

PHP

Manipulation des formulaires avec PHP

Des formulaires pour faire quoi ?

- Les formulaires HTML permettent :
 - De saisir des renseignements en vue d'un traitement
 - Calculs
 - Interrogation d'une base de données
 - Etc;
 - De transmettre des données entre les pages du site
- Ils sont indispensables pour avoir des sites interactifs :
 - Création d'un espace membre
 - Personnalisation du contenu d'un site par un client
 - Interaction avec un visiteur
 - Création d'une page de recherche
 -

PHP – Les Formulaires

CRÉATION

Déclaration d'un formulaire

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
    <title>Page vide</title>
    <meta http-equiv="content-type" content="text/html; charset=utf-8" />
    <meta http-equiv="Content-Style-Type" content="text/css" />
</head>


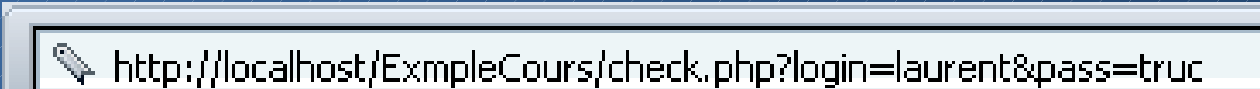
<body>
```

```
<form action="form_send.php" method="post">
    ....
</form>
```

```
</body>
</html>
```

Méthode d'envoi et traitement

```
<form action="traitement_formulaire.php" method="post">  
...  
</form>
```

- ACTION :
Chemin vers le fichier qui va traiter les données envoyées par le formulaire.
- METHOD :
 - *POST* : envoie les variables dans l'entête de la requête HTTP (donc invisible)

 - *GET* : envoie les variables à la suite de l'adresse (donc visible)


Les champs texte

- Sur une ligne : balise *input* avec ***type="text"***

```
<form action="traitement_formulaire.php" method="post">
  <p>
    Entrez votre nom :<br />
    <input name="nom" type="text" value="?" size="25" /><br />
    <input type="submit" value="valider" /><br />
  </p>
</form>
```

Entrez votre nom :

- Mot de passe : balise *input* avec ***type="password"***

```
<form action="traitement_formulaire.php" method="post">
  <p>
    Entrez votre mot de passe :<br />
    <input type="password" name="mdp" value="" size="25" /><br />
    <input type="submit" value="valider" /><br />
  </p>
</form>
```

Entrez votre mot de passe :



Attention : les objets de formulaire doivent être placés entre les balises `<form></form>`

Zone de texte

- Créer une zone de texte sur plusieurs lignes : balise *textarea*

```
<form action="traitement_formulaire.php" method="post">
  <p>
    Entrez votre commentaire :<br />
    <textarea name="comment" cols="35" rows="5">
      Ceci est un commentaire
      sur plusieurs
      lignes.
    </textarea>
    <br /><input type="submit" value="valider" /><br />
  </p>
</form>
```

Entrez votre commentaire :

Ceci est un commentaire
sur plusieurs
lignes.

valider

Boutons de formulaire

- Créer des boutons :
balise *input* avec l'un des types suivants :
type="submit", type="reset", type="button"

```
<form action="form_send.php" method="post">
  <p>
    Entrez votre e-mail :<br />
    <input type="text" name="mail" size="35" /><br />
    <input type="submit" name="envoyer" value="Valider" />
    <input type="reset" name="RAZ" value="Effacer" />
    <input type="button" name="Annu" value="Annuler" />
  </p>
</form>
```

Entrez votre e-mail :


Les boutons de type *submit* et *reset* doivent obligatoirement être placés entre les balises `<form>` et `</form>`, sinon ils n'auront aucun effet

Image cliquable

- Créer un bouton de validation avec une image : balise ***input*** avec ***type="image"***

