

Créer un thème Wordpress

Partie 1 : Les bases

CONTENU

| | |
|--|---|
| Anatomie d'un thème Wordpress | 1 |
| Créer la base d'un thème Wordpress | 3 |
| Minimum requis pour déclarer un thème dans Wordpress | 3 |
| Le fichier « style.css » | 3 |
| Le fichier « index.php » | 4 |
| Le fichier « functions.php » | 4 |
| Activer le thème..... | 5 |
| L'entête et le pied de page | 6 |
| Header.php | 6 |
| Footer.php | 6 |
| Mise à jour du fichier « index.php » | 7 |
| Découverte des fonctions Wordpress..... | 7 |
| Index.php | 7 |
| Header.php | 7 |
| Footer.php | 7 |
| Ajouter le logo dans l'entête du site | 8 |
| Conclusion..... | 8 |

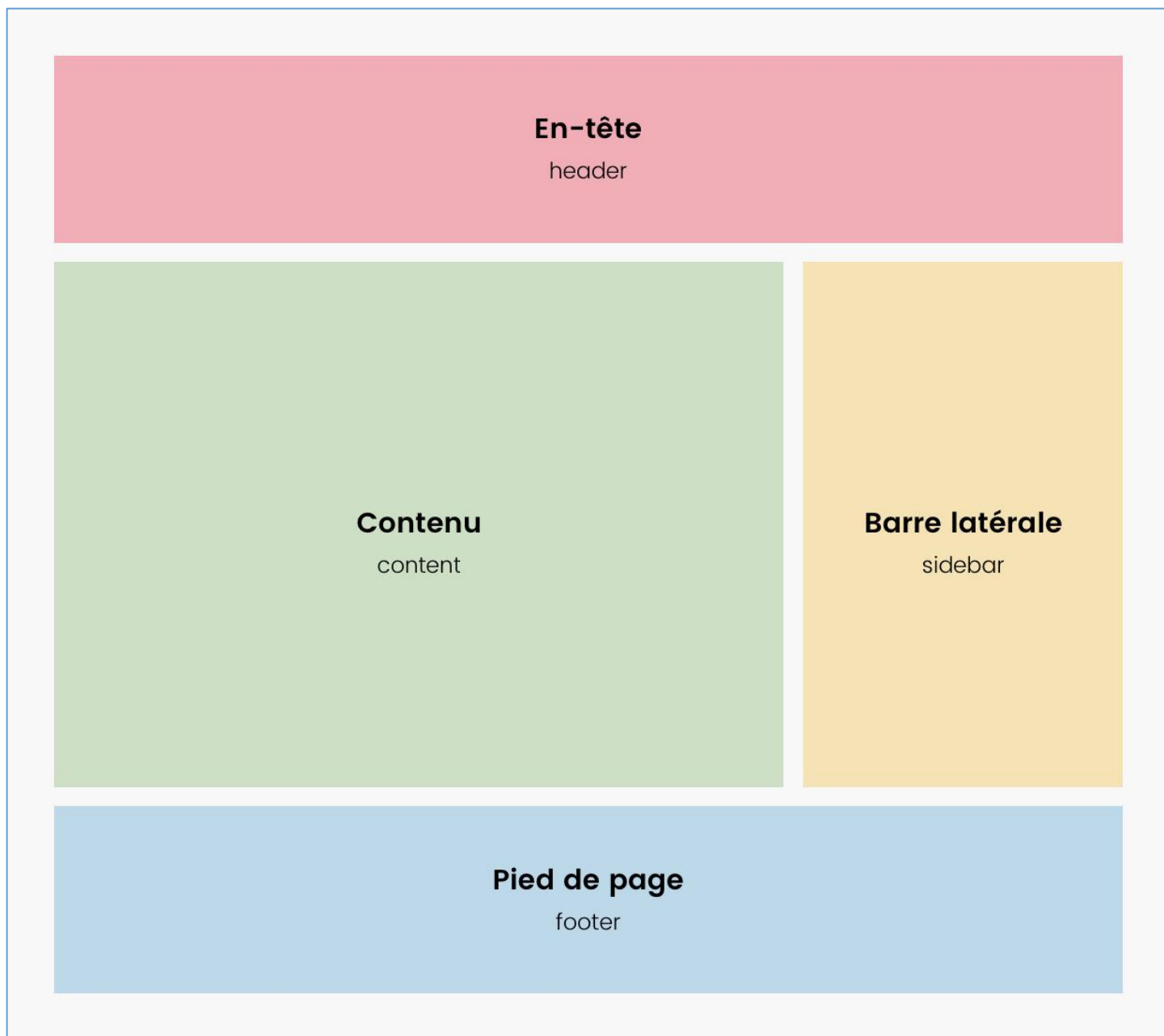
Ce cours est très largement inspiré du site CapitaineWP et de la formation en ligne accessible à cette adresse :

