

Les formulaires : Transmettre des données de page en page

¹Un formulaire est constitué de champs et de contrôles : zones de texte, boutons radio, cases à cocher, listes déroulantes... Chacun de ces éléments se caractérise par un attribut « *name* » qui définira tout simplement le nom de la variable à récupérer en PHP avec la valeur rentrée par l'utilisateur.

Un formulaire HTML se définit avec les balises `<form>` et `</form>`. L'attribut "action" de cette balise permet de spécifier la page qui traitera les données fournies dans le formulaire par l'internaute ; l'attribut « *method* » la méthode de transmission (GET ou POST).

Ces deux méthodes peuvent être utilisées dans l'envoi de données via les formulaires. Nous préconisons l'emploi de la méthode POST car elle cache les informations transmises (qui transitent par l'URL dans le cas de la méthode GET). De plus, elle permet d'envoyer des données importantes en taille (la méthode GET se limite à 255 caractères) et assure la gestion de l'envoi de fichiers.

1 Les formulaires : la méthode POST

Exemple simple de formulaire

```
<form action="resultat.php" method="post">
Entrez votre prénom : <input type="text" name="prenom" />
<input type="submit" value="valider" />
</form>
```

Ce qui donnera sur votre navigateur, si vous rentrez « Bruno » :

Sur cet exemple, l'utilisateur a la possibilité de spécifier une seule information : son prénom, cette donnée sera exploitable dans la page « resultat.php » sous la forme d'une variable `$_POST["prenom"]`² (composée de l'attribut « *name* » de la zone de texte à remplir) dont la valeur sera égale à la chaîne de caractères entrée.

Ainsi on aura par exemple dans la page « resultat.php » :

```
<?php
echo "Prénom tapé par l'utilisateur : ".$_POST['prenom'];
?>
```

Prénom tapé par l'utilisateur : Bruno

Voici un exemple de traitement des variables qui émanent des différents contrôles et champs renseignés par un internaute sur un formulaire d'échange de liens.

```
<form method="POST" action="traitement.php">
Votre adresse e-mail<input type="text" name="mail" value="adresse e-mail" size="30" maxlength="55"><br/>
<input type="checkbox" NAME="mailing" value="oui" checked> abonnement gratuit à notre-planete.info<br/>
Description de votre site<br/>
<textarea name="description" rows="3" cols="60"></textarea><br/>
Type d'échange souhaité :<br/>
<select name="demande">
<option selected value="echange de liens">Echange de liens</option>
<option value="partenariat">Partenariat</option>
</select><br/>
Dans quelle rubrique souhaitez-vous être présent ?<br/>
```

¹ Source : http://www.notre-planete.info/PHP/cours_8.php

² on peut écrire indistinctement : `$_POST['prenom']`, `$_POST["prenom"]` ou `$_POST[prenom]`

```
<input type="radio" name="rubrique" value="environnement" />Environnement / Ecologie<br/>
<input type="radio" name="rubrique" value="geographie" />Géographie<br/>
<input type="radio" name="rubrique" value="photos" />Photos<br/>
<input type="submit" value="valider" />
</form><br/>
```

Ce qui donne sur votre navigateur, une fois qu'un demandeur a rempli le formulaire :

Votre adresse e-mail

abonnement gratuit à notre-planete.info

Description de votre site

Type d'échange souhaité :

Dans quelle rubrique souhaitez-vous être présent ?

Environnement / Ecologie

Géographie

Photos

Nous remarquerons que l'appel à la page de traitement des données est « traitement.php » : on peut tout à fait appeler la même page en utilisant une variable globale PHP intitulée `$_SERVER['PHP_SELF']` qui signifie la page en cours.

Ainsi, l'affichage du formulaire et l'exploitation de ses résultats seront opérés sur la même page. Ceci implique que dans le code PHP, un test soit effectué pour savoir si le formulaire a été rempli ou non, par exemple, en testant si une des variables du formulaire a été déclarée avec la fonction `isset()`.

