

# Initiation à Wordpress 3

Licence Pro ATC – IUT d'Arles 2010

Arnaud Tanielian - [arno.tanielian@gmail.com](mailto:arno.tanielian@gmail.com) - @Danetag

## Wordpress

---

0 – Le projet .....	5
1 – Avant de Commencer .....	5
Qu'est-ce qu'un CMS ? A quoi ça sert ? .....	5
Let me introduce you: Wordpress! .....	7
Pourquoi utiliser Wordpress ? .....	7
Le principe de Wordpress .....	7
2 - Lancement de Wordpress .....	10
Logiciels indispensables.....	10
Serveur Apache, MYSQL .....	10
Editeur de texte .....	10
Browsers.....	11
Plugins FF .....	11
Installation de Wordpress .....	12
Mettre tout en place .....	12
L'arborescence.....	15
Front Office .....	18
Back Office.....	19
L'installation d'un thème .....	20
Installation d'un plugin.....	22
Installation d'un widget.....	23
3 - Insérer du texte .....	26
Ecrire son premier article : découverte de l'éditeur visuel.....	26
Section 1 : Editeur de texte enrichi (ou WYSIWYG) .....	27
Section 2 : Publier.....	28
Section 3 : Catégorie/Mots-clefs.....	28
Section 4 : Portfolio Post, extrait, rétroliens ... ..	29

4 - Modifier le contenu .....	30
Modifier les pages et articles .....	30
Création des pages .....	30
Nouveauté Wordpress 3 : Le menu .....	30
Mise en place des pages .....	32
Créer vos propres « Modèles Pages » .....	32
Nouvelle fonction de WordPress 3 : "Custom Post Types" .....	34
Une fonction pour customiser l'admin .....	34
Déclarer le custom post type .....	34
De retour sur nos pages .....	36
Nouvelle fonction de WordPress 3 : "Custom Taxonomy" .....	39
Retour à la méthode my_custom_init() .....	39
Afficher de beau projet : custom loop ! .....	40
5 - Modifier la présentation .....	43
Modifier le thème par défaut : Twenty Ten .....	43
Les modifications de bases .....	43
L'éditeur de thème .....	43
Comment intégrer son propre template ? .....	44
Monter la page HTML .....	44
Les standards CSS .....	44
Partir du template de base .....	44
6 - Gérer son site en ligne .....	45
Choisir un hébergement + nom de domaine .....	45
Exporter .....	45
Mettre en production .....	45
Importer .....	46
Quelques astuces .....	46
De beaux liens pour un beau référencement .....	46
7 - Annexes .....	47

Option : Création d'un Vhost.....	47
8 – Source .....	49

## 0 – Le projet

Construire un site personnel, de type portfolio et/ou blog, afin d'avoir un site complet et maintenu à jour sur le net.

Objectif : Avoir une présence sur le net, pouvoir y montrer ses travaux, son CV. Pourquoi pas tenir à jour un blog, qui peut être un vrai plus dans la vie professionnel.

## 1 – Avant de Commencer

### Qu'est-ce qu'un CMS ? A quoi ça sert ?

Les CMS - Content Management System sont une famille de logiciels destinés à la conception et à la mise à jour dynamique de site web ou d'application multimédia. Ils partagent les fonctionnalités suivantes :

- ils permettent à plusieurs individus de mettre à jour une administration, où les utilisateurs sont définis par des rôles, avec des actions possibles ou non sur des ressources.

Ex : Un rôle « super admin » à tous les privilèges (création, édition, suppression) sur un article, un rôle « éditeur » ne peut pas créer mais seulement éditer un article, un simple utilisateur ne peut pas accéder à l'administration.

- ils fournissent une chaîne de publication (*workflow*) offrant par exemple la possibilité de mettre en ligne le contenu des documents. On peut donc faire passer un article par plusieurs « états » : brouillon, non-publié puis publié.
- ils permettent de séparer les opérations de gestion de la forme et du contenu : Système de Template, interface d'administration déjà prête.
- ils permettent de structurer le contenu (utilisation de *FAQ*, de *blogs*, de *forums de discussion*, etc.). Selon le type de CMS, le contenu est donc déjà structuré comme un blog ou un forum par exemple.

La plupart du temps, c'est concrètement un ensemble de fichiers à télécharger et à mettre en place sur son FTP ou serveur local. La 1ère fois que l'on accède à l'index, on est en général invité à entrer les informations pour la connexion à la base de donnée, puis quelques informations générales comme le titre ou le login/mdp de l'admin principal.

**L'avantage** principal d'un CMS est donc **la rapidité** du déploiement pour un site dynamique. C'est une solution « clef en main », adaptée pour des projets à petits budgets souvent (Cela dépend du projet et du CMS bien sûr. Par exemple, *Drupal* est assez compliqué de prise en main et très puissant).

L'interface d'administration étant déjà faite (back office), il « ne reste plus » qu'à s'occuper du front office. Et encore, avec le système de templates, cette tâche peut aussi être facilitée.

**L'inconvénient** est donc la **personnalisation**. Il existe des templates tout fait, des widgets ou des plugins pour avoir plus de fonctionnalités, ou encore du « développement » spécifique à chaque CMS pour customiser son front office, mais on ne pourra jamais être aussi libre qu'un développement « à la main » ou avec des frameworks type Zend Framework ou Cake PHP.

Il existe de nombreux CMS, avec chacun ses spécificités, type, avantages et inconvénients, adaptés aux projets, aux clients et aux développeurs à différents niveaux de compétences. Par exemple :

### **Blog**

- Wordpress
- DotClear

### **Site**

- Joomla!
- Drupal
- Spip (à bannir)

### **Forum**

- phpbb /fluxbb

Etc. Il ne existe énormément, des payants comme des gratuits.

**Une liste non exhaustive** : <http://www.scriptol.fr/cms/liste.php>

J'ai décidé de monter avec vous un portfolio avec **Wordpress** pour plusieurs raisons :

- Pouvoir différencier une partie « blog (spécifique à Wordpress) et une partie « portfolio » pour présenter vos projets à la sortie de la licence
- Profiter du coup du référencement possible de votre contenu facilement grâce à Wordpress, ce qui peut être intéressant.
- Ne pas avoir à faire de développement compliqué car ce n'est pas votre spécialité
- Avoir une interface d'administration toute prête
- Pouvoir ajouter un système de galerie à l'aide de plugins.
- Pour personnaliser votre front office grâce à des templates existants ou (mieux), votre propre template

## Let me introduce you: Wordpress!

**WordPress** est un *ystème de gestion de contenu* libre écrit en PHP et reposant sur une base de données *MySQL*. WordPress est surtout utilisé comme *moteur de blog*, mais ses fonctionnalités lui permettent également de gérer n'importe quel site Web, et c'est là l'avantage !

### Pourquoi utiliser Wordpress ?

La majorité des marketeurs, développeurs et graphistes sont d'accords sur ce point : le premier avantage de WordPress est de fournir un outil simple d'utilisation tout en étant performant.

En effet, l'interface d'utilisation de WordPress s'apparente beaucoup à celle d'un traitement de texte classique (un peu comme Word), du coup elle est très facile à prendre en main.

De plus, **la communauté** très forte autour de Wordpress est un avantage déterminant pour lui, et pour un CMS en général : C'est elle qui fait sa renommée. C'est la communauté qui développe des plugins, des templates ... Et donc enrichie Wordpress tous les jours.

Entre autres, pourquoi utiliser Wordpress ? Car :

- WordPress est un logiciel libre
- WordPress est simple d'utilisation
- Wordpress a une communauté forte : Plugins, thèmes, widgets, du gratuit au payant.
- WordPress est optimisé pour les moteurs de recherches
- etc.

### Le principe de Wordpress

Wordpress est donc surtout utilisé comme un **blog**. Cependant, on peut très bien l'utiliser pour faire un site web de contenu. Il repose sur 2 principes :

- Les posts : Basiquement, cela correspond aux articles. Un titre, du contenu...
- Les pages : Typiquement, on peut retrouver des pages comme « Contact », « à propos »... Une page est personnalisable, et peut bien sûr contenir des posts.

La dernière version de Wordpress est la version 3. Elle était attendue et apporte un certain nombre de changements.

On peut noter :

### > *D'un type "blog" vers un type "site"*

« Just another Wordpress Blog » disparaît ... au profit de « Just another Wordpress site ». Il semblerait que WordPress soit en train de migrer doucement du moteur de blog vers le CMS complet.

### > *Custom Post Type*

Il est maintenant possible de définir des « types de posts » avec des propriétés différentes. Par exemple, on peut créer un type de post « Film » qui sera relié à un réalisateur, des acteurs, etc. **Ou encore un type de post « projet » qui sera relié à des clients, type de projet, technologies utilisées ...**

Difficilement automatisable, il faut mettre les mains dans le cambouis et coder quelques lignes pour arriver à créer des types de posts. (C'est ce que nous ferons tout à l'heure)

**Tutorial** : <http://wpengineer.com/impressions-of-custom-post-type/> (en anglais)

### > *Gestion des menus*

Il est possible, directement depuis l'admin, de configurer les menus qui apparaissent dans la sidebar. On peut leur choisir un titre et y ajouter des éléments :

- lien vers une page du Blog
- lien vers une catégorie
- « custom link » avec ancre et URL personnalisée

Cette fonctionnalité est un bel effort puisque dorénavant, quand on aura besoin d'un menu un peu personnalisé, on n'aura plus à modifier les fichiers PHP du thème.

Donc en gros, pouvoir générer automatiquement notre menu !

**Tutorial** : <http://www.fran6art.com/tutoriels/la-cr...rdpress-3/> (en français)

### > *Multi-blog: le Network*

WordPress MU a été intégré dans WordPress. Il devient donc possible de créer très simplement plusieurs blogs avec la même installation. Cependant, on retrouve vite les limites de WordPress MU, qui ne gère pas le multi-domaine.

Vous ne pourrez donc avoir que des plates-formes du type « blog1.domaine.com » ou « domaine.com/blog1 ».

## > ***TwentyTen, le nouveau thème par défaut***

Le thème par défaut de WordPress change du tout au tout. Ici : <http://2010dev.wordpress.com/>

## > ***Mise à jour de tous les éléments inhérents à WordPress***

JQuery, Json, Prototype, Scriptaculous et SWFObject sont intégrés.

## > ***Une vraie page "404"***

WordPress renvoi désormais un réel entête de page d'erreur 404 lors que la page demandée n'existe pas. Et ça, c'est bien pour le référencement naturel.

## > ***Quelques blogs faits en Wordpress***

Sans doute des blogs/sites que vous connaissez, mais en autre et parmi des milliers :

- Smashing Magasine (UX design, Graphisme ...) <http://www.smashingmagazine.com/>
- Mashable (Social Media, Business ...) <http://mashable.com/>
- AE2I <http://ae2i-iutarles.fr>
- Nicoptère (Développement AS3 spécialisé bitmap/3D) <http://en.nicoptere.net/>
- GrgrDvrt (développeur AS3 spécialisé en bitmap) <http://grgrdvrt.com/blog/>
- Photoshop Lady (Tutos Graphisme, Motion...) <http://www.photoshoplady.com/>
- Spoon Graphics (veille, tuto graphisme ...) <http://www.blog.spoongraphics.co.uk/>
- Pas qu'un peu (veille graphique, photo...) <http://blog.pasquunpeu.fr/>
- ... et tellement d'autres

## 2 - Lancement de Wordpress

### Logiciels indispensables

Avant de commencer le projet, faisons un point sur quelques outils indispensables au bon développement d'un projet web. Il en existe énormément, j'en ai retenu quelques-uns que j'utilise au quotidien, qui ont l'avantage d'être **gratuit**.

### Serveur Apache, MYSQL ...

> **WAMP** (MAMP, LAMP...) : En téléchargeant et en installant WAMP, un serveur apache ainsi qu'une base MYSQL est installé, pour pouvoir développer en local

> **Easy PHP** : idem que WAMP, mais moins utilisé

### Editeur de texte

> **Notepad ++** : Libre et gratuit, il a l'avantage d'être léger et simple d'utilisation. Couleur syntaxique, plugins, cela peut être un très bon outil. Mais il est souvent utilisé pour faire quelques petites retouches « vite fait », lorsque l'on veut simplement modifier un fichier et ne pas ouvrir l'artillerie lourde.

> **NetBeans** : Libre et gratuit aussi, c'est un de mes favoris. Il n'est pas trop lourd, et existe pour différents langages. Nous prendrons celui optimisé pour le PHP.

Rendez-vous sur <http://netbeans.org/downloads/index.html>

La version optimisée PHP offre plusieurs fonctionnalités clefs :

- Visualiser l'arborescence d'un projet
- Auto-complétion sur le PHP, HTML, CSS, jQuery
- CTRL + Clic
- support du SVN, FTP : Pouvoir travailler directement sur un serveur
- etc.

