
Formulaires et scripts PHP

1. Les formulaires

1.1 Création d'un formulaire HTML

La création d'un formulaire nécessite la connaissance de quelques balises HTML indispensables. Un formulaire commence toujours par la balise `<form>` et se termine par la balise `</form>`.

```
<form method="post" action "fichier.php">
// corps du formulaire
</form>
```

- L'attribut **action** indique l'URL du fichier qui traitera les données du formulaire. Entre ces deux balises se situent les balises qui créeront les différents types de champs que contiendra le formulaire.
- L'attribut **enctype** détermine le type de données envoyées par le formulaire: ce sont des informations textuelles ou des fichiers joints. Cet attribut est optionnel et vaut par défaut "application/x-www-form-urlencoded". Par contre, si l'on souhaite attacher des fichiers, il faut attribuer la valeur "multipart/form-data".
- L'attribut **id** est utilisé pour des manipulations de DOM à l'aide de scripts tels que JavaScript.
- La méthode d'envoi **method** prend les valeurs *get* ou *post*. La méthode *get* ajoute les noms des variables et leur valeur à l'URL du script chargé de les traiter. Elle a pour inconvénient de rendre visibles les données dans la barre d'adresse du navigateur. De plus, la longueur totale est limitée à 255 caractères, ce qui rend impossible la transmission d'un volume de données important. On lui préférera donc la méthode *post* avec laquelle le volume des données n'est pas limité et qui présente une plus grande garantie de confidentialité des données.
- La balise `<input>` permet de créer des composants d'aspect et de rôle très différents. La différenciation de ces composants s'effectue en définissant la valeur de leurs attributs, et notamment de l'attribut **type**. L'attribut **name** est obligatoire, car c'est lui qui permet d'identifier les champs côté serveur et ainsi de récupérer leur contenu.:

- **Champ de saisie de texte uniligne :**

```
<input type="text" name="nom_du_champ" value="chaîne" size="30" >
```

- **Champ de saisie de mot de passe :**

```
<input type="password" name="code" size="30">
```

- **Boutons d'envoi et d'effacement :**

```
<input type="submit" value="Envoyer">
```

```
<input type="reset" name="efface" value="Effacer">
```

- **Case à cocher et bouton radio :**

```
<input type="checkbox" name="case1" value="valeur_case">
```

```
<input type="radio" name="radio1" value="valeur_radio">
```

- **Liste de sélection avec options à choix unique :**

```
<select name="select1" size="1">
```

```
<option value="un">choix1 </option>
```

```
<option value="deux">choix2 </option>
```

```
</select>
```

- **Liste de sélection avec options à choix multiples :**

```
<select name="select" size="1" multiple>
```

```
<option value="un">choix1 </option>
```

```
<option value="deux">choix2 </option>
```

```
</select>
```

1. Récupération des données dans PHP

Lors de la soumission du formulaire, PHP crée autant de variables globales qu'il y a de champs nommés dans le formulaire et leur attribue comme nom la valeur de l'attribut HTML `name="..."` défini pour chaque champ et comme valeur, la valeur de l'attribut HTML `value="..."`. Le script PHP désigné dans l'attribut *action* recevra ces noms de variables et leur valeur pour leur appliquer un traitement éventuel.

Exemple:

Le formulaire (form.html) de la page d'accueil va proposer à l'utilisateur deux champs à remplir. Appelons les nom et prenom. Dans la balise `<form>` l'attribut *action* a pour valeur "resultat.php", indiquant que c'est ce fichier qui traitera les informations.

Nom ?

Prenom ?

Abonné ?

Figure 1

Quand l'utilisateur aura complété les champs du formulaire et cliqué sur le bouton d'envoi, PHP transmettra au fichier resultat.php trois variables globales nommées \$nom, \$prenom, et \$abonne, dont on pourra alors utiliser les valeurs.

form.html	resultat.php
<pre> <html> <head> <title>Saisie des données</title> </head> <body> <form method="get" action="resultat.php"> Nom ? <input type="text" size=15 name="nom"></br> Prenom ? <input type="text" size=15 name="prenom"> </br>
 Abonné ? <input type="checkbox" name="abonne" value="oui">
 <input type="submit" value="Envoyer" name="submit"> </form> </body> </html> </pre>	<pre> <html> <head> <title> ACCUEIL </title> </body> <?php echo "Bonjour ", \$_GET["prenom"], " ", \$_GET["prenom"],"
"; if (\$_GET["abonne"]=="oui") { echo "Vous bénéficiez d'une remise de 10%";} else { echo "Devenez membre M.", \$nom ; } ?> </body> </html> </pre>

