

Chapitre 6

Gérer les formulaires et les liens

1. Vue d'ensemble

1.1 Introduction

Dans les sites web dynamiques, il est très souvent nécessaire d'interagir avec l'utilisateur.

En HTML, il existe principalement deux méthodes pour interagir avec un utilisateur :

- les liens (balise `<a>`) ;
- les formulaires (balise `<form>`).

Des scripts PHP peuvent être utilisés pour traiter le clic de l'utilisateur sur un lien ou la saisie de l'utilisateur dans un formulaire.

1.2 Les liens

Le lien est la technique de base qui permet à un utilisateur de naviguer entre les différentes pages d'un site.

Un lien HTML est défini entre les balises `<a>` et ``.

Syntaxe simplifiée

```
<a  
  [ href="url" ]  
  [ id="identifiant_lien" ]  
  [ target="cible" ]
```

```
>
...
</a>
```

Les attributs de la balise `<a>` sont les suivantes :

- `href` URL (*Uniform Resource Locator*) relative ou absolue qui est appelée par le lien.
- `id` Identifiant du lien. Si la page HTML contient plusieurs liens, l'identifiant permet de les différencier. En ce qui nous concerne, cet identifiant ne présente pas d'intérêt car il n'est pas récupéré dans le script de traitement du lien. Par contre, il peut être utilisé côté client, en JavaScript par exemple.
- `target` Cible (par exemple une autre fenêtre) dans laquelle ouvrir l'URL cible. Par défaut, l'URL cible s'affiche dans la même fenêtre.

L'URL peut contenir des paramètres qui permettent de passer des informations d'une page à une autre.

Syntaxe

```
url_classique?nom=valeur[&...]
```

Le point d'interrogation (?) introduit la liste des paramètres de l'URL séparés par le caractère esperluette (&) ; chaque paramètre est constitué par un couple nom/valeur sous la forme `nom=valeur` :

```
www.monsite.com/info/accueil.php?prenom=Olivier
chercher.php?prenom=Olivier&nom=HEURTEL
```

Exemple

– Script `page1.php`

```
<?php
// Initialisation d'une variable.
$nom='Olivier';
?>
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr">
  <head><meta charset="utf-8" /><title>Page 1</title></head>
  <body>
    <div>
      <!-- lien vers la page 2 en passant la valeur de $nom
           dans l'URL -->
      <a href="page2.php?nom=<?=$nom ?>">Page 2</a>
    </div>
  </body>
</html>
```

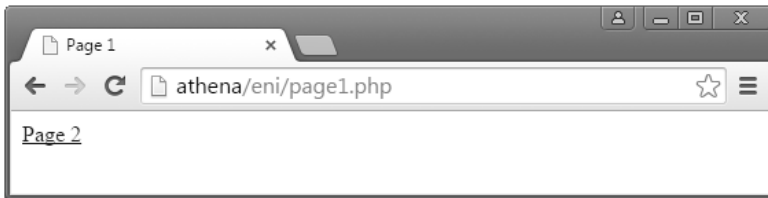
Chapitre 6

- Source de la page dans le navigateur

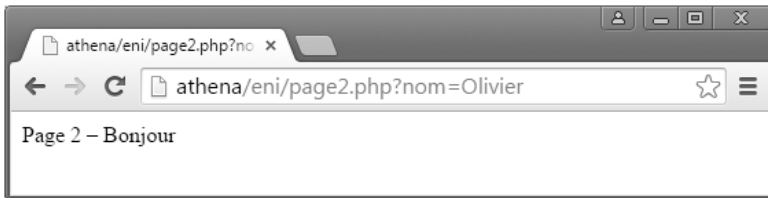
```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr">
  <head><meta charset="utf-8" /><title>Page 1</title></head>
  <body>
    <div>
      <!-- lien vers la page 2 en passant la valeur de $nom
           dans l'URL -->
      <a href="page2.php?nom=Olivier">Page 2</a>
    </div>
  </body>
</html>
```

Résultat

- Affichage de la page 1 :



- Résultat du clic sur le lien :



Pour l'instant, aucun nom n'est affiché dans la deuxième page. La variable `$nom` définie dans le script `page1.php` n'est pas disponible dans le script `page2.php` (voir le chapitre Introduction à PHP, section Les bases du langage PHP - Variables - Portée et durée de vie). De plus, notre script ne contient aucune instruction permettant de récupérer les données passées dans l'URL ; nous verrons comment procéder à la section Récupérer les données d'une URL ou d'un formulaire.

