

HTML : Hyper Text Markup Language

Achref El Mouelhi

Docteur de l'université d'Aix-Marseille
Chercheur en programmation par contrainte (IA)
Ingénieur en génie logiciel

`elmouelhi.achref@gmail.com`



Plan

- 1 Introduction
- 2 Outils
- 3 Concept de balises
- 4 Structure d'une page HTML
- 5 Attributs d'une balise
- 6 Organisation du texte
- 7 Titres

8 Listes

- `` **et** ``
- `` **et** ``
- type
- start
- `<dl>`, `<dt>` **et** `<dd>`

9 Images

10 Liens

- `<a>`
- download
- target
- `<base>`

11 Tableaux

- `<table>`, `<tr>` **et** `<td>`
- `<caption>` **et** `<th>`
- `colspan`
- `rowspan`
- `<thead>`, `<tbody>` **et** `<tfoot>`
- `<colgroup>` **et** `<col>`

12 Formulaires

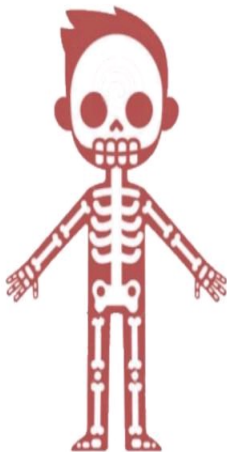
- Zones de saisie
- Zones de choix
- Boutons
- Barres de progression
- Autres balises/attributs pour les formulaires

13 Encore un peu de multimédia

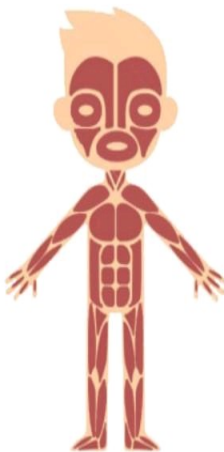
Plan

- 14 Objets
- 15 Classification des balises
- 16 Restructuration d'une page web avec HTML5
- 17 Caractères spéciaux
- 18 Attributs globaux
- 19 ARIA
- 20 Données
- 21 HTML5 et compatibilité des navigateurs
- 22 Conventions et bonnes pratiques
- 23 Code HTML5 valide

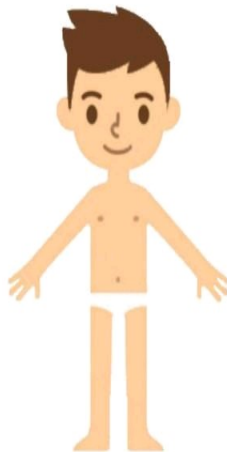
HTML



HTML



JavaScript



CSS

HTML

HTML : Hyper Text Markup Language

- Ce n'est pas un langage de programmation
- C'est plutôt un langage de description
 - composé de plusieurs balises (tags)
 - interprété par le navigateur
- Il ne nécessite pas un éditeur de texte particulier
- Il est créé en 1991 par Tim Berners-Lee et est standardisé depuis 1994 par W3C

En parlant Standardisation : deux organismes

- W3C (World Wide Web Consortium) :
 - organisme de standardisation fondé par Tim Berners-Lee
 - chargé de promouvoir la compatibilité des technologies web (HTML, XML, CSS, SOAP...)
- WHATWG (Web Hypertext Application Technology Working Group) :
 - collaboration non officielle des différents développeurs de navigateurs Web (Mozilla Foundation, Opera Software, Apple...)
 - chargé de développement de nouvelles technologies sur la base sur la base des implémentations actuelles d'Internet Explorer

HTML

Le 28 mai 2019

- Signature d'une collaboration entre W3C et WHATWG
- But : développement d'une version unique des spécifications HTML et DOM

HTML

HTML : évolution

- HTML1 : première version créée par Tim Berners-Lee en 1991.
- HTML2 : deuxième version, apparue en 1994. On commence à parler de W3C.
- HTML3 : apparue en 1996 avec plusieurs nouveautés comme les tableaux, les scripts, le positionnement du texte autour des images, etc.
- HTML4 : apparue en 1998 avec la possibilité :
 - d'utiliser de frames (découpage d'une page en plusieurs parties),
 - des améliorations sur les formulaires,
 - d'utiliser des feuilles de style (CSS).
- **HTML5** : finalisée en octobre 2014 et permet :
 - d'inclure facilement des vidéos,
 - d'ajouter plus de précisions sur les champs d'un formulaire

HTML

De quoi on a besoin ?

- un navigateur
- un éditeur de texte

Quelques navigateurs

- Google chrome : `https://www.google.com/chrome/`
- Mozilla firefox :
`https://www.mozilla.org/fr/firefox/new/`
- Edge (installé par défaut sous **Windows**)
- ...

Quelques éditeurs de texte

- **Sublime text** : <https://www.sublimetext.com/3>
- **Atom** : <https://atom.io/>
- **Notepad++** :
<https://notepad-plus-plus.org/download/v7.5.7.html>
- **Brackets** : <http://brackets.io/>
- ...

HTML

Quelques éditeurs de texte

- **Sublime text** : <https://www.sublimetext.com/3>
- **Atom** : <https://atom.io/>
- **Notepad++** :
<https://notepad-plus-plus.org/download/v7.5.7.html>
- **Brackets** : <http://brackets.io/>
- ...

CodePen : une solution en ligne

Trois éditeurs en parallèle : un pour HTML, un pour CSS et un pour JavaScript

<https://codepen.io/>

Utiliser un IDE (Environnement de développement intégré) ?

