

Wordpress

création de thèmes

Créer un thème

- Créer un dossier dans `wp_content/themes`
- Créer un fichier `style.css` dans ce dossier
- Indiquer le nom du thème dans les commentaires



```
/*  
Theme Name: Mon super theme  
*/
```

Dans l'administration de Wordpress, le thème est incomplet, il manque `index.php`

Page principale

Dans le dossier du thème, le fichier `index.php` représente l'affichage par défaut de toutes les pages du site



```
<h1> Page par défaut </h1>
```

Insertion des en-têtes en pieds

Les fonctions `wp_head()` et `wp_footer()` insèrent les éléments récurrents à chaque page. C'est par ce biais que l'on obtient les scripts, les styles et même les barres de navigation.



```
<?php wp_head(); ?>

<h1>Hello</h1>

<?php wp_footer(); ?>
```

Affichage du contenu

```

● ● ●
<?php wp_head(); ?>

<?php
// Boucle sur l'ensemble des articles
while(have_posts()):
    // article en cours
    the_post();
?>


    <h2><?php the_title() ?></h2>

<?php endwhile ?>

<?php wp_footer(); ?>
```

Les actions


Lors du traitement de la requête http Wordpress réalise quantités d'opérations, les étapes de ce processus sont jalonnées de hooks, des événements sur lesquels on peut ajouter notre propre code pour modifier le comportement de Wordpress.



```
// Dans functions.php
function mytheme_init_action(){
    echo "Wordpress est initialisé";
}
// hook et fonction de callback
add_action( "init", "mytheme_init_action" );
```

Les filtres

Les filtres sont similaires aux actions, mais au lieu de simplement réaliser un traitement, ils retournent une valeur qui modifie une variable de Wordpress.

```
  
// Dans functions.php  
  
function my_theme_custom_login_message( $message ) {  
    return "$message <h2>Merci de nous faire confiance</h2>";  
}  
add_filter( "login_message", "my_theme_custom_login_message" );
```

Trouver les hooks disponibles

Deux possibilités, Internet et la doc officielle de Wordpress

Autres liens

- hooks
- filters

“ On peut également inspecter le code de Wordpress et chercher les appels aux fonctions `do_action` et `apply_filter` pour trouver les hooks qui nous intéressent.


Petit défi

Essayer de créer un filtre qui remplace par "***" les gros mots dans les commentaires

- Trouver le filtre qui s'applique à l'affichage du commentaire
- Concevoir l'algorithme chargé du remplacement

“ *Pour pouvoir afficher les articles et les commentaires, copier les fichiers `single.php` et `comments.php` depuis le thème `twenty_twentyone`*

Correction

```
  
// Dans functions.php  
  
function my_theme_bad_words_replace( $comment_text ) {  
    $badWords = [ "con", "débile", "enfoiré" ];  
  
    foreach( $badWords as $word ){  
        $comment_text = str_ireplace( $word, "***", $comment_text );  
    }  
  
    return $comment_text;  
}  
  
add_filter( "comment_text", "my_theme_bad_words_replace" );
```

Les ressources

importation des ressources (css, js, fonts) dans toutes les pages du thème



```
// fichier functions.php
function script_and_styles(){
    // L'uri du thème sur lequel on se trouve
    $uri = get_theme_file_uri();

    // Ajout d'une feuille de style

    wp_enqueue_style("my-theme-bootstrap-css", "$uri/assets/bootstrap.css");

    // Ajout des scripts
    wp_enqueue_script("my-theme-jquery", "$uri/assets/jquery.js");

    wp_enqueue_script(
        "my-theme-bootstrap-js",
        "$uri/assets/bootstrap.js",
        // Tableau des dépendances
        [ "my-theme-jquery" ]
    );
}

// hook pour l'exécution de la fonction
add_action('wp_enqueue_scripts', 'script_and_styles');
}
```

A vous

- Créer votre propre thème
- Ajouter Bootstrap et votre propre fichier css
- Créer une page `index.php` qui affiche les titres des articles sur une grille de deux colonnes

Personnalisation de l'en-tête

Pour personnaliser l'en-tête il faut créer un fichier **header.php** qui contiendra le code du début de chaque fichier. Un bon endroit pour placer ce qui doit être partagé dans chaque page (navigation, barre de titre ...)

Pour que le contenu reste contextuel on peut utiliser les fonctions Wordpress.

