

Formulaires HTML



1. Qu'est-ce qu'un formulaire ?

En HTML, un formulaire est un espace de saisie dans une page web pouvant comporter plusieurs zones, ou champs. On peut y saisir du texte, cocher des cases, effectuer un choix dans une liste de termes prédéfinis, appuyer sur des boutons, etc. Ces zones de saisie assurent une interaction entre la personne utilisant le site web et la machine (serveur) qui héberge ce site. Les boutons des formulaires permettent d'envoyer les données entrées par l'utilisateur puis d'exécuter un script (créé en JavaScript, PHP, Perl...) côté serveur permettant de les traiter puis de renvoyer une page web. Le but de ce document est d'aider à créer un formulaire en HTML que nous ferons traiter ensuite par un script PHP disponible sur Internet.

2. Création d'un formulaire

Un formulaire s'insère simplement dans le code HTML en utilisant la balise `<form>`. Cependant, cette balise nécessite obligatoirement l'utilisation d'attributs bien spécifiques. En effet, il est indispensable d'indiquer de quelle manière les données saisies par l'utilisateur seront envoyées aux serveur et quel traitement elles vont subir. Le premier attribut est `method`. C'est lui qui indique la méthode d'envoi des données saisies. Il peut prendre deux valeurs :

- `method="get"` : valeur en général peu adaptée car l'envoi est limité à 255 caractères et les informations seront envoyées dans l'adresse de la page. Elle est à déconseiller.
- `method="post"` : valeur qui permet d'envoyer un grand nombre d'informations sans que les données saisies transitent par la barre d'adresse. Elle est davantage recommandée.

Le second attribut est `action`. Sa valeur est l'adresse du script qui traitera les informations envoyées. En général, ce programme se situe sur le serveur hébergeant le site web. Ici, nous utiliserons un fichier `php`.

Dans le code HTML, l'introduction d'un formulaire se fait donc avec le code suivant :

```
<form action="adresse_du_fichier.php" method="post">
  <!-- intérieur du formulaire -->
</form>
```

3. Contenu d'un formulaire

Un formulaire peut contenir plusieurs types d'éléments selon les informations à demander et la nature des réponses à apporter.

Les exemples donnés plus bas sont disponibles dans un fichier `html` mis à disposition.

1. Les zones de saisie de texte.

Il en existe deux types :

(a) Zone de texte à une ligne

pour saisir des textes relativement courts comme, par exemple, un nom, un pseudo, un mot de passe... Elle se crée en utilisant la balise `<input />`. Cette balise est dite orpheline, c'est-à-dire qu'elle ne se ferme pas avec `</input>`. Les attributs obligatoires à utiliser pour créer une zone de texte à une ligne :

- `type` : indique le type d'entrée (texte ou mot de passe).
- `name` : donne un nom à la zone de texte.

Un attribut `id` optionnel permet, lorsque l'utilisateur clique sur le label de la zone de texte, que le curseur se place automatiquement sur la zone à remplir.

Pour que l'utilisateur sache quel contenu écrire, on utilise un `<label>` qui peut se placer avant ou après la balise `<input />`, selon que l'on veuille l'afficher avant ou après. Cette balise `<label>` doit contenir un attribut `for` prenant la même valeur que `name` et `id` pour faire le lien avec le label.

Il est possible d'indiquer un contenu par un texte grisé disparaissant à la saisie, de proposer un texte par défaut, etc.

Exemple Le code :

```
<label for="nom">Votre nom : </label>
<input type="text" name="nom" id="nom"
  placeholder="Tapez votre nom ici" /><br/>
<label for="spe">La spécialité NSI c'est génial : </label>
<input type="text" name="spe" id="spe" size="3"
  maxlength="3" value="oui" /><br/>
<label for="mdp">Un petit texte secret : </label>
<input type="password" name="mdp" id="mdp" size="20"
  value="123456" /><br/>
```

Donne :

Votre nom :

La spécialité NSI c'est génial :

Un petit texte secret :

(b) Zone de texte multi-lignes

Si la saisie à faire nécessite la présence de plusieurs lignes de texte, comme pour un commentaire ou une description par exemple, il faut utiliser la balise `<textarea>` qui elle n'est pas orpheline. Un contenu par défaut est donc donné entre les deux balises.

Exemple Le code suivant :

```
<label for="vacances">Racontez vos vacances : </label><br/>
<textarea name="vacances" id="vacances" rows="10" cols="60">
  J'ai fait un voyage en Angleterre...
</textarea>
```

Donne :

Racontez vos vacances :

J'ai fait un voyage en Angleterre...



2. Choix multiples

(a) Les cases à cocher

Le type checkbox des balises `<input />` est fait pour cela.

Exemple Le code :

```
<label style="font-weight: bold; text-decoration: underline;">
Cochez vos franchises cinématographiques préférées :</label><br/>
<br/>
<input type="checkbox" name="bond" id="bond" checked />
<label for="bond">James Bond.</label><br/>
<input type="checkbox" name="starwars" id="starwars" />
<label for="starwars">Star Wars.</label><br/>
<input type="checkbox" name="avengers" id="avengers" />
<label for="avengers">Avengers.</label><br/>
<input type="checkbox" name="mission" id="mission" checked />
<label for="mission">Mission impossible.</label><br/>
```

donne :

Cochez vos franchises cinématographiques préférées :

- James Bond.
- Star Wars.
- Avengers.
- Mission impossible.

