

# HTML5

## Les balises et leurs attributs

### Les balises

Les pages HTML sont remplies de ce qu'on appelle des balises. Celles-ci sont invisibles à l'écran pour vos visiteurs, mais elles permettent à l'ordinateur de comprendre ce qu'il doit afficher. Les balises se repèrent facilement. Elles sont entourées de « chevrons », c'est-à-dire des symboles < et >, comme ceci : **<balise>**

A quoi est-ce qu'elles servent ? Elles indiquent la nature du texte qu'elles encadrent. Elles veulent dire par exemple : « Ceci est le titre de la page », « Ceci est une image », « Ceci est un paragraphe de texte », etc.

On distingue deux types de balises : les balises en paires et les balises orphelines.

#### Les balises en paires :

Elles s'ouvrent, contiennent du texte, et se ferment plus loin. Voici à quoi elles ressemblent :

```
<title>Ceci est un titre</title>
```

On distingue une balise ouvrante ( **<title>** ) et une balise fermante ( **</title>** ) qui indique que le titre se termine. Cela signifie pour l'ordinateur que tout ce qui n'est *pas* entre ces deux balises... n'est pas un titre.

#### Les balises orphelines :

Ce sont des balises qui servent le plus souvent à insérer un élément à un endroit précis (par exemple une image). Il n'est pas nécessaire de délimiter le début et la fin de l'image, on veut juste dire à l'ordinateur « Insère une image ici ».

Une balise orpheline s'écrit comme ceci :

```
<img />
```

### Les attributs

Les attributs sont un peu les options des balises. Ils viennent les compléter pour donner des informations supplémentaires. L'attribut se place après le nom de la balise ouvrante et a le plus souvent une valeur, comme ceci :

```
<balise attribut="valeur">
```

À quoi cela sert-il ? Prenons la balise **<img />** que nous venons de voir. Seule, elle ne sert pas à grand chose. On pourrait rajouter un attribut qui indique le nom de l'image à afficher :

```

```

## Structure de base d'une page HTML5

Ce code correspond à la base d'une page web en HTML5 :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Titre</title>
  </head>
  <body>

  </body>
</html>
```

**Insérer un commentaire :** Un commentaire est une balise HTML avec une forme bien spéciale :

```
<!-- Ceci est un commentaire -->
```

**Le doctype :** Elle est indispensable car c'est elle qui indique qu'il s'agit bien d'une page web HTML.

Ce n'est pas vraiment une balise comme les autres (elle commence par un point d'exclamation), vous pouvez considérer que c'est un peu l'exception qui confirme la règle.

**La balise `<html>` :** C'est la balise principale du code. Elle englobe tout le contenu de votre page. Comme vous pouvez le voir, la balise fermante `</html>` se trouve tout à la fin du code !

**L'en-tête `<head>` et le corps `<body>` :** Une page web est constituée de deux parties :

- L'en-tête `<head>` : cette section donne quelques informations générales sur la page comme son titre, l'encodage (pour la gestion des caractères spéciaux), etc. Cette section est généralement assez courte. Les informations que contient l'en-tête ne sont pas affichées sur la page, ce sont simplement des informations générales à destination de l'ordinateur. Elles sont cependant très importantes !
- Le corps `<body>` : c'est là que se trouve la partie principale de la page. Tout ce que nous écrirons ici sera affiché à l'écran. C'est à l'intérieur du corps que nous écrirons la majeure partie de notre code.

### Les métadonnées `<meta/>`

Représente toute information de métadonnées qui ne peut pas être représentée par un des autres éléments de métadonnées. Les attributs pour cette balise sont :

- **charset** : Cet attribut indique l'encodage des caractères pour le document.
- **name** : Cet attribut définit le nom d'une métadonnée au niveau du document.
- **content** : Cet attribut fournit la valeur associée avec l'attribut **name** suivant le contexte utilisé.

```
<!-- l'encodage de la page (pour le navigateur et pour le référencement) -->
<meta charset="UTF-8">
<!-- les description de la page (pour le référencement) -->
<meta name="description" content="Free Web tutorials">
<!-- les mots clés de la page (pour le référencement) -->
<meta name="keywords" content="HTML, CSS, JavaScript">
<!-- l'auteur de la page (pour le référencement) -->
<meta name="author" content="John Doe">
```

### Le titre principal de la page `<title>`

C'est le titre de votre page. Le titre ne s'affiche pas dans votre page mais en haut de celle-ci (souvent dans l'onglet du navigateur).

### La balise de liaison `<link>`

Définit la relation entre le document actuel et une ressource externe.

```
<link rel="stylesheet" href="styles.css" type="text/css">
```

### La balise de script `<script>`

Utilisé pour intégrer ou faire référence à un script exécutable. Cela fait généralement référence à du code JavaScript.

Les attributs possibles sont :

- **src** : Cet attribut définit l'URI d'un script externe.
- **type** : Cet attribut indique le type de script.

Soit en faisant appel au fichier qui contient le code :

```
<script src="mon_script.js" type="text/javascript"></script>
```

Ou en intégrant directement le code soit dans le `<head>` en dant le `<body>` :

```
<script>
    alert("Bonjour") ;
</script>
```

### La balise de style

**<style>**

Contient des informations de mise en forme pour un document ou une partie d'un document. Les attributs possibles sont :

- **type** : Cet attribut définit le langage de la feuille de style. Par défaut la valeur est **text/css**.
- **media** : indique le média auquel la mise en forme doit s'appliquer. La valeur par défaut est **All**. Il existe autres valeurs (screen, print, ...)

