

Les formulaires

Table des matières

1.	Les requêtes http.....	2
2.	Les inputs TEXTE.....	2
3.	Les inputs SELECT	3
4.	Traiter les données avec \$_POST	4

1. Les requêtes http

En HTML, un formulaire permet d'interagir avec le serveur en définissant une **méthode**. Les plus courantes sont **POST** et **GET**. Pour cela, on va définir un attribut **method** dans notre balise **<form>**.

L'attribut **action**, lui, définit la **destination** du formulaire. Dans notre exemple, le formulaire envoie ses données vers la page contact.php.

```
<form method="POST" action="contact.php">
```

Dans le formulaire, on peut avoir plusieurs types d'**input**, que l'on va ranger en deux catégories : Les inputs **texte**, et les inputs de **sélection**. Chaque input a un attribut **name** qui permet de définir le nom de la clé que l'on va retrouver dans le tableau **\$_POST**.

2. Les inputs TEXTE

Pour définir le comportement de notre **input**, on va utiliser l'attribut **type**. On peut lui donner plusieurs valeurs :

- text
- password
- number
- email

et enfin la balise **<textarea>**

Voici un exemple :

```
<input type="text" name="prenom" placeholder="Prénom"/>
```

Les inputs de type texte comme **text**, **email**, **password**, peuvent aussi prendre des attributs **minlength** et **maxlength** pour contrôler la longueur de la chaîne de caractères admise. Exemple :

```
<input type="text" name="nom" placeholder="Nom" maxlength="10"/>
```

Pour les inputs de type **number**, on va plutôt utiliser les attributs **min** et **max**, comme ici :

```
<input type="number" name="age" min="0" max="20" placeholder="Âge"/>
```

Enfin, pour les longs textes, on peut utiliser la balise **<textarea>** pour afficher un input large avec une barre de scroll.

```
<textarea name="mon_texte" rows="6" cols="50"></textarea>
```

3. Les inputs SELECT

Les input de type **select** permettent de forcer l'utilisateur à faire un choix parmi une **sélection donnée**. On en dénombre trois :

- Les <select>
- Les input de type radio
- Les input de type checkbox

Les <select> permettent de choisir une option parmi une liste. On lui donne des balises <option> avec un attribut **value** que l'on retrouvera dans \$_POST. On l'écrit ainsi :

```
<select name="departement">
  <option value="">Choisissez votre département</option>
  <option value="42">Loire</option>
  <option value="03">Allier</option>
  <option value="63">Puy-de-Dôme</option>
  <option value="43">Haute-Loire</option>
  <option value="23">Creuse</option>
  <option value="38">Isère</option>
</select>
```

Les inputs de type radio permettent d'imposer un et **un seul** choix parmi un nombre défini. On donne à chaque input de type radio le même **name** pour que le navigateur soit capable de forcer le choix d'un seul élément. Voici la syntaxe :

```
Sexe : <br>
<input type="radio" name="sexe" value="H">
<label for="H">Homme</label><br>
<input type="radio" name="sexe" value="F">
<label for="F">Femme</label><br>
```

Et enfin, on a les inputs de type **checkbox** qui permettent de choisir **plusieurs** éléments parmi une liste. Pour que \$_POST récupère bien les multiples choix, on lui donne un nom avec des accolades [], pour qu'il comprenne que l'on a affaire à un tableau. Voici la syntaxe :

```
Comment êtes-vous habillés ? : <br>
<input type="checkbox" name="habits[]" value="Chemise">
<label for="Chemise">Chemise</label><br>
<input type="checkbox" name="habits[]" value="Pantalon">
<label for="Pantalon">Pantalon</label><br>
<input type="checkbox" name="habits[]" value="Chaussettes">
<label for="Chaussettes">Chaussettes</label><br>
<input type="checkbox" name="habits[]" value="Chaussures">
<label for="Chaussures">Chaussures</label><br>
```

4. Traiter les données avec \$_POST

Une fois que le formulaire est envoyé, si son attribut **method** est défini sur « **POST** », alors on peut récupérer les données du formulaire dans la variable PHP **\$_POST**.

Cette variable, tout le temps accessible, se présente sous forme de tableau associatif faisant le lien entre la **clé** et la **valeur**. La clé prendra la valeur de l'attribut **name**, et la valeur prendra ce qui a été saisi dans l'input.

Si l'on a saisi plusieurs choix dans notre liste de checkbox par exemple, alors la variable **\$_POST** nous donnera un **tableau** contenant toutes les cases cochées.

Voici le contenu de **\$_POST** après avoir envoyé le formulaire réalisé ici :

The diagram illustrates the structure of the `$_POST` variable. It shows an array of size 9 with the following elements:

- `'nom' => string 'SAVOCA' (Length=6)`: The key `'nom'` is annotated as "Nom de notre champ dans le HTML : c'est la clé". The value `'SAVOCA'` is annotated as "VALEUR de la variable `$_POST['nom']`".
- `'prenom' => string 'Nicolas' (Length=7)`
- `'age' => string '20' (Length=2)`
- `'password' => string 'testkjfhjzfhjo' (Length=15)`
- `'email' => string 'n.savoca@osengo.fr' (Length=18)`
- `'mon_texte' => string 'Test je vous dis bonjour' (Length=24)`
- `'departement' => string '03' (Length=2)`
- `'sexe' => string 'H' (Length=1)`
- `'habits' => array (size=4)`: This is a nested array. Its elements are:
 - `0 => string 'Chaussettes' (Length=11)`: The value `'Chaussettes'` is annotated as `$_POST['habits'][0]`.
 - `1 => string 'Chaussures' (Length=10)`
 - `2 => string 'T-shirt' (Length=7)`
 - `3 => string 'Short' (Length=5)`

Additional annotations include:

- "Type de la variable `$_POST['nom']`" pointing to the `string` type.
- "TYPE et la TAILLE de la VARIABLE `$_POST`" pointing to the `array (size=9)` declaration.
- "Pour accéder à cette valeur, on peut appeler `$_POST['habits']`, et lui demander son index 0, ce qui nous donne :