

L1-ST5-Internet et Programmation

162

### Le langage Html : Les Formulaires

Avant d'introduire quelques fonctionnalités du langage Javascript, nous allons compléter le langage Html avec la possibilité d'afficher des objets pré-dessin     savoir:

### LES FORMULAIRES

L1-ST5-Internet et Programmation

163

### Le langage Html : Les Formulaires

Un formulaire est un ensemble de champs permettant de saisir des donn  es : une fois les champs remplis, on peut « soumettre » les informations du formulaire. Les donn  es sont alors envoy  es du poste client vers le serveur qui a envoy   le formulaire.

L'utilisation d'un formulaire n  cessite (pour l'exploitation des informations) de faire appel   un « programme » capable de faire des interpr  tations (ex: page ASP, programme CGI, script etc...)

D  claration d'un formulaire.

Les balises **<FORM> ... </FORM>**. Ce couple sert   d  limiter le d  but et la fin du formulaire.

L'attribut **ACTION** : donne le nom de la page, du programme CGI, qui recevra les informations lors de la soumission.

L1-ST5-Internet et Programmation

164

### Le langage Html : Les Formulaires

L'attribut **ACTION** : donne le nom de la page, du programme CGI, qui recevra les informations lors de la soumission.

**METHOD** : d  termine la m  thode de passage des param  tres du formulaire vers l'« appelant » .

**GET** : les param  tres (et leur valeur) seront pass  s dans l'URL du CGI appel   (**ACTION**) de la fa  on suivante : nomCGI? param1=val1&param2=val2&...&paramn=valn.

**POST** : les param  tres sont pass  s sous le m  me format, mais ils ne sont pas concat  n  s   l'URL. Ils sont au contraire pass  s dans le corps de la requ  te HTTP (ce qui permet  galement la transmission par courrier).

L1-ST5-Internet et Programmation

165

### Le langage Html : Les Formulaires

L'attribut **NAME** : donne un nom au formulaire (qui permettra ensuite son utilisation avec JavaScript).  
**ENCTYPE** qui permet de spécifier le format dans lequel sont envoyées les données  
 également **TARGET** qui avec JavaScript permet de choisir la cible.

Exemple:

```
<FORM ACTION="mailto:machin@truc.com" METHOD="post"
ENCTYPE="text/plain" NAME="nom_du_formulaire">
déclaration des différents éléments (champs) constituant le
formulaire.
</FORM>
```

Cette déclaration permettra de définir un formulaire qui au moment de la soumission enverra un courriel à l'adresse machin@truc.com

### Le langage Html : Les Formulaires : la forme

Différentes formes peuvent être utilisées pour la collecte d'informations

The image shows several examples of HTML form elements: a text input with 'Valeur d'origine', a password input with asterisks, a 'validation' button, a button labeled 'Appuyez sur le bouton', a text input with 'valeur initiale' and an 'Annuler' button, three checkboxes labeled 'Cocher la case 1', 'Cocher la case 2', and 'Cocher la case 3', three radio buttons labeled 'choix 1', 'ou choix 2', and 'ou choix 3', a large empty text area, and a dropdown menu labeled 'Menu: Opt 2'.

### Le langage Html : Les Formulaires : la forme

Pour la définition d'un élément du formulaire, on utilise le tag **<INPUT>**, ensuite avec l'attribut **TYPE** on peut définir la forme et la fonctionnalité de l'élément. Et avec l'attribut **NAME** on peut donner un nom à l'élément du formulaire.

Voici les formes de formulaires visualisées précédemment:

Attribut **TYPE** avec la valeur:

“**TEXT**” permet de définir une zone de saisie de chaîne de caractères. On peut affecter une valeur de départ avec l'attribut **VALUE**. L'attribut **SIZE** permet de fixer la taille de la zone de saisie (en nombre de caractères).

Exemple

```
<FORM>
<INPUT TYPE="text" VALUE="Valeur d'origine"
NAME="nom_du_formulaire" SIZE="40">
</FORM>
```

The image shows a screenshot of a text input field with a red border, containing the text 'Valeur d'origine'.

### Le langage Html : Les Formulaires : la forme

#### Remarque:

De manière générale, pour chacune des formes que l'on va décrire dans la suite, on pourra trouver (ou devra suivant l'utilisation) la possibilité de donner un nom avec l'attribut **NAME** ainsi qu'avec l'attribut **VALUE** une valeur (soit par défaut, soit à préciser obligatoirement suivant l'utilisation du formulaire).

En fait, lors de la soumission (exemple envoi d'un courriel) les « **NAME** » et « **VALUE** » sont concaténés (avec un signe « = » entre les deux) avant d'être envoyé.

### Le langage Html : Les Formulaires : la forme

Attribut **TYPE** avec la valeur:

“ **PASSWORD** ” permet de définir une **zone de saisie de mot de passe** (saisie masquée). On peut affecter une valeur de départ (mot de passe par défaut) avec l'attribut **VALUE**. L'attribut **SIZE** permet de fixer la taille de la zone de saisie (en nombre de caractères) (**NAME** donner un nom).

Exemple

```
<FORM>
  <INPUT TYPE="password" VALUE="Valeur d'origine"
    NAME="nom de ref" SIZE="40">
</FORM>
```



### Le langage Html : Les Formulaires : la forme

Attribut **TYPE** avec la valeur:

“ **SUBMIT** ” permet de définir un **bouton de « soumission »**. La valeur affichée dans le bouton est fixée avec l'attribut **VALUE**. L'attribut **WIDTH** permet de fixer la taille du bouton (**NAME** donner un nom).

