

Les formulaires HTML5

420-JLA-JQ Introduction au web

Notes de cours

HTML5 introduit de nombreuses nouveautés pour les formulaires pour améliorer l'aide à la saisie et les contrôles disponibles pour l'utilisateur. Plusieurs attributs simples à mettre en place améliorent la prise en charge des formulaires, ***tout en se passant de JavaScript.***

Champs obligatoires (required)

L'attribut **required** permet de rendre obligatoire le remplissage d'un champ et bloquer la validation du formulaire.

L'attribut required peut être placé sur les éléments <input> <textarea>

```
<!-- avec la valeur required -->  
<input type="text" required="required">
```

CSS:

```
:required {  
    border: 2px dotted orange;  
}
```

Champ requis



Invalide / Valide CSS

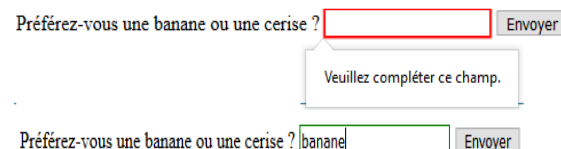
Lorsqu'un élément est invalide, deux choses ont lieu :

- L'élément possède la pseudo-classe CSS **:invalid** ; cela vous permet d'appliquer un style spécifique sur les éléments invalides. De la même façon, les éléments valides possèdent la pseudo-classe CSS **:valid**.
- Si l'utilisateur tente d'envoyer les données, le navigateur bloquera le formulaire et affichera le message d'erreur.

```
<form>  
  <label for="choix">Préférez-vous une banane ou une cerise ?</label>  
  <input id="choix" required>  
  <button>Envoyer</button>  
</form>
```

```
input:invalid {  
    border: 1px solid red; }
```

```
input:valid { border: 1px solid green; }
```



Préférez-vous une banane ou une cerise ? Envoyer

Veuillez compléter ce champ.

Préférez-vous une banane ou une cerise ? Envoyer

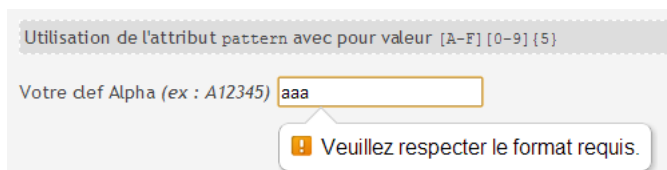
Validation (pattern)

Les patterns s'utilisent comme les expressions régulières. Avec HTML5, il ne contient pas de point d'ancrage `^` au début et `$` à la fin.

L'attribut **pattern** permet d'accepter une expression régulière.

```
<input type="text" pattern="[A-F][0-9]{5}">
```

Ce champ attend une valeur numérique d'au moins 5 chiffres précédés d'une lettre majuscule comprise entre A et F. Si le format n'est pas respecté, le navigateur en informe l'utilisateur.



Exemple :

```
<form>
  <label for="choix">Préférez-vous une banane ou une cerise ?</label>
  <input id="choix" pattern="banane|cerise">
  <button>Envoyer</button>
</form>
```

```
input:invalid {
  border: 1px solid red; }
```

Préférez-vous une banane ou une cerise ?

```
input:valid { border: 1px solid green; }
```

Préférez-vous une banane ou une cerise ?

Format (placeholder)

L'attribut **placeholder** permet de décrire brièvement un champ que voulez saisir. Donne le format que l'utilisateur doit entrer.

```
<form id="frmClient" action="">
```

```
<input id="txtTel" placeholder="888-888-8888" title="XXX-XXX-XXXX" required
pattern="[0-9]{3}-[0-9]{3}-[0-9]{4}">
```

```
<input id="btnEnvoyer" type="submit">
```

```
</form>
```

confirm

Fonction en javascript qui affiche une boîte de dialogue. Le message doit être entre apostrophe.

Exemple :

```
<input id="btnReinitialiser" type="reset" onclick="return confirm('Désirez-vous réinitialiser?')">
```

submit et reset

1. Il fonctionne uniquement si vous êtes dans un formulaire c.à.d. entre les balises <form> </form>
2. Vous devez changer le type des boutons pour submit et reset.

```
<input id="btnEnvoyer" type="submit">
```

```
<input id="btnReinitialiser" type="reset">
```

3. Vous devez ajouter dans la balise form les attributs method (get/post), action(fichier.html/fichier.php), créer la fonction événementielle onsubmit avec le confirm.

```
<form id="frmClient" action="reponse.html" method="get" onsubmit="return confirm('Désirez-vous envoyer les infos?')">
```

4. Pour le message du reset, vous avez deux endroits pour créer la fonction événementielle :
 - sur la balise input avec la fonction onclick:

```
<input id="btnReinitialiser" type="reset" onclick="return confirm('Désirez-vous réinitialiser?')">
```

- sur la balise form avec la fonction onreset :

```
<form id="frmClient" action="reponse.html" method="get" onreset="return confirm('Désirez-vous réinitialiser?')">
```