

## Chapitre 2 : langage HTML

# Plan

---

I. Introduction

II. HTML5: Éléments de base

III. Liens hypertextes et multimédias

IV. Formulaires

# I. Introduction

## 1. HTML: définition

---

- ▶ HTML (HyperText Markup Language) est un langage de balisage(Markup) utilisé pour créer des pages web.
- ▶ Il permet de **définir**, **structurer** et d'**organiser** le contenu de la page Web.
- ▶ En utilisant des mots de code balisés (<>), HTML permet de définir divers éléments tels que les *titres*, les *paragraphes*, les *images*, les *liens*, etc.
- ▶ Les navigateurs web interprètent ensuite ce code HTML pour afficher le contenu et la mise en page du document.
- ▶ HTML est un langage universel et standardisé qui est incontournable pour créer des sites web. Il est souvent utilisé en combinaison avec d'autres technologies telles que CSS (Cascading Style Sheets) pour **définir la présentation visuelle de la page**, et JavaScript pour ajouter des fonctionnalités interactives à la page.
- ▶ La dernière version d'HTML est HTML5, qui apporte de nouvelles fonctionnalités et améliorations par rapport aux versions précédentes, en particulier en ce qui concerne l'ajout de sémantique aux pages Web.

## 2. HTML: un langage de balisage

---

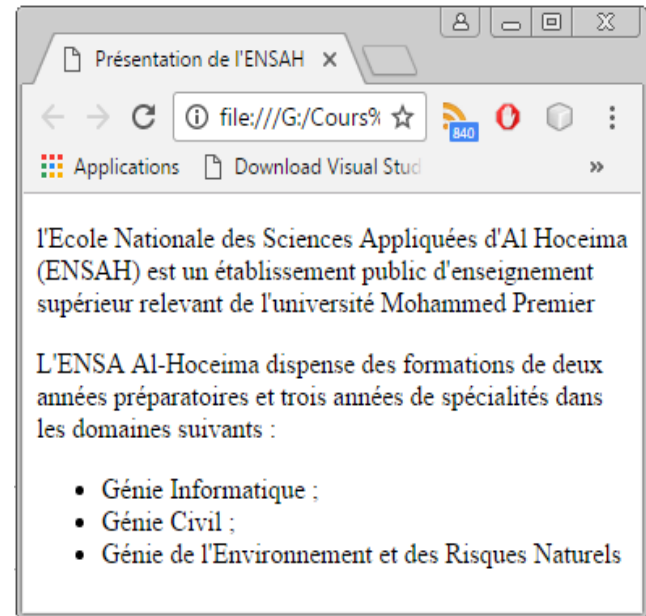
- ▶ Une balise est un mot clé, une commande du langage HTML insérée dans le corps du document pour introduire un effet particulier (Exemples : début de mise en gras, fin de mise en gras, début de tableau...).
- ▶ Une balise commence toujours par un signe "<" et se finit toujours par un signe ">". La plupart des balises doivent être ouvertes et fermées (les balises en paires) pour délimiter leur zone d'influence. La balise fermante contient la même commande que la balise ouvrante, mais précédée d'un caractère /.
- ▶ Exemple : pour mettre un texte en gras en HTML, on utilise la commande **b**. Cet élément a une balise ouvrante <b> et une balise fermante </b>. Par exemple, pour mettre le mot "ENSAH" en gras dans la phrase "Je suis étudiant à l'ENSAH.", on peut utiliser la balise <b> comme ceci: "Je suis étudiant à l'<b>ENSAH</b>." Ce code HTML fera que le mot "ENSAH" sera affiché en gras à l'écran.

.... la suite

- Pour décrire un hypertexte, le langage HTML insère des **balises** dans le texte du document; un fichier portant l'extension « .html »

