



EXPERT

2^e édition

WordPress

Développez avec PHP

extensions, widgets et thèmes avancés
(théorie, TP, ressources)

Fichiers complémentaires
à télécharger



Laurent DUMOULIN

Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence de l'ouvrage **EI24WOR** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Chapitre 1 Avant-propos

1. Introduction	17
2. À qui s'adresse cet ouvrage ?	18
3. Prérequis	18
4. Objectifs du livre	19

Chapitre 2 Introduction à WordPress

1. Introduction	21
2. Qu'est-ce que WordPress ?	21
2.1 La licence GNU GPL	22
2.2 Les avantages de WordPress	23
2.3 Les inconvénients de WordPress	23
3. Le codex WordPress : guide de référence	24
4. Installer WordPress	26
4.1 Créer la base de données	28
4.2 Transférer les fichiers WordPress	29
4.3 Configurer le site	31
4.4 Les différents identifiants	37
5. L'administration et les menus	37
5.1 Tableau de bord	41
5.1.1 Accueil	41
5.1.2 Mises à jour	42

5.2	Articles	46
5.2.1	Tous les articles	46
5.2.2	Ajouter	47
5.2.3	Les blocs du volet de droite	52
5.2.4	Les blocs masqués	55
5.2.5	Catégories	57
5.2.6	Étiquettes	58
5.3	Médias	60
5.3.1	Bibliothèque	60
5.3.2	Modifier une image	61
5.3.3	Ajouter	63
5.4	Pages	64
5.5	Commentaires	67
5.6	Apparence	68
5.6.1	Thèmes	68
5.6.2	Personnaliser	72
5.6.3	Widgets	79
5.6.4	Menus	81
5.6.5	Éditeur	84
5.7	Extensions	85
5.7.1	Extensions installées	85
5.7.2	Ajouter	86
5.7.3	Éditeur	88
5.8	Utilisateurs	89
5.8.1	Tous les utilisateurs	89
5.8.2	Ajouter	90
5.8.3	Votre profil	91
5.9	Outils	93
5.9.1	Outils disponibles	93
5.9.2	Importer	94
5.9.3	Exporter	94

- 5.10 Réglages 95
 - 5.10.1 Général 95
 - 5.10.2 Écriture 97
 - 5.10.3 Lecture 98
 - 5.10.4 Discussion 99
 - 5.10.5 Médias 102
 - 5.10.6 Permalien 102
 - 5.10.7 Les autres onglets 104
- 6. Conclusion 104

Chapitre 3
WordPress et PHP

- 1. Introduction 105
- 2. La structure des fichiers et dossiers WordPress 106
 - 2.1 Les fichiers et dossiers à la racine du site 106
 - 2.2 Le dossier wp-content 107
- 3. La base de données WordPress 109
- 4. La structure et les fichiers principaux d'un thème 112
 - 4.1 Les fichiers principaux 113
 - 4.2 Les autres fichiers 116
 - 4.3 Les dossiers éventuellement présents
à la racine d'un thème 117
- 5. Les fonctions d'inclusion 118
- 6. Les marqueurs conditionnels 119
- 7. La boucle et ses fonctions 122
 - 7.1 Les principales fonctions d'affichage dans une boucle 123
 - 7.2 Les principales fonctions pour récupérer
des variables dans une boucle 127
- 8. Les fonctions pour les textes dans les fichiers PHP 130
- 9. Les chemins dans les URL 132