Liste des raccourcis clavier :

[http://netbeans.org/project\\_downloads/usersguide/shortcuts.pdf](http://netbeans.org/project_downloads/usersguide/shortcuts.pdf)

## Browsers

> **Firefox** : indispensable au développement pour ses nombreux plugins.

> **Autres navigateurs** : pour tester la compatibilité entre navigateur. Personnellement, j'ai une préférence pour naviguer avec Chrome et développer avec Firefox.

## Plugins FF

> **Firebug** : Sans doute le plugin le plus connu. Il permet entre autres :

- De visualiser n'importe quel élément du DOM en un seul clic
- De visualiser les échanges entre le client et le serveur (GET, POST ...)
- De changer le CSS à la volée
- D'écrire et exécuter du JS à la volée
- Etc.

> **HTML Tidy** : Un simple validateur HTML qui informe des erreurs. Ne prends pas encore en charge l'HTML 5

> Web developer bar : Offre de nombreuses fonctionnalités comme :

- Visualiser une page web sans CSS, sans JS ...
- Outils de mesures
- Etc.

> Vide cache : Surtout utile lorsque l'on développe en Flash, mais peut servir en un seul clic pour supprimer vider le cache

## Installation de Wordpress

C'est parti, on allume WAMP ! Rendez-vous ensuite sur <http://wordpress.org/download/> et télécharger la dernière version de Wordpress.

### Mettre tout en place

#### *Arborescence de son www*

Tout d'abord, nous allons mettre en place le projet. Pour cela, créer un dossier « Wordpress » dans le répertoire « www » de votre machine. Vous pouvez aussi adopter une architecture de fichiers pour propre, comme par exemple avoir un dossier « project », un dossier « test ».

Une bonne pratique :

- Project
  - Client
    - Projet
- Test
- Library

De ce fait, chaque Client peut avoir plusieurs projets, on a un espace de test, et un dossier qui contient nos librairies, layouts ...

Ceci fait, décompresser l'ensemble du fichier Wordpress téléchargé à l'intérieur du dossier « Wordpress » qui vient d'être créé.

#### *Mise en place de la BDD*

**BDD** = Base De Données (ou DB = Data Base en anglais)

Wordpress étant un CMS, reposant donc sur le principe d'un site dynamique, il faut mettre en place la base de données. Inutile de s'embarrasser à savoir comment Wordpress gère sa base, ce n'est pas le plus important. Il faut simplement créer une base dans laquelle Wordpress puisse créer les tables dont il aura besoin, et si possible un accès spécialement dédié à cela, c'est-à-dire un login/mdp pour l'accès à la BDD qui soit dédié à Wordpress (et qui n'est pas tous les droits d'un root par exemple). Ceci dit, c'est

une étape pas indispensable, mais normalement indispensable en production (cela n'est pas toujours le cas).

L'outil le plus commun, c'est **phpmyadmin**.

Rendez-vous donc sur <http://127.0.0.1/phpmyadmin>

Le couple Login/mdp par défaut est :

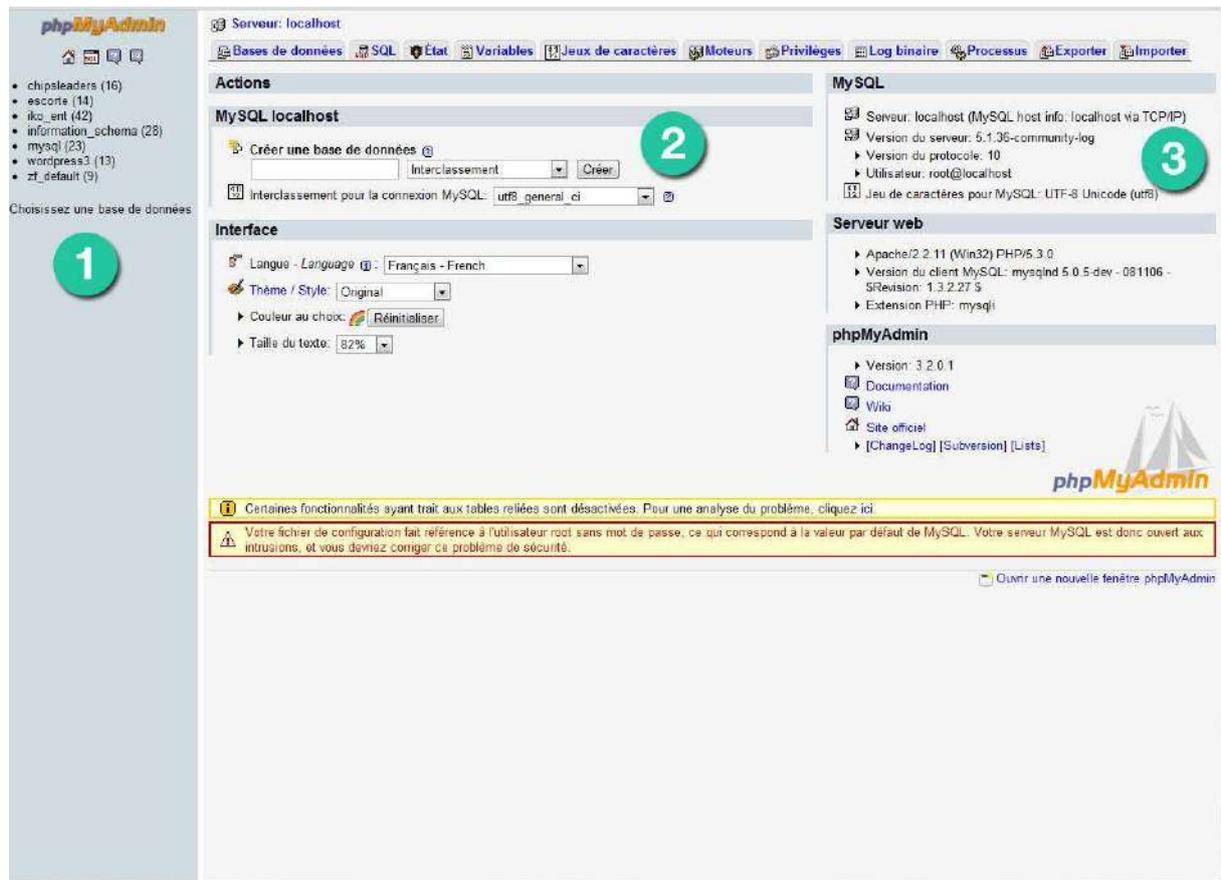
Login : root

Mot de passe : (vide)

Il y a donc 2 possibilités : garder cet accès « root », ou créer son propre utilisateur. Pour des raisons de facilité nous n'allons pas créer d'utilisateurs.

**Aller plus loin** : Créer un utilisateur MYSQL avec phpmyadmin :

[http://www.phpcs.com/tutoriaux/CREER-UTILISATEUR-MYSQL-AVEC-DROITS-SPECIFIQUES-SUR-BASE\\_275.aspx](http://www.phpcs.com/tutoriaux/CREER-UTILISATEUR-MYSQL-AVEC-DROITS-SPECIFIQUES-SUR-BASE_275.aspx)



## 1 – Liste des BDD

De base, on a une base « mysql » et « test », mais on retrouve ici l'ensemble des bases créés.

## 2 – Créer une nouvelle base de données

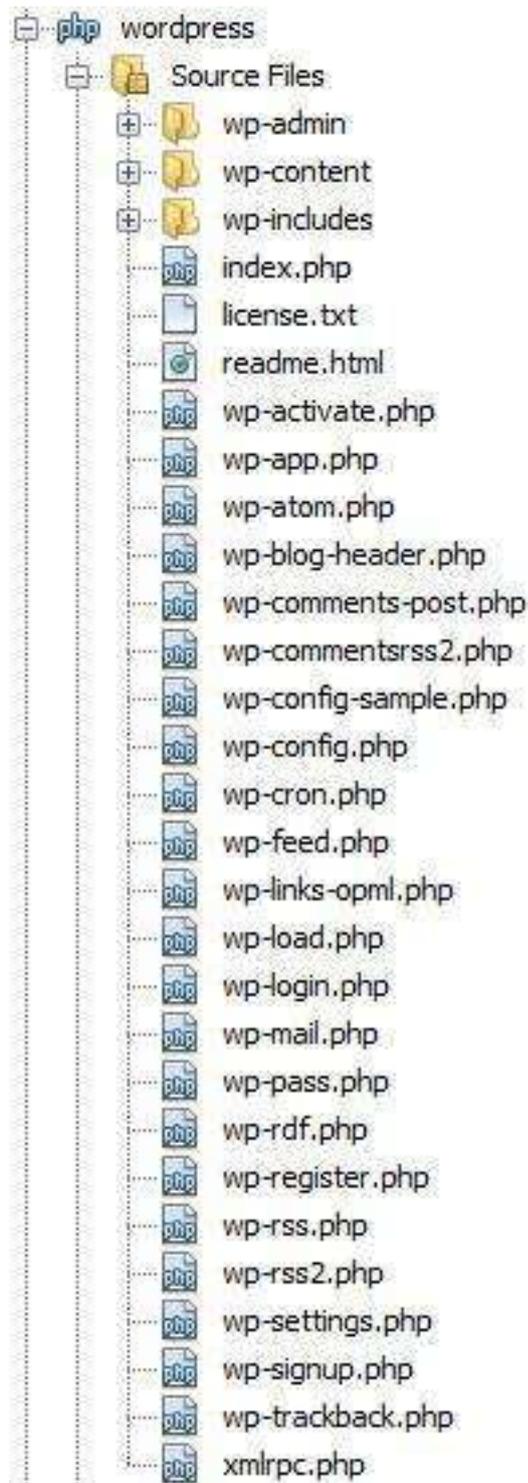
C'est ici que nous allons nous attarder. Il suffit simplement de rentrer un nom à notre base. Nous privilégierons toujours les interclassements utf8\_general\_ci, par souci d'avoir une constante : l'utf-8 (BDD, encodage de fichier, encodage HTML ...). Idem, privilégier les « \_ », pas d'espace, pas de caractères spéciaux (éèà ...)

Appelons-là : wordpress

## 3 – Infos

Ici la liste des infos sur le serveur, votre connexion, encodage de caractère ...

## L'arborescence



WordPress est organisé d'une certaine façon. Voici quelques trucs à savoir.

## *wp-admin*

C'est le dossier qui contient l'administration du site. Ainsi, nous pouvons accéder à l'interface d'administration grâce à <http://local.wordpress/wp-admin> (si vous utilisez un virtual host, sinon l'adresse devrait ressembler à quelque chose comme <http://127.0.0.1/project/wordpress>). Il ne faut rien toucher dans ce dossier

## *wp-content*

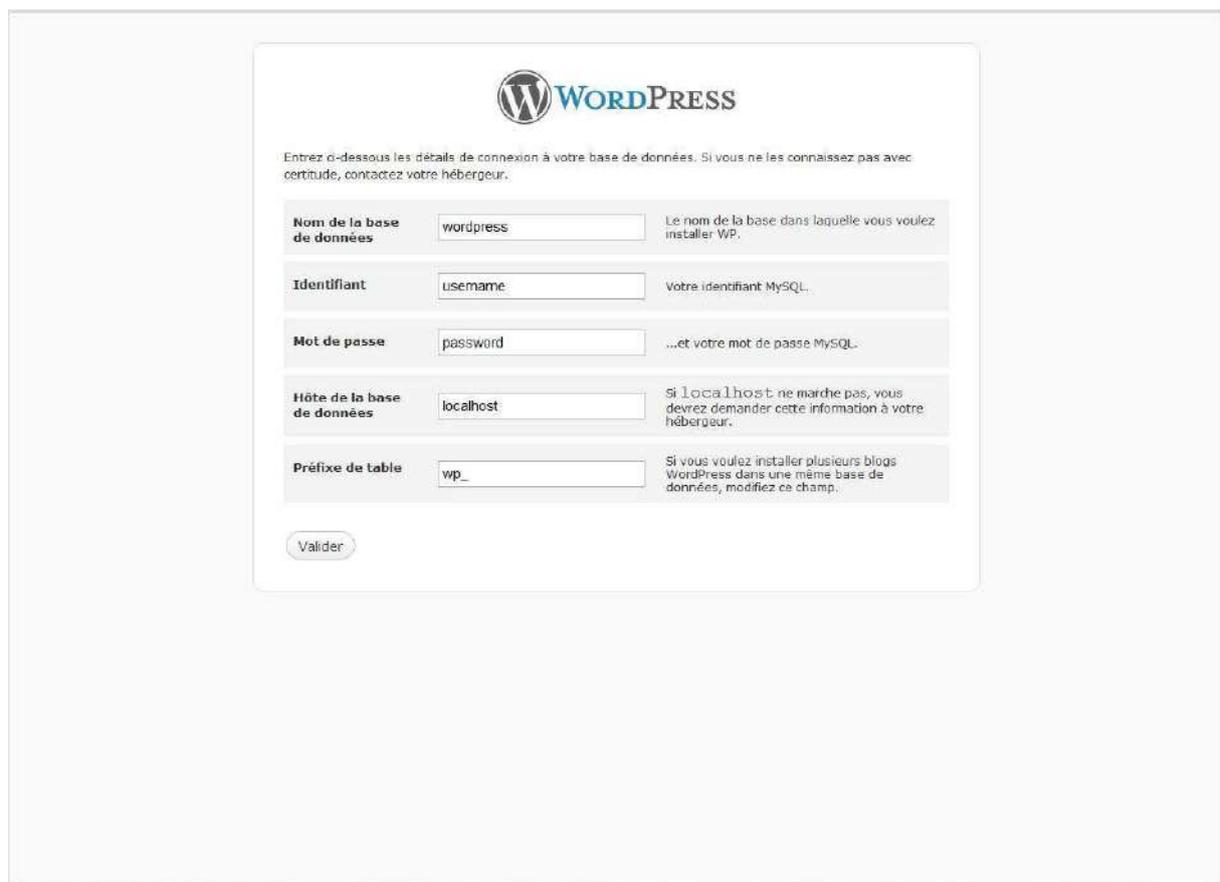
Ce dossier contient nos plugins et nos thèmes.