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title>Présentation de l'ENSAH</title>
</head>
<body>
<p>
L'Ecole Nationale des Sciences Appliquées d'Al Hoceima (ENSAH)
est un établissement public d'enseignement supérieur
relevant de l'université Mohammed Premier
</p>
<p>
L'ENSA Al-Hoceima dispense des formations de deux années préparatoires
et trois années de spécialités dans les domaines suivants :
</p>
<ul>
<li>Génie Informatique ;</li>
<li>Génie Civil ;</li>
<li>Génie de l'Environnement et des Risques Naturels</li>
</ul>
</body>
</html>
```

Code HTML



Son interprétation par le navigateur

- **NB:** L'édition du code HTML peut être réalisée en utilisant un éditeur de texte simple tel que Bloc-notes. Cependant, il est recommandé d'utiliser un éditeur dédié qui connaît déjà le code HTML, tel que Sublime Text, qui offre des fonctionnalités supplémentaires pour faciliter l'édition du code HTML.

### 3. Eléments et attributs

---

- ▶ HTML contient un ensemble **d'éléments** délimités par des **balises**. Ces éléments **permettent de définir** des **liens hypertextes**, des **titres**, des **paragraphes**, des **listes**, des **tableaux**, des **images**, etc....**Exemple** : l'élément `<img>` permet d'insérer une image dans le document.
- ▶ Un élément peut comporter zéro ou plusieurs **attributs**, choisis parmi un ensemble spécifique à cet élément ou un ensemble commun à tous les éléments HTML.
- ▶ **Les attributs modifient les propriétés des balises html**. Ils permettent en quelque sorte de **spécifier l'action de l'élément**. Ils se placent après le nom de la balise ouvrante et le plus souvent une valeur, comme ceci : Syntaxe : `<element attribut="valeur">`
- ▶ Dans **certains cas**, ils sont **essentiels**, tels que l'attribut `src` pour `<img>`, précisant le nom du fichier image à afficher dans la page. Exemple : ``
- ▶ Dans **d'autres cas**, ils peuvent être **facultatifs**, tels que `class` ou `id`, qui permettent de nommer les éléments pour interagir avec ceux-ci, ou pour leur affecter des propriétés de style. Exemple : `<p id="introduction">`.

## 4. Bien éditer son document

---

- ▶ Lors de l'écriture des balises en HTML, voici quelques règles simples qu'il vaut mieux respecter :
  - ▶ HTML5 permet le mélange des lettres majuscules et minuscules dans les noms d'éléments et leurs attributs, *mais il est recommandé d'utiliser le minuscule*.
  - ▶ **Fermeture des balises**: en HTML5 il est possible de ne pas fermer tous les éléments (par exemple l'élément `<p>`), *mais même si, il est recommandé de les fermer* pour **délimiter la zone d'influence**,
  - ▶ Pour les éléments vides comme `<meta charset="UTF-8">`, leur fermeture est optionnelle.
  - ▶ Les balises doivent être correctement imbriquées.
    - ▶ correct : `<a><b><c>contenu</c></b></a>`.
    - ▶ incorrect : `<a><b><c>contenu</a></c></b>`.
  - ▶ Il est de préférence de mettre **les valeurs des attributs entre des guillemets**.
  - ▶ Ajout des commentaires : il vaut mieux commenter le code html; cela se fait en utilisant la syntaxe suivante: `<!-- commentaire -->`

## II. HTML5: Éléments de base

### 1. Une version récente : HTML5

---

- ▶ HTML5 est la dernière évolution des standards qui définissent HTML. Il s'agit de la **nouvelle version** du *langage* HTML **présentant des révisions majeures** par rapport aux versions antérieures et qui se focalisaient juste sur le contenu des pages Web.
- ▶ De **nouveaux éléments**, **attributs** et **comportements** ont été rajoutés pour de nouveaux usages, afin de répondre aux besoins grandissants des internautes.
- ▶ HTML5 a introduit de nouvelles balises et attributs pour améliorer la **sémantique du code HTML** et permettre une meilleure compréhension de la structure de la page Web. Ces balises incluent <header>, <nav>, <section>, <article>, <footer>, <aside>, etc.
- ▶ Egalement, l'HTML 5 offre la **possibilité de l'ajout de média riches** (vidéo, audio, SVG) à une page Web, sans avoir besoin de plugins tiers tels que Flash. Les balises <audio> et <vidéo> permettent d'intégrer facilement des fichiers audio et vidéo à une page.
- ▶ HTML5 apporte aussi des fonctionnalités permettant le développement d'applications web interactives, à savoir :
  - ▶ une **API de drag and drop** en combinaison avec l'**attribut draggable**
  - ▶ une **API d'édition** en combinaison avec le nouvel **attribut contenteditable**.
  - ▶ une **API de dessin 2D** utilisé avec la nouvelle **balise canvas**



## 2. Document html5: la structure générale

---

`<!DOCTYPE html>`

`<!-- spécifier qu'il s'agit d'un document html5. Il faut la préciser au début de tout document en HTML car cela permet au navigateur ne supportant pas encore le HTML5 de passer en mode standard et ainsi d'ignorer les balises inconnues -->`

`<html>`

`<!-- C'est la balise principale du code. Elle englobe tout le contenu de la page-->`

`<head>`

`<title> titre de la page </title>`

`<meta charset="type_encodage" />`

`<!-- c'est l'en-tête du document. Cette section donne quelques informations générales sur la page comme: Le titre du document (balise <title> ... </title>), l'encodage pour la gestion des caractères spéciaux, des déclarations ou des appels de feuilles de style CSS, des fonctions ou des liens vers des fichiers JavaScript -->`

`</head>`

`<body>`

`<!-- C'est le corps du document. Elle contient la partie principale de la page. C'est cette partie du document HTML qui sera affichée dans la fenêtre du navigateur et donc visible par l'internaute -->`

`</body>`

`</html>`

### 3. L'entête <head>

---

- ▶ La section <head> d'une page HTML est utilisée pour inclure des **informations sur la page elle-même**, telles que le **titre de la page**, les **feuilles de style CSS**, les **scripts JavaScript**, les **balises meta** et les icônes de site. Voici quelques éléments couramment placés dans la section <head> :
  - ▶ La balise <meta > : sont utilisées pour fournir des informations sur la page aux navigateurs et aux moteurs de recherche, telles que l'encodage des caractères, la description et les mots-clés. Elles aident à mieux comprendre le contenu de la page.
  - ▶ La balise <title> : définit le titre de la page qui est affiché dans l'onglet du navigateur et dans les résultats de recherche.
  - ▶ Les balises <link> : sont utilisées pour lier la page web avec le CSS et avec une icône de site (une image représentative de la page dans les onglets de navigateur et les signets).
  - ▶ La balise <script >: est utilisée pour lier la page web avec un fichier JavaScript.

## 4. La balise <meta /> : cas d'utilisation

---

- ▶ Les balises <meta> en HTML sont utilisées pour fournir des métadonnées sur un document HTML. Ces métadonnées n'apparaissent pas directement sur la page web mais fournissent des informations sur celle-ci.
  - ▶ `<meta charset="valeur">` : permet de définir l'encodage des caractères c-à-d comment les caractères spéciaux vont s'interpréter et s'afficher comme: les caractères accentués, caractères arabes, caractères Tifinagh, Idéogrammes chinois et japonais, etc. La valeur `UTF-8` est la méthode d'encodage la plus couramment utilisée aujourd'hui, elle couvre pratiquement tous les symboles de toutes les langues.
  - ▶ `<meta name="keywords" content="Mot-clé1, Mot-clé2, Mot-clé3">` : définir des mots-clés qui caractérisant la page vis-à-vis des moteurs recherche.
  - ▶ `<meta name="description" content="Description ...">`: description du contenu de la page (utile pour les moteurs de recherche)
  - ▶ `<meta name="author" content="Nom de l'auteur">`: définir l'auteur de la page.

## 5. Structuration de la page

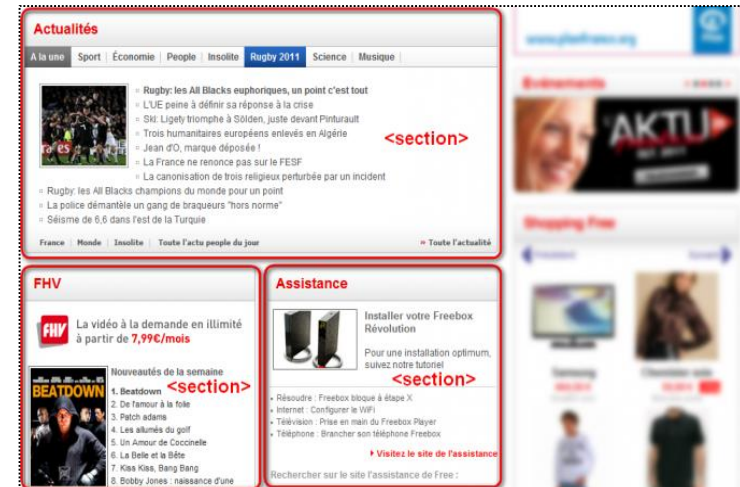
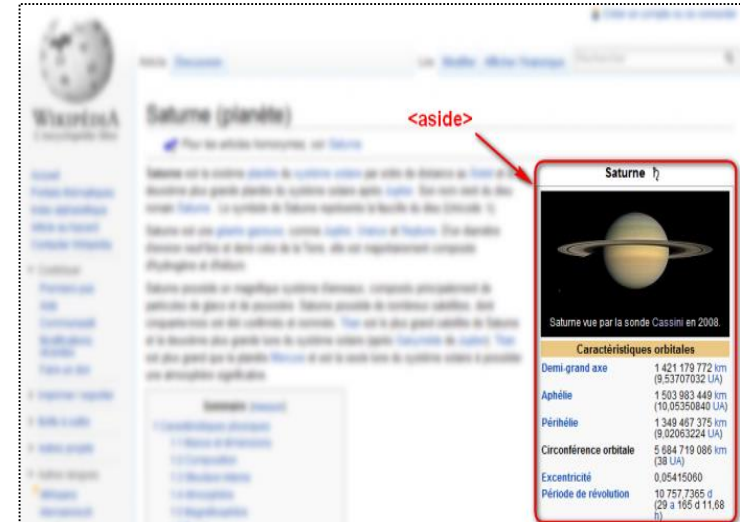
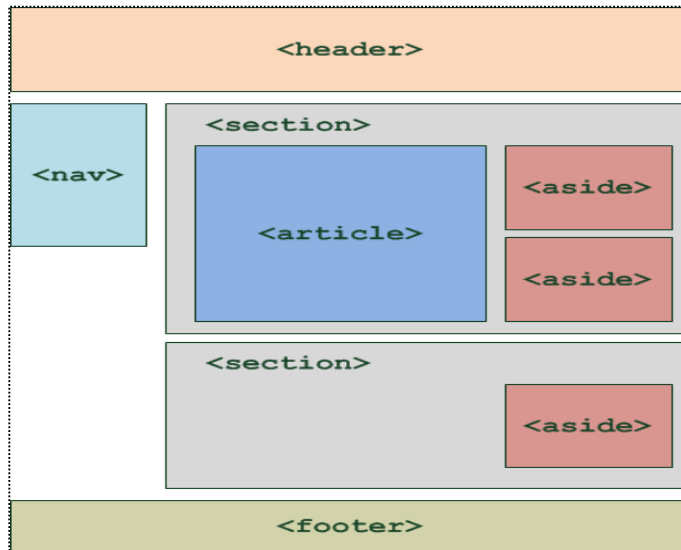
---

- ▶ La structure d'une page Web, à savoir la structure sémantique de ce qui est entre `<body>` et `</body>`, est fondamentale pour la présentation de la page à l'utilisateur.
- ▶ HTML 5 propose un ensemble d'éléments pour cette structuration à savoir : `<article>`, `<header>`, `<section>`, `<footer>`, `<nav>`, `<aside>`, `<main>`.
- ▶ Voici un exemple d'un schéma de structuration : un entête, un pied de page et le contenu principal.

