



Université de Bretagne Occidentale

UN PROJET

Avec ou sans WordPress ?

2016/17

Benjamin SIMIER

DEUST T-MIC

U.F.R Sciences et Technologiques
de Brest



Nom d'utilisateur ou adresse e-mail

Mot de passe

Se souvenir de moi

Se connecter

Remerciements

Je remercie l'équipe d'enseignants et d'intervenants grâce à qui j'ai appris beaucoup de choses pour mener à bien mes projets pendant les deux ans de formation

Je souhaite aussi remercier Alain Fessant qui est toujours présent pour nous en DEUST T-MIC.

Sommaire

Introduction	Page 4
1. Présentation de WordPress	Page 6
2. Création d'un site sous WordPress	Page 8
a. Installation locale du CMS	Page 9
b. Installation d'un thème	Page 14
c. Création d'un thème sur-mesure	Page 15
3. Création d'un site sans WordPress	Page 17
a. Préparation de l'environnement de développement	Page 18
b. Faut-il utiliser un framework ?	Page 19
4. CSS vs SCSS	Page 20
5. Mise en ligne	Page 23
6. Conclusion	Page 26
Glossaire	Page 28
Sitographie	Page 31
Annexes	Page 32

INTRODUCTION



Utilisateur ou adresse e-mail

Mot de passe

Rester connecté

Se connecter

[Mot de passe oublié ?](#)

[Retour vers Deust Conf'](#)

Introduction

Lors de cette deuxième année de DEUST T-MIC j'ai réalisé plusieurs stages, mais surtout, j'ai réalisé plusieurs projets, des sites web. Ces sites web étaient destinés à des clients ou étaient des projets personnels.

Dans ce rapport je vais donc vous parler de la réalisation de ces projets plutôt que d'un stage en particulier.

Quand je commence un projet je me pose toujours pleins de questions avant de le démarrer, quelques questions me semble plus pertinent que d'autres...

- Faut-il utiliser le **CMS** WordPress ?
- Ou vaut-il mieux créer le site sans CMS ?

- Si j'utilise **WordPress**, dois-je créer un thème sur mesure ?
- Ou puis-je télécharger et installer un thème ?

- Si j'utilise WordPress, quel **plugins** installer ?

- Que ce soit avec ou sans WordPress, vais-je utiliser du **CSS** ?
- Ou vais-je utiliser du **SCSS** ?

- Vais-je devoir utiliser un **framework** ?
- Si oui, le quel ?

Je vais commencer par vous présenter le CMS WordPress, puis son installation en **local**, suivi de l'installation d'un thème. Après avoir vu l'installation d'un thème nous allons voir la création d'un thème sur-mesure, et enfin la mise en ligne du site.

Nous allons ensuite voir la création d'un site sans utiliser WordPress, la préparation de l'environnement en local, l'utilisation d'un framework, et la mise en ligne du site

Enfin, nous allons voir quelle est la différence entre le CSS et le SCSS. Puis nous allons conclure le rapport.

Bonne lecture !

1. PRÉSENTATION DE WORDPRESS



Utilisateur ou adresse e-mail

Mot de passe

Rester connecté

Se connecter

[Mot de passe oublié ?](#)

[Retour vers Deust Conf'](#)

1. Présentation de WordPress

WordPress est un CMS (content management system, ou système de gestion de contenu).

Ce CMS est écrit en **PHP** et utilise une base de données **SQL**, il peut s'installer sur tout les serveurs qui utilisent eux même ces technologies.

Né en 2003, WordPress était à la base utilisé pour créer des petits blogs. Aujourd'hui il est utilisé par 27 % des sites web dans le monde.

WordPress n'est pas le seul CMS existant, il a plusieurs concurrents, dont Joomla, créé en 2015, il est utilisé par 3,4 % des sites web. Sont deuxième grand concurrent est Drupal, né en avant WordPress (2001), il est utilisé par 2,2 % des sites web.

Il faut aussi savoir que 53,4 % des sites web n'utilisent pas de CMS, mais nous verons cela plus en détail dans la deuxième partie du rapport.

WordPress.org Français

Accueil Thèmes Extensions Installation Téléchargement Versions Traduire WordPress Blog À propos

Bienvenue

WordPress constitue le *nec plus ultra* en matière de plateformes sémantiques de publication personnelle, alliant esthétique, standards du Web et ergonomie. Gratuit, WordPress n'en est pas moins inestimable. Sous licence GPLv2+, WordPress est un logiciel libre et gratuit.

Plus simplement, WordPress est l'outil qu'il vous faut si vous voulez avancer au moyen d'un système de gestion de contenu, plutôt que de vous battre contre lui.

Téléchargement

La dernière version francisée de WordPress sera toujours accessible depuis cette page. Il vous suffit de cliquer sur le bouton se trouvant ici à droite pour la télécharger.

S'il vous venait l'envie ou le besoin de télécharger l'une des versions précédentes de WordPress en français, elles restent disponibles et accessibles ici-même.

La version francisée est intégralement réalisée par l'équipe de l'association [WordPress-Francophone](#). Outre [un blog](#) proposant les dernières nouvelles importantes du monde de WordPress, son site donne accès à [un forum d'entraide en français](#), où vous pourrez poser des

Télécharger WordPress 4.7.5
 .zip — 8.8 Mo
 Télécharger au format .tar.gz — 8.1 Mo

Ressources
 Pour vous aider à installer ou utiliser WordPress, consultez notre documentation dans votre langue.

- Association WordPress-Francophone
- Blog de WPFR

Page d'accueil Wordpress.org

2. CRÉATION D'UN SITE **SOUS** WORDPRESS



Utilisateur ou adresse e-mail

Mot de passe

Rester connecté

Se connecter

[Mot de passe oublié ?](#)

[Retour vers Deust Conf'](#)

2. Création d'un site sous WordPress

Quand je crée un site web pour un client je me demande toujours si je vais lui proposer WordPress, son utilisation est tellement simple.

Un fois connecté à l'administration, l'utilisateur peut créer des pages, des articles, les modifier, les supprimer, organiser son menu, ...

La création d'un thème n'est pas plus compliqué que la création d'un site sans utilisation de WordPress... C'est même plus simple... Et les plugins nous simplifie encore plus la création du site.

Après avoir répondu à la question «WordPress ou pas ?», je dois me demander si je réalise un thème sur-mesure ou si je vais acheter, télécharger un thème tout prêt pour l'installer et le modifier.

