



# Cours - formulaires HTML

**Note importante** : Ce document est conçu comme un guide de référence, mais n'est pas exhaustif. La capacité à rechercher des informations complémentaires et à résoudre des problèmes de manière autonome est une compétence essentielle pour tout développeur. Si vous rencontrez des difficultés ou si certaines étapes ne fonctionnent pas comme prévu, n'hésitez pas à consulter d'autres ressources (documentation officielle, tutoriels YouTube, forums spécialisés comme Stack Overflow). Apprendre à trouver l'information manquante et à déboguer par vous-même est probablement la compétence la plus précieuse que vous développerez dans votre parcours professionnel.

W3Schools.com

W3Schools offers free online tutorials, references and exercises in all the major languages of the web. Covering popular subjects like HTML, CSS, JavaScript, Python, SQL,

[https://www.w3schools.com/html/html\\_forms.asp](https://www.w3schools.com/html/html_forms.asp)



## Introduction aux formulaires HTML

Les formulaires HTML sont des éléments essentiels pour collecter des données auprès des utilisateurs d'un site web. Ils permettent d'interagir avec les visiteurs en recueillant des informations telles que des coordonnées, des commentaires, des préférences, etc. Les formulaires sont également à la base de nombreuses fonctionnalités interactives comme les recherches, les inscriptions, les connexions et les paiements en ligne.

# Les balises HTML pour les formulaires

## Balise `<form>`

La balise `<form>` est l'élément conteneur principal pour créer un formulaire HTML. Elle définit le début et la fin du formulaire et contient tous les éléments interactifs.

```
<form action="/submit-form" method="post">
  <!-- Éléments du formulaire -->
</form>
```

Attributs principaux de la balise `<form>` :

- `action` : URL où les données du formulaire seront envoyées
- `method` : méthode HTTP utilisée pour envoyer les données (généralement "get" ou "post")
- `enctype` : spécifie comment les données du formulaire sont encodées lors de leur soumission (important pour l'envoi de fichiers)

## Balise `<label>`

La balise `<label>` définit une étiquette pour un élément de formulaire. Elle est essentielle pour :

- L'accessibilité : aide les lecteurs d'écran à identifier les champs
- L'expérience utilisateur : permet de cliquer sur l'étiquette pour donner le focus au champ associé

```
<label for="username">Nom d'utilisateur :</label>
<input type="text" id="username" name="username">
```

L'attribut `for` doit correspondre à l'attribut `id` de l'élément de formulaire associé.

## Balise `<input>`

La balise `<input>` est l'élément le plus utilisé dans les formulaires. Elle peut créer différents types de champs selon la valeur de son attribut `type`.

```
<input type="text" id="name" name="name" placeholder="Votre nom">
```

## Types d'input courants :

- `text` : champ de texte sur une ligne
- `number` : champ pour saisir des nombres
- `password` : champ masquant la saisie (affiche des astérisques ou points)
- `email` : champ pour saisir une adresse e-mail (avec validation)
- `button` : bouton cliquable
- `checkbox` : case à cocher
- `radio` : bouton radio (choix unique dans un groupe)
- `date` : sélecteur de date
- `file` : sélection de fichier
- `submit` : bouton pour soumettre le formulaire
- `reset` : bouton pour réinitialiser le formulaire

## Balise `<select>`

La balise `<select>` crée une liste déroulante permettant de sélectionner une ou plusieurs options.

```
<select id="country" name="country">
  <option value="">Choisissez un pays</option>
  <option value="fr">France</option>
  <option value="be">Belgique</option>
  <option value="ch">Suisse</option>
  <option value="ca">Canada</option>
</select>
```

- La balise `<option>` définit chaque choix disponible dans la liste
- L'attribut `value` définit la valeur qui sera envoyée au serveur

## Balise `<textarea>`

La balise `<textarea>` crée un champ de texte multiligne, idéal pour les commentaires ou descriptions.

```
<textarea id="message" name="message" rows="4" cols="50" placeholder="Votre message ici..."></textarea>
```

- `rows` : nombre de lignes visibles
- `cols` : nombre approximatif de caractères visibles par ligne

## Les attributs essentiels des éléments de formulaire

### Attribut `id`

L'attribut `id` donne un identifiant unique à l'élément. Il est utilisé pour :

- Associer une étiquette ( `<label>` ) à un champ
- Cibler l'élément avec JavaScript ou CSS
- Créer des liens d'ancrage dans la page

```
<input type="text" id="username" name="username">
```

### Attribut `name`

L'attribut `name` est essentiel pour l'envoi des données. Il définit le nom de la variable qui sera envoyée au serveur.

```
<input type="text" name="email" id="email">
```

Sans cet attribut, les données du champ ne seront pas transmises lors de la soumission du formulaire.