```
<body>
  <header> <!-- Contenu de l'en-tête de la page --></header>
  <nav> <!-- Menu de navigation de la page --></nav>
  <main>
    <!-- Contenu principal de la page -->
    <article>
      <section>
        <h1>Titre de la Page</h1>
        <p>Ceci est le contenu principal de la page.</p>
      </section>
    </article>
  </main>
  <footer><!-- Pied de page de la page --></footer>
</body>
```

.... la suite

## ► D'autres exemples:



- ▶ Ces balises décrivent la nature de certains éléments du contenu d'une page. Elles n'ont aucune influence sur la structure du contenu et bien souvent aucune influence sur la présentation et l'affichage de la page.
- ▶ Elles permettent seulement d'expliquer à l'ordinateur « Ceci est l'en-tête », « Ceci est le pied de page », etc. Elles n'indiquent pas, contrairement à ce qu'on pourrait penser, où doit être placé le contenu. C'est le rôle du CSS.
- ▶ Ce sont des balises dites sémantiques, leur but c'est de permettre à des programmes automates (Exemple: robots d'indexation) de mieux comprendre la structure d'une page web et par conséquent de faciliter son indexation.
- ▶ Le Web sémantique est encore en développement et l'impact d'un bon contenu structuré dans des balises sémantiques adaptées n'est pas encore bien senti auprès des moteurs de recherche, mais il est préférable d'adopter cette bonne pratique dès maintenant.
- ▶ Aujourd'hui, les balises HTML5 sont largement utilisées par les développeurs pour structurer leurs pages Web, mais certains continuent d'utiliser des balises génériques telles que <div>. En effet, avant l'arrivée de ces nouvelles balises HTML5, la balise <div> était souvent utilisée pour diviser une page en sections.

.... la suite

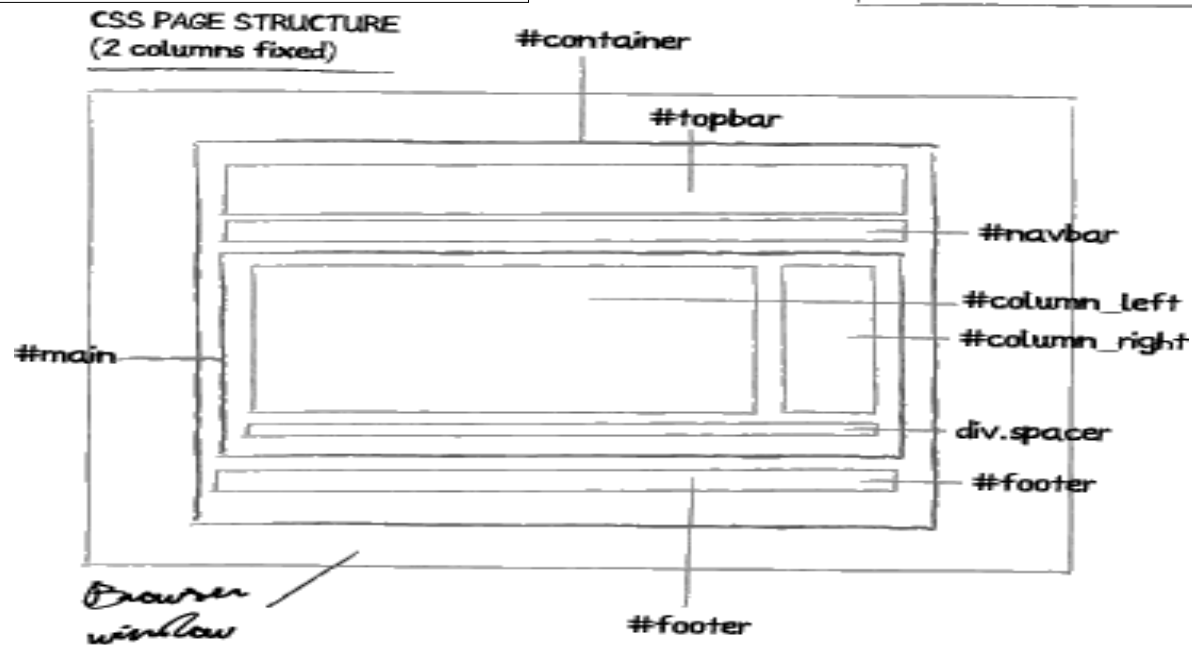
## ► Organisation par des `<div>` :