```
<form action="form_send.php" method="post">
<p>
  Entrez votre e-mail :<br />
  <input type="text" name="mail" size="35" />
  <input type="image" name="img_send" src="images/send.gif" alt="logo"/>
</p>
</form>
```

Entrez votre e-mail :



(img_send_x, img_send_y)

Lorsque l'on clique sur l'image, les données du formulaire sont transmises ainsi que les coordonnées du point cliqué accessible par `name_x` et `name_y`.

Cases à cocher et boutons radio

- Définir une case à cocher :
balise *input* avec *type="checkbox"*

```
<form action="./form.php" method="post">
  <p>
    Recevoir les news par e-mail
    <input type="checkbox" name="q1" value="oui" checked="checked" />
    <br /><input type="submit" value="Envoyer" /><br />
  </p>
</form>
```

Recevoir les news par e-mail ☒

- Boutons radio :
balise *input* avec *type="radio"*

```
<form action="./form.php" method="post">
  <p>
    Vous êtes :
    <br /><input type="radio" name="genre" value="h" checked="checked" /> Un homme
    <br /><input type="radio" name="genre" value="m" /> Une femme
    <br /><input type="radio" name="genre" value="" /> Autre...
    <br /><input type="submit" value="Soumettre" /><br />
  </p>
</form>
```

Vous êtes :

☒ Un homme

☐ Une femme

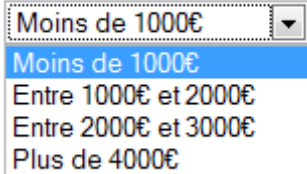
☐ Autre...

Liste déroulante / Liste de sélection

- Définir une liste déroulante : balise ***select***

```
<form action="./form.php" method="post">
  <p>Vous gagnés : <br />
  <select name="salaire">
    <option value="1">Moins de 1000€</option>
    <option value="2">Entre 1000€ et 2000€</option>
    <option value="3">Entre 2000€ et 3000€</option>
    <option value="4">Plus de 4000€</option>
  </select>
</p>
</form>
```

Vous gagnés :



Moins de 1000€

Entre 1000€ et 2000€

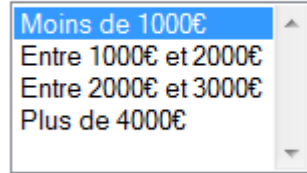
Entre 2000€ et 3000€

Plus de 4000€

- Liste de sélection : attribut ***size***

```
<form action="./form.php" method="post">
  <p>Vous gagnés : <br />
  <select name="salaire" size="5">
    <option value="1">Moins de 1000€</option>
    <option value="2">Entre 1000€ et 2000€</option>
    <option value="3">Entre 2000€ et 3000€</option>
    <option value="4">Plus de 4000€</option>
  </select>
</p>
</form>
```

Vous gagnés :



Moins de 1000€

Entre 1000€ et 2000€

Entre 2000€ et 3000€

Plus de 4000€

Liste multiple

- Définir une liste de sélection multiple : balise ***select*** avec les attributs ***size*** et ***multiple***

```
<form action="./form.php" method="post">
  <p>Selectionnez les langages que vous connaissez :<br />
  <select name="id_langage[]" multiple size="6">
    <option value="1">PHP</option>
    <option value="2">JAVA</option>
    <option value="3">Javascript</option>
    <option value="4">C</option>
    <option value="5">C++</option>
    <option value="6">C#</option>
  </select>
</p>
</form>
```

Selectionnez les langages que vous connaissez :

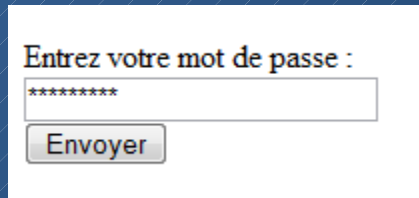


Attention au nom de l'identifiant, c'est un choix multiple, il faut donc utiliser un tableau, d'où l'utilisation des crochets []

Champ caché

- Les champs cachés servent à envoyer une valeur sans que l'utilisateur ne le voit et sans qu'il puisse la modifier.
- Définir un champ caché :
balise ***input*** avec ***type="hidden"***