Quelque soit la méthode de soumission choisie, le formulaire enverra ses informations à une page de destination spécifiée grâce à la propriété *action*. Un script PHP permet de gérer les informations saisies dans un formulaire HTML. Il est possible d'accéder à des variables qui ont été prédéfinies. Celles ci peuvent contenir des informations sur le serveur web, sur le navigateur utilisé par le client, etc. Il est possible d'accéder aux paramètres passés par la méthode GET en appelant le tableau `_GET[]` et en accèdent à la valeur ayant pour clé le nom du paramètre (spécifié dans l'attribut *name*)

Superglobale	Description
<code>\$ _GET</code>	contient les variables passées par la méthode GET
<code>\$ _POST</code>	contient les variables passées par la méthode POST

Prenons l'exemple de la figure 1 (form1.html) et affichons les variables `_GET["nom"]`, `_GET["prenom"]`, `_GET["abonne"]`, `$nom`, `$prenom`, et `$abonne`. Le script d'affichage pourra alors être : (resultat1.php)

```
<html>
<body>
<h1>Informations saisies</h1>
<?php
    // Récupération de données passées par la méthode GET
    echo " _GET: [\ "nom\ "] = ", $_GET["nom"], "<br/>";
    echo " _GET[\ "prenom\ "] = ", $_GET["prenom"], "<br/>";
    echo " _GET[\ "abonne\ "] = ", $_GET["abonne"], "<br/>";
?>
</body>
</html>
```

Le résultat est :

```
_GET:["nom"]= chahir
_GET:["prenom"]= youssef
_GET:["abonne"]= oui
```

Vous l'aurez compris, s'il s'agissait d'un formulaire utilisant la méthode POST, il aurait fallu faire appel au tableau `$_POST`. De la même manière, on peut manipuler toute autre information associée à un élément d'un formulaire (radio, select, textarea).

2. PHP et formulaires :

2.1 Radio buttons

Le principe est le même pour tout autre élément qu'un champ texte (radio button, case à cocher, liste de sélection) d'un formulaire, encore faut-il savoir comment ces derniers réagissent.

Prenons l'exemple des "radio buttons". Il est possible de préciser la valeur associée à un radio button (par défaut **on**) en utilisant l'attribut **value**. Il est ainsi possible d'avoir le formulaire suivant.

```
<html> <body>
  <form action=" formulaire_moral.php " method="get" >
    Etes-vous ?<br />
    <input type="radio" name="moral" value="supercontent" />
    Super content<br />
    <input type="radio" name="moral" value="content" />
    Content<br />
    <input type="radio" name="moral" value="pascontent" />
    Pas content<br />
    <input type="radio" name="moral" value="demauvaisehumeur" />
    De mauvaise humeur<br />
    <input type="submit" value="Ma réponse"/>
  </form> </body></html>
```

Tous les radio buttons d'un même groupe possèdent le même nom mais la valeur associée varie de l'un à l'autre. Pour connaître le moral du visiteur, il suffit alors de faire appel à `$_GET["moral"]` dans `formulaire_moral.php` :

```
<?php
    // Récupération de paramètres passés par la méthode GET
    // avec PHP>=4.1
    echo "<b>Vous êtes </b><br />", $_GET["moral"];
?>
```

2.2 Checkboxes (cases à cocher)

Les cases à cocher fonctionnent à peu près sur le même principe que les "radio buttons" mais la différence majeure c'est qu'il est cette fois possible d'en cocher plusieurs.

Tout comme pour les "radio buttons", il est possible de préciser la valeur associée à une case à cocher (par défaut **on**) en utilisant l'attribut **value**. Pour y avoir accès, on utilise un suffixe "[]" à l'attribut name dans le formulaire HTML. Ce suffixe permet de stocker dans un tableau PHP toutes les valeurs sélectionnées parmi les cases à cocher.

```
<html>
  <body>
    <form action="formulaire_checkbox.php">
      Qu'est-ce que vous aimer ?<br />
      <input type="checkbox" name="loisirs[]" value="shopping" />
      Faire du shopping<br />
      <input type="checkbox" name="loisirs[]" value="sport" />
      Faire du sport<br />
      <input type="checkbox" name="loisirs[]" value="theatre" />
      Aller au théâtre<br />
      <input type="checkbox" name="loisirs[]" value="cinema" />
      Aller au cinéma<br />
      <input type="submit" value="Ma réponse"/>
    </form>
  </body>
</html>
```

