

HTML & CSS – TP Formulaires

Pour créer des formulaires il faut :

- Le formulaire en HTML (zones de saisie, listes...)
- Une mise en forme CSS
- Un langage de développement côté serveur (PHP) qui récupère les données et gère les connexions avec le serveur web.

Exemple de formulaire

The diagram illustrates a movie form with various input fields and controls. Labels on the left are connected to the form elements by orange arrows:

- Champs de saisie sur une ligne** points to the **Titre** text input field.
- Boutons radios (choix unique)** points to the **Genre** section, which includes radio buttons for **Comédie**, **Drame**, **Historique**, and **Thriller**.
- Cases à cocher (choix multiples)** points to the **Disponible en :** section, which includes checkboxes for **Français**, **Japonais**, **Allemand**, and **Anglais**.
- Liste déroulante** points to the **Faites un choix** dropdown menu, currently showing **Choix 1**.
- Champ dépôt de fichier** points to the **Affiche du film** section, which includes a **Choisir un fichier** button and the text **Aucun fichier choisi**.
- Zone de saisie sur plusieurs lignes** points to the **Résumé du film** text area.
- Bouton de soumission du form** points to the **Valider** button at the bottom.

Première balise à connaître : la balise `<form>` `</form>`

Cette balise **indique le début et la fin du formulaire**, elle regroupe donc tous les champs du formulaire.

2 attributs essentiels :

- **Action** : url de la page qui traite les informations saisies dans le formulaire.

- **Method** : mode de transmission des paramètres au programme, avec deux valeurs possibles, get ou post.
 1. Get : envoie les données au serveur par l'intermédiaire de l'URL, dans l'en-tête de la requête http (les valeurs envoyées sont visibles par l'internaute et limitées en caractères).
Exemple : `http://localhost/créer_film.php?titre=Nikita&annee=1995...`
 2. Post : envoie les données au serveur dans le corps de la requête http et non dans l'URL. les valeurs envoyées sont invisibles et le nombre de caractères plus important.

Exemple de code :

```
<form action="envoi-form.php" method="post"> ... </form>
```

Ces détails seront approfondis en cours de PHP. Ici nous nous intéresserons essentiellement à l'interface HTML et non à l'interaction avec un langage côté serveur.

L'élément incontournable des formulaires : la balise `<input>`

C'est la balise la plus utilisée. Elle permet de demander à l'utilisateur de la page des informations textuelles (par exemple son prénom), un choix entre plusieurs options ou même de sélectionner un fichier à envoyer.

Chaque champ `<input>` a un attribut **name** et **type**.

Name : cette valeur n'est pas visible par l'internaute. Ce nom sera utile pour identifier les champs avec le langage php.

Type : définit la nature du champ `<input>`.

Exemple `<input type="text" name="contact_nom">` ce champ est de type text
--> c'est une zone de saisie.

Types fréquemment utilisés :

`Type="text"` Champ de saisie.

`Type="password"` Le texte saisi au clavier n'apparaît pas en clair.

`Type="checkbox"` Les cases à cocher, choix multiples.

`Type="radio"` Les boutons radios, choix unique.

`Type="submit"` Bouton de validation/d'envoi.

`Type="email"` Ce champ attend au minimum un caractère suivi d'un @ suivi à son tour d'un caractère.

`Type="tel"` Sur les supports tactiles le clavier proposé sera numérique.

`Type="url"` Tous les types d'URL sont admis (ftp://, mailto:, http://, etc.).

On peut ajouter un certain nombre d'attributs à la balise `<input>` pour préciser son comportement:

size : taille visuelle de la zone.

maxlength : limite le nombre de caractères saisis.

value : pour pré-remplir le champ avec une valeur par défaut.

placeholder : texte qui s'affiche dans la zone de saisie, il disparaît dès que le visiteur clique à l'intérieur du champ.

required : pour déclarer un champ obligatoire. Tant que ce champ n'est pas rempli le formulaire ne peut pas être envoyé.

Approfondir la balise input <https://developer.mozilla.org/fr/docs/Web/HTML/Element/input>

La balise <select>

Balise pour créer une liste déroulante. <select> est un conteneur dans lequel on doit énumérer, avec les balises <option>, tous les choix possibles.

```
<select name="realisateurs">
  <option value="hitchcock">Alfred Hitchcock</option>
  <option value="tarantino">Quentin Tarantino</option>
  <option value="besson">Luc Besson</option>
  <option value="miller">Henri Miller</option>
</select>
```

Approfondir la balise select <https://developer.mozilla.org/fr/docs/Web/HTML/Element/select>

Approfondir la balise option <https://developer.mozilla.org/fr/docs/Web/HTML/Element/option>

La balise <textarea>

Pour saisir un texte libre sur plusieurs lignes.

```
<textarea name="nom-zone" rows="10" cols="50">Texte facultatif</textarea>
```

Cols et *Rows* indiquent le nombre de colonnes et de lignes de la zone de saisie. Cette zone peut également être dimensionnée directement en CSS.

Pour approfondir *textarea* <https://developer.mozilla.org/fr/docs/Web/HTML/Element/textarea>

La balise <label>

"Son utilisation permet aux navigateurs d'associer cet intitulé au champ de formulaire. Ainsi, un clic sur l'intitulé donnera le focus au champ de formulaire, ce qui est très pratique par exemple pour les boutons radio ou cases à cocher parfois trop petits pour certains utilisateurs. Ce mécanisme est important également pour les lecteurs d'écran, qui ont besoin d'étiquettes explicites pour associer chaque champ de formulaire à un texte." Source Alsacréations

```
<label for="contact_nom">Nom</label>
<input type="text" id="contact_nom">
```

À lire pour comprendre l'usage de la balise label :

<http://www.alsacreations.com/astuce/lire/6-utiliser-element-label-input-formulaires.html>

<https://www.accede-web.com/notices/html-css-javascript/7-formulaires/7-1-label-for-id-champs-intitule/>

CSS : Les pseudo-classes de formulaire

Pour cibler les éléments de formulaire en fonction de leur état, on utilise les pseudo-classes :

:focus pour styler un élément ciblé par l'utilisateur (souris, clavier, tactile)

:valid

:invalid

:required cible un élément ayant l'attribut html required

:checked cible un élément coché comme un bouton-radio ou une checkbox

Les fondamentaux à connaître sur les formulaires sont ici

- <https://learn-the-web.algonquindesign.ca/topics/forms/>
- <https://openclassrooms.com/courses/apprenez-a-creeer-votre-site-web-avec-html5-et-css3/les-formulaires-8>

ATTN : Openclassrooms utilise l'ancienne syntaxe HTML5. Ne mettez pas de / dans les balises orphelines.

`<input type="text" />` NON

`<input type="text">` OUI

`
` NON

`
` OUI