

## Module 5

# WordPress 4.9

Content Management System



## Table des matières

<b>WORDPRESS.....</b>	<b>4</b>
<b>Les gestionnaires de contenu.....</b>	<b>4</b>
C.M.S.....	4
Les solutions issues du libre.....	5
Sites vitrines et blogs.....	5
Sites de vente en ligne.....	6
<b>Environnement serveur.....</b>	<b>6</b>
PHP5/MySQL.....	6
Site statique.....	6
Site dynamique.....	6
Technologies.....	6
En local sous WAMP.....	7
Présentation du logiciel.....	7
Utilisation.....	8
<b>WordPress.....</b>	<b>9</b>
Wiki.....	9
Sites de références.....	9
Back-office.....	10
Thèmes.....	13
Extensions (plugins).....	13
Optimisation.....	14
<b>Création de thème.....</b>	<b>15</b>
Organisation des fichiers sources.....	15
Principaux fichiers du thème.....	16
Les fonctions PHP dans WordPress.....	17
Les inclusions.....	18
header.php.....	20
footer.php.....	22
sidebar.php (widgets).....	23
Les menus du site.....	23
La boucle (The loop).....	24
single.php.....	25

page.php.....	26
index.php.....	28
archives.php.....	29
404.php.....	29
search.php.....	30
WP_Query.....	31
Meta Boxes.....	32
Custom Post Types.....	33

# WORDPRESS

## Les gestionnaires de contenu

### C.M.S

Content Management System (système de gestion de contenu).

Outil permettant d'administrer simplement un site Web, à la façon d'un blog grâce à une interface d'administration du site (=back-office).

Également, plate-forme de développement contenant et organisant les fichiers sources du site : **.php**, **.html**, **.css...**

Chaque C.M.S (Drupal, SPIP, Prestashop, Joomla) à sa propre interface d'administration

### Back-office WordPress 4

The screenshot shows the WordPress 4 back-office interface. The top navigation bar includes the site name 'Le blog du Webmaster', a user profile 'Salutations, damien', and options for 'Options de l'écran' and 'Aide'. The left sidebar contains a menu with items like 'Tableau de bord', 'Articles', 'Médias', 'Pages', 'Commentaires', 'Contact', 'Apparence', 'Extensions', 'Utilisateurs', 'Outils', 'Réglages', 'Photo Gallery', and 'Réduire le menu'. The main content area is titled 'Articles' and shows a list of articles. The list has columns for 'Titre', 'Auteur', 'Catégories', 'Mots-clés', and 'Date'. Three articles are listed: 'Formation Webmaster - 31 place Web' (published 25/03/2014), 'Actualité 1' (published 24/03/2014), and 'Bonjour tout le monde !' (published 24/03/2014). The interface also includes search and filter options.

Si vous avez de solides connaissances en PHP, MySQL, HTML, CSS, JavaScript, vous pourrez développer votre propre solution de gestion de contenu. C'est ce que font certaines agences Web.

Cependant, de plus en plus d'agences et d'indépendants utilisent des logiciels issus du libre. Le code source étant ouvert, ce sont des milliers de développeurs qui les font évoluer en créant des plugins (fonctionnalités) et thèmes prêt à l'emploi.

## Les solutions issues du libre

Les plus utilisés en France 2017

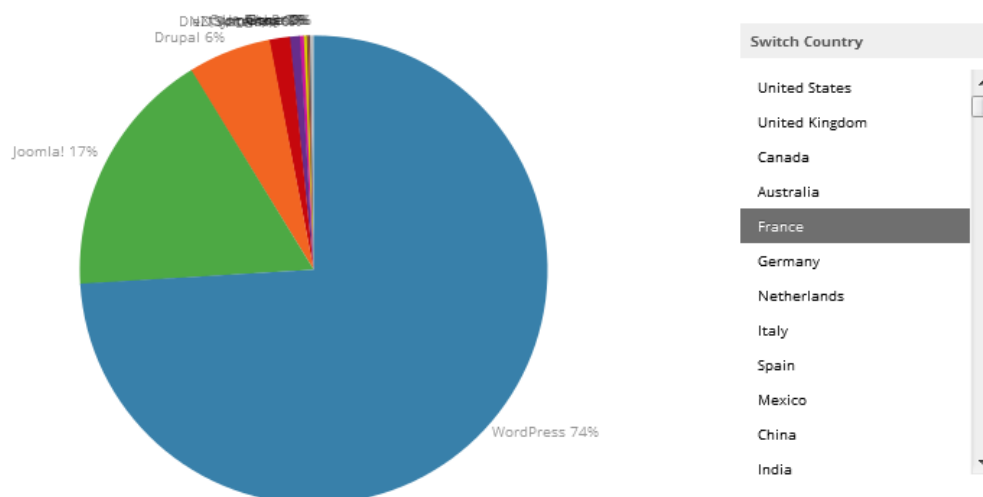
### Sites vitrines et blogs

- [Wordpress](#)
- [joomla](#)
- [Drupal](#)

### Part d'utilisation des C.M.S en France 2016

## ■ ■ Open Source usage in France

Statistics for websites using Open Source technologies in France which use the .FR Extension



Source de l'image : <http://trends.builtwith.com/cms/open-source/country/France>

## Sites de vente en ligne

- [Prestashop](#)
- [Magento](#)

# Environnement serveur

## PHP5/MySQL

### Site statique

Au début du Web, les sites étaient construits avec des pages dites "statiques".

C'est à dire que pour modifier votre site (textes, images, liens) vous devez modifier directement le code HTML/CSS de la page. [voir les modules HTML/CSS]

Également, pour créer une page, il faut recréer un fichier HTML, ce qui rend l'administration d'un tel site fastidieuse voire impossible pour les personnes n'ayant pas de compétences en HTML/CSS/JavaScript.

