

LES FORMULAIRES HTML5 - DÉVELOPPER DES PAGES WEB EN LIEN AVEC UNE BASE DE DONNÉES

-
- P/2** Introduction
 - P/3** 1. Les champs de saisie
 - P/5** 2. Les zones de saisie multi-lignes
 - P/6** 3. Les cases à cocher et les boutons radio
 - P/8** 4. Les listes de choix (déroulantes ou non)
 - P/10** 5. Label
 - P/11** 6. Encadré et regroupement de zones de saisie
 - P/12** 7. Saisie d'un nom de fichier à joindre
 - P/13** 8. Données masquées
 - P/14** 9 saisie assistée de données particulières
 - 9.1 Adresse email
 - 9.2 Numéro de téléphone
 - 9.3 Choix de dates
 - 9.4 Saisie d'un nombre
 - 9.5 Choix d'une couleur
 - P/16** 10. Soumission (envoi) d'un formulaire
 - 10.1 Submit et Reset
 - 10.2 Les boutons input type button
 - 10.3 Les boutons graphiques button
 - 10.4 Les boutons image
 - P/18** Crédits
-

Introduction

● Introduction



Les formulaires permettent à l'internaute d'effectuer une saisie et d'envoyer les informations saisies depuis son navigateur vers le serveur Web afin d'y être traitées (grâce à un programme tournant sur le serveur Web, script ASP.Net, JSP ou PHP).

Les saisies dans un formulaire peuvent s'effectuer par des éléments graphiques HTML comme des champs de saisie, des zones de texte multiligne, des listes de choix déroulantes, des cases à cocher ou des radio-boutons et bien entendu des boutons de commandes ; HTML5 apporte de nouveaux éléments graphiques permettant une meilleure ergonomie et un meilleur contrôle de saisie (saisie de nombres, de dates, d'adresse email, de couleur, ou encore curseur et barre de progression).

Les formulaires sont encadrés par les balises **<form>** et **</form>**.

La balise **<form>** possède les attributs suivants :

Attributs	Utilité
action	Indique l'url du script serveur (asp .net, jsp, php...) qui doit s'exécuter lors de l'envoi (« la soumission ») du formulaire.
method	post : les données saisies dans le formulaire sont envoyées dans le corps du message http. Elles peuvent être volumineuses et n'apparaissent pas dans la barre d'adresse du navigateur. get : les données saisies sont ajoutées à l'url (visibles en barre d'adresse du navigateur)
enctype	Spécifie le format des données envoyées "text/plain" : cas d'un formulaire classique "multipart/form-data" : cas d'un formulaire avec fichier attaché
name et id	Définit le nom du formulaire et son identifiant (utile pour réaliser ultérieurement des contrôles de saisie en script JavaScript)

Chaque information du formulaire sera identifiée par un élément graphique HTML auquel on précise les attributs **name** et **id** ; *id* sera utilisé pour manipuler les données en JavaScript (contrôles de saisie...) et *name* sera le nom de la variable reçue par le script sur le serveur Web.



Remarque

Dans ce document, nous nous préoccupons uniquement de la présentation des formulaires, de leur design et leur définition en HTML, en vue de leur traitement ultérieur.

1. Les champs de saisie

L'internaute peut saisir une information dans un champ de saisie par l'intermédiaire de l'élément

<input type = "text" .../>.

Cette balise possède les attributs suivants :

Attribut	Utilité
name et id	Nom et identifiant de la donnée. Name est le nom sous lequel l'information saisie ici sera connue et récupérée dans le script serveur déclenché par le formulaire.
size	Permet de définir un nombre de caractères visibles pour le champ.
maxlength	Permet de définir un nombre de caractères maximum saisissable.
value	Texte défini comme valeur par défaut.
readonly	Indique si le contenu de la zone de saisie est modifiable ou pas.
disabled	Indique si la zone de saisie est active ou pas.
required	Indique si cette zone de saisie est obligatoire.
pattern	Permet de préciser une règle de validation à l'aide d'une « expression régulière ».
placeholder	Valeur du texte d'invite qui disparaît dès que l'élément prend le focus.
autofocus	Indique la zone de saisie qui doit prendre le curseur dès le chargement de la page (un seul par page !).
autocomplete	Indique si le mécanisme d'autocomplétion du navigateur doit être activé ou non sur cette zone.

