

# Premier pas avec WordPress

# Sommaire

Introduction.....	3	II. Installation.....	7
I. WordPress : un CMS pas comme les autres ..	4	A. Installation locale ou distante ? .....	7
A. Qu'est-ce qu'un CMS?.....	4	B. Installation en local .....	7
B. WordPress.com et installation sur un serveur.....	4	C. Windows .....	7
C. Structure et fonctionnement de WordPress.....	5	D. Mac .....	15
D. Sites statiques et dynamiques .....	5		
E. La structure WordPress .....	6		
F. Le fonctionnement modulaire .....	6		

Crédits des illustrations: © Skill & You

## Les repères de lecture



Retour au chapitre



Définition



Objectif(s)



Espace Élèves



Vidéo /  
Audio



Point important /  
À retenir



Remarque(s)



Pour aller  
plus loin



Normes et lois



Quiz

# Introduction

WordPress est un CMS (Content Management System ou « Système de Gestion de Contenu ») basé sur l'utilisation d'une base de données. Contrairement aux idées reçues, WordPress n'est pas un logiciel mais un programme qui s'exécute sur un serveur distant ou local. C'est un ensemble de fichiers qui, lorsqu'ils sont exécutés ensemble rendent l'utilisation de WordPress possible.

Il représente un CMS gratuit et libre, c'est-à-dire que tous les fichiers sont éditables, ce qui le rend personnalisable à volonté et évolutif.

WordPress est un très bon support pour créer et gérer aussi bien un site classique, qu'un blog, un site commercial ou encore un site *corporate*, un site magazine, une galerie ou un portfolio. On peut dire que tout est possible avec WordPress (avec un peu de temps et quelques compétences) car c'est un moteur auquel se rajoutent des thèmes, des extensions et du contenu.

À vous de définir l'orientation que vous souhaitez lui donner.

L'objectif de ce cours est de comprendre comment mettre en place le CMS WordPress puis d'en appréhender son organisation et son utilisation.



### Un CMS (Content Management System)

en anglais ou SGC (Système de Gestion de Contenu) en français est, comme son nom l'indique, un **gestionnaire de contenu**. Il s'agit d'un moteur de publication permettant d'organiser, gérer et publier des pages en agrégeant des contenus (image, vidéos, textes, données chiffrées, etc.).

## I. WordPress : un CMS pas comme les autres

### A. Qu'est-ce qu'un CMS ?

Concrètement, la personne chargée d'une publication se connecte via un navigateur web au back-office (ou l'administration du CMS). Ce back-office est l'interface de gestion du site et s'oppose au front-office qui est le site lui-même. Lorsqu'un CMS est correctement paramétré et/ou personnalisé, la mise à jour et la maintenance d'un site sont donc grandement simplifiées et permettent de traiter un grand nombre de pages et de données.

La plupart des sites de nos jours ne sont plus de simples pages HTML statiques mais fonctionnent via un CMS. Si l'on peut tout à fait créer un CMS de A à Z, ce qui nécessite tout de même des compétences en HTML-CSS et Php-MySQL. La plupart du temps, il s'agit simplement de personnaliser un outil déjà existant, par exemple, d'adapter à ses besoins spécifiques une version standard d'un CMS comme **Drupal** ou **WordPress**.

Non seulement un CMS permet de gérer des contenus, mais également les personnes contribuant à la publication. On définit des rôles (administrateur, contributeur, validateurs, etc.), on définit des droits (modification, validation, publication) permettant à un groupe de personnes de participer à ce que l'on nomme un système de « *workflow* ».

#### Exemple

*Un article est créé par un rédacteur (a) avant d'être révisé et publié par un éditeur (b). La chaîne de travail du « workflow » peut être plus ou moins complexe selon les besoins, les intervenants et les outils utilisés.*

Enfin, un CMS permet également de gérer la personnalisation de contenus par l'intermédiaire de comptes personnalisés. Ainsi, lorsque vous achetez un produit sur un site d'e-commerce, il vous est possible de bénéficier du suivi de la commande que vous avez effectuée. Ce suivi montre des informations qui ne concernent que vous et personne d'autre.

### B. WordPress.com et installation sur un serveur

En réalité, il existe deux possibilités lorsqu'on veut créer un site avec WordPress. On peut utiliser le CMS pour son propre compte (en installation distante à l'aide d'un hébergeur ou en local sur sa machine lorsqu'on veut personnaliser le CMS avant la publication distante) ou bénéficier de l'offre d'hébergement gratuite WordPress.com.

**WordPress.com** vous offre la possibilité d'héberger votre blog gratuitement. Vous pouvez également accéder à des options supplémentaires moyennant un abonnement qui se décompose en plusieurs formules. La version « hébergement » offre l'avantage de la gratuité et de la simplicité, les mises à jour du CMS sont faites automatiquement et la configuration est réduite au strict minimum.

De plus, **WordPress.org** vous offre des fonctionnalités qui ne sont pas présentes sur la version indépendante. Cette solution, pourtant, a le défaut de ses avantages. La personnalisation est en effet limitée, le nombre de thèmes n'est pas aussi abondant que pour la version indépendante, certaines fonctionnalités offertes par les plug-ins en sont absentes et enfin vous ne pouvez pas faire de publicité, ce qui vous interdit des ressources.

Au contraire si vous optez pour la solution avec un hébergeur indépendant vous aurez tout le loisir de personnaliser votre site à votre convenance, de choisir le nom de domaine que vous souhaitez, d'installer tous les plug-ins ou thèmes que vous désirez. Par contre vous devrez vous occuper en permanence de la maintenance du site, des mises à jour de WordPress et de la sécurité.