10. La fonction <code>bloginfo()</code>	133
11. La fonction <code>wp_nav_menu()</code>	136
12. Les hooks (filtres et actions)	142
12.1 Les actions.	142
12.2 Les filtres	144
12.3 Ajouter des filtres et des actions dans une classe	145
12.4 Supprimer les filtres et les actions.	146
12.5 Tester les filtres et les actions	146
12.6 L'ordre d'exécution des actions	147
13. Les globales	150
13.1 Les globales dans les boucles	150
13.2 Les globales pour détecter le navigateur	152
13.3 Les globales pour les serveurs.	152
13.4 Les globales pour les versions de WordPress	152
13.5 Les globales diverses	153
14. Les classes de WordPress	153
15. La classe <code>WP_Query</code> et les requêtes sur le contenu	154
15.1 Les méthodes.	155
15.2 Les arguments	156
15.2.1 Trier par catégories	156
15.2.2 Trier par étiquettes	157
15.2.3 Trier par auteurs	157
15.2.4 Trier par pages et par articles	157
15.2.5 Choisir l'ordre d'affichage	158
15.2.6 Plus de filtres	158
15.2.7 La concaténation d'arguments	159
15.3 Plusieurs boucles sur la même page.	159
15.4 D'autres fonctions pour faire des requêtes	159
15.5 <code>WP_Query</code> et les fonctions	161
16. La classe <code>wpdb</code> et les requêtes au format SQL	162
16.1 Les méthodes utiles	162
16.2 Les principales méthodes de requêtes et leurs arguments	163

- 16.3 Créer des tables pour les plugins
avec la fonction dbdelta() 167
- 17. La classe WP_rewrite et la réécriture d'URL 168
 - 17.1 Passer une variable dans une URL
avec la fonction add_rewrite_tag() 168
 - 17.2 Réécrire l'URL avec la fonction add_rewrite_rule() 170
 - 17.3 Régénérer les règles d'écriture 170
- 18. Les shortcodes 171
 - 18.1 Créer des shortcodes 172
 - 18.2 Ajouter des arguments aux shortcodes 173
 - 18.3 Récupérer du contenu 174
 - 18.4 Utiliser les shortcodes dans des fichiers PHP 175
- 19. Conclusion 175

Chapitre 4

Les thèmes enfants

- 1. Qu'est-ce qu'un thème enfant ? 177
- 2. Créer et utiliser un thème enfant 178
 - 2.1 Créer un dossier pour le thème enfant 178
 - 2.2 Le fichier style.css 178
 - 2.3 Les modèles de page 180
 - 2.4 Le fichier functions.php 180
 - 2.5 Les chemins des dossiers enfants 181
 - 2.6 Charger un dossier languages dans le thème enfant 182
- 3. Des thèmes enfants pour les extensions 182

Chapitre 5**Personnaliser le site avec le fichier `functions.php`**

1. Introduction	183
2. Créer des fonctions simples en PHP	184
2.1 Afficher une image	184
2.2 Afficher une publicité	185
3. Créer une fonction fil d'Ariane simple	186
4. Créer une fonction pour afficher des articles d'une catégorie avec l'objet <code>WP_Query</code>	190
4.1 Une fonction simple avec <code>get_posts()</code>	192
4.2 La même fonction avec l'objet <code>WP_Query</code>	194
5. Ajouter un lien « Lire la suite » aux résumés	195
6. Ajouter des feuilles de style et des scripts	196
6.1 Les fonctions d'ajout	197
6.2 Des scripts mis à disposition	199
6.3 Inclure les styles et les scripts avec des hooks	199
6.4 Créer la fonction dans le fichier <code>functions.php</code>	200
7. Créer des emplacements pour les menus	200
8. Créer des sidebars et des zones pour les sidebars	202
9. Customiser un thème avec <code>add_theme_support()</code>	207
9.1 Les formats	208
9.1.1 Ajouter un format	208
9.1.2 Fonctionnement	209
9.1.3 Utilisation	210
9.2 Les photos vignettes (thumbnails)	211
9.2.1 Définir une taille de vignette	212
9.2.2 Créer une seconde vignette	212
9.3 Les arrière-plans (background)	213
9.4 Les en-têtes (headers)	217
9.5 Le logo	219
10. Ajouter d'autres types d'articles	221

11. Ajouter des taxonomies spécifiques 224
12. Conclusion 226

Chapitre 6
Les champs personnalisés

1. Qu'est-ce qu'un champ personnalisé ? 227
2. Afficher les champs personnalisés avec the_meta() 228
3. Afficher les champs personnalisés avec get_post_meta() 229
4. Automatiser les champs personnalisés avec add_post_meta() 230
5. La table wp_postmeta 233
6. L'extension ACF (Advanced Custom Fields) 234
 6.1 Créer des groupes de champs. 238
 6.2 Récupérer et afficher la valeur des champs
 personnalisés dans les modèles de page. 240

