

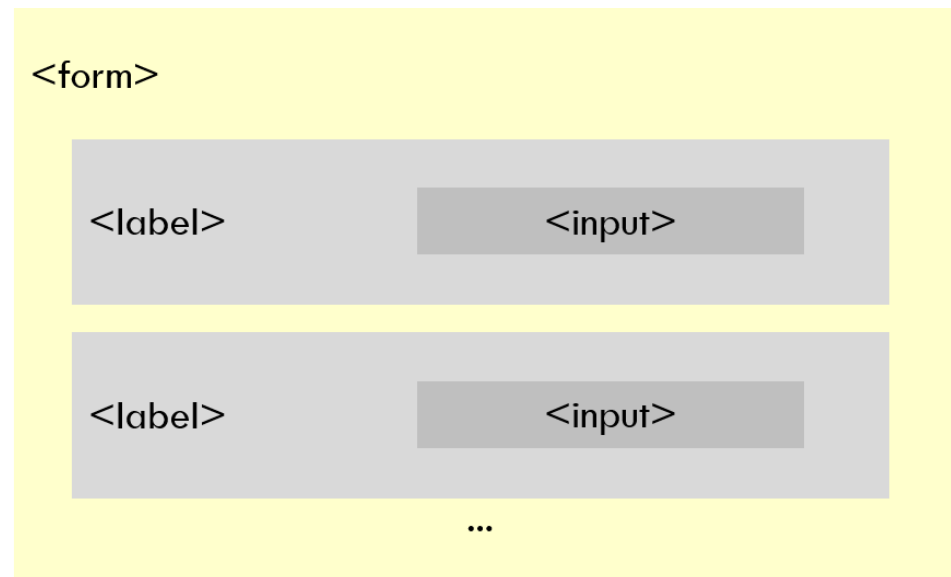
HTML

Formulaires

Boris.Fritscher@he-arc.ch

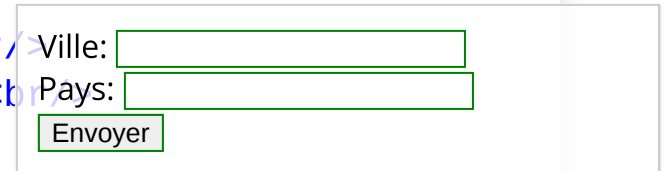
HTML < >

< > sert à préparer un ensemble de données de l'utilisateur, et les envoyer au serveur, chaque donnée est saisie dans une paire d'éléments < > et < > exemple simple:



HTML < >

```
<form action="https://httpbin.org/post" method="post">  
  <label>Ville: <input name="city_name"/></label> <br/>  
  <label>Pays: <input name="country_name"/></label> <br/>  
  <input type="submit" value="Envoyer"/>  
</form>
```



Ville:

Pays:

L'attribut **action** définit l'emplacement (une URL) où doivent être envoyées les données collectées par le formulaire.

L'attribut **method** définit la méthode HTTP utilisée pour envoyer les données (cela peut être « GET » ou « POST »).

https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Mon_premier_formulaire_HTML

Composition d'un formulaire HTML: la palette des éléments

Champs de saisie

Boîtes à cocher

<input type="checkbox"/> Voiture	<input type="radio"/> Email
<input checked="" type="checkbox"/> Train	<input type="radio"/> Téléphone
<input type="checkbox"/> Vélo	<input type="radio"/> Courrier

Champ multi-lignes

Liste déroulante de valeurs

Chat ▼

Chien
Chat
Hamster

Boutons à cliquer

Envoyer le formulaire 

https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Les_blocs_de_formulaires_natifs

HTML Form

Tous les éléments peuvent être désactivé avec l'attribut

```
<input type="text" value="Alice" disabled> <br/>  
<input type="button" value="Sauver" disabled>
```



Alice

Sauver

Élément < = " ">

Champs de saisie avec du texte sur une seule ligne.

