

Module: Développement Web

Mohamed CHERRADI

UAE / ENSAH

Formateur Web



m.cherradi@uae.ac.ma

Chapitre 2

Langage HTML



Plan

Langage HTML

Historique du html
Structure d'une page html
HTML5



Organisation du contenu

La maquette d'un site web
Les balises sémantiques
...



Les éléments multimédia

Audio, vidéo, image, ...



Formulaires & Frames

Les types des éléments
Frameset

...

Introduction

Language HTML



Introduction

- ◆ **HTML** (HyperText Markup Language) : C'est un langage de balisage (markup) permettant d'écrire de l'hypertexte (hypermedia). Il fait appel à des balises pour structurer le document.
- ◆ Son rôle est de **gérer et organiser le contenu**. C'est donc en HTML qu'on **d'écrit ce qui doit être affiché sur la page** : du texte, des liens, des images...
- ◆ Ce langage universel et incontournable permet de **représenter les sites et applications web**.
- ◆ Il est utilisé pour **créer et représenter visuellement une page web**. C'est ce qui détermine le contenu et la présentation de base d'une page web mais pas ses fonctionnalités.
- ◆ Il permet d'enrichir un texte avec des informations **structurelles, sémantiques** et de **présentation**.
- ◆ Il est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation (feuilles de style ou CSS)
- ◆ **Remarque:** HTML **n'est pas à proprement parlé un langage de programmation**, mais plutôt un langage qui permet de mettre en forme du contenu.

Historique

- ◆ 1990 : HTML est créé par Tim Berner-Lee au Centre Européen de Recherche Nucléaire (CERN)
- ◆ 1995 : HTML 2.0 normalisation par l'Internet Engineering Task Force (IETF)
- ◆ 1996 : HTML 3.2 ajout des tables, des applets (Java), etc.
- ◆ 1996 : normalisation par le W3C (World Wide Web Consortium).
- ◆ 1998 : HTML 4.01 ajout des feuilles de styles, des frames, etc.
- ◆ 2000 : XHTML 1.0 : reformulation de HTML 4 en XML 1.0.
- ◆ 2002-2006 : XHTML 2.0.
- ◆ 2007- maintenant : HTML5 : apporte de nombreuses améliorations : nouvelles fonctionnalités pour les formulaires, possibilité d'inclure facilement des vidéos, des balises sémantiques pour l'indexation, etc.

```
<body>
  <applet code="NomDeLaClasseDeLApplet.class" width="300" height="300">
    <!-- Paramètres optionnels de l'applet -->
    <param name="parametre1" value="valeur1">
    <param name="parametre2" value="valeur2">
  </applet>
</body>
```


Structure d'une page HTML

- ◆ Toutes les pages HTML utilisent la même structure de base, dont voici les ingrédients:

Structure d'une page HTML

```
<!-- commentaire -->
<html>
  <head>      L'en-tête du document <head> est réservée aux méta-informations (les informations relatives au document)
  .....
  </head>
  <body>      Le conteneur <body> contient ce qui doit être affiché: texte, images, sons...
  .....
  </body>
</html>
```

- ◆ Les commentaires peuvent être placés n'importe où dans un document HTML:

- ▶ <!-- Ceci est un commentaire -->
- ▶ /* Ceci est aussi un commentaire */

- ◆ Pour écrire du code HTML, on utilise des éditeurs de texte (IDE) dédiés à l'écriture de code : VS Code, Sublime

< > Text, Atom, Notepad++, ...

Introduction

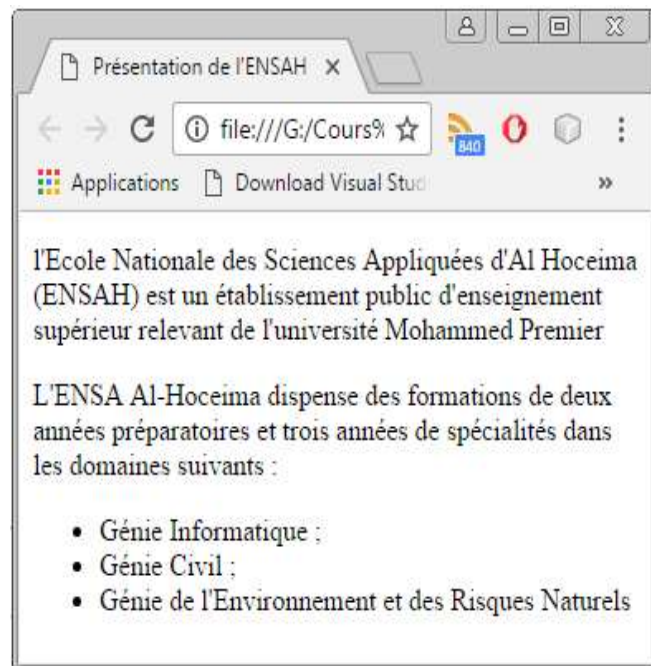
- ◆ Une balise est un mot clé, une commande du langage HTML insérée dans le corps du document pour introduire un effet particulier (Exemples : début de mise en gras, fin de mise en gras, début de tableau...).
- ◆ Une balise commence toujours par un signe "<" et se finit toujours par un signe ">". La 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 on utilise la commande **b**. La balise ouvrante est `` et la balise fermante est ``. Donc si, dans la phrase "Je suis étudiant à l'ENSAH.", je veux mettre le mot 'ENSAH' en gras comme ceci "Je suis élève ingénieur à l'**ENSAH**.", le code HTML correspondant sera : Je suis élève ingénieur à l'``ENSAH``.

