

6 : LES FORMULAIRES

I. Créer un formulaire

Un formulaire est une zone interactive de la page, dans laquelle vos visiteurs peuvent saisir des informations.

On délimite un formulaire avec la balise `<form>`, à laquelle il faut ajouter deux attributs : **method** (mode d'envoi des données) et **action** (page vers laquelle le visiteur sera redirigé après envoi du formulaire, et qui traitera les informations).

Exemple :

```
index.html
```

```
<p>Texte avant le formulaire</p>
<form>
  <p>Texte à l'intérieur du formulaire</p>
</form>
<p>Texte après le formulaire</p>
```

Vous devez ajouter deux attributs à la balise `<form>` :

method : cet attribut indique par quel moyen les données vont être. Il existe deux solutions pour envoyer des données sur le Web :

- `method="get"` : c'est une méthode en général assez peu adaptée car elle est limitée à 255 caractères. La particularité vient du fait que les informations seront envoyées dans l'adresse de la page (`http://...`), il est plutôt recommandé d'utiliser l'autre méthode : `post` ;
- `method="post"` : c'est la méthode la plus utilisée pour les formulaires car elle permet d'envoyer un grand nombre d'informations. Les données saisies dans le formulaire ne transitent pas par la barre d'adresse ;

action : c'est l'adresse de la page ou du programme qui va traiter les informations. Cette page se chargera de vous envoyer un e-mail avec le message si c'est ce que vous voulez, ou bien d'enregistrer le message avec tous les autres dans une base de données.

Cela ne peut pas se faire en HTML et CSS ; on utilisera en général le PHP.

```
index.html
```

```
<p>Texte avant le formulaire</p>
<form method=''post'' action=''traitement.php''>
  <p>Texte à l'intérieur du formulaire</p>
</form>
<p>Texte après le formulaire</p>
```

II. Les zones de saisie basiques

Zone de texte monoligne :

L'image suivante montre à quoi ressemble une zone de texte monoligne.



Pour insérer une zone de texte dans une ligne, on va utiliser la balise `<input />`.

On retrouvera cette balise plusieurs fois dans la suite de ce chapitre. À chaque fois, c'est la valeur de son attribut **type** qui va changer.

Pour créer une zone de texte à une ligne, on doit écrire :

```
index.html

<input type='text' />
```

Ce n'est pas encore suffisant : il faut donner un nom à votre zone de texte. Ce nom n'apparaît pas sur la page, mais il vous sera indispensable par la suite. En effet, cela vous permettra (en PHP par exemple) de reconnaître d'où viennent les informations : vous saurez que tel texte est le pseudo du visiteur, tel texte est le mot de passe qu'il a choisi, etc.

Pour donner un nom à un élément de formulaire, on utilise l'attribut **name**. Ici, on va supposer qu'on demande au visiteur de rentrer son pseudo :

```
index.html

<input type='text' name='pseudo' />
```

Essayons donc de créer un formulaire très basique avec notre champ de texte :

```
index.html

<form method='post' action='traitement.php'>
  <p><input type='text' name='pseudo' /></p>
</form>
```

Les libellés :

Cette zone de texte est bien jolie mais si votre visiteur tombe dessus, il ne sait pas ce qu'il doit écrire. C'est justement le rôle de la balise `<label>` :

```
index.html

<form method='post' action='traitement.php'>
  <p>
    <label>Votre pseudo</label> : <input type='text'
name='pseudo' />
  </p>
</form>
```

Ce code donne exactement le résultat que vous avez pu observer sur l'image précédente.

Mais cela ne suffit pas. Il faut lier le label à la zone de texte.

Pour ce faire, on doit donner un nom à la zone de texte, non pas avec l'attribut **name** mais avec l'attribut **id** (que l'on peut utiliser sur toutes les balises).

Un **name** et un **id** sur le champ ? Cela ne va-t-il pas faire double emploi ?

Pas du tout ! En fait, l'**id** est utilisé pour identifier l'élément HTML pour pouvoir y accéder et le manipuler. Il est donc unique pour cet élément.

