

PHP et les formulaires

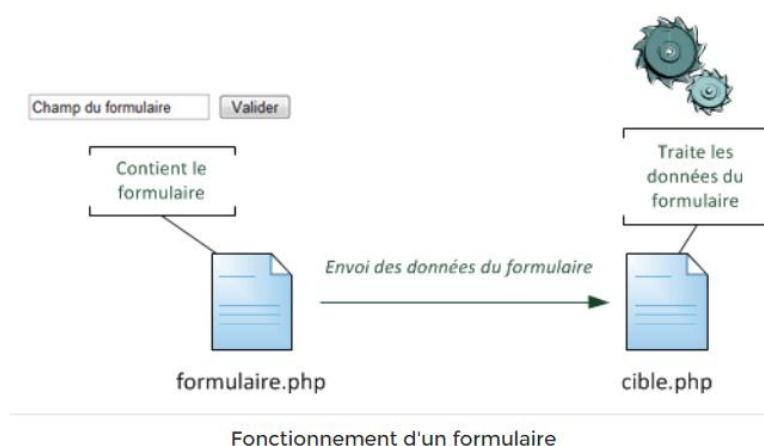
Dr Edouard Ngor SARR
Maitre-Assistant
UCAO-Saint Michel
Edouard.sarr@ucao.edu.sn

1. Introduction

Encore une fois on retrouve la notion de formulaire qui est un des points les plus important dans la réalisation d'un site application web dynamique. Les formulaires constituent le principal moyen pour vos visiteurs d'entrer des informations sur votre site. Les formulaires permettent de créer une **interactivité**. C'est un moyen très adaptés pour réaliser une attractivité entre le visiteur et le site en question. De la saisie jusqu'à la récupération des données, nous allons voir toutes les notions de base qu'il faut connaître lorsque l'on manipule les formulaires. Dans ce contexte, on verra une utilisation fréquente des super globales.

2. Envoie des données du formulaire : Les méthodes GET et POST

Les deux méthodes sont à peu près les mêmes puisque toutes les deux permettent d'envoyer des données. Toutefois, chacun a sa façon de gérer les choses et le choix des méthodes utilisées nécessite la vigilance du développeur.



a. Formulaire avec méthode GET

La méthode GET consiste à transmettre les données via l'URL. Il est intéressant de l'utiliser dans le cas où on veut faire transiter des données de pages en pages. Cependant, il faut avoir conscience que les données seront transmises par http et donc visibles dans la barre d'adresse. **Cet aspect mérite une attention particulière parce que de la sécurité des données est menacée.** Dans le cas où les données sont confidentielles, mieux vaut chercher à ne pas utiliser cette méthode. Si toutefois



on ne peut s'en passer, il faut essayer de cacher les valeurs transmises en utilisant les cryptages par exemple.

b. Formulaire avec méthode POST

Une autre méthode pour l'envoi des formulaires est la méthode POST. Il est obligatoire de l'utiliser dans certains cas comme l'envoi de fichiers, envoi de données lourdes, envoi de données confidentielles. La méthode POST est également recommandée si le formulaire déclenche une action spécifique qui doit être renouvelée à chaque fois. Avec la méthode POST, il est possible de revenir en arrière et de re-soumettre le formulaire.

NB : Les champs cachés : Les champs cachés constituent un type de champ à part. C'est un code dans votre formulaire qui n'apparaîtra pas aux yeux du visiteur, mais qui va quand même créer une variable avec une valeur. On peut s'en servir pour transmettre des informations fixes.

```
<input type="hidden" name="codesecret" value="S@RRIS2020" />
```

À l'écran, sur la page web on ne verra rien. Mais dans la page cible, une variable sera créée, et elle aura la valeur « S@RRIS2020 » !

