

# HTML



**Mars 2022**

© AFPA - Gilles Vanderstraeten  
gillesvds@adok.info

## Classes technologiques



**Semantics**

Offline & Storage

Device Access

Connectivity

Multimedia

3D, Graphics & Effects

Performance & Integration

CSS 3

## **Connaissances requises**

- Aucune.

## **Outils requis**

- Notepad++ ([//www.notepad-plus-plus.org](http://www.notepad-plus-plus.org))
- Edge ([//www.microsoft.com/edge](http://www.microsoft.com/edge))
- Firefox ([//www.firefox.com](http://www.firefox.com))
- Chrome ([//www.google.com/chrome](http://www.google.com/chrome))

## Documentation

- **WHATWG**  
[//developers.whatwg.org](http://developers.whatwg.org)
- **W3C**  
[www.w3.org/TR/html5](http://www.w3.org/TR/html5)
- **Mozilla**  
[//developer.mozilla.org/fr/docs/Web/HTML/Element](http://developer.mozilla.org/fr/docs/Web/HTML/Element)
- **Microsoft**  
[//html5labs.interoperabilitybridges.com](http://html5labs.interoperabilitybridges.com)

## Contenu

- Historique
- Présentation
- Balises de base
- Balises de structuration (1 TD)
- Balises de petite structuration
- Balises d'accrochage CSS
- Lexiques
- Listes

## Contenu (suite)

- Tableaux
- Liens
- Images
- Audio & Video
- Dessin vectoriel
- Formulaire (1 TD)
- Frames
- Attributs globaux

## Historique



- **1992**  
Au CERN de Genève, Tim Berners Lee (UK) invente le WEB en créant le protocole HTTP, **le langage HTML**, un serveur web et un navigateur.  
Il base HTML sur le très complexe méta-langage SGML.
- **1994**  
Tim Berners Lee crée le W3C ([//www.w3.org](http://www.w3.org)) dont il devient le président.

## Historique (suite)

- **1995** : **HTML 2** ajoute les formulaires.
- **1996** : **HTML 3** ajoute les tableaux.
- **1997** : **HTML 4** ajoute les *frames*.
- **2000** : Le W3C stoppe HTML et recommande XHTML 1.
- **2004**  
Apple, Mozilla et Opera fondent le WHATWG ([//www.whatwg.org](http://www.whatwg.org)) pour développer HTML 5.
- **2006**  
Le W3C recommande XHTML 1.1 et développe XHTML 2 qui ajoute les XForms et les XFrames mais est incompatible avec les anciens navigateurs.
- **2007**  
Le W3C abandonne XHTML 2 et collabore avec le WHATWG.
- **2014**  
Le W3C publie la recommandation HTML 5.  
[//www.w3.org/TR/html5](http://www.w3.org/TR/html5)

## Présentation

- **Langage de description côté client.**
- **Arborescence de balises (*tag*) hiérarchiques.**
- Eléments sémantiques forts, ex : `<img>`.
- Eléments sémantiques faibles, ex : `<div>`.
- Eléments techniques, ex : `<html>`.
- **Un élément peut contenir des attributs.**
- Un attribut est un couple : nom = valeur.



Les règles CSS peuvent modifier la sémantique des éléments HTML.



Ce cours ne liste que les éléments et attributs principaux pris en charge uniformément par Chrome, Firefox et IE sous Windows 7.

## Présentation

### **WHATWG *Living Standard* vs W3C *Recommendation***

- **WHATWG**

- Pas de version arrêtée, pas de numérotation, pas de norme.
- Une documentation agréable mais mouvante au jour le jour.
- Proche des implémentations des derniers navigateurs.
- Leur point de vue : [//wiki.whatwg.org/wiki/FAQ#The\\_WHATWG](http://wiki.whatwg.org/wiki/FAQ#The_WHATWG).

- **W3C**

- Des versions numérotées et normalisées.
- A terme, une documentation complète mais indigeste.
- 15 ans entre les recommandations HTML 4 et HTML.
- Constamment dépassé par les derniers navigateurs.
- Leur point de vue : [//www.w3.org/html/landscape](http://www.w3.org/html/landscape).

