

Module 5

WordPress 4.4

Content Management System

Table des matières

WORDPRESS.....	4
Environnement serveur.....	4
PHP5/MySQL.....	4
Site statique.....	4
Site dynamique.....	4
Technologies.....	4
En local sous WAMP.....	5
Présentation du logiciel.....	5
Utilisation.....	5
Les gestionnaires de contenu.....	7
C.M.S.....	7
Les solutions issues du libre.....	8
Sites vitrines et blogs.....	8
Sites de vente en ligne.....	8
WordPress.....	9
Wiki.....	9
Sites de références.....	9
Back-office.....	10
Thèmes.....	11
Extensions (plugins).....	11
Optimisation.....	12
Création de thème.....	13
Organisation des fichiers sources.....	13
Principaux fichiers du thème.....	14
Les inclusions.....	15
Les fonctions.....	16
Gestion des Widgets.....	19
Gestion du menu.....	19
La boucle (The loop).....	21
Affichage d'article- single.php.....	23
Affichage de page- page.php.....	23

Autres pages.....	25
WP_Query.....	28
Meta Boxes.....	29
Custom Post Types.....	30

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 4.1

The screenshot shows the WordPress 4.1 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 three published articles:

Titre	Auteur	Catégories	Mots-clés	Date
Formation Webmaster - 31 place Web	damien	Actualités	—	25/03/2014 Publié
Actualité 1	damien	Actualités	—	24/03/2014 Publié
Bonjour tout le monde !	damien	Non classé	—	24/03/2014 Publié

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 plug-ins (fonctionnalités) et thèmes prêt à l'emploi.

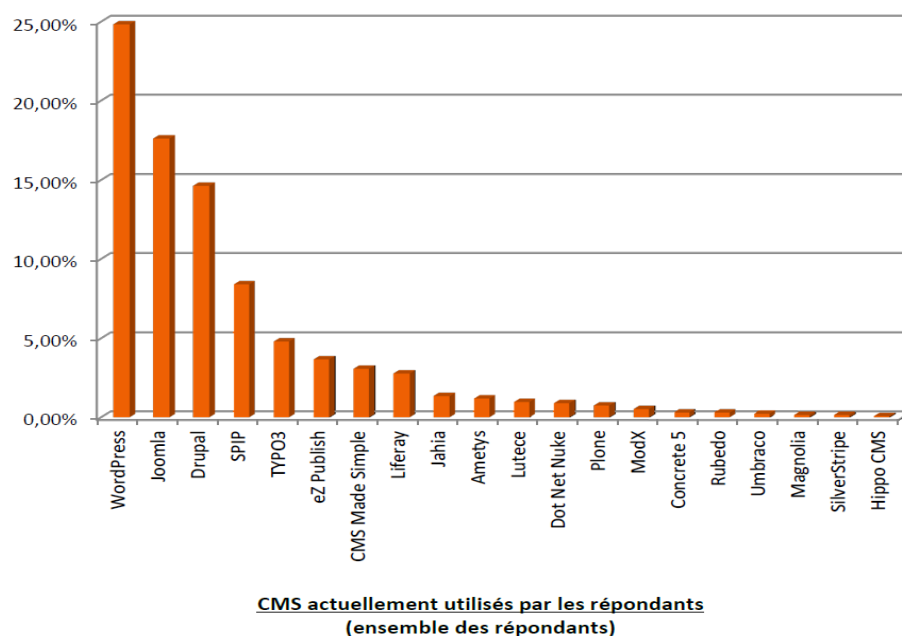
Les solutions issues du libre

Les plus utilisés en France 2014

Sites vitrines et blogs

- [Wordpress](#)
- [Joomla](#)
- [Drupal](#)
- [SPIP](#)

Part d'utilisation des C.M.S en France 2013 :



Source de l'image : Smile - <http://www.smile.fr/Livres-blancs/Gestion-de-contenu-et-ged/Enquete-cms>

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.

Utilisation

Démarrez Wamp, vous verrez apparaître dans la barre des taches une **icône verte**.