<https://capitainewp.io/formations/developer-theme-wordpress/>

ANATOMIE D'UN THEME WORDPRESS

Tous les sites sont plus ou moins réalisés de la même manière, du moins pour les fondations. Voyons comment WordPress se base sur ces concepts de base pour la conception de thèmes.

Si nous prenons l'exemple d'un blog, nous pouvons découper l'interface utilisateur en 4 zones distinctes :



L'en-tête (header) : contient le titre du site, le logo et le menu principal. Cette partie est commune à toutes les pages du site.

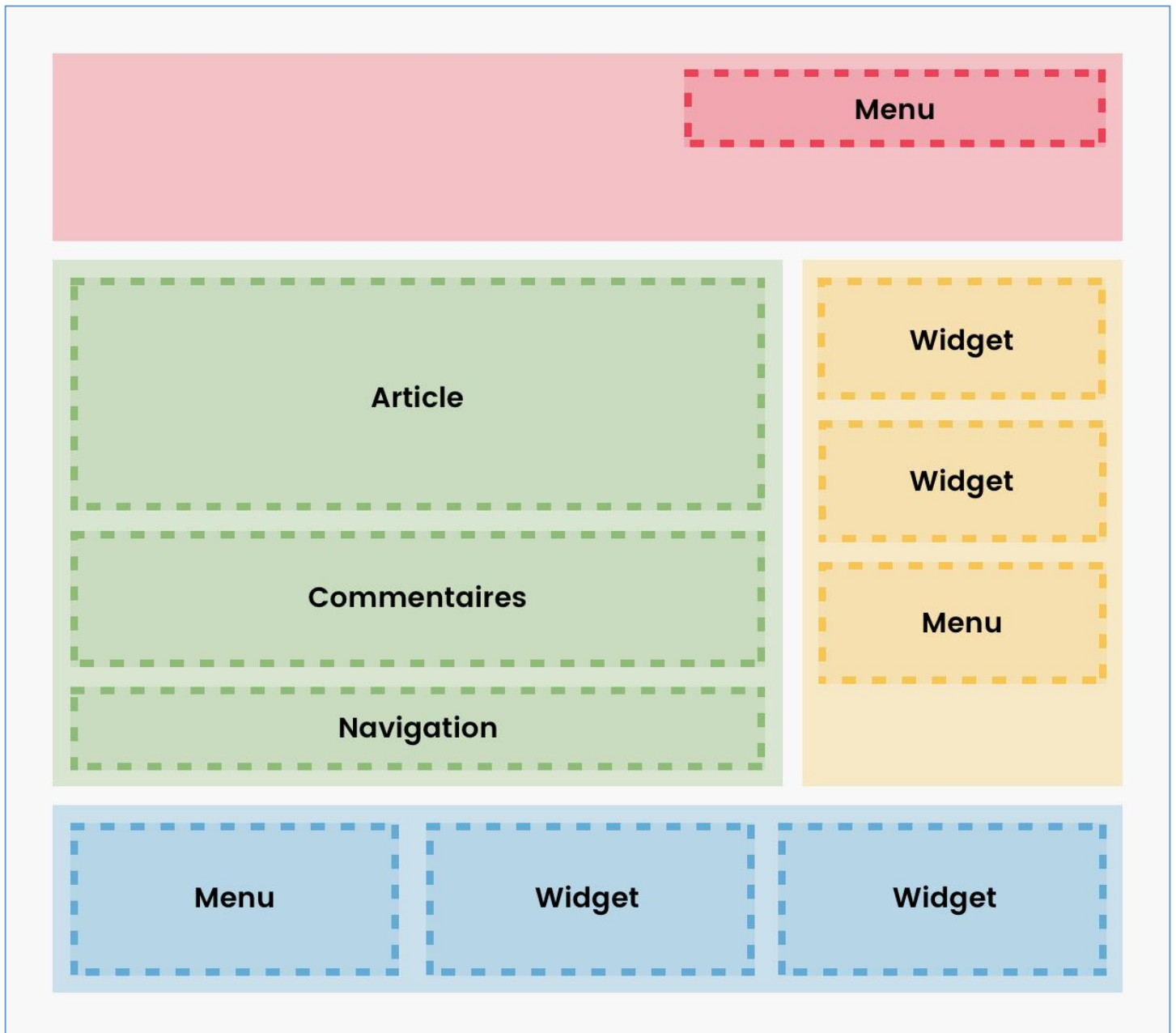
Le pied de page (footer) : identique pour toutes les pages du site.