- Console auto-intégrée
- Auto-complétion
- Auto-compilation
- Coloration syntaxique
- Meilleure structuration du projet

HTML

Quel IDE pour **HTML** ?

- **Visual Studio Code**
- `code.visualstudio.com/download`

© Achref EL MOUËL

HTML

Quel IDE pour **HTML** ?

- **Visual Studio Code**
- `code.visualstudio.com/download`

Visual Studio Code (ou VSC) , pourquoi ?

- Gratuit
- Offrant la possibilité d'intégrer des éditeurs de texte connus (comme Sublime Text, Atom...)
- Pouvant s'adapter selon le langage de programmation
- Extensible via l'installation de quelques centaines d'extensions

Quelques recommandations (raccourcis) pour VSC

- Pour activer la sauvegarde automatique : aller dans `File > AutoSave`
- Pour indenter son code : `Alt Shift f`
- Pour faire une sélection multiple : `Ctrl f2`

Pour créer un projet sous VSC

- Allez dans `File > Open Folder...`
- Cliquez sur `Nouveau dossier` et saisissez `cours-html`
- Cliquez sur le dossier `cours-html` puis sur le dossier `Sélectionner un dossier`
- Créez un fichier `index.html` dans `cours-html`
- Dans `index.html`, saisissez `html : 5` puis cliquez sur `Entree`

HTML

Code généré

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
    scale=1.0">
  <title>Document</title>
</head>

<body>

</body>

</html>
```

Extension **Live Server**

- Installez l'extension **Live Server**
- Faites un clic droit sur `index.html`
- Cliquez sur `Open with Live Server`

HTML

Syntaxe

```
<balise>  
    objet  
</balise>
```

© Achref EL MOUELHI ©

HTML

Syntaxe

```
<balise>  
    objet  
</balise>
```

ou bien (les balises orphelines ou auto-fermantes)

```
[objet] <balise/>
```

HTML

Syntaxe

```
<balise>  
    objet  
</balise>
```

ou bien (les balises orphelines ou auto-fermantes)

```
[objet] <balise/>
```

Avec HTML5, plus besoin de / pour les balises orphelines

```
[objet] <balise>
```


HTML

Syntaxe

```
<balise>  
    objet  
</balise>
```

ou bien (les balises orphelines ou auto-fermantes)

```
[objet] <balise/>
```

Avec HTML5, plus besoin de / pour les balises orphelines

```
[objet] <balise>
```

Commentaire : balise particulière

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

HTML

Propriétés

- Les balises doivent être ouvertes puis fermées récursivement, comme des parenthèses ([...]{(...)})
- Tout ce qui est entouré par deux balises s'appelle élément
- Un élément peut éventuellement contenir du texte, liens, balises...
- Un texte en clair (non-entouré par une balise) est affiché
- Les balises non reconnues seront ignorées (non-affichées)
- Les commentaires en **HTML** sont ignorés par le navigateur
- Les commentaires ne sont pas affichés mais restent visibles dans le code source de la page et ne doit donc jamais contenir une information confidentielle.

HTML

Composition

- Le contenu d'une page **HTML** est compris entre deux balises `<html>`
- Une page **HTML** est composée de deux parties
 - une entête : `<head>`
 - un corps : `<body>`

© Achref EL

HTML

Composition

- Le contenu d'une page **HTML** est compris entre deux balises `html`
- Une page **HTML** est composée de deux parties
 - une entête : `head`
 - un corps : `body`

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
  </body>
</html>
```

HTML

DOCTYPE

- Ce n'est pas une balise
- C'est facultatif
- C'est une directive permettant de préciser qu'il s'agit d'un document **HTML** et indiquant sa version

```
<!-- HTML4.01 transitional -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
  Transitional//EN"
  "http://www.w3.org/TR/html4/loose.dtd">

<!-- HTML5 -->
<!DOCTYPE html>
```

Que peut contenir <HEAD> ?

```
<head>
  <meta charset="utf-8">
  <meta name="description" content="Formation HTML">
  <title>Formation 2017</title>
</head>
```

<head> : les méta informations

- <title> titre du document (affiché par le navigateur en haut de la page)
- <link> pour référencer un fichier (CSS par exemple)
- <style> pour inclure du code CSS
- <meta> peut contenir :
 - des informations sur le codage
 - des informations pour les moteurs de recherche

Que peut contenir `<body>` ?

```
<body>  
  les informations qui seront affichées dans le navigateur  
</body>
```

© Achref EL MOUELHI ©

Que peut contenir `<body>` ?

```
<body>  
  les informations qui seront affichées dans le navigateur  
</body>
```

Dans le `<Body>`

- texte
- tableau
- image/vidéo/document
- menu
- lien
- formulaire
- liste
- ...

Attribut

- Les balises, dans certains cas, ne suffisent pas.
- On leur associe donc des attributs.
- Certains attributs :
 - prennent une seule valeur
 - prennent plusieurs valeurs chacune attribuée à une clé
 - ne prennent pas de valeur

HTML

Exemple d'attribut acceptant une seule valeur

```
<input type=text>
```

© Achref EL MOUELHI ©

HTML

Exemple d'attribut acceptant une seule valeur

```
<input type=text>
```

Une balise peut avoir plusieurs attributs

```
<input type=text placeholder="saisir votre nom">
```

HTML

Exemple d'attribut acceptant une seule valeur

```
<input type=text>
```

Une balise peut avoir plusieurs attributs

```
<input type=text placeholder="saisir votre nom">
```

Exemple d'attribut n'acceptant pas de valeur

```
<input type=text placeholder="saisir votre nom" readonly>
```

HTML

Exemple d'attribut acceptant plusieurs valeurs

```
<p style="color: white; background-color: red">  
    Bonjour  
</p>
```

© Achref EL MOUL

HTML

Exemple d'attribut acceptant plusieurs valeurs

```
<p style="color: white; background-color: red">  
    Bonjour  
</p>
```

Remarques

Avec **HTML5**, la valeur d'un attribut peut être entourée par des guillemets, des apostrophes comme elle peut ne pas être entourée si elle ne contient pas d'espace.

