

Leçon 6 : Introduction aux formulaires et iframes

UVCI
APR-2102-2-L6



The screenshot shows a web form titled "Formulaire de réservation de billet". It is divided into two main sections: "Vos coordonnées personnelles" and "Votre voyage".

Vos coordonnées personnelles

At the top of this section are four radio buttons for gender: ☒ Mr, ☐ Mme, ☐ Mlle, and ☐ M. Below these are two text input fields: "Nom :" and "Prénom :".

Votre voyage

This section contains three elements: a "Destination :" text input field, a "Pays :" dropdown menu currently showing "France", and a "Type transport :" row with three checkboxes: ☐ Avion, ☒ Bateau, and ☒ Train.

DIABATE Nagbégna, ©2017 Université Virtuelle
de Côte d'Ivoire (UVCI) - tous droits réservés

Table des matières



I - Objectifs	3
II - Section 1 : Définition, rôle et déclaration des cadres et des formulaires	4
1. 1. les Iframes	4
2. 2. Les formulaires	4
III - Section 2 : Contenu d'un formulaire	6
1. 1. La balise <code><fieldset> </fieldset></code>	6
2. 2. La balise <code><input/></code>	6
3. 3. Balise <code><textarea></code>	8
4. 4. Balise <code><select></code>	8
5. Exercice : Testez vos connaissances avant le devoir !	9
IV - Solutions des exercices	10



Objectifs

À la fin de cette leçon, vous serez capable de :

- *comprendre* l'idée et le fonctionnement des formulaires ; et,
- *concevoir* des formulaires avec HTML ; et
- *utiliser* les cadres ou iframes

Section 1 : Définition, rôle et déclaration des cadres et des formulaires



1. 1. les Iframes

Référence : ce cours est principalement inspiré du cours sur le site *Chiny.me*

La balise `<iframe>` est utilisée pour intégrer, insérer un autre document dans le document HTML courant sans influencer le fonctionnement de l'un d'eux. La balise `<iframe>` est prise en charge dans tous les principaux navigateurs. Les attributs spécifiques à cet élément sont :

- *Name* : Spécifie le nom de la fenêtre
- *Height* : Spécifie la hauteur de la fenêtre
- *Width* : Spécifie la largeur de la fenêtre
- *Src* : Spécifie la source du média
- *frameborder* : pour ajouter une bordure et indiquer l'épaisseur de la bordure

scrolling : Si la page qui doit s'afficher est plus longue que en hauteur de ce que est défini, on peut ajouter le défilement avec `scrolling="yes"`.

Exemple :

```
1 <iframe src="moncv.html" name="CvFrame" id="CvFrame" width="200" height="150"
  scrolling="no" frameborder="0">
2 Cette partie devrait afficher la page de mon CV dans ma page courante (ici)
3 </iframe>
```

2. 2. Les formulaires

Un formulaire HTML quant à lui, est ensemble de questions/réponses permettant de rendre la page Web plus interactive en la rendant capable de dialoguer avec l'internaute à travers des champs de saisie (qui reçoivent les réponses aux questions) et boutons (pour les validations ou annulations). Un formulaire permet à l'internaute de saisir du texte ou de valider un choix ou encore de sélectionner une entrée. Ces informations seront ensuite (dans la plupart des cas) envoyées au serveur pour les traiter. Les formulaires sont beaucoup utiles dans le cadre des sites web dynamique car sont généralement traités avec les langages de programmation avancé tels que PHP, JSP, ASP.

Pour déclarer (ouvrir) un formulaire, on utilise la balise `<form>` puis on termine (ferme) le formulaire avec la balise `</form>`. Cette balise a plusieurs attributs dont :

- *action= "..."* : c'est la destination du formulaire : page (ou script) appelé lorsqu'on soumet le formulaire. C'est cette page qui recevra toutes les données du formulaire. Si cet attribut est laissé vide, le formulaire sera envoyé à la page qui l'a affiché.
- *method= "..."* : `method="POST"` ou `method="GET"`. La méthode `"POST"` signifie que les valeurs transmises seront cachées. Utiliser `method="GET"` signifie que le contenu passera dans l'URL de la

page : *?champ=valeur&champ2=valeur2* ... Beaucoup moins esthétique, surtout lorsqu'il y a des champs texte, des boutons, et que la longueur est limitée à 255 caractères.