Dans ce cas, `$_GET["loisirs"]` est un tableau contenant les valeurs des cases cochées. Il suffit alors de le parcourir comme dans l'exemple suivant: (formulaire_checkbox.php)

```
<?php
  $loisirs = $_GET["loisirs"];
  if (sizeof($loisirs) !=0) {
    foreach ($loisirs as $lois) {
      echo $lois."<br />";
    }
  }
?>
```

2.3 Select (liste de sélection)

Liste de sélection unique

Les listes de sélection (à sélection unique) fonctionnent à peu près sur le même principe que les "radio buttons" la seule différence vient du fait que la valeur par défaut n'est pas **on** mais le label de l'option sélectionnée. (formulaire_selection_unique1.html)

```
<html> <body>
  <form action="formulaire_moral.php" method="get" >
    Etes-vous ?<br />
    <select name="moral">
      <option>Super content</option>
      <option>Content</option>
      <option>Pas content</option>
      <option>De mauvaise humeur</option>
    </select><br />
    <input type="submit" value="Ma réponse"/>
  </form>
</body></html>
```

Il est possible de savoir quelle option a été sélectionnée en faisant appel à `$_GET["moral"]`. Toutefois travailler avec les labels n'est pas toujours chose très pratique (notamment si votre site est disponible en différentes langues). Heureusement, il est là, encore possible de préciser la valeur associée à chaque option (ici, nous associons une note à chaque option).
(formulaire_selection_unique2.html)

```
<html> <body>
  <form action="formulaire_moral1.php" method="get" >
    Etes-vous ?<br />
    <select name="moral">
      <option value="3">Super content</option>
      <option value="2">Content</option>
      <option value="1">Pas content</option>
      <option value="0">De mauvaise humeur</option>
    </select><br />
    <input type="submit" value="Ma réponse"/>
  </form>
</body></html>
```

Et le script `formulaire_moral1.php` peut être :

```
<?php
$moral = $_GET["moral"];
switch ($moral
{
  case 3: $moral="Super content ";break;
  case 2: $moral="Content"; break;
  case 1: $moral="Pas content";break;
  case 0: $moral="De mauvaise humeur";break;
}
echo "<b>Vous êtes</b><br />$moral ";
?>
```

Liste de sélection multiple

Tout comme pour les cases à cocher, la meilleure façon de gérer les liste de selection multiple consiste à donner à la liste un nom de tableau comme suit: (formulaire_select_multiple.html)

```
<html>
  <body>
    <form action="formulaire_select_multiple.php">
      Qu'est-ce que vous aimez ?<br />
      <select name="loisirs[]" multiple="multiple">
        <option value="shopping">Faire du shopping</option>
        <option value="sport">Faire du sport</option>
        <option value="theatre">Aller au théâtre</option>
        <option value="cinema">Aller au cinéma</option>
      </select>
      <input type="submit" value="Ma réponse"/>
    </form>
  </body>
</html>
```

Dans ce cas, `$_GET["loisirs"]` est un tableau contenant les valeurs des cases cochées. Il suffit alors de le parcourir comme dans l'exemple suivant: (formulaire_select_multiple.php)

```

<?php
    // $_GET["loisirs"] contient un tableau de valeurs
    $loisirs = $_GET["loisirs"];

    echo "<b>Vous aimez </b><br />";
    for ($i=0; $i<count($loisirs); $i++) {
        echo $loisirs[$i]."<br />";
    }
?>

```

L'ensemble des informations transmises au script PHP appelé par le formulaire sera associé à une seule variable soit `$_GET` ou `$_POST`, de type tableau associatif. A l'aide de ce tableau, on connaît à la fois la valeur d'un champ et le nom associé à l'information.

```

<?php
    /* On peut tester le contenu de cette variable
       Ici c'est la méthode post */

    while (list($cle,$valeur)=each($_POST)) {
        echo $cle, " - ", $valeur,"<br/>";
        if (is_array($valeur)){
            echo " Taille :", count($valeur),"<br/>";
            while (list($cle1,$valeur1)=each($valeur)) {
                echo $cle1, " :\t", $valeur1,"<br/>";
            }
        }
    }
?>

```

3. Gestion de formulaires

3.1 Saisie des données et gestion des erreurs

L'affichage des éléments transmis ne se fait que si tous les éléments ont été renseignés, sinon les éléments manquants sont précisés. On affichera, alors, dans le formulaire les informations manquantes et celles qui sont déjà saisies. Pour cela, on va accéder directement au formulaire par le script php et on va gérer les éléments manquants dans la page du formulaire. De même la gestion des éléments cochés (ou les boutons pressés) sera traitée de la même manière, en gardant l'état des cases.