Lorsque l'on veut définir un champ de saisie de type mot de passe qui n'affiche pas les caractères saisis, il faut utiliser la balise suivante qui se définit de la même manière :

<input type = "password"/>.



Exemple

Exemple d'un champ de saisie :

Code HTML correspondant :

```
<form name="saisienom" action="..." method="get">
```

```
<input type="text" name="nom" id="nom" size="50" maxlength="50" value="Julie DUBOIS" />
```

```
</form>
```

HTML5 permet de définir une liste de valeurs à proposer, liste qui s'affine au fur et à mesure de la frappe (principe de l'*auto-completion*) grâce à l'élément **<datalist>** ; cet élément n'a aucun rendu graphique mais il peut être lié à un champ de saisie de données de texte grâce à l'attribut `list` :

```
<input type="text" name="pays" id="pays" size="50" maxlength="50" list="list_pays" />
```

```
<datalist id="list_pays">
```

```
<option value="Belgique" />
```

```
<option value="Canada" />
```

```
<option value="France" />
```

```
<option value="Luxembourg" />
```

```
</datalist>
```

2. Les zones de saisie multi-lignes

Lorsque l'on veut saisir du texte long sur plusieurs lignes, on passe par des zones de saisie multi-lignes à l'aide de l'élément `<textarea>...</textarea>`.

Cette balise possède les attributs suivants :

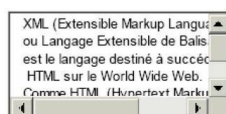
Attribut	Utilité
Name et id	Nom de la donnée. Name est le nom sous lequel l'information saisie ici sera connue et récupérée dans le script serveur déclenché par le formulaire.
rows	Nombre de lignes affichées dans la zone de saisie.
cols	Nombre de caractères affichés par ligne dans la zone de saisie.
readonly	Indique que le contenu de la zone de saisie n'est pas modifiable.
disabled	Indique si la zone de saisie est active ou pas.
required	Indique si cette zone de saisie est obligatoire.
placeholder	Valeur du texte d'invite qui disparaît dès que l'élément prend le focus.

La valeur par défaut contenue dans la zone de saisie multi-ligne est le texte compris entre les balises `<textarea>` et `</textarea>`.



Exemple

Exemple d'une zone de saisie multi-ligne :



Code HTML correspondant :

```
<form name="saisiecommentaire" action="..." method="get">  
<textarea name="commentaire" id="commentaire" cols="30" rows="5">
```

XML (Extensible Markup Language, ou Langage Extensible de Balisage) est le langage destiné à succéder à HTML sur le World Wide Web. Comme HTML (Hypertext Markup Language)...

```
</textarea>
```

```
</form>
```

3. Les cases à cocher et les boutons radio

Les données peuvent être sélectionnées par l'internaute dans des cases à cocher (plusieurs choix), ou dans des boutons radio (choix unique).

Pour cela, on utilise les éléments suivants :

<input type = "checkbox">, pour les cases à cocher, **<input type = "radio">**, pour les boutons radio.

Ces balises possèdent les attributs suivants :

Attribut	Utilité
name et id	Nom de la donnée. Name est le nom sous lequel l'information saisie ici sera connue et récupérée dans le script serveur déclenché par le formulaire. Dans un même groupe de boutons radio, tous les éléments doivent porter le même name de manière à être exclusif l'un de l'autre.
value	Valeur (chaîne de caractères) correspondant à l'option à choisir.
checked	Indique que l'option sera sélectionnée par défaut (un bouton radio par groupe DOIT être présélectionné).
disabled	Indique si la zone de saisie est active ou pas.



Exemple

Exemple de cases à cocher :

☐ Bleu ☒ Rouge ☐ Vert

Code HTML correspondant :

```
<input type="checkbox" name="couleur1" value="Bleu" /> Bleu
```

```
<input type="checkbox" name="couleur2" value="Rouge" checked="checked" /> Rouge
```

```
<input type="checkbox" name="couleur3" value="Vert" /> Vert
```



Attention (important pour les traitements côté serveur en script PHP, ASP...)