Le **name**, lui, réfère à la variable du formulaire que l'élément concerne. Ici, il n'y a qu'un seul élément qui pourra référer à la variable pseudo ; **name** et **id** auront donc la même valeur. Mais lorsque nous utiliserons des **checkbox** ou des **radio**, plusieurs éléments correspondront à la même variable. Par exemple, la variable couleur avec un élément pour rouge, un pour bleu et un pour vert. Ils auront donc le même **name**, mais pas le même **id**.

Pour lier le label au champ, il faut lui donner un attribut **for** qui a la même valeur que l'**id** du champ... Le mieux est de le voir sur un exemple :

```
index.html

<form method='post' action='traitement.php'>
  <p>
    <label for='pseudo' >Votre pseudo</label> : <input
type='text' name='pseudo' id='pseudo' />
  </p>
</form>
```

Essayez de cliquer sur le texte « Votre pseudo » : vous allez voir que le curseur se place automatiquement dans la zone de texte correspondante.

Quelques attributs supplémentaires :

On peut ajouter un certain nombre d'autres attributs à la balise `<input />` pour personnaliser son fonctionnement :

- on peut agrandir le champ avec **size** ;
- on peut limiter le nombre de caractères que l'on peut saisir avec **maxlength** ;
- on peut préremplir le champ avec une valeur par défaut à l'aide de **value** ;
- on peut donner une indication sur le contenu du champ avec **placeholder**. Cette indication disparaîtra dès que le visiteur aura cliqué à l'intérieur du champ.

Dans l'exemple suivant, la zone de texte contient une indication permettant de comprendre ce qu'il faut saisir ; le champ fait 30 caractères de long mais on ne peut écrire que 10 caractères maximum à l'intérieur :

`index.html`

```
<form method='post' action='traitement.php'>
  <p>
    <label for='pseudo'>Votre pseudo :</label>
    <input type='text' name='pseudo' id='pseudo' placeholder='Ex : Zozor' size='30' maxlength='10' />
  </p>
</form>
```

Voici le rendu du champ dans son état initial en figure suivante.

Zone de mot de passe :

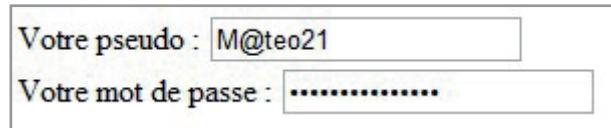
Vous pouvez facilement faire en sorte que la zone de texte se comporte comme une « zone de mot de passe », c'est-à-dire une zone où on ne voit pas à l'écran les caractères saisis. Pour créer ce type de zone de saisie, utilisez l'attribut **type="password"**.

Je complète mon formulaire. Il demande maintenant au visiteur son pseudo et son mot de passe :

`index.html`

```
<form method='post' action='traitement.php'>
  <p>
    <label for='pseudo'>Votre pseudo :</label>
    <input type='text' name='pseudo' id='pseudo' />
    <br />
    <label for='pass'>Votre mot de passe :</label>
    <input type='password' name='pass' id='pass' />
  </p>
</form>
```

Les caractères ne s'affichent pas à l'écran, comme sur l'image suivante.



Zone de texte multiligne :

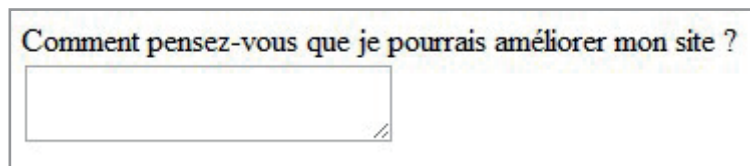
Pour créer une zone de texte multiligne, on change de balise : nous allons utiliser `<textarea>` `</textarea>`.

Comme pour tout autre élément du formulaire, il faut lui donner un nom avec **name** et utiliser un **label** qui explique de quoi il s'agit.

`index.html`

```
<form method=''post'' action=''traitement.php''>
  <p>
    <label for=''ameliorer''>Comment pensez-vous que je pourrais
    améliorer mon site ?</label><br />
    <textarea name=''ameliorer'' id=''ameliorer''></textarea>
  </p>
</form>
```

Et voici le résultat en image :



On peut modifier la taille du `<textarea>` de deux façons différentes.