La barre latérale (sidebar) : accueille des widgets et autres informations annexes.

Le contenu (content) : contient le contenu principal du site, c'est ici que s'afficheront vos articles.



Dans chacune de ces parties, nous intégrerons les composants typiques d'un site web :



Le rôle de ces composants est de rendre le thème dynamique et administrable.

Nous y retrouvons :

Les menus : le menu principal dans l'entête, et parfois un menu secondaire dans la sidebar ou le footer.

Les commentaires : à la suite d'un article (uniquement sur les pages « articles »).

Les widgets : dans les sidebars ou le footer pouvant afficher les derniers articles, les catégories, un agenda...

La navigation : pour aller à l'article suivant ou précédent.

Dans la suite de ce document, vous apprendrez à utiliser ces zones et composants.

Dans un premier temps, nous allons mettre en place le strict minimum pour déclarer notre thème.

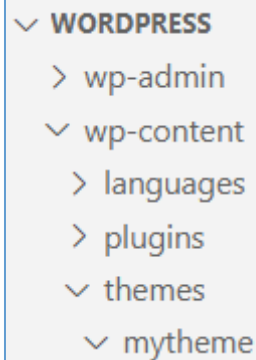
CREER LA BASE D'UN THEME WORDPRESS

La première chose à faire est de créer la structure de code minimale qui permettra à notre thème d'être visible et activable dans l'interface d'administration de WordPress.

MINIMUM REQUIS POUR DECLARER UN THEME DANS WORDPRESS

Les thèmes sont stockés dans le dossier « **wp-content/themes** » de notre installation Wordpress. Chaque thème est représenté dans un dossier.

La première chose à faire est de créer un nouveau dossier qui va accueillir les fichiers de notre thème.



Je l'appelle « mytheme » puis je l'ouvre avec mon éditeur de code. Dans ce dossier, nous allons créer quelques fichiers :

style.css : (obligatoire) permet de déclarer le thème dans Wordpress et qui contiendra les styles CSS du thème.

index.php : (obligatoire) contiendra l'unique page de notre thème.

functions.php : dans lequel on va pouvoir activer et ajouter des fonctionnalités ;

screenshot.png : permet d'afficher un aperçu du thème dans l'interface d'administration de WordPress. Format PNG obligatoire.

LE FICHIER « STYLE.CSS »

Le fichier « style.css » permet de déclarer le thème dans Wordpress.