Quelques attributs standards

- `class` : nom de classe (*à voir dans le cours CSS*)
- `id` : identifiant unique dans la page (*à voir dans le cours CSS*)
- `style` : style CSS de l'élément
- ...

Autres clés de `style` pour le format d'un texte

- `style="font-size:60%;"` : pour la taille du texte
- `style="text-align:center;"` : pour l'alignement du texte
- `style="font-family:arial;"` : pour le font du text
- ...

© AC

HTML

Autres clés de `style` pour le format d'un texte

- `style="font-size:60%;"` : pour la taille du texte
- `style="text-align:center;"` : pour l'alignement du texte
- `style="font-family:arial;"` : pour le font du text
- ...

Pour choisir une couleur

https://www.w3schools.com/colors/colors_picker.asp

Clés standards de l'attribut `style`

- `width` : pour la largeur d'un élément
- `height` : pour la hauteur d'un élément
- ...

HTML

`width` et `height` peuvent être utilisés comme attributs pour les balises suivantes

- `canvas`
- `embed`
- `iframe`
- `img`
- `input`
- `object`
- `video`

Les paragraphes

- `<p>...</p>` : pour délimiter un paragraphe
- `
` : pour aller à la ligne
- `...` : pour mettre un texte en gras
- `<u>...</u>` : pour souligner un texte
- `<i>...</i>` : pour mettre un texte en italic
- `<hr>` : pour afficher une ligne horizontale

HTML

Les paragraphes

- `<p>...</p>` : pour délimiter un paragraphe
- `
` : pour aller à la ligne
- `...` : pour mettre un texte en gras
- `<u>...</u>` : pour souligner un texte
- `<i>...</i>` : pour mettre un texte en italic
- `<hr>` : pour afficher une ligne horizontale

Il ne faut jamais oublier qu'on utilise **HTML** pour la structure et **CSS** pour le style

Pour organiser un texte, on peut plutôt utiliser

- `...` : pour mettre un texte bien en valeur
- `...` : pour mettre un texte un peu en valeur
- `<mark>...</mark>` : pour marquer un texte

Autres balises de formatage

- `<abbr>` : abréviation
- `<time>` : heure
- `<address>` : adresse formatée (en italique)
- `<cite>` : citation (en italique)
- `<code>` : élément de code informatique
- `<pre>` : texte pré-formaté
- `` : texte supprimé dans un document (barré)
- `<ins>` : texte inséré dans un document (souligné)
- `<sub>` : indice
- `<sup>` : exposant

Autres balises de formatage

- `<q>` : texte entre guillemets (quote)
- `<var>` : variable ou expression mathématique (écrite en italique)
- `<kbd>` : texte représentant un raccourci clavier
- `<dfn>` : définition
- ...

HTML

Exemple avec abbr

```
<p>Le <abbr title="World Wide Web Consortium">W3C</abbr></p>
```

© Achref EL MOUELHI ©

HTML

Exemple avec `abbr`

```
<p>Le <abbr title="World Wide Web Consortium">W3C</abbr></p>
```

Exemple avec `del` et `ins`

```
<p>La dernière version de HTML est <del>quatre</del><ins>cinq</ins></p>
```

© Achref EL MOUADHI ©

HTML

Exemple avec `abbr`

```
<p>Le <abbr title="World Wide Web Consortium">W3C</abbr></p>
```

Exemple avec `del` et `ins`

```
<p>La dernière version de HTML est <del>quatre</del><ins>cinq</ins></p>
```

Exemple avec `var` et `sup`

```
<p><var>x<sup>2</sup></var> = 0 => <var>x</var> = 0</p>
```

HTML

Exemple avec `abbr`

```
<p>Le <abbr title="World Wide Web Consortium">W3C</abbr></p>
```

Exemple avec `del` et `ins`

```
<p>La dernière version de HTML est <del>quatre</del><ins>cinq</ins></p>
```

Exemple avec `var` et `sup`

```
<p><var>x<sup>2</sup></var> = 0 => <var>x</var> = 0</p>
```

Exemple avec `kbd`

```
<p>Appuyez sur <kbd>Ctrl</kbd> + <kbd>C</kbd> pour copier sous Windows.  
</p>
```

Exemple avec details et summary

```
<details>
  <summary>Programmation par contraintes</summary>
  <p>
    un paradigme de programmation apparu dans les années
    1970 permettant de résoudre des problèmes
    combinatoires de grande taille (wiki).
  </p>
</details>
```

Règles générales

- Les navigateurs ajoutent automatiquement un espace entre chaque paragraphe
- Bien que le texte 'flottant' (non inclus dans un élément de la page) soit affiché par les navigateurs, il vaut mieux pour des raisons de 'style' inclure la totalité du texte utile de la page dans des balises, `<p>` notamment.