```
<div id="navbar"> ...navbar content... </div>
<div id="main">
  <div id="column_left">
    ...column left content...
  </div>
  <div id="column_right">
    ...column right content...
  </div>
  <div class="spacer"> </div>
</div> <!-- close #main content -->
</div> <!-- close #container -->
```

INDEX.PHP HTML  
DESIGN

*to force #container  
light*

```
<body>
<div id="container">
  <div id="topbar"> ... </div>
  <div id="navbar"> .. </div>
  <div id="main">
    <div id="column_left"> ... </div>
    <div id="column_right"> .. </div>
    <div class="spacer"/>
  </div>
</div>
</body>
```



## 6. Balises de forme et de structure

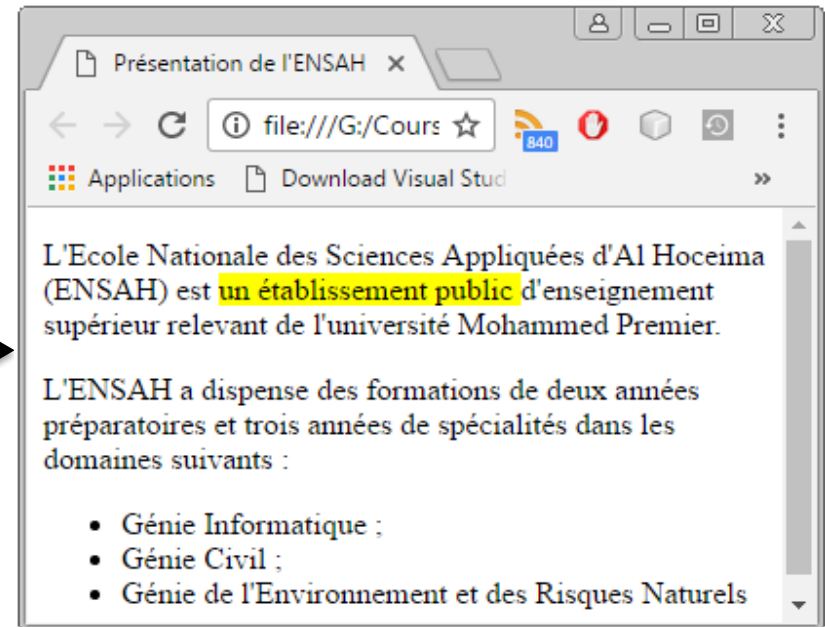
- ▶ De nombreuses balises HTML sont dédiées à "la mise en forme" et l'organisation du texte du document. Certaines existent depuis la toute première version de HTML, d'autres ont été introduites plus récemment dans HTML5. Voici une liste non exhaustive :
  - ▶ `<p>` : définit un paragraphe contenant une ou plusieurs phrases.
  - ▶ `<ul>` et `<ol>` : sont utilisés pour spécifier une liste d'éléments ( `<li>` ). `<ul>` pour une liste à puces, `<ol>` pour une liste numérotés.
  - ▶ `<br>` : permet de créer un saut de ligne.
  - ▶ `<h1>` à `<h6>` : utilisées pour la hiérarchisation des titres.
  - ▶ `<strong>` : utilisée dans un paragraphe pour mettre une partie de texte en gras.
  - ▶ `<u>` : pour souligner un texte.
  - ▶ `<del>` : définit un texte qui a été supprimé (exemple: ~~texte~~ )
  - ▶ `<mark>` : utilisée pour marquer une partie du texte en jaune.
  - ▶ `<hr/>` : utilisée pour la mise en place d'une ligne de séparation horizontale
  - ▶ `<cite>` : utilisée pour le titre d'une œuvre, d'un document ou d'un évènement.
  - ▶ `<i>` : utilisée dans un paragraphe pour indiquer qu'une partie du texte est différent du reste. Il ne faut pas utiliser cette balise lorsque `<strong>`, `<b>`, `<cite>`, `<dfn>`, `<em>`, `<q>` ou `<small>` peuvent être utilisées.
  - ▶ `<div>` : balise sans valeur sémantique réelle, elle sert de conteneur pour une mise en forme en CSS. Cette balise doit être utilisée en dernier recours, lorsqu'aucune autre balise ne peut convenir.



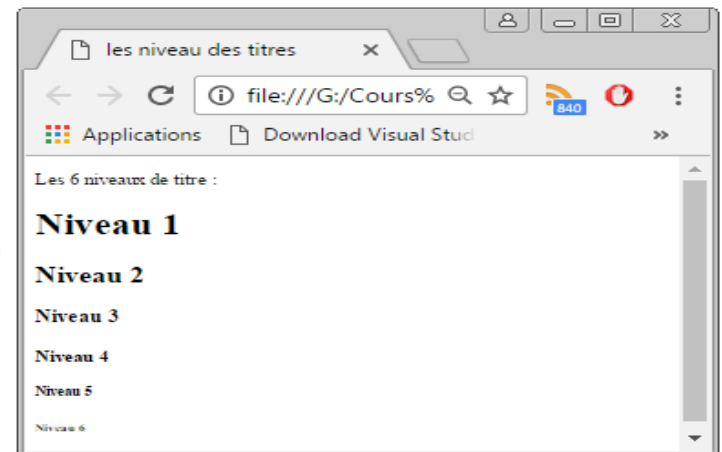
.... la suite

## Exemples :

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title> Présentation de l'ENSAH</title>
</head>
<body>
<p>
L'Ecole Nationale des Sciences Appliquées
d'Al Hoceima <abbr> (ENSAH) </abbr> est
<mark> un établissement public </mark>
d'enseignement supérieur relevant
de l'université Mohammed Premier.
</p>
<p>
L'ENSAH a dispense des formations de deux
années préparatoires et trois années de
spécialités dans les domaines suivants :
</p>
<ul>
  <li> Génie Informatique ; </li>
  <li>Génie Civil ;</li>
  <li>Génie de l'Environnement </li>
</ul>
</body>
</html>
```



```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title> les niveau des titres</title>
</head>
<body>
<p> Les 6 niveaux de titre : </p>
<h1>Niveau 1</h1>
<h2>Niveau 2</h2>
<h3>Niveau 3</h3>
<h4>Niveau 4</h4>
<h5>Niveau 5</h5>
<h6>Niveau 6</h6>
</body>
</html>
```



## 6. Balise <table> : définir des tableaux

- ▶ Un tableau HTML est défini avec la balise <table>.
- ▶ Chaque ligne de la table est définie avec la balise <tr>. Un en-tête de table est défini avec la balise <th> . Par défaut, les titres des tableaux sont en gras et centré. Les cellules sont définies avec la balise <td>.

