

## CHAPITRE IV : LE FORMULAIRE

### I) INTRODUCTION :

Les formulaires permettent aux utilisateurs de saisir des données ; ces données sont ensuite traitées par le serveur.

Par exemple, lorsque vous vous connectez sur le site quelconque, vous saisissez les caractéristiques du trajet qui vous intéresse puis le serveur traite votre demande et vous affiche les trains correspondant. Le serveur peut traiter lui-même la demande de l'utilisateur ou l'enregistrer / la transmettre à un être humain. C'est le cas lorsque vous écrivez des messages dans des forums électroniques. Ou bien encore lorsque vous rédigez votre avis sur un site.

### II) LA DECLARATION D'UN FORMULAIRE

La définition des différents éléments du formulaire se fait en utilisant la balise **<FORM>**. Il ne faut pas oublier d'écrire la balise **</FORM>** pour déclarer la fin du formulaire.

L'action à réaliser pour traiter le formulaire doit être précisée en utilisant les deux attributs suivants :

1- L'attribut **action** : désigne le fichier PHP qui va traiter, sur le serveur, les informations saisies dans le formulaire;

2- L'attribut **method** : définit la méthode de transfert des données vers le serveur. Les deux valeurs possibles sont **get** et **post**. La syntaxe habituelle est donc :

```
<form method="post" action="traiteform.php">
<fieldset>
<legend>Ceci est le titre de mon formulaire</legend>
<!-- la suite sera le corps du formulaire -->
...
</fieldset>
</form>
```

La balise **<fieldset>** permet, à l'intérieur d'un formulaire, de créer des blocs visuels permettant d'améliorer la présentation des formulaires. La balise sert ensuite à donner un titre au bloc de formulaire.

### III) LES ELEMENTS DE FORMULAIRES :

Les éléments de formulaires sont répartis en 3 classes :

- **input** : Champs de saisie de texte et différents types de boutons
- **select** : Liste (menus déroulants et ascenseurs)

- **textarea** : Zone de saisie de texte libre .

L'attribut **name** est obligatoire, car c'est lui qui permet d'identifier les données côté serveur et ainsi de récupérer le contenu saisi par l'utilisateur. Il doit être unique, sauf dans le cas d'un ensemble composé de plusieurs éléments (boutons radio par exemple). Seuls les boutons **submit** et **reset** n'ont pas besoin de cet attribut.

L'attribut **value** est commun à ces balises, mais son rôle est différent selon le **type** d'élément utilisé.

La balise **<label>** est juste pour les différents libellés.

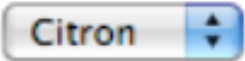
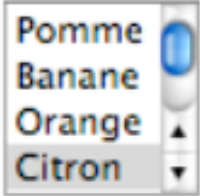
### III-1) L'élément INPUT :

Cet élément est utilisé pour définir des zones d'entrée de texte simple, des boutons, des cases à cocher ou des boutons radio. Le type d'élément dont il s'agit sera précisé en utilisant l'attribut **type**.

Type	Code	Résultat
sans	<code>&lt;input name="ident" /&gt;</code>	<input type="text"/>
	<code>&lt;input name="ident" value="Par défaut" /&gt;</code>	<input type="text" value="Par défaut"/>
submit	<code>&lt;input type="submit" value="Envoyer" name="Envoyer" /&gt;</code>	<input type="submit" value="Envoyer"/>
checkbox	<code>&lt;input type="checkbox" name="pfm[]" value="linux" checked = "checked" /&gt; Linux&lt;br/&gt; &lt;input type="checkbox" name="pfm[]" value="dos" /&gt; Dos&lt;br/&gt; &lt;input type="checkbox" name="pfm[]" value="win" /&gt; Windows</code>	<input checked="" type="checkbox"/> Linux <input type="checkbox"/> Dos <input type="checkbox"/> Windows
radio	<code>&lt;input type="radio" name="media" value="cd" checked= "checked"/&gt; CD-ROM&lt;br/&gt; &lt;input type="radio" name="media" value="dk" /&gt; Disquette</code>	<input checked="" type="radio"/> CD-ROM <input type="radio"/> Disquette
password	<code>&lt;input type="password" name="pass" size="4"/&gt;</code>	<input type="password" value="...."/>
reset	<code>&lt;input type="reset" value="Effacer" /&gt;</code>	<input type="reset" value="Effacer"/>

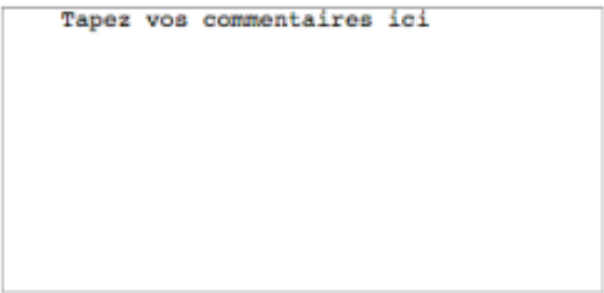
### III-2) L'élément SELECT :

Cet élément sert à définir des listes (menus déroulants ou ascenseurs). Il est utilisé avec l'élément **<option>**.