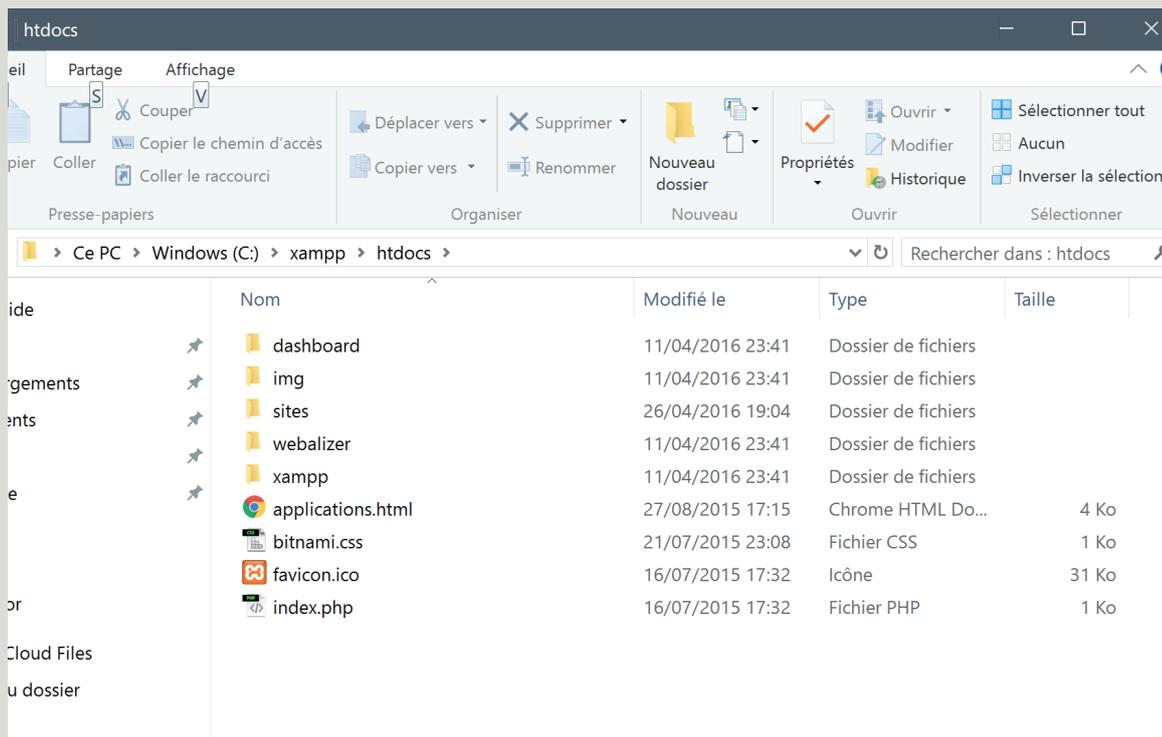
a. Installation locale du CMS

Je choisis d'utiliser le logiciel **Xampp** car c'est avec celui-ci que je travaille avec tout les jours et je suis à l'aise avec.

Xampp est le logiciel qui permet de démarrer des services web tels qu'**Apache** qui permet de lire les fichiers PHP avec un navigateur web, pour faire simple c'est Apache qui traduit le PHP au navigateur web qui lui va retransmettre l'affichage du site sur les écrans. Utiliser Apache c'est bien mais grâce à Xampp nous pouvons faire mieux et d'ailleurs ce que je vais faire si je veux utiliser WordPress, je dois lancer en plus d'Apache une base de données SQL ainsi que le panel d'administration **PhpMyAdmin**, la base de données SQL permet d'enregistrer des données tels que des mots de passe, des articles, ... Quand on utilise Apache et SQL nous pouvons quasiment tout faire sur le web.

Maintenant que nous avons défini ensemble les termes techniques voyons l'installation de WordPress en lui-même.

Tout d'abord vous devez télécharger les fichiers WordPress disponibles sur le site officiel à l'adresse : <https://fr.wordpress.org>, les fichiers sont contenus dans une archive *.zip lors du téléchargement, il va donc falloir les extraire puis les placer dans le dossier « htdocs » qui se trouve là où vous avez installé Xampp, en général à la racine d'un disque, par exemple **C:\xampp\htdocs**.



Dossier C:\xampp\htdocs

Il existe une autre méthode qui permet de ne pas aller télécharger le fichier sur internet, de décompresser l'archive, ...

Cette méthode c'est l'utilisation d'une ligne de commande, en une ligne vous allez télécharger.

Avant de pouvoir utiliser cette ligne il va vous falloir installer un logiciel comme «**Git Bash**» et le module «**Bower**»

Une fois installé, ouvrez Git Bash à l'endroit où vous voulez installer WordPress et tapez la ligne suivante : **bower install wordpress**

Bower va alors se charger d'aller récupérer la dernière version de WordPress et va la décompresser directement.

La version qu'il va télécharger sera la dernière mais si vous voulez une version plus ancienne je vous conseille d'aller voir la documentation de Bower.

(Pour utiliser Bower il vous faudra installer Node et NPM).

```

benja@SURFACE-BENJAMIN MINGW64 /c/xampp/htdocs/sites/Nouveau dossier
$ bower install wordpress
bower wordpress#*                cached https://github.com/WordPress/WordPress.git#4.7.5
bower wordpress#*                validate 4.7.5 against https://github.com/WordPress/WordPress.git#4.7.5
bower wordpress#^4.7.5          install wordpress#4.7.5

wordpress#4.7.5 bower_components\wordpress

benja@SURFACE-BENJAMIN MINGW64 /c/xampp/htdocs/sites/Nouveau dossier
$

```

Git : bower install wordpress

Une fois les fichiers extraits et placés dans le dossier «htdocs» vous allez devoir vous rendre, grâce à un navigateur web dans le panel administrateur de la base de données SQL qui se trouve à l'adresse : <http://localhost/phpmyadmin> (bien évidemment il faut démarrer Apache et SQL à l'aide de Xampp avant tout).

Si c'est votre première utilisation de Xampp, je vous conseille de créer un mot de passe administrateur pour la base de données SQL, pour cela cliquez sur «Comptes d'utilisateurs» suivit de «Changer les privilèges» de la ligne «root – localhost» puis «Modifier le mot de passe» , à ce moment ou lors du rechargement de la page vous allez avoir un problème car le panel administrateur va se bloquer mais pas de panique nous allons y remédier, pour cela je vais vous demander de vous rendre dans le dossier où est installer Xampp, souvenez-vous de notre exemple précédent : **C:\xampp** mais cette fois ci il faudra aller dans le dossier «**phpMyAdmin**» afin de trouver et ouvrir le fichier config.inc.php (ouvrez le avec Notepad++ par exemple) pour y éditer la ligne suivante **\$cfg['Servers'][\$i]['password'] = ''**; comme cela **\$cfg['Servers'][\$i]['password'] = 'VotreMotDePasse'**; , maintenant vous pouvez retourner sur votre navigateur et y actualiser la page, le problème est résolu.

Revenons sur notre panel pour y ajouter une nouvelle base de données, pour ça rien de plus simple il vous suffit de cliquer sur «Nouvelle base de données», ajoutez-y le nom souhaité et faites «Exécuter».

Récapitulons, nous avons maintenant WordPress dans le dossier «**htdocs**» et

une base de données, il ne nous reste plus qu'à lier les deux.
Pour cela, rendez-vous dans votre navigateur à l'adresse <http://localhost> suivit du nom du dossier où est WordPress dans le «[htdocs](#)»

À présent il ne vous reste plus qu'à lancer l'installation du CMS pour lier Apache et SQL mais je vais vous laisser continuer seul puisque tout y est expliqué, il vous suffit de suivre les instructions données par WordPress.
(Voir annexe 1 page 32).

Pour aller plus loin, il y a une chose que je fais et qui me semble importante quand on a plusieurs projets sur le disque dur. J'aime avoir des **nom de domaine** http://mon_site.dev plutôt que http://localhost/mon_site. Cela permet de mieux se retrouver dans ses projets.