Les titres : de plus grand au plus petit

- `<h1>...</h1>`
- `<h2>...</h2>`
- `<h3>...</h3>`
- `<h4>...</h4>`
- `<h5>...</h5>`
- `<h6>...</h6>`

Les listes

- `...` : une liste non-ordonnée (non-numérotée)
- `...` : une liste ordonnée
- `...` : un élément d'une liste
- `<dl>...</dl>` : une liste de description
- `<dt>...</dt>` : un terme d'une liste
- `<dd>...</dd>` : la description d'un terme

HTML

Exemple d'une liste nom numérotée

```
<ul>  
  <li>Barcelone</li>  
  <li>Marseille</li>  
  <li>Manchester</li>  
</ul>
```

L'affichage

- Barcelone
- Marseille
- Manchester

HTML

Exemple d'une liste numérotée

```
<ol>
  <li>Barcelone</li>
  <li>Marseille</li>
  <li>Manchester</li>
</ol>
```

L'affichage (sans les puces)

- 1 Barcelone
- 2 Marseille
- 3 Manchester

HTML

Si on veut remplacer les chiffres par des lettres

```
<ol type="A">  
  <li>Barcelone</li>  
  <li>Marseille</li>  
  <li>Manchester</li>  
</ol>
```

Les valeurs possibles pour l'attribut type

- 1
- A
- a
- I
- i

HTML

Pour commencer d'un autre chiffre que 1

```
<ol start="5">  
  <li>Barcelone</li>  
  <li>Marseille</li>  
  <li>Manchester</li>  
</ol>
```

L'affichage (sans les puces)

- 5 Barcelone
- 6 Marseille
- 7 Manchester

HTML

Il est possible d'imbriquer les listes

```
<ul>
  <li>France</li>
  <li>Italie
    <ul>
      <li>Milan</li>
      <li>Turin</li>
    </ul>
  </li>
  <li>Espagne</li>
</ul>
```

L'affichage

- France
- Italie
 - Milan
 - Turin
- Espagne

HTML

Exemple d'une liste de description

```
<dl>  
  <dt>France</dt>  
  <dd> Bleu blanc rouge</dd>  
  <dt>Allemagne</dt>  
  <dd> Noir jaune rouge</dd>  
</dl>
```

L'affichage

France
 Bleu blanc rouge
Allemagne
 Noir jaune rouge

Les images

- La balise `` permet d'insérer soit
 - une image locale (``), ou
 - à distance (``)
- Cette balise a deux attributs obligatoires : `src` et `alt`
- Une image n'est pas 'incluse' dans un document. Elle est référencée par son adresse
- Si l'image référencée n'est pas accessible, le navigateur peut afficher soit une zone rectangulaire de bonnes dimensions, soit un texte alternatif

HTML

Les attributs recommandés pour garantir un affichage correct

- `src` : URL de l'image
- `alt` : texte à afficher si image indisponible (indispensable pour être **W3C Valid**)
- `height` : hauteur
- `width` : largeur



HTML

Les attributs recommandés pour garantir un affichage correct

- `src` : URL de l'image
- `alt` : texte à afficher si image indisponible (indispensable pour être **W3C Valid**)
- `height` : hauteur
- `width` : largeur

Exemple

```

```

Pour ajouter une légende à une image, on ajoute les deux balises `figure` **et** `figcaption`

```
<figure>  
    
  <figcaption>Légende de la figure.</figcaption>  
</figure>
```

Les liens

- `<a>...` : insérer un lien (interne ou externe)
- pour créer un lien vers un autre document local ou distant
` monsite ` ou
` Page 2 `
- pour pointer vers un signet ('name' ou 'id') dans un document
` vers le haut de page `
- href est l'attribut de l'élément `<a>` le plus important puisqu'il indique la destination du lien.

HTML

Lien pour télécharger un fichier

```
<a href="http://www.lsis.org/elmouelhia/courses/web/coursCSS.pdf" download="cours.pdf">
  Télécharger mon cours CSS
</a>
```

© Achref EL MOUELHI

HTML

Lien pour télécharger un fichier

```
<a href="http://www.lsis.org/elmouelhia/courses/web/coursCSS.pdf" download="cours.pdf">
  Télécharger mon cours CSS
</a>
```

Remarques

- La valeur de l'attribut `download` est optionnel.
- Par défaut, le nom du fichier téléchargé est égal à celui du fichier original.
- Ici, le fichier téléchargé aura le nom `cours.pdf`.
- L'attribut `download` fonctionne seulement si le fichier à télécharger a le même nom du domaine que l'application.

HTML

Lien pour ouvrir une nouvelle fenêtre

```
<a href="http://www.lsis.org/elmouelhia/" target="_blank">  
  Ma page  
</a>
```

HTML

Pour définir une base pour tous les liens de la page

```
<base href="http://www.lsis.org/elmouelhia/">
```

© Achref EL MOUËL

HTML

Pour définir une base pour tous les liens de la page

```
<base href="http://www.lsis.org/elmouelhia/">
```

Ainsi, on peut simplifier la création de liens

```
<a href="teaching.html">Mes cours</a>
```

En cliquant sur ce lien, on se redirige vers

<http://www.lsis.org/elmouelhia/teaching.html>

Les tableaux

- Les tables constituent un mode privilégié de présentation d'information structurée.
- Une table (balise `<table>`) est divisée en lignes et colonnes
- Une bordure délimite ou non les cellules de la table (Balise `<td>`), organisée en lignes (balise `<tr>`)
- La table préserve en permanence l'aspect visuel de colonnes dont toutes les cellules ont la même largeur.
- En revanche, les lignes peuvent avoir des hauteurs différentes

HTML

Quelques balises de tableau

- `<table>` : pour déclarer un tableau
- `<tr>` : pour déclarer une ligne dans un tableau (table row)
- `<td>` : pour déclarer une cellule dans un tableau (table data)