Attribut	Description
name	le no <div></div>

Elément < = " ">

Saisir un mot de passe sans que celui-ci ne soit lisible à l'écran

```
<label>
  Mot de passe: <input type="password"
                    name="password"
                    value="12345678" />
</label>
```

Mot de passe:

<https://developer.mozilla.org/fr/docs/Web/HTML/Element/input/password>

Élément <input type="checkbox" value="" />

pour saisir un choix de valeurs parmi plusieurs valeurs

Attribut	Description
----------	-------------

name	le nom de la variable envoyée au serveur.
value	la valeur envoyée au serveur si la case est cochée.
checked	si présent, la case à cocher sera cochée.

```
<label>
  <input type="checkbox" name="v" value="voiture"/>
  Voiture Individuelle
</label> <br/>
<label>
  <input type="checkbox" name="v" value="train" checked="checked"/>
  Train ou tram
</label> <br/>
<label>
  <input type="checkbox" name="v" value="velo"/>
  Vélo ou trottinnette
</label>
```

- ☐ Voiture Individuelle
- ☒ Train ou tram
- ☐ Vélo ou trottinnette

Élément `<` `=` `"` `"` `>`

pour saisir un choix **unique** parmi plusieurs valeurs

Attribut	Description
name	le nom de la variable envoyée au serveur.
value	la valeur envoyée au serveur si la case est cochée.
checked	si présent, la case à cocher sera cochée.

Important même pour connecter les inputs

```
<label>
  <input type="radio" name="contact" value="email"/>
  Email
</label> <br/>
<label>
  <input type="radio" name="contact" value="telephone"/> Téléphone
</label> <br/>
<label>
  <input type="radio" name="contact" value="courrier" checked="checked"/>
  Courrier
</label>
```

- ☐ Email
- ☐ Téléphone
- ☒ Courrier

Élément < >

Espace de saisie de plusieurs lignes de texte

Attribut	Description
name	le nom de la variable envoyée au serveur.
cols	la largeur visible du contrôle de saisie.
rows	le nombre de lignes de texte visibles pour le contrôle.

```
<label>
  Commentaire:<br/>
  <textarea cols="30" rows="4" name="comment"></textarea>
</label>
<label>
  Recette:<br/>
  <textarea cols="20" rows="3" name="recette">1. Préparer
2.
3.</textarea>
</label>
```

Commentaire:

Recette:

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

Élément <select>

Liste déroulante de valeurs à choisir

Attribut	Description
name	le nom de la variable envoyée au serveur.
value	la valeur envoyée au serveur si l'option est sélectionnée.
selected	l'option sera sélectionnée

```
<label>Choisir un animal :  
  <select name="pet">  
    <option value="dog">Chien</option>  
    <option value="cat" selected>Chat</option>  
    <option value="hamster">Hamster</option>  
  </select>  
</label>
```

Choisir un animal : Chat ▼

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

Élément <select>

Liste déroulante de valeurs à choisir

Attribut	Description
----------	-------------

multiple	si présent, la sélection multiple est possible.
----------	---

size	Nombres de lignes affiché en mode liste.
------	--

```
<label>Veuillez choisir un ou plusieurs animaux: <br/>
  <select name="pets" multiple size="10">
    <optgroup label="Animaux à 4-jambes">
      <option value="Chien">Chien</option>
      <option value="chat">Chat</option>
      <option value="hamster" disabled>Hamster</option>
    </optgroup>
    <optgroup label="Animaux volants">
      <option value="perroquet">Perroquet</option>
      <option value="macaw">Macaw</option>
      <option value="albatros">Albatros</option>
    </optgroup>
  </select>
</label>
```

Veuillez choisir un ou plusieurs animaux:

Animaux à 4-jambes

Chien
Chat
Hamster

Animaux volants

Perroquet
Macaw
Albatros

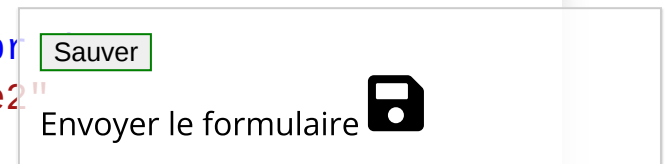
Élément < = " ">

bouton qui provoque l'envoi des données du formulaire

Attribut	Description
value	texte affiché sur le bouton.

on peut aussi utiliser une image/icône comme bouton: un même formulaire peut avoir plusieurs "submit buttons", chacun portant un nom différent.

```
<form action="https://httpbin.org/post">
  <input type="submit" name="save1" value="Sauver"/> <br>
  Envoyer le formulaire <input type="image" name="save2"
    src="images/save-solid.svg" alt="Sauver" />
</form>
```



<https://developer.mozilla.org/fr/docs/Web/HTML/Element/input/submit>

Élément < >

bouton qui provoque l'envoi des données du formulaire

Attribut	Description
type	submit, reset ou button

```
<form action="https://httpbin.org/post">
  <label>Nom: <input type="text" name="nom"/><br/>


  <button type="button">ne fait rien</button><br/>

  <button type="submit">Envoyer
    
  </button>

  <button type="reset">Effacer</button>
</form>
```

Nom:

ne fait rien

 Envoyer

Effacer

<https://developer.mozilla.org/fr/docs/Web/HTML/Element/button>

Élément <

== "

" >

Transmettre avec les données du formulaires des données supplémentaires non affichées sur la page.

