



Chapitre 3 : Gérer les formulaires et les liens





Voici les chapitres que nous allons aborder:

- Envoi des données via un formulaire?
- Envoi des données via url
- Envoi d'un fichier image
- Sécurité des données transférées: Faille XSS





- Les éléments d'un formulaires

- Balise **<form>**. Attributs obligatoires **method** et **action**.

- Peut être composé de (**INPUT type=...**):

- Texte (**text**)

- Cases à cocher (**checkbox**)

N'afficher que les
pdf ☐ tex ☐ doc ☐

- Bouton d'envoi, reset (**submit, reset**)

- Password (**password**)

- Bouton radio (**radio**)

H ☒ F ☐

- Ou de liste de sélection (**SELECT name=**), zone de texte (**textarea**)

Tout
Seconde
Terminale
Université
Quatrième



- Attributs : **method** et **action**

- ✓ La procédure de soumission d'un formulaire est Contrôlée par 2 attributs de la balise <FORM> : Method et Action

- ↳ **METHOD** détermine la manière dont les données sont transmises au serveur. Il peut prendre les valeurs **GET** et **POST**.

- ↳ **ACTION** spécifie la page cible pour les données soumises.

```
<form method="post" action="destination.php">
```

OU

```
<form method="get" action="destination.php">
```



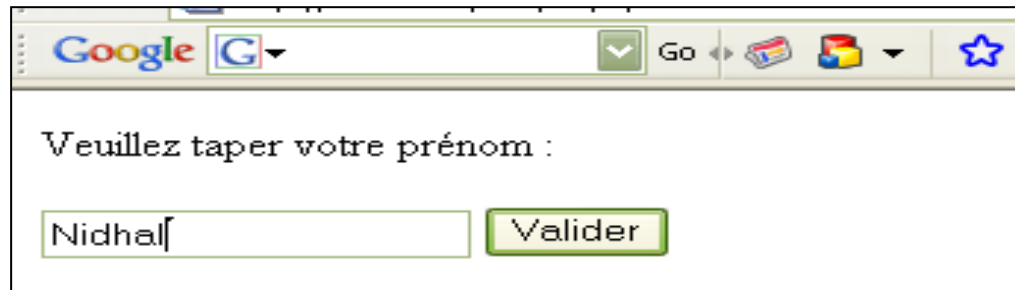
- ✓ Quand dans un site web un formulaire est rempli et envoyé, le contenu des champs saisis est transféré à la page destination sous forme de variables.
- ✓ Ce passage de variables ou de paramètres peut se faire de deux manières : en GET ou en POST

```
<html>
<body>
<!--Envoi d'un formulaire en POST -->
  <form method="post" action="destination.php">
<input type="text" name="nom" size="12"><br>
<input type="submit" value="OK">
</form>
<!--Envoi d'un formulaire en GET - >
<form method="get" action=" destination.php">
...
</form>    </body> </html>
```

■ Avec la méthode post

- POST demande à l'explorateur d'empaqueter toutes les données et de les transmettre au serveur.
- Les données ne sont pas visibles par l'internaute au moment de l'envoi
- Exemple soit deux fichiers

✓ [appel.html](#) contient la zone du texte à saisir

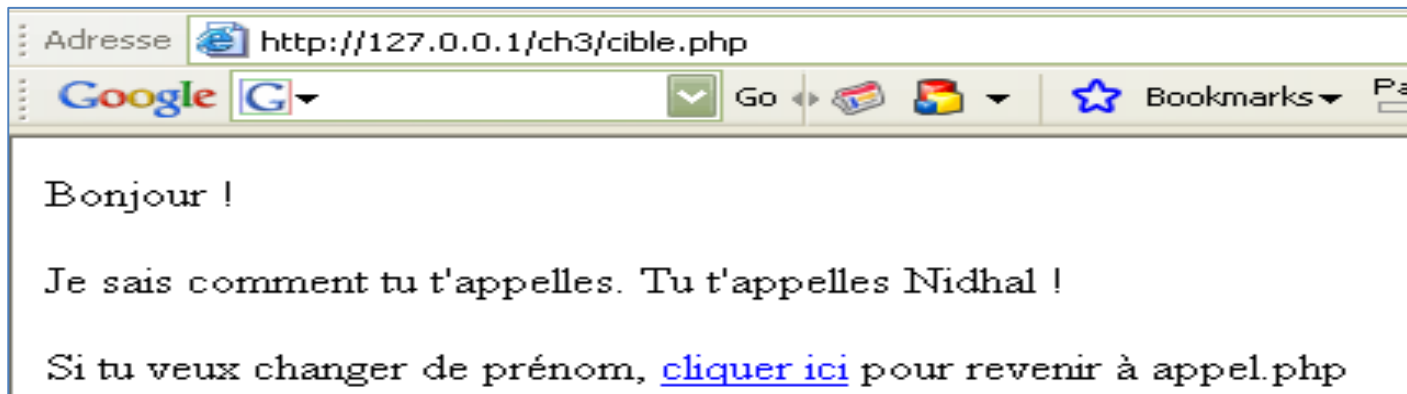


Google G Go

Veillez taper votre prénom :

Nidhal Valider

✓ [cible.php](#) affiche le contenu de la zone de texte



Adresse http://127.0.0.1/ch3/cible.php

Google G Go Bookmarks Pa

Bonjour !

Je sais comment tu t'appelles. Tu t'appelles Nidhal !