Code	Résultat
<pre>&lt;select name="menu"&gt;   &lt;option value="pomme"&gt;Pomme&lt;/option&gt;   &lt;option value="banane"&gt;Banane&lt;/option&gt;   &lt;option value="orange"&gt;Orange&lt;/option&gt;   &lt;option value="citron" selected="selected"&gt;     Citron&lt;/option&gt;   &lt;option value="peche"&gt; Pêche&lt;/option&gt;   &lt;option value="poire"&gt; Poire&lt;/option&gt; &lt;/select &gt;</pre>	
<pre>&lt;select name="menu" size= "4"&gt;   ... &lt;/select&gt;</pre>	

### III-3) L'élément TEXTAREA

Il permet de créer une zone de texte de plusieurs lignes. Il faut spécifier sa taille avec les attributs **rows** et **cols**.

Code	Résultat
<pre>&lt;textarea name="comm" rows="10" cols="40"&gt;   Tapez vos commentaires ici &lt;/textarea&gt;</pre>	

## IV) LES LISTES :

Les listes sont très utilisées en HTML pour ordonner de l'information hiérarchiquement, en utilisant ou non des chiffres pour numéroter les sections et sous-sections. On parle alors de listes non-ordonnées (<UL>) et de listes ordonnées (<OL>). Le langage HTML définit trois types de listes :

- La liste ordonnée ;
- La liste non ordonnée ;
- La liste de définition.

### IV-1) LA LISTE ORDONNEE :

Conteneur	Type de liste	Effet Visuel
<pre>&lt;ol&gt;   &lt;li&gt; article 1 &lt;/li&gt;   &lt;li&gt; article 2 &lt;/li&gt; &lt;/ol&gt;</pre>	Liste ordonnée	article 1 article 2 article 3

## Variantes d'affichage des listes ordonnées

a)

<OL>

<LI>premier point

<LI>deuxième point

<OL>

<LI>premier point secondaire

<LI>deuxième point secondaire

</OL></OL>

donne:

1. premier point

2. deuxième point

1. premier point secondaire

2. deuxième point secondaire

b)

<OL TYPE=A>

<LI>premier item

<LI>deuxième item

</OL>

donne:

A. premier item

B. deuxième item

**NB :** Les variantes <OL TYPE=a>, <OL TYPE=I> donnent respectivement des lettres minuscules, et des lettres majuscules en chiffres romains.

## IV-2) LA LISTE NON-ORDONNEE :

Conteneur	Type de liste	Effet Visuel
<ul> <li> article 1 </li> <li> article 2 </li> </ul>	Non ordonnée	article 1 article 2 article 3

## Variantes d'affichage des listes non-ordonnées

c)

<UL>

<LI>premier point

<LI>deuxième point

</UL>

donne:

• premier point

• deuxième poin

d)

<UL>

<LI>premier point

<LI>deuxième point

<UL>

donne:

• premier point

• deuxième point

○ premier point secondaire

○ deuxième point secondaire

<LI>premier point secondaire  
 <LI>deuxième point secondaire  
 </UL></UL>

**NB** : On peut changer ces choix en utilisant la commande <UL TYPE=disc>, <UL TYPE=circle> ou <UL TYPE=square>

## IV-2) LISTE DEFINITIONS

Une définition est une annotation HTML qui permet d'inscrire d'abord un terme à définir pour lui ajouter en-dessous et en retrait (alinéa) sa définition.

<DL> indique une zone de définitions ("definitions list") à venir ci-dessous (une ou plusieurs)

<DT> ("definition term") indique le premier terme à définir

<DD> donne la définition du premier terme en retrait d'un alinéa

</DL> ("definition list") indique la fin de la zone de définitions.

Pour mettre plusieurs définitions une à la suite des autres, il suffit de répéter la séquence <DT><DD> autant de fois que l'on en a besoin à l'intérieur de la zone. On peut aussi donner plus d'une définition à un terme en réutilisant la commande <DD> plus d'une fois sous la ligne <DT>.

Conteneur	Type de liste	Effet Visuel
<dl> <dt>Terme</dt> <dd>Définition</dd> </dl>	De définition	article 1 définition 1 article 2 définition 2

## V) ATTRIBUTS SPECIFIQUES AUX LISTES:

Attribut	Valeur	Effet Visuel
COMPACT		resserre l'interligne
PLAIN		supprime les puces
SEQNUM		définit le premier numéro
START		définit le premier numéro
CONTINUE		repart du numéro où il s'était arrêté à la liste précédente
TYPE (pour les listes ordonnées)	1 A a I i	numérotation chiffrée (par défaut) numérotation en capitales numérotation en bas de casse numérotation en chiffres romains (I, II, III, IV ...) numérotation en chiffres romains en bas de casse
TYPE (pour les listes non-ordonnées)	circle square disc	puce circulaire puce carrée puce en disque