```
<style media="all" type="text/css">
  p {
    color: blue;
    background-color: yellow;
  }
</style>
```

## Organiser le texte et le média

### Les paragraphes

**<p>**

La plupart du temps, lorsqu'on écrit du texte dans une page web, on le fait à l'intérieur de paragraphes. Le langage HTML propose justement la balise **<p>** pour délimiter les paragraphes.

```
<p>Bonjour et bienvenue sur mon site !</p>
```

- **<p>** signifie « Début du paragraphe » ;
- **</p>** signifie « Fin du paragraphe ».

### Sauter une ligne

**<br>**

En HTML, si vous appuyez sur la touche Entrée, cela ne crée pas une nouvelle ligne comme vous en avez l'habitude. C'est une balise **orpheline** qui sert juste à indiquer qu'on doit aller à la ligne : **<br>**.

### Ligne horizontale

**<hr>**

Il représente une ligne horizontale.

### Les titres

**<h<sub>n</sub>>**

En HTML, on a le droit d'utiliser six niveaux de titres différents. On a donc six balises de titres différentes :

- **<h1> </h1>** : signifie « titre très important ».
- **<h2> </h2>** : signifie « titre important ».
- **<h3> </h3>** : pareil, c'est un titre un peu moins important
- **<h4> </h4>** : titre encore moins important.
- **<h5> </h5>** : titre pas important.
- **<h6> </h6>** : moins d'importance.

### Les listes

Les listes nous permettent souvent de mieux structurer notre texte et d'ordonner nos informations. Nous allons découvrir ici deux types de listes :

- Les listes non ordonnées ou listes à puces ;
- Les listes ordonnées ou listes numérotées ou encore énumérations.

#### Liste non-ordonnée

**<ul>**

Permet de créer une liste d'éléments sans notion d'ordre (il n'y a pas de « premier » ni de « dernier »).

```
<ul>
  <li> Football</li>
  <li> Handball</li>
  <li> Volleyball</li>
</ul>
```

- **<ul></ul>** délimite toute la liste.
- **<li></li>** délimite un élément de la liste (une puce).

## Liste ordonnée

<ol>

Une liste ordonnée fonctionne de la même façon, seule une balise change : il faut remplacer <ul></ul> par <ol></ol>. À l'intérieur de la liste, on ne change rien : on utilise toujours des balises <li></li> pour délimiter les éléments.

```
<ol>
  <li>Je me lève.</li>
  <li>Je prend mon petit déjeuner.</li>
  <li>Je vais au lycée.</li>
</ol>
```

Les attributs possibles pour la balise <ol> :

- **reversed** : indique que les éléments de la liste sont dans l'ordre inverse (par exemple : 3/, 2/, 1/, ...)
- **type** : Définit le type de numérotation :
  - **a** pour les lettres minuscules
  - **A** pour les lettres majuscules
  - **i** pour les chiffres romains minuscules
  - **I** pour les chiffres romains majuscules
  - **1** pour les chiffres (par défaut)
- **start** : Un nombre entier à partir duquel on commence à compter pour les éléments de la liste. Toujours un chiffre arabe même lorsque le type de numérotation est constitué de lettres ou de chiffres romains ou autre

```
<ol type='a' start='6' reversed>
  <li>Je me lève.</li>
  <li>Je prend mon petit déjeuner.</li>
  <li>Je vais au lycée.</li>
</ol>
```

f. Je me lève.  
e. Je prend mon petit déjeuner.  
d. Je vais au lycée.

## Créer des liens <a>

On peut faire un lien d'une page a.html vers une page b.html, mais on peut aussi faire un lien **URL** vers un autre site (par exemple, <http://www.siteduzero.com>). Dans les deux cas, nous allons voir que le fonctionnement est le même.

Pour faire un lien, la balise que nous allons utiliser est très simple à retenir : <a>. Il faut cependant lui ajouter un attribut, **href**, pour indiquer vers quelle page le lien doit conduire.

Comment faire un lien de la page 1 vers la page 2, il suffit d'écrire comme cible du lien le nom du fichier vers lequel on veut amener. Par exemple : <a href="page2.html">.

## L'attribut target