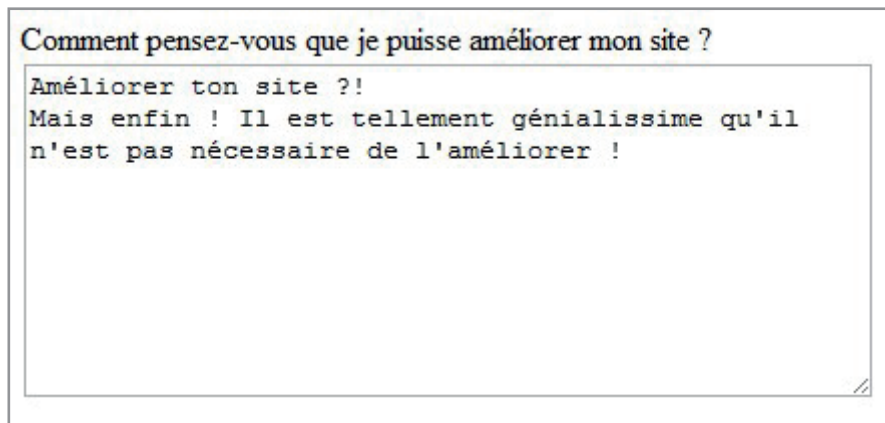
En CSS : il suffit d'appliquer les propriétés CSS **width** et **height** au `<textarea>`.

Avec des attributs : on peut ajouter les attributs **rows** et **cols** à la balise `<textarea>`. Le premier indique le nombre de lignes de texte qui peuvent être affichées simultanément, et le second le nombre de colonnes.

Vous pouvez préremplir le `<textarea>` avec une valeur par défaut. Dans ce cas, on n'utilise pas l'attribut **value** : on écrit tout simplement le texte par défaut entre la balise ouvrante et la balise fermante.

```
<form method=''post'' action=''traitement.php''>
  <p>
    <label for=''ameliorer''>
      Comment pensez-vous que je puisse améliorer mon site ?
    </label>
    <br />
    <textarea name=''ameliorer'' id=''ameliorer'' rows=''10''
    cols=''50''>
      Améliorer ton site ?!
      Mais enfin ! Il est tellement génialissime qu'il n'est pas né
      cessaire de l'améliorer !
    </textarea></p></form>
```

Et voici le résultat :



Les zones de saisie enrichies :

Email

Vous pouvez demander à saisir une adresse e-mail :

```
index.html
```

```
<input type='email' />
```

Le champ vous semblera a priori identique, mais votre navigateur sait désormais que l'utilisateur doit saisir une adresse e-mail. Il peut afficher une indication si l'adresse n'est pas un e-mail ; c'est ce que fait Firefox, par exemple (figure suivante).



URL

Avec le type **url**, on peut demander à saisir une adresse absolue (commençant généralement par `http://`) :

```
index.html
```

```
<input type='url' />
```

Même principe : si le champ ne vous semble pas différent sur votre ordinateur, sachez que celui-ci comprend bel et bien que le visiteur est censé saisir une adresse. Les navigateurs mobiles affichent par exemple un clavier adapté à la saisie d'URL.

Numéro de téléphone

Ce champ est dédié à la saisie de numéros de téléphone :

```
index.html
```

```
<input type='tel' />
```

Sur smartphone par exemple, un clavier adapté (avec les chiffres) s'affiche lorsqu'on doit remplir le champ.

Nombre

Ce champ permet de saisir un nombre entier :

```
index.html
```

```
<input type='number' />
```

Le champ s'affichera en général avec des petites flèches pour changer la valeur (figure suivante).

Vous pouvez personnaliser le fonctionnement du champ avec les attributs suivants :

- **min** : valeur minimale autorisée ;
- **max** : valeur maximale autorisée ;
- **step** : c'est le « pas » de déplacement. Si vous indiquez un pas de 2, le champ n'acceptera que des valeurs de 2 en 2 (par exemple 0, 2, 4, 6...).

