

## WIM2 - Programmation Web côté Serveur – TD Machine 1 & 2

- Pour votre rapport, créez un fichier odt de nom « votrenom\_votreprenom\_WIM2\_TDM\_1\_2 » dans lequel vous mettrez les réponses aux exercices, les informations que vous souhaitez retenir et les commandes git que vous avez exécutées.
- Sur votre compte github, si ce n'est pas encore fait, créer un repository privé de nom « wim2-votrenom-votreprenom ». Ajoutez votre enseignant comme collaborateur.
- Sur le répertoire ROOT du serveur Wamp (www), cliquez-droit et sélectionnez Git Bash Here dans le menu. Dans la fenêtre de commande tapez : git clone suivi de l'adresse de votre repository wim2. Un répertoire wim2-x-y (où x est votre nom et y votre prénom) a été créé dans www. Faites de ce répertoire le dossier courant en tapant la commande cd wim2-x-y.
- Chaque fois que vous avez terminé un exercice, vous devez ajouter le script php que vous avez réalisé à votre repository wim2-x-y (entrez : « git add monfichier.php » puis « git commit -m 'Exo num ...' »).
- A la fin du TDM n'oubliez pas de synchroniser votre dépôt wim2xx local avec votre dépôt sur github (commandes : « git fetch » puis « git push origin master »).

Remarque : Dans l'explorateur de fichiers de windows, vous devez faire en sorte que les éléments masqués et les extensions des noms de fichier soient affichés.

Sous win7, cliquez sur « Organiser » puis sur « Options des dossiers et recherche », puis sur l'onglet « Affichage ». Cochez « Afficher les fichiers, dossiers et lecteurs cachés » et décochez « Masquer les extensions des fichiers dont le type est connu ».

Sous win10, onglet « Affichage », cochez « Extensions de noms de fichiers » et « Elements masqués »

### Exercice 1 : Compléments de PHP

Le langage PHP comporte un certain nombre de fonctions permettant de gérer les URL. Nous allons en étudier quelques unes dans cet exercice. Créez un **sous-répertoire ex1** dans votre répertoire wim2.

#### A. Etude de la fonction parse\_url

##### 1. parse\_url

Sur le site php.net et recherchez la page de cette fonction. Notez dans votre rapport ce que fait cette fonction et sa signature. Expliquez ses paramètres et sa valeur de retour.

##### 2. Test sur des exemples

Réalisez un script PHP, encodé en utf-8 sans BOM, de nom test\_parse\_url\_1.php, qui affiche les résultats de l'application de cette fonction aux URL suivantes que vous mettrez dans un tableau de strings :

http://username:password@hostname.com/path/file.php?arg1=value1&arg2=value2#anchor

http://hostname.com:8080/path/file.php?arg1=value1&arg2=value2#anchor

http://subdom.hostname.com/path/file.php?arg1=value1&arg2=value2#anchor

//www.example.com/path/file.php?arg1=value1&arg2=value2#anchor

www.example.com/path/file.php?arg1=value1&arg2=value2#anchor

/path/file.php?arg[]=value1&arg[]=value2#anchor

path/file.php?arg1=value1&arg2=value2#anchor

Les résultats sont-ils conformes à ce que vous attendiez ? Expliquez.

##### 3. Test de l'effet du deuxième argument

Réalisez un script PHP de nom test\_parse\_url\_2.php qui reprend les 7 URL précédentes mais ajoute un deuxième argument dans l'appel de fonction prenant successivement les valeurs : PHP\_URL\_USER, PHP\_URL\_PORT, PHP\_URL\_SCHEME, PHP\_URL\_HOST, PHP\_URL\_PATH, PHP\_URL\_QUERY, PHP\_URL\_FRAGMENT (une seule valeur par URL).

## B. Etude de la fonction parse\_str

### 1. parse\_str

Sur le site php.net et recherchez la page de la fonction parse\_str. Notez dans votre rapport ce que fait cette fonction et sa signature. Expliquez ses paramètres et sa valeur de retour.

### 2. Test sur des exemples

Réalisez un script PHP de nom test\_parse\_str.php qui affiche les résultats de l'application de la fonction parse\_str aux querystring des URL suivantes :

http://localhost/drupal/module1/fonction1/?arg1=value1&arg2=value2#anchor

http://localhost/drupal/index.php?q=module1/fonction1&arg[]=value1&arg[]=value2#anchor

http://localhost/drupal/index.php?q=module1/fonction1?arg1=value1&arg2=value2#anchor

Les résultats sont-ils conformes à ce que vous attendiez ? Expliquez.