## *wp-includes*

Dossier contenant des fichiers propres au moteur de Wordpress. C'est ici que nous modifierons quelques fonctions.

Si vous avez créé un virtual host (voir Annexes), rendez-vous directement sur <http://local.wordpress/> (ou sinon, l'adresse devrait ressembler à quelque chose comme <http://127.0.0.1/project/wordpress>)

Là, Wordpress nous invite à entrer les infos concernant le blog, la base de données et le login/mdp de l'administration.



The screenshot shows the WordPress installation database configuration screen. At the top is the WordPress logo. Below it is a heading: "Entrez ci-dessous les détails de connexion à votre base de données. Si vous ne les connaissez pas avec certitude, contactez votre hébergeur." There are five input fields with labels and descriptions:

Nom de la base de données	<input type="text" value="wordpress"/>	Le nom de la base dans laquelle vous voulez installer WP.
Identifiant	<input type="text" value="username"/>	Votre identifiant MySQL.
Mot de passe	<input type="text" value="password"/>	...et votre mot de passe MySQL.
Hôte de la base de données	<input type="text" value="localhost"/>	Si localhost ne marche pas, vous devrez demander cette information à votre hébergeur.
Préfixe de table	<input type="text" value="wp_"/>	Si vous voulez installer plusieurs blogs WordPress dans une même base de données, modifiez ce champ.

At the bottom left is a "Valider" button.

**Nom de la base de données** : Correspond au nom que nous avons donné à la base lors de sa création. Ici, « wordpress »

**Identifiant** : Si vous n'avez pas créé de User spécifique, l'identifiant principal est « root »

**Mot de passe** : aucun (vide)

**Hôte de la base de données** : En local comme dans beaucoup de cas, ce sera le terme « localhost ». Ceci dit, il arrive que dans certain cas, cela correspond à d'autres valeurs (par exemple chez OVH, cela peut correspondre à quelque chose comme « mysql-5.7 », mais cela est indiqué dans vos identifiants)

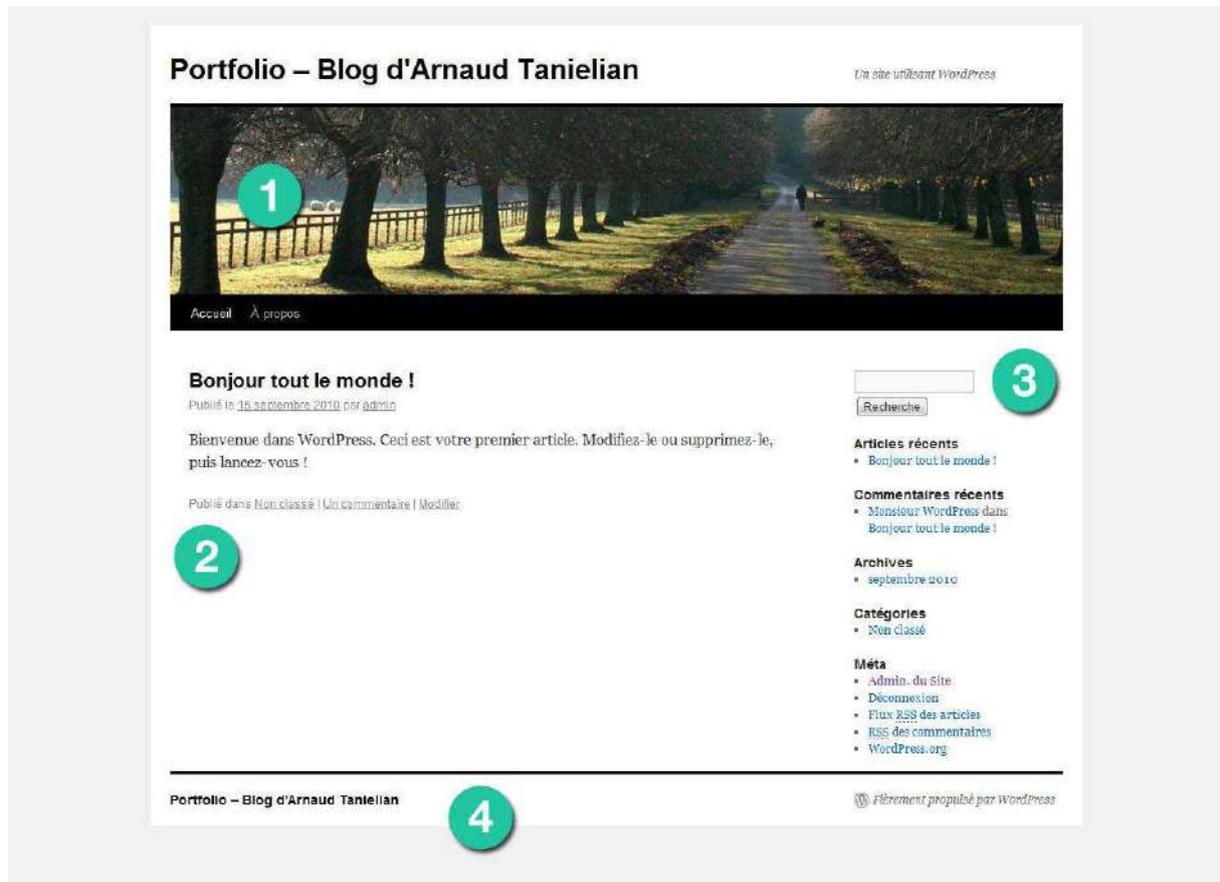
Une fois remplie, il ne reste plus qu'à fournir son propre couple login/mot de passe pour accéder à l'administration, et c'est prêt !

### *Les fichiers principaux d'un thème*

Regardons de plus près le contenu du dossier Twenty ten, le thème par défaut du nouveau Wordpress. Voici quelques fichiers à retenir :

- 404.php : page affichée lors d'une erreur 404. Enfin une page personnalisable !
- archive.php : Fichier servant à générer la page des archives
- comments.php : Fichier servant à générer les commentaires
- header.php : Fichier générant le header. Appeler lors d'un get\_header()
- footer.php : Fichier générant le footer. Appeler lors d'un get\_footer()
- index.php : Notre page d'index ! Contient l'affichage de la page d'accueil, avec notamment une loop qui affiche les derniers articles (voir plus loin pour loop)
- loop.php : Nouveau avec Wordpress 3. Ce fichier sert à générer l'ensemble des post, avec une pagination entre page.
- page.php : Sert à générer une page
- single.php : Sert à générer un article
- sidebar.php : Sert à générer la colonne sur le côté (droit ou gauche)
- style.css : Fichier CSS principal

## Front Office



Cette page correspond à l'index du site.

Basiquement, l'interface est découpée :

### **1 - Header**

Comprenant en général le titre du blog, une image et le menu

### **2 - Content (Zone de contenu)**

Sur l'index, on retrouve en général une liste d'une dizaine d'article, du moins de « preview » d'article. L'idée ici est de simplement affichée une preview (définie en back office) pour inviter le lecteur à cliquer et lire l'article en entier.

### **3 - Sidebar**

Contient plusieurs liens et fonctionnalités, dont un champ de recherche, des archives, ou encore des liens externes.

### **4 - Footer**

On peut y trouver ici aussi une liste de lien, ou encore un twitterWall ou des coordonnées.

## Back Office

The screenshot shows the WordPress dashboard with a sidebar on the left containing menu items: Tableau de bord, Articles, Médias, Liens, Pages, Commentaires, Apparence, Extensions, Utilisateurs, Outils, and Réglages. The main content area is titled 'Tableau de bord' and includes several widgets: 'Aujourd'hui' (Today) showing 1 article, 1 page, 1 category, 0 keywords, 1 comment, 1 approved, 0 pending, and 0 spam; 'Commentaires récents' (Recent comments) showing a comment from 'Monsieur WordPress'; 'Liens entrants' (Inbound links); 'Extensions' (Plugins) listing 'ourSTATSWidget' and 'Tag Search'; and 'Mises à jour récentes' (Recent updates). On the right side, there is a 'Press-Minute' form for creating a new post, a 'Brouillons récents' (Recent drafts) section, a 'Blog WordPress' section with recent posts, and 'Autres actualités de WordPress (en français)' (Other WordPress news in French).

- Tableau de bord : Espace regroupant une vue d'ensemble de son blog. Statistiques, derniers commentaires, backlink...
- Articles : Permet de gérer les articles
- Médias : Permet de gérer les médias (vidéos, images, PDF ...)
- Pages : Permet de gérer les pages.
- Commentaires : Permet de gérer les commentaires. Par défaut, un commentaire n'est pas publié en ligne tant que l'administrateur ne l'a pas validé.
- Apparence : Permet de gérer les différents thèmes.
- Extensions : Permet de gérer les plugins.
- Utilisateurs : Permet de définir les utilisateurs du blog (membres comme administrateurs)
- Outils : Différent réglages, comme l'import/export
- Réglages : Titre du blog, type de lien...

## L'installation d'un thème

Un thème, c'est vraiment ce qui va caractériser l'habillage de votre site. Il en existe des milliers, du gratuit au payant, du mal codé au valide W3C... Wordpress propose une gestion très simplifiée de vos thèmes, vous permettant d'en avoir plusieurs et de switcher simplement de l'un à l'autre, sans changer le fond mais simplement la forme.

Il est cependant conseillé de prendre un thème (ou son propre thème) et de se l'approprier (ajouts de widgets, customisation CSS ...) afin d'avoir un blog à l'identité propre et remarquable sur le net (pas perdre l'utilisateur à chaque venu s'il voit un blog différent)

Tout d'abord, on en choisit un (gratuit)

**Liste de site de thèmes :** <http://www.makeuseof.com/tag/10-resources-free-professional-wordpress-themes/>

**Liste 2010 de smashing magazine :**

<http://www.smashingmagazine.com/2010/08/19/100-free-high-quality-wordpress-themes-for-2010/>

J'ai choisi ce thème : <http://empirethemes.com/cumulus-a-portfolio-theme/> pour le projet. Une fois le .zip téléchargé, il suffit simplement de le déposer dans le dossier wp-content/themes.

Ensuite, rendez-vous dans l'administration dans la rubrique « Apparence ».

Portfolio – Blog d'Arnaud Tan... Installer des thèmes | Salutations, admin | Se déconnecter

[Tableau de bord](#) [Gérer les thèmes](#) [Installer des thèmes](#) [Aide](#)

**Thème actuel**

**Twenty Ten 1.1 par the WordPress team**

The 2010 theme for WordPress is stylish, customizable, simple, and readable — make it yours with a custom menu, header image, and background. Twenty Ten supports six widgetized areas (two in the sidebar, four in the footer) and featured images (thumbnails for gallery posts and custom header images for posts and pages). It includes stylesheets for print and the admin Visual Editor, special styles for posts in the « Asides » and « Gallery » categories, and has an optional one-column page template that removes the sidebar.

Tous les fichiers de ce thème se trouvent dans `/themes/twentyten`.