III. Les éléments d'options

HTML vous offre une ribambelle d'éléments d'options à utiliser dans votre formulaire. Ce sont des éléments qui demandent au visiteur de faire un choix parmi une liste de possibilités. Nous allons passer en revue :

- les cases à cocher ;
- les zones d'options ;
- les listes déroulantes.

Les cases à cocher :

Créer une case à cocher ? Rien de plus simple ! Nous allons réutiliser la balise `<input />`, en spécifiant cette fois le type **checkbox** :

```
index.html
```

```
<input type='checkbox' name='choix' />
```

Rajoutez un `<label>` bien placé, et le tour est joué !

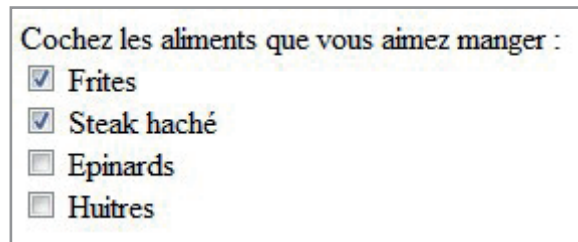
```
index.html
```

```
<form method='post' action='traitement.php'>
  <p>
    Cochez les aliments que vous aimez manger :<br />
    <input type='checkbox' name='frites' id='frites' />
  <label for='frites'>Frites</label><br />
</form>
```

index.html suite

```
<input type=''checkbox'' name=''steak'' id=''steak'' />
<label for=''steak''>Steak haché</label><br />
    <input type=''checkbox'' name=''epinards'' id=''epinards'' />
<label for=''epinards''>Epinards</label><br />
    <input type=''checkbox'' name=''huitres'' id=''huitres'' />
<label for=''huitres''>Huitres</label>
</p>
</form>
```

Et voici le résultat sur l'image suivante :



N'oubliez pas de donner un nom différent à chaque case à cocher, cela vous permettra d'identifier plus tard lesquelles ont été cochées par le visiteur.

Enfin, sachez que vous pouvez faire en sorte qu'une case soit cochée par défaut, avec l'attribut **checked** :

index.html

```
<input type=''checkbox'' name=''choix'' checked />
```

Les zones d'options :

Les zones d'options vous permettent de faire un choix (et un seul) parmi une liste de possibilités. Elles ressemblent un peu aux cases à cocher, mais il y a une petite difficulté supplémentaire : elles doivent être organisées en groupes. Les options d'un même groupe possèdent le même nom (**name**), mais chaque option doit avoir une valeur (**value**) différente.

La balise à utiliser est toujours un **<input />**, avec cette fois la valeur **radio** pour l'attribut **type**.

Les choses seront plus claires sur l'exemple ci-dessous :

index.html

```
<form method=''post'' action=''traitement.php''>
  <p>
    Veuillez indiquer la tranche d'âge dans laquelle vous vous
    situez :<br />
```


index.html suite

```
<input type='radio' name='age' value='moins15' id='moins15'
/> <label for='moins15'>Moins de 15 ans</label><br />

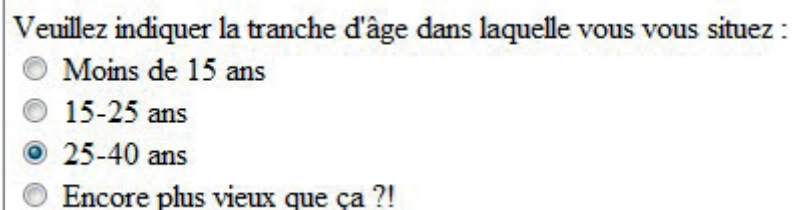
<input type='radio' name='age' value='medium15-25' id='me-
dium15-25' /> <label for='medium15-25'>15-25 ans</label><br />

<input type='radio' name='age' value='medium25-40' id='me-
dium25-40' /> <label for='medium25-40'>25-40 ans</label><br />

<input type='radio' name='age' value='plus40' id='plus40' />
<label for='plus40'>Encore plus vieux que ça ?!</label>

</p>
</form>
```

Ce qui nous donne la figure suivante :



Veuillez indiquer la tranche d'âge dans laquelle vous vous situez :

- ☐ Moins de 15 ans
- ☐ 15-25 ans
- ☒ 25-40 ans
- ☐ Encore plus vieux que ça ?!