Attribut	Description
name	le nom de la variable envoyée au serveur.
value	la valeur envoyée au serveur si la case est cochée.

```
<input type="hidden" name="session_id" value="ab324e23"/>
```

Méthode très utilisée pour reprendre des données saisies ou connues précédemment pour le prochain formulaire, dans le cas d'une suite de formulaires pour un même traitement.

Exemple: Achat sur un Web Shop, le numéro de la commande n'est pas saisi par l'utilisateur, mais il est fabriqué par le serveur et repassé en "hidden" de formulaires en formulaires jusqu'au formulaire final de paiement.

Exemple de formulaire complet

```
<h3>Devenez membre de notre association:</h3>
<form method="get"
      action="https://httpbin.org/post">
  <p>
    <label>Nom: <input type="text" name="nom"/></label>
  </p>
  <p>
    <label>Prénom: <input type="text" name="prenom"/></label>
  </p>
  <p>
    <label>e-Mail: <input type="text" name="eMail"/></label>
  </p>
  <p>
    <label>Pays:
      <select name="pays">
        <option value="suisse"
                  selected="selected">Suisse</option>
        <option value="allemagne">Allemagne</option>
        <option value="france">France</option>
        <option value="etatsunis">&Eacute;tats-Unis</option>
      </select>
    </p>
```

Devenez membre de notre association:

Nom:

Prénom:

e-Mail:

Pays:

Date de naissance:

- ☒ Membre Actif
☐ Membre Soutien
☐ Section Fourchette

Regroupements d'éléments < >

Structurer des éléments d'un formulaire en sous-groupes

```
<fieldset>
  <legend>titre pour le groupe</legend>
  <label><input name="..." /></label>
  ... champs ou boutons du formulaire ...
</fieldset>
```

https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Comment_structurer_un_formulaire_HTML

Exemple de formulaire complet - fieldsets

```
<h3>Devenez membre de notre association:</h3>
<form method="post"
      action="http://stepbin.org/post"
      enctype="multipart/form-data">
  <fieldset>
    <legend>Données personnelles</legend>
  <p>
    <label>Nom: <input type="text">
```

HTTP GET ou POST

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. Du coup, si un formulaire est envoyé avec cette méthode, les données envoyées au serveur sont ajoutées à l'URL.

La méthode POST est un peu différente. C'est la méthode que le navigateur utilise pour demander au serveur une réponse prenant en compte les données contenues dans le corps de la requête HTTP : « Hé serveur ! vois ces données et renvoie-moi le résultat approprié ». Si un formulaire est envoyé avec cette méthode, les données sont ajoutées au corps de la requête HTTP.

https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Envoyer_et_extraire_les_donn%C3%A9es_des_formulaires

HTTP 0

```
POST / HTTP/1.1
Host: foo.com
Content-Type: application/x-www-form-urlencoded
Content-Length: 13

say=Hi&to=Mom
```

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

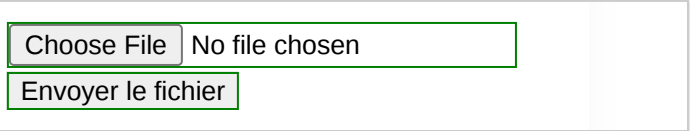
Avantage de la méthode GET (avec les données dans l'URL): On peut garder un bookmark de l'URL complet (avec données), on peut revenir en arrière sur la même page ("Back"), on peut utiliser cet URL dans un lien Hypertexte (< = " " >) Remarque: le browser doit encoder certains caractères spéciaux dans l'URL de la requête quand ils font partie des valeurs passées (espaces, accents, etc.)