Chapitre 7
Les modèles de page

1. Qu'est-ce qu'un modèle de page ? 243
2. Créer des modèles de page pour le bloc Attributs de la page 244
3. Créer d'autres modèles de page 246

Chapitre 8
Créer un thème et des fonctionnalités avancés

1. Introduction 249
2. Créer le thème. 251
3. Customiser le thème de base 254
 3.1 Nettoyer le thème et préparer la base 254
 3.2 Modifier la page d'accueil. 255

4. Ajouter un logo avec la fonction <code>add_theme_support()</code>	257
5. Ajouter un onglet Options du thème	260
5.1 La fonction <code>add_theme_pages()</code>	260
5.2 Créer l'onglet.	261
5.3 Les bonnes pratiques	262
5.4 Appeler le fichier dans <code>functions.php</code>	263
6. Mettre en place la page d'options	264
7. Ajouter une feuille de style CSS et un script JavaScript	265
8. Ajouter une image bannière.	267
8.1 Créer une méthode comprenant un champ input de type file	267
8.2 Créer le code HTML	268
8.3 Vérifier le fichier téléchargé	269
8.4 Transférer l'image avec la fonction <code>wp_handle_upload()</code>	271
8.5 Utiliser les options pour enregistrer en base de données	271
8.6 Afficher l'image sur le site	274
8.7 Insérer une image par défaut dès l'activation du thème	274
8.8 Créer un aperçu dans l'administration	277
9. Créer un bouton qui remet les options par défaut	278
10. Choisir une couleur avec Iris (color picker).	280
10.1 Ajouter le script JavaScript et le CSS Iris, palette de WordPress	280
10.2 Ajouter la couleur par défaut	281
10.3 Ajouter le formulaire	281
10.4 Ajouter le code JavaScript pour faire fonctionner Iris	282
10.5 Enregistrer les informations.	284
10.6 Afficher le résultat	284
11. Ajouter des textes sur la page d'accueil	285
11.1 Ajouter les codes aux méthodes de la classe	286
11.2 Ajouter un éditeur WYSIWYG avec la fonction <code>wp_editor()</code> .	287
11.3 Enregistrer les informations.	290
11.4 Afficher le résultat	290
11.5 Ajouter du code CSS pour rendre le thème responsive	291

12. Ajouter une sidebar dans le footer pour la page d'accueil 298
13. En résumé 305
14. Conclusion 307

Chapitre 9
Les extensions et les widgets

1. Introduction 309
2. Les widgets 312
3. Les extensions utiles 320
4. Les extensions diaporama 335
5. Les extensions pour l'administration 341
6. Les grosses extensions 357
7. Les extensions pour les développeurs 364

Chapitre 10
Créer une extension simple en PHP

1. Introduction 367
2. Préparer les éléments 367
3. Afficher l'extension dans l'administration 368
4. Créer le code PHP de base 371
5. Créer un shortcode 373
6. Créer un onglet dans le menu d'administration 374
6.1 Afficher un onglet principal 375
6.2 Ajouter des fonctions pour les sous-menus
d'un onglet principal de WordPress 376
6.3 Ajouter des fonctions pour les sous-menus
d'un onglet principal personnalisé 377
6.4 Créer l'onglet 377

- 7. Créer la même extension en objet 378
- 8. Le dossier mu-plugins 381
- 9. Conclusion 382

Chapitre 11

Créer une extension widget en PHP

- 1. Introduction 383
- 2. Configurer l'extension 384
- 3. La classe WP_Widget 385
- 4. Configurer le widget 387
- 5. Enregistrer le widget avec la fonction register_widget() 388
- 6. Afficher le widget grâce à un hook 390
- 7. Créer le formulaire du widget 390
 - 7.1 Ajouter les champs HTML 390
 - 7.2 Ajouter les méthodes get_field_id() et get_field_name() 391
 - 7.3 Ajouter des paramètres par défaut 392
- 8. Enregistrer et mettre à jour les options du widget 395
- 9. Afficher le widget dans la sidebar 396