Introduction

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

```
<!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

Éléments 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 `` 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 `` , 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">`.

Types d'Eléments

◆ Un élément HTML peut être constitué d'une :

▶ **Balise en paires** (ouvrante et fermante) et d'un contenu.

● Exemple: `<title>`Ceci est le titre de ma 1ère page Web`</title>`

▶ **Balise unique** qu'on dit **orpheline** (pas besoin de balise de fin). Ne contiennent pas de texte, n'ont pas de balise de fermeture

● Exemple: `
`, ``

◆ **Remarque:** Ces éléments sont classifiées en deux catégories:

◆ Élément bloc (div) : break line

◆ Élément inline (span)

◆ **Caractères spéciaux:** `$nbsp;`, `<`, `>`;

Les attributs liés aux événements

- ◆ onclick="action",
- ◆ onchange
- ◆ onsubmit
- ◆ ondblclick,
- ◆ onkeydown,
- ◆ onkeypress,
- ◆ onkeyup,
- ◆ onmouseover,
- ◆ onmouseout,
- ◆ onload
- ◆ onfocus
- ◆ ...

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.
- ◆ Il propose de **nouveaux éléments**, **attributs** et **comportements** pour de nouveaux usages, afin de répondre aux besoins grandissants des internautes.
- ◆ En plus de la **possibilité de l'utilisation de média riches** (vidéo, audio, SVG, etc.), HTML5 apporte des fonctionnalités permettant le développement d'**applications** web **interactives**.
- ◆ HTML 5 introduits des nouveaux éléments pour permettre d'utiliser des nouvelles API, à savoir :
 - ▶ une API de drag and drop en combinaison avec l'attribut draggable
 - ▶ une API pour jouer des vidéos et des sons/musiques utilisé avec les nouvelles balises video et audio une API de dessin 2D utilisé avec la nouvelle balise canvas.
 - ▶ une API utilisée pour les applications hors-lignes avec le nouvel attribut manifest="demo.appcache" une API d'édition en combinaison avec le nouvel attribut contenteditable (localStorage, sessionStorage, ...).
 - ▶ une API qui permet l'accès à l'historique pour prévenir les problèmes de bouton retour-en-arrière.

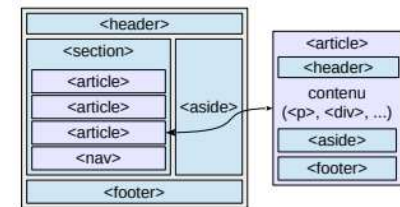


Une version récente : HTML5

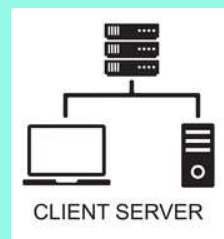
- ◆ **HTML5 = HTML + CSS+ JavaScript** : Il s'agit bien d'un abus de langage, mais un abus justifié : le contenu (HTML) étant bien dissocié de la forme (CSS), mais peu exploitable pour les visiteurs sous une forme brute sans mise en pages, et peu dynamique sans langage de script pour des interactions avec le contenu de la page lui-même.

- ◆ Les nouveautés de HTML5:

- ▶ Les champs sémantiques: [article](#), [section](#), [nav](#), [header](#), [footer](#), [aside](#), ...
- ▶ De nouveaux éléments, comme par exemple: [number](#), [color](#), [date](#), [datetime](#), [range](#), [email](#), [week](#), [tel](#), [search](#), [reset](#), ...
- ▶ Intégrer ce qu'on appelle des attributs de validation à travers l'utilisation de [regex](#) ([required](#) [pattern=regex](#))
- ▶ Avant HTML5, pour intégrer l'[audio](#) et/ou [vidéo](#) on utilise des plugins comme flash media player
- ▶ Canvas et SVG : HTML5 inclut le balisage [<canvas>](#) qui permet de dessiner des graphiques, des animations. Ainsi, il prend en charge l'utilisation de l'élément [<svg>](#) pour créer des graphiques vectoriels.



Organisation du contenu



Document HTML5 : La structure générale

◆ `<!DOCTYPE >` : type du document.

► **Exemple:** `<!DOCTYPE html >` : Document de type HTML5

◆ `<html></html>`: balise principale englobant le contenu de la page Web

◆ `<head></head>`: en-tête du document

◆ `<meta/>`: permet de rajouter **d'autres informations concernant le document**. Balise non obligatoire. Cette balise peut avoir plusieurs attributs tels que :

► **charset** : type d'encodage de la page, important pour caractères spéciaux (ex. `<meta charset="utf-8" />`)

► **name** : a généralement la valeur **author**, **description** ou **keywords**. S'il existe il est suivi de l'attribut **content**.

► **content** : bref description de ce qui est indiqué dans name.

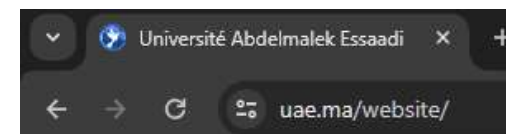
◆ `<title></title>`: titre de la page Web

◆ `<link rel="icon" href="icon_path" type="image/icon type">` : Logo icon d'un onglet

◆ `<script type="text/javascript" src="scripts.js"></script>` : Ajouter un script JS

◆ `<body></body>`: contenu principal de la page Web

```
<!DOCTYPE html>
<html>
  <head>
    <title>FSTH -Site Officiel</title>
    <meta charset=utf-8" />
    <meta name="description" content="Site de la FST d'Al Hoceima (FSTH)" />
    <meta name="author" content="Tarik" />
    <meta name="keywords" content="FSTH, MIP, BCG, MI, EEA, ..." />
    <meta name="generator" content="Adobe Dreamweaver" />
  </head>
```



L'encodage des éléments

- ◆ La première chose que l'on fait d'habitude sur une page web c'est déclarer l'encodage de caractère utilisé.
- ◆ C'est lui qui détermine comment les caractères spéciaux vont s'afficher comme: les caractères accentués, caractères arabes, caractères Tifinagh , Idéogrammes chinois et japonais, etc.
- ◆ HTML5 propose d'utiliser l'élément `<meta>` d'une façon plus simple : `<meta charset=valeur/>`
- ◆ Il existe plusieurs techniques d'encodage utilisées en fonction des langues. L'attribut *charset* peut prendre l'un des valeurs suivantes: UTF-8, ISO- 8859-1, OEM 775, Windows-1253... L'encodage le plus utilisé aujourd'hui c'est UTF-8; il prend en charge pratiquement tous les symboles de toutes les langues.
- ◆ L'élément `<meta charset="UTF-8">` se place dans l'entête juste après la balise `<head>`.

Bien à 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 les éléments vides comme `<meta charset="UTF-8">`, il est optionnel de les fermer.
 - ▶ Les balises doivent être correctement imbriquées.
 - **correct** : `<a><c>contenu</c>`.
 - **incorrect** : `<a><c>contenu</c>`.
 - ▶ 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 -->` ou bien `/* Commentaire sur une ou plusieurs lignes */`

Balise 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 :

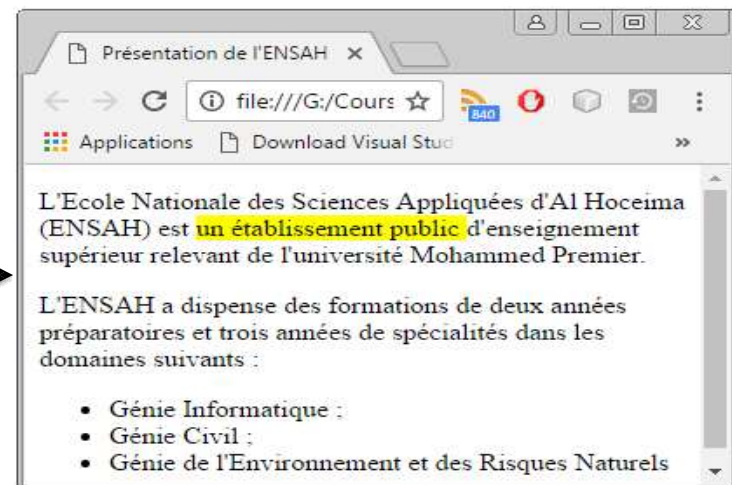
- ▶ `<p>` : définit un paragraphe contenant une ou plusieurs phrases.
- ▶ `` et `` : sont utilisés pour spécifier une liste d'éléments (``). `` pour une liste à puces, `` pour une liste numérotés.
- ▶ `
` : permet de créer un saut de ligne.
- ▶ `<h1>` à `<h6>` : utilisées pour la hiérarchisation des titres.
- ▶ `` : utilisée dans un paragraphe pour mettre une partie de texte en gras.
- ▶ `<u>` : pour souligner un texte.
- ▶ `` : 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 ``, ``, `<cite>`, `<dfn>`, ``, `<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.

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>

