

Développement web S1

– 5. Gestion des formulaires –

Christophe BLANC

– BUT MMI –
IUT CLERMONT AUVERGNE
UNIVERSITÉ CLERMONT AUVERGNE
WWW.CHRISTOPHE-BLANC.FR

2023-2024

- ✓ Un 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.
- ✓ Il Un formulaire HTML est défini entre les balises `<FORM>` et `< \FORM>`

Syntaxe

```
<FORM  
  [ ACTION = "url_de_traitement" ]  
  [ METHOD = "GET" | "POST" ]  
  [ NAME = "nom_formulaire" ]  
>  
...  
</FORM>
```

- ✓ **ACTION** : URL relative ou absolue, qui va traiter le formulaire, en ce qui nous concerne, un script PHP. Si rien n'est spécifié, la même URL que la page est utilisée
- ✓ **METHOD** : 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.
- ✓ **NAME** : nom du formulaire. Si la page HTML contient plusieurs formulaires, le nom permet de les différencier. En ce qui nous concerne, ce nom ne présente pas d'intérêt car il n'est pas récupéré dans le script de traitement du formulaire.

Gestion des formulaires

Exemple

```
<FORM>
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">Feminin
<INPUT TYPE="radio" NAME="sexe" VALUE="?" CHECKED>Ne sait pas
</BR>Photo :
<INPUT TYPE="file" NAME="photo" VALUE="" SIZE="50">
</BR>Couleurs preferees :
<INPUT TYPE="checkbox" NAME="bleu"> Bleu
<INPUT TYPE="checkbox" NAME="blanc"> Blanc
<INPUT TYPE="checkbox" NAME="rouge"> Rouge
<INPUT TYPE="checkbox" NAME="nesaitpas" CHECKED> Ne sait pas
</BR>Langue :
<SELECT NAME="langue">
    <OPTION VALUE="E">Espagnol
    <OPTION SELECTED VALUE="F">Francais
    <OPTION VALUE="I">Italien
</SELECT>
...
```

Gestion des formulaires

Exemple

```
...
</BR>Fruits preferes :</BR>
<SELECT NAME="fruits" MULTIPLE SIZE="8">
  <OPTION VALUE="A">Abricots
  <OPTION VALUE="C">Cerises
  <OPTION VALUE="F">Fraises
  <OPTION VALUE="P">Peches
  <OPTION SELECTED VALUE="?">Ne sait pas
</SELECT>
</BR>Commentaire :</BR>
<TEXTAREA NAME="Commentaire" ROWS="4" COLS="50">
</TEXTAREA>
</BR>
<INPUT TYPE="submit" NAME="OK" VALUE="OK">
</FORM>
```

Gestion des formulaires

Exemple

Nom : Mot de passe :

Sexe : Masculin Feminin Ne sait pas

Photo : Aucun fichier choisi

Couleurs preferees : Bleu Blanc Rouge Ne sait pas

Langue : ▾

Fruits preferes :

- Abricots
- Cerises
- Fraises
- Peches
- Ne sait pas

Commentaire :

Gestion des formulaires

Interaction entre un formulaire et un script PHP

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).

Gestion des formulaires

Interaction entre un formulaire et un script PHP

Trois méthodes sont utilisables pour faire interagir un formulaire et un script PHP :

- ✓ placer le formulaire dans un document HTML "pur" (.html), le formulaire ne contient alors aucun élément dynamique, et indiquer le nom du script PHP qui doit traiter le formulaire dans l'option *ACTION* de la balise `<FORM>`
- ✓ placer le formulaire dans un script PHP (par exemple pour construire une partie du formulaire dynamiquement) et faire traiter le formulaire par un autre script PHP (mentionné dans l'option *ACTION* de la balise `<FORM>`)
- ✓ placer le formulaire dans un script PHP (par exemple pour construire une partie du formulaire dynamiquement), et le faire traiter par le même script PHP (mentionné dans l'option *ACTION* de la balise `<FORM>` ou appelé par défaut si cette option n'est pas présente).

Gestion des formulaires

Interaction entre un formulaire et un script PHP : exemple première méthode

Document html saisie.html

```
<HTML>
<HEAD><TITLE>Saisie</TITLE></HEAD>
<BODY>
<FORM ACTION="traitement.php" METHOD="POST">
Nom : <INPUT TYPE="text" Name="nom" VALUE=""></BR>
<INPUT TYPE="submit" NAME="OK" VALUE="OK">
</FORM>
</BODY>
</HTML>
```

Script PHP traitement.php

```
<?php
/* A faire
    -recuperer les informations saisies
    -faire le traitement
    -afficher une nouvelle page
*/
?>
```

Le résultat du clic sur le bouton OK est une page vide car, pour l'instant, le script de traitement ne fait rien.