Le choix de l'une ou l'autre des solutions se fera en fonction de vos compétences, du temps que vous souhaitez consacrer à la personnalisation et également du coût. Si vous avez peur des enjeux techniques auxquels vous allez être confronté, la solution WordPress.com vous permettra en quelques minutes de posséder un blog. L'inconvénient est que vous n'aurez pas la main sur tous les éléments du site et surtout que l'adresse de votre site sera construite comme suit : « **nom-de-votre-site.wordpress.com** ».

N'oubliez pas, toutefois, que l'on peut toujours passer d'une version à l'autre puisqu'il est très facile dans les deux solutions d'exporter ou d'importer les contenus que vous avez saisis.

## C. Structure et fonctionnement de WordPress

### 1. Sites statiques et dynamiques

Concrètement WordPress est un gestionnaire de site dynamique. C'est-à-dire qu'il fonctionne avec une base de données contrairement à un site statique. Pour rappel, le fonctionnement d'un site statique est le suivant : le navigateur affiche la page HTML qui se trouve sur le serveur distant selon le schéma suivant :

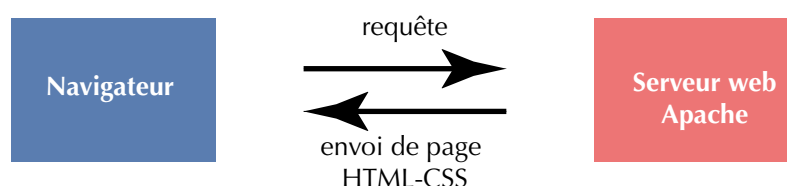


Fig. 1

Un site dynamique est d'une structuration plus complexe. Le navigateur envoie une requête sous la forme d'une URL, le **serveur web** (Apache, par exemple) envoie une information au **serveur d'application** (PHP, par exemple) qui récupère les données dans le **serveur de base de données** (MySQL, par exemple) afin de construire à la volée une page HTML qui est renvoyée vers le navigateur.

Soit le schéma suivant :

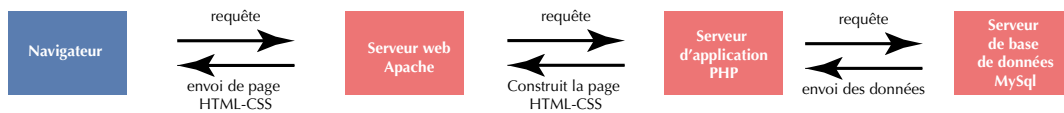


Fig. 2

## 2. La structure WordPress

Lorsque vous allez entrer dans l'univers de WordPress vous allez aborder toute une série de notions et d'outils parfois propres au CMS, parfois partagés avec d'autres. Par défaut WordPress se compose, comme on l'a dit plus haut, d'une interface de gestion appelée indifféremment administration ou back-office sur laquelle on se connecte via un navigateur afin de paramétrer le site (le front-office appelé également le front) ou créer des contenus (pages, articles, etc.). Il va donc être question de **widgets** (petits blocs de fonctionnalités que l'on peut disposer à sa guise sur la page), de **plug-ins** ou d'**extensions** (fonctionnalités supplémentaires qu'il s'agit d'installer et parfois de paramétrer avant de pouvoir les utiliser) et des **thèmes** (l'apparence du site, mais également certaines fonctionnalités ou widgets supplémentaires, voire même une philosophie d'organisation et de présentation des contenus).

## 3. Le fonctionnement modulaire

On peut décrire le fonctionnement de WordPress comme un système de briques modulaires. Un premier module permet de faire fonctionner le moteur de publication et apporte les fonctionnalités de base. On ajoute ensuite des modules de deuxième niveau qui vont apporter des fonctionnalités supplémentaires. Enfin, un troisième niveau de modules permet de générer l'apparence et le design graphique du site web. À tout instant il est possible d'ajouter ou retirer les modules de deuxième ou de troisième niveau.

Cette souplesse est presque totalement indépendante des contenus qui ont pu être saisis dans l'interface d'administration.

Si l'on devait décomposer le fonctionnement de WordPress, on aurait :

- **Première couche** : le noyau (le moteur) ;
- **Deuxième couche** : les plug-ins (des fonctionnalités supplémentaires) ;
- **Troisième couche** : le thème (l'apparence).

En réalité, lorsqu'on installe WordPress, par défaut, on se retrouve avec un noyau minimal. On aura donc nativement le moteur de publication mais également certaines fonctionnalités et un thème qui s'appelle « **Twenty + un nombre** » (Twenty-ten, Twenty-eleven, etc.). Il faut également savoir que certains plug-ins ou thèmes peuvent faire planter WordPress ou le ralentir. En effet, les thèmes et les plug-ins étant souvent élaborés par des développeurs indépendants, il n'est pas rare que leur installation engendre des conflits, soit entre les plug-ins soit avec le noyau.

## II. Installation

### A. Installation locale ou distante ?

On peut se poser la question de l'intérêt de l'installation locale de WordPress. En effet, pourquoi s'embêter à configurer un serveur local alors qu'un hébergement offre l'avantage de posséder tous les outils permettant de faire tourner correctement le CMS ?

Pourquoi passer du temps à manipuler WordPress sur sa machine alors que l'on va devoir à terme publier celui-ci sur un serveur et répéter alors tout le processus d'installation ?

En réalité, il ne faut pas considérer que l'un et l'autre sont antinomiques ou qu'il s'agit d'une perte de temps. Lorsque l'on construit un site web, après l'étape de conception, il faut mettre en place l'outil de publication ou le CMS. Si l'on travaille directement sur le serveur distant, on risque d'avoir des contenus accessibles alors même qu'ils ne sont pas finalisés. Dans le même ordre d'idées, comment travailler sur une refonte de site (changement de CMS, passage d'un site statique à un site « fièrement propulsé » par WordPress) sans posséder au moins temporairement un serveur de test et/ou de développement. De plus, lors de la phase de découverte de WordPress, il est tout à fait intéressant de pouvoir tester et expérimenter allègrement sans se soucier du résultat.