Chapitre 12

Créer une extension avec PHP/MySQL

- 1. Introduction 399
- 2. Préparer les éléments 401
- 3. Construire la classe 402
- 4. Créer et supprimer la table 404
 - 4.1 Créer la table 404
 - 4.2 Supprimer la table lors de la désactivation de l'extension 407
 - 4.3 Supprimer la table lors de la suppression de l'extension 409

- 5. Ajouter le sous-menu de l'extension et la page de configuration . . 411
 - 5.1 Ajouter le sous-menu 411
 - 5.2 Afficher la page d'administration 412
 - 5.3 Créer le formulaire pour enregistrer les cartes 412
- 6. Ajouter les fichiers JavaScript et CSS 414
 - 6.1 Ajouter les fichiers 414
 - 6.2 Faire appel au hook d'action load-(page). 415
 - 6.3 Vérifier les champs du formulaire avec jQuery 416
- 7. Enregistrer les cartes 418
 - 7.1 Récupérer et vérifier les variables 418
 - 7.2 Enregistrer la carte avec une requête INSERT 419
 - 7.3 Rediriger après l'insertion. 421
- 8. Créer un menu avec les différentes cartes. 422
 - 8.1 Récupérer toutes les cartes avec une requête SELECT 422
 - 8.2 Créer le menu avec une boucle 423
- 9. Afficher les pages cartes 424
 - 9.1 Mettre en place un second template 424
 - 9.2 Ajouter le menu au second template. 425
- 10. Afficher l'aperçu d'une carte et ajouter son titre 426
 - 10.1 Récupérer l'id de la carte avec une requête SELECT 426
 - 10.2 Ajouter le code JavaScript de l'API Google Maps 427
- 11. Mettre à jour les cartes. 429
 - 11.1 Créer le formulaire pour la mise à jour des cartes 430
 - 11.2 Récupérer et vérifier les variables 431
 - 11.3 Mettre à jour les informations avec une requête UPDATE . . . 432
 - 11.4 Rediriger l'utilisateur vers la carte après la mise à jour 434
- 12. Supprimer les cartes 435
 - 12.1 Créer un bouton pour supprimer la carte 435
 - 12.2 Récupérer et vérifier les variables 436
 - 12.3 Supprimer la carte avec une requête DELETE 436
 - 12.4 Rediriger l'utilisateur après la suppression 437

- 13. Afficher la carte sur le site côté utilisateur avec un shortcode 438
 - 13.1 Ajouter l'appel JavaScript à l'API Google Maps 438
 - 13.2 Créer des shortcodes pour chaque carte 438
 - 13.3 Afficher le code sur la page de chaque carte 439

Chapitre 13

Créer une extension avec Ajax et JSON

- 1. Introduction 443
- 2. Préparer les éléments 444
 - 2.1 Configurer l'extension 444
 - 2.2 Construire la classe 445
 - 2.3 Créer la table 446
 - 2.4 Supprimer la table lors de la suppression de l'extension 447
 - 2.5 Ajouter un sous-menu et la page de configuration 448
 - 2.6 Ajouter les fichiers JavaScript et CSS 449
- 3. Créer le shortcode et l'ajouter au site 451
 - 3.1 Créer le shortcode 451
 - 3.2 Insérer le shortcode 452
 - 3.3 Afficher le code dans l'administration pour les utilisateurs . . . 454
- 4. Vérifier les champs du formulaire en jQuery 455
- 5. Effectuer la requête Ajax 457
 - 5.1 WordPress et Ajax 457
 - 5.2 Transmettre des variables PHP au fichier JavaScript 457
 - 5.3 Générer une clé de sécurité 458
 - 5.4 La fonction Ajax 458
- 6. Récupérer les données et enregistrer une adresse e-mail 460
 - 6.1 Les hooks d'action wp_ajax_(action)
et wp_ajax_nopriv_(action) 460
 - 6.2 Vérifier la clé de sécurité 461
 - 6.3 Enregistrer l'adresse e-mail avec une requête d'insertion 461
 - 6.4 Vérifier les doublons avant l'insertion 462