```
<a href="mapage.html" target="pr2">Aller vers ma page</a>
```

L'attribut **target** indique où le lien va être ouvert, elle peut prendre les valeurs suivantes :

- Un nom d'une **iframe**, comme l'exemple ci-dessus.
- **\_self** : le contexte de navigation actuel c'est-à-dire la page où se trouve le lien.
- **\_blank** : une nouvelle page ou onglet.
- **\_parent** : le contexte de navigation parent c'est-à-dire dans la page parent.
- **\_top** : le contexte de navigation le plus haut c'est-à-dire dans la page du parent plus haut.

## Un lien vers une ancre

Une **ancre** est une sorte de point de repère que vous pouvez mettre dans vos pages HTML lorsqu'elles sont très longues.

En effet, il peut alors être utile de faire un lien pour que le visiteur puisse sauter directement à la partie qui l'intéresse.

Pour créer une ancre, il suffit de rajouter l'attribut **id** à une balise qui va alors servir comme un nom de l'ancre. Cela nous servira ensuite pour faire un lien vers cette ancre. Par exemple :

```
<h2 id="mon_ancre">Titre</h2>
```

Ensuite, il suffit de créer un lien comme d'habitude, mais cette fois l'attribut **href** contiendra un dièse (#) suivi du nom de l'ancre.

```
<a href="#mon_ancre">Aller vers l'ancre</a>
```

## Les images `<img/>`

La balise qui va nous permettre d'insérer une image est `<img />` !

Les attributs pour cette balise sont :

- **src** : il permet d'indiquer où se trouve l'image que l'on veut insérer.
- **alt** : cela signifie « texte alternatif ». C'est-à-dire un court texte qui décrit ce que contient l'image. Ce texte sera affiché à la place de l'image si celle-ci ne peut pas être téléchargée.

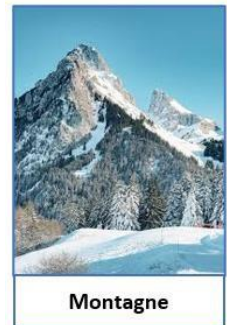
```
<p>Voici une photo que j'ai prise lors de mes dernières vacances
: <br>
    
</p>
```

## Les figures `<figure>` et `<figcaption>`

`<figure>` spécifie un conteneur autonome qui peut contenir un élément qui est généralement une image.

`<figcaption>` définit une légende pour cet élément.

```
<figure>
  
  <figcaption>
    Montagne
  </figcaption>
</figure>
```



## Les multimédias `<video>`, `<audio>` et `<source>`

`<video>` permet d'insérer une vidéo dans la page Web.

`<audio>` permet d'insérer un audio dans la page Web.

Les attributs à utiliser pour `<video>` et `<audio>` :

- **src** : pour spécifier le fichier source de vidéo ou de l'audio.
- **controls** : Si l'attribut est présent, le navigateur affichera des contrôles pour que l'utilisateur puisse gérer la lecture, le volume, et le déplacement du curseur de lecture.

```
<video src="media/cours1.mp4" controls></video>
<audio src="media/symphonie9.mp3" controls></audio>
```

Parfois, il est indispensable de donner plusieurs formats du vidéo ou de l'audio pour que le navigateur ou l'utilisateur choisit laquelle des formats est adéquate pour lui. Donc, dans ce cas on a recours à la balise `<source>` pour avoir plusieurs sources du même média. La balise `<source>` contient les attributs suivants :

- **src** : pour spécifier le fichier source de vidéo ou de l'audio (obligatoire).
- **type** : indique le type de la ressource (optionnel).
- **media** : indique une requête media (hors programme). Elle mentionné seulement avec la balise `<picture>` (hors programme aussi).

```
<video controls>
  <source src="media/cours1.mp4" type="video/mp4"/>
  <source src="media/cours1.webm" type="video/webm"/>
</video>
<audio controls>
  <source src="media/symphonie9.mp3" type="audio/mp3"/>
  <source src="media/ symphonie9.ogg"/>
</audio>
```

On peut ajouter des autres attributs soit au vidéo ou à l'audio comme :

- **autoplay** : indique que le média doit automatiquement être lancée.
- **muted** : le son est coupé.

```
<video src="media/cours1.mp4" controls autoplay muted></video>
```

Le vidéo  
"media/cours1.mp4"  
apparaît sur le navigateur avec  
ces contrôles, il va être lancé  
automatiquement avec le son  
coupé.

## Citation

### <cite>

Utilisé pour mentionner le titre d'un œuvre (livre, film, chanson, ...)

```
<p>Albert Einstein a publié son livre :
  <cite>La théorie de la relativité restreinte et
générale</cite> vers l'année 1916
</p>
```

Albert Einstein a publié son livre :  
*La théorie de la relativité restreinte  
et générale* vers l'année 1916

## Adresse

### <address>

Utilisé pour mentionner l'adresse physique ou numérique.

```
<p>Pour nous contacter, envoyer à :
  <address>webmaster@openclass.com</address >
Ou rejoindre l'adresse :
  <address>5, Avenue Ali Balhouen, Tunis 1002</address >
</p>
```

Pour nous contacter, envoyer à :  
*webmaster@openclass.com*  
Ou rejoindre l'adresse :  
*5, Avenue Ali Balhouen, Tunis 1002*

## Marquage de texte

### <mark>

Utilisé pour marquer un texte à cause de son importance dans le contexte

```
<p>Search results for "salamander":
Several species of <mark>salamander</mark> inhabit the
temperate rainforest of the Pacific Northwest.
Most <mark>salamander</mark>s are nocturnal, and hunt for
insects, worms, and other small creatures.</p>
```

Search results for "salamander": Several  
species of **salamander** inhabit the  
temperate rainforest of the Pacific  
Northwest. Most **salamander**s are  
nocturnal, and hunt for insects, worms,  
and other small creatures.

## Divulgence des détails

### <details> et <summary>

La balise **<details>** sert à révéler ou cacher une information (texte, table, image, ...) lors du clic sur un élément contenu dans la balise **<summary>**. Si l'attribut **open** est présent, l'information à révéler est visible initialement.

