

Chapitre 16 - Html - Formulaires

Les formulaires HTML permettent d'instaurer un dialogue entre un utilisateur et un site web ou une application. Ils permettent à l'utilisateur d'envoyer des données au site web. La plupart du temps, ces données sont envoyées à des serveurs web mais la page peut aussi les intercepter et les utiliser elle-même.

Un formulaire HTML est composé d'un ou plusieurs widgets. Ceux-ci peuvent être des zones de texte (sur une seule ligne ou plusieurs lignes), des boîtes à sélection, des boutons, des cases à cocher ou des boutons radio. La plupart du temps, ces items sont associés à un libellé qui décrit leur rôle. On voit tout cela dans ce chapitre.

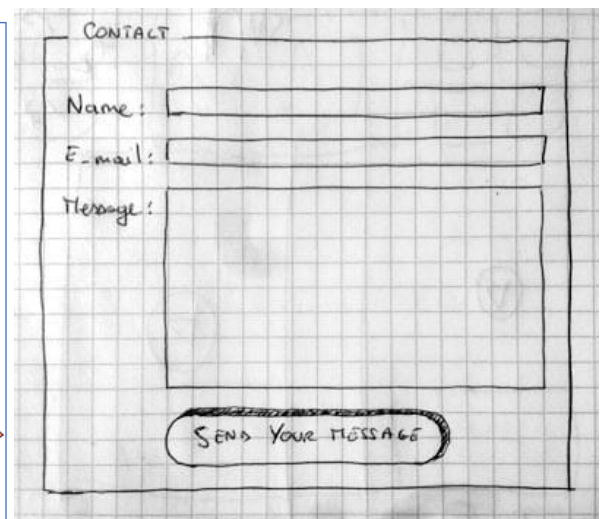
1- EXEMPLE D'UN FORMULAIRE BASIQUE DE CONTACT :

Code Html :

```
<body>
<form action=" " method=" " >
  <fieldset>
    <legend>CONTACT</legend>
    <div>
      <label for="n">Name:</label>
      <input type="text" id="n" name="nomDeFamille">

      <label for="e">Email:</label>
      <input type="email" id="e" name="adresseMail">

      <label for="m">Message:</label>
      <textarea type="text" id="m" name="texteMessage"></textarea>
    </div>
    <div class="bouton">
      <button type="submit" name="boutonEnvoyer">Send your message</button>
    </div>
  </fieldset>
</form>
```



Code Css

```
fieldset{
  max-width: 600px;
  margin:auto;
  border-radius: 5px;
}

form fieldset :nth-child(2){
  /*border: 1px solid red;*/      display: grid;
  grid-template-columns: 1fr 5fr;
}

.bouton{
  text-align: center;
}

button {
  padding:5px;
  border-radius: 10px;
  box-shadow: 5px -5px #AAA;
  margin:10px 0 2px 0;
}
```

Point Cours :

- Un formulaire est composé de widgets qui sont placés dans un élément html `<form>` :

```
<form action="/trait" method="post">...</form>
```

- Les widgets qui permettent à l'utilisateur de saisir une donnée sont généralement des éléments `<input>`. Ce sont des balises orphelines. Ils sont de type *inline* :

```
<input type="text" id="n" name="nomDeFamille">
```

- L'attribut '*type*' de `<input>` définit la nature de la donnée saisie. L'apparence du widget dépend de la valeur de cet attribut. Cet attribut *type* peut prendre les valeurs : *text, email, tel, url, number, range, date, password, ...*
- L'attribut '*name*' de `<input>` définit le nom de la variable envoyée au serveur, qui contiendra la donnée saisie.
- Un texte descriptif accompagne généralement ces éléments `<input>`. Ils sont placés dans un élément html `<label>`. Pour lier ces 2 éléments, l'attribut *for* de `<label>` doit être le même que l'attribut *id* de `<input>`.

```
<label for="n">Name:</label>  
<input type="text" id="n" name="nomDeFamille">
```

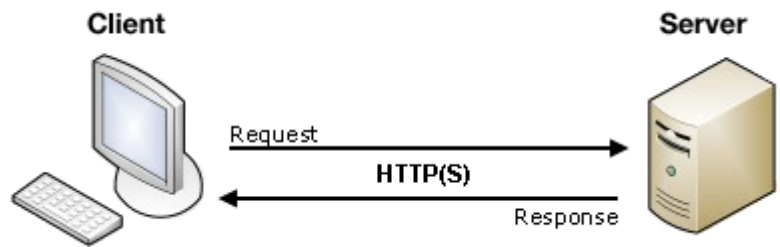
- Le widget qui permet à l'utilisateur de saisir un texte de grande taille est un élément html `<textarea>`. Il est de type *inline* et est constitué d'une double balise.
- Pour le widget bouton, il existe 2 éléments html appropriés :

- ```
<button type="submit" name="boutonEnvoyer"> message</button>
```

- ```
<input type="submit" name="boutonEnvoyer" value="Send  
your message">
```

2- ENVOI DES DONNEES VERS LE SERVEUR :

Le web se fonde sur une architecture client/serveur. Le client (généralement un navigateur Web) envoie une requête à un serveur (le plus souvent un serveur web), en utilisant le protocole HTTPS. Le serveur répond à la requête en utilisant le même protocole.