## Présentation

### **Compatibilité**

- **Niels Leenheer (NL)** : rapport de compatibilité.  
[//www.html5test.com](http://www.html5test.com)
- **Alexis Deveria (NL)** : grille de compatibilité recommandée par Mozilla.  
[//www.caniuse.com](http://www.caniuse.com)
- **Remy Sharp (UK)** : tests de compatibilité.  
[//html5demos.com](http://html5demos.com)
- **Microsoft** : *screenshots*, rapport de compatibilité...  
[//www.modern.ie/fr-fr](http://www.modern.ie/fr-fr)
- **StatCounter** : parts de marché des navigateurs.  
[//gs.statcounter.com](http://gs.statcounter.com)

# Présentation

## *Document-type*

- Obligatoire avant tout autre balise.
- `<!doctype html>`

## Commentaires

- `<!-- ... -->`

## HTML ou XHTML ?

- Toujours intéressant de pouvoir parser le document en XML donc :
  - **Respecter la hiérarchie.**  
Ex : `<b>Exemple <i>simple</i></b>`
  - **Auto-fermer les balises vides.**  
Ex : `<img ... />`
  - **Coupler les attributs booléens.**  
Ex : `<input ... required="required"/>`

# Présentation

## Les caractères spéciaux

- Indispensables :
  - Espace insécable : `&nbsp;`; (*non breakable space*)
  - `<` : `&lt;`; (*less than*)
  - `>` : `&gt;`; (*great than*)
  - `&` : `&amp;`; (*amperstand*)

# Présentation

## Structure minimale

```
<!doctype html>
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Ma page minimale</title>
  </head>
  <body>
    ... contenu ...
  </body>
</html>
```

## Balises de base

**<html>**

- **Racine du document.**
- Obligatoire et unique.



## Balises de base

### <head>

- **En-tête du document.** Hormis le titre, son contenu n'est pas visible.
- Premier enfant de <html>.
- Obligatoire et unique.
- Peut contenir :
  - <meta/>
  - <title>
  - <base/>
  - <link/>
  - <style>
  - <script>

## Balises de base

### <meta/>

- **Information sur la nature du document.**
- Si présent(s), obligatoirement dans <head>.
- Plusieurs utilisations :
  - **<meta charset="utf-8"/>**  
Toujours utiliser UTF-8 pour les pages web !
  - **<meta name="description" content="..." />**  
Description du document (SEO).
  - **<meta name="keywords" content="..." />**  
Mot-clés du document (SEO).
  - **<meta name="viewport"**  
    **content="width=device-width,**  
    **user-scalable=no"/>**  
Pour les sites destinés aux mobiles. Voir aussi :  
[//developer.mozilla.org/fr/docs/Mozilla/Mobile/Balise\\_meta\\_viewport](https://developer.mozilla.org/fr/docs/Mozilla/Mobile/Balise_meta_viewport).

## Balises de base

### `<title>`

- **Titre de la page** (apparaît dans la barre de titre).
- Si présent, unique dans `<head>`.
- Contient uniquement du texte.

### `<base/>`

- **URL de base** pour toutes les URLs relatives du document.
- Si présent, unique dans `<head>`.
- Élément vide.
- Attributs :
  - **`href="URL"`**  
URL de base.

## Balises de base

### `<link/>`

- **Chargement de fichiers.**
- Principalement utilisé pour fichiers (*external CSS*) et icônes de page.
- Si présent(s), obligatoirement dans `<head>`.
- Attributs :
  - **`href="URL"`**  
URL du fichier.
  - **`rel="stylesheet | icon"`**
  - **`type="text/css | image/png"`**
  - **`media="mediaQuery"`**  
Les *media-query* sont abordées dans le cours CSS.

## Balises de base

### <style>

- **Inclusion de règles CSS** (*internal CSS*).
- Si présent(s), obligatoirement dans <head>.
- Contient uniquement des règles CSS.
- Attributs :
  - **type="text/css"**
  - **media="mediaQuery"**  
Les *media-query* sont abordées dans le cours CSS.