### Site dynamique

Il est aujourd'hui possible, grâce à une interface d'administration (back-office) de faire toutes les modifications sur le contenu : textes, images, messages de forum. Tout ceci étant accessible à une personne n'ayant aucune connaissance en HTML/CSS.

De nouvelles fonctionnalités sont alors possibles : forum, espace membre, formulaires etc... grâce aux fichiers PHP inclus dans le gestionnaire de contenu (C.M.S).

## Technologies

Pour réaliser un tel site, il faudra avoir recours à deux nouveaux langages : PHP (*Hypertext Preprocessor*) et MySQL (*Structured Query Language*).

### Exemple de script PHP

```
<?php
$id_agent=$_POST['id_agent'];
// récupération d'un identifiant envoyé par formulaire
?>
```

### Exemple de requête SQL

```
DELETE FROM db_competence_agent WHERE id_agent = $id_agent;
// Supprimer une ligne en fonction de l'identifiant
```

Pour héberger des sites dynamiques, la plupart des hébergeurs (OVH, 1and1, nFrance ...) ont un environnement **LAMP** :

**Linux** : système d'exploitation

**Apache** : serveur Web

**MySQL** : serveur de base de données

**PHP** : langage de script

### Le PHP s'insère dans le HTML, soit directement

```
<strong><?php echo($donnees['titre']); ?></strong> - <em> le <?php
echo($donnees['date_creation_fr']); ?></em>
```

### Soit par inclusion de script externe

```
<?php
include 'header.php';
?>
```

## En local sous WAMP

### Présentation du logiciel

+ [WampServer](#) est un logiciel gratuit à installer sur votre ordinateur personnel et qui va vous permettre de travailler en local dans les mêmes conditions que si votre site était hébergé sur un serveur distant. Le dossier contenant votre site sera placé dans **C:\wamp\www**. Vous pourrez ainsi placer plusieurs dossiers qui correspondront à plusieurs sites ayant chacun leur base de donnée propre.

+ **UwAmp** Il est aussi possible d'utiliser une version portable (sans installation) de Wamp qui peut donc « tourner » sur clé USB. C'est la solution que nous privilégions : <https://www.uwamp.com/fr/>

Télécharger la version zippée

-  [Télécharger Zip \(79 Mo\)](#)

## Utilisation

Pour démarrer UwAmp, double-clic sur le fichier « UwAmp.exe » vous verrez apparaître dans la barre des tâches une **icône verte et bleue**.

Clic gauche dessus, le menu apparaît :

- **Navigateur www** va vous permettre d'accéder à votre site et déclenchera son ouverture dans votre navigateur par défaut. L'URL de votre site en local aura la forme :  
`http://localhost/webmaster-blog/`
- **PHPMysqlAdmin** vous permettra d'accéder à l'interface de gestion de vos bases de données (à manier avec précaution !)
- **Dossier www** est un raccourci qui vous permettant d'accéder directement au dossier racine de votre site : **tous les fichiers sources**.
- **Apache/PHP/MySQL** vous permettent d'avoir des renseignements sur votre configuration en local, par exemple PHP 5.3.5, voire de la modifier (fichier php.ini) ...



# WordPress



## Wiki

WordPress est un système de gestion de contenu libre écrit en PHP, reposant sur une base de données MySQL, et distribué par [Automattic](#). WordPress est surtout utilisé comme moteur de blog, mais ses fonctionnalités lui permettent également de gérer n'importe quel site web. Il est distribué selon les termes de la GNU GPL. Le logiciel est aussi à l'origine du service WordPress.com.

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

## Sites de références

- Site officiel : <http://wordpress.org/>  
Version Française (téléchargement) : <http://fr.wordpress.org/>
- Documentation technique : <http://codex.wordpress.org/>  
Version Française : <http://codex.wordpress.org/fr:Accueil>
- Forum d'entraide (fr) : <http://www.wordpress-fr.net/support/>
- Blog Wordpress (en ligne) : <http://wordpress.com/>  
Version Française : <http://fr.wordpress.com/>

**TP25\_installation\_wordpress\_local**

# Back-office

## Accès

<http://localhost/webmaster-blog/wp-admin/>

## Back-office du site

Le menu principal (colonne gauche) est partagé en trois section :

- le tableau de bord : page d'accueil du back-office
- les contenus [Articles - Médias - Pages - Commentaires]
- les paramètres du site [Apparence - Extensions - Utilisateurs - Outils - Réglages]

## Contenus "pages" ou "articles" ?

Pages et articles sont sensiblement la même chose : elles contiennent un titre, du contenu

- Les pages seront utilisées pour publier du contenu dont la date de publication importe peu.  
Ex : contact, présentation, mentions légales...

Si vous installez un thème, regardez dans la page s'il propose des "modèles de page"

- Les articles (posts) sont utilisés pour publier des informations comme des billets dans un blog (la date compte). Ex : Actualités...

Documentation : <https://codex.wordpress.org/fr:Pages>

### **Widget**

Apparence > Widgets. Spécificité de Wordpress, il permettra d'installer des applications en un clic, c'est une sorte "mini-plugin" offrant une fonctionnalités : calendrier, derniers posts, formulaire de recherche...

### **Gestion des menus Front-office**

Apparence > Menus. Vous pouvez construire très facilement le/les menu du site Web.

Un menu peut être constitué de pages, articles, catégories, ou liens libres.

### **Utilisateurs**

Il peut y avoir plusieurs rédacteurs pour un même site. Lors de l'ajout d'un nouvel utilisateur, il est possible de déterminer quels seront ses "pouvoirs" (ecrire, publier...).