Clic gauche dessus, le menu apparaît :

- **Localhost** 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/`
- **phpMyAdmin** vous permettra d'accéder à l'interface de gestion de vos bases de données (à manier avec précaution !)
- **www directory** est un raccourci 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/>

[TP24_installation_wordpress_local]

Back-office

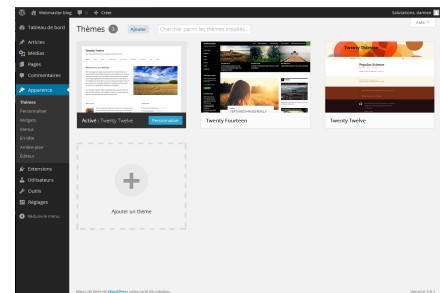
Accès

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

Menu principal back-office

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...
- 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

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

Vous pouvez construire très facilement le menu du site Web (front end) : **apparence > menu**.

Un menu peut être constitué de pages, catégories, ou liens libres. Pas d'article directement (passer par une catégorie).

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

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://wordpress.org/themes/browse/new/>

Les thèmes offrent plus ou moins d'option de configurations. Par exemple :

- [Virtue](#) - Theme très complet avec beaucoup d'options possibles (zone de widget, mise en page...)
- [ZeeMinty](#) – Theme plus simple à mettre en place (moins d'options).

ATTENTION : lorsque vous changez de thème, **les paramètres des menus et zones de widgets** (différents selon les thèmes) **seront à refaire**.

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.

[TP25_parametrage_theme]

Extensions (plugins)

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

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/>

[TP26_extensions_basiques]

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');
```

[TP27_extensions_avancees]

[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

[TP29_modifications_css_wordpress]

Principaux fichiers du thème

Les fichiers constitutifs d'un thème sont contenus dans wp-content > themes > [mon-thème]. 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 thème 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 */

```

- **index.php**

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

Tout thème doit 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 thème :

- Page > page.php
- Post (article) > single.php
- Catégorie > 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

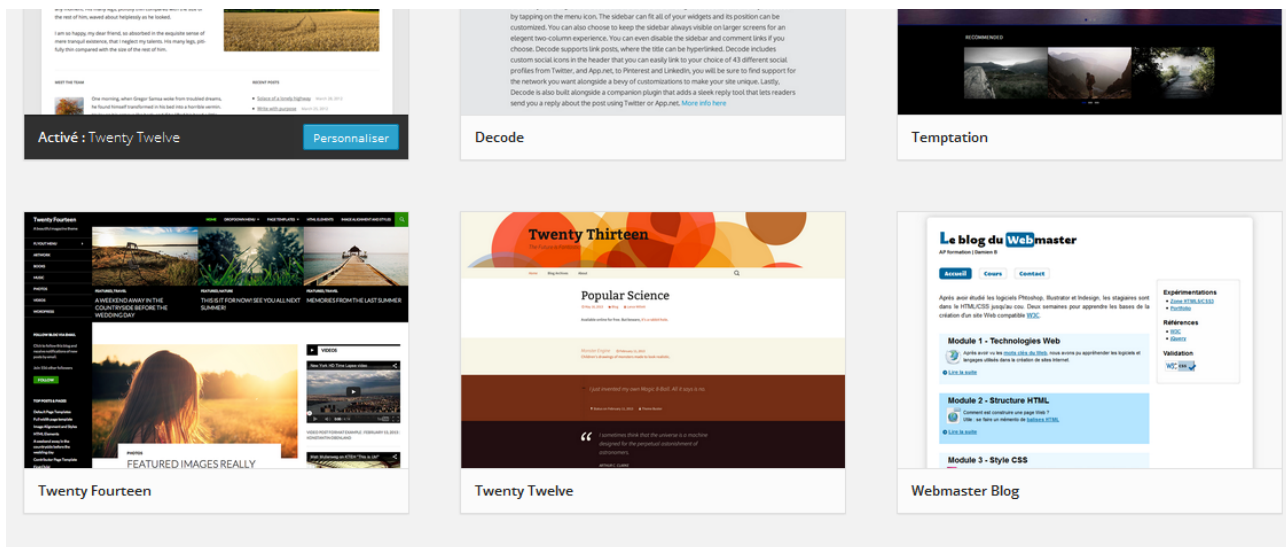
Si l'un ou plusieurs fichiers suivants n'existent pas : category.php, tag.php, author.php, date.php, la page renvoyée sera : **archive.php**

- **functions.php**

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

- **screenshot.png**

Taille : 880px / 660px, affiche la vignette en back-office



[TP30_theme_wordpress_structure]

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 codes suivants :

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

[TP31_theme_wordpress_inclusions]

Les fonctions

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.