- 6.5 Transmettre la réponse à la requête Ajax 463
- 6.6 Ajouter une image de chargement (loader). 465
- 7. Afficher et gérer les adresses e-mail dans l'administration. 466
 - 7.1 Ajouter une méthode listant les adresses e-mail. 466
 - 7.2 Créer un tableau avec toutes les adresses e-mail 466
 - 7.3 Créer une liste d'adresses e-mail
pour la copier-coller facilement 470
- 8. Supprimer des adresses e-mail 472
 - 8.1 Créer un événement JavaScript sur le bouton de suppression. 472
 - 8.2 Transmettre les variables au fichier JavaScript. 473
 - 8.3 Créer la requête Ajax 474
 - 8.4 Créer le hook et la méthode pour la requête Ajax 474
 - 8.5 Créer la méthode de suppression. 475
 - 8.6 Supprimer et transmettre la réponse à la requête Ajax 475
 - 8.7 Mettre à jour la liste et vérifier que le tableau n'est pas vide. . 477
 - 8.8 Ajouter une image de chargement (loader). 477

Chapitre 14
Soumettre une extension à WordPress

- 1. Introduction 479
- 2. Les prérequis WordPress. 479
 - 2.1 La licence. 480
 - 2.2 Le fichier readme.txt. 481
- 3. Soumettre l'extension. 482
 - 3.1 Créer un compte sur wordpress.org 482
 - 3.2 Envoyer l'extension. 482
 - 3.3 Le dossier SVN 484

Chapitre 15

Traduire le thème et les extensions

1. Introduction 485
2. WordPress et les langues 486
 - 2.1 Le dossier wp-content/languages 486
 - 2.2 Les fichiers de langue du dossier wp-content/languages 486
 - 2.3 D'autres dossiers languages 487
 - 2.4 Le fonctionnement 488
3. Utiliser le logiciel Poedit 489
 - 3.1 Télécharger et installer le logiciel 490
 - 3.2 Créer le fichier de langue 491
 - 3.3 Traduire 494
4. Utiliser les fichiers dans un thème ou une extension 497
5. Traduire les nouvelles chaînes 498
6. Traduire les pages et les articles 498

Chapitre 16

Optimiser et sécuriser un site

1. Introduction 499
2. Optimiser un site 500
3. Sécuriser un site 504

Chapitre 17

Sauvegarder un site

1. Introduction 509
2. Télécharger les fichiers par FTP 510
3. Sauvegarder la base de données 510
4. Rétablir le site en cas de crash 512

Chapitre 18
Mettre en ligne ou migrer son site

- 1. Introduction515
- 2. Importer la base de données.516
- 3. Modifier les URL avec des requêtes SQL.517
- 4. Transférer les fichiers et modifier le fichier wp-config.php.519

Chapitre 19
WordPress MU

- 1. Introduction523
- 2. Qu'est-ce que WordPress MU ?523
- 3. Transformer un site en WordPress MU524
 - 3.1 Activer le multisite525
 - 3.2 Configurer et choisir les URL.525
 - 3.3 Modifier les fichiers wp-config.php et .htaccess.526
- 4. Gérer un réseau527
 - 4.1 Dans le panel d'administration du réseau528
 - 4.2 Dans le panel d'administration de chaque site534

Annexes

- 1. Les liens utiles537
- 2. Glossaire539
 - 2.1 Les langages de programmation de WordPress539
 - 2.2 Les termes WordPress, du Web et de programmation540

- Index547

Chapitre 3

WordPress et PHP

1. Introduction

WordPress est un CMS entièrement conçu en PHP programmation orientée objet, il est donc normal que WordPress ait développé ses propres API, classes, méthodes, fonctions... pour simplifier la vie des développeurs.

Ce chapitre décrit la place prépondérante de PHP dans WordPress, la structure et les fonctions récurrentes de WordPress. Il sert de référence pour l'ensemble du livre, vous pouvez très bien le sauter et vous y référer par la suite lors de la personnalisation du fichier `functions.php`, de la création de thèmes avancés ou lors de la création d'extensions...

Ce chapitre est un condensé des possibilités qu'offre WordPress, il va à l'essentiel. Alors n'hésitez pas à consulter la documentation officielle, le codex ou de nombreux tutoriels gratuits, présents sur la Toile, pour approfondir vos connaissances sur un sujet précis.