[http://codex.wordpress.org/Rôles\\_et\\_Permissions#Super\\_pouvoirs\\_pour\\_le\\_propriétaire\\_du\\_blog](http://codex.wordpress.org/Rôles_et_Permissions#Super_pouvoirs_pour_le_propriétaire_du_blog)

## **TP Guidé : prise en main du back-office**

### **Faite une nouvelle installation de WordPress (nouveau site) : demo-wp**

**+ Supprimez les articles et pages présents**

**+ Créez 2 catégories : Blog / Créations**

**+ Créez 4 articles : Actualité 1 / Actualité 2 / Création 1 / Création 2**

**Insérez dans ces articles du faux texte (lorem ipsum).**

**Pour chaque article, choisissez une image mise en avant au format paysage. Cette image doit être optimisée pour le web (voir images > divers)**

**Mettez l'actualité 1 en avant sur la page d'accueil (visibilité). Testez dans cet article l'éditeur : gras, listes, liens, titres, citations...**

**Actualité 1 et actualité 2 seront affectés à la catégorie blog. Création 1 et Création 2 à la**

**catégorie Créations (décochez "non classé)**

**Création 1 aura les étiquettes : HTML5, CSS3, Photoshop. Création 2 aura les étiquettes : Photoshop, Wordpress**

**+ Créez 2 pages : A propos / Contact**

**Insérez dans ces pages du faux texte (lorem ipsum).**

**Choisissez une image mise en avant au format paysage (cette image doit être optimisée pour le web).**

# Thèmes

C'est la grande force de WordPress !

Une impressionnante galerie de thèmes gratuits (ou presque...) qui vont permettre de changer l'habillage du site : <https://fr.wordpress.org/themes/browse/new/>

Les thèmes offrent plus ou moins d'options de configurations, par exemple :

- [Virtue](#) - Theme très complet avec beaucoup d'options possibles
- [Twenty Sixteen](#), thème plus simple à paramétrer.

Certains thèmes intègrent des plugins/widgets. Donc il vaut mieux **d'abord se fixer sur un thème** et dans un second temps travailler : menus, widgets, extensions.

Pour voir à quoi le thème ressemble avant de l'installer, oubliez « l'aperçu » (bouton haut/droite) car le rendu n'est pas très fidèle. Lorsque l'option est disponible, rendez-vous plutôt sur le site de l'éditeur :

[Page d'accueil du thème →](#)

## TP26\_paramétrage\_thème

# Extensions (plugins)

Les plugins permettent d'étendre les fonctionnalités de wordpress. Ils peuvent être gratuits ou payants, plus ou moins simples à configurer.

ATTENTION : Sur un site en production, **n'installer que les plugins absolument nécessaires** (en fonction de vos besoins). S'il existe une solution sans passer par un plugin, privilégiez-la.

Les plugins ralentissent le site et augmentent les probabilités d'incompatibilités (entre plugins et au changement de version de WordPress). De plus ils « alourdissent » le menu du back-office.

Vérifiez toujours que l'extension est **compatible avec votre version de WordPress**.

- **Tous les plugins (authentifiés par Wordpress) :** <http://wordpress.org/plugins/>

- Plugins sélectionnés (fr) : <http://wpformation.com/plugins-installation-wordpress/>
- Top plugins (fr) : <http://www.echodesplugins.li-an.fr/category/top-plugins/>

## TP27\_extensions\_wordpress

# Optimisation

Voir en ligne : <https://bulledev.com/optimiser-wordpress/>

- Dans wp-config.php (désactive les révisions d'articles)

```
define( 'WP_POST_REVISIONS', false );
```

- Dans wp-config.php - a la fin du fichier (augmente la taille mémoire)

```
define('WP_MEMORY_LIMIT', '96M');
```

## TP28\_woocommerce

# Création de thème

## Organisation des fichiers sources

Dans les fichiers présents directement à la racine du site, à noter :

- **index.php** : 1er fichier ciblé par le navigateur, sa seule fonction est de charger le thème choisi en back-office.
- **wp-config** : fichier créé lors de l'installation de Wordpress, sont enregistrées les informations de connexion à la base de donnée.

Dossiers racine :

- **wp-admin** : fichiers sources du back-office. Les modifications sur ces fichiers ne peuvent être effectuées que par des développeurs confirmés.
- **wp-includes** : ensemble des fonctions php pour la partie front-end. Constitue le noyau (Core) du logiciel. Domaine également réservé aux développeurs back-end car une mauvaise manipulation dans ces fichiers peut entraîner des bugs.
- **wp-content** : ce dossier contient tous les fichiers HTML/CSS gérant l'affichage des pages du site. C'est donc quasi-exclusivement dans les fichiers sources de ce dossier qu'interviendra le développeur front-end

Contenu de wp-content :

- **languages** : gestion des traductions
- **plugins** : toutes les extensions chargées depuis le back-office (quel que soit le theme choisi) viennent se placer dans ce dossier. Vous pouvez déposer directement un plugin dans ce dossier et l'activer par la suite en back-office
- **themes** : affichage du site à l'écran. Contient les différents themes par défaut + éventuellement ceux chargés par vous. Un seul des dossiers contenu dans thèmes est actif (celui sélectionné en back-office).
- **upgrade** : gestion des mises à jour
- **upload** : tous les fichiers chargés depuis le back-office sont placés (classés) dans ce dossier

## Principaux fichiers du thème

Les fichiers constitutifs d'un thème sont contenus dans wp-content > themes > [mon-theme]. Tout ce qui va permettre l'affichage du site à l'écran (HTML/CSS)

- **style.css**

Ce fichier contiendra le CSS de votre site.

Reprendre une feuille de style d'un autre theme et compléter les informations enregistrées en commentaires en début de page. Elles seront affichées en back-office.