Dans les échanges HTTP, le navigateur adresse au serveur Web uniquement les checkbox qui sont cochées.



Exemple

Exemple de boutons radio :

☐ Bleu ☐ Rouge ☒ Vert

Code HTML correspondant :

```
<input type="radio" name="couleur" value="Bleu" /> Bleu
```

```
<input type="radio" name="couleur" value="Rouge" /> Rouge
```

```
<input type="radio" name="couleur" value="Vert" checked="checked" /> Vert
```



Attention

- Les boutons radio représentent les différentes valeurs possibles d'une même information (ici, la couleur choisie) ; les boutons radio correspondant à la même information

doivent donc porter le même nom.

Une seule variable sera transmise au serveur Web.

- Plusieurs zones de saisie différentes et **non exclusives les unes des autres** peuvent porter le même nom dans une page HTML, comme dans le cas des cases à cocher qui

permettent de **sélectionner un ou plusieurs éléments** d'une liste (pe : les messageries e-mail Web).

Dans ce cas, il est préférable de terminer le nom HTML de la zone par des crochets de manière à ce que JavaScript et les scripts serveurs (PHP, ASP...) les reconnaissent automatiquement comme des **tableaux de variables** :

```
<tr><td><input type="checkbox" name="matricule[]" value="123" />Dupont</td></tr>
```

```
<tr><td><input type="checkbox" name="matricule[]" value="133" />Durand</td></tr>
```

```
<tr><td><input type="checkbox" name="matricule[]" value="423" />Duwald</td></tr>
```

4. Les listes de choix (déroulantes ou non)

L'internaute peut sélectionner les informations qu'il désire dans une liste pré-renseignée.

La balise **<select>** permet de définir la liste des choix possibles.

Chaque choix sera défini par une balise **<option>**, contenue dans l'élément **<select>**.

Les attributs possibles de la balise **<select>** sont les suivants :

Attribut	Utilité
name et id	Nom de la donnée. Name est le nom sous lequel l'information saisie ici sera connue et récupérée dans le script serveur déclenché par le formulaire.
size	Si size vaut 1 (valeur par défaut), c'est une liste déroulante. Si size > 1, c'est une liste de choix avec ascenseur qui s'affiche sur plusieurs lignes.
multiple	Dans le cas où size > 1 uniquement. Autorise la sélection de plusieurs valeurs.
value	Valeur de l'option sélectionnée par l'utilisateur.

Les attributs possibles de la balise **<option>** sont les suivants :

Attribut	Utilité
value	Valeur pour l'option correspondante ; elle donnera sa valeur à la boîte select.
selected	Détermine le choix proposé par défaut.



Exemple

Exemple de liste déroulante :



Code HTML correspondant :

```
<select name="couleur">  
<option value="Bleu" selected="selected">Bleu</option>  
<option value="Rouge">Rouge</option>  
<option value="Vert">Vert</option>  
</select>
```




Exemple

Exemple de liste de choix :

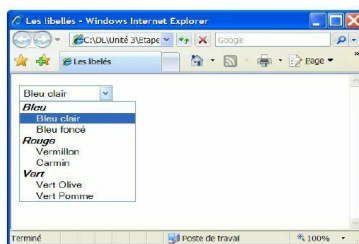


Code HTML correspondant :

```
<select name="couleur" size="3" multiple="multiple">
<option value="Bleu">Bleu</option>
<option value="Rouge">Rouge</option>
<option value="Vert">Vert</option>
</select>
```

Il est possible de regrouper ces options entre elles à l'aide de la balise **<optgroup></optgroup>**.

L'attribut label permet de donner un nom au groupe d'options. Le nom du groupe sera ici visible par le visiteur mais il ne pourra pas cliquer dessus.



```
<select name="couleur" id="couleur" >
  <optgroup label="Bleu">
    <option value="bleuClair">Bleu clair</option>
    <option value="bleuFoncé">Bleu foncé</option>
  </optgroup>
  <optgroup label="Rouge">
    <option value="vermillon">Vermillon</option>
    <option value="carmin">Carmin</option>
  </optgroup>
  <optgroup label="Vert">
    <option value="vertOlive">Vert Olive</option>
    <option value="vertPomme">Vert Pomme</option>
  </optgroup>
</select>
```

5. Label

Il est souvent nécessaire d'ajouter un libellé explicatif à une zone de saisie ; cela peut se faire en HTML, sous forme de texte tout simple ou encore à l'intérieur d'un élément `... `. Pour ajouter du sens à ce libellé et le lier à la zone de saisie, HTML a introduit récemment l'élément `<label>` dont l'attribut **for="xx"** relie ce label à une zone de saisie identifiée par son attribut **id**.