```



Structuration de la page

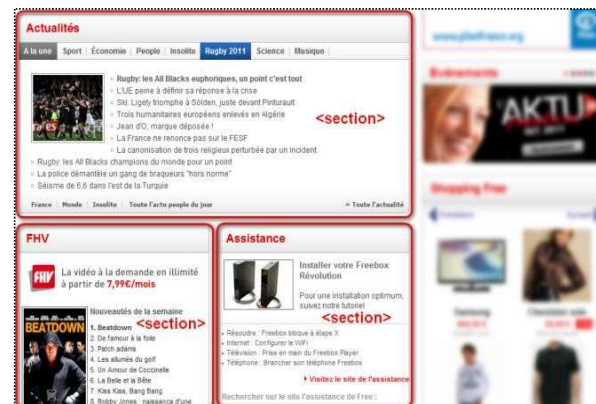
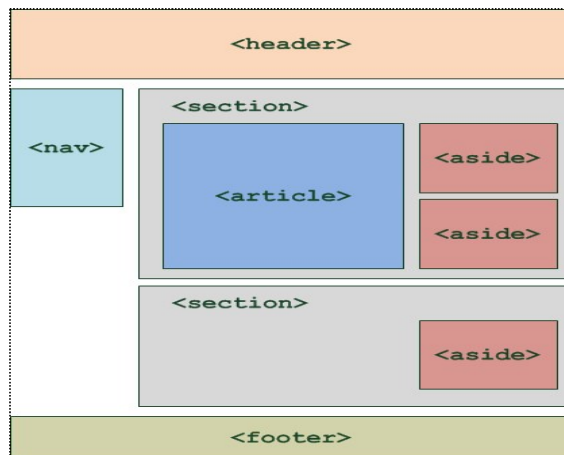
- ◆ La structure d'une page Web, à savoir la structure sémantique de ce qui est entre `<body>` et `</body>`, est fondamentale pour la présentation de la page à l'utilisateur.
- ◆ HTML 5 propose un ensemble d'éléments pour cette structuration à savoir : `<article>`, `<header>`, `<section>`, `<footer>`,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

```
<body>

  <header>
    <nav <!-- Navigation principale du site--> </nav>
  </header>
  <article>
    <header>
      <h1 <!-- titre de l'article --> </h1>
    </header>
    <div <!-- Une image par exemple --> </div>
    <section>
      <p <!-- des paragraphes --> </p>
    </section>
  </article>
  <footer>
    <!-- footer de la page -->
  </footer>

</body>
```

D'autres exemples



Structuration de la page

- ◆ 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 automatiques (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 à ses débuts 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 (<https://validator.w3.org/nu/#textarea>).
- ◆ À l'heure actuelle, ces balises ne sont pas très utilisées. Les développeurs utilisent encore des balises génériques comme `<div>` à la place pour englober les différentes portions de contenu de la page. D'ailleurs, c'est comme cela qu'on faisait avant l'arrivée de ces nouvelles balises HTML5.

Organisation par <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

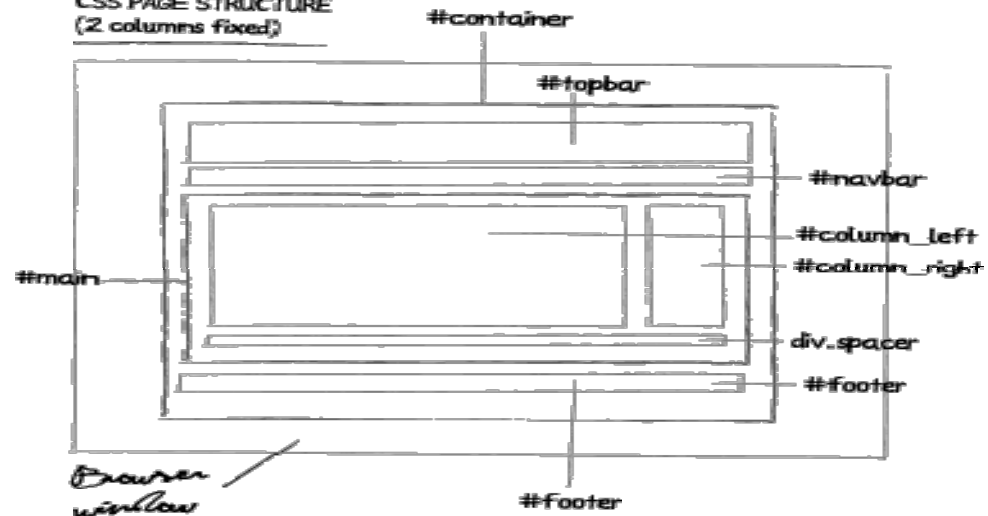
*to force #container
height*

```

<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>

```

CSS PAGE STRUCTURE
(2 columns fixed)



Définir un tableau

- ◆ 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>`.