Exemple de formulaire utilisant la méthode POST :

```
<HTML>

  <BODY>
    <form method="post" action="questionnaire.php">
      <label for="nom">Nom:</label>
      <input type="text" name="nom" id="nom" />
      <label for="prenom">Prénom:</label>
      <input type="text" name="prenom" id="prenom" />
      <label for="email">Email:</label>
      <input type="text" name="email" id="email" />
      <label for="titre">Titre : </label>
      <select name="titre">
        <option value="Mme" selected="selected">Madame</option>
        <option value="Mlle">Mademoiselle</option>
        <option value="Mr">Monsieur</option>
      </select>
    </form>
  </BODY>
</HTML>
```



3. Récupération des données

Dans l'attribut action d'un formulaire, on spécifie l'adresse du fichier dans lequel les données sont renvoyées et traitées. Les données transmises se trouvent souvent sous forme de tableau dans des variables qu'on appelle des superglobales :

- Le tableau `$_GET` permet de récupérer toutes les données envoyées par la méthode GET. Il contient toutes les données transmises via l'URL.
- Le tableau `$_POST` contient les données envoyées par la méthode POST.
- Le tableau `$_FILES` est spécifique à l'envoi de fichiers. Il contient les informations du fichier quand l'utilisateur télécharge un fichier via un formulaire.

Pour récupérer une donnée, on utilise généralement le nom du champ comme index du tableau `$_POST`.

- La fonction `isset()` permet de savoir si une variable a été affectée ou non. Si vous ne cochez pas la case musique, la variable `$_POST['musique']` ne contient pas de valeur. **Et puisque'on en parle, profitons pour voir ce qu'est la fonction `empty()`.**
- La fonction `'empty()'` a le même rôle que la fonction `isset()` sauf qu'elle renvoie true lorsque la variable vaut 0 ou non définie.

Exemple Code dans questionnaire.php

```
<?php
    $tr="";
    $pr="";
    $nm="";
    $em="";
    if(isset($_POST['titre'])){
        $tr= $_POST['titre'];
    }
    if(isset($_POST['prenom'])){
        $pr= $_POST['prenom'];
    }
    if(isset($_POST['nom'])){
        $nm= $_POST['nom'];
    }
    if(isset($_POST['email'])){
        $em= $_POST['email'];
    }
    }
?>
```

4. Téléchargement d'un fichier

Pour télécharger un fichier, le champ XHTML associé est un input de type FILE.

Voici un exemple de code qu'on mettra dans un fichier upload.html



```
<form action="enregistre.php" method="post" enctype="multipart/form-data">
  <p>
    <input type="file" name="fichier" size="40" />
    <input type="hidden" name="MAX_FILE_SIZE" value="2048">
    <input type="submit" value="Envoyer" />
  </p>
</form>
```

Une fois le formulaire validé, les valeurs seront postées dans enregistre.php donc voici le code :

```
<?php
    $nom_fichier= $_FILES['fichier']['tmp_name'];
    echo "nom temporaire:". $nom_fichier;
    $nom_destination='C:\wamp\www\img\image.jpg';
    move_uploaded_file($nom_fichier, $nom_destination);
?>
```

La variable globale utilisée pour les fichiers est \$_FILE .

Au moment du téléchargement, PHP essaie de mettre le fichier dans un répertoire temporaire et lui donne un nom temporaire.

- C'est la fonction move_uploaded_file qui permet de déplacer le fichier dans le répertoire dans lequel vous voulez le mettre. Ensuite, vous pouvez lui spécifier un nom à votre aise.
- Le champ caché 'MAX_FILE_SIZE', quant-à lui, permet de spécifier la taille maximum du fichier que l'on va télécharger.

5. Exercice d'application

Sans copier, ni copier/coller le cours, concevez une page qui s'appelle imc.php, et propose un formulaire sous la forme suivante :

- Entrez votre prénom :
- Entrez votre taille (sous la forme 1.70) :
- Entrez votre poids (en kilos) :

Ensuite vous traiterez le formulaire en PHP pour que quand la personne clique, elle voit apparaître :

```
Bonjour (son prénom)
Votre IMC (indice de masse corporelle) est exactement : (son imc)
Vous avez (une corpulence normale ou bien Vous êtes en surpoids, ou bien Vous êtes obèse etc...)
```

Pour la petite opération, l'IMC, c'est le poids divisé par la taille divisé par la taille... En voici la syntaxe :

\$poids/(\$taille*\$taille)