

## Le formulaire

La balise **form** encadre tous les *widgets*. L'attribut *action* indique l'URL du script chargé de traiter les données saisies et *method* indique la façon dont les données sont envoyées :

- GET : envoi des données via l'URL
- POST : envoi des données via l'entête HTTP (HTTP header)

```
<!-- Formulaire GET -->
<form action="script.php" method="get" >
  <!-- On place ici les widgets -->
</form>

<!-- Formulaire POST -->
<form action="script.php" method="post" >
  <!-- On place ici les widgets -->
</form>
```

## Widgets de saisie

### Texte simple *input[type=text]*

Zone de saisie de texte d'une seule ligne.

```
<input type="text" name="nom"
  value="Valeur par défaut" size="48" />
```

Texte :

### Mot de passe *input[type=password]*

Zone de saisie de texte sur une seule ligne dans la saisie est protégée (chaque caractère marqué par une astérisque).

```
<input type="password" name="pass"
  value="" size="16" />
```

Mot de passe :

### Aire de texte *textarea*

Zone de saisie de texte sur plusieurs lignes.

```
<textarea name="corps_de_texte" rows="8" cols="32">
  Saisissez votre texte ici...
</textarea>
```

Textarea :

Saisissez votre texte  
ici...

Auteur : Eric Bellot. Mis à jour le 2009-08-21

Document publié selon les termes de la licence Creative Commons  
CC-BY-SA : <http://creativecommons.org/licenses/by-sa/2.0/fr/>

## Widgets de sélection

### Bouton radio *input[type=radio]*

La sélection d'un bouton désélectionne les autres.

Les boutons radio devant fonctionner ensemble doivent recevoir le même attribut *name*.

```
<input type="radio" name="ouinon" value="1"
  checked="1"/>
<input type="radio" name="ouinon" value="0" />
```

Boutons radio :



### Case à cocher *input[type=checkbox]*

```
<input name="case1" type="checkbox" value="oui"
  checked="1" />
<input name="case2" type="checkbox" value="oui" />
```

Case cochée :

Case décochée :

### Choix multiple *select[multiple]*

Zone de sélection multiple (Ctrl+clic et Maj+clic).

Important : remarquez les deux crochets à la suite du nom *couleur[]*. C'est un code qui permet de renvoyer les résultats sous forme de tableau à PHP.

L'attribut *multiple* n'est doté d'une valeur que pour obtenir un code conforme au XHTML. Quelle que soit la valeur, dès que l'attribut *multiple* est présent, il est actif.

```
<select name="couleurs[]" size="4" multiple="1">
  <option value="vert" selected="1">Vert</option>
  <option value="bleu">Bleu</option>
  <option value="rouge">Rouge</option>
  <option value="jaune">Jaune</option>
</select>
```

Choix multiple :



### Menu déroulant *select*

Zone de sélection unique : menu déroulant.

Le menu déroulant s'obtient avec un code identique à celui du « choix multiple » dont on a supprimé l'attribut *multiple*.

Si l'attribut *value* n'est pas défini, c'est le texte entre les balises qui est envoyé.

```
<select name="couleurs" size="1">
  <option value="green">Vert</option>
  <option value="blue">Bleu</option>
  <option value="red">Rouge</option>
  <option value="yellow">Jaune</option>
</select>
```

Menu déroulant :



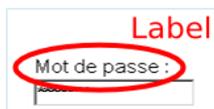
## Widgets de désignation

### Label

Étiquette d'un widget (requis pour l'accessibilité)

L'attribut `for` correspond à l'attribut `name` du widget désigné par le label.

```
<label for="name_element">Nom du widget</label>
<label>Nom du widget : <br />
  <input name="nom" type="text" size="48" />
</label>
```



### Fieldset

Permet de donner un titre et un cadre à un ensemble de widgets.

```
<fieldset>
  <legend>Jeu de widget (fieldset):</legend>
  <!-- Placer ici les widgets. -->
</fieldset>
```



## Boutons Input

### Input Submit input[type=submit]

Le bouton **Submit** permet de soumettre le formulaire, c'est à dire de déclencher l'envoi des données.

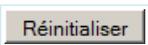
```
<input type="submit" name="validation"
value="Valider" />
```



### Input Reset input[type=reset]

Le bouton **reset** permettant de réinitialiser le formulaire. Les valeurs saisies par l'utilisateur sont effacées ou remises aux valeurs par défaut.

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



### Input Image input[type=image]

Le bouton **Image** est un bouton de soumission du formulaire comme le bouton **Submit** sauf que le contenu est une image.

```
<input type="image" name="validation"
src="disquette.png" alt="Valider" />
```



Les valeurs renvoyées par le bouton **Image** sont les coordonnées du clic. En PHP, si le nom (`name`) du bouton est `validation`, les valeurs numériques dans `$_POST` ou `$_GET` seront `'validation_x'` et `'validation_y'`.

### Input Button input[type=button]

Le bouton `button` n'a pas de comportement par défaut. En général, son comportement est défini par une action JavaScript.

```
<input type="button" value="Note"
onclick="window.open('note.html');" />
```



## Boutons poussoir (push buttons)

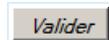
L'élément `button` est analogue à `input[submit]` ou `input[reset]`, mais il permet d'insérer un contenu complexe entre ses deux balises : texte mis en forme, image, etc.

### button submit button[type=submit]

Analogue à `input[submit]`, le `button[submit]` permet de valider un formulaire.

```
<!-- Contenu : texte balisé -->
<button type="submit" name="validation">
  <em>Valider</em>
</button>

<!-- Contenu : image -->
<button type="submit" name="enregistrer">
  
</button>
```



### button reset button[type=reset]

Le `button[reset]` est analogue au widget `input[reset]` et permet de réinitialiser un formulaire.

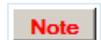
```
<!-- Contenu : aucun (image d'arrière-plan) -->
<button type="reset" name="annuler"
style="width:32px;height:32px;border:none;
background: #fff url('cancel.png')
no-repeat scroll center center;" />
```



### bouton générique button[type=button]

Aucune action n'est définie pour le bouton générique. En général, son comportement est défini par une action JavaScript.

```
<button type="button"
onclick="window.open('note.html');">
  <strong style="color:red">Note</strong>
</button>
```



## Autres Widgets

### Transfert de fichiers input[type=file]

Permet de sélectionner un nom de fichier. Le widget `cache` `MAX_FILE_SIZE` doit être placé avant le widget `file` pour indiquer la taille maximum acceptée pour les fichiers (en octets). La balise `form` doit recevoir l'attribut `enctype="multipart/form-data"` pour accepter les fichiers.

```
<input name="MAX_FILE_SIZE" type="hidden"
value="3000" />
<input name="fichier" type="file" />
```



### Hidden input[type=hidden]

Permet d'envoyer une valeur sans la faire apparaître dans le formulaire vu par l'internaute.

Attention : la valeur est visible dans le code HTML.

```
<input name="invisible" type="hidden" value="33" />
```