- *id="..."*, *name="..."* : Sont utilisés pour identifier le formulaire dans la page .

Supposons que nous voulons créer un formulaire d'inscription nommé "inscription" utilisant la méthode "*POST*" et faisant appel à la page "*inscription.php*" pour assurer le traitement après envoi.

Le code HTML ressemblera à ceci:

```
1 <form id="inscription" name="inscription" method="POST" action="inscription.php">
2   <!-- contenu du formulaire -->
3 </form>
```

Section 2 : Contenu d'un formulaire



La balise `<form>` contient ce que l'on appelle les champs de formulaire. Ce sont des objets comme des zones de texte ou des boutons. Ces champs peuvent être intégrés de diverses manières.

1. 1. La balise `<fieldset>` `</fieldset>`

Cette balise sert à encadrer un ensemble de champs pour regrouper des objets de même connotation.

Exemple : utiliser un *fieldset* pour les coordonnées personnelles (nom, prénom, adresse), un autre pour le choix du produit (quantité, couleur, taille ...). Cette délimitation n'est pas obligatoire mais apporte un plus sur l'aspect ergonomique.

2. 2. La balise `<input/>`

La balise *input* est la balise essentielle des formulaires, car elle permet de créer un bon nombre d'éléments "interactifs". La syntaxe de cette balise est la suivante :

```
1 <input type="Nom du champ" value="Valeur par défaut" name="Nom de l'élément"/>
```

L'attribut *name* est essentiel car il permettra au script de traitement de connaître le champ associé à la paire nom/valeur, c'est-à-dire que le nom du champ sera suivi du caractère "=" puis de la valeur entrée par l'utilisateur, ou dans le cas contraire de la valeur par défaut repéré par l'attribut *value*.

- *L'attribut type* : Il permet de préciser le type d'élément que représente la balise *INPUT*, voici les valeurs que ce champ peut prendre :

- *text*: Permet d'avoir une zone de texte normale (adapté pour les noms, emails, login...)
- *password*: Permet d'avoir une zone de mot de passe où les caractères sont masqués.
- *radio*: Pour créer un bouton radio. Il s'agit d'un petit cercle qu'on peut cocher et qui permet de faire un choix unique. Le fait de cocher un autre bouton radio décoche le premier.
- *checkbox*: Permet de créer une case à cocher. C'est un petit carré qu'on peut cocher ou décocher. Il est destiné à faire des choix multiples.
- *file*: Permet de créer le champ de chargement de fichier (comme une pièce jointe d'un email).
- *date* : Permet de créer le champ de sélection de date (HTML5)
- *number* : Permet de créer le champ de saisi ou sélection qui ne recevra que les nombres (HTML5)
- *url* : Permet de créer le champ de saisi d'adresse URL (HTML5)
- *hidden*: Pour créer un champ caché. C'est comme une zone de texte mais qui n'est pas visible sur le navigateur. Il sert à stocker provisoirement des données jusqu'à la phase de traitement.
- *reset*: Permet de créer un bouton d'annulation. Si le formulaire contient des champs qu'on a déjà saisis, ce bouton permet de tout initialiser.
- *submit*: C'est le bouton d'envoi. C'est lui qui permet de poster le formulaire vers la page définie dans l'attribut action de la balise `<form>`.

- *L'attribut name* : Permet de donner un nom au champ de formulaire. Le nom sert d'identifiant du champ et doit être unique.

- *L'attribut value* : Sert à définir la valeur par défaut d'un champ. S'il s'agit d'une zone de texte, le fait de déclarer l'attribut value force le navigateur à initialiser ce champ avec le texte faisant office de valeur de l'attribut. Si le champ est un bouton, alors le texte de l'attribut value constitue l'étiquette (ou label). C'est le texte visible sur le bouton.
- *L'attribut size* : Contient un entier comme valeur. Il définit la largeur du champ en caractères. Par défaut, une zone de texte mesure 20 caractères ; cela veut dire qu'on peut voir jusqu'à 20 caractères à la fois même si le texte est plus grand.