### Attribut `required`

L'attribut `required` indique que le champ doit obligatoirement être rempli avant de pouvoir soumettre le formulaire.

```
<input type="text" id="name" name="name" required>
```

C'est une validation côté client qui empêche la soumission si le champ est vide.

## Attribut `checked`

L'attribut `checked` s'applique aux éléments `<input>` de type `checkbox` ou `radio`. Il définit l'état coché par défaut.

```
<input type="checkbox" id="newsletter" name="newsletter" checked>
<label for="newsletter">S'abonner à la newsletter</label>
```

## Attributs `min` et `max`

Ces attributs définissent les valeurs minimales et maximales pour les types numériques (`number`, `date`, `range`, etc.).

```
<input type="number" id="age" name="age" min="18" max="120">
```

## Autres attributs utiles

- `placeholder` : texte d'exemple affiché dans le champ vide
- `value` : valeur par défaut du champ
- `disabled` : désactive le champ (non modifiable et non envoyé)
- `readonly` : rend le champ non modifiable mais toujours envoyé
- `pattern` : expression régulière pour valider la saisie
- `autocomplete` : active/désactive l'autocomplétion du navigateur

## Exemple de formulaire complet

```
<form action="/submit-form" method="post">
  <div class="form-group">
    <label for="full-name">Nom complet:</label>
    <input type="text" id="full-name" name="full_name" required>
  </div>

  <div class="form-group">
    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required>
  </div>
```

```

<div class="form-group">
  <label for="age">Âge:</label>
  <input type="number" id="age" name="age" min="18" max="120">
</div>

<div class="form-group">
  <label>Genre:</label>
  <label><input type="radio" name="gender" value="male"> Homme</label>
  <label><input type="radio" name="gender" value="female"> Femme</label>
  <label><input type="radio" name="gender" value="other"> Autre</label>
</div>

<div class="form-group">
  <label for="country">Pays:</label>
  <select id="country" name="country">
    <option value="">Sélectionnez votre pays</option>
    <option value="fr">France</option>
    <option value="be">Belgique</option>
    <option value="ch">Suisse</option>
    <option value="ca">Canada</option>
  </select>
</div>

<div class="form-group">
  <label for="message">Message:</label>
  <textarea id="message" name="message" rows="4" cols="50"></textarea>
</div>

<div class="form-group">
  <label>
    <input type="checkbox" name="terms" required>
    J'accepte les conditions d'utilisation
  </label>
</div>

```

```
<div class="form-group">
  <label>
    <input type="checkbox" name="newsletter" checked>
    Je souhaite recevoir la newsletter
  </label>
</div>

<div class="form-group">
  <button type="submit">Envoyer</button>
  <button type="reset">Réinitialiser</button>
</div>
</form>
```

## Bonnes pratiques

1. **Accessibilité** : Toujours associer des étiquettes ( `<label>` ) aux champs de formulaire
2. **Organisation** : Structurer les formulaires en sections logiques avec des groupes ( `<fieldset>` et `<legend>` )
3. **Validation** : Utiliser les attributs de validation HTML5 (required, pattern, min, max) pour une première validation côté client
4. **Feedback** : Fournir un retour visuel sur les erreurs et les champs valides
5. **Sécurité** : Toujours valider à nouveau les données côté serveur (ne jamais faire confiance uniquement à la validation côté client)
6. **Mobile-friendly** : S'assurer que le formulaire est utilisable sur les appareils mobiles
7. **Labels clairs** : Utiliser des labels descriptifs pour guider l'utilisateur

**Protection intellectuelle** : Ce cours, est une création originale de **MANJAL Younes** et est protégé par le droit d'auteur conformément au Code de la propriété intellectuelle français (article L.111-1). Toute reproduction, modification, diffusion ou utilisation, même partielle, sans autorisation

écrite préalable est strictement interdite. L'intelligence artificielle a été utilisée uniquement comme outil d'assistance pour la correction orthographique et la reformulation de certaines phrases, sous ma supervision et validation. Le contenu pédagogique et intellectuel reste entièrement ma création personnelle.