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

Tout ce qui concerne le code propre à Wordpress se trouve dans le Codex (essentiellement en Anglais) : <http://codex.wordpress.org/fr:Accueil>

La liste des fonctions et leur utilisation (Anglais) : https://codex.wordpress.org/Function_Reference

Exemple de fonction d'affichage

Fonctions permettant l'affichage dynamique du titre et du slogan d'un site

Back-office

Titre de site et description ▲

Titre du site

Slogan

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>
```

Rendu (avec CSS lié)

Le blog du Webmaster

Site de test WordPress

Principales fonctions contenues dans :

- **header.php**

```

<!DOCTYPE html>
<html <?php language_attributes(); ?>>
<head>
  <meta charset="<?php bloginfo('charset'); ?>" />
  <!--[if lt IE 9]>
    <script src="http://html5shim.googlecode.com/svn/trunk/html5.js"></script>
  <![endif]-->
  <title><?php wp_title( '|', true, 'right' ); ?></title>
  <link rel="shortcut icon" href="<?php echo get_template_directory_uri(); ?>
>/favicon.ico" />
  <?php wp_head(); ?>
</head>

<body>
<!-- Début conteneur -->
<section id="conteneur">
  <!-- Début entete -->
  <header id="entete">
    <h1 class="site">
      <a href="<?php bloginfo('url'); ?>">
        <?php bloginfo('name'); ?>
      </a>
    </h1>
    <p class="sous_titre"><?php bloginfo('description'); ?></p>
  </header>
  <!-- Fin entete -->

```

Code HTML généré

<html <?php language_attributes(); ?>>	<html lang="fr-FR">
<meta charset="<?php bloginfo('charset'); ?>" />	<meta charset="UTF-8" />
<title><?php wp_title(' ', true, 'right'); ?></title>	<title>Portfolio Le blog du webmaster</title>
<head> ... <?php wp_head(); ?> </head>	Très important Permet aux plugins d'insérer du code dans le <head> (css, js). Lien vers le fichier "style.css". A placer juste avant la fermeture du </head>
<h1><?php bloginfo('name'); ?></h1>	<h1>Le blog du Webmaster</h1>
<h1> <a href="<?php bloginfo('url'); ?>"> <?php bloginfo('name'); ?> </h1>	<h1> Le blog du Webmaster </h1>
<p><?php bloginfo('description'); ?></p>	<p>Site de test Wordpress</p>

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

- **footer.php**

<pre>... <?php wp_footer(); ?> </body></pre>	<p>Très important</p> <p>Permet aux plugins d'insérer du code avant la fermeture de <body></p>
--	---

Des fonctions particulières à votre thème se trouvent dans le fichier **functions.php**, dans certains tutoriaux il peut être demandé d'apporter des modifications à ce fichier.

Gestion des Widgets

Vous pouvez inclure les widgets Wordpress à l'endroit que vous voulez dans votre template. Le plus souvent dans l'inclusion **sidebar.php**

L'inclusion se fait avec le code suivant :

Inclusion dans la colonne droite

```
<div id="colonne">
    <?php dynamic_sidebar(); ?>
</div>
```

Note : Les différents Widget sont inclus dans des balises <aside>

Gestion du menu

Vous pouvez appeler le menu principal paramétré depuis le back-office, nom donné au menu : "header" :

```
<?php wp_nav_menu( array( 'theme_location' => 'header', 'menu_class' => 'header-nav' ) ); ?>
```

Deux menus différents dans un site

Pour rajouter un menu différent en pied de page par exemple (mentions légales, liens ...) :

1- Créez un nouveau menu en back-office nommé "footer", incluant les pages concernées.

2- Modifier le fichier **functions.php**, concernant la fonction **register_nav_menu()**;

```
// This theme uses wp_nav_menu() in two locations.
register_nav_menu( 'header', __( 'Menu principal', 'webmasterblog' ) );
register_nav_menu( 'footer', __( 'Menu pied de page', 'webmasterblog' ) );
```

2- Vous pouvez alors utiliser dans **footer.php**