Vous pouvez également garder la partie "reset" de la feuille de style ainsi copiée.

```

1  /*
2  Theme Name: Webmaster Blog
3  Theme URI: http://wordpress.org/themes/webmaster-blog
4  Author: Damien B
5  Author URI: http://31-place-web.net/
6  Description: Theme simple bleu et blanc responsive design
7  Version: 1.0
8  Tags: web, blog, bleu, blanc, responsive
9
10 */

```

### TP29\_modifications\_css\_wordpress

- **index.php**

Page d'accueil du site. Sera également affiché si aucun autre template n'est disponible pour afficher une page demandée.

Tout theme donc être composé d'au moins ces 2 fichiers : index.php et style.css. En fonction du type de contenu à afficher, différents fichiers php pourront être appelés.

Voici les autres principaux fichiers qui peuvent être utilisés dans un theme :

- Page > page.php
- Article (Post) > single.php
- Categorie > category.php
- Archive (classée par date) > date.php
- Resultat de recherche > search.php
- Mot clé > tag.php
- 404 (page introuvable) > 404.php
- Auteur > author.php

Si l'un ou plusieurs fichiers suivants n'existent pas : category.php, tag.php, author.php, date.php, la page renvoyée sera : **archive.php**

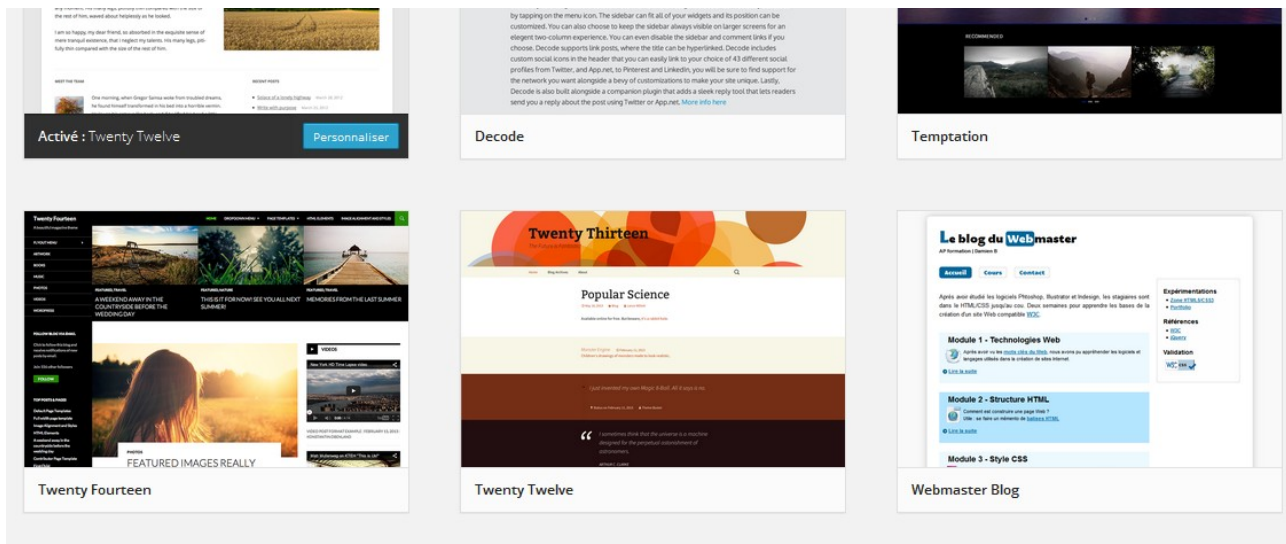


- **functions.php**

Active les fonctionnalités prévues par le thème : menu de navigation, miniatures, fonctions personnalisées...

- **screenshot.png**

Affiche la vignette en back-office



## TP30\_theme\_wordpress\_structure

## Les fonctions PHP dans WordPress

Pour rendre le site dynamique (administrable) il va falloir remplacer le texte et images "en dur" dans le HTML par des fonctions PHP permettant de modifier ces éléments depuis le back-office WordPress.

Les fonctions PHP dans WordPress seront utilisées pour des tâches très variées : afficher un titre, une liste d'articles, insérer un menu, des widgets, afficher le contenu des articles...

- Tout ce qui concerne le code propre à Wordpress se trouve dans le Codex (essentiellement en Anglais) : <http://codex.wordpress.org/fr:Accueil>
- Une refonte de la documentation est en cours : <https://developer.wordpress.org/>
- La liste des fonctions et leur utilisation (Anglais) : [https://codex.wordpress.org/Function\\_Reference](https://codex.wordpress.org/Function_Reference)

**Exemple de fonctions d'affichage : `bloginfo()` ;** (ici pour l'affichage du titre et du slogan du site) :

#### Back-office

Titre de site et description ▲

Titre du site

Le blog du Webmaster

Slogan

Site de test WordPress

#### Code PHP

```
<?php bloginfo('name'); ?>
```

```
<?php bloginfo('description'); ?>
```

#### CODE PHP et HTML

```
<header id="entete">
    <h1 class="site"><?php bloginfo('name'); ?></h1>
    <p class="sous_titre"><?php bloginfo('description'); ?></p>
</header>
```

## Les inclusions

Dans un soucis de simplification et de maintenabilité, il va être possible d'inclure des parties de code qui sont les mêmes de pages en pages.

Généralement :

- Entete > `header.php`
- Pied de page > `footer.php`
- Colonne latérale > `sidebar.php`

Ainsi par exemple, une modification dans l'inclusion `header.php` se fera dans les pages : `single.php`, `tag.php`, `page.php`, `index.php`...