Si tu veux changer de prénom, [cliquer ici](#) pour revenir à appel.php



■ Avec la méthode **get**

- La méthode GET passe les variables et leurs données via l'URL d'une manière visible dans l'URL :

`http://127.0.0.1//cible.php?nom=Cherif&prenom=Nidhal`

- Exemple de script associé à une requête avec la méthode GET :

```
<?php
    echo "Nom: $_GET['nom'] <BR>";
    echo "Prenom: $_GET['prenom'] <BR>";
?>
```

1. Ecrire le code du formulaire dans le fichier « **commentaire.php** »

Vos commentaires

Aidez-nous à améliorer notre site:

Votre nom

Vos commentaires:

Votre adresse de courrier électronique:

Indiquez les éléments du site que vous aimez:

Design du site ☐ Les liens ☐ Facilité de navigation ☐ Les images ☐

2. En cliquant sur soumettre les données saisies seront envoyées et affichées dans le même fichier



- Envoi des données via un formulaire au script php du même fichier

```
<?php
    if (isset($_POST['submit']))
    { //Récupération des paramètres passés
        $prenom = $_POST['prenom'];
        $nom = $_POST['nom'];
        //affichage des paramètres
        echo "<center>Bonjour $prenom $nom</center>";
    }
    else
    {
?> <!-- Affichage du formulaire de saisie -->
<form method="post" action="formulaire.php">
    Nom : <input type="text" name="nom" size="12"><br>
    Prénom : <input type="text" name="prenom" size="12">
    <input type="submit" name="submit" value="OK">
</form>
```

formulaire.php



- Variable d'environnement : **`$_SERVER['PHP_SELF']`**
 - `$_SERVER['PHP_SELF']` retournera le nom du fichier en cours d'exécution « formulaire.php »
 - la destination du formulaire (action="formulaire.php") aurait pu ne pas être nominative mais exploiter une variable serveur `PHP_SELF` puisque les paramètres sont envoyées à la page elle même.

Cela donnera donc :

```
<form method="post" action="<?php  
    echo $_SERVER['PHP_SELF'];  
?>" >
```

à la place de :

```
<form method="post" action="formulaire.php">
```



- Des paramètres ou variables peuvent passer d'une page source vers une page destination sans transiter par un formulaire pour leur envoi. Les hyperliens peuvent être des vecteurs de passage de paramètre.

- Syntaxe

```
<!--Syntaxe d'envoi -->  
<a  
href=destination.php?variable1=contenu1&variable2=contenu2&  
...>    Lien </a>
```

- La récupération des paramètres dans la page destination se fait par le tableau \$_GET



```
<input type="file" name=" photo" />
```

Parcourir...

- les informations concernant les champs de type **file** sont enregistrées dans le tableau superglobal **\$_FILES** ;
- les fichiers sont temporairement placés dans le dossier temporaire du serveur.

• Entête du formulaire

```
<form method="post" action="page.php" enctype="multipart/form-data">
```

- **enctype** : le type d'encodage des données
- **multipart/form-data** : spécifie que le formulaire envoie des données binaires (fichier) et du texte (champs de formulaire)



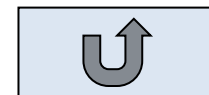
<?php

```
$Nom=$_FILES[ ' photo ' ][ ' name '];  
$Taille=$_FILES[ ' photo ' ][ ' size '];  
$Type=$_FILES[ ' photo ' ][ ' type '];  
$Nom_tmp=$_FILES[ ' photo ' ][ ' tmp_name '];  
if(is_uploaded_file($_FILES[ ' photo ' ][ ' tmp_name '])==true)  
    // on passe à la suite  
    // déplacement du fichier temporaire  
    move_uploaded_file($_FILES[ ' photo ' ][ ' tmp_name '],  
                      $_FILES[ ' photo ' ][ ' name ']);  
else die ("problème d'envoi du fichier");
```

- Appel.html

```
<html>
<body>
<!--Envoi d'un formulaire en POST -->
  <p>Veuillez taper votre prénom : </p>
<form action="cible.php" method="post">
<input type="text" name="prenom" >
<input type="submit" value="Valider" >
</form>

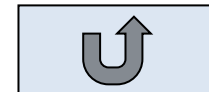
</body>
</html>
```



- cible.php

```
<html>
<body>
<p>Bonjour !</p>
<p>Je sais comment tu t'appelles. Tu t'appelles
  <?php
    echo    $_POST['prenom'];
  ?>  </p>
<p>Si tu veux changer de prénom,
<a href="appel.html">cliquer ici</a> pour revenir à
appel.html</p>

</body>
</html>
```





- XSS (plus officiellement appelée Cross-Site Scripting) est une faille permettant l'injection de code HTML ou JavaScript dans des variables mal protégées.

Imaginons la page suivante :

```
<html>
  <form method="post" action="connexion.php">
    <input type="texte" name="pseudo" />
    <input type="submit" value="Connexion" />
  </form>

</html>
```

Qui envoie ses paramètres à la page connexion.php :

```
<?php
  echo "Bonjour ".$_POST['pseudo']." !"

?>
```

L'utilisateur peut entrer dans le formulaire **Cherif** ou script javascript au lieu du nom



```

```

- **Solution** : la fonction `htmlspecialchars()` . Cette fonction permet de filtrer les symboles du type <, & ou encore " , en les remplaçant par leur équivalent en HTML. Par exemple :
 - Le symbole & devient &
 - Le symbole " devient "
 - Le symbole ' devient '