```
<details>
  <summary>Les détails ici</summary>
  Dans cette zone, vous verrez toutes les détails.</details>
```

```
<details open>
  <summary>Les détails ici</summary>
  Dans cette zone, vous verrez toutes les détails.</details>
```

**A**

► Les détails ici

**B**

▼ Les détails ici

Dans cette zone, vous verrez toutes  
les détails.

**open** absent → A est l'état initial || **open** présent → B est l'état initial || par un clic, A mène vers B et B mène vers A

## Élément de sortie

### <output>

Représente un conteneur dans lequel on peut injecter le résultat d'un calcul ou d'une action utilisateur.

```
<form oninput="result.value=parseInt(a.value)+parseInt(b.value)">
  <input type="number" name="a"/> +
  <input type="number" name="b"/> =
  <output name="result"></output>
</form>
```

21 + 23 = 44

## Les conteneurs

### Cadre intégré

### <iframe>

Utilisé pour intégrer une page HTML dans une autre. Une page HTML qui contient une cadre **<iframe>** est considéré comme contexte de navigation **parent** pour ce cadre. Les attributs nécessaires pour cette balise sont :

- **src** : l'adresse de la page HTML qui remplira ce cadre (peut être une URL).
- **name** : un nom pour la cadre qui va servir après comme **target** pour les éléments **<a>** ou **<form>**.

```
<iframe src="inscription.html" name="frame2"
  width="400" height="300">
</iframe>
```

On peut mentionner les dimensions du cadre ici  
dans le HTML ou le rapporter au CSS.

## Les balises universelles

### <span> et <div>

Il arrivera parfois que vous ayez besoin d'appliquer une class (ou un id) à certains mots qui, à l'origine, ne sont pas entourés par des balises. En effet, le problème de class, c'est qu'il s'agit d'un attribut. Vous ne pouvez donc en mettre que sur une balise. Si, par exemple, je veux modifier uniquement « **bienvenue** » dans le paragraphe suivant :

```
<p>Bonjour et bienvenue sur mon site !</p>
```

Cela serait facile à faire s'il y avait une balise autour de « **bienvenue** ». Par chance, on a inventé... la balise **<span>**.

#### Code HTML

```
<p>Bonjour et <span class="salutations">bienvenue</span>
sur mon site !</p>
```

#### Code CSS

```
.salutations {
  color: blue; }
```

Si maintenant nous voulons regrouper plusieurs balises :

#### Code HTML

```
<div class="montexte">
  <p>Bonjour et bienvenue sur mon site !</p>
  <p>Le site est en train de construction</p>
  <h2>Faites une visite</h2>
</div>
```

#### Code CSS

```
.montexte {
  color: blue;
}
```

En fait, on a inventé deux balises dites **universelles**, qui n'ont aucune signification particulière. Il y a une différence minime (mais significative !) entre ces deux balises :

- **<span> </span>** : c'est une balise de type **inline**, c'est-à-dire une balise que l'on place au sein d'un paragraphe de texte, pour sélectionner certains mots uniquement.
- **<div> </div>** : c'est une balise de type **block**, qui entoure un bloc de texte. Les balises **<p>**, **<h1>**, etc. sont de la même famille. Ces balises ont quelque chose en commun : elles créent un nouveau « bloc » dans la page et provoquent donc obligatoirement un retour à la ligne.

## Les tableaux

Pour créer des tableaux dans notre page HTML, on a recouru aux balises suivantes :

- **<table>** : permet de définir un tableau (obligatoire).
- **<caption>** : permet de définir une légende (titre) pour le tableau.
- **<thead>** : délimite l'entête du tableau.
- **<tbody>** : délimite le corps du tableau.
- **<tfoot>** : délimite le pied du tableau.
- **<tr>** : définit une ligne dans le tableau (obligatoire).
- **<th>** : définit une cellule d'entête (ou spécial) dans le tableau.
- **<td>** : définit une cellule dans le tableau (obligatoire).

Person	Most interest in	Age
Chris	HTML tables	22
Dennis	Web accessibility	45
Sarah	JavaScript frameworks	29
Karen	Web performance	36
Average age		33



