

I. Les formulaires

Lorsque l'on crée un site web, plusieurs types d'utilisation s'offre à nous. Dans notre cas, nous avons principalement utilisé nos sites web afin de présenter de l'information à un utilisateur. Il peut arriver que nous ayons besoin de récupérer des informations en provenance de l'utilisateur. Pour ce faire, une des méthodes les plus utilisées reste le formulaire.

a. Création du formulaire

Dans ce TP, nous allons compléter la page contact que nous avons créée lors du TP précédent. Pour ce faire, vous allez créer un nouveau projet nommé « TP3_nom_prénom » dans lequel vous allez copier les sources du TP précédent. Si ce dernier a été correctement réalisé, vous devriez disposer à minima :

- D'un dossier « HTML » contenant toutes les pages de votre site à l'exception de « index.html ».
- D'un dossier « IMG » contenant toutes les images de votre site, quelles que soient leurs extensions.
- D'un dossier « CSS » contenant toutes les feuilles de style de votre site.

Dans le dossier HTML doit se trouver un fichier « contact.html » (le créer si ce n'est pas le cas), ouvrez-le et reproduisez-y le style que vous avez déjà utilisé dans la page « index.html ».

Ajouter dans le cœur de la page la balise suivante :

```
<form name="contactForm" id="contactForm" method="GET" action="https://www.w3schools.com/action_page.php">
  <!-- -->
</form>
```

Il s'agit d'une balise « form », elle a pour but de rassembler les éléments de formulaire qui seront présentés à l'utilisateur. En d'autres termes, l'intégralité des autres balises de saisie que nous serons amenés à voir durant ce TP devra toujours se trouver dans une balise de type « form ».

On peut identifier dans cette balise « form » 4 attributs :

- « name » qui donne un moyen de sélectionner le formulaire par son nom.
- « id » qui est un identifiant unique permettant de sélectionner le formulaire par son id.
- « method » qui désigne la méthode utilisée pour l'envoi des données. Il existe 2 types de méthodes à savoir : GET et POST.
- « action » qui désigne la page ciblée par le formulaire qui recevra les données au moment de l'envoi. **Attention :** La page ciblée recevra toutes les données du formulaire quelle que soit la méthode d'envoi (GET ou POST), toutefois, par souci de clarté nous préférons utiliser dans ce TP la méthode GET qui à l'inverse de la méthode POST permet l'envoi des données en clair au travers de l'URL. En temps normal la page ciblée par l'attribut « action » est une page PHP provenant de votre propre site, mais étant donné que nous n'avons pas les compétences (JavaScript ou PHP) pour pouvoir traiter les formulaires nous serons contraints d'utiliser une page de test disponible sur « w3schools ».

b. Structure du document (maquette)

Voici un exemple de structure que vous pouvez donner à votre formulaire :

La maquette du formulaire est divisée en deux sections principales :

- Besoin d'un renseignement :**
 - Nom :
 - Prénom :
 - Sexe :
 - Email :
 - Téléphone :
 - Message :
- Nous envoyer votre CV :**
 - Nom :
 - Prénom :
 - Email :
 - Votre CV :

Entre les sections, il y a des champs supplémentaires :

- Date naissance : - -
- Langue : ☒ Fr ☐ En ☐ Autre
- Votre avis sur le site : 10
- Réponse par : ☐ SMS ☐ e-mail ☐ Pigeon voyageur

En bas du formulaire, il y a deux boutons : **Envoyer** et **Réinitialiser**.

Voici une liste des éléments que l'on retrouve dans ce formulaire :

- Deux zones de regroupement afin d'identifier la partie du formulaire « besoin d'un renseignement » de la partie du formulaire « Nous envoyer votre CV ».
- Un champ de saisie texte pour le prénom.
- Une liste déroulante à choix limité pour le sexe.
- Un champ de saisie mail pour l'adresse e-mail.
- Un champ de saisie pour le téléphone.
- Trois champs de saisie numérique pour la date de naissance.
- Une zone de sélection à choix unique pour la langue.
- Un curseur pour permettre la notation.
- Une zone de cases à cocher permettant un choix multiple.
- Un bouton permettant l'envoi des données.
- Un bouton permettant de réinitialiser le formulaire.

La liste proposée ci-dessus comporte le minimum obligatoire que doit contenir votre formulaire. Vous pouvez bien entendu rajouter d'autres éléments en vous aidant de « w3schools ». (https://www.w3schools.com/html/html_form_elements.asp)

c. Les regroupements

Les regroupements permettent d'identifier des zones au sein de votre formulaire, dans notre cas votre formulaire devra disposer d'au moins deux zones distinctes.

```
<fieldset>
  <legend>Besoin d'un renseignement :</legend>
  <!-- Eléments de formulaire ici -->
</fieldset>
```

d. Les champs texte

Les champs texte sont les champs permettant à l'utilisateur de saisir du texte et sont utilisables au travers des balises « input ». Les champs de texte sont généralement associés à une balise de type label afin de spécifier à l'utilisateur ce que l'on attend de lui :

```
<label for="nom">Nom :</label>  
<input type="text" id="nom" name="nom" value="Mickey">
```