Mots-clés : black, blue, white, two-columns, fixed-width, custom-header, custom-background, threaded-comments, sticky-post, translation-ready, microformats, rtl-language-support, editor-style

**Thèmes disponibles**



**Cumulus 1.2 par Empire Themes**

*A light, fluffy portfolio theme.*

[Activer](#) | [Prévisualiser](#) | [Supprimer](#)

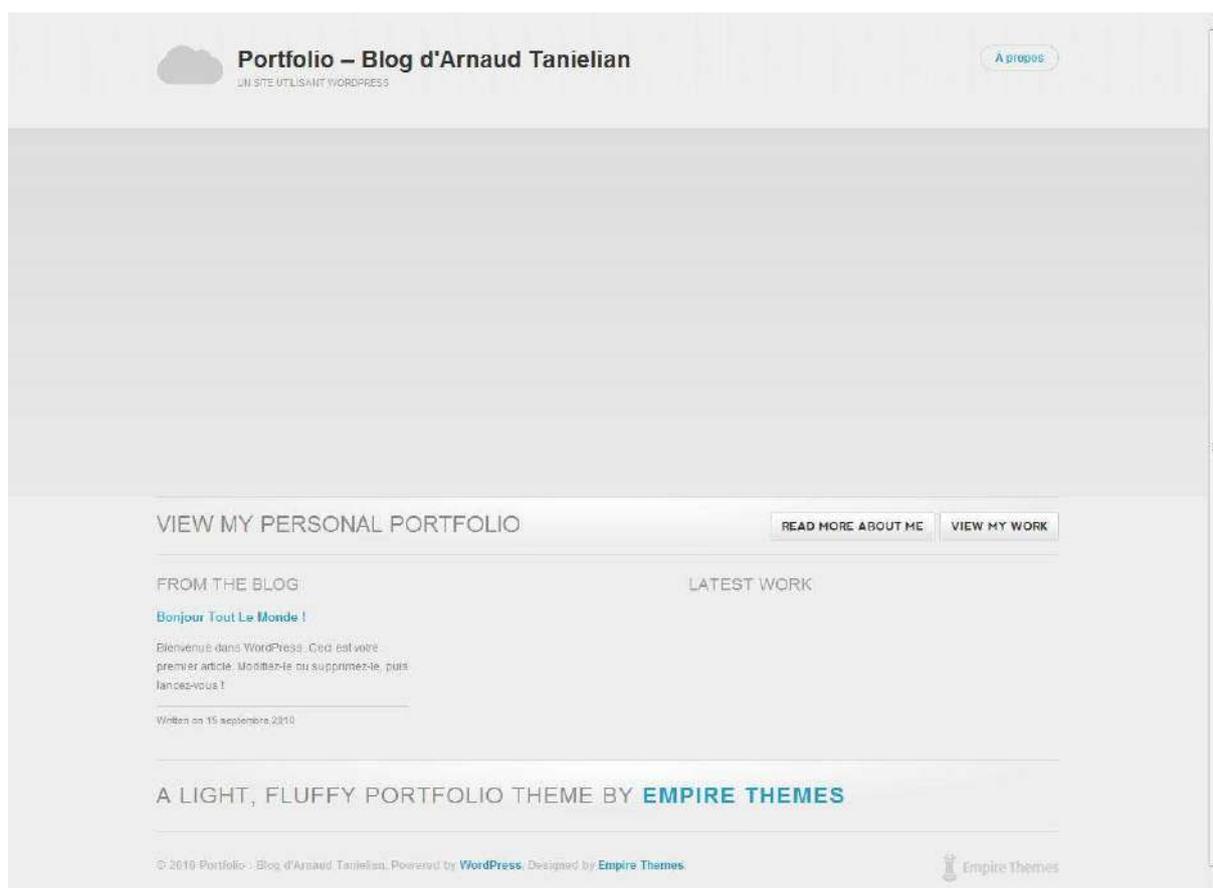
Tous les fichiers de ce thème se trouvent dans `/themes/cumulus`.

Mots-clés : two-column, widget-ready, portfolio

Merci de faire de WordPress votre outil de création. | [Documentation](#) | [Forums d'entraide](#) Version 3.0.4

On peut voir que le thème Cumulus 1.2 est disponible dans la rubrique « thèmes disponibles ».

Il ne reste plus qu'à cliquer dessus, puis « activer ». En se rendant sur le front, on peut constater le changement :



YeY, on avance !

## Installation d'un plugin

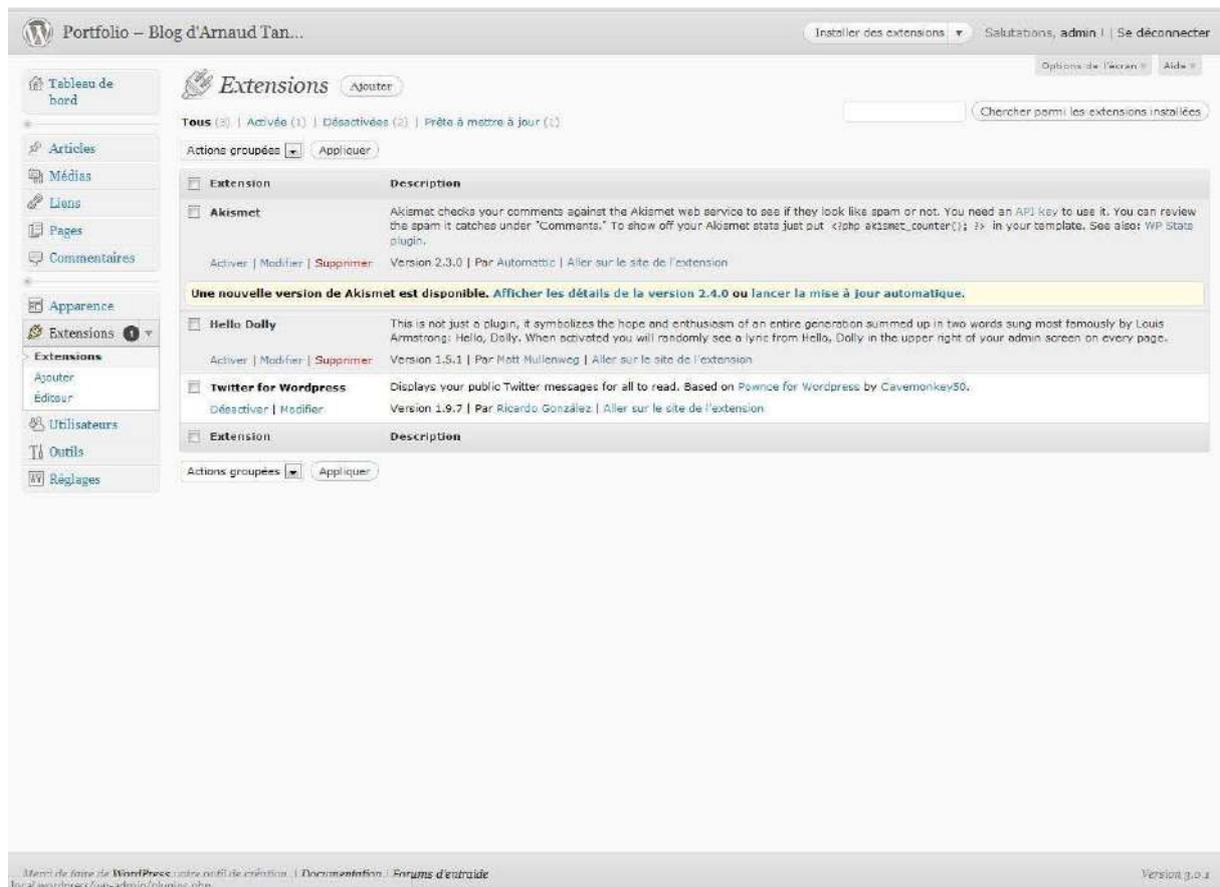
**Liste de plugins :** <http://wordpress.org/extend/plugins/>

Qu'est-ce qu'un plugin ? Un plugin est une extension de votre blog. Il a pour but de rajouter des fonctionnalités multiples et diverses, dans plusieurs domaines : Nouvelles fonctionnalités à vos articles (galeries d'images, partage réseaux sociaux ...), SEO (google analytics) ...

J'ai choisi le plugin **Share This** <http://wordpress.org/extend/plugins/share-this/>, qui permet de partager facilement sur les réseaux sociaux le lien vers son post ou projet. L'intérêt de ce plugin c'est qu'il adapte les dernières bonnes techniques (à mon sens) du partage : Nombre de tweets et de likes, puis un espace « share » pour accéder à tous les autres.

**Installation :** Il suffit de simplement télécharger, puis, déposer le dossier décompressé dans le répertoire « plugins »

Pour la suite, rendez-vous dans la rubrique « extensions » de l'administration. Cette rubrique regroupe l'ensemble des plugins ou widgets installés et permet de les gérer. L'avantage, c'est qu'on est alerté en cas de mise à jour d'un plugin !



A noter : Une fois le plugin installé, la configuration du plugin se trouve le plus souvent dans la rubrique « Réglages »

## Installation d'un widget

Liste de widgets : [http://codex.wordpress.org/WordPress\\_Widgets](http://codex.wordpress.org/WordPress_Widgets)

Il y a beaucoup de widgets disponibles. On peut retrouver entre autres le célèbre Tag Cloud (nuage de tag), RSS, Search ...

Un widget, c'est presque comme un plugin mais pas tout à fait. **C'est avant tout un plugin « d'habillage » de votre blog.**

C'est en général (et par défaut) utilisé dans la sidebar, la fameuse colonne à droite ou à gauche de l'écran. Mais ça peut tout aussi bien être utilisé dans le footer ou autre.

De plus, avec un widget, aucun besoin d'aller trifouiller le code !

Attention ceci dit, car pour que votre widget marche, votre thème doit être « widget ready », c'est-à-dire qu'il puisse accueillir et afficher votre widget. Pour cela, regarder sur la page d'information de l'auteur du thème.

**Aller plus loin :** Comment rendre mon thème « widget ready » ?

<http://automatic.com/code/widgets/themes/>

**Aller plus loin** : Comment rendre « widgetisable » un autre élément du DOM (footer par exemple) ?

<http://www.themelab.com/2009/04/25/add-a-widgetized-footer-to-your-wordpress-theme/>

**Code en bref**

### *Fonctions.php*

```
(register_sidebar(array('name'=>'Footer',  
  
    'before_widget' => '<div class="footer-left">',  
  
    'after_widget' => '</div>',  
  
    'before_title' => '<h3>',  
  
    'after_title' => '</h3>',  
  
));
```

### *Footer.php*

```
<?php if ( !function_exists('dynamic_sidebar') || !dynamic_sidebar("Footer") ) : ?>  
  
<!-- Afficher contenu à la place du widget -->  
  
<?php endif; ?>
```

J'ai choisi ce widget : <http://wordpress.org/extend/plugins/twitter-for-wordpress/> pour permettre l'affichage des derniers tweets d'un compte Twitter.

Après avoir téléchargé le pack, placer le dossier dézippé dans le répertoire « plugins » du dossier wp-content.

Ensuite, rendez-vous dans l'administration, rubrique « extension » pour activer l'extension.

Maintenant, c'est là la différence avec un plugin : Rendez-vous dans la rubrique apparence > widget.

Vous trouverez ici la liste des widgets, et des parties du site dans lesquels vous pourrez « glisser » tel ou tel widget. Selon le thème installé, les « parties du sites » sont widgetisable ou pas. Par exemple, le thème de base est pratiquement entièrement widgetisable : sidebar, footer ...

The screenshot shows the WordPress administration interface for a site titled "Portfolio – Blog d'Arnaud Tan...". The user is logged in as "salutations, admin" and can access "Nouvel article", "Options de l'écran", and "Aide".

The main content area is titled "Widgets" and is divided into two sections:

- Widgets disponibles:** A list of 16 widgets that can be added to the sidebar. Each widget has a title and a brief description:
  - Archives:** Une archive mensuelle des articles de votre site.
  - Articles récents:** Les articles les plus récents de votre site.
  - Calendrier:** Un calendrier des articles de votre site.
  - Catégories:** Une liste ou un menu déroulant des catégories.
  - Commentaires récents:** Les commentaires les plus récents.
  - Flux:** Articles en provenance de n'importe quel flux RSS ou ATOM.
  - Liens:** Vos liens préférés.
  - Méta:** (Dé)connexion, administration, flux et autres liens internes à WordPress.
  - Menu personnalisé:** Utilisez ce widget pour ajouter l'un de vos menus en tant que widget.
  - Nuage de mots-clés:** Vos mots-clés les plus utilisés, au format « nuage ».
  - Pages:** Les pages statiques de votre site WordPress.
  - Recherche:** Un formulaire de recherche pour votre site.
  - Recherche:** Recherche.
  - Texte:** Texte ou code HTML arbitraire.
- Widgets désactivés:** A list of widgets that have been removed from the sidebar but whose settings are preserved. Currently, it shows "Pages" and "Calendrier".

On the right side of the interface, there is a sidebar titled "Colonne latérale 1" which currently contains a "Twitter #1" widget.

The left sidebar contains navigation links: "Tableau de bord", "Articles", "Médias", "Liens", "Pages", "Commentaires", "Apparence" (with sub-links for Themes, Widgets, Menus, Theme Options, Éditeur), "Extensions", "Utilisateurs", "Outils", and "Réglages".