```
<form action="./forme.php" method="post">
  <p>
    Entrez votre mot de passe :<br />
    <input type="hidden" name="encryption" value="md5" />
    <input type="password" name="mdp" value="" size="25" /><br />
    <input type="submit" value="Envoyer" /><br />
  </p>
</form>
```



Entrez votre mot de passe :

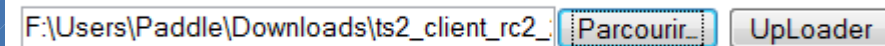
Envoyer

Envoi de fichiers

- Le code suivant permet à l'utilisateur d'envoyer un fichier disponible sur son ordinateur.

```
<form action="upload.php" method="post" enctype="multipart/form-data">
  <p>
    <input type="file" name="fichier" size="40" />
    <input type="submit" value="UpLoader" />
  </p>
</form>
```

- Pour cela, on utilise la **balise *input* avec *type="file"***



Il est nécessaire de spécifier dans la balise *form* qu'il ne s'agit pas de données textuelles. C'est le rôle l'instruction suivante : ***enctype="multipart/form-data"***

PHP – Les Formulaires

TRAITEMENT DES DONNÉES AVEC PHP

Utilisation des superglobales

- Les données envoyées par un formulaire se retrouvent dans des superglobales.
- Les superglobales sont des tableaux associatifs dont la portée est globale.
- Pour récupérer les informations d'un formulaire, on peut utiliser l'une des superglobales suivantes :
 - **\$_GET[]** : données envoyées via l'URL
 - **\$_POST[]** : données envoyées via l'entête HTTP
 - **\$_FILES[]** : informations sur les fichiers envoyés par le visiteur.
 - **\$_REQUEST[]** : fusion de **\$_GET[]**, **\$_POST[]** et **\$_COOKIE[]**. En cas de conflits, **\$_COOKIE[]** est prioritaire, puis **\$_POST[]**.

Exemple de récupération d'une donnée

● Fichier *form_mail.html* : création du formulaire

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Page vide</title>
  <meta http-equiv="content-type"
    content="text/html; charset=utf-8" />
  <meta http-equiv="Content-Style-Type" content="text/css" />
</head>

<body>

  <form action="form_mail.php" method="post">
    <p>
      Entrez votre e-mail :<br />
      <input type="text" name="mail" size="35" />
      <input type="submit" name="send_mail" value="Envoyer" />
    </p>
  </form>

</body>
</html>
```

Entrez votre e-mail :

Envoyer

Exemple de récupération d'une donnée

- Fichier *form_mail.php* : traitement des données

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Page vide</title>
  <meta http-equiv="content-type"
    content="text/html; charset=utf-8" />
  <meta http-equiv="Content-Style-Type" content="text/css" />
</head>

<body>

<?php
  if (isset($_POST['send_mail'])) {
    echo "<p>Votre mail est " . $_POST['mail'] . "</p>";
  } else {
    echo '<p>Utilisez la page <a href="./form_mail.html">form_mail.html</a>
    pour renseigner votre mail</p>';
  }
?>