### B. Installation en local

#### 1. Windows

Pour Windows on peut installer un package comprenant le serveur Apache, la base de données MySQL ainsi que le serveur d'application PHP. Il existe plusieurs solutions : **XAMP** ou **WAMP**. XAMP est un logiciel libre que l'on peut installer indifféremment sur Windows, Mac ou Linux. WAMP est une solution spécialement dédiée à Windows comme l'indique le « W ».

##### a. Installation de WAMP

La première étape est donc de télécharger le package sur le site [wampserver.com](http://wampserver.com). Choisissez WAMP 32 bits ou 64 bits suivant votre système d'exploitation. Une fois le téléchargement terminé on lance le fichier d'installation et on installe WAMP à la racine, c'est-à-dire que l'on doit avoir une adresse locale de type : « C:\wamp ». On vous demande également quel va être le navigateur par défaut, il suffit d'aller chercher le « .exe » de votre navigateur favori. Vous avez également, une fois l'installation terminée, la possibilité de paramétrer les valeurs de la fonction « mail () » de PHP (adresse et serveur mail) ou de sauter cette étape si vous ne savez pas ce que cela veut dire.

Ensuite, lancer WAMP comme pour n'importe quel logiciel en passant par le bouton démarrer de Windows. Une icône apparaît en bas à droite de la barre de tâche de Windows.



Fig.3

Celle-ci est rouge si les serveurs n'ont pas démarré. L'icône peut également être orange, ce qui signifie que les serveurs démarrent.

Il arrive qu'elle reste bloquée sur cette couleur. Cela signifie qu'il y a un problème d'installation. Quand l'icône se stabilise en vert, les serveurs ont démarré et sont prêts à fonctionner.



Fig.4

Lorsqu'on démarre WAMP, suivant la configuration, les serveurs peuvent démarrer automatiquement ou attendre qu'on leur demande de démarrer. Pour les faire démarrer, si l'icône est rouge, il suffit de cliquer gauche sur celle-ci puis sur clique « *Start All Services* ».

#### b. Création de la base de données

On va démarrer phpMyAdmin en cliquant sur l'icône verte puis sur « *phpMyAdmin* », comme sur la copie d'écran ci-dessous :

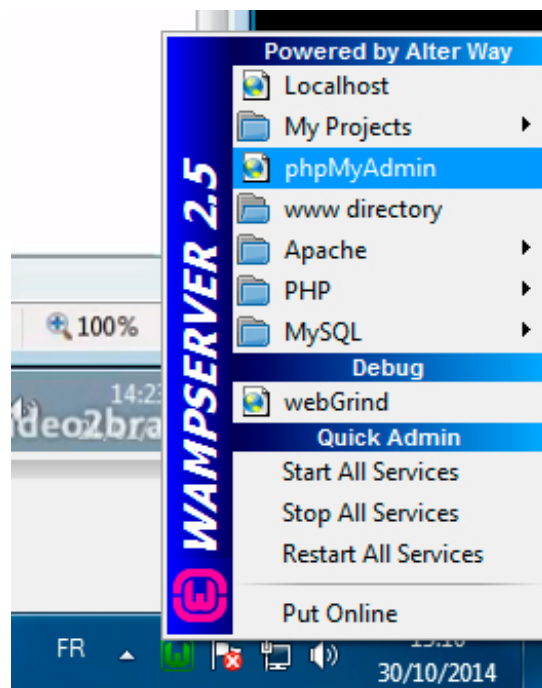


Fig.5



Lorsqu'on se retrouve sur l'interface de phpMyAdmin, il faut cliquer sur le bouton « Bases de données » :

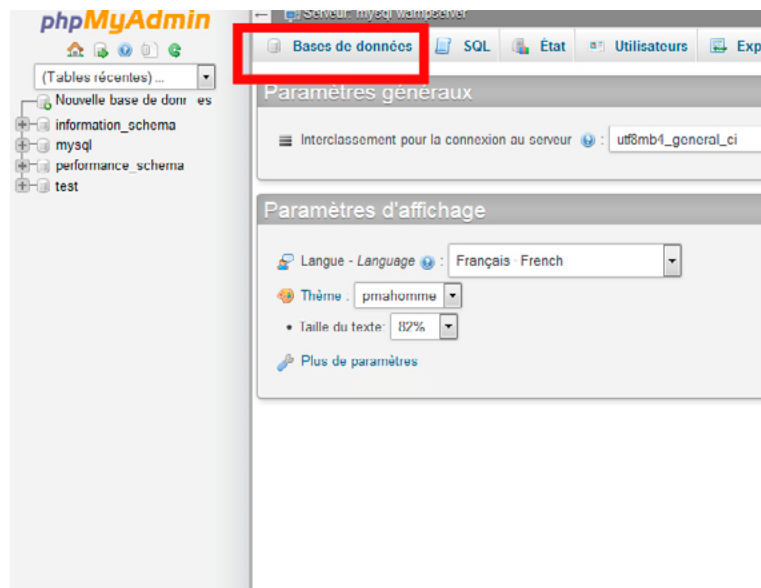


Fig.6

Dans le champ « **Créer une base de données** », on indique le nom de la base sans accent, espace ni majuscule. Puis on clique sur « créer » :