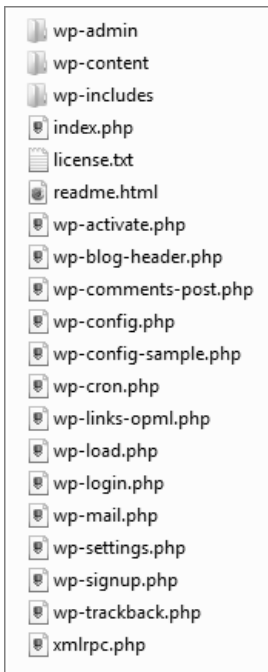
Tous les mots-clés et termes techniques de chaque chapitre vous permettent de mener votre propre enquête sur le Net. Avec une communauté très active, il serait dommage de ne pas profiter de l'expérience de toutes ces personnes, prêtes à vous aider gratuitement. Vous trouverez énormément de ressources, avec de nombreux exemples de code.

2. La structure des fichiers et dossiers WordPress

Connaître l'organisation des fichiers dans WordPress est important. La structure est toujours la même, pour un petit site ou pour un gros site : le core ou cœur de WordPress ne change pas.

2.1 Les fichiers et dossiers à la racine du site

Les sites WordPress ont tous la même structure. À la racine, vous trouverez les dossiers wp-admin, wp-content, wp-includes et les fichiers WordPress.



Fichiers et dossiers à la racine du site WordPress.

Il ne faut pas modifier ou supprimer certains fichiers ou dossiers natifs de WordPress, en l'occurrence ceux qui font partie du core, le site risquerait de ne plus fonctionner. De plus, la mise à jour de WordPress écrase toutes les modifications.

Le dossier **wp-admin** contient l'ensemble des fichiers concernant l'administration du site. Il ne faut pas modifier des fichiers dans ce dossier.

Le dossier **wp-content** contient l'ensemble des fichiers des thèmes, des extensions et des médias. C'est dans ce dossier que s'effectuent les modifications et personnalisations du site.

Le dossier **wp-includes** contient l'ensemble des fichiers du core de WordPress (classes, fonctions, scripts...). Il ne faut pas modifier des fichiers dans ce dossier.

Le fichier **wp-config.php** est un fichier généré lors de l'installation de WordPress, à partir du fichier wp-config-sample.php. Les informations qui ont servi à la configuration du site se trouvent dans ce fichier : le nom de la base de données, le mot de passe, l'URL de la base de données, le préfixe des tables, les clés de sécurité, les langues... Ce fichier est modifié lors du changement d'hébergeur, lors de l'optimisation du site, du débogage, lors d'un changement de langue... La suppression du fichier wp-config.php réaffiche la page de configuration.

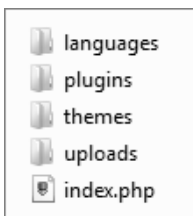
Les autres fichiers .php servent au core de WordPress et il ne faut absolument pas les modifier ou les supprimer. Cependant, vous pouvez trouver à la racine du site des fichiers qui ont été générés par une extension, certains pourront être modifiés : .htaccess, robots.txt ou sitemap.xml...

2.2 Le dossier wp-content

Le dossier **wp-content** est le dossier servant à la création de thèmes, d'extensions, de fonctions. Il contient les dossiers **languages**, **plugins**, **themes** et le fichier **index.php**.

D'autres dossiers apparaissent lors d'ajout d'images, comme le dossier **uploads**, et lors de mises à jour, tel le dossier **upgrade**.

D'autres dossiers peuvent encore apparaître selon l'installation de certaines extensions, comme le dossier **cache**.



Dossiers et fichiers du répertoire wp-content, lors de l'installation de WordPress.

Le dossier **languages** : contient les fichiers de langue de WordPress. L'installation manuelle s'effectue en téléchargeant par FTP les fichiers de langue dans ce dossier. Les fichiers de langue ont les extensions .mo et .po/.pot.

Le dossier **plugins** : contient l'ensemble des extensions installées. De base, le dossier contient deux extensions : Akismet et Hello Dolly. L'installation manuelle s'effectue en téléchargeant par FTP le dossier contenant les fichiers de l'extension dans le dossier plugins.

