

Editions ENI

PHP 7

Développez un site web dynamique et interactif (2^e édition)

Collection
Ressources Informatiques

Extrait

Chapitre 6

Gérer les formulaires et les liens

1. Vue d'ensemble

1.1 Introduction

Dans les sites web dynamiques, il est très souvent nécessaire d'interagir avec l'utilisateur.

En HTML, il existe principalement deux méthodes pour interagir avec un utilisateur :

- les liens (balise `<a>`) ;
- les formulaires (balise `<form>`).

Des scripts PHP peuvent être utilisés pour traiter le clic de l'utilisateur sur un lien ou la saisie de l'utilisateur dans un formulaire.

1.2 Les liens

Le lien est la technique de base qui permet à un utilisateur de naviguer entre les différentes pages d'un site.

Un lien HTML est défini entre les balises `<a>` et ``.

Syntaxe simplifiée

```
<a  
  [ href="url" ]  
  [ id="identifiant_lien" ]  
  [ target="cible" ]
```

```
>
...
</a>
```

Les attributs de la balise `<a>` sont les suivantes :

- `href` URL (*Uniform Resource Locator*) relative ou absolue qui est appelée par le lien.
- `id` Identifiant du lien. Si la page HTML contient plusieurs liens, l'identifiant permet de les différencier. En ce qui nous concerne, cet identifiant ne présente pas d'intérêt car il n'est pas récupéré dans le script de traitement du lien. Par contre, il peut être utilisé côté client, en JavaScript par exemple.
- `target` Cible (par exemple une autre fenêtre) dans laquelle ouvrir l'URL cible. Par défaut, l'URL cible s'affiche dans la même fenêtre.

L'URL peut contenir des paramètres qui permettent de passer des informations d'une page à une autre.

Syntaxe

```
url_classique?nom=valeur[&...]
```

Le point d'interrogation (?) introduit la liste des paramètres de l'URL séparés par le caractère esperluette (&) ; chaque paramètre est constitué par un couple nom/valeur sous la forme `nom=valeur` :

```
www.monsite.com/info/accueil.php?prenom=Olivier
chercher.php?prenom=Olivier&nom=HEURTEL
```

Exemple

– Script `page1.php`

```
<?php
// Initialisation d'une variable.
$nom='Olivier';
?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr">
  <head><meta charset="utf-8" /><title>Page 1</title></head>
  <body>
    <div>
      <!-- lien vers la page 2 en passant la valeur de $nom
           dans l'URL -->
      <a href="page2.php?nom=<?=$nom ?>">Page 2</a>
    </div>
  </body>
</html>
```

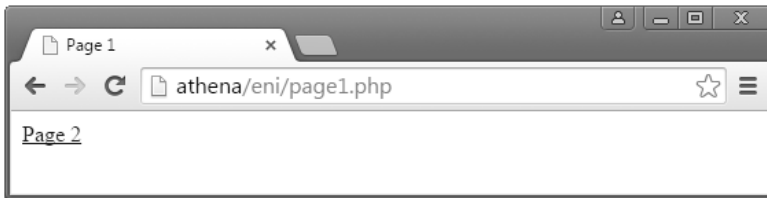
Chapitre 6

- Source de la page dans le navigateur

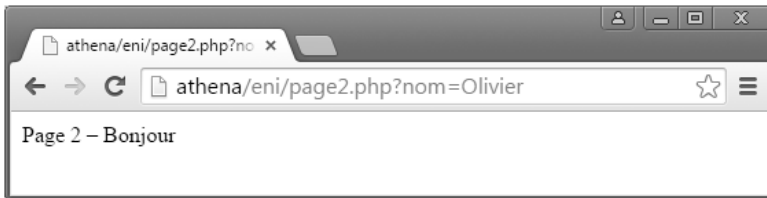
```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr">
  <head><meta charset="utf-8" /><title>Page 1</title></head>
  <body>
    <div>
      <!-- lien vers la page 2 en passant la valeur de $nom
           dans l'URL -->
      <a href="page2.php?nom=Olivier">Page 2</a>
    </div>
  </body>
</html>
```

Résultat

- Affichage de la page 1 :



- Résultat du clic sur le lien :



Pour l'instant, aucun nom n'est affiché dans la deuxième page. La variable `$nom` définie dans le script `page1.php` n'est pas disponible dans le script `page2.php` (voir le chapitre Introduction à PHP, section Les bases du langage PHP - Variables - Portée et durée de vie). De plus, notre script ne contient aucune instruction permettant de récupérer les données passées dans l'URL ; nous verrons comment procéder à la section Récupérer les données d'une URL ou d'un formulaire.

1.3 Les formulaires

1.3.1 Petit rappel sur les formulaires

Le formulaire est un outil de base indispensable pour les sites web dynamiques puisqu'il permet à l'utilisateur de saisir des informations et donc d'interagir avec le site.

Un formulaire HTML est défini entre les balises `<form>` et `</form>`.