Le code qui permet de récupérer les différentes valeurs saisies par l'utilisateur est le suivant :

```
<?php
$mail = $_POST["mail"];
echo "votre adresse e-mail est : $mail<br />";
$mailing = (isset($_POST["mailing"])? "oui" : "non");
echo "Avez-vous souhaité vous inscrire sur la lettre d'informations ? $mailing<br />";
$description = htmlentities($_POST["description"], ENT_QUOTES);
echo "et sa description est la suivante : <br />".nl2br($description)."<br />";
$rubrique = $_POST["rubrique"];
echo "pour la rubrique ".$rubrique;
?>
```

Remarques :

- pour la case à cocher la variable prend la valeur true ou false si la case a été cochée ou non. Un simple test (écrit sous une forme raccourcie ici) attribue à la variable `mailing` la valeur "oui" ou "non". En fait, si la case a été cochée, la variable « name » correspondante est déclarée, sinon elle n'existe pas ;
- l'attribut « name » des boutons radios est le même car ils appartiennent à une même liste de choix. Le bouton radio sélectionné crée une variable nommée `$_POST["rubrique"]` qui est égale à la valeur notée dans l'attribut « value » s'il existe, sinon sa valeur sera égale à la chaîne de caractères spécifiée entre les balises `<input>` et `</input>` ;
- la liste d'options fonctionne de la même façon pour la valeur de la variable `$_POST["demande"]`, l'attribut « name » étant déclaré dans la balise `<select>`

- nous utilisons la fonction **htmlspecialchars()** qui transforme certains caractères du texte entré par l'utilisateur dans la zone de texte en d'autres qui ne permettront pas l'exécution de code HTML par une personne malintentionnée ;
- nous employons la fonction **nl2br()** qui permet de conserver les sauts de ligne effectués dans la zone de texte

vosre adresse e-mail est : bruno.martin@ifosupwavre.be
Avez-vous souhaité vous inscrire sur la lettre d'informations ? oui
et sa description est la suivante :
Notes de cours pour webmaster.
pour la rubrique photos

Ces données pourraient ensuite être insérées dans une base de données pour y être conservées et exploitées par la suite.

2 Passage de variables dans l'URL : la méthode GET

Dans ce cas, les variables et les valeurs qu'elles prennent sont déclarées directement dans l'URL c'est à dire via la balise de lien HTML ``. Les variables sont ensuite exploitables sur la page cible en PHP.

La récupération des variables se fait via la syntaxe : `$\$variable = \$_GET['variableurl']$` au lieu de `$\$_POST$` précédemment.

Exercice : Les formulaires

- 1 Ecrivez un programme qui, en fonction d'un formulaire, vérifiera le chiffre « posté » et affichera le grade obtenu de l'étudiant. **form1.php – resultat1.php**

Quelle est votre moyenne?

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11

0≤moyenne <12 : refusé
12≤moyenne <14 : satisfaction
14≤moyenne <16 : distinction
16≤moyenne <18 : grande distinction
18≤moyenne <20 : la plus grande distinction

- 2 Réalisez un programme qui teste un login et un mot de passe transmis par un formulaire pour l'autorisation d'accès à une page. **form2.php – resultat2.php**

- 3 Affichez les résultats « postés » de votre formulaire sur une page php. **form3.php – resultat3 .php**

Prévoyez au moins 5 pays dans la liste déroulante...