</body>
</html>
```

\$_REQUEST[], \$_POST[] ou \$_GET[] ?

- **\$_POST[]** : si les données du formulaire sont transmises par la méthode *post*
- **\$_GET[]** : si les données du formulaire sont transmises par la méthode *get*
- **\$_REQUEST[]** : utilisable quelque soit la méthode de passage utilisée, à la condition que l'on soit certain qu'il n'y ait pas de collisions possibles *ie* deux variables de même nom.

Avantage : On peut modifier la façon de passer les données sans avoir à modifier le code de la page PHP qui les traitent. Cette méthode est préférable aux deux autres dès lors que l'on sait exactement ce que l'on fait.

Traitement du texte

- Quelques fonctions utiles pour traiter les données venant d'un champ texte :
 - Les fonctions ***addslashes()*** et ***stripslashes()*** permettent, respectivement, d'ajouter et de supprimer les backslashes ("\\") devant les caractères le nécessitant (guillemets simples : ', guillemets doubles : ", backslash : \, valeur NULL)
 - La directive PHP *magic_quotes_gpc* est activée par défaut, et elle appelle *addslashes()* sur toutes les données GET, POST et COOKIE.
 - La fonction *get_magic_quotes_gpc()* retourne TRUE si la directive *magic_quotes* est activée, FALSE sinon.
 - La fonction ***nl2br()*** transforme les retours à la ligne (\n) en "
"
 - La fonction ***htmlspecialchars()*** transforme les caractères interprétables afin qu'ils soient affichés et non analysés.

Exemple : *form_textarea.html*

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Page vide</title>
  <meta http-equiv="content-type"
    content="text/html; charset=utf-8" />
  <meta http-equiv="Content-Style-Type" content="text/css" />
</head>

<body>

  <form action="form_textarea.php" method="post">
    <p>
      Entrez votre texte :<br />
      <textarea name="comment" cols="35" rows="6">
        Si j'écris comme ça en sautant des
        lignes et en utilisant des apostrophes
        la phrase b<a est vrai mais pas b>a,
        je peux avoir des problèmes.
      </textarea>
      <br /><input type="submit" name="send_comment" value="Envoyer" />
      <input type="reset" name="reset_comment" value="RaZ" />
    </p>
  </form>

</body>
</html>
```

Entrez votre texte :

Si j'écris comme ça en sautant des
lignes et en utilisant des
apostrophes
la phrase b<a est vrai mais pas b>a,
je peux avoir des problèmes.

Envoyer

RaZ

Exemple : *form_textarea.php* simpliste !

```
<?php
if (isset($_REQUEST['send_comment'])) {
    $texte=$_REQUEST['comment'];

    echo "<p>".$texte."</p>";
} else {
    header('location: form_textarea.html');
}

?>
```


Code HTML généré par *form_textarea.php*

- Le code précédent génère le code suivant :

```
<p>Si j\'écris comme ça en sautant des  
lignes et en utilisant des apostrophes  
la phrase b<a est vrai mais pas b>a,  
je peux avoir des problèmes.  
</p>
```

- Ce qui produit cet affichage dans le navigateur :

Si j\'écris comme ça en sautant des lignes et en utilisant des apostrophes la phrase ba, je peux avoir des problèmes.

Exemple : *form_textarea.php* corrigé !

```
<?php
    if (isset($_REQUEST['send_comment'])) {
        if (get_magic_quotes_gpc()) $texte=stripslashes($_REQUEST['comment']);
        else $texte=$_REQUEST['comment'];

        $texte=htmlentities($texte, ENT_COMPAT, "UTF-8");
        echo nl2br("<p>".$texte."</p>");
    } else {
        header('location: form_textarea.html');
    }

?>
```

Code HTML généré par *form_textarea.php*

- Le code précédent génère le code suivant :

```
<p>Si j'&eacute;cris comme &ccedil;a en sautant des <br />
lignes et en utilisant des apostrophes <br />
la phrase b<lt;a est vrai mais pas b<gt;a,<br />
je peux avoir des probl&egrave;mes.<br />
</p>
```

- Ce qui produit cet affichage dans le navigateur :

Si j'écris comme ça en sautant des lignes et en utilisant des apostrophes la phrase b<a est vrai mais pas b>a, je peux avoir des problèmes.