Le dossier **themes** : contient l'ensemble des thèmes installés. De base, il y a trois thèmes : Twenty Sixteen, Twenty Fifteen et Twenty Fourteen. L'installation manuelle s'effectue en téléchargeant par FTP le dossier contenant les fichiers et dossiers du thème dans le dossier themes.

Le dossier **uploads** : contient l'ensemble des médias téléchargés. Pour les images, WordPress crée trois formats d'image et les classe selon les paramètres entrés dans l'administration. Par défaut WordPress classe les médias dans des dossiers ayant pour nom l'année, avec à l'intérieur des dossiers ayant pour nom les mois en chiffres (01, 02, ... 12). Certaines extensions y créent parfois des dossiers pour y stocker divers fichiers.

Le dossier **upgrade** : accueille les fichiers compressés des mises à jour de thèmes ou d'extensions. Avant leur décompression, WordPress les télécharge dans ce dossier. Si la mise à jour se passe mal, vous retrouvez le fichier compressé dans ce dossier. Ce dossier apparaîtra lorsque vous effectuerez une mise à jour.

Le fichier **index.php** : sert uniquement à la sécurité du dossier wp-content. Ce fichier vide permet d'afficher une page blanche au lieu de l'arborescence du site en cas de dysfonctionnement. Vous le retrouvez dans différents dossiers.

3. La base de données WordPress

WordPress installe douze tables dans la base de données. Pour le constater, connectez-vous à votre base de données. C'est là où sont stockées toutes les informations de votre site (contenu des articles, des pages, des catégories, des étiquettes, etc., la configuration de WordPress, des extensions, des thèmes, etc., les informations sur les utilisateurs...).

Dans le livre, les exemples sont illustrés avec le préfixe wp45 (par défaut, WordPress utilise le préfixe wp_). Normalement, le préfixe est choisi lors de l'installation, il se peut donc qu'il soit différent si vous l'avez changé pour des questions de sécurité. Pensez à changer le préfixe lors de l'utilisation des exemples.