Nom	Prénom	Age
Jill	Smith	50
Eve	Jackson	20

- ▶ Il est possible de fusionner des cellules horizontalement avec l'attribut `colspan` ou verticalement avec `rowspan`. Il faut indiquer combien de cellules doivent être fusionnées.
- ▶ A noter que la balise <table> ne comporte plus d'attributs en HTML5 comme c'était le cas avec l'ancienne version.

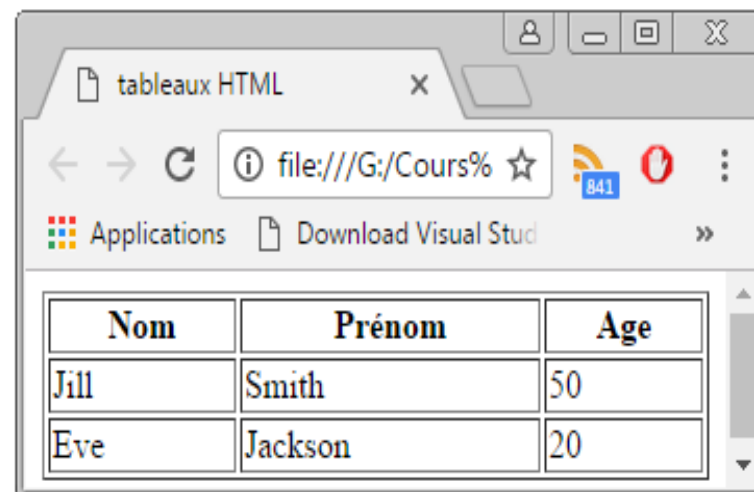
.... la suite

## ► Exemples :

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

<table style="width:100%">
  <tr> <th>Nom</th> <th>Prénom</th> <th>Age</th> </tr>
  <tr> <td>Jill</td> <td>Smith</td> <td>50</td> </tr>
  <tr> <td>Eve</td> <td>Jackson</td> <td>20</td></tr>
</table>

</body>
</html>
```

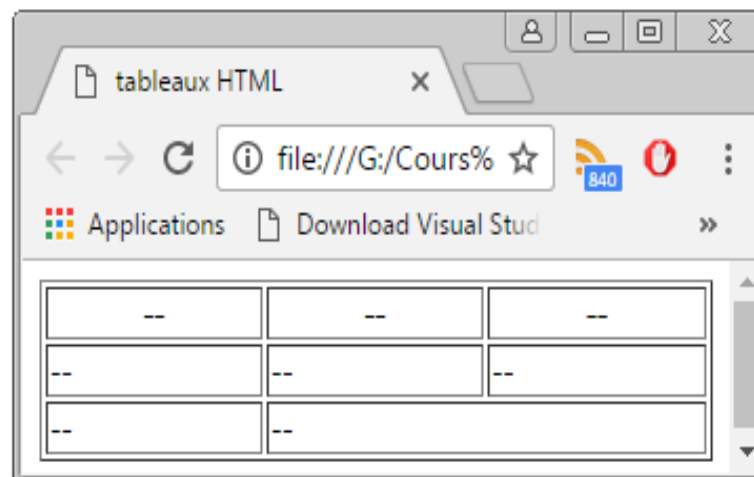


Nom	Prénom	Age
Jill	Smith	50
Eve	Jackson	20

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

<table style="width:100%">
  <tr> <th> -- </th> <th> -- </th> <th> -- </th> </tr>
  <tr> <td> -- </td> <td> -- </td> <td> -- </td> </tr>
  <tr> <td> -- </td> <td colspan="2"> -- </td> </tr>
</table>

</body>
</html>
```



--	--	--
--	--	--
--	--	

# III. Liens hypertextes et multimédias

## 1. Manipulation des liens

---

- ▶ Dans le cas général, un site web est composé de plusieurs pages. L'utilisateur peut passer en un clic d'une page à une autre grâce à des liens hypertextes.
- ▶ Ces liens sont utilisés généralement pour aller vers :
  - ▶ Une autre page du site.
  - ▶ Un endroit de la page en cours.
  - ▶ Un endroit dans une autre page du site.
  - ▶ Une page d'un autre site situé sur le Web.
  - ▶ Une adresse électronique.
  - ▶ Un fichier à télécharger.
- ▶ En HTML, les liens sont définis avec l'élément `<a>` et son attribut `href` dont la syntaxe est : `<a href="url" target=" " type=" " > link text </a>`
  - ▶ `href` est l'attribut permettant de spécifier l'adresse de destination (du lien).
  - ▶ `link text` : le **texte du lien**, c'est la partie visible (la partie cliquable). En cliquant sur pour aller à l'adresse indiquée.

► D'autres attributs de l'élément <a> :

- L'attribut **target** : spécifie l'endroit où peut s'ouvrir le document lié. Sa syntaxe est: `<a target="_blank|_self|_parent|_top|framename">`. On n'utilise plus que **\_blank**(nouvel onglet/fenêtre). Les autres valeurs sont presque obsolètes. Elles étaient utilisés avec les "frameset", un élément abandonné en HTML5.
- L'attribut **type**: indique au navigateur le type de la cible si celui-ci n'est pas un document HTML, par exemple un fichier son ou une image.
- Points d'ancrage internes: sont utilisées pour permettre aux lecteurs de sauter à des parties spécifiques d'une page Web. Elles peuvent être utiles si la page Web est très longue. Deux étapes pour faire une ancre :
  - *Création d'un point d'ancrage* : `<h2 id="mon_ancree">Titre </h2>`
  - *Ajout d'un lien vers* : il suffit de créer un lien, mais cette fois l'attribut **href** contiendra un dièse (#) suivi du nom de point d'ancrage `<a href="#mon_ancree">Aller vers l'ancree</a>`