```
<?php wp_nav_menu( array( 'theme_location' => 'footer', 'menu_class' => 'footer-  
nav' ) ); ?>
```

La boucle (The loop)

Une boucle va permettre d'afficher le contenu d'un ou plusieurs articles ou de 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 Post, j'affiche ça = BOUCLE -->
<?php endwhile; ?>
```

Cette boucle peut être couplée à une fonction conditionnelle pour prévoir le fait qu'une boucle peut ne rien renvoyer (s'il n'y a pas d'articles enregistrés).

Structure avec condition

```
<?php if (have_posts()) : ?>
<!-- Si j'ai des Posts, j'affiche cette partie = CONDITION -->

  <?php while (have_posts()) : the_post(); ?>
  <!-- Pour CHAQUE Post, j'affiche ça = BOUCLE -->
  <?php endwhile; ?>

<?php else : ?>
  <!-- Si il n'y a pas de Post, j'affiche cette partie = FIN CONDITION -->
<?php endif; ?>
```

Exemple d'utilisation d'une boucle - Derniers articles affichés en page d'accueil

```

<!-- Début page -->
<section id="page">

    <?php if (have_posts()) : ?>
    <!-- Si condition est vérifiée (=il y a des articles), s'affiche une fois -->
    <h1>Actualités</h1>

        <!-- Début de la boucle, un ou plusieurs résultats -->
        <?php while (have_posts()) : the_post(); ?>
            <article class="index">
                <h2><?php the_title(); ?></h2>
                <?php the_post_thumbnail('thumbnail') ?>
                <!-- Image à la une affichage vignette -->
                <p><?php the_excerpt(); ?></p> <!-- Résumé -->
                <p class="clear">
                    <a href="<?php the_permalink(); ?>" > Lire la suite</a>
                </p>
            </article>
        <?php endwhile; ?>
    <!-- Fin de la boucle -->

    <!-- Si la boucle ne renvoie rien (pas d'article) afficher ceci : -->
    <?php else : ?>
        <p>Il n'y a Pas d'actualités</p>
    <?php endif; ?>

</section>
<!-- Fin page -->

```

- Paramétrer `<?php the_excerpt(); ?>` : http://codex.wordpress.org/the_excerpt
- Paramétrer `<?php the_post_thumbnail() ?>` : http://codex.wordpress.org/Post_Thumbnails
La taille des images (small-medium-large) est paramétrable en back-office : Réglages > Média

Affichage d'article- single.php

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

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

http://codex.wordpress.org/Template_Tags#Post_tags

Exemple d'article simple

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

<section id="page">

    <?php while (have_posts()) : the_post(); ?>
        <?php the_post_thumbnail('medium') ?>
        <h2><?php the_title(); ?></h2>
        <?php the_content(); ?>
    <?php endwhile; ?>

</section>

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

Il est possible d'afficher un forum en bas de page avec le code :

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

Affichage de page- page.php

Ce template servira à afficher des modèles de pages (A propos, contact...). L'environnement sera le même que pour single.php avec une fonction if et la **boucle while**. Le bout de code ci-dessus *Exemple d'article simple* sera opérationnel pour une page article.

Plusieurs gabarits

Il est possible d'avoir un affichage différent pour certaines pages.

Dans ce cas, créer un nouveau fichier PHP (que l'on nomme comme on veut) et faire une déclaration en début de fichier comme dans l'exemple ci-dessous.

Cette nouvelle page peut être placée dans le dossier **[page-templates]**, supprimez les modèles qui s'y trouvent déjà si vous avez fait une duplication de thème.

Exemple de modele de page sans colonne latérale

```

<?php
/*
Template Name: Page sans colonne
*/
?>

<?php get_header(); ?>
<!-- Il suffira de créer un CSS #page-large -->
<section id="page-large">

    <?php while (have_posts()) : the_post(); ?>
        <h2 class="titre_article"><?php the_title(); ?></h2>
        <?php the_content(); ?>
    <?php endwhile; ?>

</section>

<!-- sans sidebar -->
<?php get_footer(); ?>

```

Vous pouvez activer ce nouveau gabarit depuis le back-office pour une page déterminée. Le modèle par défaut étant celui de page.php) :

Attributs de la page ▲

Parent

(pas de parent) ▼

Modèle

Page sans colonne ▼

Ordre

0

Besoin d'aide ? Utiliser l'onglet Aide présent dans le coin supérieur droit de votre écran.