Dans ces pages, l'inclusion des fichiers grâce aux fonctions suivantes :

- `<?php get_header(); ?>` | `<?php get_sidebar(); ?>` | `<?php get_footer(); ?>`

**index.php après inclusions :**

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

<!-- Début conteneur principal-->
<main class="principal conteneur flex">
  <!-- Liste des articles-->
  <section id="page">
    <!-- Article 1-->
    <article class="index">
      <!-- Contenu de l'article-->
      </article>
    <!-- Fin article 1-->
  </section>
  <!-- Fin liste des articles -->

  <?php get_sidebar(); ?>
</main>
<!-- Fin conteneur principal-->

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

**TP31\_theme\_wordpress\_inclusions**

## header.php

```

<!DOCTYPE html>
<html <?php language_attributes(); ?>>

<head>
    <meta charset="<?php bloginfo('charset'); ?>" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link href="https://fonts.googleapis.com/css?
family=Montserrat:400,500,700%7COpen+Sans" rel="stylesheet">
    <link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.1.0/css/all.css"
integrity="sha384-1KuwvrZot6UHsBSfcMvOkWwlCMgc0TaWr+30HWE3a41taBwTZhyTEggF5tJv8tbt"
crossorigin="anonymous">
    <link rel="pingback" href="<?php bloginfo( 'pingback_url' ); ?>">
    <?php wp_head(); ?>
    <!--/l/ wp_head() - !important! - appels javascript et css des plugins et charge les
principaux css du theme-->
</head>

<body id="top" <?php body_class(); ?>>
    <!-- Début entete-->
    <header class="fond">
        <div class="conteneur">
            <div id="entete">
                <h1 class="logo">
                    <a href="<?php bloginfo('url'); ?>"><?php
bloginfo('name'); ?></a>
                </h1>
                <p><?php bloginfo('description'); ?></p>
            </div>
        </div>
        <!--Navigation à travailler (cf "les menus du site" ci-dessous)-->
        <nav id="menu">
            <ul>
                <li><a class="actif" href="index.html">Accueil</a></li>
                <li><a href="portfolio.html">Portfolio</a></li>
                <li><a href="zone-html5.html">Labo</a></li>
                <li><a href="contact.html">Contact</a></li>
            </ul>
        </nav>
        <nav id="menu-mobile">
            <i class="fas fa-bars"></i>
            <ul>
                <li><a class="actif" href="index.html">Accueil</a></li>
                <li><a href="portfolio.html">Portfolio</a></li>
                <li><a href="zone-html5.html">Labo</a></li>
                <li><a href="contact.html">Contact</a></li>
            </ul>
        </nav>
        <!--fin note - zone de navigation à travailler (statique)-->
    </header>
    <!-- Fin entete-->

```

PHP	Code HTML généré
<code>&lt;html &lt;?php language_attributes(); ?&gt;&gt;</code>	<code>&lt;html lang="fr-FR"&gt;</code>
<code>&lt;meta charset="&lt;?php bloginfo( 'charset' ); ?&gt;" /&gt;</code>	<code>&lt;meta charset="UTF-8" /&gt;</code>
<code>&lt;title&gt;&lt;?php wp_title( ' ', true, 'right' ); ?&gt;&lt;/title&gt;</code> <i>(A mettre si n'est pas géré dans fonctions.php - Si vous utilisez [theme-base] fourni dans ce cours, vous n'avez pas à insérer cette ligne)</i>	<code>&lt;title&gt;Portfolio   Le blog du webmaster&lt;/title&gt;</code>
<code>&lt;link rel="pingback" href="&lt;?php bloginfo( 'pingback_url' ); ?&gt;"&gt;</code>	Documentation : <a href="https://codex.wordpress.org/Introduction_to_Blogging#Pingbacks">https://codex.wordpress.org/Introduction_to_Blogging#Pingbacks</a>
<code>&lt;head&gt;</code> ... <code>&lt;?php wp_head(); ?&gt;</code> <code>&lt;/head&gt;</code>	<b>Très important</b> Permet aux plugins d'insérer du code dans le <code>&lt;head&gt;</code> (css, js). Lien vers le fichier "style.css". A placer juste avant la fermeture du <code>&lt;/head&gt;</code>
<code>&lt;h1&gt;&lt;?php bloginfo('name'); ?&gt;&lt;/h1&gt;</code>	<code>&lt;h1&gt;Le blog du Webmaster&lt;/h1&gt;</code>
<code>&lt;h1&gt;</code> <code>&lt;a href="&lt;?php bloginfo('url'); ?&gt;"&gt;</code> <code>&lt;?php bloginfo('name'); ?&gt;</code> <code>&lt;/a&gt;</code> <code>&lt;/h1&gt;</code>	<code>&lt;h1&gt;</code> <code>&lt;a href="http://localhost/webmaster-blog"&gt;</code> <code>Le blog du Webmaster</code> <code>&lt;/a&gt;</code> <code>&lt;/h1&gt;</code>
<code>&lt;p&gt;&lt;?php bloginfo('description'); ?&gt;&lt;/p&gt;</code>	<code>&lt;p&gt;Site de test Wordpress&lt;/p&gt;</code>
<code>&lt;body &lt;?php body_class(); ?&gt; &gt; &lt;/body&gt;</code>	Généré dans pour un article par exemple : <code>&lt;body class="post-template-default single single-post postid-186 single-format-standard"&gt;&lt;/body&gt;</code>

- Ne pas mettre les balises : `<meta name="description">` - `<meta name="author">` ... qui seront générées par le plugin Yoast

- Ne pas mettre le lien vers le favicon `<link rel="shortcut icon" >`

- Des fonctions particulières à votre thème se trouvent dans le fichier fonctions.php, comme par exemple de générer dynamiquement le `<title>` (c'est le cas dans webmasterblog).