Pourquoi avoir mis le même nom pour chaque option ? Tout simplement pour que le navigateur sache de quel « groupe » le bouton fait partie.

Essayez d'enlever les attributs **name**, vous verrez qu'il devient possible de sélectionner tous les éléments d'options. Or, ce n'est pas ce que l'on veut ; c'est pour cela qu'on les « lie » entre eux en leur donnant un nom identique.

Vous noterez que cette fois on a choisi un id différent de **name**. En effet, les valeurs de **name** étant identiques, on n'aurait pas pu différencier les **id** (et vous savez bien qu'un **id** doit être unique !). Voilà donc pourquoi on a choisi de donner à l'**id** la même valeur que **value**.

Si vous avez deux zones d'options différentes, il faut donner un **name** unique à chaque groupe.

Les listes déroulantes :

Les listes déroulantes sont un autre moyen élégant de faire un choix parmi plusieurs possibilités. Le fonctionnement est un peu différent. On va utiliser la balise **<select>** **</select>** qui indique le début et la fin de la liste déroulante. On ajoute l'attribut **name** à la balise pour donner un nom à la liste.

Puis, à l'intérieur du **<select>** **</select>**, nous allons placer plusieurs balises **<option>** **</option>** (une par choix possible). On ajoute à chacune d'elles un attribut **value** pour pouvoir identifier ce que le visiteur a choisi.

Voici un exemple d'utilisation :

index.html

```
<form method='post' action='traitement.php'>
  <p>
    <label for='pays'>Dans quel pays habitez-vous ?</label><br/>
    <select name='pays' id='pays'>
      <option value='france'>France</option>
      <option value='espagne'>Espagne</option>
      <option value='italie'>Italie</option>
      <option value='royaume-uni'>Royaume-Uni</option>
      <option value='canada'>Canada</option>
      <option value='etats-unis'>États-Unis</option>
      <option value='chine'>Chine</option>
      <option value='japon'>Japon</option>
    </select>
  </p>
</form>
```

Le résultat obtenu est représenté à la figure suivante :



Dans quel pays habitez-vous ?

Espagne

France

Espagne

Italie

Royaume-Uni

Canada

Etats-Unis

Chine

Japon

Si vous voulez qu'une option soit sélectionnée par défaut, utilisez cette fois l'attribut **selected** :

index.html

```
<option value='canada' selected>Canada</option>
```

Vous pouvez aussi grouper vos options avec la balise `<optgroup>` `</optgroup>`. Dans notre exemple, pourquoi ne pas séparer les pays en fonction de leur continent ?

index.html

```
<form method='post' action='traitement.php'>
  <p>
    <label for='pays'>Dans quel pays habitez-vous ?</label><br/>
    <select name='pays' id='pays'>
      <optgroup label='Europe'>
        <option value='france'>France</option>
        <option value='espagne'>Espagne</option>
        <option value='italie'>Italie</option>
        <option value='royaume-uni'>Royaume-Uni</option>
      </optgroup>
      <optgroup label='Amérique'>
        <option value='canada'>Canada</option>
        <option value='etats-unis'>Etats-Unis</option>
      </optgroup>
      <optgroup label='Asie'>
        <option value='chine'>Chine</option>
        <option value='japon'>Japon</option>
      </optgroup>
    </select>
  </p>
</form>
```

Le résultat obtenu est représenté à la figure suivante :



The screenshot shows a web form with the title "Dans quel pays habitez-vous ?". Below the title is a dropdown menu. The menu is currently open, showing a list of countries. The countries are grouped by continent, with the continent name in bold. The countries listed are: Espagne (highlighted), France, Italie, Royaume-Uni, Amérique, Canada, Etats-Unis, Asie, Chine, and Japon.

IV. Finaliser et envoyer le formulaire

Nous y sommes presque. Il ne nous reste plus qu'à agrémenter notre formulaire de quelques dernières fonctionnalités (comme la validation), puis nous pourrions ajouter le bouton d'envoi du formulaire.