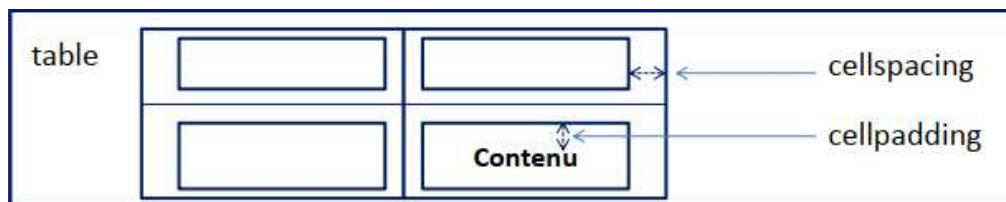
| Nom | Prenom | Age |
|------|---------|-----|
| Jill | Smith | 50 |
| Eve | Jackson | 20 |

- ◆ 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.

Définir un tableau

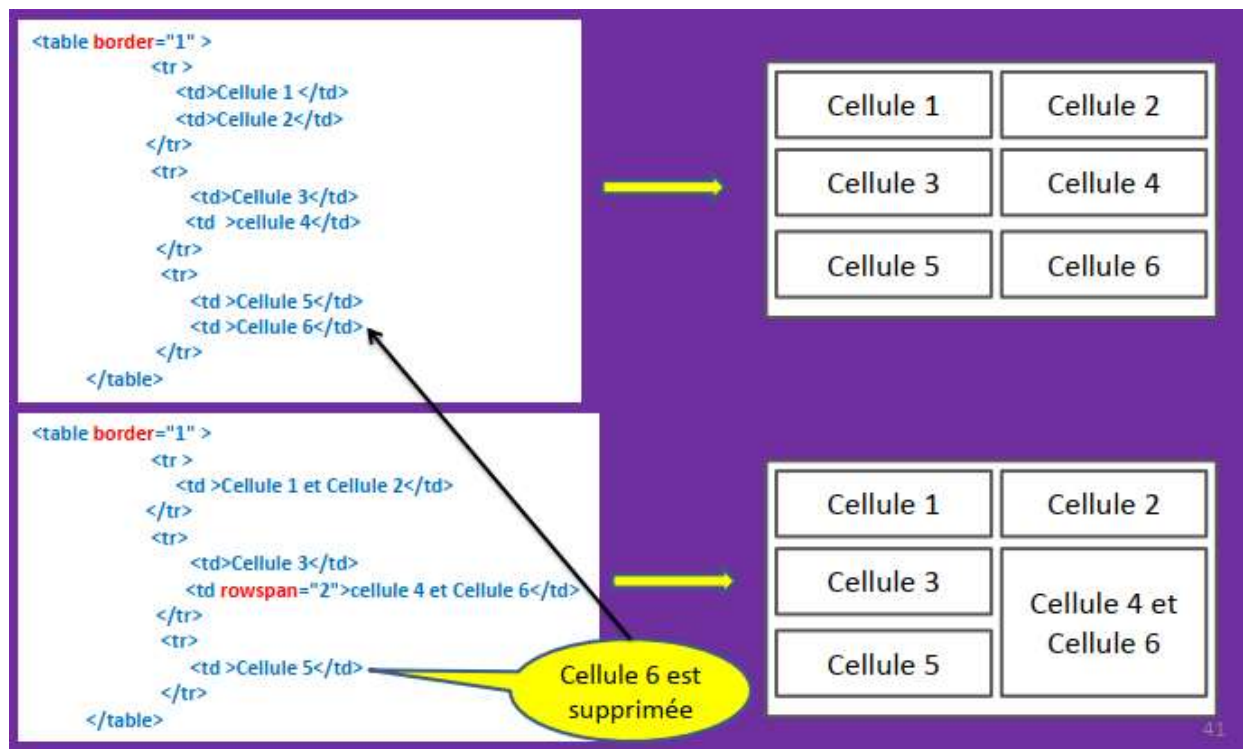
◆ La balise `<table>` peut accueillir de nombreux attributs :

- ▶ **border** : définit, en pixels, la taille de la bordure entourant le tableau (aussi les cellules).
- ▶ **width** : définit en pixels ou en pourcentage (%) la largeur du tableau.
- ▶ **height** : définit en pixels la hauteur du tableau.
- ▶ **cellspacing** : définit en pixels ou en pourcentage l'espacement entre les différentes cellules du tableau.
- ▶ **cellpadding** : définit en pixels ou en pourcentage (%) la marge interne des cellules