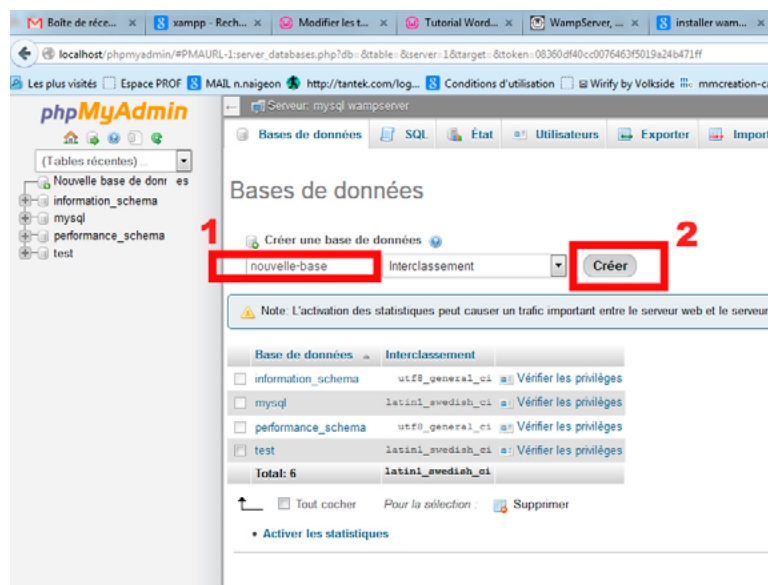


Fig.7

### c. Installation de WordPress

On télécharge la dernière version de WordPress à partir du site <https://fr.wordpress.com/>.

On récupère une archive dont le nom est « **WordPress-numéro-de-version-fr\_FR** ». On récupère le dossier WordPress qui se trouve à l'intérieur et on va le coller dans le dossier « www » de WAMP.

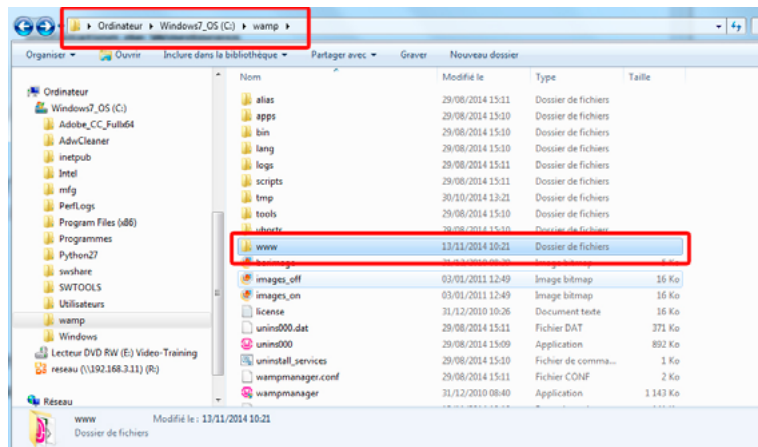


Fig.8

Vous pouvez renommer le dossier WordPress si vous le désirez, mais c'est avant de commencer l'installation du CMS qu'il faut le faire, soit au moment de copier le dossier dans le fichier « www ».

Étudions ce que l'on obtient avec le navigateur quand on se rend sur le serveur local dont l'adresse dans le navigateur est « localhost ».

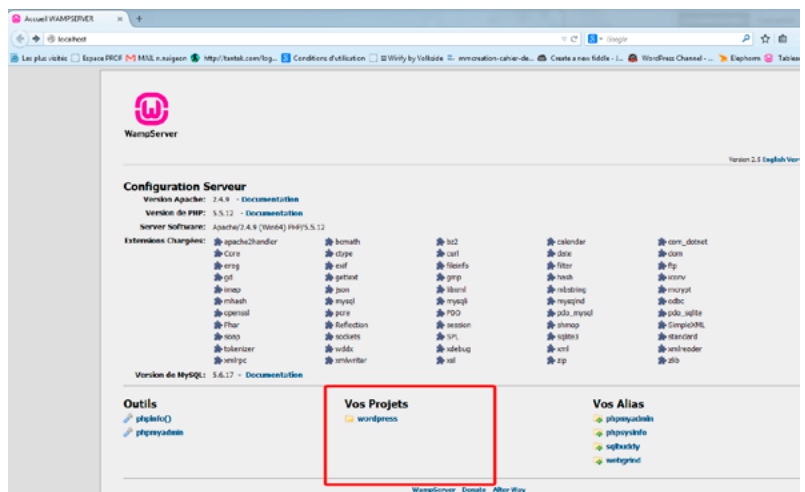


Fig.9

La zone « Vos Projets » correspond en fait à tout ce que l'on trouve dans le dossier « www » de WAMP. Pour faire une petite expérience, ajoutons-y un dossier nommé « mon-site » et un fichier HTML qu'on appellera « page.html ».

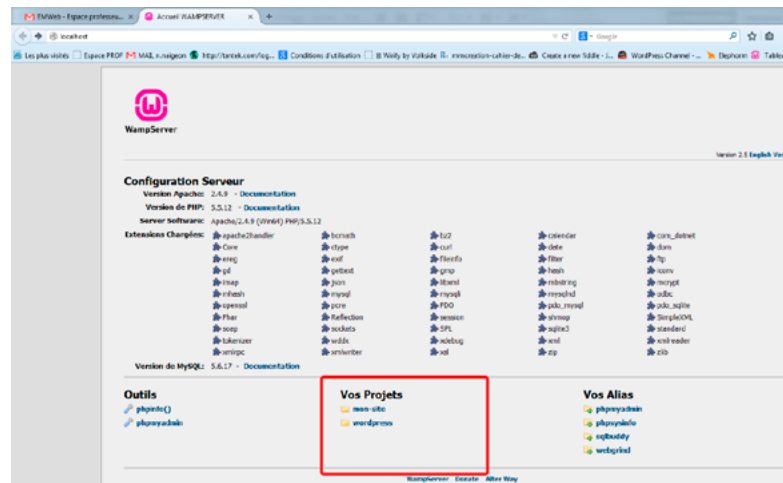
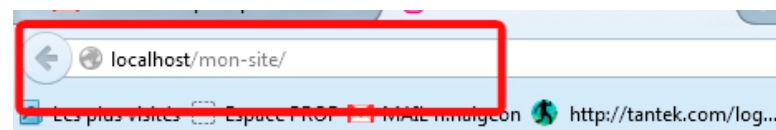


Fig. 10

On remarque qu'une nouvelle icône apparaît dans la zone « Vos Projets » qui correspond bien au dossier « mon-site » que l'on a ajouté. Par contre, notons que l'on aperçoit aucun fichier. Continuons notre exploration en cliquant sur le dossier « mon-site » qui contient des fichiers HTML, entre autres. Ici il faut dire que depuis WAMP 2.5, la philosophie a changé, il faut donc soit écrire directement : **https://localhors/nom-du-dossier**, soit modifier certains paramètres directement dans les fichiers WAMP.