Chaque message d'erreur est rangé dans un tableau `$erreur`, dont les champs correspondent aux champs du formulaire. Le contrôle du formulaire commence par un test des erreurs possibles et affichage des messages d'erreur correspondants. Ensuite, on réaffiche la page du formulaire avec le contenu des champs déjà remplis correctement (variables déjà saisies et les boutons déjà cochés). Pour cela, on utilise un tableau `$coche` qui servira à mémoriser l'état des boutons et/ou des cases à cocher.

Par exemple, si on veut afficher l'état d'un bouton radio ou de liste, on va procéder de la manière suivante :

a) Radio buttons et liste de sélection unique:

Le script contient :

```
$coche["Nom_Bouton"]="CHECKED";      et      $coche["Nom_Liste"]="SELECTED";
```

Lorsque le formulaire sera renvoyé au navigateur, `<?php echo $coche["..."]; ?>` inscrira "CHECKED" (resp. "SELECTED") pour la case déjà remplie ce qui cochera la case (resp. sélectionne la ligne), ou n'inscrira rien (chaîne vide) pour les cases qui n'ont pas été remplies.

Exemple 1: (Boutons radio)

Script html :

```
...
<input type="radio" name="moral" value="supercontent" <?php echo $coche["supercontent"]; ?
>/>
Super content<br />
<input type="radio" name="moral" value="content" <?php echo $coche["content "]; ?>/>
Content<br />
<input type="radio" name="moral" value="pascontent" <?php echo $coche["pascontent"]; ?>/>
Pas content<br />
<input type="radio" name="moral" value="demauvaisehumeur" <?php echo
$coche["demauvaisehumeur"]; ?>/>
De mauvaise humeur<br />
...
```

Script php:

```
<?php
//memorise l'état du bouton coché
$moral=$_GET["moral"]; $coche[$moral]="CHECKED";
echo "<b>Vous êtes </b><br />",$moral;
?>
```

Exemple 2: (Liste de sélection unique)

Script html :

```
...
<select name="sexe">
  <option <?php echo $coche["Monsieur"]; ?>>Monsieur</option>
  <option <?php echo $coche["Madame"]; ?>>Madame</option>
  <option <?php echo $coche["Mademoiselle"]; ?
>>Mademoiselle</option>
</select>
...
```

Script php:

```
<?php
//memorise l'état du bouton coché
$sexe=$_GET["sexe"];
$coche[$sexe]="SELECTED";
echo "<b>Vous êtes </b><br />",$sexe;
?>
```

b) Checkboxes et Liste de sélection multiple :

Comme les noms des éléments choisis ont été rangés dans un tableau, on va utiliser une boucle comme ceci :

```
foreach ($Tab_case as $case) { $coche[$case]="CHECKED"; }
et
foreach ($Tab_lig as $ligne) { $coche[$ligne]="SELECTED"; }
```

Lorsque le formulaire sera renvoyé au navigateur, <?php echo \$coche["..."]; ?> inscrira "CHECKED" (resp. "SELECTED") pour toutes les cases choisies (resp. les lignes sélectionnées) ce qui cochera les case (resp. sélectionne la ligne), ou n'inscrira rien (chaîne vide) pour les cases qui n'ont pas été remplies.

Exemple 1: Checkbox

Script.html :

```
....
  Qu'est-ce que vous aimer ?<br />
  <input type="checkbox" name="loisirs[]" value="shopping" <?php echo
$coche["shopping"]; ?>/>
  Faire du shopping<br />
  <input type="checkbox" name="loisirs[]" value="sport" <?php echo $coche["sport"]; ?
>/>
  Faire du sport<br />
  <input type="checkbox" name="loisirs[]" value="theatre" <?php echo
$coche["theatre"]; ?>/>
  Aller au théâtre<br />
  <input type="checkbox" name="loisirs[]" value="cinema" <?php echo
$coche["cinema"]; ?>/>
  Aller au cinéma<br />
...
```

Script.php

```
<?php
  $loisirs = $_GET["loisirs"];
  echo "<b>Vous aimez </b><br />";

  foreach ($loisirs as $loisir) {
    $coche[$loisir]="CHECKED";
    echo $loisir."<br />";
  }
?>
```

3.2) Démarche générale :

I) On accède directement au formulaire par le script php (Formulaire.php) dans lequel on va :

- 1) Inclure le formulaire si c'est la première fois. Pour cela, on peut tester l'existence des variables correspondant aux noms du formulaire (*)

```
if ( (!isset($_GET["Titre"]))&&(!isset($_GET["Extrait"])) )
  include("Formulaire.html");
```

- 2) Sinon, il faut d'abord récupérer les données saisies grâce au tableau \$_GET ou \$_POST et gérer les messages d'erreur. Chaque message d'erreur est rangé dans un tableau \$erreur, dont les champs correspondent aux champs du formulaire.

```
$Titre=$_GET["Titre"];
$Titre=trim($Titre);
if ($Titre=="") {
  $erreur["Titre"]="il manque un titre";
}
...
```

On prévoit l'affichage des messages d'erreur dans la page html du formulaire.