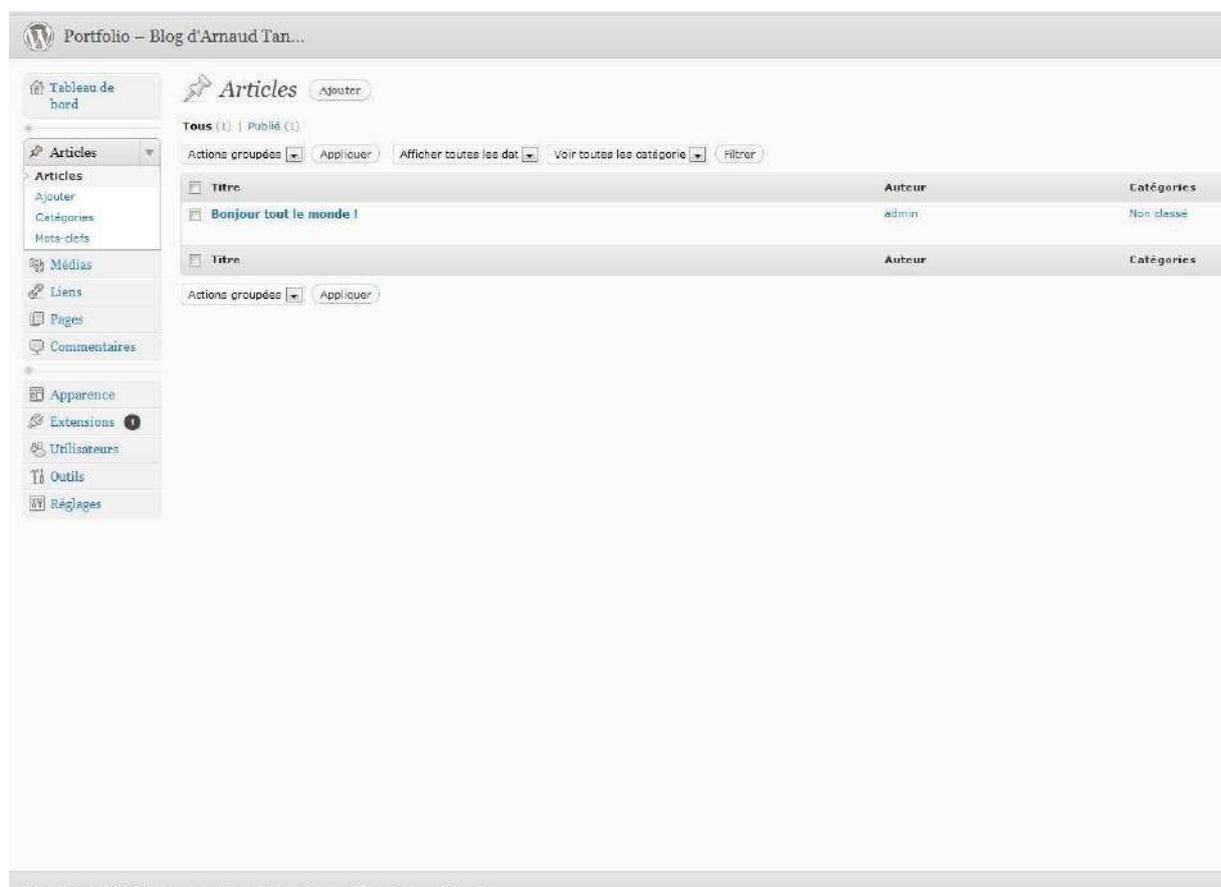
### 3 - Insérer du texte

#### Ecrire son premier article : découverte de l'éditeur visuel

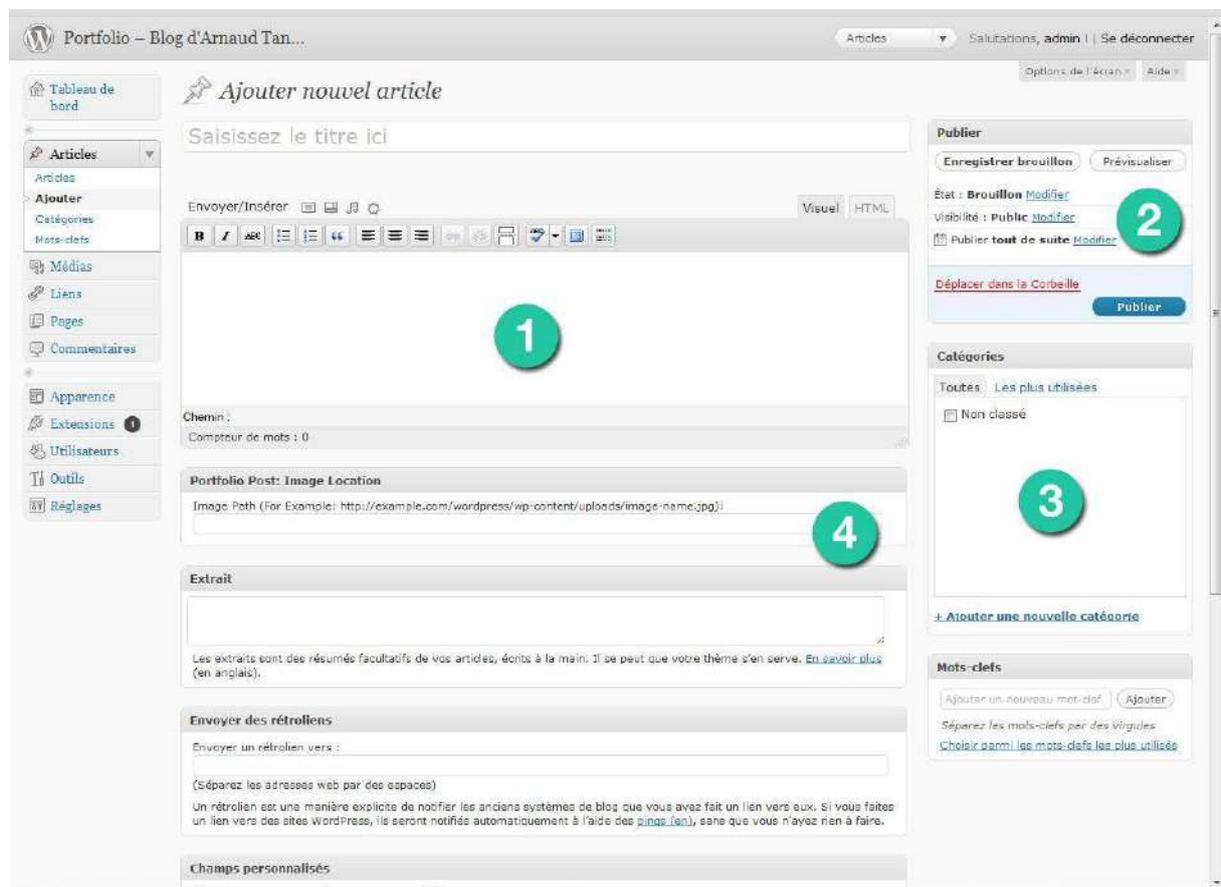
Un article, c'est le principal élément d'un blog. Que cela soit un blog sur le graphisme, développement ou même sur votre vie, c'est ce qui fait vivre votre blog, lui donne de l'intérêt, fait revenir les visiteurs sur votre site.

Un bon article est un article unique. Ne vous contentez pas de copier/coller des articles trouvés ici et là (si vous le faites, ne pas oublier de citer la source). Ecrivez du contenu pertinent, « unique », en bon français, intéressant. Vous pouvez aussi choisir une ligne directrice précise (la « geek écolo » : <http://www.thegreengeekette.fr/> , ou encore la critique décalée et osée : <http://www.megaconnard.com/> ...) A vous de trouver votre identité sur le net. On peut tout aussi bien tenir un blog classique et normal, juste attention de ne pas tenir une copie de copie de copie d'un vrai blog ;)

Bref, allons jeter un œil dans l'administration, dans la rubrique article. D'une manière classique, nous retrouvons l'ensemble des articles, avec la possibilité d'en ajouter, modifier, supprimer ...



Clignons sur le bouton ajouter. Voici l'interface d'ajout de nouvel article :



C'est une interface claire, simple, mais découpé en plusieurs sections.

## Section 1 : Editeur de texte enrichi (ou WYSIWYG)

**WYSIWYG**: What You See Is What You Get.

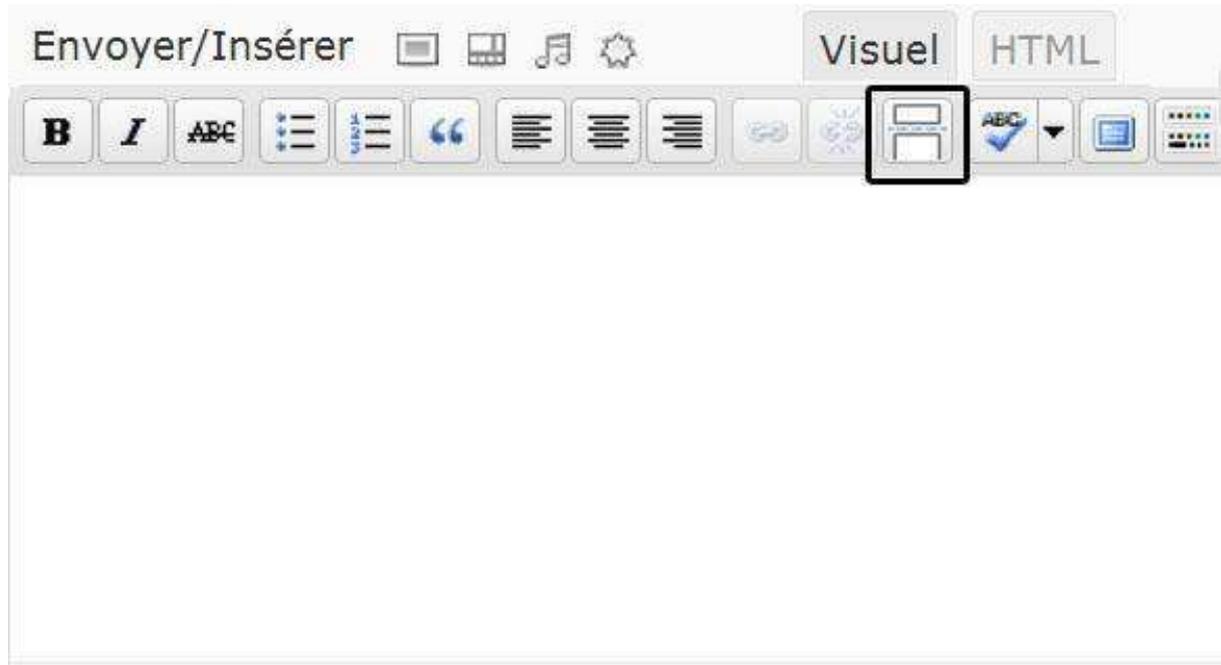
Un éditeur de texte enrichi est un éditeur proposant des fonctionnalités pour enrichir un texte, de façon à formater un texte brut en sortie HTML. Le but est d'avoir un contenu présentable sur le net, valide W3C (cela dépend des éditeurs), référencable, accessible ...)

**A noter** : Il existe énormément d'éditeur de texte, qui peut se trouver sous forme de plugins JS ou PHP.

Un WYSIWYG vous offre beaucoup de fonctionnalités : Paragraphes, caractères gras, souligné, couleurs, insertion d'images ... Un peu comme si on se retrouvait face à une page Word.

Le WYSIWYG offre l'avantage de présenter le contenu selon 2 versions : Visuel et HTML. Visuel pour avoir un rendu purement visuel, et HTML pour avoir le rendu sous forme de balises HTML. On peut, à tout moment de la rédaction d'un article, switcher entre les 2 rendus pour équilibrer ce que l'on voit avec ce que l'on va générer. (Par exemple, pour rajouter des balises *alt* aux images, ou virer les *<br/>* rajouté lors des sauts de lignes.)

Zoom sur un bouton très utile : La balise « More »



Cette balise sert à définir à quel endroit l'article sera tronqué en page d'accueil puis affiché à la suite « Lire la suite »

Le reste des fonctionnalités est assez classique, comparé à un éditeur de texte comme Word. L'important étant de bien formater son texte (utiliser l'icône « liste » pour créer des listes, qui va générer des `<ul><li>` côté HTML ...), cela en va de votre crédibilité, et pour un web plus propre et plus accessible/référencable.

## Section 2 : Publier

C'est ici que ce trouve les différentes informations sur l'état d'un post : **Brouillon** (il s'enregistre automatiquement), **non-publié**, **publié**. On peut aussi voir sa visibilité (rendre un post public ou privé), c'est-à-dire restreindre les droits de lecture d'un post (en d'autres termes, il faut être loggé sur le site, et avoir les droits nécessaires pour voir le post. Cela reste un cas assez exceptionnel, car un blog a normalement vocation à être public, mais on ne sait jamais, surtout avec la vulgarisation de Wordpress en tant que site, et non blog).

C'est tout bête, mais il ne faut pas oublier de publier son article pour qu'il puisse être affiché en front

## Section 3 : Catégorie/Mots-clefs

C'est l'endroit où l'on peut associer un post à une catégorie. On peut par exemple imaginer des catégories comme « Développement », « Graphisme », « UX Design » ... Si

l'on veut, on peut n'associer qu'une seule catégorie/article, histoire de pouvoir faire par exemple une page qui ne parle que de graphisme, et une autre que de développement...