Exemple

```
<FORM>
  <INPUT TYPE="submit"
    VALUE="validation">
</FORM>
```



### Le langage Html : Les Formulaires : la forme

Attribut **TYPE** avec la valeur:

“ **BUTTON** ” permet de définir un **bouton simple** (pas d'appel de type CGI, utilisation avec javascript). La valeur affichée dans le bouton est fixée avec l'attribut **VALUE**. L'attribut **WIDTH** permet de fixer la taille du bouton (**NAME** donner un nom).

Exemple

```
<FORM>
  <INPUT TYPE="button"
    VALUE="Appuyez sur le bouton">
</FORM>
```



### Le langage Html : Les Formulaires : la forme

Attribut **TYPE** avec la valeur:

“ **RESET** ” permet de définir un **bouton de ré-initialisation** (remet les valeurs fixées par **VALUE** aux formulaires). La valeur affichée dans le bouton est fixée avec l'attribut **VALUE**. L'attribut **WIDTH** permet de fixer la taille du bouton (**NAME** donner un nom).

Exemple

```
<FORM>
  <INPUT TYPE="text" VALUE="valeur initiale">
  <INPUT TYPE="reset" VALUE="Annuler">
</FORM>
```



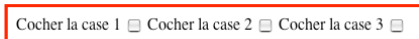
### Le langage Html : Les Formulaire : la forme

Attribut **TYPE** avec la valeur:

“**CHECKBOX**” permet de définir une **case à cocher** (on peut ajouter l’attribut **CHECKED** si l’on souhaite que la case soit cochée à l’apparition de celle-ci) On peut attribuer avec **VALUE** une valeur par défaut différente du texte associé (et **NAME** pour donner un nom).

Exemple

```
<FORM>
Cocher la case 1 <INPUT TYPE="checkbox" NAME="case1"
VALUE="OK">
Cocher la case 2 <INPUT TYPE="checkbox">
Cocher la case 3 <INPUT TYPE="checkbox">
</FORM>
```



L1-STIS-Internet et Programmation

174

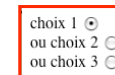
### Le langage Html : Les Formulaire : la forme

Attribut **TYPE** avec la valeur:

“**RADIO**” permet de définir une **série de boutons à cocher à choix unique** (toutes les valeurs de **NAME** doivent être identiques). Un seul « bouton » pourra être affecté de l’attribut **CHECKED**. On peut attribuer avec **VALUE** une valeur par défaut différente du texte associé.

Exemple

```
<FORM>
choix 1 <INPUT TYPE="radio" CHECKED
NAME="bouton" VALUE="B1"><BR>
ou choix 2 <INPUT TYPE="radio" NAME="bouton"
VALUE="B2"> <BR>
ou choix 3 <INPUT TYPE="radio" NAME="bouton"
VALUE="B3"> <BR>
</FORM>
```



L1-STIS-Internet et Programmation

175

### Le langage Html : Les Formulaire : la forme

Les balises **<TEXTAREA> ... </TEXTAREA>** :

permettent de définir une **zone de saisie de texte** (comme <INPUT TYPE="text">) avec plusieurs lignes de saisie.

On peut adjoindre les attributs:

**Name** : pour fixer le nom de l’objet de saisie.

**Cols** : fixe le nombre de colonnes de la zone de saisie.

**Rows** : fixe le nombre de lignes de la zone de saisie.

Exemple

```
<FORM>
<TEXTAREA NAME="Zone" COLS="40" ROWS="5">
</TEXTAREA>
</FORM>
```



L1-STIS-Internet et Programmation

176

### Le langage Html : Les Formulaire : la forme

Les balises **<SELECT> ... </SELECT>** :

permettent de définir une **liste** ou un **menu déroulant** (l’attribut **NAME** est unique pour la liste (ou menu)).

On définit ensuite les choix du menu avec **<OPTION>**

On peut y adjoindre l’attribut **SELECTED** pour faire un choix par défaut.

On peut attribuer avec **VALUE** une valeur par défaut différente du texte associé.

Exemple

```
<FORM>
<SELECT NAME="Menu">
<OPTION> Menu: Opt 1
<OPTION> Menu: Opt 2
<OPTION SELECTED> Menu: Opt 3
</SELECT>
</FORM>
```



L1-STIS-Internet et Programmation

177

### Le langage Html : Les Formulaires : la forme

Les balises **<SELECT> ... </ SELECT >** :

permettent de définir une **liste** ou un **menu déroulant**.

On peut en utilisant l'attribut **SIZE** faire apparaître une liste (plusieurs lignes)

Et on peut adjoindre l'attribut **MULTIPLE** pour autoriser plusieurs choix

Exemple:

```
<FORM>
  <SELECT NAME= "menu" SIZE=4 MULTIPLE>
    <OPTION VALUE="entrée1" >assiette scandinave
    <OPTION VALUE="entrée2" SELECTED>charcuterie
    <OPTION VALUE="plat1">aile de raie
    <OPTION VALUE="plat2">gigot d'agneau
    <OPTION VALUE="fromage" SELECTED>fromage
    <OPTION VALUE="dessert">dessert maison
  </SELECT>
</FORM>
```