La fonction `wp_head` insère les ressources déclarées précédemment dans le hook

`wp_enqueue_scripts`




```
<!DOCTYPE html>
<html <?php language_attributes(); ?>>
<head>
  <meta charset="<?php bloginfo('charset'); ?>">
  <title>
    <?php wp_title(); ?> - <?php bloginfo( 'name' ); ?>
  </title>
  <?php wp_head(); ?>
</head>

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

Le pied de page

Le fichier `footer.php` contient la fin du document html. Comme `header.php`, il est conçu pour définir le contenu partagé par toutes les pages du site.



```
<footer>
    My footer
</footer>

<?php wp_footer(); ?>

</body>
</html>
```

Ajout des en-têtes et pieds à la page index

Les fonctions `get_header` et `get_footer` insèrent leur contenu respectifs dans nos modèles



```
<!-- Insertion de header.php -->
<?php get_header() ?>

<?php
// Boucle sur l'ensemble des articles
while(have_posts()):
    // article en cours
    the_post();
?>

    <h2><?php the_title() ?></h2>

<?php endwhile ?>

<!-- Insertion de footer.php -->
<?php get_footer() ?>
```

Les éléments de page

Élément	fichier	fonction
en-tête	header.php	get_header()
pied de page	footer.php	get_footer()
barre latérale	sidebar.php	get_sidebar()
commentaires	comment.php	comments_template()
barre de navigation	...	wp_nav_menu()

Les "template tags" des articles

Description	fonction
le titre de l'article	<code>the_title()</code>
le texte de l'article	<code>the_content()</code>
l'auteur de l'article	<code>the_author()</code>
le résumé de l'article	<code>the_excerpt()</code>
l'id de l'article	<code>the_ID()</code>
la date de l'article	<code>get_the_date()</code>
les tags de l'article	<code>the_tags()</code>
la catégorie de l'article	<code>the_category()</code>
une meta info de l'article	<code>the_post_meta(meta)</code>
le lien vers l'article	<code>the_permalink()</code>
les liens suivant et précédent	<code>posts_nav_links()</code>
les liens de pagination	<code>wp_link_pages()</code>

“

*Les fonctions `the_something()` affichent dans la page pour obtenir la valeur il existe des fonctions `get_the_something()`
Pour la date cependant la fonction `the_date()` pose problème quand on est dans une boucle, il vaut mieux utiliser `get_the_date()`*

Les fichiers de contenu

Description	fichier
écran par défaut	index.php
liste des articles	home.php
page d'accueil	front-page.php
affichage d'un article	single.php
affichage d'une page	page.php
liste des ressources (articles, auteurs, catégories ...)	archive.php
page d'erreur	404.php
formulaire de recherche	searchform.php
résultat de recherche	search.php
commentaires	comments.php
en-tête	header.php
pied	footer.php
barre latérale	sidebar.php

A vous

- Créer un fichier `single.php` pour afficher le détail d'un article (utiliser les templates tag)
- Créer un fichier `page.php` pour afficher le contenu des pages
- Créer une page d'erreur personnalisée

Factoriser avec les template_parts

Il s'agit simplement d'un include à la sauce Wordpress, cependant il est plus intéressant car il fonctionne bien avec les héritages de thèmes.



```
<?php
    // Affiche les contenu du fichier parts/article-content.php
    get_template_part("parts/article_content");
?>
```

Afficher le menu



```
<!-- dans header.php après l'ouverture de body -->  
<nav class="nav" role="navigation">  
    <?php wp_nav_menu() ?>  
</nav>
```

Si on a plusieurs menus, on peut passer en argument le nom du menu que l'on veut afficher comme ceci :




```
<?php wp_nav_menu([ "theme_location" =>"menu-principal" ]) ?>
```

Référencer les menus

Afficher un menu créé par l'administrateur c'est bien mais il faut connaître le nom de ce menu. Dans le cas où le développeur du thème n'est pas lui-même l'administrateur du site il ne peut savoir a priori quel nom sera utilisé.

Pour pallier ce problème le développeur peut référencer des emplacements de menu déjà nommés. L'administrateur indiquera alors que tel ou tel menu se charge dans tel ou tel emplacement

```
  
// Dans fonctions.php  
  
function my_theme_setup(){  
    // id et description du menu  
    register_nav_menu("menu-1", "Menu principal");  
    register_nav_menu("menu-2", "Menu secondaire");  
}  
  
add_action( 'after_setup_theme', 'my_theme_setup' );
```

```
  
<?php wp_nav_menu([ "theme_location" =>"menu-1" ]) ?>
```

Afficher les widgets

Pour afficher les widgets, comme pour les menus, il faut créer un espace ou charger ces widgets