Syntaxe simplifiée

```
<form
  [ action="url_de_traitement" ]
  [ method="GET"|"POST" ]
  [ id="identifiant_formulaire" ]
  [ target="cible" ]>
...
</form>
```

Les attributs de la balise `<form>` sont les suivants :

- | | |
|---------------------|--|
| <code>action</code> | URL (<i>Uniform Resource Locator</i>) relative ou absolue qui va traiter le formulaire (en ce qui nous concerne, un script PHP). Cet attribut est obligatoire pour se conformer à la recommandation XHTML stricte. |
| <code>method</code> | Mode de transmission vers le serveur des informations saisies dans le formulaire.
GET (valeur par défaut) : les données du formulaire sont transmises dans l'URL.
POST : les données du formulaire sont transmises dans le corps de la requête. |
| <code>id</code> | Identifiant du formulaire. Si la page HTML contient plusieurs formulaires, l'identifiant permet de les différencier. En ce qui nous concerne, cet identifiant ne présente pas d'intérêt car il n'est pas récupéré dans le script de traitement du formulaire. Par contre, il peut être utilisé côté client, en JavaScript par exemple. |
| <code>target</code> | Cible (par exemple une autre fenêtre) dans laquelle ouvrir l'URL cible. Par défaut, l'URL cible s'affiche dans la même fenêtre. |

Entre les balises `<form>` et `</form>`, il est possible de placer des balises `<input>`, `<select>` ou `<textarea>` pour définir des zones de saisie.

Exemple (formulaire HTML complet)

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr">
  <head>
    <meta charset="utf-8" />
    <title>Saisie</title>
  </head>
  <body>
    <form action="saisie.php" method="post">
      <div>
        Nom :
        <input type="text" name="nom" value=""
          size="20" maxlength="20" />
        Mot de passe :
        <input type="password" name="mot_de_passe" value=""
          size="20" maxlength="20" />
        <br />Sexe :
        <input type="radio" name="sexe" value="M" />Masculin
        <input type="radio" name="sexe" value="F" />Féminin
        <input type="radio" name="sexe" value="?"
          checked="checked" />Ne sait pas
        <br />Photo :
        <input type="file" name="photo" size="50" />
        <br />Couleurs préférées :
        <input type="checkbox" name="couleurs[bleu]" />Bleu
        <input type="checkbox" name="couleurs[blanc]" />Blanc
        <input type="checkbox" name="couleurs[rouge]" />Rouge
        <input type="checkbox" name="couleurs[pas]"
          checked="checked" />Ne sait pas
        <br />Langue :
        <select name="langue">
          <option value="E">Espagnol</option>
          <option value="F" selected="selected" >Français</option>
          <option value="I">Italien</option>
        </select>
        <br />Fruits préférés :<br />
        <select name="fruits[]" multiple="multiple" size="8">
          <option value="A">Abricots</option>
          <option value="C">Cerises</option>
          <option value="F">Fraises</option>
          <option value="P">Pêches</option>
          <option value="?" selected="selected">
            Ne sait pas</option>
        </select>
        <br />Commentaire :<br />
        <textarea name="commentaire" rows="4" cols="50"></textarea>
        <br />
        <input type="hidden" name="invisible" value="123" /><br />
      </div>
    </form>
  </body>
</html>
```