## 2. Insertion des images et des figures

---

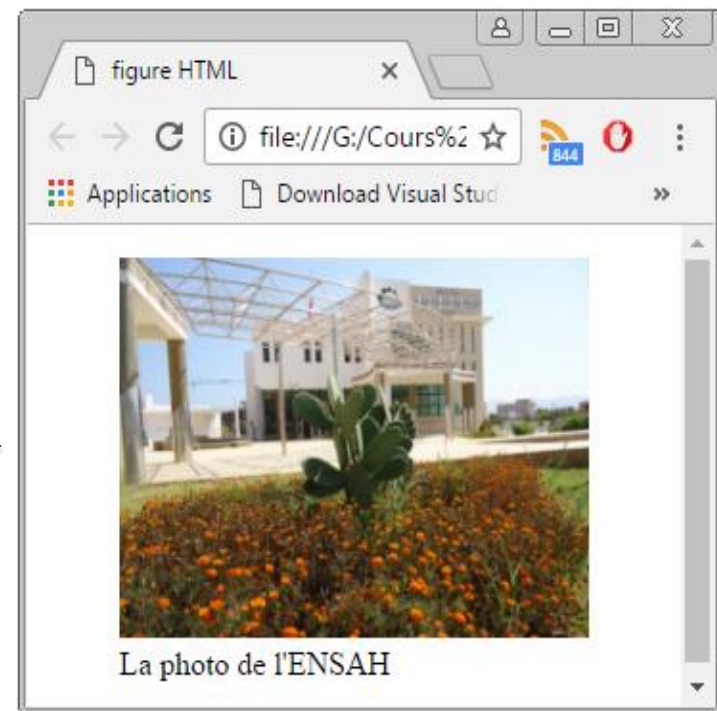
- ▶ En HTML, les images sont définies avec la balise `<img>`.
- ▶ La balise `<img>` est vide, il contient des attributs uniquement, et ne dispose pas d'une balise de fermeture.
- ▶ Sa syntaxe : ``
  - ▶ L'attribut `src` spécifie l'URL de l'image.
  - ▶ L'attribut `alt` fournit un texte alternatif pour une image, si l'image n'est pas affichée pour une raison quelconque.
- ▶ Un exemple :  
`<img src= "ensah.jpg" alt= "Photo de ENSAH" style="float:right;width:42px;height:42px;border:0; "/>`

.... la suite

- ▶ La mise en forme d'une figure est réalisée grâce à l'élément `<figure>`.
- ▶ Cet élément est utilisé pour regrouper un ou plusieurs médias ( illustrations, diagrammes, exemples de code, photos...)
- ▶ Pour ajouter une légende (un titre) à une figure, on utilise l'élément `<figcaption>`.
- ▶ Exemple :

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

<figcaption>La photo de l'ENSAH </figcaption>
</figure>
</body>
</html>
```



### 3. Insertion de l'Audio et de la vidéo

---

- ▶ En plus de la possibilité de l'insertion des images dans un document html, HTML5 donne la possibilité d'insérer de l'audio et la vidéo.
- ▶ Avant l'arrivée de HTML5, les fichiers audio/vidéo ne pouvaient être lus dans un navigateur qu'en utilisant un plugin, tel que Flash.
- ▶ Pour l'Audio, HTML5 propose l'élément `<audio>`. Exemple d'utilisation :

```
<audio controls>
```

```
<source src="test.mp3" type="audio/mpeg">
```

Ce navigateur ne supporte pas ce type de fichier.

```
</audio>
```

- ▶ Pour la vidéo, HTML5 propose l'élément `<video>`. Exemple d'utilisation :

```
<video width="320" height="240" controls>
```

```
<source src="movie.mp4" type="video/mp4">
```

Ce navigateur ne supporte pas ce type de fichier.

```
</video>
```



.... la suite

- ▶ L'attribut '`controls`' ajoute des éléments de contrôle tels que : lecture, pause et le volume.
- ▶ L'élément `<source>` permet de spécifier des fichiers audio alternatives que le navigateur peut choisir. Par défaut, il sélectionne le premier format reconnu.
- ▶ Le texte entre les balises `<audio>` et `</ audio>` ou `<video>` et `</ video>` ne sera affiché que lorsque le navigateur ne prend pas en charge l'un des éléments.
- ▶ Plusieurs formats sont supportés pour chacun des deux éléments :
  - ▶ Pour `<audio>` : .mp3, .wav, .ogg, .aac, .m4a.
  - ▶ Pour `<video>` : .mp4, .m4v, .webm, .ogv.

# IV. Formulaires

## 1. Présentation

- ▶ Les formulaires occupent une place prépondérante dans la conception et l'exploitation d'une application ou un site Web. Il suffit de penser à tous les sites à vocation commerciale pour lesquels ces formulaires sont indispensables.
- ▶ C'est en effet la seule façon de recevoir des informations provenant directement de l'utilisateur final, et en outre, structurées selon les besoins du concepteur.

The form is divided into several sections by horizontal lines. The first section contains: 'Civilité\*' with radio buttons for 'Madame', 'Mademoiselle', and 'Monsieur'; 'Nom\*' and 'Prénom(s)\*' with text input fields; 'Date de naissance\*' with three dropdown menus for day, month, and year; and 'Pays de naissance\*' with a dropdown menu showing '---'. The second section contains: 'Courriel\*' with a text input field; 'Téléphone\*', 'Mobile', and 'Fax' each with a text input field. The third section contains: 'Adresse\*' with a large text input field; 'Code postal\*', 'Ville\*', and 'Pays\*' each with a text input field, where 'Pays\*' shows '---'. The final section contains: 'Votre fonction\*' with a text input field.

## 2. Formulaire : déclaration

---

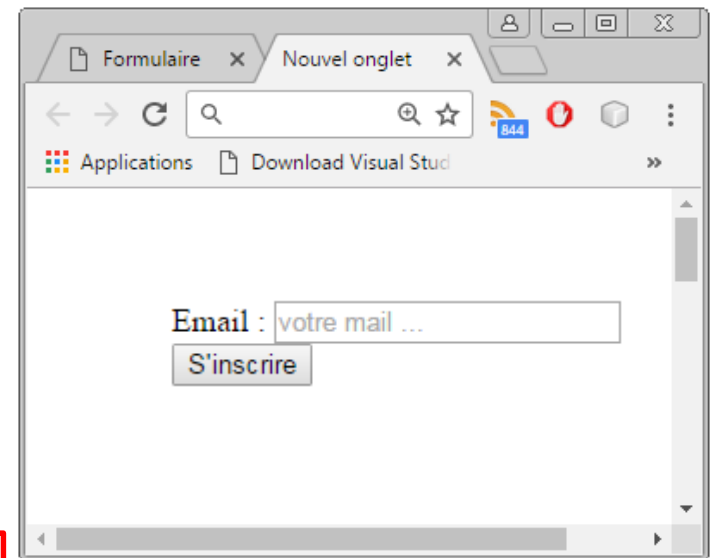
- ▶ L'élément permettant de définir un formulaire est `<form>`. Il est utilisé pour collecter les entrées utilisateur.
- ▶ Les attributs usuels de la balise `<form>` sont :
  - ▶ **action**: il sert à indiquer l'action avec laquelle les données seront récupérées et traitées. On distingue deux types d'action :
    - ▶ *Externe*: les données de formulaire seront traitées par une machine serveur. Dans ce cas, il faut indiquer dans l'action, l'adresse d'une page, d'un programme (PHP, Python, ASP, JSP, ...) ou d'un courrier électronique, situant sur la machine serveur. Par exemple :
      - ▶ `action = "traitement.php"`.
      - ▶ `action="mailto:mon_email@serveur"`.
    - ▶ *Interne*: c'est lorsque les données d'un formulaire sont traitées en interne (côté client) par du JavaScript, l'attribut action reste vide `action=""`.

- ▶ **method**: cet attribut indique par quel moyen les données vont être envoyées. Il existe deux méthodes :
  - ▶ **method="get"** : les informations seront envoyées en clair dans l'adresse de la page (http://...). C'est une méthode en général assez peu adaptée car elle est limitée à 255 caractères.
  - ▶ **method="post"** : c'est la méthode la plus utilisée pour les formulaires car elle permet d'envoyer un grand nombre d'informations. Les données saisies dans le formulaire ne transitent pas par la barre d'adresse.
- ▶ **enctype**: l'attribut **enctype** spécifie sous quel format informatique seront transmises les données du formulaire. Il est utilisé dans le cas de la méthode post.
  - ▶ **application/x-www-form-urlencoded** : c'est le choix par défaut. Tous les caractères sont codés avant d'être envoyés (les espaces sont convertis en symboles "+" et les caractères spéciaux sont convertis en valeurs ASCII HEX)
  - ▶ **text/plain**: les espaces sont convertis en symboles "+", mais aucun caractère spécial n'est codé.
  - ▶ **multipart/form-data**: ce type d'encodage est utilisé lorsque le formulaire contient des fichiers à envoyer au serveur. Les données sont alors encodées sous forme de plusieurs parties séparées, chaque partie contenant une partie différente des données du formulaire.

### 3. Formulaire: éléments

- ▶ Un formulaire HTML peut contenir des éléments pour différents types d'entrée, comme les champs de texte, des cases à cocher, des listes déroulantes, boutons radio, et plus encore.
- ▶ Exemple :