Voici un exemple de formulaire en HTML qui tient lieu de Travaux Pratique

et le code HTML pour réaliser ce formulaire est le suivant :

```

1 <form method="POST" action="inscription.php">
2 <fieldset><legend>Formulaire d'inscription</legend>
3   Nom : <input type="text" name="" value=""><br><br>
4   Prénoms: <input type="text" name="" value=""><br><br>
5   Civilité<br>
6   <input type="radio" name="civilite" value="Mr">Monsieur <br>
7   <input type="radio" name="civilite" value="Mme">Madame <br><br>
8   sexe:
9     <select name="sexe" align="left">
10       <option value="H">Homme</option>
11       <option selected value="F">Femme</option>
12     </select><br><br>
13   login : <input type="text" name="login" value=""><br><br>
14   mot de passe : <input type="password" name="" value=""><br><br>
15   Présentation : <br>
16   <textarea name="" rows="" cols=""></textarea><br><br>
17   <input type="submit" name="submit" value="Valider">
18 </fieldset>
19 </form>

```

Vous pouvez le tester chez vous, balise après balise pour voir les différentes étapes de construction.

3. 3. Balise `<textarea>`

La balise `<textarea>` permet de définir un espace de texte. Un espace de texte est une grande zone de texte qui permet d'écrire des paragraphes entiers. Elle dispose des attributs suivants:

- *cols* : définit la largeur en nombre de caractères de l'espace de texte. Le fait de dépasser la largeur définie entraîne un retour automatique à la ligne.
- *rows* : définit la hauteur de l'espace de texte en nombre de ligne. Si le texte dépasse le nombre de lignes défini, alors une barre de défilement qui permet d'atteindre le reste des lignes apparaît.

4. 4. Balise `<select>`

La balise `<select>` permet de déclarer une liste de sélection (liste déroulante). Pour remplir cette liste avec les valeurs, on doit déclarer une autre balise à l'intérieur, c'est la balise `<option>`. Dans la balise `<option>` on déclare le mot à afficher (en tant qu'élément de la liste).

[Solution n°1 p 10]

- ☐ un bouton
- ☐ une zone de texte dont la valeur par défaut est T1
- ☐ une zone de texte nommée T1

- ☐ une zone de texte
- ☐ une case à cocher
- ☐ une case d'option

- ☐ réinitialiser les contrôles du formulaire
- ☐ envoyer les données du formulaire par E-mail
- ☐ envoyer les données du formulaire

- ☐ `<p></p>`
- ☐ `<formulaire></formulaire>`
- ☐ `<form></form>`

- ☐ value
- ☐ id
- ☐ name

Solutions des exercices

> Solution n°1

Exercice p. 9

Exercice

- ☐ un bouton
- ☐ une zone de texte dont la valeur par défaut est T1
- ☒ une zone de texte nommée T1

`type=text` désigne une zone de texte et `name=T1` indique le nom du champ de saisie

Exercice

- ☐ une zone de texte
- ☒ une case à cocher
- ☐ une case d'option

`type=checkbox` désigne une case à cocher et le « ON » signifie que le champ est actif (coché par défaut)

Exercice

- ☐ réinitialiser les contrôles du formulaire
- ☐ envoyer les données du formulaire par E-mail
- ☒ envoyer les données du formulaire

Exercice

- ☐ `<p></p>`
- ☒ `<formulaire></formulaire>`
- ☐ `<form></form>`

Exercice

- ☐ value
- ☐ id
- ☒ name

sachez que l'attribut "`id`" permet de créer des ancres pour positionner le navigateur directement dans un endroit de la page. C'est un attribut lu par le navigateur et est utile dans l'utilisation avec JavaScript et HTML mais indispensable dans le traitement des données du formulaire.