A ne pas confondre avec les tags (ou « mots-clefs »), juste en dessous. L'idée ici est d'associer autant de tags que l'on veut par post.

#### **Section 4 : Portfolio Post, extrait, rétroliens ...**

Ce sont des fonctionnalités un peu transverses, pas forcément utilisées à chaque fois, mais qui sont des compléments à votre article.

## 4 - Modifier le contenu

### Modifier les pages et articles

Il est temps d'apprendre à manipuler un peu le blog pour pouvoir le personnaliser selon nos besoins.

Nous allons découper le site en 2 parties : une partie « blog » et une partie « portfolio ».

**Pour plus de facilité, les exemples suivants seront traités avec le thème de base.**

#### Création des pages

Le site aura 4 pages :

- **Accueil** : La page d'accueil du site, elle offre un aperçu rapide des derniers articles ou projets
- **Blog** : Contient tous les articles
- **Portfolio** : Contient tous les projets
- **A propos** : Indique le propos du site, qui le tient, voir un formulaire de contact.

Modifiez la page « à propos » qui existe déjà, et créez les 2 autres.

#### Nouveauté Wordpress 3 : Le menu

La nouveauté de cette version, c'est la création de **menus personnalisés**.

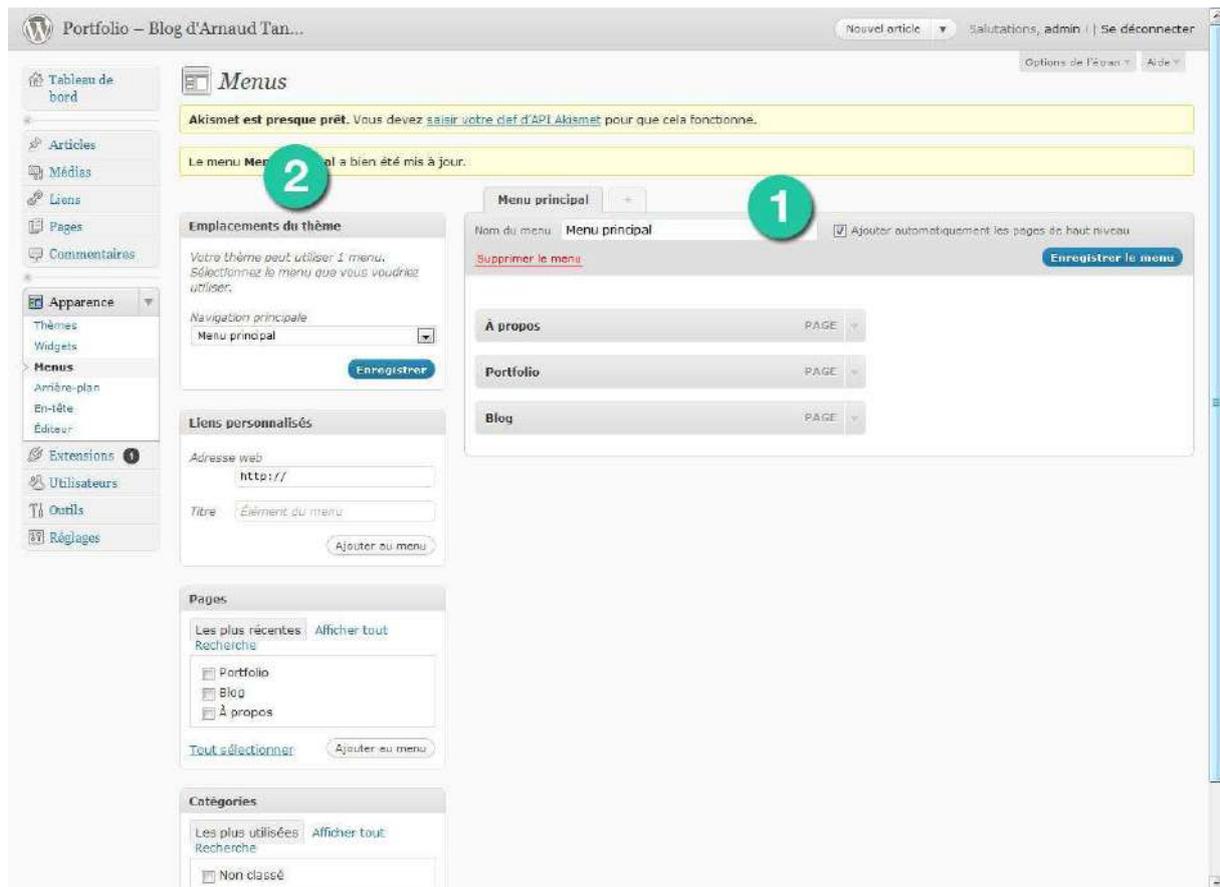
Le but ? Pouvoir créer un menu customisé en générant automatiquement un menu en fonction de ses pages, articles, catégories, ou n'importe quel autre lien. Comme les widgets, il faut que le thème soit « menu ready ». Le thème de basant étant compatible, pas de problème.

**Aller plus loin** : rendre son thème « menu ready » :

<http://www.fran6art.com/tutoriels/la-creation-de-menus-sous-wordpress-3/> ,  
<http://wpchannel.com/creer-menus-personnalises-wordpress-3-0/>

A noter que le menu peut être utilisé en tant que widget (donc rendre un menu directement dans une partie du site, par exemple, le footer)

Pour se faire, rendez-vous dans Apparence > Menus



## 1 : Créer le menu

Cliquez sur le bouton **Créer menu**. Il faut lui donner un nom, car on peut créer plusieurs menus.

## 2 : Personnalisez le menu

Dans la partie gauche, vous noterez la présence de 3 modules :

- **Liens externes** : très pratique pour insérer des liens vers d'autres sites Web ;
- **Pages** : pour ajouter des pages WordPress dans votre menu ;
- **Catégories** : pour insérer des catégories WordPress dans votre barre de navigation ;

Il suffit donc de cocher les éléments de votre choix – Les pages « Blog », « Portfolio » et « A propos », ainsi que le lien « Accueil », puis de cliquer sur **Ajouter au menu**.

A noter que l'on peut cliquer sur « Ajouter automatiquement les pages de haut niveau ».

Pour réarranger l'ordre du menu, glissez / déposez les différents éléments tout en sachant qu'il est possible de créer des sous-menus. Pour ce faire, décalez l'élément vers la droite pour créer une imbrication.

N'oubliez pas de cliquer sur **Sauvegarder le menu** pour enregistrer vos modifications.

Constatez le résultat en front. Ajustez en conséquence votre menu selon vos besoins.

Vous pouvez ajouter un second menu en cliquant sur le bouton en forme de plus, situé juste à côté du menu nouvellement créé sous l'administration WordPress.

## Mise en place des pages

Maintenant que notre menu est en place, nous voulons personnaliser nos pages.

Mais l'éditeur de Wordpress n'offre pas (hormis extensions) la possibilité de coder directement.

Nous allons donc créer puis construire les pages pour qu'elles puissent afficher le contenu adapté.

### Comment faire ?

Expliquons tout d'abord la démarche de Wordpress pour afficher une page.

Wordpress cherche plusieurs fichiers de modèle dans votre thème actif. Le premier trouvé sera utilisé pour afficher la page demandée. Plus bas vous trouverez l'ordre dans lequel WordPress effectue sa recherche :

- 1 > La page sélectionnée "Modèle Page". C'est-à-dire, le fichier « blog.php » pour la page blog
- 2 > page.php : le fichier de base qui génère la page automatiquement depuis ce que l'on a mis dans l'éditeur
- 3 > index.php : autrement dit, l'accueil.

## Créer vos propres « Modèles Pages »

Les fichiers définissant chacun des « Modèles Page » se situent dans le dossier de votre thème.

Pour créer un nouveau Modèle pour une Page vous devez créer un fichier. Pour cela et pour plus de facilité, copiez/collez le fichier « page.php », et renommez-le « blog.php ». De ce fait, nous partons du code de base d'une page. De plus, ce fichier va « override » le fichier page.php, c'est-à-dire passer devant lui en termes d'exécution de code.

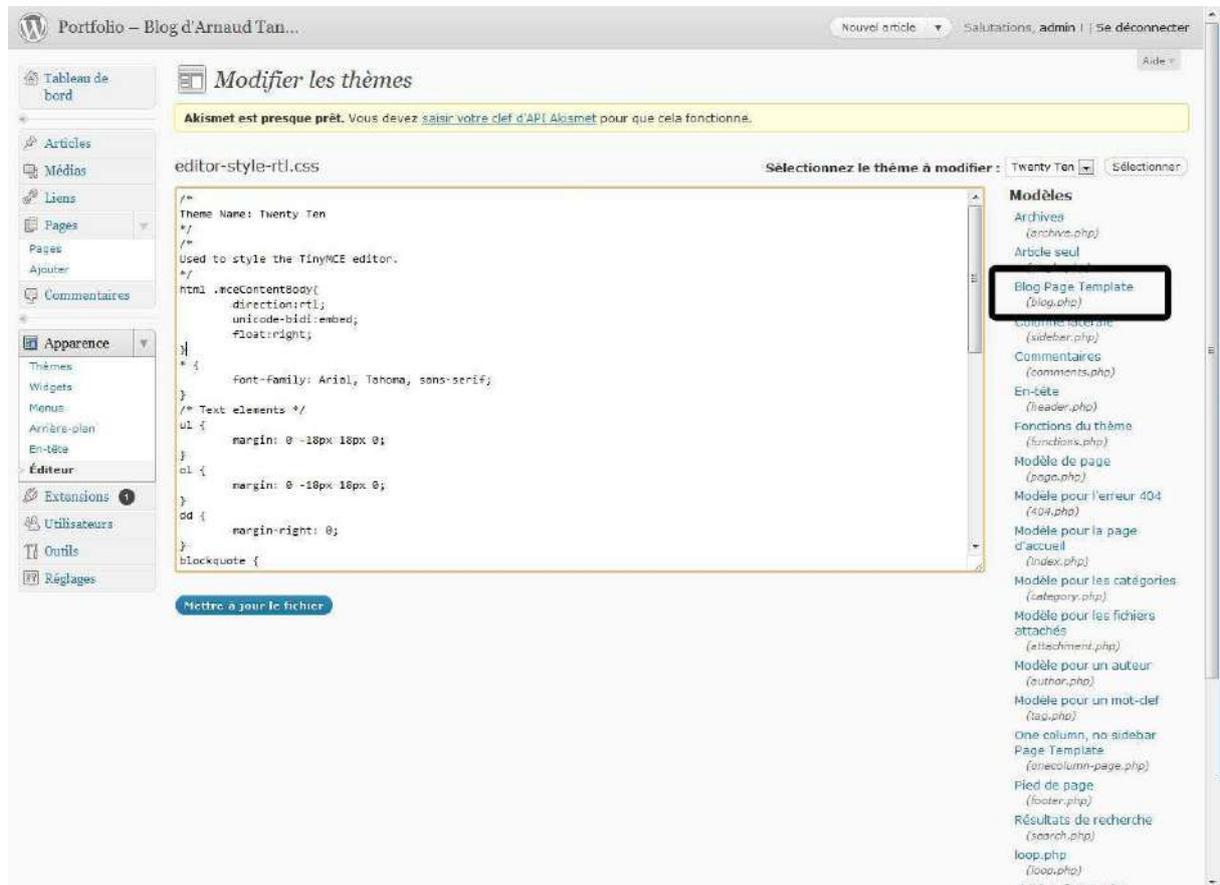
Pour cela, il faut tout d'abord éditer le fichier blog.php et donner un nom de Template :

```
<?php
/*
Template Name: Blog
*/
?>
```

Cette étape est indispensable pour que Wordpress puisse reconnaître notre page. On garde le reste du code pour le moment.

Maintenant, rendez-vous dans Apparence > Editeur.

On peut constater que le fichier « blog .php » est présent, sous le nom de « Blog page Template »



Maintenant, retournons dans la gestion des pages, et éditons la page « blog ».

Dans la section « Attributs de la page », on peut maintenant trouver notre modèle « Blog » dans la sélection de Modèle. On lie donc la page au modèle « blog ».

Et voilà, on n'a plus qu'à travailler le fichier blog.php pour personnaliser exclusivement la page blog !

Faisons le même travail pour la page « portfolio ». Avant de nous jeter dans la configuration de ces 2 pages, voyons d'abord le principe des « Custom Post Types »

## Nouvelle fonction de WordPress 3 : "Custom Post Types"

C'est la possibilité de créer ses propres types d'articles. En effet, de bases, un article est considéré comme tel. Grâce au « custom post types », on peut maintenant configurer ses propres « types » de posts. Ici, l'intérêt est de garder le post de base, mais de créer un nouveau type de post, le post de type « **projet** ».