## Balises de base

### <script>

- **Inclusion ou chargement d'un script.**
- Contient uniquement un script.
- Attributs :
  - **type="text/javascript"**
  - **src="URL"**  
Chargement d'un script
  - **async="true | false"**  
Si `src` présent : chargement asynchrone ou non (ordre).
  - **defer="true | false"**  
Exécution différée après chargement complet du document ou pas.



Parce que cette balise peut être vide (chargement) ou pas (inclusion), les navigateurs ne tolèrent pas qu'elle soit auto-fermée lorsqu'elle est vide.

## Balises de base

### `<body>`

- **Corps du document.** Son contenu est visible.
- Second enfant de `<html>`.
- Obligatoire et unique.

## Balises de structuration

### `<nav>`

- **Regroupement de liens de navigation.**

### `<header>`

- **En-tête d'une partie.**

### `<footer>`

- **Pied d'une partie.**

### `<article>`

- **Contenu autonome.** Devrait comporter un en-tête.

### `<section>`

- **Section d'un contenu autonome.** Devrait comporter un en-tête.

## Balises de structuration

`<aside>`

- Contenu textuel annexe.

`<figure>`

- Contenu média annexe.

`<figcaption>`

- Légende d'un média annexe.

## Balises de structuration

`<h1> . . . <h6>`

- **Titre d'un contenu.** Du plus important au moins important.



Ne peut contenir que du contenu phrasé : **PAS DE** `<h_>`, `<p>`, `<div>`.

`<p>`

- **Paragraphe.**



Ne peut contenir que du contenu phrasé. Voir ci-dessus.

## TD : structuration sémantique

- Aidez-vous de ce document pour coder cette page.
- Seul le code HTML est demandé, pas le code CSS, mais vous devez respecter la sémantique voulue par l'auteur.

## Balises de petite structuration



**Ne peuvent contenir que du contenu phrasé.** Voir plus haut.

**<abbr>**

- **Abbréviation.**
- Attributs :
  - **title="description"**  
Description de l'abbréviation (infobulle).

**<address>**

- **Coordonnées.**

## Balises de petite structuration



**Ne peuvent contenir que du contenu phrasé.** Voir plus haut.

`<blockquote>`

- Citation longue.

`<q>`

- Citation courte (une ligne).

## Balises de petite structuration



**Ne peuvent contenir que du contenu phrasé.** Voir plus haut.

`<code>`

- Code-source.

`<kbd>`

- Ligne de commande.

`<mark>`

- Surlignement.

## Balises de petite structuration



**Ne peuvent contenir que du contenu phrasé.** Voir plus haut.

**<sub>**

- **Indice.**

**<sup>**

- **Exposant.**

## Balises de petite structuration



**Ne peuvent contenir que du contenu phrasé.** Voir plus haut.

**<strong>**

- **Important.** Devrait remplacer **<b>** (*Bold*=gras) dans ce cas.

**<em>**

- **Emphase.** Devrait remplacer **<i>** (*Italic*) dans ce cas.

**<del>**

- **Suppression.** Devrait remplacer **<s>** (*Strikethrough*=barré).

**<ins>**

- **Insertion.** Devrait remplacer **<u>** (*underline*=soulignement) dans ce cas.



## Balises de petite structuration

`<br/>`

- **Saut de ligne** (*BReak line*). Uniquement dans un poème, une adresse...

`<hr/>`

- **Séparation** (*Horizantal Rule*). De plus en plus rare.

## Balises de petite structuration

`<button/>`

- **Bouton**. Aucune action : programmation javascript obligatoire.
- Attributs :
  - **value="texte"**  
Texte affiché sur le bouton.

`<progress>`

- **Barre de progression.**
- Ne peut contenir que du contenu phrasé. Voir plus haut.
- Exemple :

```
<progress value="70" max="100"></progress>
```



Essentiellement utilisée en programmation javascript. Le contenu ne s'affiche que dans les navigateurs ne supportant pas cet élément.

## Balises d'accrochage CSS

`<div>`

- Conteneur sans sémantique de type *block*.

`<span>`

- Conteneur sans sémantique de type *inline* (*span=portée*).



**Ne peut contenir que du contenu phrasé.** Voir plus haut.



Tous deux sont destinés à l'accrochage de règles CSS (*CSS hooks*).



