PHP #4

PHPDocumentor

&

PHP au sein de Wordpress

# Contexte d’utilisation de PHPDocumentor

1. Exécuter *Putty* que vous avez déjà utilisé en Réseau.
2. Se connecter en SSH sur webmmi.iut-tlse3.fr
3. Télécharger le logiciel ***PHP Documentor*** directement vers votre espace ftp (Ne vous déplacez pas dans public\_html !) :

wget <https://phpdoc.org/phpDocumentor.phar>

1. Ajouter (sur ce fichier seulement !) les droits d’exécution :

chmod u+x phpDocumentor.phar

1. Créer un dossier dans lequel sera créée la documentation de vos fichiers php :

mkdir public\_html/phpdoc

# Génération de la documentation automatique

1. Lancer le scan de vos fichiers php pour créer leur documentation à partir des commentaires automatiques que vous y avez fait figurer :

./phpDocumentor.phar -d public\_html/*php* -t public\_html/phpdoc

*(Le cas échéant, remplacer php par le dossier dans lequel vous avez vos fichiers php.)*

1. Dans [http://webmmi.iut-tlse3.fr/~*login*/phpdoc](http://webmmi.iut-tlse3.fr/~/phpdoc) vérifier la documentation créée :

Par exemple dans « **Files** », vérifier que vos scripts exercices*n*.php montrent les variables et les fonctions définies.

# Contexte des exercices dans wordpress

1. Télécharger depuis *Moodle* les fichiers de départ vers votre espace personnel.

Pour cette séance, vous ferez les exercices demandés dans certains fichiers de votre *wordpress* sur webmmi. Rappelez-vous qu’**il faut téléverser le fichier à chaque modification** pour le tester.

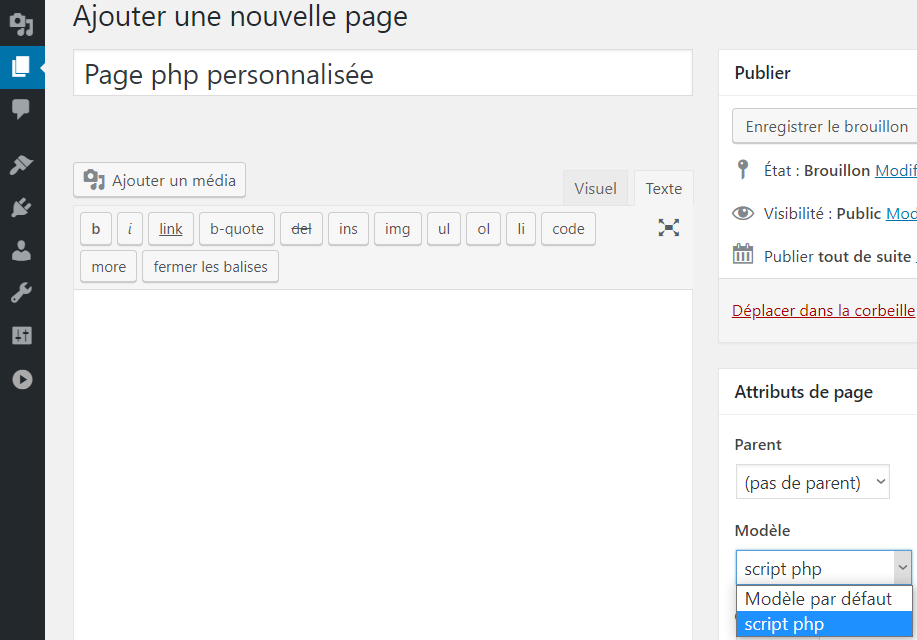
1. Se connecter à l’administration de votre *wordpress*.
2. Repérer dans **Apparence** > **Thèmes**, repérer le thème activé.

***Attention : dans le module d’enseignement « Intégration web – Wordpress », vous serez amené à changer et même à créer un thème enfant.***

1. Via *FileZilla*, repérer le dossier de ce thème dans l’arborescence de *wordpress* :

/public\_html/wordpress/wp-content/themes/

# 1ère possibilité : intégrer un script php à un modèle de page personnalisée

1. Parmi les fichiers de départ récupérés, téléverser le modèle de page page-scriptphp.php dans le dossier du thème.
2. Dans l’administration de *wordpress*, ajouter une nouvelle page et choisir « **script php** » comme **Modèle** dans les **Attributs de page** :
3. Tester l’aperçu de cette page (vous pouvez bien sûr la publier également) : du code php spécifique s’affiche en cohérence avec le thème.
4. Éditer le fichier page-scriptphp.php
5. Repérer le début et la fin du programme php de la partie « main ».
6. Y coller le programme php d’un de vos exercices php précédents.

*Remarque : si votre programme contient l’instruction include (c’est-à-dire si le fichier nécessite un autre fichier, comme connexion.inc.php), vous devez téléverser également cet autre fichier. (Ultérieurement nous verrons où placer ces fichiers annexes.)*

1. Tester **l’affichage de votre programme php dans *wordpress***.

À partir de maintenant, si c’est approprié vous pouvez effectuer les prochains exercices php dans une page *wordpress* comme celle-ci.

# 2ème possibilité : code court *wordpress*

1. Parmi les fichiers de départ récupérés, éditer le fichier function titre\_livre.php
2. Via *FileZilla*, éditer le fichier **functions.php** du thème.
3. Copier-coller le programme php de function titre\_livre.php A LA FIN du fichier functions.php

Le programme que l’on rajoute va créer le **shortcode « titre\_livre »** avec le titre d’un livre retourné par la fonction titre\_premier\_livre()

1. Dans l’administration de *wordpress*, ajouter un nouvel article qui fait apparaître le shortcode…

**[titre\_livre]**

1. Tester l’aperçu de cet article (vous pouvez bien sûr le publier également) : **une variable php spécifique s’affiche** en cohérence avec le contenu *wordpress*.
2. Consulter la documentation de la fonction *wordpress* add\_shortcode : <https://developer.wordpress.org/reference/functions/add_shortcode/> et identifier :
   * Le rôle du 1er paramètre : la chaine de caractère qu’on va mettre entre crochets pour appeler la fonction en second paramètre
   * Le rôle du 2nd paramètre : la fonction qu’on souhaite appeler avec le shortcode
   * S’il y a une valeur de retour ? ce que fait la fonction en 2nd paramètre
3. En vous appuyant sur la documentation, ainsi qu’en reprenant la création du short-code pour le titre, éditer le fichier functions.php et y **créer un short-code pour le prix** du livre.
4. Ajouter l’affichage du prix dans l’article précédent.
5. Tester l’utilisation de ces 2 short code dans d’autre article, dans une page.

# *3ème possibilité : exécution d’un code php lors d’une action de wordpress (concept de hook) (Pour aller plus loin…)*

1. Copier-coller la fonction php\_footer() depuis le fichier function php\_footer.php dans le fichier functions.php
2. Consulter la documentation de la fonction *wordpress* add\_action pour faire exécuter la fonction php\_footer() quand *Wordpress* affiche le *footer*.
3. Tester sur votre site *wordpress*.

# *Vérification de la documentation automatique (Pour aller plus loin…)*

1. Corriger les erreurs éventuelles affichées par PHP Documentor.