Regrouper les champs :

Si votre formulaire grossit et comporte beaucoup de champs, il peut être utile de les regrouper au sein de plusieurs balises `<fieldset>`. Chaque `<fieldset>` peut contenir une légende avec la balise `<legend>`. Regardez cet exemple :

```
<form method="post" action="traitement.php">

  <fieldset>
    <legend>Vos coordonnées</legend> <!-- Titre du fieldset -->

    <label for="nom">Quel est votre nom ?</label>
    <input type="text" name="nom" id="nom" />

    <label for="prenom">Quel est votre prénom ?</label>
    <input type="text" name="prenom" id="prenom" />

    <label for="email">Quel est votre e-mail ?</label>
    <input type="email" name="email" id="email" />

  </fieldset>

  <fieldset>
    <legend>Votre souhait</legend> <!-- Titre du fieldset -->

    <p>
      Faites un souhait que vous voudriez voir exaucé :

      <input type="radio" name="souhait" value="riche" id="riche" /> <label for="riche">Etre
riche</label>
      <input type="radio" name="souhait" value="celebre" id="celebre" /> <label
for="celebre">Etre célèbre</label>
      <input type="radio" name="souhait" value="intelligent" id="intelligent" /> <label
for="intelligent">Etre <strong>encore</strong> plus intelligent</label>
      <input type="radio" name="souhait" value="autre" id="autre" /> <label for="autre">Autre...
</label>
    </p>

    <p>
      <label for="precisions">Si "Autre", veuillez préciser :</label>
      <textarea name="precisions" id="precisions" cols="40" rows="4"></textarea>
    </p>
  </fieldset>
</form>
```

Le résultat obtenu est représenté à la figure suivante :

Vos coordonnées

Quel est votre nom ?

Quel est votre prénom ?

Quel est votre e-mail ?

Votre souhait

Faites un souhait que vous voudriez voir exaucé :

☐ Etre riche

☐ Etre célèbre

☐ Etre **encore** plus intelligent

☐ Autre...

Sélectionner automatiquement un champ :

Vous pouvez placer automatiquement le curseur dans l'un des champs de votre formulaire avec l'attribut **autofocus**. Dès que le visiteur chargera la page, le curseur se placera dans ce champ. Par exemple, pour que le curseur soit par défaut dans le champ « prénom » :

index.html

```
<input type='text' name='prenom' id='prenom' autofocus />
```

Rendre un champ obligatoire :

Vous pouvez faire en sorte qu'un champ soit obligatoire en lui donnant l'attribut **required** :

index.html

```
<input type='text' name='prenom' id='prenom' required />
```

Le navigateur indiquera alors au visiteur, si le champ est vide au moment de l'envoi, qu'il doit impérativement être rempli.

Le bouton d'envoi :

Il ne nous reste plus qu'à créer le bouton d'envoi. Là encore, la balise **<input />** vient à notre secours. Elle existe en quatre versions :

- **type="submit"** : le principal bouton d'envoi de formulaire. C'est celui que vous utiliserez le plus souvent. Le visiteur sera conduit à la page indiquée dans l'attribut **action** du formulaire ;
- **type="reset"** : remise à zéro du formulaire ;

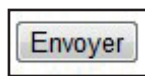
- **type="image"** : équivalent du bouton **submit**, présenté cette fois sous forme d'image. Rajoutez l'attribut **src** pour indiquer l'URL de l'image ;
- **type="button"** : bouton générique, qui n'aura (par défaut) aucun effet. En général, ce bouton est géré en JavaScript pour exécuter des actions sur la page.

On peut changer le texte affiché à l'intérieur des boutons avec l'attribut **value**.
Pour créer un bouton d'envoi, on écrira donc par exemple :

```
index.html
```

```
<input type='submit' value='Envoyer' />
```

Ce qui nous donne ceci :



Lorsque vous cliquez sur le bouton « Envoyer », le formulaire vous amène alors à la page indiquée dans l'attribut **action**. Il faudra créer la page PHP qui traite les informations saisies dans le formulaire.