Pour se faire il vous suffit d'aller dans le dossier où est installé Xampp, puis dans [apache > conf > extra](#) et enfin d'ouvrir le fichier [httpd-vhosts.conf](#).
Et ajoutez y ceci :

```
<VirtualHost *:80>  
  ##ServerAdmin webmaster@dummy-host.example.com  
  DocumentRoot «C:/xampp/htdocs/mon_site»  
  ServerName mon_site.dev  
  ##ServerAlias www.dummy-host.example.com  
  ##ErrorLog «logs/dummy-host.example.com-error.log»  
  ##CustomLog «logs/dummy-host.example.com-access.log» common  
</VirtualHost>
```

`DocumentRoot «C:/xampp/htdocs/mon_site»` est le chemin vers vos fichiers
`ServerName mon_site.dev` est le nouveau nom de domaine de votre site

Si c'est le premier nom de domaine que vous ajoutez, ajoutez y aussi ce code :

```
<VirtualHost *:80>  
  ##ServerAdmin webmaster@dummy-host.example.com  
  DocumentRoot «C:/xampp/htdocs»  
  ServerName localhost  
  ##ServerAlias www.dummy-host.example.com  
  ##ErrorLog «logs/dummy-host.example.com-error.log»  
  ##CustomLog «logs/dummy-host.example.com-access.log» common  
</VirtualHost>
```

Ces lignes vont vous permettre de garder «localhost» accessible.

```

25 # match a ##ServerName or ##ServerAlias in any <VirtualHost> block.
26 #
27 #
28 # localhost
29 <VirtualHost *:80>
30     ##ServerAdmin webmaster@dummy-host.example.com
31     DocumentRoot "C:/xampp/htdocs"
32     ServerName localhost
33     ##ServerAlias www.dummy-host.example.com
34     ##ErrorLog "logs/dummy-host.example.com-error.log"
35     ##CustomLog "logs/dummy-host.example.com-access.log" common
36 </VirtualHost>
37 #
38 #
39 # Projets
40 #
41 # faismoiunepizza.com
42 <VirtualHost *:80>
43     ##ServerAdmin webmaster@dummy-host.example.com
44     DocumentRoot "C:/xampp/htdocs/sites/projet-wp"
45     ServerName faismoiunepizza.com
46     ##ServerAlias www.dummy-host.example.com
47     ##ErrorLog "logs/dummy-host.example.com-error.log"
48     ##CustomLog "logs/dummy-host.example.com-access.log" common
49 </VirtualHost>
50 #
51 #
52 # lafinebouche camion.dev
53 <VirtualHost *:80>

```

Mon fichier httpd-vhosts.conf

Pour terminer la création de votre nom de domaine de développement, il faudra vous rendre dans `C:\Windows\System32\drivers\etc` et ouvrir le fichier «`hosts`», puis ajouter la ligne suivante : `127.0.0.1 mon_site.dev` (Windows vous demandera sûrement d'avoir les droits d'administration).

Un fois enregistré, stopper Apache puis relancez le.

Vous pouvez maintenant accéder à votre site depuis votre domaine plutôt que par <http://localhost>

```

19 # localhost name resolution is handled within DNS itself.
20 #   127.0.0.1      localhost
21 #   ::1           localhost
22
23 127.0.0.1 localhost
24
25 # Projets
26 127.0.0.1 faismoiunepizza.com
27 127.0.0.1 lafinebouche camion.dev
28 127.0.0.1 pinbook.dev
29 127.0.0.1 symfony.dev
30 127.0.0.1 jetrouveunstage.dev
31 127.0.0.1 randonnenepal.dev
32 #

```

Mon fichier hosts

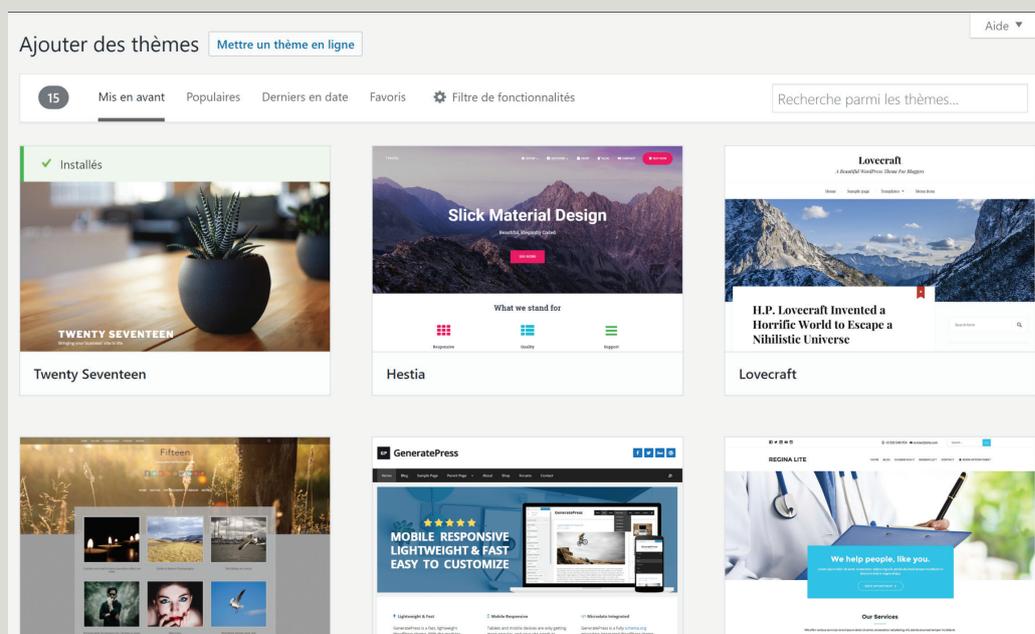
b. Installation d'un thème

WordPress permet d'installer simplement des thèmes pour personnaliser son site et le rendre unique, pour cela vous pouvez, soit télécharger un thème sur internet (gratuit ou payant), soit installer directement un thème depuis votre administration WordPress.

Si vous choisissez d'installer directement un thème depuis votre administration il faudra aller dans **Apparence > Thèmes > Ajouter**.

Vous allez pouvoir faire votre choix parmi des milliers de thèmes.

Cliquez ensuite sur installer, puis activer (Vous pouvez aussi prévisualiser les thèmes sur votre site avant de les installer).

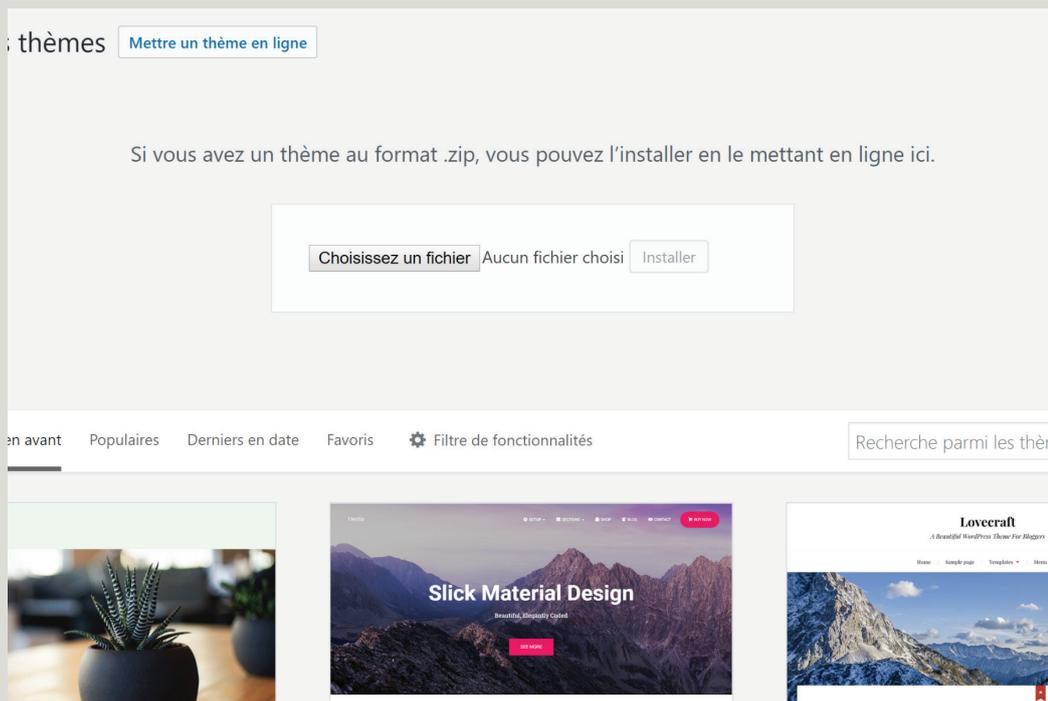


Choix d'un thème dans le panel

L'autre méthode consiste à télécharger le thème sur internet et de l'installer manuellement... Ici deux méthodes.

