

# 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 du 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, pour l'instant vide -->
</form>
```

Voyons les différentes façons de coder certaines des informations que peut être amené à saisir un utilisateur de site web.

## 3. Les zones de saisie de texte

### a. Zone de texte à une ligne

Comme son nom l'indique, elle sert à 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'il n'est pas nécessaire de la fermer avec `</input>`. Les attributs obligatoires à utiliser pour créer une zone de texte à une ligne :

- **type="text"** : indique que du texte est demandé à l'utilisateur.

- **name="nom"** : donne un nom à la zone de texte.

Ce code minimal n'affiche qu'une case vide dans laquelle l'utilisateur ne sait pas quoi taper. Il faut donc lui attribuer un **label** et le lier à la zone de texte. Cela se fait avec la balise `<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** pour faire le lien avec le label.

Si dans la balise `input` on ajoute un attribut **id**, lorsque l'utilisateur clique sur le label de la zone de texte, le curseur se placera automatiquement sur la zone à remplir.

Exemple :

```
<label for="nom">Votre nom : </label>
<input type="text" name="nom" id="nom" />
```

Attributs facultatifs de la balise `input` :

**size="40"** : permet de fixer la taille du champ de saisie de texte. Le nombre entre guillemets fixe cette taille.

**Maxlength="20"** : limite le nombre de caractères à saisir avec la valeur entre guillemets.

**value="Oui"** : pré-remplit la zone de texte avec une valeur par défaut.

**placeholder="Tapez votre nom ici"** : donne une indication sur ce qu'il faut taper dans la zone de saisie de texte, indication qui disparaît dès que l'utilisateur commence à taper.

**type="password"** : crée une zone de mot de passe, c'est-à-dire que ce qui est tapé dans la zone de saisie ne sera pas affiché à l'écran.

## Exercice 1

Taper le code HTML nécessaire à l'affichage reproduit ci-dessous.

Votre nom :

La spécialité NSI c'est génial :

Un petit texte secret :

## b. Zone de texte multi-ligne

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`. De la même manière que précédemment, il faut lui donner un nom et la faire précéder de la balise `label`.

Exemple :

```
<label for="description">Racontez vos vacances : </label><br/>
<textarea name="description" id="description"></textarea>
```

Attributs optionnels :

**rows="10"** : permet de fixer le nombre de lignes de la zone de texte.

**cols="40"** : permet de fixer le nombre de colonnes de la zone de texte.

## Exercice 2

Taper le code HTML nécessaire à l'affichage reproduit ci-dessous.

**Racontez vos vacances :**

J'ai fait un voyage en Angleterre...

## 4. Choix multiples

### a. Les cases à cocher

Souvent, plutôt que de demander au visiteur d'énumérer ses choix, il est plus facile de lui faire cocher des cases parmi une liste pré-établie. Le type **checkbox** est fait pour ça.

Syntaxe :

```
<input type="checkbox" name="Bond" id="Bond" />
<label for="Bond">James Bond.</label>
```

### Exercice 3

Taper le code HTML nécessaire à l'affichage reproduit ci-dessous.

#### 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 :

```
<form action="traitement.php" method="post">
Dans quelle classe êtes-vous ?<br/>
<input type="radio" name="Classe" value="Seconde" id="Seconde" />
<label for="Seconde">Seconde</label>.<br/>
<input type="radio" name="Classe" value="Première" id="Première" />
<label for="Première">Première</label>.<br/>
<input type="radio" name="Classe" value="Terminale" id="Terminale" />
<label for="Terminale">Terminale</label>.<br/>
<br/>
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="Seckinger" id="Seckinger" />
<label for="Seckinger">M Seckinger</label>.<br/>
<input type="radio" name="Prof" value="Thévenon" id="Thévenon" />
<label for="Thévenon">M Thévenon</label>.<br/>
</form>
```

### Exercice 4

Tester ce code pour bien comprendre la différence avec les cases à cocher.

### 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 :

```
<label for="Prof">Quel est votre professeur de NSI ?</label><br/>
<Select name="Prof" id="Prof">
<option value="Malingrey">M Malingrey</option>
<option value="Seckinger">M Seckinger</option>
<option value="Thévenon">M Thévenon</option>
</select>
```

#### Exercice 5

Tester le code.

## 5. Derniers réglages et envoi du formulaire

### a. Présentation et astuces

Il faut toujours s'assurer qu'un formulaire soit lisible sans difficulté par l'utilisateur sous peine de décourager ce dernier. Si le formulaire 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.

Syntaxe :

```
<fieldset>
<legend>Etat civil</legend>
<!-- insérer les champs ici -->
</fieldset>
<fieldset>
<legend>Scolarité</legend>
<!-- insérer les champs ici -->
</fieldset>
```

Et on peut obtenir un affichage similaire à ceci :

<b>Etat civil</b> Nom : <input type="text"/> Prénom : <input type="text"/>
<b>Scolarité</b> Dans quelle classe êtes-vous ? <input type="radio"/> Seconde. <input type="radio"/> Première. <input type="radio"/> Terminale.

Dans un formulaire, certains champs sont plus important que d'autres. Il est donc nécessaire de les mettre en valeur, voire de rendre leur saisie obligatoire. Pour cela il existe les attributs **autofocus** et **required**.

Exemple :

```
<input type="text" name="nom" id="nom" autofocus required />
```

## b. Création du bouton d'envoi du formulaire

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 remettre le formulaire à zéro.

Exemple :

```
<input type="submit" value="Valider vos choix" />  
<input type="reset" value="Remise à zéro" />
```

### Exercice 6

Écrire le code HTML pour obtenir un formulaire ressemblant à la capture d'écran suivante :

The screenshot shows a web form with three main sections:

- Etat civil**: Contains three input fields for 'Nom', 'Prénom', and 'Âge'.
- Scolarité**: Contains a dropdown menu for 'Classe' (set to 'Seconde'), a group of radio buttons for 'Série' (S, ES, L, Indéterminée), and a checkbox for 'Redoublant(e)'.
- Orientation**: Contains a heading 'Qu'envisagez-vous pour la suite de vos études ?' followed by five checkboxes: 'Université', 'BTS', 'IUT', 'Ecole de commerce', and 'Ecole d'ingénieurs'. Below this is a text area for 'Commentaires éventuels'.

At the bottom of the form are two buttons: 'Valider vos choix' and 'Remise à zéro'.

Rendre obligatoire le remplissage de chacun des champs nommés nom, prenom, age et classe. Faire en sorte que la sélection de la classe soit vide au lancement de la page, et que la série soit indéterminée par défaut. Noter qu'il y a un élément input d'un type différent de ceux vus plus haut pour l'âge : il s'agit du type **number**. Tester alors le formulaire en utilisant le fichier php ayant pour adresse : <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 (évoquée en début de document).

Pour information, le contenu de ce 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>
```