```
<table>
  <caption>Front-end web developer course
  2021</caption>
  <thead>
    <tr> <th>Person</th>
    <th>Most interest in</th>
    <th>Age</th> </tr>
  </thead>
  <tbody>
    <tr> <th>Chris</th>
    <td>HTML tables</td>
    <td>22</td> </tr>
    <tr> <th>Dennis</th>
    <td>Web accessibility</td>
    <td>45</td> </tr>
    <tr> <th>Sarah</th>
    <td>JavaScript frameworks</td>
    <td>29</td> </tr>
    <tr> <th>Karen</th>
    <td>Web performance</td>
    <td>36</td> </tr>
  </tbody>
  <tfoot>
    <tr> <th colspan="2">Average age</th>
    <td>33</td>
  </tr>
  </tfoot>
</table>
```

- Les balises ci-dessous donnent une structure au tableau. Pour une bonne mise en forme, le CSS est indispensable.
- L'attribut **colspan** sert à fusionner plusieurs colonnes dans une seule (ici 2).
- Il existe aussi l'attribut **rowspan** qui sert à fusionner plusieurs lignes dans une seule.



## Les évènements

Les évènements permettent d'effectuer une action à des **moments** bien précis, cette **action** peut être :

- Un code HTML comme `onfocus="input1.value=input2.value+input3.value"`
- Un code CSS comme `onfocus="style='color :blue;'"`
- Un code JavaScript comme `onfocus="mafonction()"` ou `onfocus="alert('Bienvenue')"`

### Evènement de la page : onload

Se déclenche lorsque la page est complètement chargée.

```
<body onload="alert('Bienvenue')">
```

Lorsque la page est chargée, un message de Bienvenue sera affiché.

### Evènements des éléments du formulaire : onsubmit, oninput, onblur, onfocus, onchange

- **onsubmit** : se déclenche lorsqu'un formulaire est soumis au serveur (validé).
- **oninput** : se déclenche dès que la valeur d'un élément change.
- **onblur** : se déclenche au moment où l'élément perd le focus.
- **onfocus** : se déclenche au moment où l'élément obtient le focus.
- **onchange** : se déclenche lorsque la valeur d'un élément change.

La différence entre **oninput** et **onchange** est que **oninput** se déclenche directement au moment où la valeur de l'élément change alors que **onchange** ne se déclenche pas immédiatement lorsque la valeur de l'élément change mais attend que cet élément perde le focus pour se déclencher, autrement : **onchange = oninput and onblur**

```
<input onblur="style='color:red;'">
```

Lorsque l'élément input perd le focus, son contenu devient de couleur rouge.

### Evènements de la souris : onclick, onmouseover, onmouseout

- **onclick** : se déclenche lors d'un clic sur l'élément.
- **onmouseover** : se déclenche lorsque le pointeur de la souris survole l'élément.
- **onmouseout** : se déclenche lorsque le pointeur de la souris se déplace en dehors de l'élément.

```
<h1 onmouseout="style='background-color:red;'">
```

Lorsque le pointeur de la souris quitte l'élément h1, son arrière-plan devient de couleur rouge.

### Evènements du clavier : onkeydown, onkeyup

- **onkeydown** : se déclenche lorsque l'utilisateur appuie sur une touche du clavier.
- **onkeyup** : se déclenche lorsque qu'une touche du clavier qui a été pressée est relâchée.

HTML

```
<input id="A" onkeydown="donner1(event)">
<input id="B" onkeydown="donner2(event)">
```

JS

```
function donner1(e) {
    alert(e.code)
}
function donner2(e) {
    alert(e.keyCode)
}
```

Le mot clé **event** représente la touche appuyée. Si le curseur est dans l'input A et on appuie sur une touche du clavier, onkeydown est déclenché et la fonction donner1 est exécutée. (De même pour B et donner2).  
donner1 affiche en message le nom de la touche (si on appuie sur la touche F il affiche **KeyF**)  
donner2 affiche en message le code ascii de la touche (si on appuie sur la touche F il affiche **70**)

**event** : mot clé passé en paramètre, fait référence à la touche appuyée

**e.code** : fait référence à la touche appuyée (si touche R appuyée, e.code donne KeyR)

**e.keyCode** : fait référence au code ASCII de la touche appuyée (si touche R appuyée, e.keyCode donne 82)



## Evènements du média : onplay, onpause

- **onplay** : se déclenche lorsqu'un élément média (audio ou vidéo) est démarré.
- **onpause** : se déclenche lorsqu'un élément média (audio ou vidéo) est mis en pause ou atteint la fin.

```
<video onplay="marche()" onpause="arret()" controls>
  <source src="media/video4.mp4"/>
</video>
```

Lorsque le vidéo est mis en marche, **onplay** se déclenche et fait appel à la fonction JS **marche()**. Et lorsque le vidéo est arrêté ou terminé, **onpause** se déclenche et fait appel à la fonction JS **arret()**.

## Les éléments de formulaire

### Ouvrir un formulaire **<form>**

La balise **<form>** sert comme conteneur des éléments de formulaire. Les attributs possibles pour cette balise sont :

- **name** : affecte un nom à la balise.
- **action** : détermine l'emplacement (local ou URL) du fichier (généralement de type PHP) qui va traiter le formulaire après sa validation.
- **method** : indique par quel moyen les données de formulaire seront envoyées. Il existe deux méthodes :
  - **"get"** : c'est une méthode en général assez peu adaptée car elle est limitée à 255 caractères. Les informations seront envoyées dans l'adresse URL de la page (http ...).
  - **"post"** : c'est la méthode la plus utilisée car elle permet d'envoyer un grand nombre d'informations. Cette méthode est plus sécurisée car les données saisies dans le formulaire ne transitent pas par la barre d'adresse.
- **target** : indique où le fichier d'action va être ouvert, elle peut prendre les valeurs suivantes :
  - Un nom d'une **iframe**, comme l'exemple ci-dessus.
  - **\_self** : le contexte de navigation actuel c'est-à-dire la page où se trouve le lien.
  - **\_blank** : une nouvelle page ou onglet.
  - **\_parent** : le contexte de navigation parent c'est-à-dire dans la page parent.
  - **\_top** : le contexte de navigation le plus haut c'est-à-dire dans la page du parent plus haut.

### Regroupement **<fieldset>** et **<legend>**

La balise **<fieldset>** permet de regrouper un ensemble des éléments de formulaire et affecter un titre ou une description à cet ensemble à l'aide de la balise **<legend>**.

### Libellé **<label>**

La balise **<label>** permet de définir un libellé pour un élément de formulaire.