## footer.php

```

<!-- Début pied de page-->
<footer class="conteneur">
  <div id="pied" class="flex">
    <small><a href="#" target="_blank">31placeweb.net</a></small>
    <!--zone de navigation à travailler-->
    <nav id="menu-pied">
      <ul>
        <li><a href="#">Crédits</a></li>
        <li><a href="#">Mentions légales</a></li>
        <li><a href="#">Contact</a></li>
      </ul>
    </nav>

    <ul>
      <li><a href="#"><i class="fab fa-facebook-square fa-lg"></i></a></li>
      <li><a href="#"><i class="fab fa-twitter-square fa-lg"></i></a></li>
      <li><a href="#"><i class="fab fa-linkedin fa-lg"></i></a></li>
    </ul>
    <!--zone de navigation à travailler-->
  </div>
  <a href="#top" class="top js-scrollTo"><i class="fa-lg fas fa-chevron-up"></i></a>
</footer>
<!-- Fin pied de page-->

<?php wp_footer(); ?>
<!--/2/ wp_footer() - !important! - appels javascript des plugins-->
</body>

</html>

```

```

<?php wp_footer(); ?>
</body>

```

### Très important

Permet aux plugins d'insérer du code avant la fermeture de <body>

## sidebar.php (widgets)

Vous pouvez inclure les widgets Wordpress à l'endroit que vous voulez dans votre template. Le plus souvent dans l'inclusion **sidebar.php**. La zone de widget a été travaillée au préalable dans : Apparence > Widgets, nous pouvons l'inclure au template grâce à la fonction :

### Inclusion dans la colonne droite

```
<!-- Début colonne-->
<aside id="colonne">
    <?php dynamic_sidebar('colonne'); ?>
</aside>
<!-- Fin colonne-->
```

L'identifiant '**colonne**' est défini dans **functions.php**, vous pouvez modifier les paramètres de la fonction : **webmasterblog\_widgets\_init()**.

### Nouvelle zone de Widgets

Vous pouvez bien entendu créer une nouvelle zone de widgets (en pied de page par exemple) : dupliquez l'ensemble du bloc **register\_sidebar()** dans et changez le paramètre : id=>'pied' (ainsi que *name* et *description* associés qui seront visibles en back-office)

Appelez cette nouvelle zone de widgets dans votre footer.php avec :

```
<footer>
    <?php dynamic_sidebar('pied'); ?>
</footer>
```

## Les menus du site

### Plusieurs menus

Vous pouvez déclarer plusieurs menus dans **functions.php** en modifiant les paramètres de la fonction : **register\_nav\_menus()** (3 menus déclarés dans le theme webmasterblog : principal – mobile – pied).

+ Vous pouvez appeler dans **header.php** le **menu principal** paramétré depuis le back-office identifiant donné dans **register\_nav\_menus() : 'entete'**

```
<?php wp_nav_menu( array( 'theme_location' =>
'principal', 'container'=>'nav', 'container_id'=>'menu' ) ); ?>
```

+ En dessous de ce menu pour insérer une navigation mobile :

```
<!--Menu mobile-->
<?php wp_nav_menu( array( 'theme_location' =>
'mobile','container'=>'nav', 'container_id'=>'menu-mobile' ) ); ?>
```

+ De même, dans footer.php, pour appeler le menu pied de page :

```
<!--Menu pied-->
<?php wp_nav_menu( array( 'theme_location' => 'pied','container'=>'nav',
'container_id'=>'menu-pied' ) ); ?>
```

## La boucle (The loop)

Une boucle va permettre d'afficher **les contenus des Articles et Pages** :

[http://codex.wordpress.org/fr:La Boucle](http://codex.wordpress.org/fr:La_Boucle)

### Structure simple

```
<?php while ( have_posts() ) : the_post(); ?>
    <!-- Pour CHAQUE article, j'affiche ça (titre, texte, auteur...) = BOUCLE -->
<?php endwhile; ?>
```

### Structure avec condition

```
<?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>
<!-- SI il y a au moins 1 article, j'affiche ça (titre, texte, auteur...) -->
<?php endwhile; else : ?>
<!-- SINON j'affiche ça ("Désolé pas de contenu") -->
<?php endif; ?>
```

La boucle sera aussi bien utilisée pour afficher le contenu d'un article isolé (dans single.php) que pour afficher la liste des derniers articles en page d'accueil (dans index.php). Elle prend en compte son **environnement** (page.php, archives.php) et renvoie les contenus associés sans avoir à modifier la boucle.

Tous les champs mobilisables à l'intérieur d'une boucle s'appellent des marqueurs de modèle (template tags) : titre, texte, auteur, date de publication etc... :

[https://codex.wordpress.org/fr:Marqueurs de Modele](https://codex.wordpress.org/fr:Marqueurs_de_Modele)

Nous allons voir ci-dessous l'utilisation de la boucle dans les principaux fichiers du thème :



## single.php

Ce fichier va permettre de gérer l'affichage d'**un article** (post). Une actualité par exemple. Les inclusions header, footer et éventuellement sidebar sont à placer dans ce fichier.

Le contenu de l'article sera affiché dans la boucle (`<?php the_title() ?>`...).