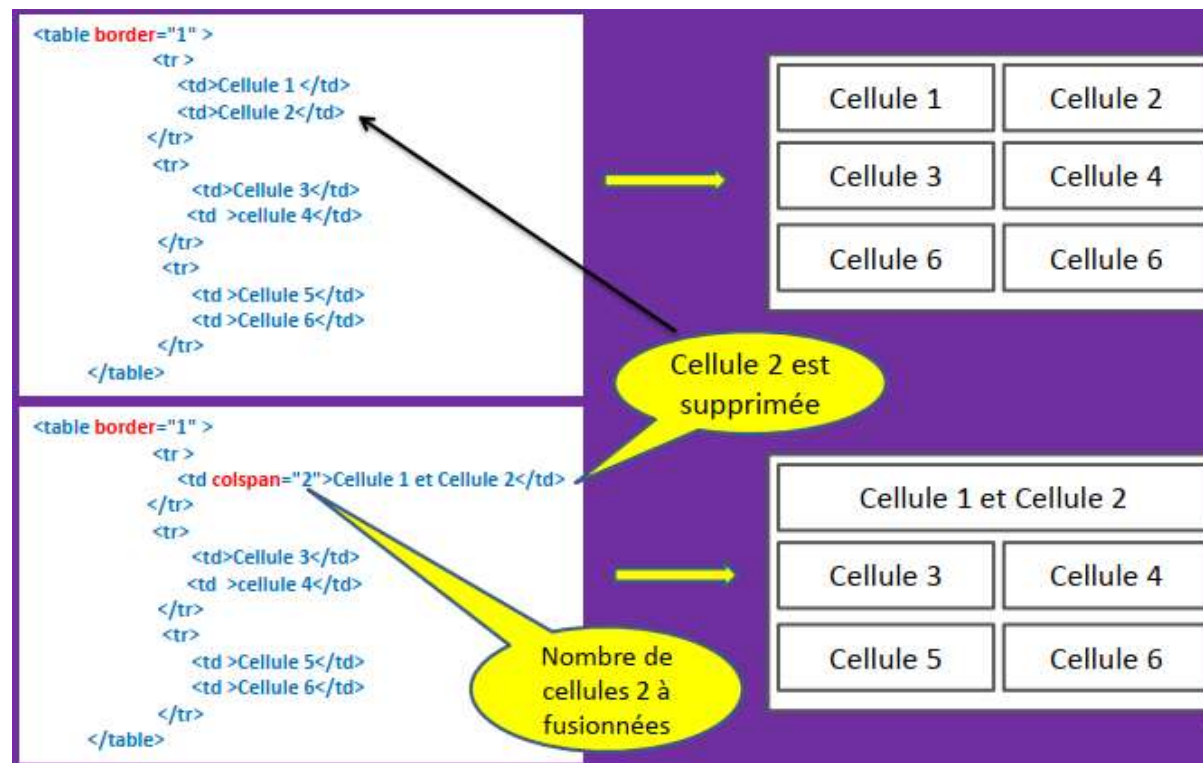
Définir un tableau

- ◆ La fusion de lignes: l'attribut `rowspan` de la balise `td` va faire la fusion des cellules horizontalement



Définir un tableau

- ◆ La fusion de colonnes : l'attribut **colspan** de la balise **td** va faire la fusion des cellules horizontalement.

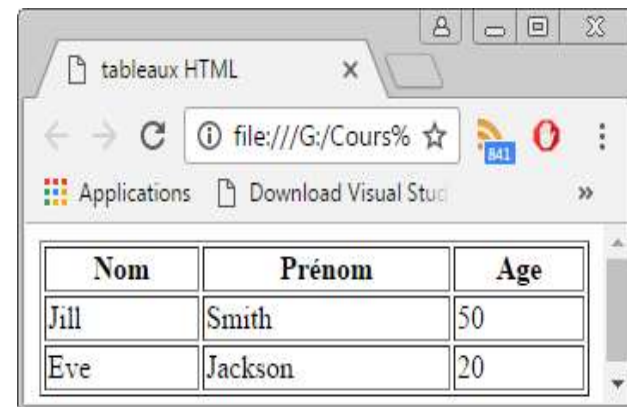


Exemples

```
<!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>
```

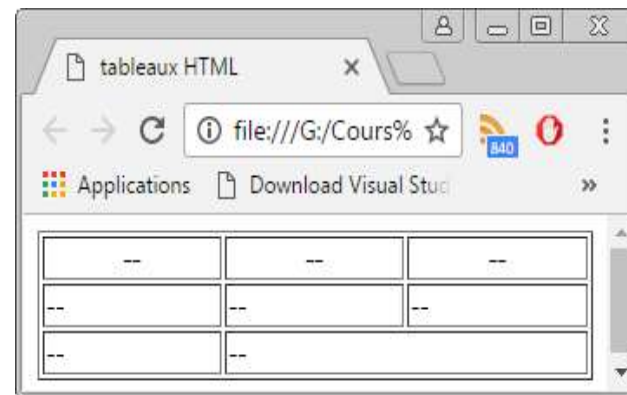


| Nom | Prénom | Age |
|------|---------|-----|
| Jill | Smith | 50 |
| Eve | Jackson | 20 |

```
<!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>
```



| -- | -- | -- |
|----|----|----|
| -- | -- | -- |
| -- | -- | |

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 ceci grâce à des liens hypertextes.
- ◆ Ces liens sont utilisés généralement pour aller vers :
 - ▶ Un endroit de la page en cours
 - ▶ Une autre page du site.
 - ▶ 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 : ` link text `
 - ▶ `href` est l'attribut permettant de spécifier l'adresse de destination
 - ▶ *link text* : le **texte du lien**, c'est la partie visible (la partie cliquable). En cliquant sur pour à aller l'adresse indiquée.

D'autres attributs de l'éléments <a>

- ◆ L'attribut `target` : spécifie l'endroit où peut s'ouvrir le document lié. Sa syntaxe est: ``. 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.
- ◆ Les ancres : 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'une ancre* : `<h2 id="mon_ancre">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 l'ancre `Aller vers l'ancre`

Les éléments multimédia



Insertion des images et des figures

- ◆ En HTML, les images sont définies avec la balise ``.
- ◆ La balise `` est vide, il contient des attributs uniquement, et ne dispose pas d'une balise de fermeture
- ◆ Sa syntaxe :

```

```

- ▶ L'attribut `src` spécifie l'URL (adresse web) de l'image.
- ▶ L'attribut `alt` fournit un texte alternatif pour une image, si l'image n'est pas affichée pour une raison quelconque.
- ▶ L'attribut `style` est utilisé pour spécifier les propriétés de style comme : hauteur et largeur, emplacement de l'image dans la page (gauche ou droite), et le bordure.

- ◆ Sa syntaxe :

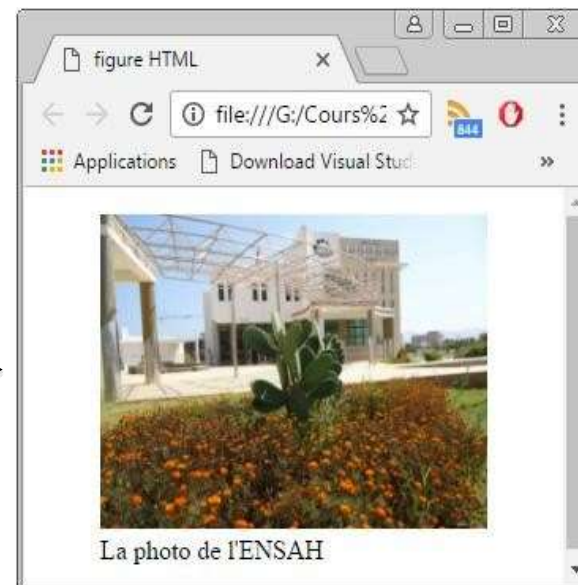
```
<img src= "ensah.jpg" alt= "Photo de ENSAH" style="float:right;width:42px;height:42px;border:0;"/>
```