## Index of /mon-site

[ICO]	<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
[PARENTDIR]	<a href="#">Parent Directory</a>		-	
[TXT]	<a href="#">chien.html</a>	2014-11-10 12:46	8.9K	
[DIR]	<a href="#">css/</a>	2014-11-13 11:16	-	
[DIR]	<a href="#">images/</a>	2014-11-13 11:16	-	
[TXT]	<a href="#">page-1.html</a>	2014-11-10 12:50	7.0K	

*Apache/2.4.9 (Win64) PHP/5.5.12 Server at localhost Port 80*

Fig. 11

Les différents éléments qui composent le dossier « mon-site » ; fichiers HTML, sous-dossiers sont visibles. On remarque également que le nom des versions des serveurs Apache et PHP ainsi que le port utilisé sont indiqués. L'adresse dans le navigateur est « **localhost/mon-site** ». Ajoutons un fichier index.html et cliquons de nouveau sur « **mon-site** » à partir de l'adresse « **localhost** ».

Cette fois-ci nous n'explorons plus le contenu du dossier par l'intermédiaire du navigateur, celui-ci nous affiche la page index.html.



Fig. 12

C'est, en effet, le propre d'un serveur web que d'afficher par défaut certains types de fichiers prédéfinis (« index.html », « index.php »).

Ceci nous permet de comprendre pourquoi lorsqu'on va cliquer sur notre dossier WordPress décompressé (ou tout autre nom de dossier si on l'a renommé), une page va s'afficher.



Fig. 13

À nouveau, si lorsque vous cliquez sur le dossier WordPress vous atterrissez sur une page d'erreur, essayez d'écrire l'adresse complète dans la barre d'adresse, en ajoutant « **localhost** » entre « http:// » et le nom du site, soit : « **http://localhost/nom-du-site** ».

On clique sur « c'est parti » pour se retrouver sur un formulaire :

WordPress - Fichier de con...

localhost/wordpress/wp-admin/setup-config.php?step=1&language=fr\_FR

Google

Les plus visités Espace PROF MAIL n.naigeon http://tante.com/log... Conditions d'utilisation Wifry by Volkside mcreation-cahier-de...

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

1 Nom de la base de données nouvelle-base Le nom de la base de données dans laquelle vous souhaitez installer WordPress.

2 Identifiant root Votre identifiant MySQL.

3 Mot de passe ...et son mot de passe MySQL.

4 Adresse de la base de données localhost Si localhost ne fonctionne pas, votre hébergeur doit pouvoir vous donner la bonne information.

5 Préfixe des tables wp\_ Si vous souhaitez faire tourner plusieurs installations de WordPress sur une même base de données, modifiez ce réglage.

Envoyer

Fig. 14

1. Le nom de la base de données créée plus haut.
2. L'identifiant MySQL, en local est toujours « root » (qui veut dire racine), si on est sur un serveur distant, l'identifiant est fourni par l'hébergeur.
3. Le mot de passe de la base de données, sur Windows : on laisse le champ vide en local mais, comme pour l'identifiant, l'hébergeur en fournit un si on est sur un serveur distant.
4. L'adresse de la base de données, en local cela sera « localhost », sur un serveur distant l'URL de la base vous sera fournie par l'hébergeur.
5. On peut indiquer un préfixe ici afin de différencier plusieurs installations de WordPress, on évitera d'utiliser **WP\_** sur un serveur distant pour des raisons de sécurité.

S'il n'y a pas d'erreur, on obtient ceci :



Fig. 15

Il suffit dès lors de remplir le formulaire suivant :

Fig. 16

1. Le titre du site.
2. L'identifiant de l'administrateur. Par défaut, WordPress utilise « *admin* », pour des raisons de sécurité, encore, sur un serveur distant, on évitera absolument de le conserver tel quel.
3. Le mot de passe de l'administrateur.
4. Votre adresse de messagerie. Cela permet par exemple de modifier son mot de passe si on le perd (évidemment dans le cas d'un WordPress hébergé).
5. Ici c'est également une option que l'on choisira uniquement si l'on a un hébergement.

On obtient ensuite :



Quel succès !

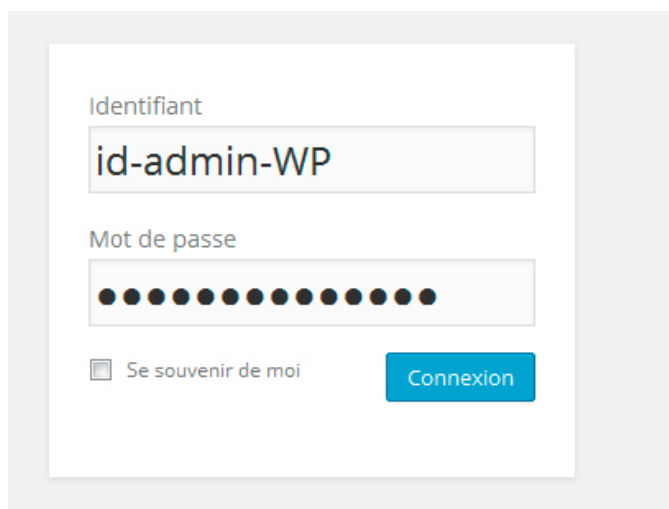
WordPress est installé. Vous attendiez-vous à d'autres étapes ? Désolé de vous décevoir ;-)

Identifiant	id-admin-WP
Mot de passe	Le mot de passe que vous avez choisi.

