

PHP - Pages / Formulaires

Pages / Formulaires

Pierre Blarre - Florian Rodriguez

Plan

- 1. Inclure d'autres fichiers - exemple
- 2. Passer de la donnée - `$_GET`
- 3. Récupérer les données d'un formulaire en PHP
- 4. Session

Inclure d'autres fichiers - exemple

exemple-include.php

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Bienvenue dans notre cours PHP</title>
5   </head>
6   <body>
7     <h1>Test inclure</h1>
8     <?php include 'test.php'; ?>
9     <div>
10      <p>Fin de test inclure</p>
11    </div>
12  </body>
13 </html>
```

test.php

```
1 <p>Le contenu de ce fichier</p>
2 <?php echo '<p>sera inclus dans un autre
3 fichier</p>'; ?>
```

Résultat

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Bienvenue dans notre cours PHP</title>
5   </head>
6   <body>
7     <h1>Test inclure</h1>
8     <p>Le contenu de ce fichier</p>
9     <p>sera inclus dans un autre fichier</p> <div>
10      <p>Fin de test inclure</p>
11    </div>
12  </body>
13 </html>
```

Exemple: <http://localhost:2023/exemple-include.php>

Inclure d'autres fichiers

- `include monFichier.php` va être remplacé à l'exact endroit où il est écrit par le code de `monFichier.php`
- Cela est très utile pour réutiliser du code par exemple pour inclure l'en-tête ou le pied page dans toutes les pages d'un site.
- Les différentes possibilités pour inclure d'autres fichiers:
 - `include` -> inclut un fichier mais ne produit pas d'erreur en cas d'échec (ex: fichier absent)
 - `require` -> inclut un fichier mais produit une erreur (fatal) en cas d'échec (ex: fichier absent)
 - `include_once`, `require_once` -> idem mais n'inclut pas le fichier si il est déjà inclus

Passer de la donnée - \$_GET

PHP inclut des variables dites “super globales”, qui sont toujours disponibles, dans n’importe quel programme PHP.

Une de ces variables superglobales est : `$_GET`

Elle permet de récupérer les données transmises dans une URL

Exemple: <http://localhost:2023/exemple-include.php?mavariabile=test>

Passer de la donnée - \$_GET

Exemple: <http://localhost:2023/exemple-include.php?mavariabile=test>

Dans cet url d'exemple ?mavariabile=test est appelé «query string»

La query string commence après le «?»

On a ensuite un ensemble clé/valeur séparé par des &

On peut récupérer l'information dans le fichier php exemple-include.php grâce à:

`$_GET['mavariabile']`

Exemple

<http://localhost:2023/exemple-get.php?maVariable=titi&maDeuxiemeVariable=toto>

exemple-get.php

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Bienvenue dans notre cours PHP</title>
5   </head>
6   <body>
7     <?php
8       echo $_GET['maVariable'].'<br>';
9       echo $_GET['maDeuxiemeVariable'].'<br>';
10    ?>
11  </body>
12 </html>
```



Résultat

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Bienvenue dans notre cours PHP</title>
5   </head>
6   <body>
7     titi<br>toto<br></body>
8 </html>
```



Récupérer les données d'un formulaire en PHP

Une des fonctionnalités offertes par le HTML est de pouvoir créer des formulaires qui pourront envoyer les données entrées dans le formulaire vers une URL.

Les formulaires HTML peuvent envoyer leurs données de 2 manières différentes, en `get` ou en `post`.

La méthode `get` va ajouter toutes les données à l'url, alors que la méthode `post` va envoyer les données dans la requête http.

La variable superglobale pour la méthode `post` est `$_POST`

Pour pouvoir ensuite ***récupérer*** ces données côté serveur, par exemple pour enregistrer les informations dans une base de données, on va utiliser PHP!

Un formulaire HTML avec la méthode Get

- Formulaire simple qui enverra une requête GET

```
1 <!-- Formulaire simple qui enverra une requête GET -->
2 <form action="http://localhost:2023/exemple-form-get.php" method="get">
3   <label for="GET-name">Nom :</label>
4   <input id="GET-name" type="text" name="nom">
5   <input type="submit" value="Enregistrer">
6 </form>
```

→ Liens vers la page de test:

<http://localhost:2023/exemple-form-get.php>

→ Enverra les données de cette manière:

<http://localhost:2023/exemple-form-get.php?nom=Toto&submit=Enregistrer>

- En PHP, on aura:
 - `$_GET['nom']` -> "Toto";
 - `$_GET['submit']` -> "Enregistrer";

Un formulaire HTML avec la méthode Post

- Formulaire simple qui enverra une requête POST

```
1 <!-- Formulaire simple qui enverra une requête POST -->
2 <form action="http://localhost:2023/exemple-form-post.php" method="post">
3   <label for="POST-name">Nom :</label>
4   <input id="POST-name" type="text" name="name">
5   <input type="submit" value="Enregistrer">
6 </form>
```



→ Liens vers la page de test:

<http://localhost:2023/exemple-form-post.php>

→ Enverra les données de cette manière:

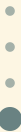
Les données ne seront pas visibles dans l'url mais sont dans la requête http envoyée au serveur. La méthode POST envoie l'information d'une autre manière.

- En PHP, on aura:
 - `$_POST['name'] -> "Pierre";`
 - `$_POST['submit'] -> "Enregistrer";`

Transmission des données

Une fois des données transmises avec l'url ou bien en méthode post, il est donc possible de les récupérer depuis un script PHP.

Une fois dans un script PHP, il sera possible d'effectuer une connexion avec une base de données et de sauvegarder ensuite ces données dans la base.



Session

- On sait envoyer de l'information entre page grâce à l'url mais si on charge d'autres pages cette information est «oubliée».
→ Session
- Les sessions permettent de conserver des variables sur toutes les pages du site.



Session - création

- Un visiteur arrive sur le site.
- On créer une session pour lui.
- PHP génère alors un numéro unique.
(ex: PHPSESSID : "ampo0n7njvesjf33b7mnfqbtca")
- Ce numéro est sauvegardé sur un cookie

Session - utilisation

- `session_start()` sur chacune des pages AVANT d'écrire le moindre code HTML ou PHP
- On a alors accès à la variable superglobale `$_SESSION` .
- On peut alors accéder à une infinité de variable de session du type `$_SESSION['maVariable1']`
- `session_destroy()` pour détruire manuellement la session

Exemple

```
1 <?php
2 // Démarrer la session
3 session_start();
4
5 // Vérifier si le compteur de visites existe dans la session
6 if (!isset($_SESSION['compteur'])) {
7     $_SESSION['compteur'] = 0; // Initialiser le compteur à 0
8 }
9
10 // Incrémenter le compteur de visites
11 $_SESSION['compteur']++;
12
13 // Afficher le nombre de visites
14 echo "Nombre de visites sur cette page : " . $_SESSION['compteur'];
15 ?>
```

→ Liens vers la page de test:

<http://localhost:2023/exemple-session.php>