Liste des champs qui peuvent être affichés (titre de l'article, catégorie, auteur, date...):

[http://codex.wordpress.org/Template\\_Tags#Post tags](http://codex.wordpress.org/Template_Tags#Post_tags)

### single.php

```
<?php get_header(); ?>
<main class="principal conteneur flex">
  <article id="page">
    <!-- DEBUT de la boucle -->
    <?php while (have_posts()) : the_post(); ?>

      <!-- Affiche l'image mise en avant de l'article-->
      <?php the_post_thumbnail('medium_large') ?>

      <div class="inter">
        <h1>
          <!-- Affiche le titre de l'article-->
          <?php the_title() ?>
        </h1>
        <p class="infos">
          <!-- Affiche la date, l'auteur et la/les catégories-->
          <span><?php the_date(); ?></span> /
          <span><?php the_author(); ?></span> /
          <span><?php the_category(', '); ?></span>
        </p>
        <!-- Affiche le contenu de l'article-->
        <?php the_content() ?>
        <div class="etiquettes">
          <p>
            <!-- Affiche les mots clés de l'article-->
            <?php the_tags('<strong>Mots clés :</strong> '); ?>
          </p>
        </div>
        <!-- affiche les commentaires de l'article -->
        <?php comments_template(); ?>
      </div>
    <!-- FIN de la boucle -->
    <?php endwhile; ?>
  </article>
  <?php get_sidebar(); ?>
</main>
<?php get_footer(); ?>
```

## Les posts formats

Il est possible de choisir un affichage différencié en fonction du type d'article : galerie, citation, video..., Voir à ce sujet : <http://tulipemedia.com/tutoriel-et-guide-sur-les-post-formats-de-wordpress/>

## page.php

Ce template servira à afficher les pages du site (A propos, contact, mentions légales...).

L'environnement sera sensiblement le même que pour single.php. sans l'affichage de la catégorie (réservées aux articles). Il n'est peut-être pas nécessaire d'afficher l'auteur de la page, la date de publication, les commentaires etc...

Dans le modele ci-dessous la page s'affiche par défaut sans colonne (pas d'inclusion de sidebar.php)

### page.php

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

<main class="principal conteneur">

    <article id="pleine-page" >
    <!-- DEBUT de la boucle -->
    <?php while (have_posts()) : the_post(); ?>

        <!-- Affiche l'image mise en avant de la page-->
        <?php the_post_thumbnail('full') ?>

        <div class="inter">
            <h1>
                <!-- Affiche le titre de la page-->
                <?php the_title() ?>
            </h1>

            <!-- Affiche le contenu de la page-->
            <?php the_content() ?>
        </div>

    <!-- FIN de la boucle -->
    <?php endwhile; ?>
    </article>

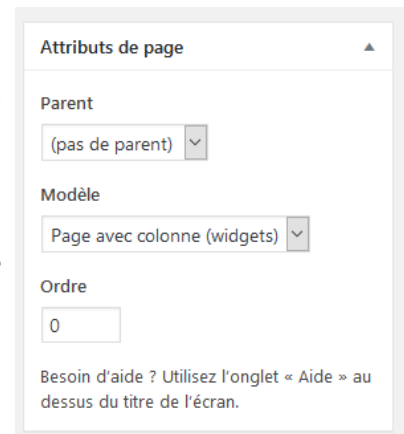
</main>

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

## Différents gabarits de pages

Il est possible d'avoir un affichage différent pour certaines pages. Par exemple, un modèle de page AVEC colonne.

Dans ce cas, créer un nouveau fichier PHP **page-colonne.php** par exemple, placé dans le répertoire **page-templates**. La déclaration de ce nouveau fichier se fait grâce au commentaire PHP **Template Name** en début de code



### page-colonne.php

```
<?php
/*
Template Name: Page avec colonne (widgets)
*/
?>

<?php get_header(); ?>

<main class="principal conteneur flex">
  <article id="page" >
    <!-- DEBUT de la boucle -->
    <?php while (have_posts()) : the_post(); ?>

    <!-- Affiche l'image mise en avant de la page-->
    <?php the_post_thumbnail('medium_large') ?>

    <div class="inter">
      <h1>
        <!-- Affiche le titre de la page-->
        <?php the_title() ?>
      </h1>

      <!-- Affiche le contenu de la page-->
      <?php the_content() ?>
    </div>

    <!-- FIN de la boucle -->
    <?php endwhile; ?>
  </article>

  <?php get_sidebar(); ?>
</main>

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

# index.php

Page d'accueil du site.

- Paramétrer `<?php the_excerpt(); ?>` : Premières lignes de texte de l'article [http://codex.wordpress.org/the\\_excerpt](http://codex.wordpress.org/the_excerpt) - (paramétrages à la fin de functions.php)
- Paramétrer `<?php the_post_thumbnail() ?>` avec des tailles d'image personnalisées "custom-size", voir la fonction **add\_image\_size** à la fin du fichier **functions.php**

## index.php

```

<?php get_header(); ?>

<!-- Début conteneur principal-->
<main class="principal conteneur flex">
  <!-- Liste des articles-->
  <section id="page">
    <!-- DEBUT BOUCLE-->
    <?php if ( have_posts() ) : while ( have_posts() ) :
the_post(); ?>

    <article class="index">
      <a href="<?php the_permalink(); ?>">
        <?php the_post_thumbnail('custom-size') ?>
      </a>
      <div class="presentation">
        <h2>
          <a href="<?php the_permalink() ?>">
            <?php the_title(); ?>
          </a>
        </h2>
        <p class="infos">
          <span><?php echo get_the_date(); ?></span>/
          <span><?php the_author(); ?></span>/
          <span><?php the_category(' , '); ?></span>
        </p>
        <?php the_excerpt(); ?>
      </div>
    </article>

    <!-- FIN BOUCLE-->
    <?php endwhile; else: ?>
    <p>
      <?php _e("Il n'y a pas de publication ce jour."); ?>
    </p>
    <?php endif; ?>
  </section>
  <!-- Fin liste des articles -->