Au début de ce fichier, nous allons déclarer les informations du thème dans un bloc de commentaire :

```

wp-content > themes > mytheme > # style.css
1  /*
2  Theme Name: My Theme
3  Theme URI: https://devoldere.net
4  Author: Mickaël DEVOLDERE
5  Author URI: https://devoldere.net
6  Description: Mon premier thème sur Wordpress !
7  Requires at least: WordPress 6.0
8  Version: 1.0
9  */
  
```

Nous y retrouvons le nom du thème, les informations sur l'auteur, la licence, la version... Seul le nom du thème est obligatoire, les autres informations sont facultatives (mais recommandées).

/!\ Attention : Pas d'espace avant les deux points « : », sinon ça ne fonctionne pas !



LE FICHIER « INDEX.PHP »

Ce fichier contiendra, pour le moment, la structure HTML du site. Plus tard, nous découperons ce fichier en plusieurs parties distinctes.

Pour le moment, nous nous contenterons d'une structure HTML minimale :

```
wp-content > themes > mytheme > index.php
1  <!DOCTYPE html>
2  <html lang="fr">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Mon super site Wordpress !</title>
7  </head>
8  <body>
9      <h1>Bienvenue</h1>
10 </body>
11 </html>
```

Nous reviendrons compléter ce fichier plus tard.

LE FICHIER « FUNCTIONS.PHP »

Ce fichier permet d'activer des fonctionnalités dans notre thème. Par défaut, Wordpress n'active que le minimum requis dans chaque thème. C'est le développeur du thème qui décide quelles fonctionnalités activer dans chacun de ses thèmes.

Par exemple, « l'image de mise en avant » d'un article est une fonctionnalité désactivée par défaut dans les thèmes.

Pour activer cette fonctionnalité, ajouter dans le fichier « functions.php » :

```
wp-content > themes > mytheme > functions.php
1  <?php
2
3  // Activer la prise en charge des images de "mise en avant" dans les articles
4  add_theme_support( 'post-thumbnails' );
```

« add_theme_support() » est une fonction de Wordpress, elle permet d'activer des fonctionnalités dans notre thème.

Il existe un certain nombre de fonctionnalités activables. Retrouvez la liste complète sur le site de Wordpress :

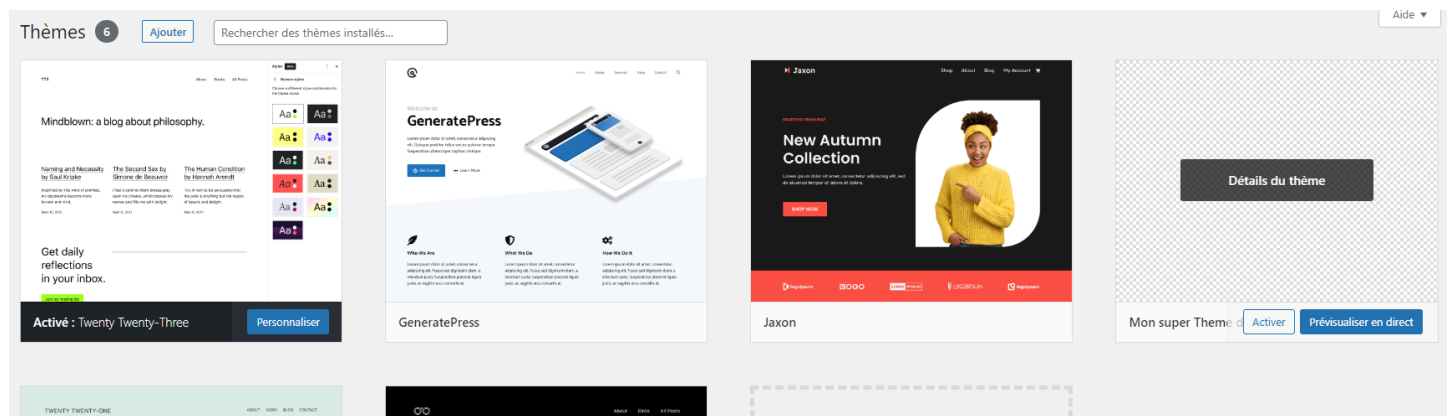
https://codex.wordpress.org/Theme_Features



ACTIVER LE THEME

Pour tester notre thème, nous pouvons désormais l'activer dans l'interface d'administration de Wordpress.