(b) Les zones d'options

Une zone d'options permet de faire un **choix unique** parmi une liste de possibilités. Elle se définit avec la balise `<input />` dont l'attribut `type` aura pour valeur `radio`. Une zone d'options se différencie des cases à cocher par le fait qu'elle doit être organisée en groupes. Les options d'un même groupe doivent avoir le même `name` (de manière à ce qu'un seul choix soit possible dans le groupe) mais une `value` différente.

Exemple Le code :

```
Quel est votre professeur de NSI ?<br/>
<input type="radio" name="Prof" value="Malingrey" id="Malingrey" />
<label for="Malingrey">M Malingrey</label>.<br/>
<input type="radio" name="Prof" value="Thévenon" id="Thévenon" checked />
<label for="Thévenon">M Thévenon</label>.
```

Donne :

Quel est votre professeur de NSI ?

- M Malingrey.
- M Thévenon.

(c) Les listes déroulantes

Pour faire un choix unique parmi une liste pré-établie, on vient de voir qu'on peut définir une zone d'options mais il existe une seconde possibilité : une liste déroulante. Pour en définir une, on doit utiliser la balise `<select>` en précisant ses attributs `name` et `id`. Ensuite on y insère autant de balises `option` que l'on veut de choix possibles.

Exemple Le code :

```
<label for="classe">Quel est votre classe ?</label><br/>
<Select name="classe" id="classe">
<option value="2">Seconde</option>
<option value="1" selected>Première</option>
<option value="0">Terminale</option>
</select>
```

Donne :

Quel est votre classe ?

Première ▾

On peut ajouter l'attribut `selected` à l'un des éléments `option` pour donner un choix par défaut (dont la valeur peut être vide et dont le champ peut être vide).

4. Éléments de présentation

Quelques éléments de présentation peuvent être ajoutés. Pour qu'un formulaire soit lisible sans difficulté par l'utilisateur s'il contient beaucoup de choses à saisir, il est conseillé de regrouper les champs en rubriques en utilisant les balises `<fieldset>` et `<legend>`. La balise `fieldset` permet d'entourer les champs sélectionnés d'un mince liseré et la balise `legend` permet de lui donner un nom. En voici la syntaxe pour un groupement de champs :

```
<fieldset>
<legend>Titre du groupe de champs</legend>
<!-- champs de formulaire -->
</fieldset>
```

Dans le fichier html d'exemples, ceux-ci sont séparés les uns des autres par ce moyen.

On peut également rendre la saisie de certains champs obligatoire. Pour cela on utilise l'attribut `required` qui permet d'assurer que la valeur est non vide lors de l'envoi.

Pour placer le curseur automatiquement sur un champ précis à l'ouverture de la page, on utilise l'attribut `autofocus`.

Dans le fichier html d'exemples, l'autofocus est mis sur la zone de texte multi-lignes, et seul le champ nom est rendu obligatoire.

5. Bouton d'envoi du formulaire ; remise à zéro

Le formulaire doit toujours intégrer un bouton d'envoi, ce qui permet à l'utilisateur de valider toutes ses saisies et à ces dernières d'être envoyées afin que le script déclaré dans la définition du formulaire puisse les traiter. Pour créer un tel bouton il suffit d'utiliser à nouveau la balise `<input />` mais en attribuant la valeur `submit` à l'attribut `type`.

On peut aussi donner la valeur `reset` à `type` si on souhaite permettre la remise à zéro du formulaire (avec ses valeurs par défaut). Voir le fichier html d'exemples pour voir la mise en place de ces éléments.

6. Exercice

À partir du fichier contenant les exemples, écrire le code HTML pour obtenir un formulaire ressemblant à la capture d'écran suivante :

État civil	
Nom : <input type="text"/>	Prénom : <input type="text" value="Tapez votre prénom ici"/>
Âge : <input type="text"/>	
Scolarité	
Classe : <input type="text" value="v"/>	
Cochez la case si vous êtes redoublant : <input type="checkbox"/>	
Orientation	
Qu'envisagez-vous pour la suite de vos études ?	
<input type="checkbox"/> BTS	
<input type="checkbox"/> BUT <input type="checkbox"/> Université	
<input type="checkbox"/> CPGE	
<input type="checkbox"/> Autre	
Si Autre, précisez :	
<input type="text"/>	
Valider vos choix	Remise à zéro

Rendre obligatoire le remplissage de chacun des champs nommés nom, prenom, age et classe.

L'autofocus sera mis sur le champ nom.

Faire en sorte que la sélection de la classe soit vide au lancement de la page.

Tester alors le formulaire en donnant comme valeur à l'attribut action de la balise `<form>` l'adresse suivante :

<http://patrick.thevenon.free.fr/tmp/traitement.php>

On veillera à essayer les deux méthodes, `get` et `post` et à observer la différence entre les deux visible, comme évoqué au début de ce document, dans l'adresse de la page web obtenue.

Pour information, le contenu du fichier traitement.php est le suivant :

```
<!DOCTYPE html>
<html>
<head>
<title>Analyse du formulaire</title>
<meta http-equiv="Content-type" content="text/html; charset=utf-8"/>
</head>
<body>
<p>Après traitement du formulaire, voici ce que l'on peut dire :</p>
<div style="border-style:double">
<?php
if (!empty($_POST))
{
echo "Vous vous appelez ".$_POST["prenom"]." ".$_POST["nom"];
echo ", vous avez ".$_POST["age"]." ans.";
echo " Vous êtes en classe de ".$_POST["classe"].".";
}
else
{
echo "Vous vous appelez ".$_GET["prenom"]." ".$_GET["nom"];
echo ", vous avez ".$_GET["age"]." ans.";
echo " Vous êtes en classe de ".$_GET["classe"].".";
}
?>
</div>
</body>
</html>
```