  <?php get_sidebar(); ?>
</main>
<!-- Fin conteneur principal-->

<?php get_footer(); ?>

```

## archives.php

Ce fichier affiche une **liste d'articles** liés à : une catégorie - **category.php**, un mot clé - **tag.php**, un auteur - **author.php**, une date - **date.php** si ces fichiers ne sont pas présents dans le theme.

La boucle « reconnaissant » seule son environnement, elle s'adaptera Cela peut être un copié/collé du fichier **index.php** ci-dessus, ou réalisez un autre affichage HTML

## 404.php

Page d'erreur, page introuvable

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

<main class="principal conteneur">

    <article id="pleine-page">
        <!-- DEBUT de la boucle -->

        <div class="inter">
            <h1>Erreur 404</h1>
            <p>Nous sommes désolé mais la page recherchée n'existe plus ou a été
déplacée, revenez à <a href="<?php bloginfo('url'); ?>">l'accueil du site</a> ou
effectuez une recherche :</p> <?php get_search_form(); ?>

        </div>
    </article>

</main>

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

# search.php

Page d'affichage des résultats d'une recherche dans le site (si moteur de recherche)

```
<?php get_header(); ?>
<main class="principal conteneur">
  <article id="pleine-page">

    <div class="inter">
      <h1>
        Résultats de la recherche :
      </h1>
      <ul>

        <!-- DEBUT de la boucle -->
        <?php if ( have_posts() ) : while ( have_posts() ) : the_post(); ?>

          <li><a href="<?php the_permalink();?>">
            <?php the_title();?></a></li>

          <!-- FIN de la boucle -->

        <?php endwhile; else: ?>
      </ul>
      <p>
        <?php _e("Aucun résultat pour cette recherche"); ?>
      </p>
    <?php endif; ?>
  </div>

</article>
</main>
<?php get_footer(); ?>
```

Vous pouvez améliorer l'affichage des résultats (avec titre, résumé, image à la une...)

**TP32\_theme\_wordpress\_finalisation**

## WP\_Query

Afin de gérer au mieux l'affichage sur des sites complexes, il faut parfois faire des requêtes élaborées. Par exemple : Tous les articles de la "catégorie 2" - Tous les articles ayant le mot clé "développement", classés par date inverse...

### Afficher sur la page d'accueil le titre et texte de la page "A propos" (identifiant=4)

```
<?php
$args= array(
    'post_type' => 'page',
    'page_id' => 4
);

$query = new WP_Query($args);
while ($query->have_posts()) : $query->the_post();
?>
<h2><?php the_title(); ?></h2>
<?php the_content();?>
<?php endwhile; ?>
```

Si une autre boucle doit être placée **après** celle-ci, ajouter (après <?php endwhile; ?>) :

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

### Autres exemples de requêtes (5 articles de la catégorie voiture par date ...)

<http://www.naxialis.com/comment-utilise-la-requete-wp-query-pour-laffichage-de-vos-articles/>

```
$args= array(
    'showposts' => 5, // nombre d'article
    'category_name' => 'voiture', // nom de la catégorie (identifiant
sans accent et majuscule)
    'orderby' => 'date', // ordre par date
    'order' => 'desc', // ordre descendant
    'tag' => 'rouge' // mot clé (identifiant sans accent et majuscule)
);
?>
```

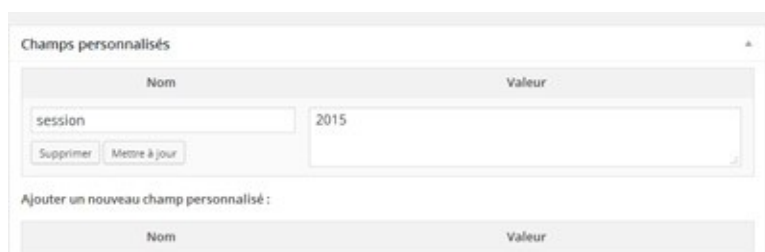
Liste des paramètres possibles pour la fonction WP\_Query :

[http://codex.wordpress.org/Class Reference/WP\\_Query#Parameters](http://codex.wordpress.org/Class_Reference/WP_Query#Parameters)

# Meta Boxes

Gérer l'affichage d'informations supplémentaires dans des zones prédéterminées.

## Back-office



## Dans page.php

```
<p>  
<strong>Session :</strong>  
    <?php echo get_post_meta($post->ID, "session", true); ?>  
</p>
```

## Aller Plus loin

Pour insérer d'autres types de champs : cases à cocher, zone de texte, upload de pdf...

Voir le tuto en ligne : <http://wabeo.fr/jouons-avec-les-meta-boxes/>

## Avec un plugin

Pour gérer plus facilement la création et l'appel de nouveaux champs personnalisés :

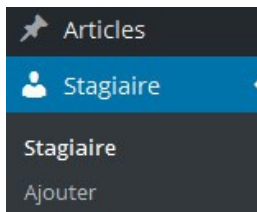
<https://www.advancedcustomfields.com/>



# Custom Post Types

Créer de nouveaux objets éditoriaux

## Back-office



## Dans functions.php

```
add_action( 'init', 'create_post_type' );
function create_post_type() {
    register_post_type( 'stagiaire',
        array(
            'labels' => array(
                'name' => __( 'Stagiaire' ),
                'singular_name' => __( 'Stagiaires' )
            ),
            'public' => true,
            'has_archive' => true,
            'menu_position' => 5,
            'menu_icon' => 'dashicons-admin-users',
        )
    );
}
```

## Documentation

[https://codex.wordpress.org/Function\\_Reference/register\\_post\\_type](https://codex.wordpress.org/Function_Reference/register_post_type)

**A contrario**, pour supprimer certains éléments du menu (Widgets, outils...):

<https://wpchannel.com/masquer-menus-administration-back-office-wordpress/>

**TP33\_export\_bdd\_wordpress**