Rendez-vous dans le menu « apparence → thèmes. Notre thème apparait à côté des autres thèmes déjà présents dans l'installation de Wordpress :



Cliquer sur le bouton « activer » permet de rendre le thème actif et désactivera automatiquement le thème précédemment activé.

Si le thème n'apparait pas dans la liste, vérifiez :

- que vous avez bien enregistré vos fichiers ;
- dans le bon dossier ;
- dans style.css, il ne faut pas mettre d'espace avant les deux points « : » ;
- que vous avez correctement écrit les commentaires « /* */ ».

Une fois activé, rendez-vous sur la partie publique de votre site et admirez le résultat 😊

Bien entendu, le thème n'affiche rien d'exceptionnel, mais nous voyons bien le contenu de notre fichier index.php s'afficher.



L'ENTETE ET LE PIED DE PAGE

Dans la première partie de ce document, nous avons placé toute la structure HTML du site dans le fichier `index.php`.

Comme nous le verrons plus loin, nous pourrions créer d'autres fichiers qui permettront de différencier l'affichage pour certains types de contenus (pages statiques, liste d'articles, un seul article etc...). Le fichier `index.php` étant le fichier par défaut si les autres n'existent pas.

Dans chacun de ces fichiers, nous devrons replacer l'entête et le pied de page.

En effet, l'entête et le pied de page de la plupart des sites web sont généralement communs à toutes les pages du site. C'est-à-dire que le contenu est le même, peu importe la page affichée.

Pour éviter toute répétition de code, l'entête et le pied de page seront placés dans des fichiers séparés respectivement :

- **header.php** pour l'entête
- **footer.php** pour le pied de page.

Séparer l'entête et le pied de page permet de respecter le concept de base « Don't Repeat Yourself » qui est l'un des concepts les plus importants en programmation.

Nous en profiterons pour commencer à inclure quelques fonctions Wordpress dans notre interface utilisateur.

HEADER.PHP

L'entête du site contient généralement le titre du site, le logo, le menu principal.

L'entête contiendra tout le code HTML à partir du début de la structure (`<html>`) jusqu'à l'ouverture de la balise `<body>`. La balise `</body>` sera fermée dans le fichier « `footer.php` ».

```
wp-content > themes > mytheme > 📄 header.php
1  <!DOCTYPE html>
2  <html <?php language_attributes(); ?>>
3  <head>
4      <meta charset="<?php bloginfo('charset'); ?>">
5      <meta name="viewport" content="width=device-width, initial-scale=1"/>
6      <?php wp_head(); ?>
7  </head>
8
9  <body <?php body_class(); ?>>
10
11 <?php wp_body_open(); ?>
```

Réduit au strict minimum pour le moment, ce fichier contiendra, plus tard, l'entête du site (titre, logo, menu...).

FOOTER.PHP

Le pied de page du site contient généralement des liens vers des pages statiques (mentions légales, RGDP etc..) et des liens vers des sites partenaires.

Footer.php ferme les balises `</body>` et `</html>`

```
wp-content > themes > mytheme > 📄 footer.php
1  <?php wp_footer(); ?>
2  </body>
3  </html>
```



MISE A JOUR DU FICHIER « INDEX.PHP »

Maintenant que notre entête et pied de page sont prêts, nous allons éditer le fichier index.php.

Index.php ne contiendra que la partie « centrale du site » (le contenu principal)

Index.php doit demander à Wordpress d'inclure l'entête et le pied de page.

```
wp-content > themes > mytheme > index.php
1  <?php get_header(); ?>
2
3  <h1>Bienvenue</h1>
4
5  <?php get_footer(); ?>
```

Il est temps de tout enregistrer et de tester.

