

- **template-{slug}.php** : un fichier de modèle spécifique basé sur le **slug** (identifiant) d'une page, par exemple **template-about.php** pour la page **À propos**.
- **template-{id}.php** : un fichier de modèle spécifique basé sur l'**ID d'une page**, par exemple **template-42.php** pour la page avec l'**ID 42**.
- **page-{slug}.php** : un fichier de modèle spécifique basé sur le **slug** (identifiant) d'une page, par exemple **page-about.php** pour la page **À propos**.
- **page-{id}.php** : un fichier de modèle spécifique basé sur l'**ID d'une page**, par exemple **page-42.php** pour la page avec l'**ID 42**.
- **single-{post_type}.php** : un fichier de modèle spécifique basé sur le **type de publication (post_type)**, par exemple **single-portfolio.php** pour les articles de type **Portfolio**.
- **single.php** : le fichier de modèle par défaut pour les articles de blog individuels.
- **index.php** : le fichier de modèle par défaut pour les autres types de contenu, y compris les archives, les pages d'accueil, les pages de recherche, etc.
- **archive-{post_type}.php** : un fichier de modèle spécifique pour les **archives d'un type de publication** particulier, par exemple **archive-portfolio.php** pour les archives du type de publication **Portfolio**.

Récupérer l'ID d'une publication

Pour rappel, l'**ID d'une publication** peut être récupéré depuis le tableau de bord WordPress, en accédant à la liste des publications dans le menu **Articles** ou **Pages**.

Survolez une publication avec votre souris et vous verrez l'**ID** apparaître dans l'URL affichée en bas de votre navigateur.

Par exemple, **post.php?post=123&action=edit** où **123** est l'**ID de la publication**.

Méthode déconseillée : utiliser le **slug** ou l'**ID** dans la gestion de vos templates peut provoquer des problèmes. Ces valeurs peuvent être modifiées, par exemple si la page est supprimée puis recréée (ce qui modifie l'**ID**) ou lors d'un changement de permalien (ce qui modifie le **slug**).

Le cas des templates d'archives

Archive	Template
Archive des catégories d'articles	category-\$slug.php ou category-\$id.php Ex. une catégorie Promotion qui aurait un affichage différent traité par le template category-promotion.php
Archive d'auteurs	author-\$nicename.php ou author-\$id.php avec \$nicename qui correspond au pseudo de l'utilisateur.
Archive des contenus personnalisés	archive-\$posttype.php avec \$posttype le nom du type de contenu personnalisé.
Archive par calendrier	date.php

Body classes

Le Template Hierarchy joue également un rôle dans la génération des classes du corps du contenu (**body classes**), c'est-à-dire les classes incluent dans la balise **<body>** de votre page HTML.

```
<body <?php body_class(); ?>>
```

Exemple de rendu :

```
<body class="sfwd-topic-template-default single single-sfwd-topic postid-6785 logged-in admin-bar wp-custom-logo wp-embed-responsive footer-on-bottom hide-focus-outline link-style-standard content-title-style-normal content-width-normal content-style-boxed content-vertical-padding-show non-transparent-header mobile-non-transparent-header learndash-cpt learndash-cpt-sfwd-topic learndash-template-ld30 learndash-cpt-sfwd-topic-6785-current learndash-cpt-sfwd-courses-1207-parent learndash-cpt-sfwd-lessons-3308-parent learndash-embed-responsive customize-support header-is-fixed">
```

Ceci permet alors de générer des règles CSS spécifiques, exemple pour une classe en particulier (**single...**).

Grâce au Template Hierarchy, le développeur a la possibilité de créer des templates en fonction du type de contenu à afficher.

Il est important de noter que la hiérarchie des templates WordPress peut varier en fonction des thèmes et des configurations spécifiques.

Certains thèmes peuvent inclure des fichiers de modèle supplémentaires ou personnaliser la hiérarchie par défaut pour répondre à leurs besoins spécifiques.

En comprenant la hiérarchie des templates WordPress, vous pouvez créer des fichiers de modèle personnalisés pour afficher des pages spécifiques ou des types de contenu particuliers de manière

contrôlée et flexible.