6. Encadré et regroupement de zones de saisie

L'élément **<fieldset>** permet de regrouper divers éléments de formulaire qui ont un sens commun dans un encadré ; cet élément n'a qu'un but de présentation à l'utilisateur et n'est pas porteur de données en lui-même. Il est judicieux de donner un libellé à ce groupe de zones à saisir grâce à l'élément **<legend>**.



Exemple

Extrait de code HTML correspondant :

<fieldset>

<legend>Personal Information**</legend>**

<u>L</u>ast Name: tabindex="1" accesskey="L" />

First **<u>N</u>**ame: tabindex="2" accesskey="N" />

<u>A</u>ddress: tabindex="3" accesskey="A" />

</fieldset>



Remarque

En complément, pour une meilleure ergonomie, l'attribut **tabindex** permet de déterminer l'ordre de passage d'une zone à l'autre par frappe de la touche de tabulation, et l'attribut **accesskey** permet de déterminer un raccourci clavier.

7. Saisie d'un nom de fichier à joindre

L'élément `<input type="file"... />` permet d'activer l'explorateur de disques du poste utilisateur pour sélectionner un fichier à joindre.

Cet élément se limite à déterminer le nom et le chemin d'accès au fichier (que l'utilisateur pourrait aussi bien saisir lui-même dans la zone de texte) mais il ne sait rien du contenu de ce fichier (il faudra du traitement supplémentaire piloté par le script serveur pour effectuer la lecture et le transfert du contenu du fichier).

- Exemple



Code HTML correspondant :

```
<p>Sélectionner le fichier à envoyer : <input type="file" name="fichier" id="fichier" size="50" /></p>
```

8. Données masquées

Le développeur peut inclure dans un formulaire des données qui ne seront pas affichées mais qui feront partie intégrante du lot d'informations du formulaire ; cela peut être utile par exemple dans un dialogue de type assistant pas à pas où, chaque page contenant une partie seulement des données, le traitement de toutes les informations sera effectué après la dernière étape.

Le développeur peut donc insérer ou propager des données de page en page grâce à l'élément `<input type="hidden" name=... id=... value=.../>`.

9 saisie assistée de données particulières

HTML5 apporte de nouveaux éléments pour simplifier et sécuriser la saisie de données particulières courantes.

9.1 Adresse email

L'élément `<input type="email" ... />` permet de saisir une adresse email vraisemblable dans sa forme (présence correcte des signes point et arobas...).

Attention : en aucun cas, cet élément n'est capable de vérifier sur le serveur de mail l'existence du compte associé !

9.2 Numéro de téléphone

L'élément `<input type="tel" ... />` permet de saisir un numéro de téléphone vraisemblable dans sa forme (chiffres, présence correcte des signes +, parenthèses...);

Attention : en aucun cas, cet élément n'est capable de vérifier l'existence du compte associé !

9.3 Choix de dates

Pour choisir une date du calendrier, rien ne vaut l'élément `<input type="date" />` ; il est complété par des variantes permettant de choisir une semaine (`<input type="week" />`), un mois (`<input type="month" />`), une heure (`<input type="time" />`) ou encore une date et une heure (`<input type="datetime" />`).

9.4 Saisie d'un nombre

L'élément `<input type="number" />` permet de saisir une donnée composée de chiffres ; selon le navigateur, des petits curseurs permettent de faire défiler les nombres jusqu'à la valeur voulue.

Quand la valeur attendue se situe dans une fourchette de valeurs prédéfinies, on peut avantageusement utiliser un curseur graphique grâce à l'élément `<input type="range" />` dont on paramètre le comportement à l'aide des attributs **min**, **max** et **step**.