Connexion

Fig. 17

En cliquant sur connexion on tombe sur l'écran suivant :



Identifiant

id-admin-WP

Mot de passe

☐ Se souvenir de moi

Connexion

Fig. 18

Il suffit de remplir les champs avec les valeurs que l'on a rentrées précédemment pour se connecter au back-office ou administration de WordPress.

## 2. Mac

Pour Mac on peut également installer un package comprenant le serveur Apache, la base de données MySQL ainsi que le serveur d'application PHP. Le package le plus populaire est MAMP - le M vaut pour « Macintosh ».

### a. Installation de MAMP

La première étape est donc de télécharger le package sur le site <https://www.mamp.info/en/>. On va sur la page *Downloads* pour télécharger le package qui se compose à la fois de MAMP et de MAMP-PRO (version payante offrant plus de fonctionnalités mais non nécessaire pour notre sujet).

Une fois le téléchargement terminé on lance le fichier d'installation (.pkg) et on installe MAMP. Le login et le mot de passe de la machine sont demandés par l'installateur pour effectuer l'opération. Le logiciel se retrouve automatiquement dans le dossier « Applications ». Attention le package installe également MAMP-PRO qui ne nous intéresse pas ici.

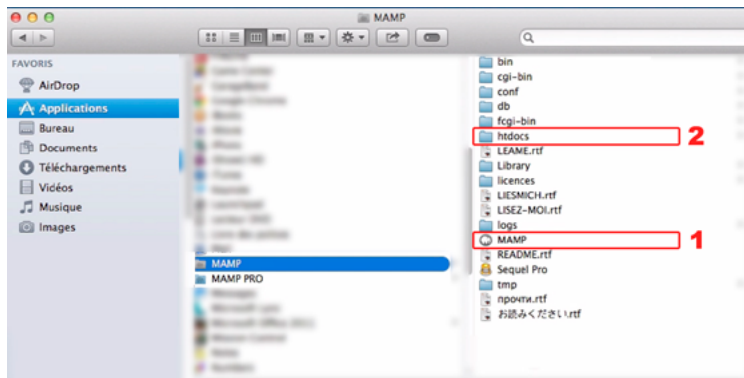


Fig. 19

Dans le dossier MAMP on trouve en (1) l'application qui démarre MAMP et également le dossier « htdocs » (2) qui va accueillir tous les fichiers des sites en local. Lorsqu'on démarre MAMP il faut faire attention à bien choisir MAMP et non MAMP-PRO.

Une fois que l'on a lancé MAMP, il faut demander à démarrer les serveurs. On observe que l'icône de démarrage passe au vert et que les cases Apache et MySQL sont également en vert - ce qui signifie que les serveurs ont démarré et sont prêts à fonctionner.

Si la page d'accueil n'a pas automatiquement été chargée, on peut demander à le faire en cliquant sur « Ouvrir la page d'accueil ».

Celle-ci a une adresse en : **localhost:8888/MAMP/**

## b. Création de la base de données

Une fois que l'on a ouvert la page d'accueil sur le navigateur il suffit de cliquer sur : « **Outils > phpMyAdmin** ». On va démarrer phpMyAdmin en cliquant sur l'icône vert puis sur « phpMyAdmin », comme sur la copie d'écran ci-dessous :



Fig. 20



Lorsqu'on se retrouve sur l'interface de phpMyAdmin, il faut cliquer sur le bouton « Bases de données » :

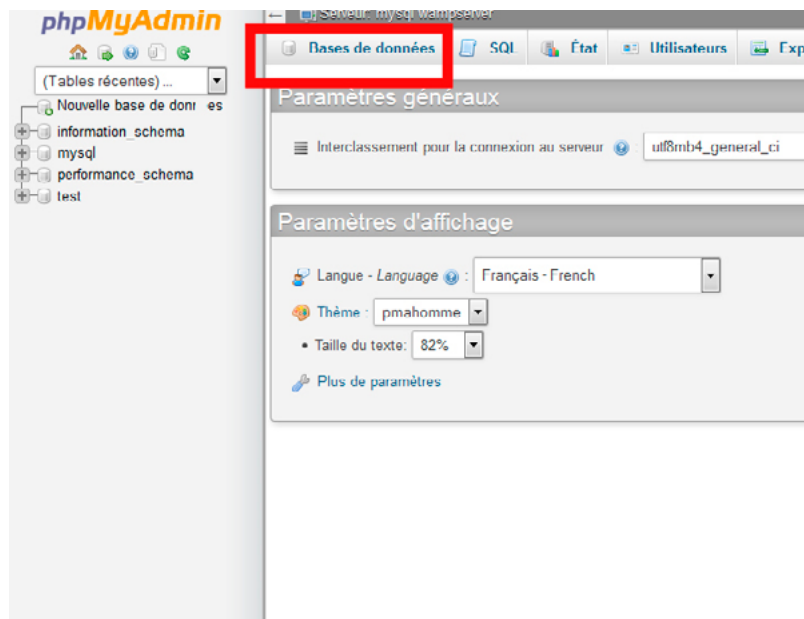


Fig.21

Dans le champ « créer une base de données », on indique le nom de la base sans accent, espace ni majuscule. Puis on clique sur « créer » :