Le formulaire Html n'est rien d'autre qu'un moyen commode et convivial de configurer une requête HTTPS pour envoyer des données à un serveur.

Dans le protocole HTTPS, pour atteindre une ressource sur Internet, le navigateur envoie une requête à une URL. Ces requêtes sont composées de deux parties : un en-tête (header) qui contient les métadonnées sur les capacités du navigateur, et un corps (body) qui contient les informations nécessaires au serveur pour effectuer la requête.

Dans la balise ouvrante de l'élément `<form action=" " method=" ">`, 2 attributs sont importants :

- L'attribut `action=" "` indique au serveur ce qu'il faut faire après l'envoi du formulaire. Par exemple, si on écrit `action="/tchat"`, lorsque la requête envoyée par le navigateur arrive au serveur, celui-ci reconnaît le mot clé « *tchat* ». Le serveur exécute alors un algorithme qui a été codé par le webmaster du site.
- L'attribut `method=" "` indique au navigateur quel protocole est utilisé pour véhiculer les données vers le serveur. Les principales méthodes utilisées sont GET et POST.

Point Cours :

- La méthode **GET** est utilisée par le navigateur pour demander au serveur de renvoyer une certaine ressource. "Hé le serveur, je veux cette ressource." Dans ce cas, le navigateur envoie un corps vide.
Si un formulaire est envoyé avec cette méthode, les données envoyées au serveur sont ajoutées à **l'URL** de la requête HTTPS.
- Avec la méthode **POST**, le navigateur demande au serveur une réponse prenant en compte les données contenues dans le corps de la requête HTTPS : « Hé serveur ! vois ces données et renvoie-moi le résultat approprié ».
Si un formulaire est envoyé avec cette méthode, les données sont ajoutées au **au corps** de la requête HTTPS.

GET	POST
ASCII URL limitée à 2000 caractères/JavaScript	Pas de taille limite/ Traitement par PHP

3- AUTRES WIDGETS DISPONIBLES POUR LES FORMULAIRES :

a. BOUTONS RADIOS :

Les éléments `<input>` dont l'attribut `type` vaut `radio` sont généralement utilisés pour construire des groupes d'options parmi lesquelles on ne peut choisir **qu'une valeur**. Ces « boutons radio » sont représentés par des cercles remplis lorsqu'ils sont sélectionnés.

```
<form action="" method="">
  <div>
    <label for="a">Choix 1</label>
    <input type="radio" id="a" name="choix" value="choix1" checked />

    <label for="b">Choix 2</label>
    <input type="radio" id="b" name="choix" value="choix2" />

    <label for="c">Choix 3</label>
    <input type="radio" id="c" name="choix" value="choix3" />
  </div>
  <button type="submit" name="boutonEnvoyer" value="boutonFormulaire">Envoyer</button>
</form>
```

Le code ci-contre donnera sur le navigateur :

Choix 1 ☒ Choix 2 ☐ Choix 3 ☐

Envoyer

```
div{
  display: grid;
  grid-template-columns: 1fr 1fr;
  width: max-content;
  border: 1px solid black;
  padding: 10px;
}
```

Avec le Css ci-contre, l'affichage devient :

Choix 1 ☒

Choix 2 ☐

Choix 3 ☐

Envoyer

En cliquant sur le bouton *Envoyer*, aucune méthode n'étant spécifiée dans la balise `<form>`, c'est la méthode GET qui est choisie par défaut. On retrouve ainsi en fin d'URL, les variables nommées *choix* et *boutonEnvoyer* qui prennent les valeurs suivantes :

?choix=choix1&boutonEnvoyer=boutonFormulaire

b. BOUTONS CHECKBOXS :

Les éléments `<input>` dont l'attribut `type` vaut `checkbox` ressemblent aux éléments radios vus dans le paragraphe précédent. Les différences sont les suivantes :

- on peut choisir une ou **plusieurs valeurs**.
- ils sont représentés par des cases carrées remplis lorsqu'ils sont sélectionnés.

```
<form action="" method="">
  <div>
    <label for="a">Choix 1</label>
    <input type="checkbox" id="a" name="choix" value="choix1" checked />

    <label for="b">Choix 2</label>
    <input type="checkbox" id="b" name="choix" value="choix2" />

    <label for="c">Choix 3</label>
    <input type="checkbox" id="c" name="choix" value="choix3" />
  </div>
  <button type="submit" name="boutonEnvoyer" value="boutonFormulaire">Envoyer</button>
</form>
```

Le code ci-contre donnera sur le navigateur :

Choix 1 ☒

Choix 2 ☒

Choix 3 ☐

Envoyer

avec le Css précédent.