La première méthode est d'accéder au **FTP**, de placer le dossier de votre thème (décompressé) dans **wp-content > thèmes**, puis activez le dans l'administration WordPress (**Apparence > Thèmes**).

La deuxième façon de le faire est d'aller dans **Apparence > Thème > Ajouter > Mettre un thème en ligne**, puis retrouver votre fichier téléchargé en cliquant sur «Choisissez un fichier»



«Mettre un thème en ligne»

c. Création d'un thème sur-mesure

Comme nous avons pu le voir, nous pouvons télécharger et installer des thèmes sur WordPress, mais nous pouvons aussi créer nos propres thèmes. Dans cette partie je vais vous expliquer comment procéder.

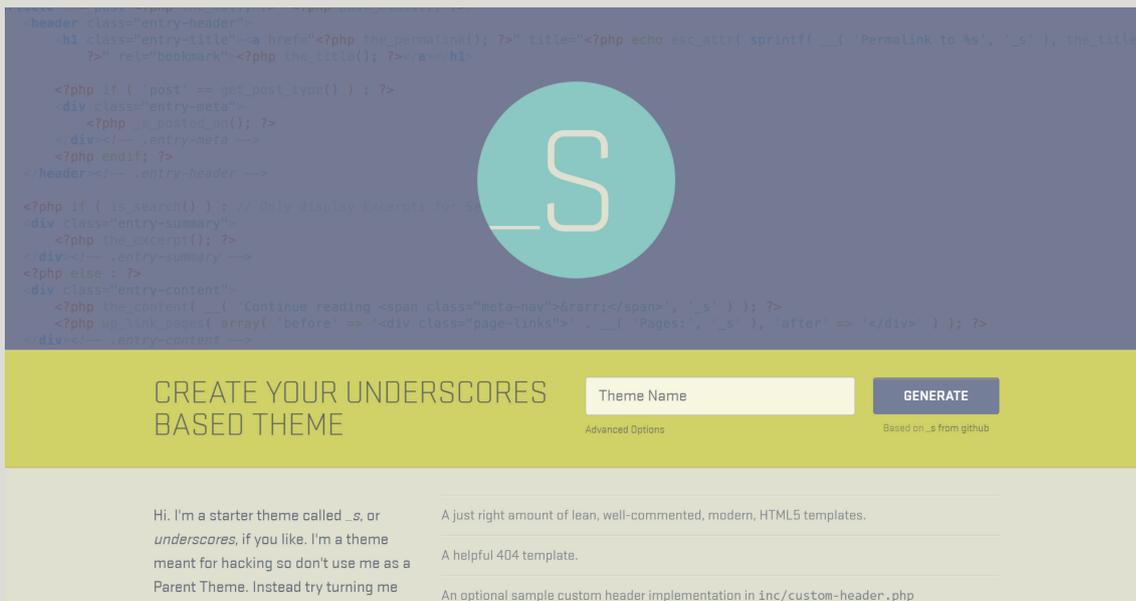
Tout d'abord, pour créer un thème WordPress il faut comprendre certaines choses :

- Nous ne pouvons pas nommer les fichiers comme bon nous semble.
- Un thème WordPress est fragile et peut cesser de fonctionner à chaque modification.
- Il y a des choses à faire et d'autres à éviter.

Pourquoi ne pouvons-nous pas nommer les fichiers comme bon nous semble ? Simplement car WordPress va lire certains fichiers et pas d'autres, par exemple vous voulez modifier la page des articles, c'est possible mais n'allez pas créer une page nommée `article.php` mais plutôt `single.php`, je vous conseille de vous rendre à l'annexe 2 page 35 qui vous présente les différents fichiers utilisables.

Enfin Il y a des choses à éviter, par exemple [page-médias.php](#) ne fonctionnera pas mais [page-medias.php](#) fonctionnera.

Un site intéressant pour de la création de votre thème WordPress, <http://underscores.me>, ce site va générer automatiquement un thème de base, c'est à dire que vous aurez tous les fichiers permettant de créer votre propre thème.



<http://underscores.me>

3. CRÉATION D'UN SITE SANS WORDPRESS



Utilisateur ou adresse e-mail

Mot de passe

Rester connecté

Se connecter

[Mot de passe oublié ?](#)

[Retour vers Deust Conf'](#)

3. Création d'un site sans WordPress

Il est vrai que WordPress est le CMS le plus utilisé au monde avec 27 % de sites web mais 53,4 % des sites n'utilisent pas de CMS.

Ces sites sont développés de A à Z, certains sont dynamiques, d'autres statiques.

Les sites statiques n'ont pas de panel d'administration, pour pouvoir les modifier il faudra passer par le code tandis que les sites dynamiques seront pour la plupart modifiables, ils utiliseront une base de données pour y stocker leurs informations.

a. Préparation de l'environnement

Afin de bien travailler il va falloir préparer son environnement de travail, son espace de travail.

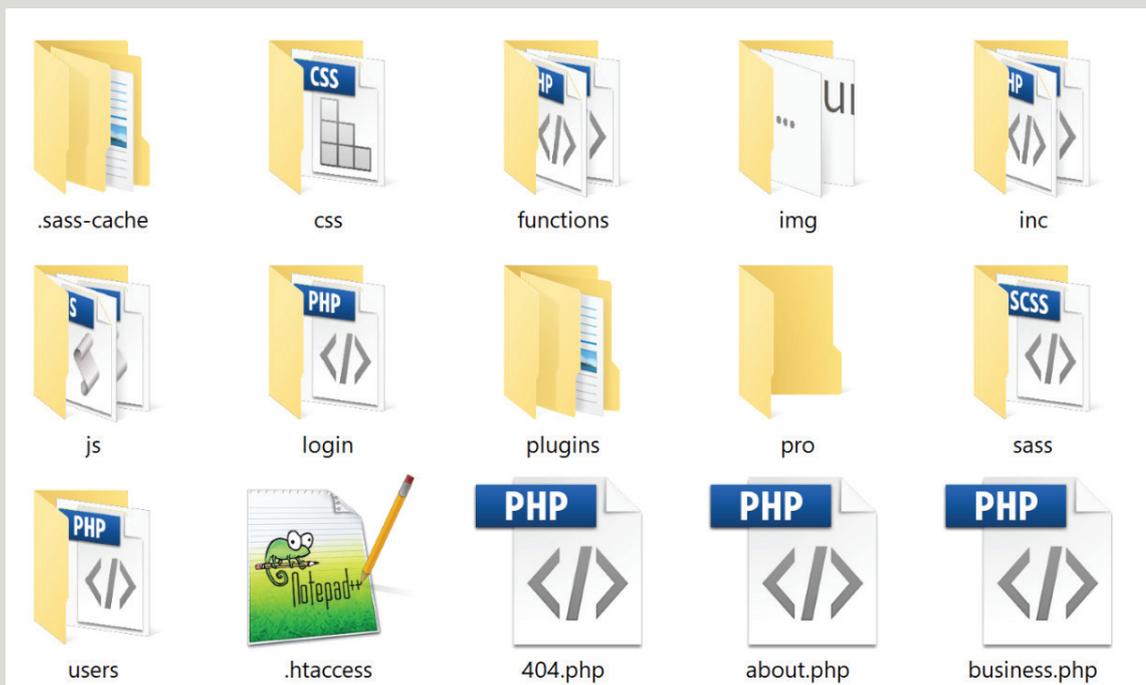
Tout d'abord il faut savoir si nous allons utiliser du PHP ou pas... Si nous utilisons que du **HTML** et du CSS nous n'allons pas avoir besoin de serveur local, dans le cas contraire un serveur local comme Xampp sera obligatoire.