Les modes *block* et *inline* sont abordés dans le cours CSS.

## Lexiques

`<dl>`, `<dt>`, `<dd>`

- **Lexique** (*Definition List, Term, Description*). Exemple :

```
<dl>
  <dt>Entropie</dt>
  <dd>
    Fonction d'état (J/K) correspondant au degré de
    désordre d'un système thermodynamique au niveau
    moléculaire.
  </dd>
  <dt>Enthalpie</dt>
  <dd>
    Fonction d'état (J) donnant l'énergie totale
    d'un système thermodynamique.
  </dd>
</dl>
```

## Listes

**<ul>, <ol>, <li>**

- **Liste** (*Unordered List, Ordered List, List Item*). Exemple :

```
<ul>
  <li>Physique</li>
  <li>Mathématiques</li>
</ul>
<ol>
  <li>Historique
    <ol>
      <li>Avant 1989</li>
      <li>Après 1989</li>
    </ol>
  </li>
</ol>
```

- Dans une <ol>, <li> peut prendre un attribut `value` recevant un entier correspondant au rang souhaité dans la liste.

## Tableaux

**<table>, <tr>, <td>**

- **Tableaux** (tr=*table row*, td=*table data*). Exemple :

```
<table>
  <tr>
    <td>1</td>
    <td>2</td>
    <td>3</td>
  </tr>
  <tr>
    <td>4</td>
    <td>5</td>
    <td>6</td>
  </tr>
</table>
```

## Tableaux

### <table>, <tr>, <td> (suite)

- Attributs de <td> pour les fusions de cellules :
  - **colspan="nbDeCellules", rowspan="nbDeCellules"**  
Fusion de colonnes ou de rangées.
- Exemple :

```
<table>
  <tr>
    <td colspan="2">1+2</td>
    <td rowspan="2">3+6</td>
  </tr>
  <tr>
    <td>4</td>
    <td>5</td>
  </tr>
</table>
```

## Tableaux

### <th>, <caption>

- **En-tête** (th=*table header*) **et légende de tableaux.** Exemple:

```
<table>
  <caption>Exemple</caption>
  <tr>
    <th>Col1</th>
    <th>Col2</th>
  </tr>
  <tr>
    <td>A</td>
    <td>B</td>
  </tr>
</table>
```



La balise <th> possède aussi les attributs rowspan et colspan.

## Tableaux

### <thead>, <tbody>, <tfoot>

- **En-tête, corps et pied de tableaux.**
- Les 3 parties peuvent être codées dans le désordre. Exemple:

```
<table>
  <caption>Exemple</caption>
  <tbody>
    <tr><td>A</td></tr>
  </tbody>
  <tfoot>
    <tr><td>Pied</td></tr>
  </tfoot>
  <thead>
    <tr><th>En-tête</th></tr>
  </thead>
</table>
```

## Liens

### <a>

- **Lien** (*Anchor*=ancre).
- Attributs :
  - **href="URL"**  
URL de destination.
  - **target="contexte"**  
Nom d'un contexte existant (fenêtre ou onglet) ou :
    - `_blank` : nouveau contexte.
    - `_parent` : contexte parent (si présent).
    - `_top` : contexte racine (si présent).
  - **media="mediaQuery"**  
Les *media-query* sont abordées dans le cours CSS.
  - **name="#ancree"**  
Nom d'une ancre locale.  
L'attribut `href` doit être absent.

## Liens

### <a> (suite)

- Exemples :

```
<a href="#bas">Bas</a>
```

```
<a href="//www.leica-camera.com">Leica</a>
```

```
<a href="//www.nikon.fr" target="_blank">Nikon</a>
```

```
<a href="mailto:gillesvds@adok.info">Gilles VDS</a>
```

```
<a href="#top">Haut</a>
```

```
<a id="bas"/>
```

- Pour les liens email voir aussi :  
[//developer.mozilla.org/en-US/docs/Web/Guide/HTML/Email\\_links](http://developer.mozilla.org/en-US/docs/Web/Guide/HTML/Email_links)

## Images

### <img/>

- **Image.**

- Attributs :

- **src="source"**  
Source de l'image (JPEG, GIF, PNG, BMP, SVG).
- **alt="texteAlternatif"**  
Texte alternatif.
- **width="largeur", height="hauteur"**  
Largeur et hauteur en pixels ou en pourcentage du parent.
- **ismap="ismap"**  
Envoie les coordonnées x, y du clic. Obligatoirement dans un <a>.
- **usemap="#carte"**  
Nom de la carte associée. Voir diapo suivante.