HTML

Exemple

```
<table border="1">
  <tr>
    <td>PHP</td>
    <td>10</td>
  </tr>
  <tr>
    <td>Java</td>
    <td>8</td>
  </tr>
</table>
```

HTML

Autres balises de tableau

- `<caption>` : pour ajouter une légende au tableau (texte associé)
- `<th>` : pour déclarer une cellule d'en-tête

HTML

Exemple

```
<table border="1">
  <caption>Moyenne par matière</caption>
  <tr>
    <th>Matière</th>
    <th>Moyenne</th>
  </tr>
  <tr>
    <td>PHP</td>
    <td>10</td>
  </tr>
  <tr>
    <td>Java</td>
    <td>8</td>
  </tr>
</table>
```

HTML

Fusionner les cellules d'une même colonne d'un tableau

```
<table border="1">
  <tr>
    <th>Nom</th>
    <th colspan="2">Mail</th>
  </tr>
  <tr>
    <td>Achref El Mouelhi</td>
    <td>achref@elmouelhi.fr</td>
    <td>elmouelhi@achref.fr</td>
  </tr>
</table>
```

HTML

Fusionner les cellules d'une même ligne d'un tableau

```
<table border="1">
  <tr>
    <th>Nom</th>
    <td>Achref El Mouelhi</td>
  </tr>
  <tr>
    <th rowspan="2">Mail</th>
    <td>achref@elmouelhi.fr</td>
  </tr>
  <tr>
    <td>elmouelhi@achref.fr</td>
  </tr>
</table>
```

HTML

Regrouper, par lignes, les éléments d'un tableau en utilisant `thead`, `tbody` et `tfoot`

```
<table>
  <thead>
    <tr>
      <th>Mois</th>
      <th>Loyer</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>January</td>
      <td>600$</td>
    </tr>
    <tr>
      <td>February</td>
      <td>700$</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Total</td>
      <td>1 300$</td>
    </tr>
  </tfoot>
</table>
```


HTML

Regrouper, par colonnes, les éléments d'un tableau en utilisant <colgroup> et <col>

```
<table border="1">
  <colgroup>
    <col style="background-color: blue">
    <col span=2 style="background-color: red">
  </colgroup>
  <tr>
    <td>Matière</td>
    <td>Coefficient</td>
    <td>Moyenne</td>
  </tr>
  <tr>
    <td>PHP</td>
    <td>2</td>
    <td>18</td>
  </tr>
  <tr>
    <td>Java</td>
    <td>3</td>
    <td>15</td>
  </tr>
</table>
```

HTML

L'attribut `style` de la balise `col` agit seulement sur les propriétés :

- `background-color`
- `border`
- `width`
- `visibility`

HTML

Autres clés de l'attribut `style` de la balise *table*

- `padding` : espace entre le contenu de la cellule et les frontières (la bordure)
- `border-spacing` : espace entre les cellules
- `width` : pour définir la largeur
- `height` : pour la hauteur
- `text-align` : pour l'alignement horizontal du texte dans une cellule (valeurs possibles : `left`, `right` et `center`)
- `vertical-align` (à utiliser dans la balise `td`) : pour l'alignement vertical du texte dans une cellule (valeurs possibles : `top`, `bottom` et `middle`)
- ...

Remarques

- Les tableaux ont été largement utilisés dans le passé pour organiser les pages des sites web.
- Cet usage est aujourd'hui obsolète et absolument découragé.
- On utilise pour cela les balises `<div>` et les styles **CSS** qui seront vus plus loin.

HTML

Déclaration d'un formulaire

```
<form method="POST ou GET" action="page web destination">  
</form>
```

© Achref EL MOUELHI ©

HTML

Déclaration d'un formulaire

```
<form method="POST ou GET" action="page web destination">  
</form>
```

Les attributs d'un formulaire

- `method` : concerne l'envoi de données et peut prendre deux valeurs.
 - `GET` : non fréquemment utilisée car limitée à 2048 caractères. En plus, les informations envoyées seront visibles dans la zone d'adresse.
 - `POST` : plus utilisée que `GET` car elle permet d'envoyer un grand nombre d'informations et les données saisies dans le formulaire ne transitent pas par la barre d'adresse.
- `action` : indique l'adresse de la page ou du programme qui va traiter les informations (généralement avec un langage autre que **HTML**).

Que peut-on avoir dans un formulaire ?

- Des zones de saisie
- Des zones de choix
- Des boutons

HTML

Une zone de saisie monoligne

```
<input type="text" name="nom zone texte">
```

© Achref EL MOUELHI ©

HTML

Une zone de saisie monoligne

```
<input type="text" name="nom zone texte">
```

Une zone de saisie multiligne

```
<textarea name="nom zone texte" id="identifiant"></textarea>
```

© Achref EL MOUËL

HTML

Une zone de saisie monoligne

```
<input type="text" name="nom zone texte">
```

Une zone de saisie multiligne

```
<textarea name="nom zone texte" id="identifiant"></textarea>
```

Un libellé associé à une zone de saisie

```
<label for="identifiant zone saisie">texte</label>
```

HTML

Une zone de saisie monoligne

```
<input type="text" name="nom zone texte">
```

Une zone de saisie multiligne

```
<textarea name="nom zone texte" id="identifiant"></textarea>
```

Un libellé associé à une zone de saisie

```
<label for="identifiant zone saisie">texte</label>
```

Exemple

```
<label for="nom">Nom :</label> <input type="text" name="nom" id="nom">
```