(Rendez-vous à la page 9 pour plus d'informations sur les serveurs local)

Pour bien préparer son environnement il faut préparer ses dossiers, par exemple : Nous allons stocker nos images dans «img» ou «images», les fichiers **JS** dans «js» ou «javascript», les fichiers CSS dans «CSS», ...

Le fichier **.htaccess** est très important, il permet beaucoup de choses comme par exemple ajouter le www, ajouter le https, réécrire les url, bloquer des ressources, ...

Un fichier **index.php** ou **index.html** sera obligatoire pour afficher votre site correctement sur les navigateurs web, disons que c'est la porte d'entrée de votre site web.



Organisation de mes fichiers et dossiers

b. Faut-il utiliser un framework ?

Les frameworks sont très intéressants, ils permettent de développer plus rapidement, plus simplement, ...

Par exemple le framework Symfony est très intéressant si on développe un gros site web, mais il l'est moins si le site est plus petit.

Il permet de travailler en équipe plus simplement, de faire des choses et de les refaire sans devoir tout développer de zéro à chaque projets.

Il en existe de toutes sortes, des frameworks PHP comme symfony, des frameworks CSS comme Bootstrap, des frameworks javascript, ou encore des frameworks **LESS** ou SCSS comme Knacss.

Personnellement j'utilise beaucoup Knacss, pour toutes les mises en pages tout est tellement plus simple.

Par exemple pour mettre du margin je vais utiliser la class «mam» pour margin+all+medium ou «mas» pour margin+all+small ou «mtm» pour margin+top+medium... Rendez vous en annexe 3 page 36 pour voir la documentation.

4. CSS VS SCSS



Utilisateur ou adresse e-mail

Mot de passe

Rester connecté

Se connecter

[Mot de passe oublié ?](#)

[Retour vers Deust Conf'](#)

4. CSS vs SCSS

En stage chez Koality j'ai appris à utiliser le SCSS avec Fabien Canu et depuis j'ai du mal à m'en passer.

Pour faire simple le SCSS c'est du CSS mais plus évolué.

Avec le SCSS nous pouvons utiliser des variables, par exemple si j'utilise la couleur `#222222` plusieurs fois dans mon projet je vais créer la variable :

`$couleur_A : #222222;` , puis je vais pouvoir l'utiliser en fond par exemple : `background-color: $couleur_A;` , et si un jour je veux remplacer la couleur `#222222` par `#23282d` je n'aurais qu'à remplacer la valeur de ma variable.

La Syntaxe imbriquée est l'une des choses que je préfère avec le SCSS, je pense qu'un exemple sera plus parlant :

```
//SCSS
.site-navigation {
  ...
  ul {
    ...
  }
  li {
    ...
  }
  a {
    ...
  }
}
```

Et voilà le CSS

```
//CSS
.site-navigation {
  ... }
.site-navigation ul {
  ... }
.site-navigation li {
  ... }
.site-navigation a {
```

Et aussi :

```
// SCSS
a{
  ...
  &:hover {
    ...
  }
}
```

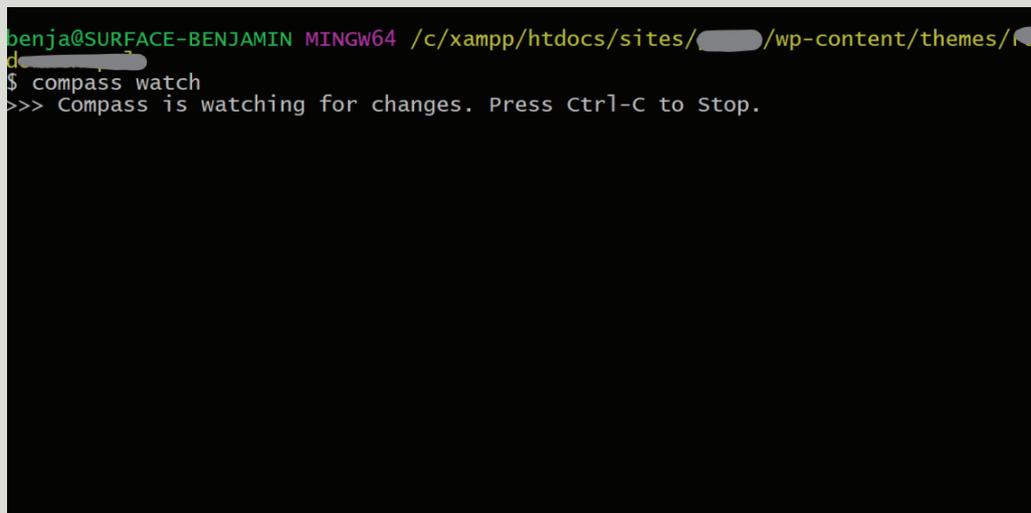
Qui donne en CSS

```
// CSS
a{
  ...
}

a:hover {
  ...
}
```

Pour pouvoir compiler le SCSS il va falloir utiliser du **Ruby**, avec Git et **Compass** il vous suffira de taper la ligne : **compass watch**

Pour pouvoir faire du SCSS il faudra taper la ligne : **compass create**



```
benja@SURFACE-BENJAMIN MINGW64 /c:/xampp/htdocs/sites/[redacted]/wp-content/themes/[redacted]
$ compass watch
>>> Compass is watching for changes. Press Ctrl-C to Stop.
```

Git : compass watch

5. MISE EN LIGNE



Utilisateur ou adresse e-mail

Mot de passe

Rester connecté

Se connecter

[Mot de passe oublié ?](#)

[Retour vers Deust Conf'](#)

5. Mise en ligne

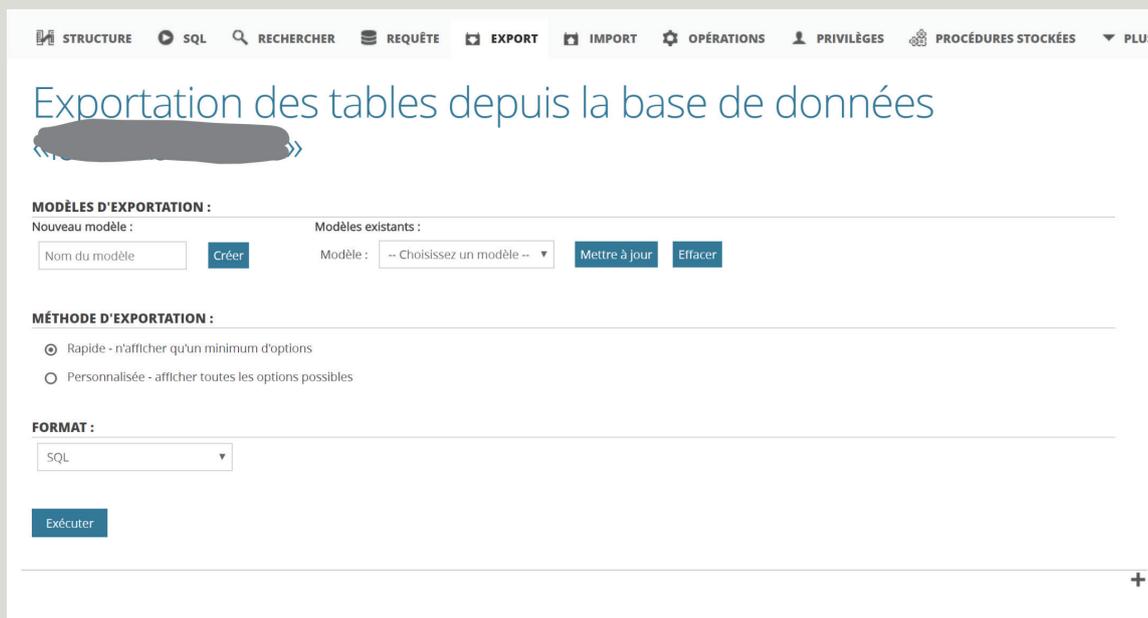
Votre projet terminé, que vous utilisiez WordPress, un autre CMS ou pas de CMS du tout il va bien falloir mettre en ligne votre nouveau site web, que le monde le voit !