```
<form action="" method="">  
  Nom et prénom : <input type="text" name="noms">  
  <br>  
  Civilité:  
  <input type="radio" name="cv" value="mlle"> Mademoiselle  
  <input type="radio" name="cv" value="mme"> Madame  
  <input type="radio" name="cv" value="m"> Monsieur  
  <br>  
  Date de naissance: <input type="date" name="naissance">  
  <br>  
  Choisissez votre ville:  
  <select name="ville">  
    <option>Tanger</option>  
    <option>Tétouan</option>  
    <option>Al Hoceima</option>  
    <option>Nador</option>  
    <option>Oujda</option>  
  </select>  
  <br>  
  <input type="email" name="email" placeholder="votre mail ...">  
  <input type="submit" name="" value="S'inscrire">  
</form>
```



## 4. Formulaire: éléments d'entrée 'input'

---

- ▶ L'élément `<input type=" " name=" " >` : grâce à l'attribut 'type', cet élément permet de différencier plusieurs types d'entrées :
  - ▶ `<input type="text" name=" ">` : Définit un champ de saisie de texte d'une ligne
  - ▶ `<input type="radio" name=" " value=" ">` : Définit un bouton radio (pour sélectionner un des nombreux choix). Il peut être utilisé pour définir une liste à choix unique.
  - ▶ `<input type="checkbox" name=" " value=" ">` : Définit des cases à cocher. Il peut être utilisé pour une liste à choix multiple.
  - ▶ `<input type="submit">` : Définit un bouton de soumission (pour soumettre le formulaire)
  - ▶ `<input type="reset">` : Définit un bouton d'actualisation des champs remplis.
  - ▶ `<input type="password" name=" ">` : Définit un champ de texte sur une seule ligne dont la valeur sera masquée.
  - ▶ `<input type="button" value="Click Me!">` : Définit un bouton.
  - ▶ Des nouveaux type input ont été ajoutés en html5 : *date, email, search, color, datetime, datetime-local, month, number, range, tel, time, url, week.*

## 5. Formulaire: attribut name, value et checked

---

- ▶ les éléments de formulaire proposent divers attributs, dont notamment 'name', 'value' et 'checked'.
  - ▶ '**name**' permet de définir un identifiant unique pour l'élément de formulaire, essentiel pour identifier les données lors de la soumission du formulaire.
  - ▶ '**value**' détermine la valeur initiale ou par défaut de l'élément de formulaire. Cette valeur est affichée lorsque la page est chargée et est envoyée au serveur si l'utilisateur ne la modifie pas avant de soumettre le formulaire.
  - ▶ '**checked**' est utilisé dans les éléments de formulaire tels que les cases à cocher ou les boutons radio pour indiquer que l'élément est pré-sélectionné ou coché par défaut lors du chargement de la page.

## ► Exemples :

- L'élément `<input type="radio" name=" " value=" " >`

Civilité :

`<input type="radio" name="cvl" value="Mme">` Madame

`<input type="radio" name="cvl" value="M">` Monsieur

`<input type="radio" name="cvl" value="Mlle">` Mademoiselle

### ► **NB :**

- l'attribut 'name' d'une liste à choix unique doit avoir une même valeur
- l'attribut 'value' est obligatoire dans ce cas.
- L'attribut 'checked' peut être utilisé pour sélectionner une valeur par défaut.

- L'élément `<input type="checkbox" name=" " value=" " >`

Vos diplômes :

`<input type="checkbox" name="d[0]" value="l">` LICENCE

`<input type="checkbox" name="d[1]" value="m">` Master

`<input type="checkbox" name="d[2]" value="ing">` Ingénieur

`<input type="checkbox" name="d[3]" value="phD" checked>` Doctorat

- **NB :** L'attribut 'name' d'une liste à choix multiple doit prendre une valeur sous forme de tableau (name='d[]'), et l'attribut 'value' est également obligatoire dans ce cas. Pour qu'une ou plusieurs cases soient cochées, il faut activer l'attribut 'checked'.



## 6. Formulaire: élément select

---

- L'élément `<select>` : permet de définir une **liste déroulante**.

- Les options de la liste sont définies grâce à la balise `<option>`. Un exemple :