HTML

Une zone de saisie monoligne

```
<input type="text" name="nom zone texte">
```

Une zone de saisie multiligne

```
<textarea name="nom zone texte" id="identifiant"></textarea>
```

Un libellé associé à une zone de saisie

```
<label for="identifiant zone saisie">texte</label>
```

Exemple

```
<label for="nom">Nom :</label> <input type="text" name="nom" id="nom">
```

Généralement on associe la même valeur aux attributs `id` et `name`

HTML

Avec HTML 5, les 3 écritures suivantes sont équivalentes

```
<input type=text value=John>
```

```
<input type="text" value="John Wick">
```

```
<input type='text' value='John Wick'>
```

HTML

Avant HTML 5

password : pour les mots de passe

© Achref EL MOUELHI ©

HTML

Avant HTML 5

`password` : pour les mots de passe

Avec HTML 5

- `email` : pour les adresses emails
- `number` : pour les nombres (attributs `min`, `max` et `step`)
- `color` : pour les couleurs
- `date` : pour les calendriers
- ...

HTML

Trois types de zones de choix

- Les cases à cocher
- Les boutons radio
- Les listes déroulantes

HTML

Cases à cocher

```
<input type="checkbox" name="nom du choix" [checked]>
```

© Achref EL MOUELHI ©

HTML

Cases à cocher

```
<input type="checkbox" name="nom du choix" [checked]>
```

Boutons radios

```
<input type="radio" name="nom du choix" [checked]>
```

© Achref EL MELHI ©

HTML

Cases à cocher

```
<input type="checkbox" name="nom du choix" [checked]>
```

Boutons radios

```
<input type="radio" name="nom du choix" [checked]>
```

Listes déroulantes

```
<select name="liste">  
<option value="valeur1" [selected]>valeur1</option>  
...  
<option value="valeurN">valeurN</option>  
</select>
```

HTML

Les listes déroulantes (le regroupement)

```
<select name="liste">
<optgroup label="label1">
<option value="valeur1">valeur1</option>
    ...
<option value="valeur1">valeurN</option>
</optgroup>
<optgroup label="label2">
<option value="valeur1">valeur1</option>
    ...
<option value="valeur1">valeurM</option>
</optgroup>
    ...
</select>
```

HTML

Les listes de données (datalist) = liste déroulante + zone de saisie

```
<input list="sports" name="sport">
```

```
<datalist id="sports">  
  <option value="football">  
  <option value="handball">  
  <option value="cross fit">  
  <option value="tennis">  
  <option value="hockey">  
</datalist>
```

HTML

Trois types de boutons

- `submit` : envoie le contenu d'un formulaire à la page indiqué dans l'action
- `reset` : remet à zéro (efface) le contenu d'un formulaire
- `button` : bouton générique qui permet de réaliser plusieurs tâches pour **HTML** (quitter une page,...) ou de déclencher un code **JavaScript**

HTML

Remarque

Tout bouton déclaré dans un formulaire avec la balise `<button>` est de type `submit`

© Achref EL MOUELFI

HTML

Remarque

Tout bouton déclaré dans un formulaire avec la balise `<button>` est de type `submit`

Pour déclarer un bouton simple (pas de type `submit`) dans une balise `form`

```
<input type="button" value="cliquer">
```


HTML

Remarque

Tout bouton déclaré dans un formulaire avec la balise `<button>` est de type `submit`

Pour déclarer un bouton simple (pas de type `submit`) dans une balise `form`

```
<input type="button" value="cliquer">
```

Ou

```
<button type=button>cliquer</button>
```

HTML

Un bouton de type `submit` peut être aussi déclaré avec la balise `input`

```
<input type="submit" value="envoyer">
```

HTML

Un bouton de type `submit` peut être aussi déclaré avec la balise `input`

```
<input type="submit" value="envoyer">
```

Ou tout simplement (uniquement dans un formulaire)

```
<button>envoyer</button>
```

HTML

Deux types de barre de progression

- `progress` : barre de progression dynamique représentant l'avancement d'une tâche.
- `meter` : barre de progression statique représentant une jauge.

HTML

Exemple avec `progress`

```
<label for="compte">Création de compte :</label>  
<progress id="compte" value="60" max="100"> 60% </progress>
```

© Achref EL MOUËL

HTML

Exemple avec `progress`

```
<label for="compte">Création de compte :</label>  
<progress id="compte" value="60" max="100"> 60% </progress>
```

Exemple avec `meter`

```
<label for="disk">Espace mémoire utilisé : </label>  
<meter id="disk" value="120" min="0" max="500">120 sur 500 GO</meter>
```

HTML

Quelques attributs ne prenant pas de valeurs

- `required` : pour indiquer qu'un champ est obligatoire
- `autofocus` : pour placer le curseur dans cet élément dès chargement de la page
- `autocomplete` : pour indiquer si on autorise auto-complétion
- `multiple` : pour indiquer qu'il est possible de choisir ou d'insérer plusieurs éléments
- `placeholder` : pour afficher un message indicatif dans un champ
- `readonly` : pour rendre le champ en lecture seule

HTML

Exemple

```
<form method="POST" action="traitement.php">
  <fieldset>
    <legend>Nom complet</legend> <!-- partie nom prénom -->
    <label for="nom">nom :</label>
    <input type="text" id="nom"><br>
    <label for="prenom">prénom :</label>
    <input type="text" id="prenom">
  </fieldset>

  <fieldset>
    <legend>Sexe</legend> <!-- partie sexe -->
    <input type="radio" name="homme" value="homme">
    <label for="homme"> Homme </label><br>
    <input type="radio" name="femme" value="femme">
    <label for="femme"> Femme </label>
  </fieldset>
</form>
```