- 4 Envoyez un Email en php (formulaire + utilisation de la fonction mail) **form4.php – resultat4.php**

- 5 Créez le formulaire suivant (**form5.html – resultat5.php**):

Celui sera renvoyé vers une page « remise.php », qui affichera la remise selon le tableau suivant

Si le type est un grossiste et qu'il a un brut hors taxes qui dépasse 10000 alors il a une remise de 10%
 Sinon Si le type est un particulier et qu'il a un brut hors taxes qui dépasse 15000 alors il a une remise de 30%
 Sinon Si le type est un détaillant et qu'il a un brut hors taxes qui dépasse 10000 alors il a une remise de 15%

- 6 **form6.php – resultat6.php**

Prévoir un formulaire

Avec 2 champs :

- Âge (liste déroulante de 1 à 99)
- Jour de la semaine (liste déroulante avec tous les jours de la semaine)

Ce formulaire redirigera vers une page (formulaire6.6.php) et affichera le prix d'entrée dans le parc d'attraction, sachant que :

- Les enfants de 3 à 11 ans paient 27€ en semaine et 29€ le we.
- Les personnes de 12 à 55 ans paient 31€ quelque soit le jour de la semaine
- Les autres ne paient pas l'entrée.

- 7 Scouts - Ecrivez un programme qui en fonction d'une variable informera l'enfant de sa catégorie **form7.php – resultat7.php**

- Moins de 5 ans : trop petit !
- Baladins: de 5 à 8 ans
- Louveteaux : de 9 à 12 ans
- Scout : de 13 à 16 ans
- Pionniers : après 17 ans

- 8 **form8.php – resultat8.php**. Réalisez le formulaire suivant :

Celui-ci redirigera vers une page où vous devez récupérer les données et vérifier si le titre n'est pas vide et que l'URL commence bien par «http:// ».

Pour ce faire aidez-vous des deux fonctions ci-dessous :

- la fonction `empty()` permet de contrôler si un champ est vide ou non
- la fonction `substr()` permet de sélectionner une sous chaîne dans une chaîne (*par exemple, les 7 premiers caractères de l'URL.... Qui doivent être égale à « http:// »*)

<http://be2.php.net/manual/fr/function.empty.php>

<http://be2.php.net/manual/fr/function.substr.php>

- 9 **form9.php – resultat9.php**

Prévoir un formulaire Avec 3 champs :

- Nom de l'article
- Prix HTV
- Code TVA (prévoir pour ce champ une liste déroulante de 1 à 3)

Ce formulaire redirigera vers une page et affichera en fonction du code TVA :

CODE TVA1 = 2,10%

CODE TVA2 = 5,50%

CODE TVA3 = 19,60%

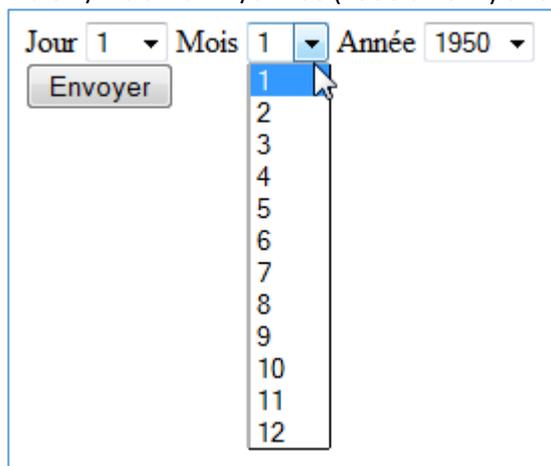
Le nom de l'article + son prix TVAC

Exemple :

Affichera à l'écran :

Ecran PRIX TVAC = 470.75155

- **10 form10.php – resultat10.php** Créez un formulaire qui permet de choisir dans des listes déroulantes (jour 1 à 31 /mois 1 à 12 /année (1950 à 2014) une date :



The image shows a web form with three dropdown menus for date selection. The first dropdown is labeled 'Jour' and has '1' selected. The second dropdown is labeled 'Mois' and has '1' selected, with a mouse cursor hovering over it, and a list of numbers from 1 to 12 is displayed below it. The third dropdown is labeled 'Année' and has '1950' selected. Below the dropdowns is a button labeled 'Envoyer'.

Affichez ensuite cette date à l'écran