```
<input type="submit" name="soumettre" value="OK" />
<input type="image" name="valider" alt="valider"
src="valider.gif" />
<input type="reset" name="effacer" value="Effacer" />
<input type="button" name="action" value="Ne fait rien" />
</div>
</form>
</body>
</html>
```

Résultat



Nom : Mot de passe :

Sexe : Masculin Féminin Ne sait pas

Photo :

Couleurs préférées : Bleu Blanc Rouge Ne sait pas

Langue : ▼

Fruits préférés :

- ▼
-
-
-
-

Commentaire :

PHP peut intervenir à deux endroits par rapport au formulaire :

- Pour la construction du formulaire, si ce dernier doit contenir des informations dynamiques.
- Pour le traitement du formulaire (c'est-à-dire des données saisies par l'utilisateur dans le formulaire).

Editions ENI

WordPress

Développez avec PHP

extensions, widgets et thèmes avancés
(théorie, TP, ressources)
(3^e édition)

Collection
Expert IT

Extrait

Chapitre 3

WordPress et PHP

1. Introduction

WordPress étant un CMS entièrement conçu en PHP, langage orienté 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.

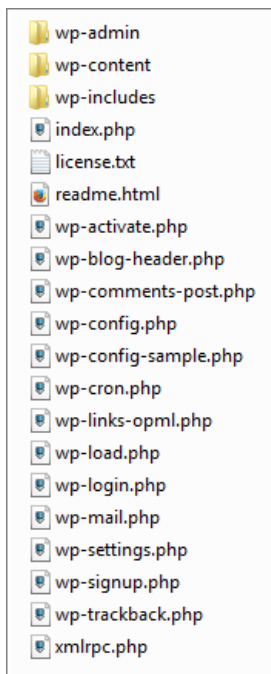
Tous les mots-clés et termes techniques de chaque chapitre vous permettent de mener votre propre enquête sur le Net. Avec une communauté très active, il serait dommage de ne pas profiter de l'expérience de toutes ces personnes, prêtes à vous aider gratuitement. Vous trouverez énormément de ressources, 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, des langues... C'est dans ce dossier que s'effectuent les modifications et personnalisations principales 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, la configuration de la langue principale du site... 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.

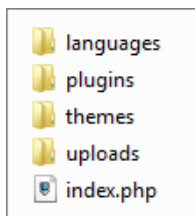
Les autres fichiers .php servent au core de WordPress et il ne faut absolument pas les modifier ou les supprimer. Cependant, vous pouvez trouver à la racine du site des fichiers qui ont été générés par une extension, par WordPress ou ajoutés par vous-même. Certains fichiers pourront donc être modifiés : .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, des thèmes et des extensions. L'installation manuelle s'effectue en téléchargeant par FTP les fichiers de langue dans ce dossier et les sous-dossiers concernés. 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. L'ensemble des plugins s'affichent dans l'administration, onglet **Extensions**.

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

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, dans l'onglet **Réglages - Médias**. 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. L'ensemble des médias s'affichent dans l'administration, onglet **Médias**.

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 msw_ (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.

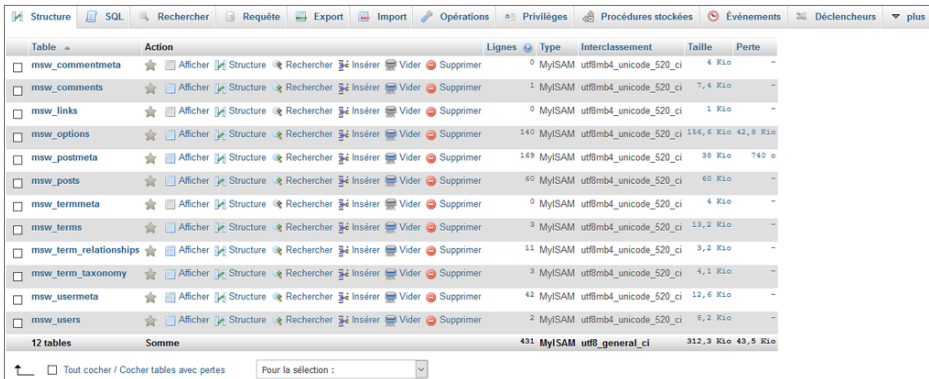


Table	Action	Lignes	Type	Interclassement	Taille	Perte
msw_commentmeta	Afficher Structure Rechercher Insérer Vider Supprimer	0	MyISAM	utf8mb4_unicode_520_ci	4 KLo	-
msw_comments	Afficher Structure Rechercher Insérer Vider Supprimer	1	MyISAM	utf8mb4_unicode_520_ci	7,4 KLo	-
msw_links	Afficher Structure Rechercher Insérer Vider Supprimer	0	MyISAM	utf8mb4_unicode_520_ci	1 KLo	-
msw_options	Afficher Structure Rechercher Insérer Vider Supprimer	140	MyISAM	utf8mb4_unicode_520_ci	156,6 KLo	42,8 KLo
msw_postmeta	Afficher Structure Rechercher Insérer Vider Supprimer	169	MyISAM	utf8mb4_unicode_520_ci	38 KLo	740 o
msw_posts	Afficher Structure Rechercher Insérer Vider Supprimer	60	MyISAM	utf8mb4_unicode_520_ci	60 KLo	-
msw_termmeta	Afficher Structure Rechercher Insérer Vider Supprimer	0	MyISAM	utf8mb4_unicode_520_ci	4 KLo	-
msw_terms	Afficher Structure Rechercher Insérer Vider Supprimer	3	MyISAM	utf8mb4_unicode_520_ci	13,2 KLo	-
msw_term_relationships	Afficher Structure Rechercher Insérer Vider Supprimer	11	MyISAM	utf8mb4_unicode_520_ci	3,2 KLo	-
msw_term_taxonomy	Afficher Structure Rechercher Insérer Vider Supprimer	3	MyISAM	utf8mb4_unicode_520_ci	4,1 KLo	-
msw_usermeta	Afficher Structure Rechercher Insérer Vider Supprimer	42	MyISAM	utf8mb4_unicode_520_ci	12,6 KLo	-
msw_users	Afficher Structure Rechercher Insérer Vider Supprimer	2	MyISAM	utf8mb4_unicode_520_ci	8,2 KLo	-
12 tables	Somme	431	MyISAM	utf8_general_ci	312,9 KLo	43,8 KLo

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 : https://codex.wordpress.org/Database_Description

Voici les tables natives de WordPress avec le préfixe de base wp :

- 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** et pour la configuration dans l'onglet **Réglages - Akismet Anti-Spam**.
- 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, mais aussi le contenu de vos posts personnalisés (custom post type), que l'on apprendra à créer.
- La table **wp_termmeta** contient les metadonnées ajoutées aux catégories. Cette nouvelle table est disponible depuis la version 4.4.