

Formulaires HTML et traitement des données avec PHP

Introduction

Les formulaires HTML permettent aux utilisateurs d'entrer et d'envoyer des données à un serveur pour traitement. En combinaison avec PHP, les formulaires deviennent des outils puissants pour construire des applications web dynamiques, comme des systèmes de connexion, des outils de recherche ou des formulaires de contact. Ce chapitre explore en profondeur la création de formulaires HTML et le traitement sécurisé de leurs données avec PHP.

1. Création de formulaires HTML

Un formulaire HTML est défini à l'aide de la balise `<form>`. Voici les attributs importants :

- **action** : Spécifie la page PHP qui traitera les données.
- **method** : Détermine la méthode d'envoi (GET ou POST).

Exemple de formulaire simple :

```
<form action="traitement.php" method="post">
  <label for="nom">Nom :</label>
  <input type="text" id="nom" name="nom" required><br><br>

  <label for="email">Email :</label>
  <input type="email" id="email" name="email" required><br><br>

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

2. Méthodes GET et POST

- **GET** : Les données sont envoyées dans l'URL. Utile pour des recherches ou liens partageables.
 - **POST** : Les données sont envoyées de manière invisible, plus sécurisé pour les formulaires sensibles.
-

3. Traitement des données avec PHP

Dans le fichier `traitement.php`, on récupère les données envoyées par le formulaire :

```
<?php
$nom = $_POST['nom'];
$email = $_POST['email'];

echo "Bonjour $nom, votre adresse email est : $email";
?>
```

Remarque : Si le formulaire utilise `method="get"`, il faut utiliser `$_GET` au lieu de `$_POST`.

4. Sécuriser les entrées utilisateurs

Il est crucial de valider et filtrer les données pour éviter les attaques (injections XSS, SQL, etc.).

Validation de base :

```
if (!empty($_POST['nom']) && !empty($_POST['email'])) {
    $nom = htmlspecialchars($_POST['nom']);
    $email = filter_var($_POST['email'], FILTER_VALIDATE_EMAIL);

    if ($email) {
        echo "Bienvenue, $nom ! Votre email est valide.";
    } else {
        echo "Adresse email invalide.";
    }
} else {
    echo "Tous les champs sont requis.";
}
```

5. Gestion des erreurs et messages utilisateur

Il est important d'afficher des messages clairs en cas d'erreurs :

```
if (empty($_POST['nom'])) {
    echo "Le champ nom est obligatoire.";
}
```

On peut aussi rediriger ou conserver les données avec `$_SESSION`.

6. Formulaires avancés

Champs de type :

- radio, checkbox, select
- file pour l'envoi de fichiers

Exemple avec plusieurs champs :

```
<form action="traitement.php" method="post">
    <label>Genre :</label>
    <input type="radio" name="genre" value="Homme"> Homme
    <input type="radio" name="genre" value="Femme"> Femme<br><br>

    <label>Langages préférés :</label><br>
    <input type="checkbox" name="langages[]" value="PHP"> PHP<br>
    <input type="checkbox" name="langages[]" value="JavaScript">
JavaScript<br><br>

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

Récupération en PHP :

```
$genre = $_POST['genre'];
$langages = $_POST['langages']; // tableau

foreach ($langages as $langage) {
    echo "Langage choisi : $langage <br>";
}
```

7. Téléversement de fichiers

```
<form action="upload.php" method="post" enctype="multipart/form-data">
    Sélectionner un fichier :
    <input type="file" name="monfichier">
    <input type="submit" value="Uploader">
</form>
<?php
if (isset($_FILES['monfichier'])) {
    $dossier = 'uploads/';
    $fichier = $dossier . basename($_FILES['monfichier']['name']);

    if (move_uploaded_file($_FILES['monfichier']['tmp_name'], $fichier)) {
        echo "Fichier uploadé avec succès.";
    } else {
        echo "Erreur lors de l'envoi du fichier.";
    }
}
?>
```

Conclusion

Les formulaires HTML associés à PHP permettent de construire des interfaces interactives avec traitement côté serveur. Il est indispensable de toujours valider et sécuriser les données pour garantir la fiabilité et la sécurité des applications web. Grâce à cette combinaison, vous pouvez gérer des connexions, des inscriptions, des contacts, ou même uploader des fichiers de manière sécurisée et performante.