Ce type de post peut par exemple regrouper une description, quelques infos sur le client, des screenshots/vidéos du projet, des tags ...

### Une fonction pour customiser l'admin

Nous allons devoir taper un peu de code, hélas. Mais nous allons faire cela pas à pas !

Tout le code suivant va être regroupé dans une fonction qui sera appelée à l'initialisation de l'administration, dans le fichier **functions.php**, en fin de fichier.

De ce fait, lorsque l'on lance l'admin, on activera toutes les fonctions décrites dans la fonction `my_custom_init()` :

```
add_action('init', 'my_custom_init');

function my_custom_init()
{
    /* notre code PHP pour rajouter les custom post type */
}
```

### Déclarer le custom post type

Maintenant, il faut rajouter le type « projet » dans l'admin :

```
register_post_type('projet', array(

    'label' => ('Projets'),
    'singular label' => ('Projet'),
    'public' => true,
    'show ui' => true,
    'capability type' => 'post',
    'hierarchical' => false,
    'supports' => array('title', 'author', 'thumbnail', 'editor')

));
```

Il faut utiliser la fonction « `register_post_type` » qui porte bien son nom : elle enregistre un nouvel élément, un nouveau type de Post. Il y a ensuite plusieurs paramètres. Ils n'y sont pas tous parce que beaucoup sont optionnels.

Les plus importants :

- « projet » correspond au nom du nouvel élément. Il sera utilisé dans les fichiers de template.
- « label » est le nom qu'on retrouvera dans la colonne de gauche de l'admin. Il est préférable de mettre un pluriel.
- « singular\_label » est l'équivalent au singulier
- « public » définit si les posts seront publics ou non.
- « show\_ui » définit si ce nouvel élément apparaît dans l'admin
- « capability\_type » définit si le nouvel élément est identique à un Post classique ou à une Page classique. Ici on veut un titre et une image miniature donc Post. En général, les Posts sont beaucoup plus intéressants car ils ont davantage de fonctions que les Pages.
- « hierarchical » définit s'il peut y avoir une hiérarchie (comme pour les Pages). Ici, c'est non car chaque élément est au même niveau.
- « supports » est un array (tableau) qui définit quels éléments d'un Post normal on veut avoir. Ici je veux juste le titre, l'auteur du Post et une miniature. Il y a pleins d'autres éléments :
  - Title
  - editor (la fenêtre principale de contenu)
  - author
  - thumbnail
  - excerpt
  - trackbacks
  - custom-fields'
  - comments
  - revisions (la sauvegarde automatique de versions passées)
  - page-attributes (choix de template pour les pages et gestion de la hiérarchie)

Maintenant, rendez-vous dans l'administration. Un nouvel élément est apparu !



Et oui, on peut désormais ajouter, gérer, supprimer, modifier des « projets » !

**Aller plus loin** : Toutes les options :

<http://justintadlock.com/archives/2010/04/29/custom-post-types-in-wordpress>

**De retour sur nos pages**

Pour rappel, nous avons maintenant 2 nouveaux fichiers: **blog.php** et **portfolio.php**

Nous voulons que la page blog n'affiche que les posts, et la page portfolio n'affiche que les projets.

Prenons l'exemple du portfolio. Le contenu du fichier est assez simple :

```
<?php

/**
Template Name: Portfolio
*/

get_header();

$home_paged = (get_query_var('paged'));
$args = array(
    'post type' => 'projet',
    'post status' => 'publish',
    'paged' => $home_paged
);
```

```

query_posts($arguments);

?>

        <div id="container">

            <div id="content" role="main">

                <?php get_template_part( 'loop');?>

            </div><!-- #content -->

        </div><!-- #container -->

<?php get_sidebar(); ?>

<?php get_footer(); ?>

```

### Explication du code :

```
get_header();
```

Permet d'afficher le header

```
$home_paged = (get_query_var('paged'));
```

Le `$home_paged` est une astuce qui permet de récupérer la valeur de la page courante et ainsi éviter de casser la pagination!

```

$arguments = array(
    'post type' => 'projet',
    'post status' => 'publish',
    'paged' => $home_paged
);

query_posts($arguments);

```

Ensuite, on prépare un tableau qui va servir de paramètre à la fonction `query_post()` ; On spécifie le « `post_type` », ici `projet`, le statut des posts, ici « `publish` » (on ne prend que les posts publiés), et on indique la page courante, ici `$home_paged`.

### La fonction `query_post()`

[http://codex.wordpress.org/Function\\_Reference/query\\_posts](http://codex.wordpress.org/Function_Reference/query_posts)

Une des fonctions les plus importantes de Wordpress. Cette fonction sert à définir ce qui doit être affiché dans « la loop » (voir plus loin). On peut passer en paramètre un tableau comme une chaîne de caractère.

Par exemple, pour n'afficher que les posts dont la catégorie est égale à 1 :

```
query_post('cat=1');
```

Ensuite, on n'a plus qu'à appeler « la loop » à l'intérieur de notre <div> container.

```
<div id="content" role="main">

    <?php get_template_part( 'loop');?>

</div><!-- #content -->
```

### La fonction `get_template_part()`

[http://codex.wordpress.org/Function\\_Reference/get\\_template\\_part](http://codex.wordpress.org/Function_Reference/get_template_part)

Elle permet d'afficher la loop, en fonction de notre `query_post`. A noter que cette fonction est nouvelle depuis la version 3, elle permet d'inclure des fichiers php.

Le fichier a proprement parlé est le fichier `loop.php`.

**The Loop:** [http://codex.wordpress.org/The\\_Loop](http://codex.wordpress.org/The_Loop)

**Aller plus loin :** Afficher les thumbnails directement dans la loop :

<http://nenuno.co.uk/creative/wordpress/adding-post-thumbnails-to-wordpress-3-0/>

N'hésitez pas à créer vos propres fichiers de loop !

Il suffit maintenant de reproduire la même chose pour la page blog pour afficher les simple posts.

## Nouvelle fonction de WordPress 3 : “Custom Taxonomy”

[http://codex.wordpress.org/Custom\\_Taxonomies](http://codex.wordpress.org/Custom_Taxonomies)

Derrière ce nom barbare, se cache tout simplement les catégories, liens et les tags. Avec WordPress 3, vous pouvez créer d’autres taxonomies. Imaginons qu’avec ce portfolio vous voulez ajouter 2 choses :

- le type de projet : un logo, un site vitrine, une animation flash....
- La (les) techno(s) utilisée(s)

On pourrait utiliser les Catégories et les Tags, mais elles sont déjà utilisées par les Posts pour le Blog. On va donc créer d’autres catégories et d’autres tags, exclusivement pour le Portfolio.

### Retour à la méthode `my_custom_init()`

On ré-ouvre le fichier `functions.php`, puis à la suite, dans la fonction `my_custom_init()`, on peut rajouter :

```
register_taxonomy( 'type', 'projet',
    array( 'hierarchical' => true,
          'label' => 'Type',
          'query_var' => true,
          'rewrite' => true ) );

register_taxonomy('techno', 'projet',
    array( 'hierarchical' => false,
          'label' => 'Techno',
          'query_var' => true,
          'rewrite' => true ) );
```

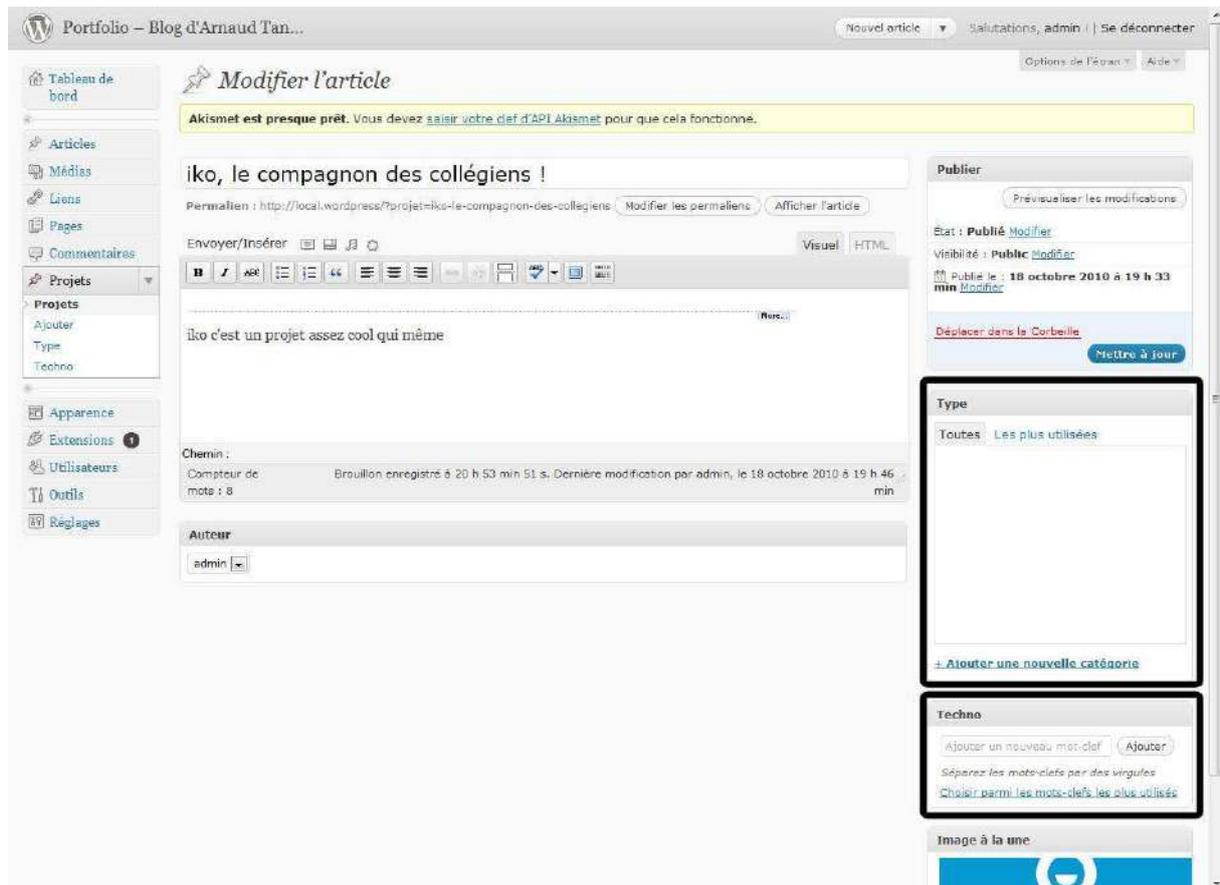
### Fonction `register_taxonomy()`

On utilise la fonction `register_taxonomy`, qui permet d’enregistrer une nouvelle taxonomie. Dans l’ordre on choisit :

- le nom de la taxonomie
- l’élément auquel il s’applique (ça peut être ‘post’, ‘page’ ou un post type qu’on a créé, en l’occurrence ‘projet’)
- ‘hierarchical’ définit s’il y a une relation parent/enfant. Si oui, c’est comme les Catégories. Si non, c’est comme les Tags.

- 'label' c'est le nom plus descriptif qui s'affichera dans l'admin
- 'query\_var' à mettre sur true pour qu'on puisse appeler cette taxonomie dans nos templates
- 'rewrite' c'est la chaîne de caractères présente dans les permaliens. Si on met "true", ça prend la valeur par défaut, c'est à dire le nom de la taxonomie.

On peut désormais gérer les types de projets comme des catégories, et les technos comme des tags !



## Afficher de beau projet : custom loop !

Et bien maintenant que l'on est bien avancé, pour afficher nos customs taxonomy, il va falloir finalement créer notre propre loop. Rien de bien compliqué !