Pour cela il va vous falloir un logiciel, **FileZella**, ce logiciel va vous permettre de vous connecter à votre hébergeur et d'y placer vos fichiers.

Pour l'hébergement, quand je fais des sites pour des clients je leur conseille **OVH** parce que c'est l'un des plus grand hébergeur et le SAV est génial. Pour le nom de domaine, pareil, OVH me semble être l'un des meilleurs.

Attention toute fois à vérifier avant d'acheter, si vous utilisez une base de données SQL en local, vous allez aussi en avoir besoin sur internet, choisissez une offre comportant une base de données SQL.

Pour la migration de vos données SQL il va falloir exporter du serveur local, puis importer sur votre nouvel environnement. Allez dans <http://localhost/phpmyadmin>, choisissez votre base de données puis faites «**EXPORT**» > «**Exécuter**», puis «**IMPORT**» sur votre nouvel hébergement.



PhpMyAdmin : EXPORT

Attention, pour WordPress, il vous faudra surement modifier un fichier avant la mise en ligne, le fichier `wp-config.php`, il se trouve à la racine du site, c'est le fichier qui contient les informations de connexion à la base de données SQL, il faudra remplacer les identifiants local par les identifiants distant.

```
21 // ** MySQL settings - You can get this info from your web host ** //
22 /** The name of the database for WordPress */
23 define('DB_NAME', ' ');
24
25 /** MySQL database username */
26 define('DB_USER', 'root');
27
28 /** MySQL database password */
29 define('DB_PASSWORD', ' ');
30
31 /** MySQL hostname */
32 define('DB_HOST', 'localhost');
33
34 /** Database Charset to use in creating database tables. */
35 define('DB_CHARSET', 'utf8mb4');
36
37 /** The Database Collate type. Don't change this if in doubt. */
38 define('DB_COLLATE', '');
39
```

Fichier : wp-config.php

6. CONCLUSION



Utilisateur ou adresse e-mail

Mot de passe

Rester connecté

Se connecter

[Mot de passe oublié ?](#)

[Retour vers Deust Conf'](#)

6. Conclusion

Pour conclure, réaliser des projets, des sites web tous différents les uns des autres permet de toucher à tout, aussi bien de réaliser un thème WordPress que de réaliser un site de A à Z en PHP ou en utilisant des framework comme Bootstrap, Knacss ou Symfony... Ou encore de créer un site WordPress en utilisant Knacss...

Mon stage chez Koality m'a permis d'utiliser du SCSS, une chose dont je ne peux plus me passer depuis.

Je pense que aujourd'hui il faut utiliser WordPress plus souvent, c'est tellement simple d'utilisation par la suite pour les utilisateurs, ils ne sont pas perdu en l'utilisant et apprennent très rapidement.

De plus il existe aujourd'hui des milliers d'extensions qui peuvent être installées sur tout les sites WordPress et un site WordPress n'est plus qu'un simple blog comme avant, nous pouvons aujourd'hui développer de vrais sites web avec ce CMS. Je recommande son utilisation.

Glossaire

CMS veut dire content management system ou en français Système de gestion de contenu est une famille de logiciels destinés à la conception et à la mise à jour dynamique de sites Web ou d'applications multimédia.

WordPress est un CMS permettant de gérer un site web simplement.

Un **plugin** est un petit logiciel web permettant d'ajouter des fonctionnalités aux CMS, dans notre cas, à WordPress.

Les feuilles de style en cascade, généralement appelées **CSS** de l'anglais Cascading Style Sheets, c'est le langage qui permet de mettre en forme de site web.

Le **SCSS** est une évolution du CSS.

Un **framework** est un ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture et des patterns, l'ensemble formant ou promouvant un « squelette » de programme. Il est souvent fourni sous la forme d'une bibliothèque logicielle.

Un serveur **local** permet de créer un serveur virtuel sur le disque dur de l'ordinateur.

Hypertext Preprocessor, plus connu sous son sigle **PHP** (acronyme récursif), est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale.

SQL (sigle de Structured Query Language, en français langage de requête structurée) est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles.

Xampp est un logiciel qui permet de créer un serveur local sur un ordinateur et de pouvoir lancer, entre autres, Apache et SQL.

Apache est le service web qui permet de lire et traduire les fonctions PHP aux

navigateurs web.

phpMyAdmin (PMA) est une application Web de gestion pour les systèmes de gestion de base de données MySQL réalisée en PHP et distribuée sous licence GNU GPL.

Le **ZIP** est un format de fichier permettant l'archivage (utilisation d'un seul fichier pour stocker plusieurs fichiers) et la compression de données (diminution de l'espace occupé sur le support numérique) sans perte de qualité.

Git est un logiciel de gestion de versions décentralisé.

Le **nom de domaine** d'un site est l'adresse visible après le http:// et se terminant par .fr par exemple.

FTP ou File Transfer Protocol (protocole de transfert de fichiers) est un protocole de communication destiné à l'échange informatique de fichiers sur un réseau TCP/IP. Il permet, depuis un ordinateur, de copier des fichiers vers un autre ordinateur du réseau, ou encore de supprimer ou de modifier des fichiers sur cet ordinateur. Ce mécanisme de copie est souvent utilisé pour alimenter un site web hébergé chez un tiers.

L'HyperText Markup Language, généralement abrégé **HTML**, est le format de données conçu pour représenter les pages web. C'est un langage de balisage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement et logiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web.

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives

LESS est un langage dynamique de génération de feuilles de style conçu par Alexis Sellier.

Ruby est un langage de programmation libre. Il est interprété, orienté objet et multi-paradigme.

Compass est un module à ajouter à Git.

FileZella est un logiciel qui permet de transférer des données sur le serveur FTP.

OVH est une entreprise française spécialisée dans les services de cloud computing. Fondé en 1999 par Octave Klaba, le groupe propose des solutions de cloud public et privé, des serveurs dédiés, de l'hébergement mutualisé, du housing (ou colocation), de l'enregistrement de noms de domaines, de la fourniture d'accès Internet par lignes ADSL, VDSL et SDSL, ainsi que de la téléphonie sur IP.

Sitographie

Les sites m'ayant permis de trouver les définitions exactes de certains termes du glossaire (consultés en juin 2017) :

https://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_contenu

https://fr.wikipedia.org/wiki/File_Transfer_Protocol

https://fr.wikipedia.org/wiki/Structured_Query_Language

<https://fr.wikipedia.org/wiki/OVH>

<https://fr.wikipedia.org/wiki/Ruby>

[https://fr.wikipedia.org/wiki/LESS_\(langage\)](https://fr.wikipedia.org/wiki/LESS_(langage))

<https://fr.wikipedia.org/wiki/JavaScript>

https://fr.wikipedia.org/wiki/Hypertext_Markup_Language

<https://fr.wikipedia.org/wiki/Git>

[https://fr.wikipedia.org/wiki/ZIP_\(format_de_fichier\)](https://fr.wikipedia.org/wiki/ZIP_(format_de_fichier))

<https://fr.wikipedia.org/wiki/PhpMyAdmin>

<https://fr.wikipedia.org/wiki/PHP>

<https://fr.wikipedia.org/wiki/Framework>

[https://fr.wikipedia.org/wiki/Sass_\(langage\)](https://fr.wikipedia.org/wiki/Sass_(langage))

https://fr.wikipedia.org/wiki/Feuilles_de_style_en_cascade

Téléchargez WordPress (En français)