Gestion des formulaires

Interaction entre un formulaire et un script PHP : exemple deuxième méthode

Document PHP saisie.php

```
<?php
//inclure un fichier qui contient des definitions de
//constantes, dont le titre de la page (TITRE_PAGE_SAISIE)
require("constantes.inc");
//initialisation d'une variable qui contient la valeur
//initiale de la zone saisie (dans la pratique, cette
//valeur vient sans doute d'ailleurs et n'est pas codee en dur
$nom="X";
//dans le code HTML qui suit, inclusion de deux petits
//bouts de code PHP pour afficher respectivement le titre
//de la page et la valeur initiale de la zone de saisie
?>
<HTML>
<HEAD><TITLE><?php echo TITRE_PAGE_SAISIE ?></TITLE></HEAD>
<BODY>
<FORM ACTION="traitement.php" METHOD="POST">
Nom : <INPUT TYPE="text" Name="nom" VALUE="<?php echo $nom?>"></BR>
<INPUT TYPE="submit" NAME="OK" VALUE="OK">
</FORM>
</BODY>
</HTML>
```

Script PHP traitement.php

```
<?php
/* A faire
    -recuperer les informations saisies
    -faire le traitement
    -afficher une nouvelle page
*/
?>
```

Gestion des formulaires

Interaction entre un formulaire et un script PHP : exemple troisième méthode

Document PHP saisie.php

```
<?php
//inclure un fichier qui contient des definitions de
//constantes, dont le titre de la page (TITRE_PAGE_SAISIE)
require("constantes.inc");
//initialisation d'une variable qui contient la valeur
//initiale de la zone saisie (dans la pratique, cette
//valeur vient sans doute d'ailleurs et n'est pas codee en dur
$nom="X";
//dans le code HTML qui suit, inclusion de deux petits
//bouts de code PHP pour afficher respectivement le titre
//de la page et la valeur initiale de la zone de saisie
?>
<HTML>
<HEAD><TITLE><?php echo TITRE_PAGE_SAISIE ?></TITLE></HEAD>
<BODY>
<FORM ACTION="saisie.php" METHOD="POST">
Nom : <INPUT TYPE="text" Name="nom" VALUE="<?php echo $nom?>"></BR>
<INPUT TYPE="submit" NAME="OK" VALUE="OK">
</FORM>
</BODY>
</HTML>
```

Le résultat du clic sur le bouton OK est la même page de nouveau affichée, car, pour l'instant, le script de traitement ne fait rien. Utiliser le même script pour l'affichage initial et le traitement nécessite d'être en mesure de faire la différence entre les deux types d'appels.

Gestion des formulaires

Interaction entre un formulaire et un script PHP : quelle méthode choisir ?

- ✓ Séparer la page HTML du script PHP présente un inconvénient au niveau de la maintenance : si des modifications sont apportées au formulaire, il y a deux fichiers à modifier.
- ✓ Inversement, si le formulaire n'a aucune partie dynamique, l'écrire dans un fichier HTML séparé du script PHP qui le traite, permet de bien séparer l'interface utilisateur (la couche "présentation") du traitement
- ✓ Dans la pratique, pour faciliter la maintenance, il est souhaitable de définir certaines valeurs présentées à plusieurs reprises dans des constantes ou des variables et d'utiliser des constantes/variables dans les pages : toutes les pages deviennent un peu dynamiques et la troisième méthode paraît optimale.

Dans la suite, nous allons rentrer dans le détail du traitement du formulaire en PHP, en utilisant des exemples construits sur le modèle de la troisième méthode.

Gestion des formulaires

Interaction entre un formulaire et un script PHP : récupérer les données saisies dans le formulaire

Par défaut, toutes les zones de formulaire sont automatiquement enregistrées dans le script PHP qui traite le formulaire, dans un tableau associatif `$_POST` pour les formulaires POST et `$_GET` pour les formulaires GET : la clé du tableau est égale au nom de la zone dans le formulaire (option NAME de la balise `<INPUT>`, `<SELECT>` ou `<TEXTAREA>` et la valeur égale à la valeur saisie dans la zone.

Gestion des formulaires

Interaction entre un formulaire et un script PHP : récupérer les données saisies dans le formulaire - Exemple

```
<HTML>
<HEAD><TITLE>Saisie</TITLE></HEAD>
<BODY>
<?php
$nom = "";
// - tester si le tableau contient une valeur associee a la cle
// portant le meme nom que la zone de saisie du formulaire
// - si oui, recuperer cette valeur dans une variable et l'afficher
if ( isset($_POST["nom"]) )
{
    $nom = $_POST["nom"];
    echo "\$_POST->\$nom</BR>";
}
?>

<FORM ACTION="saisie.php" METHOD="POST">
Nom : <INPUT TYPE="text" Name="nom" VALUE="<?php echo \$nom?>"></BR>
<INPUT TYPE="submit" NAME="OK" VALUE="OK">
</FORM>
</BODY>
</HTML>
```

\$_POST -> Olivier

Nom :