Les règles CSS définissant la largeur et la hauteur ont priorité sur les valeurs des attributs width et height.

## Images

### <img/> (suite)

- Exemples :

```

```

```
<a href="//www.adok.info/utils/form.php">  
    
</a>
```

```

```

## Images

### <map>, <area/>

- **Carte de liens sur image.** Exemple :

```
  
<map id="rx100">  
  <area shape="circle" coords="390,100,10" href="#"  
    title="Objectif rétractable"/>  
  <area shape="circle" coords="270,30,10" href="#"  
    title="Bague de commande"/>  
  <area shape="circle" coords="100,35,10" href="#"  
    title="Ecran LCD"/>  
  <area shape="circle" coords="150,155,10" href="#"  
    title="Wifi NFC"/>  
</map>
```

## Images

### <map>, <area/> (suite)

- Les valeurs possibles de l'attribut shape de <area> sont :
  - rect (rectangle ABCD) : coords="xA,yA,xC,yC".
  - circle (cercle de centre C, de rayon R) : coords="xC,yC,R".
  - poly (polygone) : coords="xA,yA,...,xN,yN".

## Audio & Video

### <audio>, <source/>

- **Média audio.**
- Format **mp3** universellement accepté.
- Exemple :

```
<audio controls="controls" autoplay="autoplay">
  <source src="orangeBlossom_yaSidi.mp3"
    type="audio/mp3"/>
</audio>
```

- Voir aussi ce tour complet de la balise <audio> :  
[//blogs.msdn.com/b/ie/archive/2011/05/13/unlocking-the-power-of-html5-lt-audio-gt.aspx](http://blogs.msdn.com/b/ie/archive/2011/05/13/unlocking-the-power-of-html5-lt-audio-gt.aspx)



La balise <audio> possède un attribut src pour se passer de <source/> mais fonctionnement incorrect si plusieurs <audio> dans la même page.



## Audio & Video

### <video>, <source/>

- **Média video.**
- Format **MPEG H.264 (audio AAC ou MP3)** universellement accepté.
- Exemple :

```
<video controls="controls" poster="leica.jpg">  
  <source src="leica.mp4" type="video/mp4"/>  
</video>
```



Les attributs `width` et `height` de `<video>` devraient toujours être renseignés (en pixels).



La balise `<video>` possède un attribut `src` pour se passer de `<source/>` mais fonctionnement incorrect si plusieurs `<video>` dans la même page.



La balise `<track/>` permet d'ajouter des sous-titres.

Voir : [//developer.mozilla.org/fr/docs/Web/HTML/Element/track](https://developer.mozilla.org/fr/docs/Web/HTML/Element/track)

## Dessin vectoriel

### <canvas/>

- **Zone de dessin vectoriel.**
- Attributs :
  - **`width="largeur"`**  
Largeur de la zone.
  - **`height="hauteur"`**  
Hauteur de la zone.
- Manipulation uniquement en JavaScript.  
Non détaillé dans ce cours. Voir :  
[//developer.mozilla.org/fr/docs/Web/Guide/Graphics/Dessiner\\_avec\\_canvas](https://developer.mozilla.org/fr/docs/Web/Guide/Graphics/Dessiner_avec_canvas)

## Formulaires

### <form>

- **Formulaire.** C'est l'enveloppe des champs de saisie et du bouton d'envoi.
- Attributs :
  - **method="get | post"**  
Méthode d'envoi. Utilisez `post` de préférence.
  - **action="URL"**  
URL de destination.
  - **target="cible"**  
Voir <a> plus haut.
  - **enctype="text/plain | multipart/form-data"**  
Encodage : `text/plain` étant la valeur par défaut, cet attribut n'est utile qu'associé à la seconde valeur lorsque le formulaire comprend des champs de type `file`. Voir plus loin.
  - **novalidate="novalidate"**  
Annule la validation automatique par le navigateur.