<https://fr.wordpress.org>

Téléchargez Xampp

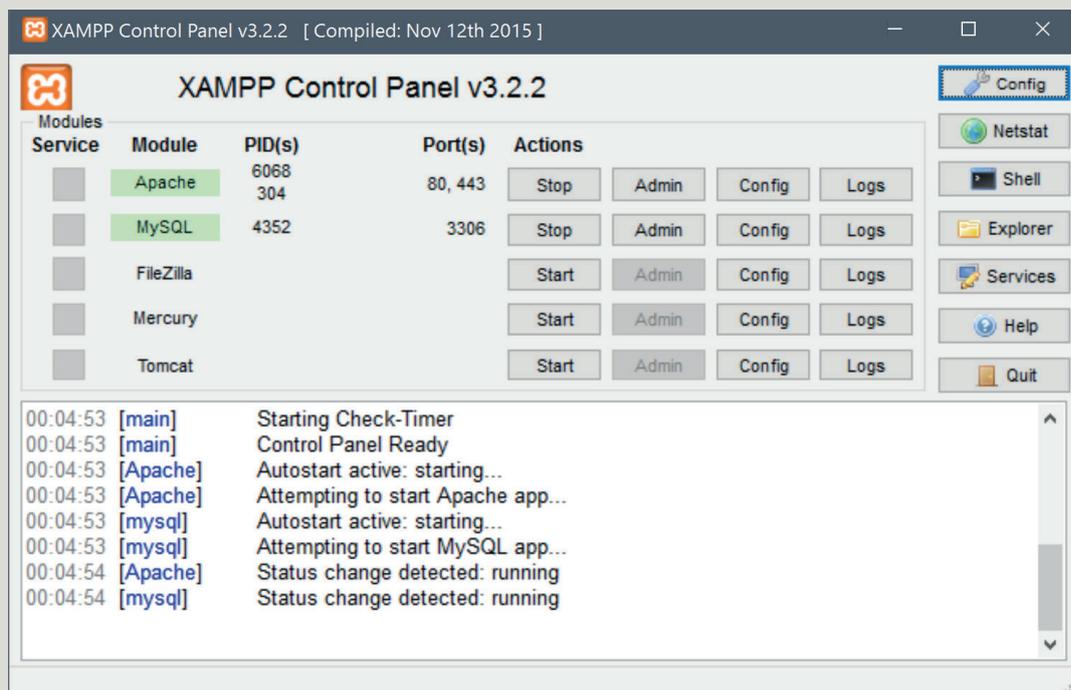
<https://www.apachefriends.org/fr/index.html>

Téléchargez FileZella

<https://filezilla-project.org>

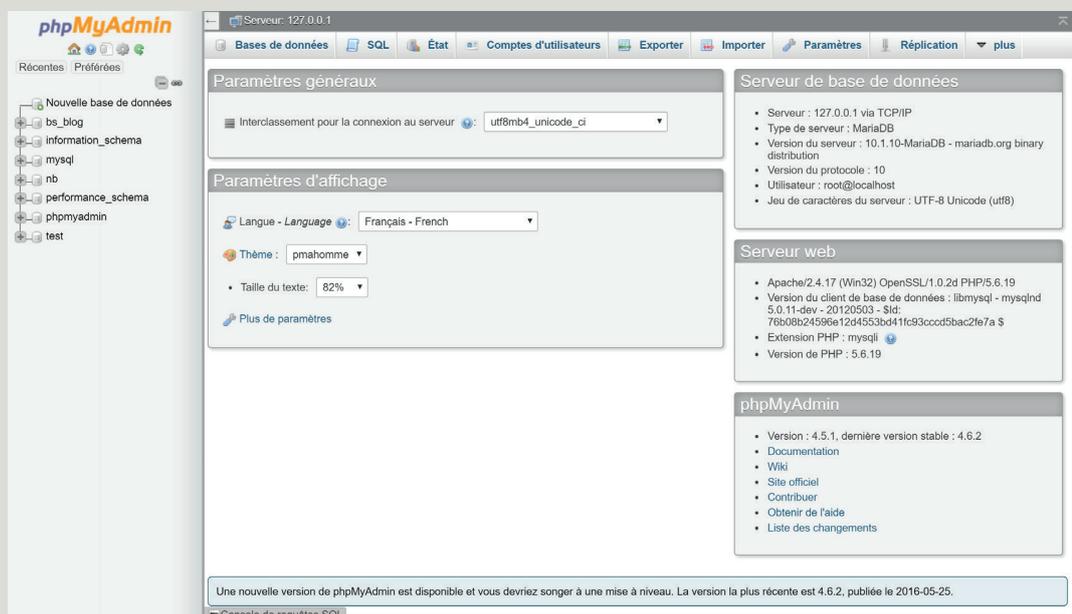
Annexes

Annexe 1

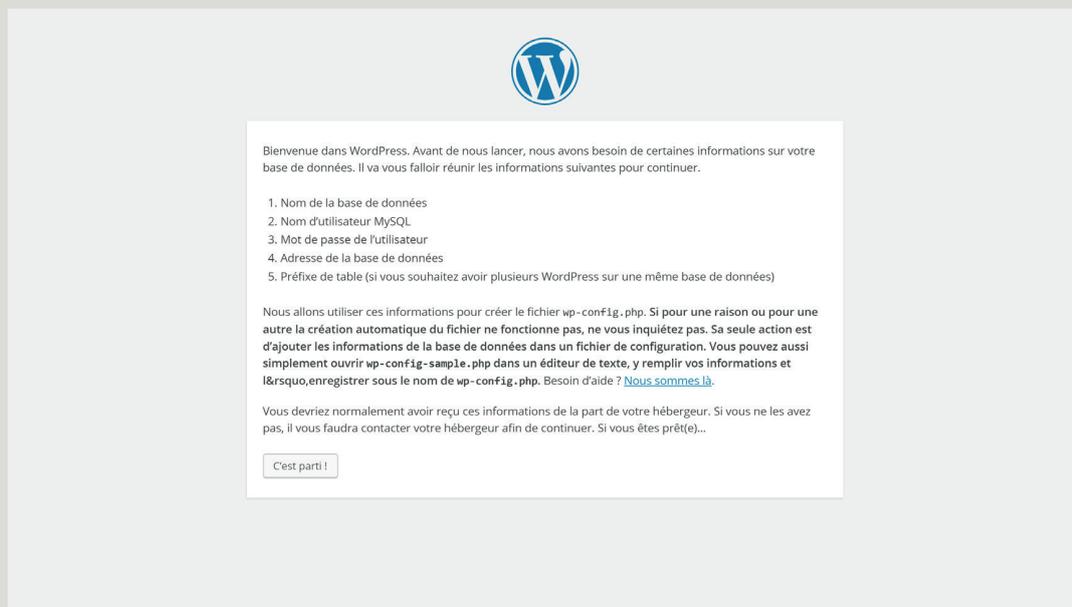


A - Logiciel Xampp

Sur l'image A de l'annexe 1 vous pouvez voir le logiciel Xampp ouvert ainsi que les services Apache et MySQL (SQL) lancés, les boutons d'actions affichent donc le texte «Stop» et les boutons «Admin» (pour administration) sont activés.



B - Panel administrateur PhpMyAdmin



C - Lancement de l'installation WordPress



Vous devez saisir ci-dessous les détails de connexion à votre base de données. Si vous ne les connaissez pas, contactez votre hébergeur.

Nom de la base de données	<input type="text" value="wordpress"/>	Le nom de la base de données avec laquelle vous souhaitez utiliser WordPress.
Identifiant	<input type="text" value="root"/>	Nom d'utilisateur MySQL.
Mot de passe	<input type="text" value="admin"/>	Mot de passe de l'utilisateur MySQL.
Adresse de la base de données	<input type="text" value="localhost"/>	Ce localhost ne marche pas, demandez cette information à l'hébergeur de votre site.
Préfixe des tables	<input type="text" value="wp_"/>	Si vous souhaitez faire tourner plusieurs installations de WordPress sur une même base de données, modifiez ce réglage.