Autres pages

- **404.php**

Page d'erreur, page introuvable

```
<?php get_header(); ?>
<section id="page">
    <h2 class="titre_article">Erreur 404</h2>
    <p>La page demandée n'existe pas ou a été déplacée</p>
    <?php get_search_form(); ?>
</section>
<?php get_footer(); ?>
```

- **archive.php**

Page d'archives (par mois, année...)

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

<section id="page">
    <?php if ( have_posts() ) : ?>
        <?php while (have_posts()) : the_post(); ?>
            <article class="index">
                <h2><?php the_title(); ?></h2>
                <?php the_post_thumbnail('thumbnail') ?>
                <?php the_excerpt(); ?>
                <p class="clear">
                    <a href="<?php the_permalink(); ?>" > Lire la suite</a>
                </p>
            </article>
        <?php endwhile; ?>
    <?php else : ?>
        <p>Il n'y a pas d'archive pour l'instant.</p>
    <?php endif; ?>
</section>

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

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

- **category.php**

Dans ce cas, liste les articles de la catégorie

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

<section id="page">
    <?php if (have_posts()) : ?>

        <?php while (have_posts()) : the_post(); ?>

            <article class="index">
                <h2><?php the_title(); ?></h2>
                <?php the_post_thumbnail('thumbnail') ?>
                <?php the_excerpt(); ?>

                <small>
                    Posté le <?php the_date(); ?> <!-- Date -->
                    dans <?php the_category(', '); ?>
                </small>

                <p class="clear">
                    <a href="<?php the_permalink(); ?>"> > Lire la suite</a>
                </p>
            </article>
        <?php endwhile; ?>
        <!-- Fin de la boucle -->
        <!-- Sinon, si la boucle ne renvoie rien (pas d'article) afficher ceci : -->
        <?php else : ?>
            <p>Il n'y a pas de post dans cette catégorie</p>
        <?php endif; ?>

    </section>

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

- **search.php**

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

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

<section id="page">
    <?php if ( have_posts() ) : ?>
        <h2 class="titre_article">Résultat de la recherche :</h2>

        <ul>
            <?php while ( have_posts() ) : the_post(); ?>
                <li><a href="<?php the_permalink(); ?>"><?php the_title();?
></a></li>

            <?php endwhile; ?>
        </ul>

        <?php else : ?>
            <h2 class="titre_article">Résultat de la recherche :</h2>
            <p>Désolé, aucun résultat n'a été trouvé pour cette recherche</p>

        <?php endif; ?>

    </section>

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

[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)
);
?>
```

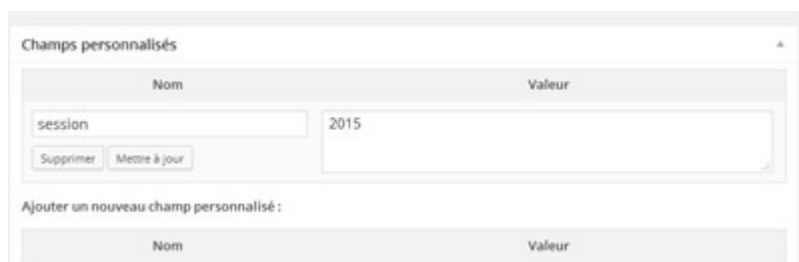
Liste des paramètres possibles pour la fonction WP_Query :

http://codex.wordpress.org/Class_Reference/WP_Query#Parameters

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>
```

Rendu

Session : 2015 |

Aller Plus loin

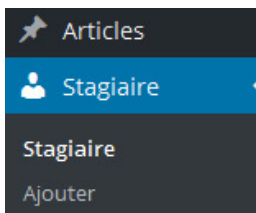
Pour insérer d'autres types de champs : cases à cocher, zone de texte, upload de pdf...

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

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

[TP33_export_bdd_wordpress]