Cases à cocher : *form_case.html*

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Page vide</title>
  <meta http-equiv="content-type"
    content="text/html; charset=utf-8" />
  <meta http-equiv="Content-Style-Type" content="text/css" />
</head>

<body>

  <form action="form_case.php" method="get">
    <p>Recevoir les news par e-mail
      <input type="checkbox" name="case_offre_on" value="ok" checked="checked" />
      <br /><input type="submit" value="Envoyer" /><br />
    </p>
  </form>

</body>
</html>
```

Case à cocher : *form_case.php*

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">

<head>
  <title>Page vide</title>
  <meta http-equiv="content-type"
    content="text/html; charset=utf-8" />
  <meta http-equiv="Content-Style-Type" content="text/css" />
</head>

<body>

<?php
  if (isset($_REQUEST['case_offre_on'])) {
    echo "La case est cochée et sa valeur est : ".$_REQUEST['case_offre_on'];
  } else {
    echo "La case n'est pas cochée !";
  }
?>

</body>
</html>
```

PHP – Les Formulaires

CRÉATION DYNAMIQUE

Génération d'un formulaire

- PHP peut envoyer au navigateur des balises HTML et donc, à fortiori des balises de formulaires.
- Il est donc bien évidemment possible de générer le formulaire dans le code PHP.

Exemple : *form_mail2.php*

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>Page vide</title>
  <meta http-equiv="content-type"
    content="text/html; charset=utf-8" />
  <meta http-equiv="Content-Style-Type" content="text/css" />
</head>

<body>

<?php
  //On teste si l'on vient du formulaire ou s'il faut le créer
  if (isset($_REQUEST['send_mail']) && (!empty($_REQUEST['mail'])))
  { //Si on vient du formulaire
    //On traite les données
    echo "<p>Votre mail est " . $_REQUEST['mail'] . "</p>";
  }
  else { //Sinon, génération de la page avec le formulaire
    echo '<form action="form_mail2.php" method="post">
      <p>Entrez votre e-mail :<br />
      <input type="text" name="mail" size="35" />
      <input type="submit" name="send_mail" value="Envoyer" /></p>
      </form>';
  }
?>

</body>
</html>
```

Ici, on utilise qu'un seul fichier PHP pour afficher et traiter le formulaire.

Avantage :

Cela permet, par exemple, de traiter les erreurs comme la saisie d'un mail non valide.

Exemple : *form_case2.php*

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>Page vide</title>
  <meta http-equiv="content-type"
    content="text/html; charset=utf-8" />
  <meta http-equiv="Content-Style-Type" content="text/css" />
</head>

<body>

<?php
  //On teste si l'on vient du formulaire ou s'il faut le créer
  if (isset($_REQUEST['send_note']) && (!empty($_REQUEST['note'])))
  { //Si on vient du formulaire
    //On traite les données
    echo "<p>Vous avez donné la note ".$_REQUEST['note']. "</p>";
  }
  else { //Sinon, génération de la page avec le formulaire
    echo '<form action="form_case2.php" method="get">
      <p>Choisissez une note :';

    for ($i=1; $i<10; $i++) echo '<input type="radio" name="note" value=".' . $i . '" />' . $i . ' ';
    echo '<input type="radio" name="note" value=".' . $i . '" checked="checked" />' . $i;

    echo '<br /><input type="submit" name="send_note" value="Envoyer" /></p></form>';
  }
?>

</body>
</html>
```

Exemple : *form_case2.php*

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
  <title>Page vide</title>
  <meta http-equiv="content-type"
    content="text/html; charset=utf-8" />
  <meta http-equiv="Content-Style-Type" content="text/css" />
</head>

<body>

<?php
  //On teste si l'on vient du formulaire ou s'il faut le créer
  if (isset($_REQUEST['send_note']) && (!empty($_REQUEST['note'])))
  { //Si on vient du formulaire
    //On traite les données
    echo "<p>Vous avez donné la note ".$_REQUEST['note']. "</p>";
  }
  else { //Sinon, génération de la page avec le formulaire
    echo '<form action="form_case2.php" method="post">';
    echo '<p>Choisissez une note :';

    for ($i=1; $i<10; $i++) echo '<input type="radio" name="note" value="'.$i.'" /> '.$i.' ';

    echo '<br /><input type="submit" name="send_note" value="Envoyer" /></p></form>';
  }
?>

</body>
</html>
```

Vous avez donné la note 3

Choisissez une note : ☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

Envoyer