Retournez sur la partie publique, et admirez le résultat (pas de changement visuel). Affichez le code source de la page vous permettra de constater que le code des 3 fichiers (header, index et footer) est bien présent.

Si la manipulation a bien fonctionné, vous devriez voir apparaître la barre d'administration WordPress (en noir en haut du site) et le titre dans l'onglet du navigateur. Concernant le code source de la page, remarquez qu'il y a du HTML en plus de ce que nous avons implémenté ! C'est dû aux fonctions que l'on vient d'ajouter dans header et footer et que l'on va analyser dans les parties suivantes.

DECOUVERTE DES FONCTIONS WORDPRESS

Revenons sur le code que ne venons d'implémenter.

INDEX.PHP

get_header() : la fonction get_header() indique à Wordpress d'inclure le fichier header.php du thème actuel.

get_footer() : la fonction get_footer() indique à Wordpress d'inclure le fichier footer.php du thème actuel.

D'une manière générale, tous les templates que nous créerons dans notre thème inclurons ces 2 fichiers de la même manière.

HEADER.PHP

language_attributes() : permet de définir la langue du document. Cette valeur est basée sur le paramètre WordPress dans « Réglages > Général > Langue du site ».

bloginfo('charset') : permet de définir l'encodage du site. Par défaut c'est UTF-8 et c'est très bien comme ça : votre site prendra en charge les caractères spéciaux, accents, caractères non-latins, émoticônes...

wp_head() : c'est par cette fonction que WordPress, votre thème et les extensions vont pouvoir venir déclarer le chargement des scripts JS et des styles CSS. On verra un peu plus tard comment en tirer parti de cette fonctionnalité.

body_class() : permet d'obtenir des noms de classe CSS en fonction de la page visitée, ce qui pourra nous faciliter la création de nos styles. Les classes déclarées dans la balise body sont gérées par Wordpress et dépendent du contexte actuel (page d'accueil, article, page statique etc...).

wp_body_open() : (depuis WordPress 5.2) permet à des extensions d'écrire du code au début du body.

FOOTER.PHP

wp_footer() : Possède le même rôle que wp_head() à savoir, charger des scripts et styles mais cette fois-ci, en bas de page. Avec les dernières recommandations W3C, cette fonction devrait être de moins en moins utilisée.



AJOUTER LE LOGO DANS L'ENTÊTE DU SITE

Retournons dans notre fichier « header.php »

Pour le moment, ce fichier contient le code HTML jusqu'à l'ouverture de la balise <body>. Nous allons maintenant ajouter l'entête du site, autrement dit, le contenu de la balise <header> qui contiendra le logo dans un premier temps.

Choisissez le logo de votre choix que vous trouverez sur le web ou copiez celui-ci :



Dans le dossier du thème, créer un dossier « images » et déposez-y le logo que vous avez choisi.

Dans header.php ajouter le code suivant après l'appel de la fonction **wp_body_open()** :

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

<header class="header">
|   <a href="<?php echo home_url( '/' ); ?>">
|   |   
|   |   </a>
|   </header>
```

Enregistrez les fichiers et retournez sur la partie publique du site.

Vous devriez voir le logo et pouvoir cliquer dessus, ce qui vous ramène à l'accueil du site grâce à la fonction **home_url()**.

D'autre part, nous utilisons la fonction **get_template_directory_uri()** afin d'obtenir l'adresse absolue (c'est-à-dire complète) du logo. Sans ça, votre image ne s'affichera pas.

En regardant le code (clic droit, afficher le code source de la page) on voit bien l'url vers l'accueil ainsi que l'url complète de l'image.

Vous avez la liberté d'étoffer l'entête avec les éléments de votre choix (images, slogan, titre etc...).

CONCLUSION

Maintenant que nous avons bien séparé l'en-tête et le pied de page du reste du site, nous allons pouvoir continuer.

Avant de rendre notre site dynamique, il va nous falloir du contenu.

Dans la partie suivante, nous allons générer du faux contenu en un clic.

--- FIN DU DOCUMENT ---