```
Choisissez votre ville : <select name="ville">
    <option value="Al Hoceima"> Al Hoceima</option>
    <option value="Nador">Nador</option>
    <option value="Oujda">Oujda</option>
</select>
```

- Par défaut, le premier élément dans la liste déroulante qui est sélectionné. Sinon, il est possible d'utiliser l'attribut *selected* pour sélectionner un autre.
  - Par défaut, l'utilisateur ne peut choisir qu'un élément du menu déroulant. Avec l'attribut *multiple* de la balise `<select>`, **plusieurs choix peuvent être effectués**. Dans ce cas, l'attribut *name* doit prendre une valeur sous forme de tableau (*name="ville[]"*).
  - Il est possible de grouper des options grâce à l'élément `<optgroup>`. Par exemple grouper les villes par région :

```
<optgroup label= "Région Tanger ">
    <option value="Al Hoceima"> Al Hoceima</option>
    <option value="Tanger "> Tanger </option>
</optgroup>
```

## 7. Formulaire: élément datalist

---

- ▶ L'élément `<datalist id=" ">` : spécifie une liste d'options prédéfinies pour un élément `<input list=" ">`. C'est un nouveau élément introduit par html5.
- ▶ Les utilisateurs verront une liste déroulante des options prédéfinies comme des données d'entrée pour `<input list=" ">`.
- ▶ L'attribut **list** de l'élément `<input>`, **doit se référer à l'attribut id** de l'élément `<datalist id=" ">`. Voici un exemple :

```
<input list="ville">  
<datalist id="ville">  
  <option value="Al hoceima">  
  <option value="Nador">  
  <option value="Oujda" >  
  <option value="Tanger">  
  <option value="Tétouan">  
</datalist>
```

## 8. Formulaire: élément `textarea` et `fieldset`

- L'élément `<textarea>` : permet de définir un champ de saisie multi-ligne (une zone de texte). Un exemple :

```
<textarea name="message" rows="10" cols="30">
```

Ceci c'est un texte

```
</textarea>
```

- L'élément `<fieldset>` : permet de regrouper des champs de formulaire. Chacun des `<fieldset>` peut contenir une légende avec la balise `<legend>`.

```
<fieldset>
  <legend>Informations personnelles :</legend>
  Civilité :
  <input type="radio" name="cvl" value="Mme"> Madame
  <input type="radio" name="cvl" value="M"> Monsieur
  <input type="radio" name="cvl" value="Mlle"> Mademoiselle
  <br>
  Nom et prénom: <input type="text" name="nom">
</fieldset>
<fieldset>
  <legend>Paramètres d'accès :</legend>

  Email : <input type="email" name="email">
  <br>
  Mot de passe : <input type="password" name="pass">
</fieldset>
<input type="submit" name="">
```

## 9. Les attributs des champs

- Les éléments de formulaire offrent une variété d'attributs. Certains de ces **attributs** sont **partagés** entre différents types d'éléments, tandis que d'autres sont **spécifiques** à certains types d'éléments.

L'attribut	Éléments qu'il l'utilise	Description
name	<code>&lt;form&gt;</code> , <code>&lt;input&gt;</code> , <code>&lt;button&gt;</code> , <code>&lt;select&gt;</code> , <code>&lt;textarea&gt;</code> , <code>&lt;fieldset&gt;</code> , <code>&lt;object&gt;</code> , <code>&lt;meta&gt;</code> , <code>&lt;map&gt;</code> , <code>&lt;output&gt;</code> , <code>&lt;param&gt;</code>	Nom de l'élément. Utilisé par exemple par le serveur pour récupérer les données envoyées par l'utilisateur.
value	<code>&lt;button&gt;</code> , <code>&lt;option&gt;</code> , <code>&lt;input&gt;</code> , <code>&lt;meter&gt;</code> , <code>&lt;progress&gt;</code> , <code>&lt;param&gt;</code>	Définit une valeur par défaut qui sera affichée dans l'élément au chargement de la page.
required	<code>&lt;input&gt;</code> , <code>&lt;select&gt;</code> , <code>&lt;textarea&gt;</code>	Indique si l'élément doit être obligatoirement rempli ou non.
placeholder	<code>&lt;input&gt;</code> , <code>&lt;textarea&gt;</code>	Indique un texte informatif à afficher pour l'utilisateur dans l'élément.
readonly	<code>&lt;input&gt;</code> , <code>&lt;textarea&gt;</code>	Indique si l'élément peut être modifié.
selected	<code>&lt;option&gt;</code>	Définit une valeur qui sera sélectionnée au chargement de la page.
checked	<code>&lt;command&gt;</code> , <code>&lt;input&gt;</code>	Indique si l'élément doit être coché au chargement de la page.
disabled	<code>&lt;button&gt;</code> , <code>&lt;command&gt;</code> , <code>&lt;select&gt;</code> , <code>&lt;option&gt;</code> , <code>&lt;optgroup&gt;</code> , <code>&lt;textarea&gt;</code> , <code>&lt;fieldset&gt;</code>	Indique si l'utilisateur peut interagir avec l'élément.
multiple	<code>&lt;input&gt;</code> , <code>&lt;select&gt;</code>	Définit si plusieurs valeurs peuvent être saisies dans un champ de type email ou file.
autofocus	<code>&lt;button&gt;</code> , <code>&lt;input&gt;</code> , <code>&lt;keygen&gt;</code> , <code>&lt;select&gt;</code> , <code>&lt;textarea&gt;</code>	L'élément doit être automatiquement sélectionné après le chargement de la page.
rows/cols	<code>&lt;textarea&gt;</code>	Définit le nombre de lignes/colonnes dans un textarea.
size	<code>&lt;input&gt;</code> , <code>&lt;select&gt;</code>	Définit la largeur de l'élément (en pixels). Si l'attribut type de l'élément est text ou password, il s'agit alors du nombre de caractères.
wrap	<code>&lt;textarea&gt;</code>	Indique comment la césure doit être gérée.
for	<code>&lt;label&gt;</code> , <code>&lt;output&gt;</code>	Décrit l'élément qui lui est associé
max	<code>&lt;input&gt;</code> , <code>&lt;meter&gt;</code> , <code>&lt;progress&gt;</code>	Définit la valeur maximum autorisée.