Commencez par copier/coller le fichier loop.php, en le renommant loop-portfolio.php

Il suffit maintenant de se rendre aux alentours de la ligne 150. C'est ici que l'on peut customiser le rendu de nos tags et catégories.

```
<?php if ( count( get_the_category() ) ) : ?>

<span class="cat-links">
```

```

        <?php printf( __( '<span class="%1$s">Posted in</span> %2$s',
'twentyten' ), 'entry-utility-prep entry-utility-prep-cat-links',
get_the_category_list( ', ' ) ); ?>

</span>

<span class="meta-sep">|</span>

<?php endif; ?>

<?php

//Type de projet
echo get_the_term_list( $post->ID, 'type', '<p>Type de projet : ', ', ', ',
'</p>' );

?>

<?php

//Tags Technos
$tags_list = get_the_term_list( $post->ID, 'techno', '<p>Technos : ', ', ', ',
'</p>' );

if ( $tags_list ):?>

    <span class="tag-links">

        <?php printf( __( '<span class="%1$s"></span> %2$s', 'twentyten' ),
'entry-utility-prep entry-utility-prep-tag-links', $tags_list ); ?>

    </span>

    <span class="meta-sep">|</span>

<?php endif; ?>

```

Ici, on utilise la fonction `get_the_term_list()`

### **Fonction `get_the_term_list()`**

Cette fonction permet d'afficher les taxonomies du post courant (accessible grâce à `$post->ID`). On indique ensuite le nom de la taxonomie (type, techno ...), puis le « formatage » html (avant, après)

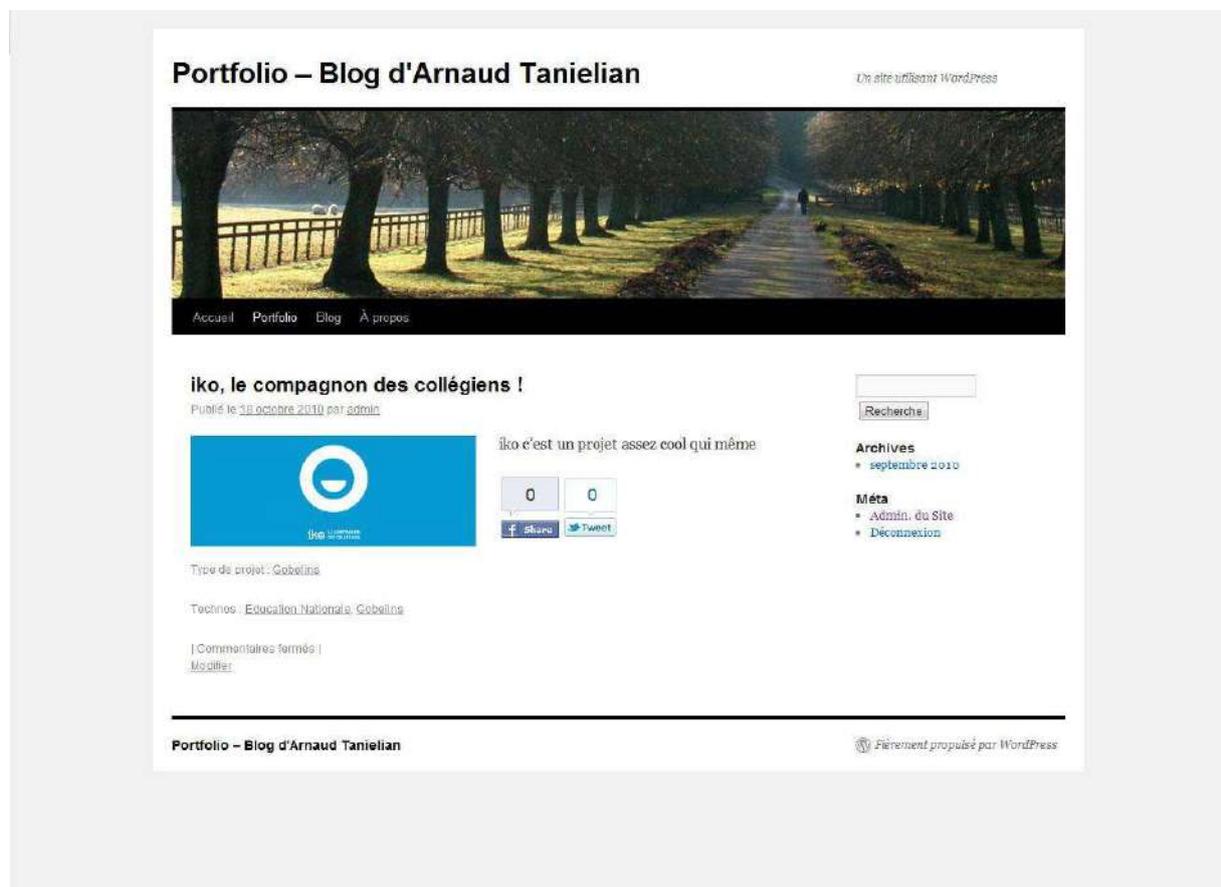
Il ne reste plus qu'à demander le rendu de cette loop custom dans le fichier `portfolio.php` !

```
<div id="content" role="main">

    <?php get_template_part( 'loop-portfolio');?>

</div><!-- #content -->
```

Maintenant, notre rubrique Portfolio :



The screenshot shows a WordPress portfolio page titled "Portfolio – Blog d'Arnaud Tanielian". The page features a large header image of a tree-lined path. Below the image is a navigation menu with links for "Accueil", "Portfolio", "Blog", and "À propos". The main content area displays a post titled "iko, le compagnon des collégiens !" published on October 18, 2010, by admin. The post includes a blue logo for "iko" and a description: "iko c'est un projet assez cool qui même". There are two social media share buttons (Facebook and Twitter) and two comment count boxes showing "0". The page also has a search bar, an "Archives" section with a link to "septembre 2010", and a "Méta" section with links for "Admin du Site" and "Déconnexion". The footer contains the site title "Portfolio – Blog d'Arnaud Tanielian" and the WordPress logo with the text "Fièrement propulsé par WordPress".

## 5 - Modifier la présentation

Il y a 2 manières de customiser un blog : en modifiant un thème... Ou en créant le sien.

### Modifier le thème par défaut : Twenty Ten

Maintenant, que l'on a vu comment modifier le fond, modifions la forme !

#### Les modifications de bases

Rendez-vous dans la rubrique Admin > Apparence.

Vous pouvez déjà régler quelques petites choses :

- En-tête : Image en header de votre site
- Arrière-plan : Background de votre site

Ce sont des modifications mineures, mais c'est un début.

#### L'éditeur de thème

Cliquez sur Apparence > Editeur.

D'ici, vous pouvez directement modifier un fichier .php, ou un fichier .css.

- Style.css : C'est LA feuille de style par défaut, celle qui contient la majorité des définitions de style de notre site.

Pour cela, n'hésitez pas à utiliser **firebug**. Il vous permet de modifier le CSS à la volée, puis de répercuter facilement les changements dans le fichier CSS.

## Comment intégrer son propre template ?

Héhé, là est tout l'enjeu. Souvent en agence, il est facile de partir de thème existant, mais beaucoup plus intéressant de monter son propre template.

### Monter la page HTML

Et oui, c'est LA première étape. La majorité du temps, une seule page est à monter (s'en suit quelques déclinaison, mais l'essentiel du code HTML est là)

### Les standards CSS

Wordpress recommande d'adopter des « best practices » pour le montage HTML, et notamment les nommages des identifiants ou class CSS.

CSS coding standards: [http://codex.wordpress.org/CSS\\_Coding\\_Standards](http://codex.wordpress.org/CSS_Coding_Standards)

Pour le montage HTML/CSS, ou même avant lors de la conception, je vous invite à suivre les nombreuses règles présentes ici :

[http://codex.wordpress.org/Site\\_Design\\_and\\_Layout](http://codex.wordpress.org/Site_Design_and_Layout)

Règles que j'espère pouvoir mettre en valeur dans un prochain cours ou tutorial ;)

### Partir du template de base

Pour plus de facilité, l'astuce consiste à copier/coller le thème twenty ten dans un autre dossier en le renommant par le nom de son propre thème. C'est une bonne base de travail.

Il ne reste plus qu'à remplacer le code HTML dans les fichiers (header, index, footer ...) petit à petit.

## 6 – Gérer son site en ligne

Maintenant que l'on a bien avancé son site en local, il est temps de le mettre en ligne !

### Choisir un hébergement + nom de domaine

De nombreuses offres d'hébergeurs existent sur internet : One&one, OVH, Gandhi ... A vous de choisir.

Un hébergement mutualisé, sur un an, coûte environ 20 euros par an (hébergement + nom de domaine). Il comprend l'essentiel : une base mysql, support de PHP, email perso, espace FTP.

### Exporter

Wordpress permet d'exporter tout votre contenu sous forme de fichier XML, importable par la suite.

Pour cela, rendez-vous dans Outils > Exporter.

### Mettre en production

Une fois votre hébergement et nom de domaine acheté, placez sur le FTP un dossier wordpress « neuf », c'est-à-dire dans l'état d'origine une fois téléchargé sur le site de Wordpress.

Car il y a une étape importante à ne pas oublier : la configuration ! En effet, il va falloir reconfigurer votre Wordpress par rapport à votre nouvel environnement, surtout par rapport à la base mysql.

Une fois les fichiers uploadés sur le serveur, visitez votre blog en ligne pour la 1<sup>ère</sup> fois afin de procéder aux réglages.

**A noter** : bien souvent, le serveur MYSQL n'est pas « localhost ». Pour OVH par exemple, cela ressemble plus à un truc du genre « MySQL 5-7 ».

Maintenant, vous avez le choix. Soit vous remplacez petit à petit les fichiers modifiés en local sur le wordpress « neuf » (template, plugins, images ...), soit vous conservez le fichier **wp-config** (qui contient votre configuration en prod) quelque part sur votre ordi, supprimez le dossier wordpress en prod, uploadé votre dossier wordpress du local, et remplacez le fichier wp-config du serveur (ouf !)

## Importer

Une fois tout cela fait, vous n'avez plus qu'à vous rendre dans la nouvelle administration en prod, puis Outils > Importer. Et le tour est joué !

## Quelques astuces

### De beaux liens pour un beau référencement

Une petite astuce pour avoir de belle URL, optimisées pour le référencement.

Rendez-vous dans Réglages > Permalien.

Choisir autre chose que la valeur par défaut, pour ma part, j'ai opté pour **Date et Titre**

**Aller plus loin :** <http://wordpress-tuto.fr/url-rewriting-1>

## 7 - Annexes

### Option : Création d'un Vhost

Il est toujours plus propre et agréable de travailler avec un vhost (virtual host).

#### Ce n'est pas une étape obligatoire

Un Vhost consiste à créer une « fausse » adresse web accessible à l'ordinateur lié à n'importe quelle adresse IP.

Ici, l'idée est de créer un vhost pour rattacher une adresse propre à notre projet.

Exemple : Au lieu d'avoir <http://127.0.0.1/projet/wordpress/wordpress3/>, on pourrait directement y accéder avec <http://local.wordpress/>

Sous Windows

Il y a 3 fichiers à modifier : httpd.conf, httpd-vhosts.conf, et host.

> *httpd.conf* (C:\wamp\bin\apache\Apache2.2.11\conf)

Il faut ici simplement autoriser le chargement du fichier vhost

```
# Virtual hosts
Include conf/extra/httpd-vhosts.conf
```

> *httpd-vhosts.conf* (C:\wamp\bin\apache\Apache2.2.11\conf\extra)

Ici, on va définir les « règles ». Il faut redéfinir le localhost, puis définir notre projet.

```
<VirtualHost *:80>
    DocumentRoot "F:/wamp/www/"
    ServerName localhost
    ServerAlias www.localhost.com
    ServerAlias www.localhost.fr
</VirtualHost>
```

```
<VirtualHost *:80>
```

```
DocumentRoot "F:\wamp\www\test\wordpress3\wordpress"
```

```
ServerName local.wordpress
```

```
</VirtualHost>
```

```
> host (C:\Windows\System32\drivers\etc)
```

Ce fichier sert à définir l'IP en relation avec le hostname (le nom de domaine)

```
127.0.0.1 localhost
```

```
127.0.0.1 local.wordpress
```

Sous Mac

C'est assez identique, seulement les fichiers ne se trouvent pas au même endroit.

On redémarre le serveur, et tadaaam ! On peut accéder à notre projet grâce à :

<http://local.wordpress/>

Ceci dit, il y a aussi une raison de cette façon de faire. Car un fichier accessible à

<http://127.0.0.1/projet/wordpress/wordpress3/> n'est pas à la racine (root). De ce fait, si je fais appel à une image, je dois passer absolument par la « base url ».

Exemple:

```
<img src=« /img/test.jpg » alt=« test » />
```

Ici, le premier « / » indique que le chemin commence à la racine. Donc avec un vhost, on est directement à la racine, donc pas de problème. Mais sans cela, nous serions obligés d'utiliser une base URL, de type :

```
<img src=«http://127.0.0.1/projet/wordpress/wordpress3/img/test.jpg » alt=« test » />
```

Un peu lourd dans son code :)

On peut passer par des chemins relatifs, mais c'est beaucoup plus hasardeux.

## **8 – Source**

*[http://fr.wikipedia.org/wiki/Syst%C3%A8me\\_de\\_gestion\\_de\\_contenu](http://fr.wikipedia.org/wiki/Syst%C3%A8me_de_gestion_de_contenu)*

*<http://fr.wikipedia.org/wiki/WordPress>*

*<http://marketingaffiliation.wordpress.com/2010/06/03/pourquoi-utiliser-un-blog-wordpress-pour-faire-de-laffiliation-sur-internet/>*

*<http://blog.elephorm.com/tutoriel-wordpress-3/>*

*<http://www.42ubik.com/blog/pourquoi-utiliser-wordpress>*

*<http://codex.wordpress.org/fr:Pages>*

*<http://www.seomix.fr/wordpress/utilisation/wordpress-3-ce-qui-va-changer/>*

*<http://bravenewworld.fr/776-les-nouveautes-de-wordpress-3.html>*