1.3 Les formulaires

1.3.1 Petit rappel sur les formulaires

Le 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.

Un formulaire HTML est défini entre les balises `<form>` et `</form>`.

Syntaxe simplifiée

```
<form
  [ action="url_de_traitement" ]
  [ method="GET"|"POST" ]
  [ id="identifiant_formulaire" ]
  [ target="cible" ]>
...
</form>
```

Les attributs de la balise `<form>` sont les suivants :

- | | |
|---------------------|--|
| <code>action</code> | URL (<i>Uniform Resource Locator</i>) relative ou absolue qui va traiter le formulaire (en ce qui nous concerne, un script PHP). Cet attribut est obligatoire pour se conformer à la recommandation XHTML stricte. |
| <code>method</code> | 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. |
| <code>id</code> | Identifiant du formulaire. Si la page HTML contient plusieurs formulaires, l'identifiant permet de les différencier. En ce qui nous concerne, cet identifiant ne présente pas d'intérêt car il n'est pas récupéré dans le script de traitement du formulaire. Par contre, il peut être utilisé côté client, en JavaScript par exemple. |
| <code>target</code> | Cible (par exemple une autre fenêtre) dans laquelle ouvrir l'URL cible. Par défaut, l'URL cible s'affiche dans la même fenêtre. |

Entre les balises `<form>` et `</form>`, il est possible de placer des balises `<input>`, `<select>` ou `<textarea>` pour définir des zones de saisie.

Exemple (formulaire HTML complet)

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr">
  <head>
    <meta charset="utf-8" />
    <title>Saisie</title>
  </head>
  <body>
    <form action="saisie.php" method="post">
      <div>
        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" />Féminin
        <input type="radio" name="sexe" value="?"
          checked="checked" />Ne sait pas
        <br />Photo :
        <input type="file" name="photo" size="50" />
        <br />Couleurs préférées :
        <input type="checkbox" name="couleurs[bleu]" />Bleu
        <input type="checkbox" name="couleurs[blanc]" />Blanc
        <input type="checkbox" name="couleurs[rouge]" />Rouge
        <input type="checkbox" name="couleurs[pas]"
          checked="checked" />Ne sait pas
        <br />Langue :
        <select name="langue">
          <option value="E">Espagnol</option>
          <option value="F" selected="selected" >Francais</option>
          <option value="I">Italien</option>
        </select>
        <br />Fruits préférés :<br />
        <select name="fruits[]" multiple="multiple" size="8">
          <option value="A">Abricots</option>
          <option value="C">Cerises</option>
          <option value="F">Fraises</option>
          <option value="P">Pêches</option>
          <option value="?" selected="selected">
            Ne sait pas</option>
        </select>
        <br />Commentaire :<br />
        <textarea name="commentaire" rows="4" cols="50"></textarea>
        <br />
        <input type="hidden" name="invisible" value="123" /><br />
      </div>
    </form>
  </body>
</html>
```

```
<input type="submit" name="soumettre" value="OK" />
<input type="image" name="valider" alt="valider"
src="valider.gif" />
<input type="reset" name="effacer" value="Effacer" />
<input type="button" name="action" value="Ne fait rien" />
</div>
</form>
</body>
</html>
```

Résultat



Nom : Mot de passe :

Sexe : Masculin Féminin Ne sait pas

Photo :

Couleurs préférées : Bleu Blanc Rouge Ne sait pas

Langue :

Fruits préférés :

-
-
-
-
-

Commentaire :

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).