```
<form name="f1" method="post" action="inscrit.php" onsubmit="return
controle()" target="_self">
<fieldset>
  <legend>Données Personnelles</legend>
  <p><label>Nom :</label><input type="text" name="nm" id="nm"/></p>
  <p><label>Prénom :</label><input type="text" name="pr" id="pr"/></p>
  <p><input type="submit" value="Valider"/></p>
</fieldset>
</form>
```

## Champs de saisie

## <input/>

### Champs de saisie textuels

### text, password, email et tel

Un champ de saisie de type **text**, **password**, **email** ou **tel** permet de saisir un texte. Cependant il y a des légères différences :

- **text** : texte quelconque.
- **password** : texte quelconque mais caché derrière un caractère spécial (\* ou .) pour des raisons de sécurité.
- **email** : texte quelconque mais qui doit comporter un @. A ne pas compter pas sur ce type pour avoir un email valide, mieux de le faire tout seul à l'aide de JavaScript.
- **tel** : texte quelconque. Il faut un contrôle JavaScript au cas de vouloir avoir des chiffres seulement.

Les attributs possibles sur ces champs de saisie sont :

- **name** : nom de l'élément.
- **value** : valeur de l'élément.
- **placeholder** : texte d'indication qui apparaît sur le champ lorsqu'il est encore sans valeur.
- **readonly** : l'élément est accessible en lecture seulement. On ne peut changer sa valeur.
- **required** : le remplissage de cet élément est obligatoire. Interdit de le laisser vide.
- **disabled** : désactiver l'élément, devient inaccessible et inchangeable.
- **list** : contient l'**id** de **<datalist>** en cas ce champ va être une liste déroulante. Cet attribut ne convient pas avec le type **password**.

```
<form name="f1">
<p><label>Nom :</label>
<input type="text" name="nm" id="nm" placeholder="Taper le nom" /></p>
<p><label>Prénom :</label>
<input type="text" name="pr" id="pr" disabled/></p>
<p><p><label>Mot de passe :</label>
<input type="password" value="hello"/></p>
</form>
```

### Champs de saisie numériques

### number et range

Un champ de saisie de type **number** ou **range** permet de saisir des nombres :

- **number** : pour saisir un nombre (entier ou décimal).
- **range** : pour saisir un nombre entier à l'aide d'un curseur à déplacer entre deux bornes.

Les attributs possibles sur ces champs de saisie sont :

- **name** : nom de l'élément.
- **value** : valeur de l'élément.
- **readonly** : l'élément est accessible en lecture seulement. On ne peut changer sa valeur.
- **required** : le remplissage de cet élément est obligatoire. Interdit de le laisser vide.
- **disabled** : désactiver l'élément, devient inaccessible et inchangeable.
- **min** : une valeur minimale du nombre à saisir.
- **max** : une valeur maximale du nombre à saisir.

```
<form name="f1">
<p><label>Age :</label>
<input type="number" name="ag" id="ag" min="10" max="100" /></p>
<p><label>Niveau :</label>
<input type="range" name="nv" id="nv" min="1" max="10" value="8"/></p>
</form>
```

**Champs de saisie de date et heure****date et time**


Un champ de saisie de type :


- **date** : permet à l'utilisateur de saisir une date en lui fournissant un calendrier pour choisir la date ou l'écrire directement en suivant le bon format.
- **time** : permet à l'utilisateur de saisir une heure en lui fournissant un montre numérique pour choisir l'heure ou l'écrire directement en suivant le bon format.

Les attributs possibles sur ces champs de saisie sont :

- **name** : nom de l'élément.
- **value** : valeur de l'élément.
- **required** : le remplissage de cet élément est obligatoire. Interdit de le laisser vide.
- **disabled** : désactiver l'élément, devient inaccessible et inchangeable.
- **min** : une valeur minimale du date ou heure à saisir.
- **max** : une valeur maximale du date ou heure à saisir.