Insertion des images et des figures

- ◆ 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>
```



Insertion de l'audio et de la vidéo

- ◆ En plus de la possibilité de l'insertion des images dans le contenu d'un document html, HTML5 donne la possibilité de l'insertion de l'audio et la vidéo.
- ◆ Avant l'HTML5, les fichiers audio/vidéo ne peuvent être lus dans un navigateur que en utilisant un plug-in (comme le flash).
- ◆ Pour l'Audio, HTML5 propose l'élément `<audio>` pour l'intégrer dans une page Web. Un 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>` pour l'intégrer dans une page Web. Un exemple d'utilisation :

Insertion de l'audio et de la vidéo

```
<video width="320" height="240" controls>  
  <source src="movie.mp4" type="video/mp4"> Ce navigateur ne supporte pas ce type  
  de fichier.  
</video>
```

- ◆ 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.
- ◆ A noter que sauf **trois formats** sont supportés pour chacun des deux éléments :
 - ▶ Pour `<audio>` : MP3, Wav et Ogg.
 - ▶ Pour `<video>` : MP4, WebM et Ogg.

Formulaires & Frames



Formulaires

- ◆ **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 l'une des techniques les plus efficace pour recevoir des informations provenant directement de l'utilisateur final, et en outre, structurées selon les besoins du concepteur.

A sample web form with the following fields and sections:

- Civilité***: Radio buttons for ☐ Madame, ☐ Mademoiselle, ☐ Monsieur
- Nom***: Text input field
- Prénom(s)***: Text input field
- Date de naissance***: Date picker with dropdowns for day (01), month (01), and year
- Pays de naissance***: Dropdown menu with "-- --"
- Courriel***: Text input field
- Téléphone***: Text input field
- Mobile**: Text input field
- Fax**: Text input field
- Adresse***: Text input field with a small map icon on the right
- Code postal**: Text input field
- Ville***: Text input field
- Pays***: Dropdown menu with "-- --"
- Votre fonction***: Text input field

Déclaration de formulaire

- ◆ 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, Perl, ASP...) 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=""`.

Déclaration de formulaire

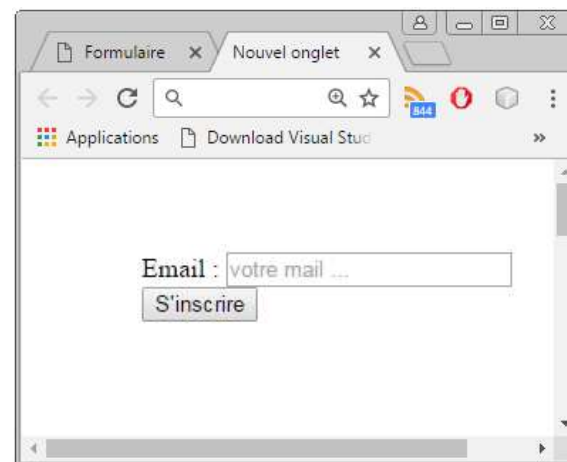
- ◆ **method**: cet attribut indique **par quel moyen les données vont être envoyées**. Il existe deux méthodes :
 - ▶ **method="get"** : les informations **seront envoyées en clair dans l'adresse de la page** (http://...). C'est une méthode en général assez peu adaptée car elle est limitée à 255 caractères.
 - ▶ **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)
 - ▶ **multipart/form-data**: aucun caractère n'est codé. Cette valeur est requise **lorsque vous utilisez des formulaires qui ont un contrôle de téléchargement de fichier.**
 - ▶ **text/plain**: les espaces sont convertis en symboles "+", mais aucun caractère

Les éléments du formulaire

- ◆ 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="">
Nom et prénom: <input type = "text"> <br>
Civilité : <input type = "radio" name="madame">Madame
<input type = "radio" name="demoiselle">Mademoiselle
<input type = "radio" name="monsieur">Monsieur<br>
Date de naissance : <input type = "date" name="datenaissance"><br>
Choisissez votre ville : <select name="ville">
<option value="Al Hoceima"> Al Hoceima</option>
<option value="Nador">Nador</option>
<option value="Oujda">Oujda</option>
</select><br>
Email : <input type="email" placeholder="votre mail ..." ><br>
<input type = "submit" value="S'inscrire">
</form>
```



Les éléments du formulaire

- ◆ L'élément `<input type = " " >` : grâce à l'attribut **'type'**, cet élément permet de différencier plusieurs types d'entrée :
 - ▶ `<input type = "text">` : Définit un champ de saisie de texte d'une ligne
 - ▶ `<input type = "radio">`: Définit un bouton radio (pour sélectionner un des nombreux choix)
 - ▶ `<input type = "checkbox">`: Définit des cases à cocher
 - ▶ `<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">` : 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`.
- ◆ 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>
```



Les éléments du formulaire

◆ 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 est sélectionnée. Sinon, il est possible d'utiliser l'attribut `selected`.
- 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**.
- Il est possible de grouper des options grâce à l'élément `<optgroup>`. Par exemple grouper les villes par région :

Choisissez votre ville :

```
<select name="ville">
  <optgroup label="Région Tanger">
    <option value="Al Hoceima"> Al Hoceima</option>
    <option value="Tanger"> Tanger </option>
  </optgroup>
  <optgroup label="Région Oriental">
    <option value="Nador">Nador</option>
    <option value="Oujda">Oujda</option>
  </optgroup>
</select>
```

Les éléments du formulaire

◆ L'élément `<datalist>` : spécifie une liste d'options prédéfinies pour un élément `<input>`. C'est un nouveau élément introduit par `html5`.

- ▶ Les utilisateurs verront une liste déroulante des options prédéfinies comme des données d'entrée pour `<input>`.
- ▶ L'attribut **de liste** de l'élément `<input>`, doit se référer à l'attribut **id** de l'élément `<datalist>`. Voici un exemple :

```
<form action="/action_page.php" method="get">
  <label for="browser">Choose your browser from the list:</label>
  <input list="browsers" name="browser" id="browser">
  <datalist id="browsers">
    <option value="Edge">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
  <input type="submit">
</form>
```

Choose your browser from the list: Envoyer

Edge
Firefox
Chrome
Opera
Safari

Les éléments du formulaire