Nota Bene

L'élément `<progress>` permet quant à lui d'afficher un niveau de progression en % grâce à une jauge horizontale (barre de progression) ; sa valeur (et donc la progression de la barre) devra être rafraîchie périodiquement par programmation.

L'élément `<meter>` et ses attributs **min**, **max**, **value**, **step**, **low**, **optimum**, permet d'afficher une valeur sur une jauge et de préciser graphiquement son placement dans une échelle prédéfinie.

Ces éléments, assez proches du précédent, ne permettent pas la saisie et sont encore très mal supportés par les navigateurs.

9.5 Choix d'une couleur

L'élément `<input type="color" />` permet d'afficher un sélecteur de couleur dans une palette affichée, ce qui est bien pratique en cas de personnalisation de la présentation d'un site par

l'utilisateur lui-même.

10. Soumission (envoi) d'un formulaire

10.1 Submit et Reset

La soumission d'un formulaire déclenche l'appel du module de traitement serveur associé, défini par l'attribut **action** de l'élément `<form >`, et communique toutes les données du formulaire au serveur (tout ce qui est placé entre `<form>` et `</form>`).

L'envoi d'un formulaire, et sa réinitialisation (remise à blanc des champs), s'effectuent en cliquant sur deux boutons spéciaux : les boutons **submit** et **reset**. Ces boutons se déclarent à l'aide de

la balise `<input type="submit">`, pour l'envoi du formulaire, et `<input type="reset">` pour la réinitialisation des champs du formulaire.



Exemple

Exemple des 2 boutons :



Code HTML correspondant :

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

```
<input type="reset" value="Recommencer" />
```



Nota Bene

Le libellé du bouton est déterminé par l'attribut **value**.

En général, on ne nomme pas les boutons (pas d'attribut **name**) car ils ne contiennent pas d'information utile et ils n'ont pas besoin d'être envoyés au serveur.

10.2 Les boutons input type button

Avec HTML 4, de nouveaux contrôles graphiques pour le Web sont apparus afin d'enrichir l'interface graphique avec l'utilisateur et de faciliter la programmation de l'interactivité.

Au-delà de la soumission d'un formulaire HTML au serveur Web et de sa réinitialisation, il est souvent nécessaire de dessiner des boutons déclenchant des actions locales assurées par le navigateur (comme des contrôles de saisie).

Les traitements correspondants doivent être définis à l'aide d'un véritable langage de programmation (HTML reste un langage de description de page) ; c'est l'objet du langage JavaScript qui sera étudié ultérieurement.

Depuis HTML 4, la nouvelle balise `<input type="button" ... />` permet de dessiner des boutons standards, ressemblant aux boutons submit et reset mais dont l'action sera déterminée par le développeur.



Exemple

Exemple de bouton d'action :



Code HTML correspondant :

```
<input type="button" value=" Tester " onclick="..." />
```



Ici sera défini le code javascript à exécuter quand l'utilisateur active ce bouton.

10.3 Les boutons graphiques button

Il est possible de donner l'apparence et le comportement d'un bouton à une portion graphique d'une page HTML grâce à la balise **<button ... > ... </button>**.

Tout ce qui est dessiné entre les deux balises (texte, image, tableau...) sera représenté dans un bouton cliquable.



Exemple

Exemple de bouton graphique :



De même, le code JavaScript à exécuter quand l'utilisateur active ce bouton sera défini dans la balise button.

Code HTML correspondant :

```
<button ...>
  <table width="150">
    <tr>
      <td width="50%"></td>
      <td><font face="Arial"><b><i>Clic-moi !</i></b></font></td>
    </tr>
  </table>
</button>
```

10.4 Les boutons image

Plutôt que de dessiner un bouton standard *submit* pour valider et soumettre un formulaire HTML, il est aussi possible de dessiner un bouton graphique de soumission avec la balise **<input type="image" src="..." />**.

L'attribut **src** définit le fichier image à utiliser.

L'image se comporte comme un bouton de validation (déclenchement de l'action associée au form et envoi des données saisies).

Crédits

OEUVRE COLLECTIVE DE L'AFPA

Sous le pilotage de la Direction de l'ingénierie

DATE DE MISE À JOUR

03/09/2021

© AFPA

Reproduction interdite

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconques. »