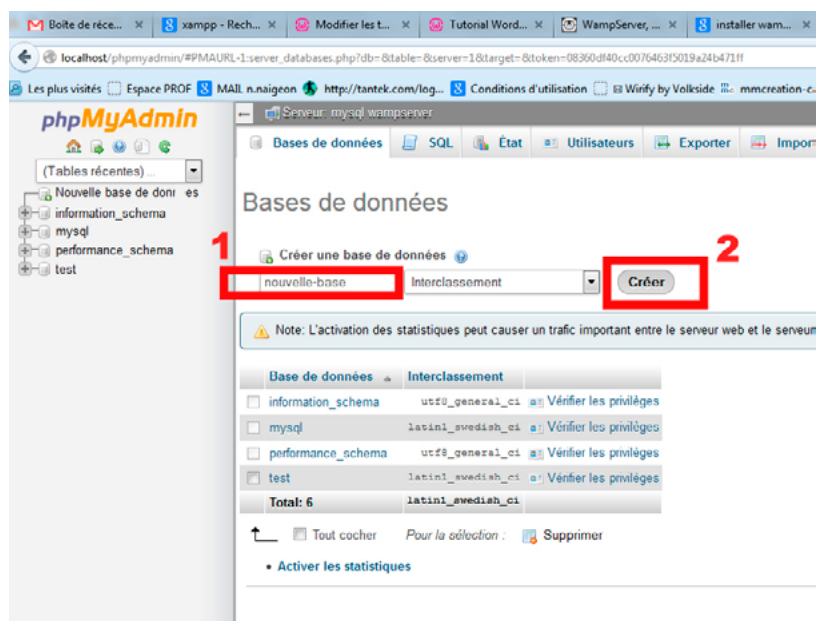


Fig.22

### c. Installation de WordPress

On télécharge la dernière version de WordPress à partir du site <https://fr.wordpress.com/>.

On récupère une archive dont le nom est « WordPress-numéro-de-version-fr\_FR ». On récupère le dossier WordPress qui se trouve à l'intérieur et on va le coller dans le dossier « **htdocs** » de MAMP.

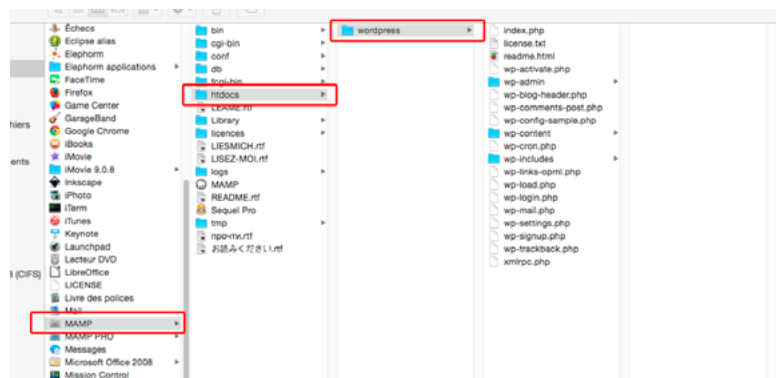


Fig.23

Vous pouvez renommer le dossier WordPress si vous le désirez, mais c'est avant de commencer l'installation du CMS qu'il faut le faire, soit au moment de copier le dossier dans le fichier « **htdocs** » ou juste après.

Étudions ce que l'on obtient avec le navigateur quand on se rend sur le serveur local, soit automatiquement au démarrage des serveurs MAMP, soit en cliquant sur « **Ouvrir la page WebStart** ».

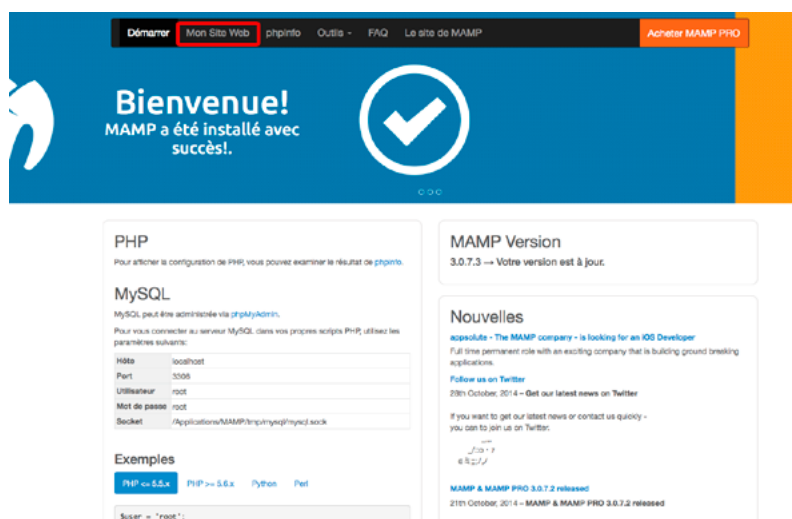


Fig.24

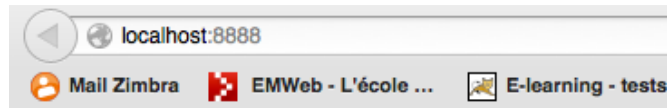
Lorsqu'on clique sur « **Mon site web** », on se rend à l'adresse « **localhost:8888** » qui correspond à l'adresse du serveur local et nous permet de voir tout ce que l'on trouve dans le dossier « **htdocs** » de MAMP.

## Index of /

- [wordpress/](#)

Fig.25

Pour faire une petite expérience, ajoutons-y un dossier nommé « **mon-site** », et rafraichissons la page.

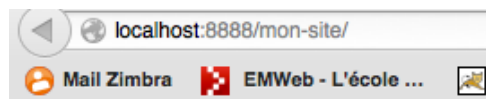


## Index of /

- [mon-site/](#)
- [wordpress/](#)

Fig.26

On remarque qu'un nouvel élément apparaît et qu'il correspond bien au dossier « **mon-site** » que l'on a ajouté. Continuons notre exploration en cliquant sur le dossier « mon-site » qui contient des fichiers HTML, entre autres.