## C. Etude des fonctions urldecode, rawurlencode, urlencode, rawurlencode et http\_build\_query

Recherchez les pages de ces fonctions sur php.net puis répondez aux questions suivantes :

1. Que font les fonctions urldecode et rawurlencode et quelle est la différence entre les deux.

2. Que font les fonctions urlencode et rawurlencode et quelle est la différence entre les deux.

3. Que fait la fonction http\_build\_query, quel est le type de ses arguments ?

### 4. Test 1 sur des exemples

4.1. Ecrivez un script PHP de nom test\_url\_code\_1.php contenant les instructions suivantes :

```
<html><head>
<meta charset="utf-8">
<title>test_url_code_1</title></head><body>
<?php
echo 'var_dump de $_GET :'; var_dump($_GET);
echo 'Ré encodage de $_GET par http_build_query($_GET) : ', http_build_query($_GET), '<br/><br/>';
echo 'urldecode(http_build_query($_GET)) = ', urldecode(http_build_query($_GET)), '<br/><br/>';

$x = $_SERVER['REQUEST_URI']; echo '$_SERVER["REQUEST_URI"] = ', $x, '<br/>';

echo 'urldecode($_SERVER["REQUEST_URI"]) = ', urldecode($x), '<br/><br/>';
?>
</body></html>
```

4.2. Appelez votre script avec l'URL suivante: 127.0.0.1/wim2/test\_url\_code.php?var1=%BA&var2=%C2%BA et expliquez les résultats obtenus.

4.3. Que se passe-t-il si vous supprimez la ligne <meta charset="utf-8"> ? Expliquez.

### 5. Test 2 sur des exemples

5.1. Ecrivez un script PHP de nom test\_url\_code\_2.php contenant les instructions suivantes :

```
<html><head>
```

```

<meta charset="utf-8">
<title>test_url_code_1</title></head><body>
<?php
$url1 = 'edit?cours=wim2&auteur=Denis Finck';
echo 'URL : ', $url1, '<br/>';
echo 'Encodage : ', urlencode($url1), '<br/><br/>';
echo 'Décodage : ', urldecode(urlencode($url1)), '<br/><br/>';
echo 'Encodage : ', rawurlencode($url1), '<br/><br/>';
echo 'Décodage : ', rawurldecode(rawurlencode($url1)), '<br/><br/>';
echo 'Décodage mixte : ', urldecode(rawurlencode($url1)), '<br/><br/>';
echo 'Décodage mixte : ', rawurldecode(urlencode($url1)), '<br/><br/>';
echo 'Double codage mixte : ', rawurlencode(urlencode($url1)), '<br/><br/>';
?>
</body></html>

```

5.2. Expliquez les résultats obtenus.

## Exercice 2 : Règles de réécriture d'URL, fichier .htaccess

Vous trouverez des informations pour vous aider dans les documents suivants:

<http://httpd.apache.org/docs/2.4/fr/rewrite/intro.html>  
<http://httpd.apache.org/docs/2.4/fr/rewrite/flags.html>  
[http://httpd.apache.org/docs/2.4/fr/mod/mod\\_rewrite.html](http://httpd.apache.org/docs/2.4/fr/mod/mod_rewrite.html)

1. Créez un sous-répertoire de nom ex2 dans votre répertoire wim2XX.
2. Dans ce répertoire créez un fichier texte de nom .htaccess. (windows supprimera le point final).

3. Ouvrez votre fichier .htaccess avec notepad++ et entrez le code suivant :

```

RewriteEngine on
#RewriteCond %{REQUEST_FILENAME} !-f
#RewriteCond %{REQUEST_FILENAME} !-d
RewriteCond %{REQUEST_URI} !=/favicon.ico
RewriteRule ^ pierre.php [L]

```

4. Pour XXX valant successivement pierre, paul, jean, créez les fichiers XXX.php de contenu :

```

<?php
echo 'Vous avez demandé le fichier '.$_SERVER['REQUEST_URI'].'<br>
echo "Je m'appelle XXX.";

```

5. Dans un navigateur entrez l'URL 127.0.0.1/wim2XX/ex2/jean.php.  
Quel résultat obtenez vous ? Expliquez.

6. Dans le fichier .htaccess remplacez la dernière ligne par :

```

RewriteRule ^p pierre.php [L]

```

Quels résultats obtenez vous pour les URL 127.0.0.1/jean.php et 127.0.0.1/paul.php ? Expliquez.

7. Dans le fichier .htaccess supprimez les # au début des lignes 2 et 3 et sauvegardez le fichier.  
Quels résultats obtenez vous pour les URL 127.0.0.1/jean.php et 127.0.0.1/paul.php ? Expliquez.

8. Dans votre rapport, expliquez chaque ligne du fichier .htaccess en détaillant chaque élément.