```
<form name="f1">
<p><label>Date de naissance :</label>
<input type="date" name="dt" id="dt" min="2002-01-01"/></p>
<p><label>Rendez-vous :</label>
<input type="time" name="tm" id="tm" min="08:00" max="12:00"
value="09:30"/></p>
</form>
```

Date de naissance :  

Rendez-vous :  

**Champs de saisie des cases à choix****checkbox et radio**

Un champ de saisie de type :

- **checkbox** : permet à l'utilisateur de cocher une case pour faire un choix. Il peut cocher plusieurs cases à la fois.
- **radio** : permet aussi à l'utilisateur de cocher une case pour faire un choix. Généralement les boutons radio sont utilisés en famille (une famille à le même **name**) pour ne pouvoir cocher qu'une seule case.

Les attributs possibles sur ces champs de saisie sont :

- **name** : nom de l'élément.
- **value** : valeur de l'élément.
- **required** : le remplissage de cet élément est obligatoire. Interdit de le laisser vide.
- **disabled** : désactiver l'élément, devient inaccessible et inchangeable.
- **checked** : si cet attribut est présent, la case sera cochée par défaut.

```
<form name="f1">
<p><label>Genre :</label>
<input type="radio" name="gr" value="H" checked/>Homme
<input type="radio" name="gr" value="F"/>Femme </p>
<p><label>Connaissance :</label>
<input type="checkbox" name="ch1" value="HTML" checked/>HTML
<input type="checkbox" name="ch2" value="CSS" checked/>CSS
<input type="checkbox" name="ch3" value="SQL"/>SQL
</p>
</form>
```

Genre : ☒ Homme ☐ Femme

Connaissances : ☒ HTML ☒ CSS ☐ SQL

Pour une famille de boutons radio (même name), on envoie au serveur (pour le PHP) le couple (**name** : **value**) où le **name** est le nom commun des boutons et le **value** est la valeur du bouton coché.

Dans l'exemple le formulaire soumet au serveur le couple ("gr" = "H").

Pour les boutons checkbox, on envoie au serveur (pour le PHP) pour chaque bouton son propre couple (**name** : **checked**) où le **name** est le nom du bouton et **checked** est une valeur booléenne qui indique si ce bouton est coché ou non. Dans l'exemple le formulaire soumet au serveur les couples ("ch1" = "true") ("ch2" = "true") ("ch3" = "false")

## Les boutons **button, reset et submit**

Un champ de saisie de type :

- **button** : un bouton qui n'a pas un rôle spécifique. Il peut être programmé pour appeler une fonction JavaScript lorsqu'il est cliqué.
- **reset** : un bouton qui réinitialise tous les champs du formulaire à leurs valeurs initiales.
- **submit** : un bouton qui soumet (valide) le formulaire au serveur.

Les attributs possibles sur ces champs de saisie sont :

- **name** : nom de l'élément.
- **value** : valeur de l'élément. C'est le texte qui apparaît sur le bouton.
- **disabled** : désactiver l'élément, devient inaccessible et inchangeable.

```
<form name="f1">
<p><label>Nom&Prénom :</label>
<input type="text" name="np" id="np"/></p>
<p><label>Date Naissance :</label>
<input type="date" name="dn" name="dn"/></p>
<p><input type="reset" value="Annuler"/>
<input type="submit" value="Valider"/></p>
</form>
```

## Zone de saisie multiligne **<textarea>**

Un champ de saisie défini par la balise **<textarea>** permet de saisir un texte sur plusieurs lignes.

Les attributs possibles sur ce champ de saisie sont :

- **name** : nom de l'élément.
- **placeholder** : texte d'indication qui apparaît sur le champ lorsqu'il est encore sans valeur.
- **readonly** : l'élément est accessible en lecture seulement. On ne peut changer sa valeur.
- **required** : le remplissage de cet élément est obligatoire. Interdit de le laisser vide.
- **disabled** : désactiver l'élément, devient inaccessible et inchangeable.
- **cols** : largeur de la zone de saisie comptée en nombre de caractères.
- **rows** : nombre de ligne visibles.
- **maxlength** : longueur maximale de la chaîne à saisir comptée en nombre de caractères.

```
<form name="f1">
<p><label>Remarques :</label>
<textarea name="rq" id="rq" cols="30" rows="4"
maxlength="100"></textarea></p>
</form>
```

## Liste classique **<select> et <option>**

La balise **<select>** permet de créer une liste déroulante. Les choix dans cette liste sont définis par la balise **<option>**.

Si une option est accompagnée par l'attribut **selected**, elle sera choisie par défaut et elle apparaît dès le début dans la zone visible de la liste.

```
<form name="f1">
<p><label>Niveau :</label>
<select name="niv" id="niv">
<option value="niv1">1ère année</option>
<option value="niv2">2ème année</option>
<option value="niv3">3ème année</option>
<option value="niv4" selected>4ème année</option>
</select></p>
</form>
```

Lors de validation, on envoie au serveur (pour le PHP) le couple (**name** : **value**) où **name** est le nom de la liste alors que **value** est la valeur de l'option choisit. Dans l'exemple le formulaire soumet au serveur le couple ("niv" = "niv4") Pour le JavaScript on utilise :  
1<sup>ère</sup> année, 2<sup>ème</sup> année, ... n'ont aucun sens pour le PHP.

## Liste avec `datalist`

### `<input>`, `<datalist>` et `<option>`

La balise `<input>` vu précédemment permet de créer une liste déroulante si elle est liée à une balise `<datalist>`. La liaison s'effectue lorsque l'attribut `list` de `<input>` prend comme valeur l'`id` de la `<datalist>`. Les choix dans cette liste sont définis par la balise `<option>`.

L'attribut `selected` ne fonctionne pas ici.