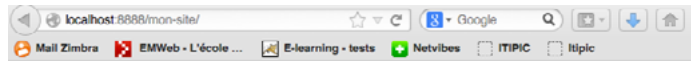


## Index of /mon-site

- [Parent Directory](#)
- [chien.html](#)
- [css/](#)
- [images/](#)
- [page-1.html](#)

Fig.27

Les différents éléments qui composent le dossier « mon-site » ; fichiers HTML, sous-dossiers sont visibles. L'adresse dans le navigateur est « **localhost:8888/mon-site** ». Ajoutons un fichier *index.html* et cliquons de nouveau sur « mon-site » à partir de l'adresse « **localhost:8888** ». Cette fois-ci nous n'explorons plus le contenu du dossier par l'intermédiaire du navigateur, celui-ci nous affiche la page *index.html*.



## Ceci est mon fichier index.html

Fig. 28

C'est, en effet, le propre d'un serveur web que d'afficher par défaut certains types de fichiers prédéfinis (« **index.html** », « **index.php** »).

Ceci nous permet de comprendre pourquoi lorsqu'on va cliquer sur notre dossier WordPress décompressé (ou tout autre nom de dossier si on l'a renommé), une page va s'afficher.

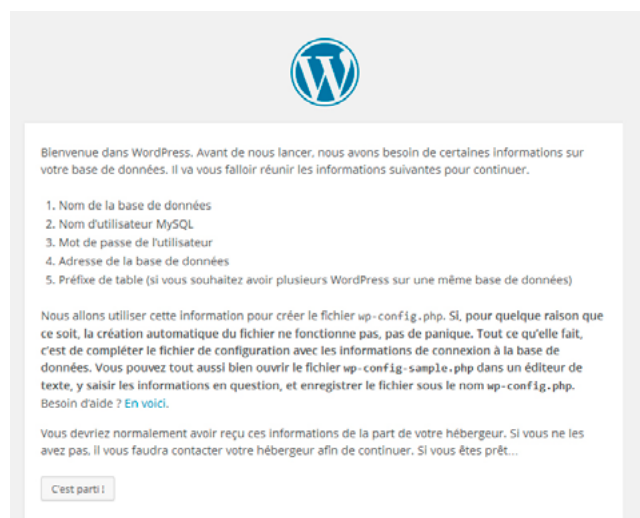


Fig. 29

On clique sur « c'est parti » pour se retrouver sur un formulaire :

 A screenshot of the WordPress database configuration form. At the top is the WordPress logo. Below it, the text reads: 'Vous devez saisir ci-dessous les détails de connexion à votre base de données. Si vous ne les connaissez pas, contactez votre hébergeur.' The form contains five input fields with labels and hints: 'Nom de la base de données' (nouvelle-base), 'Identifiant' (root), 'Mot de passe' (root), 'Adresse de la base de données' (localhost), and 'Préfixe des tables' (wp\_). Each field has a corresponding hint on the right. At the bottom, there is an 'Envoyer' button.

Fig. 30

1. Le nom de la base de données créée plus haut.
2. L'identifiant MySQL, en local c'est toujours « root » (qui veut dire racine), si on est sur un serveur distant l'identifiant est fourni par l'hébergeur.
3. Le mot de passe de la base de données, sur Mac, c'est également « root » mais comme pour l'identifiant, l'hébergeur en fournit un si on est sur un serveur distant.
4. L'adresse de la base de données, en local sera « localhost », sur un serveur distant l'URL de la base vous sera fournie par l'hébergeur.
5. On peut indiquer un préfixe ici afin de différencier plusieurs installations de WordPress, on évitera d'utiliser **WP\_** sur un serveur distant.

Si cela ne fonctionne pas, l'étape de renseignement de la base de données vous affichera un message d'erreur et vous invitera à recommencer.

S'il n'y a pas d'erreur, on obtient ceci :



Fig.31

Il suffit dès lors de remplir le formulaire suivant :

Fig.32

1. Le titre du site.
2. L'identifiant de l'administrateur. Par défaut, WordPress utilise « *admin* », pour des raisons de sécurité, sur un serveur distant, on évitera absolument de le conserver tel quel.
3. Le mot de passe de l'administrateur. Si vous êtes sur un serveur distant, préférez un mot de passe avec des majuscules et des chiffres afin de prévenir tout problème de piratage.
4. Votre adresse de messagerie. Cela permet par exemple de modifier son mot de passe si on le perd (évidemment dans le cas d'un WordPress hébergé).
5. Ici c'est également une option que l'on choisira uniquement si l'on a un hébergement.

On obtient ensuite :

## Quel succès !

WordPress est installé. Vous attendiez-vous à d'autres étapes ? Désolé de vous

Identifiant	id-admin-WP
Mot de passe	Le mot de passe que vous avez choisi.

Fig.33

En cliquant sur connexion on tombe sur l'écran suivant :

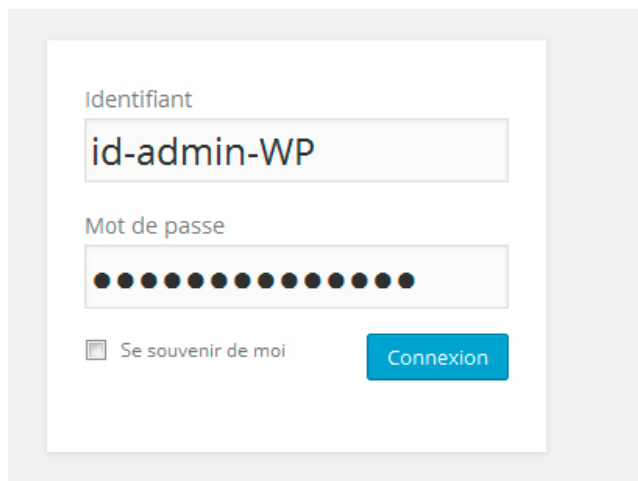


Fig.34

Il suffit de remplir les champs avec les valeurs que l'on a rentrées précédemment pour se connecter au back-office ou administration de WordPress.