HTML

Insérer un élément audio

```
<audio src="audio.mp3"></audio>
```

© Achref EL MOUELHI ©

HTML

Insérer un élément audio

```
<audio src="audio.mp3"></audio>
```

Attention

Les navigateurs ne supportent pas tous les formats audios existants

HTML

Insérer un élément audio

```
<audio src="audio.mp3"></audio>
```

Attention

Les navigateurs ne supportent pas tous les formats audios existants

Les attributs possibles

- `controls` : pour afficher les boutons lecture et pause
- `loop` : pour jouer le fichier audio en boucle
- `autoplay` : pour lire le contenu du fichier dès le chargement de la page

HTML

Insérer un élément vidéo

```
<video src="fichier.mp4"></video>
```

© Achref EL MOUELHI ©

HTML

Insérer un élément vidéo

```
<video src="fichier.mp4"></video>
```

Attention

Les navigateurs ne supportent pas tous les formats vidéos existants

HTML

Insérer un élément vidéo

```
<video src="fichier.mp4"></video>
```

Attention

Les navigateurs ne supportent pas tous les formats vidéos existants

Pour éviter ce problème : définir plusieurs formats

```
<video controls>  
  <source src="fichier.mp4">  
  <source src="fichier.webm">  
  <source src="fichier.ogv">  
</video>
```

HTML

Une deuxième solution consiste à

- héberger la vidéo sur **YouTube**
- copier l'identifiant de la vidéo
- utiliser ce dernier dans la balise `iframe`

HTML

Pour les vidéos YouTube, on peut utiliser la balise `iframe`

```
<iframe width="400" height="300" src="https://www.  
youtube.com/embed/8cm1x4bC610">  
</iframe>
```


Les attributs possibles

- `poster` : image à afficher à la place de la vidéo
- `controls` : pour afficher les boutons lecture et pause
- `loop` : pour jouer le fichier vidéo en boucle
- `autoplay` : pour lire le contenu du fichier dès le chargement de la page

HTML

object

balise permettant d'intégrer plusieurs types de contenu dans un document HTML (compatible avec tous les navigateurs mais **dépréciée** depuis 2015)

- une autre page HTML
- PDF
- flash
- image
- vidéo
- audio

HTML

object

balise permettant d'intégrer plusieurs types de contenu dans un document HTML (compatible avec tous les navigateurs mais **dépréciée** depuis 2015)

- une autre page HTML
- PDF
- flash
- image
- vidéo
- audio

Remarque

Pour les audios et vidéos, il est recommandé d'utiliser les balises `<audio>` et `<video>`

HTML

Considérons la page `fichier.html`

```
<!DOCTYPE html>
<html>
<head>
  <title>Fichier</title>
</head>
<body style="background-color:teal">
  Hello everybody
</body>
</html>
```

© Achref EL

HTML

Considérons la page `fichier.html`

```
<!DOCTYPE html>
<html>
<head>
  <title>Fichier</title>
</head>
<body style="background-color:teal">
  Hello everybody
</body>
</html>
```

Il est possible d'inclure `fichier.html` dans une autre page `index.html`

```
<!DOCTYPE html>
<html>
<head>
  <title>Index</title>
</head>
<body>
  <p> Bonjour tout le monde </p>
  <object width="400" height="200" data="fichier.html"></object>
</body>
</html>
```

Remarque

La balise orpheline `<embed>` permet également d'inclure un document HTML, une image ou un flash (n'est pas compatible avec tous les navigateurs et dépréciée depuis 2015)

HTML

inline et block

- `block` : c'est une balise qui ajoute automatiquement un retour à la ligne avant et après.
- `inline` : c'est une balise qui se situe obligatoirement à l'intérieur d'une balise `block`. Elle ne crée pas de retour à la ligne, le texte qui se trouve à l'intérieur s'écrit donc à la suite du texte précédent, sur la même ligne.

Exemple

- `block` : `<p>`, `<h1>`...
- `inline` : `<a>`...

Les balises universelles

Ce sont des balises sans aucune sémantique, utilisées souvent en CSS ou dans les formulaires.

- `...` : inline
- `<div>...</div>` : block

Quelques propriétés de balise `inline`

- Il ignore les marges `top` et `bottom` mais applique les marges `left` et `right`, ainsi que tout `padding`.
- Il ignore les propriétés `width` et `height`.

HTML

Les balises structurales

- `<header>...</header>` : l'entête de la page
- `<footer>...</footer>` : le pied de la page
- `<main>...</main>` : tout le reste de la page
- `<nav>...</nav>` : l'emplacement du menu
- `<section>...</section>` : le main peut être composé de plusieurs sections
- `<article>...</article>` : dans une section on peut définir un ou plusieurs articles
- ...