Après envoi du formulaire, les variables nommées *choix* et *boutonEnvoyer* prennent les valeurs suivantes :

C. LISTES DEROULANTES .

L'élément `<select>` permet d'obtenir une liste d'options que l'utilisateur pourra choisir.

Le code ci-contre :

```
<form action=" " method=" ">
  <select name="menu" size="1" required">
    <option value="" selected>Choisir une ou plusieurs options</option>
    <option value="choix1">choix 1</option>
    <option value="choix2">choix 2</option>
    <option value="choix3">choix 2</option>
  </select>

  <button type="submit" name="boutonEnvoyer" value="boutonFormulaire">Envoyer</button>
</form>
```

donnera sur le navigateur :

Choisir une ou plusieurs options ▼ Envoyer

ou :

Choisir une ou plusieurs options ▼ Envoyer

- Choisir une ou plusieurs options
- choix 1
- choix 2
- choix 2

Après envoi du formulaire, les variables nommées *choix* et *boutonEnvoyer* prennent les valeurs suivantes :

?menu=&boutonEnvoyer=boutonFormulaire

En rajoutant l'attribut **multiple** dans la balise ouvrante `<select>`, l'utilisateur pourra sélectionner une ou plusieurs valeurs.

d. WIDGET INPUT DE TYPE RANGE :

Les éléments `<input>` dont l'attribut *type* vaut *range* permettent à l'utilisateur d'indiquer une valeur numérique comprise entre deux bornes. La valeur précise n'est pas considérée comme importante.

```
<form action=" " method=" ">
  <div>
    <label for="v">Volume</label>
    <input type="range" id="v" name="volume" min="0" max="11" />

    <label for="b">Niveau des basses</label>
    <input type="range" id="b" name="basse" min="0" max="100" value="90" step="10" />
  </div>
  <button type="submit" name="boutonEnvoyer" value="boutonFormulaire">Envoyer</button>
</form>
```

Le code ci-contre donnera sur le navigateur :

Volume  Niveau des basses 

Ou l'affichage ci-contre avec le css donné dans le paragraphe a) (boutons radios).

Volume 
 Niveau des basses 

Après envoi du formulaire, les variables nommées *choix* et

?volume=6&basse=90&boutonEnvoyer=boutonFormulaire

boutonEnvoyer prennent les valeurs suivantes :

e. WIDGET INPUT DE TYPE DATE :

Les éléments `<input>` dont l'attribut *type* vaut *date* permettent de saisir des dates.

La valeur fournie contiendra l'année, le mois et le jour mais pas l'heure et les minutes. Les éléments `<input type="time">` et `<input type="datetime-local">` permettent respectivement de saisir des heures et des combinaisons de dates et heures.

Le code html ci-contre :

```
<form action=" " method=" ">
  <div>
    <label for="d">Start date:</label>
    <input type="date" id="d" name="datenaissance" value="2024-05-02" min="1900-01-01" max="2050-01-01" />
  </div>
  <button type="submit" name="boutonEnvoyer" value="boutonFormulaire">Envoyer</button>
</form>
```

donnera sur le navigateur

Start date: 

ou en ouvrant le menu, on aura :

Start date: 

< mai 2005 >

lun.	mar.	mer.	jeu.	ven.	sam.	dim.
25	26	27	28	29	30	1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31	1	2	3	4	5

Effacer

Si la date du 15 mai 2005 est sélectionnée, après envoi du formulaire, les variables suivantes sont transmises dans l'URL :

```
?datenaissance=2005-05-15&boutonEnvoyer=boutonFormulaire
```

4- EXERCICE :

Ecrire les codes html et css qui permettent d'obtenir le formulaire ci-dessous :

Formulaire de paiement

Les champs obligatoires sont suivis par un *.

Informations de contact

Qualité

- ☐ Monsieur
- ☐ Madame

Nom : *

e-mail : *

Mot de passe : *

Informations de paiement

Type de carte :

Numéro de carte : *

Date d'expiration : *