```
<form name="f1">
<p><label>Niveau :</label>
<input name="niv" id="niv" list="maliste"/>
<datalist id="maliste" >
<option value="niv1">1ère année</option>
<option value="niv2">2ème année</option>
<option value="niv3">3ème année</option>
<option value="niv4">4ème année</option>
<datalist></p>
</form>
```

```
<form name="f1">
<p><label>Niveau :</label>
<input name="niv" id="niv" list="maliste"/>
<datalist id="maliste" >
<option value="niv1">
<option value="niv2">
<option value="niv3">
<option value="niv4">
<datalist></p>
</form>
```

## Les éléments de structuration

Les balises suivantes servent à structurer la page Web en le décomposant en des parties dont chacune joue un rôle spécifique. NOTER BIEN que la charge principale de structuration est supportée par le CSS qui aura la tâche de placer chaque partie dans une zone spécifique.

### Entête `<header>`

Généralement c'est la zone en haut de la page qui contient un logo, slogan ou bannière de site. Les `<section>` aussi peuvent avoir des `<header>`.

### Pied `<footer>`

Généralement c'est la zone en bas de la page qui contient les liens de contact, le nom de l'auteur ou les mentions légales. Les `<section>` aussi peuvent avoir des `<footer>`.

### Navigation `<nav>`

Regroupe tous les principaux liens de navigation du site. Vous y placerez par exemple le menu principal de votre site.

### Section `<section>`

Regroupe des contenus en fonction de leur thématique. Elle englobe généralement une portion du contenu au centre de la page.

### Côté `<aside>`

Contient des informations complémentaires au document que l'on visualise. Ces informations sont généralement placées sur le côté. Il peut y avoir plusieurs blocs `<aside>` dans la page.

### Article `<article>`

Sert à englober une portion généralement autonome de la page. C'est une partie de la page qui pourrait ainsi être reprise sur un autre site. C'est le cas par exemple des actualités (articles de journaux ou de blogs).

### Zone Principale `<main>`

Représente le contenu dominant d'un document. La zone de contenu principale se compose de contenu directement lié ou développant le thème central d'un document ou la fonctionnalité centrale d'une application.

```

<header>
  <h1>ARCAN</h1>
  <h2>Carnets de voyage</h2>
</header>
<nav>
  <ul>
    <li><a href="#">Accueil</a></li>
    <li><a href="#">Blog</a></li>
    <li><a href="#">CV</a></li>
  </ul>
</nav>
<section>
  <aside>
    <h1>À propos de l'auteur</h1>
    <p>C'est moi, Arcan ! Je suis né un 23
  </aside>
  <article>
    <h1>Je suis un grand voyageur</h1>
    <p>Bla bla bla bla (texte de l'article)</p>
  </article>
</section>
<footer>
  <p>Copyright Arcan - Tous droits réservés<br>
  <a href="#">Me contacter !</a></p>
</footer>

```

<h1>ARCAN</h1> <h2>Carnets de voyage</h2>	
<ul style="list-style-type: none"> <li>• <a href="#">Accueil</a></li> <li>• <a href="#">Blog</a></li> <li>• <a href="#">CV</a></li> </ul>	<h3>À propos de l'auteur</h3> <p>C'est moi, Arcan ! Je suis né un 23 novembre 2005.</p> <h3>Je suis un grand voyageur</h3> <p>Bla bla bla bla (texte de l'article)</p>
<p>Copyright Arcan - Tous droits réservés  <a href="#">Me contacter !</a></p>	

## Les attributs globaux

Se sont des attributs qui peuvent être présent avec n'importe qu'elle balise.

### Attribut de classe : **class**

Spécifie une classe pour un élément. Généralement une classe couvre plusieurs éléments.

### Attribut d'identifiant : **id**

Spécifie un identifiant unique pour un élément. A NE PAS affecter le même identifiant à deux éléments.

### Attribut de visibilité : **hidden**

Lorsque cet attribut est présent, l'élément devient invisible.

### Attribut de langue : **lang**

Spécifie la langue dont le contenu de l'élément est écrit avec.

### Attribut de style : **style**

Permet de déterminer un style CSS pour cet élément.

### Attribut de titre : **title**

Spécifie un titre pour cet élément qui sert comme des informations supplémentaires sur cet élément.

```

<h1 id="titre1" class="montitre" lang="fr">
  Cours HTML
</h1>
<p class="montitre" lang="en">
  This HTML course has just finished
</p>
<h1 style="color : red ;" title="special bac" >
  BONNE CHANCE AU BACCALAUREAT
</h1>
<p hidden class="montitre" lang="ar">
  أتمنى أن يساعدكم على النجاح
</p>

```