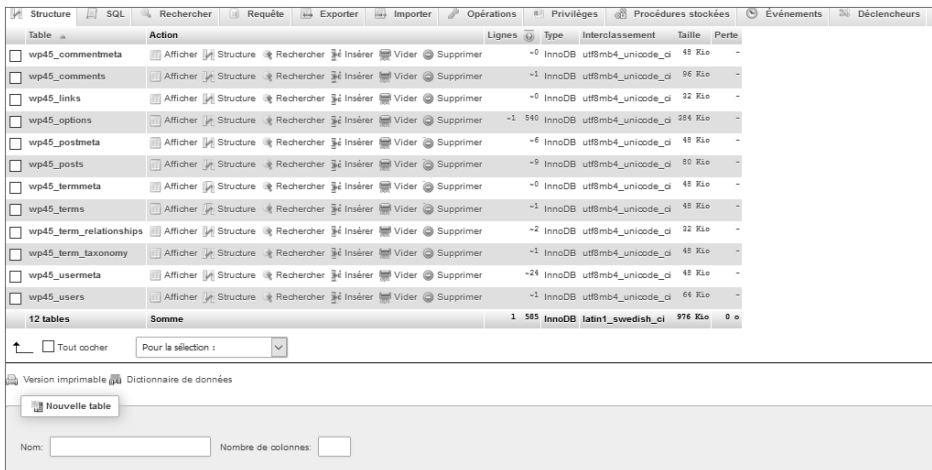


Table	Action	Lignes	Type	Interclassement	Taille	Perte
<input type="checkbox"/> wp45_commentmeta	Afficher Structure Rechercher Insérer Vidier Supprimer	-0	InnoDB	utf8mb4_unicode_ci	48 Kio	-
<input type="checkbox"/> wp45_comments	Afficher Structure Rechercher Insérer Vidier Supprimer	-1	InnoDB	utf8mb4_unicode_ci	96 Kio	-
<input type="checkbox"/> wp45_links	Afficher Structure Rechercher Insérer Vidier Supprimer	-0	InnoDB	utf8mb4_unicode_ci	22 Kio	-
<input type="checkbox"/> wp45_options	Afficher Structure Rechercher Insérer Vidier Supprimer	-1 540	InnoDB	utf8mb4_unicode_ci	284 Kio	-
<input type="checkbox"/> wp45_postmeta	Afficher Structure Rechercher Insérer Vidier Supprimer	-6	InnoDB	utf8mb4_unicode_ci	48 Kio	-
<input type="checkbox"/> wp45_posts	Afficher Structure Rechercher Insérer Vidier Supprimer	-9	InnoDB	utf8mb4_unicode_ci	80 Kio	-
<input type="checkbox"/> wp45_termmeta	Afficher Structure Rechercher Insérer Vidier Supprimer	-0	InnoDB	utf8mb4_unicode_ci	48 Kio	-
<input type="checkbox"/> wp45_terms	Afficher Structure Rechercher Insérer Vidier Supprimer	-1	InnoDB	utf8mb4_unicode_ci	48 Kio	-
<input type="checkbox"/> wp45_term_relationships	Afficher Structure Rechercher Insérer Vidier Supprimer	-2	InnoDB	utf8mb4_unicode_ci	22 Kio	-
<input type="checkbox"/> wp45_term_taxonomy	Afficher Structure Rechercher Insérer Vidier Supprimer	-1	InnoDB	utf8mb4_unicode_ci	48 Kio	-
<input type="checkbox"/> wp45_usermeta	Afficher Structure Rechercher Insérer Vidier Supprimer	-24	InnoDB	utf8mb4_unicode_ci	48 Kio	-
<input type="checkbox"/> wp45_users	Afficher Structure Rechercher Insérer Vidier Supprimer	-1	InnoDB	utf8mb4_unicode_ci	64 Kio	-
12 tables	Somme	1 585	InnoDB	latin1_swedish_ci	976 Kio	0 o

↑ Tout cacher Pour la sélection :

Version imprimable Dictionnaire de données

Nom: Nombre de colonnes:

Tables en base de données lors de l'installation de WordPress, dans phpMyAdmin.

D'autres tables apparaissent selon les thèmes, les extensions et l'utilisation du mode multisite.

Remarque

Référence au codex : http://codex.wordpress.org/Database_Description

Voici les tables natives de WordPress :

- La table **wp_commentmeta** contient des informations complémentaires concernant les commentaires. Cette table est utilisée par l'extension **Akismet**, devenue extension native de WordPress, indispensable pour éviter de se faire spammer. Vous la retrouvez dans l'administration, sous l'onglet **Extensions**.
- La table **wp_comments** contient tous les commentaires des articles et des pages.
- La table **wp_links** (facultative) regroupe tous les liens enregistrés via l'onglet **Liens** de l'administration. L'onglet **Liens** n'existe plus depuis la version 3.5, et nécessite l'utilisation de l'extension Link Manager, mais la table est toujours présente pour que les personnes qui l'utilisaient avant ne perdent pas leurs données sur les versions récentes de WordPress. Vous pouvez également activer le gestionnaire de liens en ajoutant le code suivant au fichier `functions.php` :

```
add_filter( 'pre_option_link_manager_enabled',  
    '__return_true' );
```

- La table **wp_options** contient les configurations générales du site (renseignées lors de l'installation du site) et des extensions... parmi les plus importantes ! Lors de la création de thèmes avancés, cette table est également utilisée pour y stocker des informations, grâce à des fonctions de WordPress.
- La table **wp_postmeta** contient les informations complémentaires liées aux articles ou aux pages. Cette table est directement liée à la table `wp_posts`.
- La table **wp_posts** est la plus importante, elle contient tout le contenu du site. Vous y retrouvez toutes les informations concernant les articles, les pages, les photos, les fichiers PDF et autres médias, ou les produits dans le cas d'extensions e-commerce.
- La table **wp_termmeta** contient les metadonnées ajoutées aux catégories. Cette nouvelle table est disponible depuis la version 4.4.
- La table **wp_terms** contient les catégories et les étiquettes. Cette table est directement liée aux tables `wp_term_relationships` et `wp_term_taxonomy`.