```

<form name="traitement"
method="post" action="
traitement.html">
Choisissez un nombre : <br/>
<input type="range" /> <br/>
Choisissez une couleur : <br/>
<input type="color" /> <br/>
Date (jj/mm/aaaa): <br/>
<input type="date" /> <br/>
Temps (hh:mm): <br/>
<input type="time" /> <br/>
Semaine : <br/>
<input type="week" /> <br/>
Mois : <br/>
<input type="month" /> <br/>
Date (jj/mm/aaaa hh:mm) : <br/>
<input type="datetime-local" /> <br/>
</form>

```

Choisissez un nombre :

Choisissez une couleur :

Date (jj/mm/aaaa):

Temps (hh:mm):

Semaine :

Mois :

Date (jj/mm/aaaa hh:mm) :



Les éléments du formulaire

```
<form name="traitement" method="post" action="traitement.html">
Rechercher : <br/>
  <input type="search" />
</form>
```

Rechercher :

Mention :


```
<select>
  <option selected> Choisissez</option>
  <option> Assez bien </option>
  <option> Bien</option>
  <option> Très bien </option>
</select>
```

Mention :

Choisissez ▼
 Choisissez
 Assez bien
 Bien
 Très bien

```
<form method="post" action="traitement.php">
  <label>Taper un message : </label><br />
  <textarea name="precisions" id="precisions" cols="40" rows="4">
  </textarea>
</form>
```

Taper un message :

Les éléments du formulaire

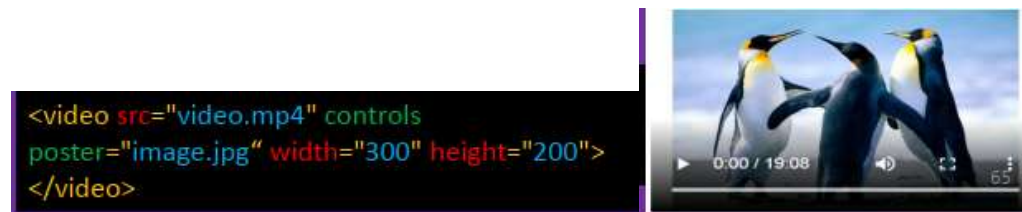
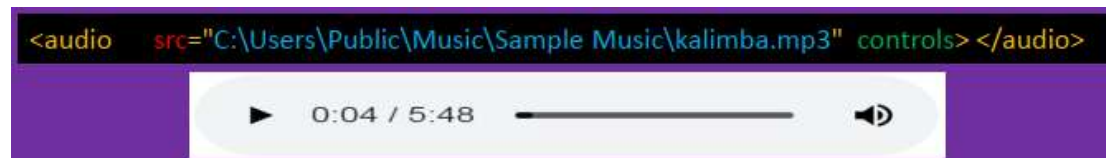
```
<form method="post" action="traitement.php">
  <label>Dans quel pays habitez-vous ?</label><br />
  <select name="pays" id="pays">
    <optgroup label="Europe">
      <option value="france">France</option>
      <option value="espagne">Espagne</option>
      <option value="italie">Italie</option>
      <option value="royaume-uni">Royaume-Uni</option>
    </optgroup>
    <optgroup label="Amérique">
      <option value="canada">Canada</option>
      <option value="etats-unis">Etats-Unis</option>
    </optgroup>
    <optgroup label="Asie">
      <option value="chine">Chine</option>
      <option value="japon">Japon</option>
    </optgroup>
  </select>
</form>
```

Dans quel pays habitez-vous ?

| | |
|-----------------|---|
| France | ▼ |
| Europe | |
| France | |
| Espagne | |
| Italie | |
| Royaume-Uni | |
| Amérique | |
| Canada | |
| Etats-Unis | |
| Asie | |
| Chine | |
| Japon | |

54

Les éléments du formulaire



| | | |
|---------|-------|--|
| En-tête | thead | |
| | | |
| | | |
| corps | tbody | |
| | | |
| | | |
| pied | tfoot | |

Regroupement des champs

- ◆ HTML donne la possibilité de **regrouper des champs de formulaire** grâce à la balise `<fieldset>`.
- ◆ Chaque `<fieldset>` **peut contenir une légende avec la balise `<legend>`**.

```
<form action="">
<fieldset style="width:30%">
  <legend>Informations personnelles</legend>
  Civilité : <input type = "radio" name="madame">Madame
             <input type = "radio" name="demoiselle">Mademoiselle
             <input type = "radio" name="monsieur">Monsieur<br>
  Nom et prénom: <input type = "text"> <br>
  Date de naissance : <input type = "date" name="datenaissance"><br>
</fieldset>