## Formulaires

### <input/>

- **Saisie.**
- Attributs :
  - **type="type"**  
Le type du champ : voir diapos suivantes.
  - **name="nom"**  
Obligatoire. Nom associé à la valeur transmise au serveur.
  - **value="valeurInitiale"**  
Optionnel sauf pour les types `checkbox` et `radio`.
  - **autofocus="autofocus"**  
Sur un seul élément par formulaire. Met le focus sur cet élément.  
Aucun effet sur le type `hidden`.
  - **disabled="disabled"**  
Désactive l'élément : la valeur ne sera pas envoyée.

## Formulaire

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

- **Champ de saisie mono-ligne.** C'est le type par défaut.
- Attributs particuliers :
  - **`size="longueur"`, `maxlength="longueurMaxi"`**  
Longueur (approximative) et nombre maxi de caractères entrables.
  - **`placeholder="invite"`**  
Texte d'invite (*prompt*) qui s'efface dès le début de la saisie.
  - **`autocomplete="off"`**  
Le champ ne sera autocomplété par les valeurs déjà entrées.
  - **`required="required"`**  
Une valeur est obligatoire.
  - **`readonly="readonly"`**  
La valeur par défaut n'est pas modifiable.
  - **`list="idDataList"`**  
Voir plus loin `<datalist>`.

## Formulaire

### `<input type="text"/>` avec `<datalist>`

- **Autocomplétion via une liste définie.** Exemple :

```
<input type="text" name="nav" list="browsers"/>
<datalist id="browsers">
  <option value="Chrome"></option>
  <option value="Firefox"></option>
  <option value="Internet Explorer"></option>
  <option value="Opera"></option>
  <option value="Safari"></option>
</datalist>
```

## Formulaires

```
<input type="password"/>
```

- **Champ de mot de passe.**
- Les caractères saisis ne sont pas affichés en clair.
- Attributs identiques au type `text` (`autocomplete` toujours à `off`).

## Formulaires

```
<input type="hidden"/>
```

- **Champ caché.**
- N'apparaît pas.
- Aucun attribut particulier.

## Formulaires

`<input type="checkbox"/>`

- **Case à cocher.**
- Attributs particuliers :
  - **checked="checked"**  
Case cochée par défaut.

## Formulaires

`<input type="radio"/>`

- **Bouton de radio.**
- Fonctionne en groupe (même valeur de `name`).
- Attributs particuliers :
  - **checked="checked"**  
Coché initialement (un seul bouton du groupe).

## Formulaires

`<input type="file"/>`

- **Choix d'un fichier.**
- Attributs particuliers :
  - **`accept="typeMIME"`**  
Propose (mais ne restreint pas) le choix d'un type de fichier.  
Exemples de types MIME :
    - `image/jpeg`
    - `image/*`
    - `application/pdf`

## Formulaires

`<input type="reset"/>`

- **Bouton de remise à zéro.** Remet l'ensemble des `<input>` du formulaire dans leurs états d'origine.
- Aucun attribut particulier. L'attribut `value` permet de personnaliser le texte affiché sur le bouton.

`<input type="submit"/>`

- **Bouton d'envoi.** Envoie les données du formulaire.
- Aucun attribut particulier. L'attribut `value` permet de personnaliser le texte affiché sur le bouton.

`<input type="button"/>`

- **Bouton sans action.** Programmation javascript obligatoire.
- Aucun attribut particulier. L'attribut `value` permet de personnaliser le texte affiché sur le bouton.

## Formulaires

`<input type="image"/>`

- **Bouton d'envoi sous forme d'image.** Les coordonnées  $x, y$  du clic sont envoyées au serveur.
- Attributs particuliers :
  - **`src="URL"`**  
URL de l'image.
  - **`value="texteAlternatif"`**  
Texte alternatif.
  - **`width="largeur" height="hauteur"`**  
Largeur et hauteur en pixels ou en pourcentage du parent.  
L'homothétie n'est pas automatiquement conservée !

## Formulaires

`<input type="email"/>`

- **Champ de saisie d'email.** Sera vérifié.
- Attributs identiques au type `text`.