```
  
// Dans functions.php  
  
function my_widgets_init() {  
    register_sidebar( array(  
        'name'          => 'Footer widgets',  
  
        'id'             => 'footer-widgets-1',  
        'description'    => 'Ajouter vos widgets ici',  
        'before_widget'  => '<section id="%1$s" class="widget %2$s">',  
        'after_widget'   => '</section>',  
        'before_title'   => '<h2 class="widget-title h5">',  
        'after_title'    => '</h2>',  
    ) );  
}  
  
add_action( 'widgets_init', 'my_widgets_init' );
```


```
  
// Dans un template  
<?php dynamic_sidebar( 'footer-widgets-1' ); ?>
```

Les images mises en avant

1- Activer dans `functions.php`

```
  
add_theme_support( 'post-thumbnails' );
```

2- Modifier les articles pour définir une image 3- Afficher l'image dans le template

```
  
<?php the_post_thumbnail() ?>
```

On peut passer en argument une chaîne de caractère pour indiquer la taille de l'image, les tailles (thumbnail, medium, medium-large, large, full). On peut également passer un tableau d'entier indiquant la largeur et la hauteur.

Tester si l'article possède une image

```


<?php
        if(has_thumbnail()){
            the_post_thumbnail("large");
        }
    ?>
</div>


```

Modèles pour les pages

Créer une page PHP commence par le commentaire suivant

```
<?php
/*
 * Template name: custom page
 */
<?
```

Dans l'administration de Wordpress choisir le modèle qui doit s'appliquer à la la page

Attributs de page



Modèle :

custom page

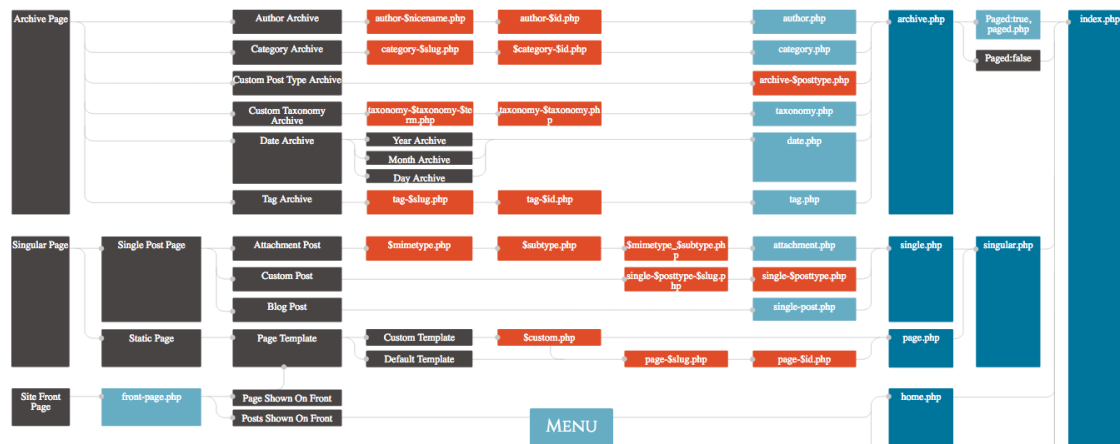


Page parente :



La hiérarchie des templates

THE WORDPRESS TEMPLATE HIERARCHY



Gérer les commentaires

méthode simple



```
<!-- dans single.php -->
<h2>Les commentaires</h2>
<!-- comments_template affiche le contenu de comments.php -->
<?php comments_template(); ?>
```



```
<!-- comments.php -->

<!-- Affichage du formulaire -->
<?php comment_form(); ?>

<!-- le nombre de commentaires -->
<p><?php comments_number(); ?> commentaires</p>

<!-- la liste des commentaires -->
<?php wp_list_comments( [
    "style"      => "div",
    "format"     => "html5",
    "avatar_size" => 74,
] );
?>
```

Thème enfant

La mention template indique le nom du thème parent (qui doit être installé dans Wordpress). Un thème enfant hérite de tous les fichiers de son parent mais peut redéfinir ses propres fichiers.



```
/*
```

```
Theme Name: Mon super theme
```

```
Template: nom du thème parent
```

```
*/
```