- 3) Actualiser dans le tableau \$case les cases déjà cochées et sélectionnés.

```

if (count($choix) == 0) {
    $erreur["choix"] = "vous n'avez pas fait de choix";
}
else {
    foreach ($choix as $case) {    $coche[$case]="CHECKED"; }
}

```

- 4) On compte le nombre d'erreur (nombre d'éléments du tableau \$erreur). S'il est supérieur à 0, c'est qu'il y a des erreurs et donc il faut recharger le formulaire. Sinon, toutes les données sont correctement saisies et le script peut appeler d'autres fonctions d'affichage ou de traitement de données !

```

if (count($erreur) == 0) {    include("Traitement.php"); }
else {
    include("Formulaire.html");
}

```

Formulaire.php :

```

<?php
if ( (!isset($_GET["Titre"]))&&(!isset($_GET["Extrait"])) )
    include("Formulaire.html");
else{
    $Titre=$_GET["Titre"];
    $Titre=trim($Titre);
    if ($Titre=="") {
        $erreur["Titre"] ="il manque un titre";
    }
    $Extrait=$_GET["Extrait"];
    $Extrait=trim($Extrait);
    if ($Extrait==""){
        $erreur["Extrait"] ="il manque un extrait";
    }
    $choix=$_GET["choix"];
    if (count($choix) == 0) {
        $erreur["choix"] = "vous n'avez pas fait de choix";
    }
    else {
        for ($i=0;$i<sizeof($choix);$i++) {
            $c=$choix[$i];$coche[$c]="CHECKED";
        }
    }
    if (count($erreur) == 0) {
        $_GET["Extrait"]=nl2br($Extrait);
        $corps = "<table bgcolor=#cccccc border>";
        $corps .="<tr><th>Elément</th><th>Valeur</th></tr>";
        while (list($cle,$val) =each($_GET)) {
            $corps .="<tr><td><u>" . $cle ;
            $corps .="</u></td><td><b>". $val. " </b></td></tr>";
            if (is_array($val)) {
                $corps .="<tr><td>". (" .count($val) .")</td><td>";
                while (list($clebis,$valbis) =each($val)) {
                    $corps .=$valbis."<br/>";
                }
            }
        }
    }
}

```

```

        $corps .="</td></tr>";
    }
}
$corps .="</table>";
echo $corps;
}
else {
    include("Formulaire.html");
}
}
?>

```

Supposons qu'on veut imposer une sélection d'au moins trois choix. Dans ce cas, il faut tester si (count(\$choix) > 3) Ce qui donne :

```

if (count($choix) < 3) {
    $erreur["choix"] = "vous n'avez pas fait de choix";
}
else {
    ...
}

```

II) Dans la page html du formulaire (Formulaire.html), On commence par l'affichage des messages d'erreur dans le cas où count(\$erreur) est différent de 0. Ensuite, on réaffiche la page du formulaire avec le contenu des champs déjà remplis correctement

Formulaire.html :

```

<html>
  <head><title>Gestion de formulaires</title></head>
  <body>
    <?php
      if (isset($erreur) && count($erreur) != 0)
      {
        while (list($cle, $val) = each($erreur)) {
          echo $cle, " : \t", $val, "<br/>";
        }
      }
    ?>
    <h2>Formulaire</h2>
    <form action="Formulaire.php" method="get" name="formGet">
      <u>Saisir un titre :</u><br>
      <input type="text" name="Titre" size="20"
        value="<?php echo $Titre; ?>" <br/>
      <u>Extrait :</u><br>
      <textarea name="Extrait" cols="30" rows="4">
        <?php echo $Extrait; ?>
      </textarea><br/>
      <u>Choix :</u><br>
      <input type="checkbox" name="choix[]" value="livre"
        <?php echo $coche["livre"]; ?>>livre<br/>
      <input type="checkbox" name="choix[]" value="roman"
        <?php echo $coche["roman"]; ?>>roman<br/>
      <input type="checkbox" name="choix[]" value="périodique"
        <?php echo $coche["périodique"]; ?>>périodique<br/>
      <input type="checkbox" name="choix[]" value="web"
        <?php echo $coche["web"]; ?>>web<br/>
      <input type="submit" name="Soumettre" value="Soumission">
    </form></body>

```