Dans le cas où l'on veut ajouter un label au champ texte, il faut spécifier l'attribut « for » qui aura pour valeur l'« id » de l'élément auquel on souhaite l'associer.

e. Les listes déroulantes

Il existe plusieurs méthodes afin de limiter les possibilités de réponse de l'utilisateur. Dans le cas d'une liste déroulante, l'objectif est de proposer une grande diversité de valeur tout en limitant l'espace utilisé :

```
<select>  
  <option value="volvo">Volvo</option>  
  <option value="saab">Saab</option>  
  <option value="mercedes">Mercedes</option>  
  <option value="audi">Audi</option>  
</select>
```

f. Les champs email

Le champ « email » permet à l'utilisateur de saisir du texte tout en lui imposant un certain nombre de contraintes. En effet, les récentes versions d'HTML permettent d'effectuer une vérification très basique au moment de la saisie d'un « email » afin de vérifier la présence d'un « @ » suivi d'un nom de domaine (XXX.XX).

Attention : il n'en reste pas moins fondamental de vérifier dès que possible que l'information qui sera envoyée ou qui a été envoyée correspond bien au format attendu. Le champ email et les messages d'erreur qu'il génère ne sont en aucun cas une sécurité suffisante.

```
<input type="email" name="emailaddress">
```

g. Les champs tel

Les champs « tel » fonctionnent exactement sur le même principe que les champs « mail » à la différence que les champs « tel » sont destinés à recevoir des numéros de téléphone. Il existe toutefois l'attribut « pattern » qui devra être utilisé obligatoirement dans le cas d'un champ « tel ».

```
<input type="tel" name="tel" pattern="" required>
```

L'attribut « pattern » va permettre d'imposer un modèle de ce qui est attendu dans le champ. A vous de trouver quel est le « pattern » à utiliser dans le cas d'un numéro de téléphone portable français.

h. Les champs numériques

Dans le cas où l'on souhaite recueillir auprès de l'utilisateur une valeur numérique, on peut soit lui laisser saisir cette valeur au clavier ou alors lui imposer un intervalle duquel il ne pourra pas sortir. Cette méthode permet notamment de borner la valeur pouvant être saisie par l'utilisateur :

```
<input type="number" name="jour" min="1" max="5">
```

i. Le curseur numérique

Le curseur numérique fonctionne de la même manière que les champs numériques, la seule différence est que le curseur numérique possède un graphisme plus travaillé :

```
<input type="range" name="points" min="0" max="10">
```

j. Les sélecteurs à choix unique

Cet élément plus couramment appelé « radio bouton » permet de disposer des choix les uns à côté des autres afin de proposer à l'utilisateur de sélectionner un des choix (**et un seul**) :

```
<input type="radio" name="age" value="30"> 0 - 30<br>
<input type="radio" name="age" value="60"> 31 - 60<br>
<input type="radio" name="age" value="100"> 61 - 100<br>
```

Attention : afin de lier les radios boutons entre eux, il faut impérativement que le nom des radios boutons soit le même.

k. Les sélecteurs à choix multiple

Cet élément plus couramment appelé « case à cocher », permet de proposer à l'utilisateur plusieurs choix, ce dernier pourra alors ne pas cocher de case, en cocher une seule ou plusieurs :

```
<input type="checkbox" name="vehicle1" value="Bike"> I have a bike<br>
<input type="checkbox" name="vehicle2" value="Car"> I have a car<br>
<input type="checkbox" name="vehicle3" value="Boat" checked> I have a boat<br>
```

Attention : À la différence des radios boutons les checkbox ne sont pas liés entre elles par défaut. On peut toutefois les regrouper dans un tableau en appliquant le même nom à chaque élément suivi de « [] » :

```
<input type="checkbox" name="vehicle[]" value="Bike"> I have a bike<br>
<input type="checkbox" name="vehicle[]" value="Car"> I have a car<br>
<input type="checkbox" name="vehicle[]" value="Boat" checked> I have a boat<br>
```

l. Un peu de CSS !

Dans le cas des champs qui sont vérifiés par le navigateur, on a la possibilité d'utiliser deux pseudo-classes afin d'appliquer un style sur le champ lorsque celui-ci est correct ou incorrect (« valid » ou « invalid »). À vous de trouver comment utiliser ces pseudo-classes et de les mettre en œuvre dans votre site.

m. Lien de dépôt

Amaury AUGUSTE : https://yncrea-my.sharepoint.com/:f/g/personal/amaury_auguste_yncrea_fr/EomfOy3kJyhBoguaMH2Hqj4B_9HkIBZExMj3rTCun3XTqw

Renaud BOYER : https://yncrea-my.sharepoint.com/:f/g/personal/amaury_auguste_yncrea_fr/Er1dRreEhP1BkI9KDRu01wEB8bfHzZWnvqVdxCiFTxH_bw

n. Pour aller plus loin

Le prochain TP traitera des API, vous pouvez essayer de comprendre ce qu'est une API et essayer d'en intégrer une dans votre site (météo par exemple).