Avantage de la méthode POST (avec les données transmises séparément de l'URL): Si on revient sur le même URL en navigant, on ne risque pas d'insérer une 2e fois les mêmes données dans la base, sauf si on repasse par le formulaire de départ. Remarque: pour transmettre certaines données (p. ex. fichiers), on est obligé de faire avec la méthode POST.

Envoi d'un fichier au serveur

< = " ">

```
<form action="https://httpbin.org/post"
      method="post"
      enctype="multipart/form-data">
  <input type="file" name="my_file"/>
  <input type="submit" value="Envoyer le fichier"/>
</form>
```



Choose File No file chosen
Envoyer le fichier

L'attribut `enctype="multipart/form-data"` est obligatoire dès qu'il y a au moins un fichier dans le formulaire, et la méthode est forcément **post**.

Ne jamais faire
confiance aux
données reçus
des utilisateurs!

Injection contenu tier

Les attaques Cross-Site Scripting (XSS) et Cross-Site Request Forgery (CSRF) sont des attaques fréquentes qui surviennent quand vous affichez des données renvoyées par un utilisateur à celui-ci ou à un autre utilisateur.

Pour éviter ces attaques, vous devez toujours vérifier les données qu'un utilisateur envoie à votre serveur et (si vous avez besoin de les afficher) essayez de ne pas afficher le contenu HTML tel que fourni par l'utilisateur. À la place, vous devez traiter les données fournies par l'utilisateur afin de ne pas les afficher verbatim.

Injection SQL

Cela consiste d'ordinaire à envoyer une requête SQL en espérant que le serveur l'exécutera

SQL Injection.

User-Id:

Password:

`select * from Users where user_id= 'srinivas '
and password = 'mypassword '`

User-Id:

Password:

`select * from Users where user_id= '' OR 1 = 1; /* '
and password = '*/-- '`

Déni de service

Limitez le volume des données entrantes pour n'autoriser que ce qui est nécessaire.

Séparation possible entre labels et contrôles

Dans certains cas, on ne peut pas mettre le contrôle `<input type="text" name="Addr" />` dans l'élément `<div>` correspondant, à cause des règles d'imbrications des balises. On utilise l'attribut **for** pour connecter à un **id**.

```
<div>
  <label for="address">Adresse:</label>
</div>
<div>
  <input id="address" type="text" name="Addr" />
</div>
```

Exercices

HTML Formulaire: Ski Club
Bootstrap

Validation Formulaire

- Champ requis
- Limitation de la taille des entrées
- Validation selon une expression régulière

```
<form>
  <label>Un mot à 3 lettre
    <input type="text" required minlength="3"
      maxlength="3">
  </label>
  <label>Un mot qui commence par a ?
    <input type="text" required pattern="a.*"><br/>
  </label>
  <label>Combien en voulez-vous ?
    <input type="number" value="1" min="1" max="10">
  </label>
  <button type="submit">Soumettre</button>
</form>
```

Un mot à 3 lettre

Un mot qui commence par a ?

Combien en voulez-vous ?

The image shows a visual rendering of the HTML form code. Red dashed boxes highlight the text input fields for 'Un mot à 3 lettre' and 'Un mot qui commence par a ?'. The 'Combien en voulez-vous ?' field is a number input with the value '1' and a 'Soumettre' button next to it.

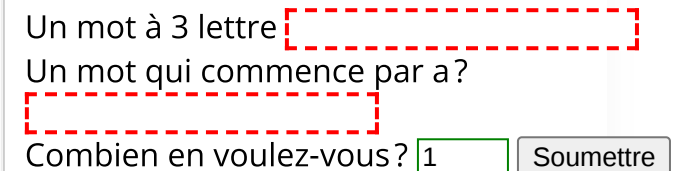
https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Validation_donnees_formulaire

Validation Formulaire et CSS

Des pseudo-class : `input:invalid` et `input:valid` sont automatiquement ajoutées aux éléments du formulaire.

```
input:invalid {  
  border: 2px dashed red;  
}  
input:valid {  
  border: 1px solid green;  
}
```

```
<form>  
  <label>Un mot à 3 lettre  
    <input type="text" required minlength="3"  
      maxlength="3">  
  </label>  
  <label>Un mot qui commence par a ?  
    <input type="text" required pattern="a.*"><br/>  
  </label>  
  <label>Combien en voulez-vous ?  
    <input type="number" value="1" min="1" max="10">  
  </label>
```



Un mot à 3 lettre

Un mot qui commence par a?

Combien en voulez-vous?