D - Lancement de l'installation WordPress



Identifiant ou adresse de messagerie

Mot de passe

Se souvenir de moi

[Mot de passe oublié ?](#)

[← Retour sur test](#)

E - Installation WordPress terminée

Annexe 2

Les différents fichiers utilisables par WordPress lors de la création d'un thème personnalisé sont :

style.css - C'est le fichier qui va permettre de donner le design du site mais aussi d'y ajouter des informations sur le thème en lui-même grâce à un bout de code en haut du fichier.

index.php - C'est le fichier qui « lance » le site, sans ce fichier votre site ne sera pas visible par les navigateurs web.

footer.php - Ce fichier est, comme son nom l'indique, le bas de page du site, c'est lui qui va afficher le pied de page sur toutes les pages du site.

header.php - Ici aussi ce fichier porte bien son nom, c'est grâce à lui que l'on affiche toutes les informations tel que le menu sur chaque page du site, c'est aussi dans celui-ci qu'on ajoute toutes les balises tels que le titre ou le lien vers le fichier css.

sidebar.php - Ce fichier est utilisé pour afficher une barre de widget par exemple sur l'un des côtés de chaque page du site, le fichier peut être renommé sidebar-left.php ou sidebar-right.php selon l'endroit où il va s'afficher.

page.php - Le fichier page.php permet de donner un style au page du site. On peut créer différents styles de page sur le même site, il suffit de renommer page-article.php par exemple et d'y ajouter un petit code en haut.

single.php - Son nom ne l'indique pas mais ce fichier permet de donner un style aux articles du site (si il y en a).

archive.php - Ce fichier permet de créer une page d'archive qui va répertorier tous les articles du site.

search.php - Ce fichier permet de donner du style à la page de recherche si une barre de recherche est ajoutée au site.

404.php - Ici aussi ce fichier porte son nom à merveille, ce fichier va permettre de personnaliser la page d'erreur 404 du site.

home.php - C'est la page d'accueil du site web.

Annexe 3



CheatSheet

Helpers de largeur

<code>.w10</code> , → <code>.w100</code>	width 10% à 100%
<code>.w100p</code> → <code>.w1140p</code>	width 100px à 1140px
<code>.mw960p</code>	max-width 960px
<code>.mw1140p</code>	max-width 1140px
<code>.wauto</code>	width: auto

Helpers d'espaces

<code>.man</code>	"margin all none" (=margin: 0)
<code>.pan</code>	"padding all none" (=padding: 0)
<code>.mas</code> <code>.mam</code> <code>.mal</code>	"margin all small medium large"
<code>.pas</code> <code>.pam</code> <code>.pal</code>	"padding all small medium large"
<code>.mts</code> <code>.mtm</code> <code>.mtl</code>	"margin top small medium large"
<code>.pts</code> <code>.ptm</code> <code>.ptl</code>	"padding top small medium large"

Helpers responsive

<code>.large-</code>	préfixe pour écrans larges +
<code>.medium-</code>	préfixe pour écrans medium
<code>.small-</code>	préfixe pour écrans small
<code>.tiny-</code>	préfixe pour écrans tiny -

Points de rupture (Sass)

<code>\$tiny</code>	544px
<code>\$small</code>	768px
<code>\$large</code>	1024px
<code>\$extra-large</code>	1440px

Variables d'espaces (Sass)

<code>\$tiny-value</code>	.5rem
<code>\$tiny-plus-value</code>	.7rem
<code>\$small-value</code>	1rem
<code>\$small-plus-value</code>	1.5rem
<code>\$medium-value</code>	2rem
<code>\$medium-plus-value</code>	3rem
<code>\$large-value</code>	4rem
<code>\$large-plus-value</code>	6rem
<code>\$extra-large-value</code>	8rem
<code>\$extra-large-plus-value</code>	12rem
<code>\$ultra-large-value</code>	16rem
<code>\$ultra-large-plus-value</code>	20rem

Variables diverses (Sass)

<code>\$base-font-size</code>	taille de police de base
<code>\$line-height</code>	hauteur de ligne de base
<code>\$h1-size ... \$h6-size</code>	taille de police des titrages
<code>\$font-stack-common,</code> <code>\$font-stack-headings,</code> <code>\$font-stack-monospace</code>	familles de police

Alignements

<code>.txtright</code> <code>.txtright</code> , <code>.txtcenter</code>	alignements de contenus (via text-align)
<code>.left</code> <code>.right</code> <code>.center</code>	alignements de blocs (via margin)

Positionnement flottant

<code>.fl</code> <code>.fr</code>	float-left float-right
<code>.clearfix</code>	contient les flottants
<code>.clear</code>	se place sous les flottants
<code>.bfc</code>	"BFC" contient les flottants et se place à côté des flottants

Positionnements variés

<code>.inbl</code>	élément en display: inline-block
--------------------	----------------------------------

Positionnement flexbox

<code>.flex-container</code>	parent flexbox (horizontal)
<code>.flex-container-v</code>	parent flexbox (vertical)
<code>.flex-item-fluid</code>	enfant flexbox de taille fluide
<code>.flex-item-center</code>	enfant centré (horizontal et vertical)
<code>.item-first</code> <code>.item-last</code>	enfant affiché avant ou après ses frères

Grilles de mise en page

<code>.grid</code>	grille monoligne
<code>.grid-2</code> → <code>.grid-12</code>	grille multilignes de largeurs égales
<code>.has-gutter</code>	application de la gouttière
<code>.has-gutter-l</code> <code>-xl</code>	gouttière large extra-large
<code>.grid-3--reverse</code>	grille 3 colonnes inversée
<code>.pull</code> <code>.push</code>	élément poussé (gauche ou droite)

Grilles taille intermédiaire (small)

<code>.grid-X(-small-Y)</code>	syntaxe type
<code>.grid-3-small-1</code>	1 col (tiny), 1 col (small), puis 3 col
<code>.grid-4-small-2</code>	1 col (tiny), 2 col (small), puis 4 col
<code>.grid-6-small-3</code>	1 col (tiny), 3 col (small), puis 6 col
<code>.grid-6-small-4</code>	1 col (tiny), 4 col (small), puis 6 col

Grilles : tailles individuelles (option)

<code>.one-half</code>	moitié de largeur du parent
<code>.one-third</code>	⅓ de largeur du parent
<code>.one-quarter</code>	¼ de largeur du parent
<code>.one-fifth</code>	⅕ de largeur du parent
<code>.one-sixth</code>	⅙ de largeur du parent
<code>.two-thirds</code>	⅔ de largeur du parent
<code>.three-quarters</code>	¾ de largeur du parent
<code>.five-sixths</code>	⅚ de largeur du parent
<code>.full</code>	toute la largeur du parent

A tilted screenshot of the WordPress login form. It features a white background with a dark grey border. At the top right is the WordPress logo. Below it are two input fields: the first is labeled 'Nom d'utilisateur ou adresse e-mail' and the second is labeled 'Mot de passe'. Below the second field is a checkbox labeled 'Se souvenir de moi'. At the bottom left of the form is a link labeled 'Mot de passe oublié ?'. At the bottom right is a dark grey button with the text 'Se connecter' in white.

UN PROJET

Avec ou sans WordPress ?

Avec ou sans WordPress ? Une question qui revient régulièrement... Dans ce rapport, je vous explique pourquoi utiliser WordPress ? Comment l'installer... Mais aussi pourquoi ne pas utiliser WordPress et quelles sont les alternatives.