<fieldset style="width:30%">
  <legend>Paramètres d'accès</legend><br>
  Email : <input type="email" placeholder="votre mail ..."> <br>
  Mot de pass : <input type="password">
</fieldset>
<br>
<input type = "submit" value="Inscription">
</form>
```

Informations personnelles

Civilité : ☐ Madame ☐ Mademoiselle ☐ Monsieur
 Nom et prénom:
 Date de naissance :

Paramètres d'accès

Email :
 Mot de pass :

Inscription

Les attributs des champs

- ◆ Les différents éléments de formulaire admettent des attribues comme : `name`, `size`, `maxlength`, `value`, `readonly` , `placeholder` (pour texte), `checked` (pour bouton), `required`, `pattern`, `height`, `width`. Il y ceux qui sont communs entre les éléments mais il y a d'autres bien spécifique.
- ◆ L'attribut '`name`' permet de définir un nom (name) unique pour l'élément. Il est utilisé pour collecter le contenu de la ligne de texte lors de la soumission du formulaire, c'est-à-dire qu'il est obligatoire de l'utiliser.
- ◆ L'attribut `size` permet de définir le nombre de caractères visibles de la ligne de texte et, par conséquent, la largeur de la zone de texte.
 - ▶ L'utilisateur peut néanmoins encoder autant de caractères qu'il souhaite même s'ils débordent du champ de la zone visible.
 - ▶ La valeur par défaut de `size` est de 20.

HTML size Attribute

Name:

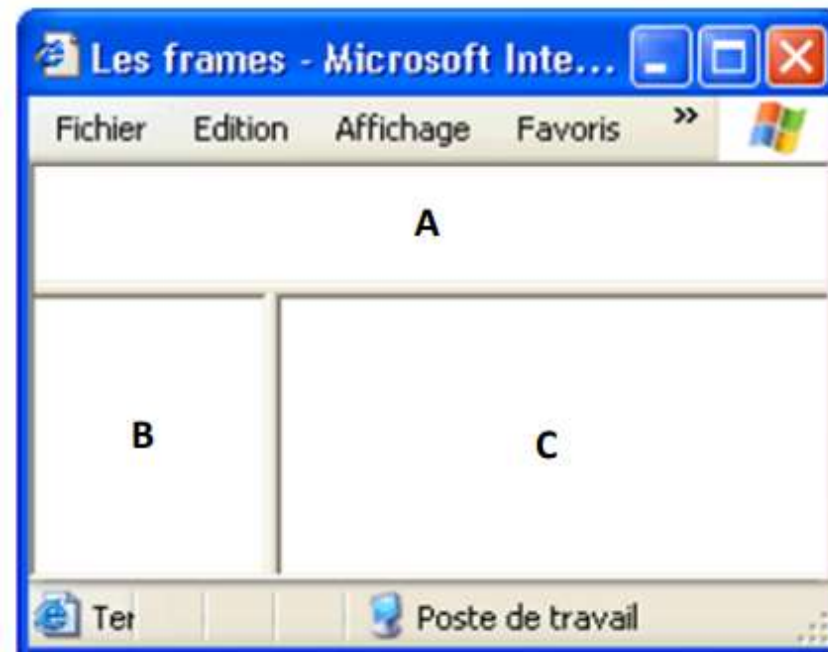
Email-id:

Les différents attributs

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

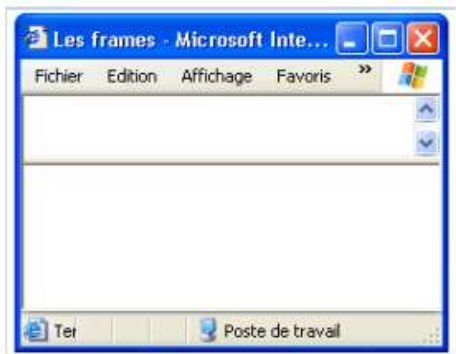
Les frames

- ◆ Avec le langage HTML, il est possible de diviser une fenêtre en cadres ou plusieurs zones appelées **frames** (cadres).
- ◆ Grâce aux **frames** (cadres), il est possible d'afficher plusieurs pages HTML, chacune dans un frame (zone ou cadre) différent.
- ◆ **Exemple** : Page Web divisée en trois frames A, B et C :



Les frames « frameset »

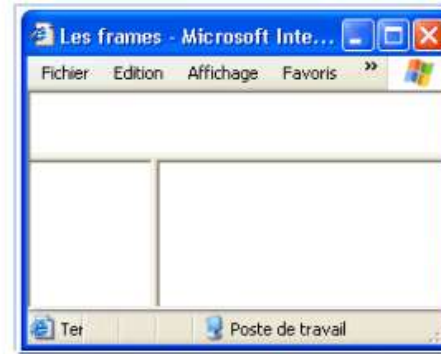
- ◆ La balise `<frameset>` permet de définir le nombre de frames et leurs dimensions. Les deux principaux attributs sont :
 - ◆ `rows="H1,H2,...Hn"` : permet de créer des frames **horizontales** (Exemple 1).
 - ◆ `cols="V1,V2,...Vn"` : permet de créer des frames **verticales** (Exemple 2).
- ◆ Remarques:
 - ◆ Les hauteurs et largeurs peuvent être données en pixels ou en pourcentage.
 - ◆ Les deux attributs `rows` et `cols` peuvent être utilisés conjointement (Exemple 3).
 - ◆ Dans un document HTML, la balise `<frameset>` remplace la balise `<body>`.



Exemple 1



Exemple 2



Exemple 3

Les frames « frameset »

◆ Pour définir le contenu de chacune des zones au sein de la balise `<frameset>`, on utilise la balise `<frame>`.

◆ Exemples:

The diagram illustrates two examples of HTML framesets and their corresponding browser rendering. Each example consists of an HTML code block on the left and a browser window on the right, connected by a blue arrow.

Top Example (Columns):

```
<html>
<head>
  <title>Mon site</title>
</head>
<frameset cols="30%,70%">
  <frame>
  <frame>
</frameset>
</html>
```

The browser window shows a single page titled "Mon site" with a file path of "file:///D:/...". The page is divided into two vertical frames: a smaller one on the left (30%) and a larger one on the right (70%).

Bottom Example (Rows):

```
<html>
<head>
  <title>Mon site</title>
</head>
<frameset rows="30%,70%">
  <frame>
  <frame>
</frameset>
</html>
```

The browser window shows a single page titled "Mon site" with a file path of "file:///D:/...". The page is divided into two horizontal frames: a smaller one at the top (30%) and a larger one at the bottom (70%).

Les frames « frameset »

- ◆ Pour ajouter le contenu de chacun des cadres, la balise doit contenir les deux attributs :
 - ◆ `src=url` : pour indiquer le fichier à placer dans la zone.
 - ◆ `name` : le nom de la zone afin qu'elle puisse devenir la cible d'un lien (on utilisera alors l'attribut `target` de la balise `<a>` pour y accéder).
- ◆ Exemples:

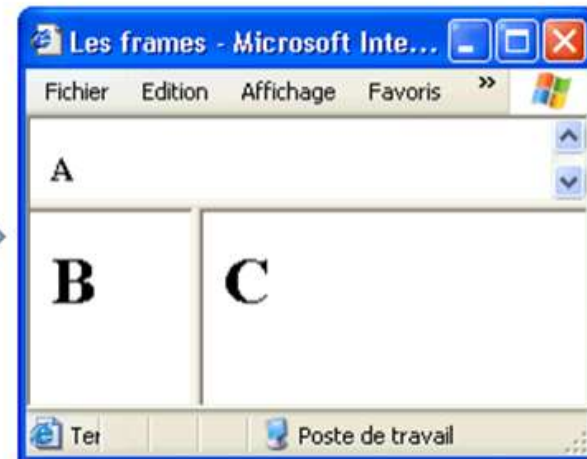
```
<html>
  <head>
    <title>Mon site</title>
  </head>
  <frameset cols="30%,70%">
    <frame src="Exemple6.html" name="gauche">
    <frame src="Exemple8.html" name="droite">
  </frameset>
</html>
```



Les frames « frameset »

- ◆ Les balises `<frameset>` peuvent être imbriquées.
- ◆ Exemples (Une page contenant trois zones : A, B et C):

```
<frameset rows="30%,70%">  
  <frame src="A.html" name="zoneA">  
  <frameset cols="30%,70%">  
    <frame src="B.html" name="zoneB">  
    <frame src="C.html" name="zoneC">  
  </frameset>  
</frameset>
```



Merci pour votre attention !!!