`<input type="url"/>`

- **Champ de saisie d'URL.** Sera vérifié.
- Attributs identiques au type `text`.

## TD : formulaire

- Consultez la documentation puis testez le support par les derniers navigateurs des types suivants :
  - `date` (calendrier)
  - `color` (*color picker*)
  - `range` (curseur)

et de la balise : `<output>`.

Solution : formulaire.htm

## Formulaires

### `<textarea>`

- **Champ de saisie multi-lignes.**
- Attributs identiques au type `text` et :
  - `cols="largeur"`  
Largeur en nombre de caractères.
  - `rows="hauteur"`  
Hauteur en nombre de lignes.
- Exemple :

```
Message<br/>
<textarea name="message" cols="30" rows="5">
  Texte initial.
</textarea>
```



## Formulaires

### <select>, <option>, <optgroup>

- **Menu déroulant** (*drop down list*). Exemple :

```
Courant
<select name="courant">
  <optgroup label="Continu">
    <option value="12">12 volts</option>
    <option value="24">24 volts</option>
  </optgroup>
  <optgroup label="Alternatif">
    <option value="110">110 volts</option>
    <option value="220"
      selected="selected">220 volts</option>
  </optgroup>
</select>
```



Les balises <optgroup> et <option> peuvent recevoir l'attribut disabled="disabled".

## Formulaires

### <fieldset>, <legend>

- Regroupement d'éléments de formulaires. Exemple :

```
<fieldset>
  <legend>Vous</legend>
  Nom <input name="nom"/><br/>
  Sexe
  <input type="radio" name="sexe" checked="checked"/>M
  <input type="radio" name="sexe"/>F
</fieldset>
```

## Formulaires

### `<label>`

- **Etiquette d'un élément de formulaire.** Exemples :

```
<label>Nom <input name="nom"></label>
```

```
<label for="nom">Nom</label>
```

```
<input id="nom" name="nom"/>
```

## Frames

### `<iframe/>`

- **Frame** (*inline frame*).
- Attributs :
  - **src="URL"**  
URL du contenu.
  - **srcdoc="chaîneHTML"**  
Contenu direct.
  - **width="largeur" height="hauteur"**  
Largeur et hauteur en pixels ou en pourcentage du parent.
  - **name="contexte"**  
Contexte utilisable comme cible dans un attribut `target` de lien.

## Attributs globaux

**`id="identifiant"`**

- Identifiant unique de l'élément.
- Accroche CSS via le sélecteur #.
- Manipulation JavaScript via `document.getElementById()`.

**`class="classeCSS"`**

- Accroche CSS via le sélecteur . (point).
- Manipulation JavaScript via `document.getElementsByClassName()`.

**`style="règlesCSS"`**

- Règles CSS directement dans l'élément (*inline CSS*).

## Attributs globaux

**`hidden="hidden"`**

- Masque l'élément.
- A n'utiliser que pour des contenus en attente de publication ou obsolètes. Sinon, utiliser la règle CSS `display:none`.

**`draggable="draggable"`**

- Rend *drag-and-droppable* l'élément.

**`title="titre"`**

- Titre associé à l'élément (infobulle).
- Signification plus riche pour `<abbr>` : voir plus haut cette balise.

## Attributs globaux

**`lang="langue"`**

- Langue de l'élément.
- Liste des abbréviations :  
[//fr.wikipedia.org/wiki/Liste\\_des\\_codes\\_ISO\\_639-1](https://fr.wikipedia.org/wiki/Liste_des_codes_ISO_639-1)

**`dir="ltr | rtl"`**

- Direction de la lecture pour l'élément.

**`tabindex="indice"`**

- Ordre de parcours des éléments via la touche *tab* du clavier.

## Attributs globaux

**`data-*= "valeur"`**

- Donnée personnalisée associée à l'élément (*custom data attribute*).
- Le nom de la donnée ne doit pas contenir de majuscules.
- Manipulation JavaScript via
  - `element.getAttribute()`
  - `element.dataset`
- Exemple :

```
<div id="article" data-taille="XXL"
                        data-stock="3">
    ...
</div>
```