HTML

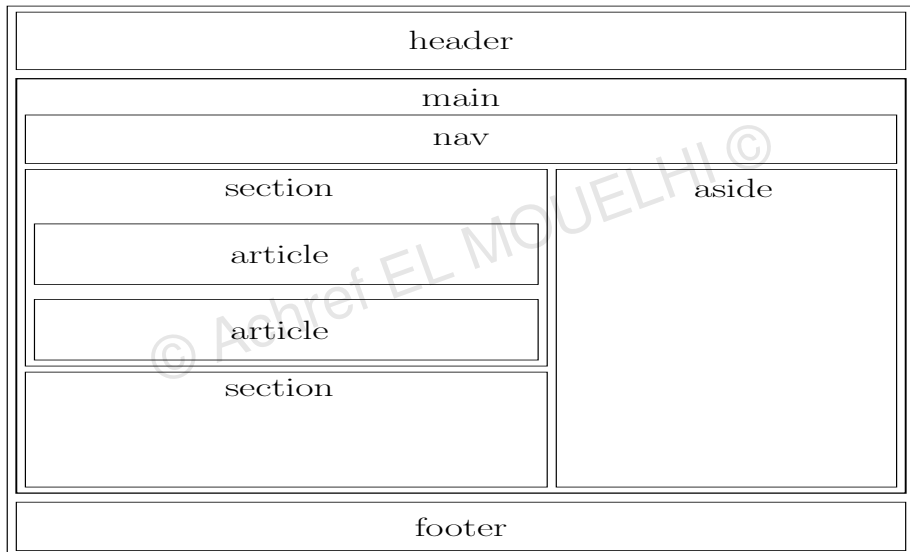
Les balises structurales

- `<header>...</header>` : l'entête de la page
- `<footer>...</footer>` : le pied de la page
- `<main>...</main>` : tout le reste de la page
- `<nav>...</nav>` : l'emplacement du menu
- `<section>...</section>` : le main peut être composé de plusieurs sections
- `<article>...</article>` : dans une section on peut définir un ou plusieurs articles
- ...

Remarque

- Ne pas confondre les deux balises `<head>` et `<header>`.
- Ces balises sont plutôt sémantiques et n'ont pas de positions fixes.

HTML



HTML

Le codage de caractères spéciaux

- `´` : é
- ``` : è
- `ê` : ê
- `ç` : ç
- `&` : &
- `æ` : æ
- `×` : ×
- `"` : "
- `<` : <
- `>` : >
- ...

HTML

Le codage de caractères spéciaux

- `´` : é
- ``` : è
- `ê` : ê
- `ç` : ç
- `&` : &
- `æ` : æ
- `×` : ×
- `"` : "
- `<` : <
- `>` : >
- ...

Autres caractères spéciaux

https://www.w3schools.com/charsets/ref_utf_latin1_supplement.asp

HTML

Quelques attributs globaux

- `id`
- `class`
- `style`
- `draggable`
- `lang`
- `hidden`
- `title` (le texte qui s'affiche en survolant un élément **HTML**)
- `accesskey` (le raccourci **Windows** permettant de mettre placer le cursus sur l'élément **HTML**)
- ...

HTML

Sous Windows, appuyez sur `alt shift e` pour accéder à ma page ou `alt shift f` pour accéder à France Football

```
<a href="http://www.lsis.org/elmouelhia" accesskey="e" title="
  alt shift e">
  Ma page
</a>
<br>
<a href="http://www.francefootball.fr" accesskey="f" title="alt
  shift f">
  France Football
</a>
```


HTML

Pour souligner les fautes d'orthographe, on peut utiliser l'attribut `spellcheck` (cliquer dans la zone de saisie et vérifier que `annees` est souligné)

```
<textarea spellcheck="true">  
    Les meilleures années, les annees 80...  
</textarea>
```

© Achref EL MOU

HTML

Pour souligner les fautes d'orthographe, on peut utiliser l'attribut `spellcheck` (cliquer dans la zone de saisie et vérifier que `annees` est souligné)

```
<textarea spellcheck="true">
    Les meilleures années, les annees 80...
</textarea>
```

S'il ne s'agit pas d'une zone de saisie, on ajoute l'attribut `contenteditable` (cliquer sur le paragraphe et vérifier que le contenu est éditable)

```
<p spellcheck="true" contenteditable>
    Les meilleures années, les annees 80...
</p>
```

Accessibilité dans le développement web

- Rendre accessible un contenu web
 - aux personnes avec des handicapes visuels et/ou auditifs
 - quels que soient leurs dispositifs d'accès (ordinateur, smart-phone, tablette...)
- **Web Accessibility Initiative (WAI)** : ensemble de normes sur l'accessibilité définies par W3C

ARIA

- **A**ccessible **R**ich **I**nternet **A**pplications (applications internet riches et accessibles)
- Ensemble d'attributs HTML
- Permettant de rendre accessible le contenu d'une application web en utilisant quelques outils d'assistance

Quelques outils d'assistance

- **ChromeVox** (pour ajouter comme extensions sous Google Chrome <https://chrome.google.com/webstore/detail/chromevox-classic-extensi/kgejglhpjiefppelpmljglcjbhoiplfn?hl=fr>)
- **Jaws**
- **VoiceOver**

L'attribut `role`

- permet d'ajouter une sémantique à nos balises
- accepte un ensemble de valeurs prédéfinies

© Achref EL MOUELHI ©

L'attribut `role`

- permet d'ajouter une sémantique à nos balises
- accepte un ensemble de valeurs prédéfinies

Exemples de rôle

- Rôles de structure : `main`, `navigation`, `article`, `heading`...
- Rôles de composant graphique : `alert`, `button`, `link`, `radio`...
- Rôles composés
 - `listbox` **et** `option`
 - `menubar` **et** (`menuitem` **ou** `menuitemcheckbox` **ou** `menuitemradio`)
 - ...

Exemple 1 avec un rôle de composant graphique

```
<div role=link>  
  <a href="#">visitez ma page</a>  
</div>
```

© Achref EL MOU

Exemple 1 avec un rôle de composant graphique

```
<div role=link>  
  <a href="#">visitez ma page</a>  
</div>
```

Exemple 2 avec un rôle de composant graphique

```
<h2 role="alert">  
  Votre formulaire ne peut être soumis, veuillez  
  remplir les champs obligatoires.  
</h2>
```

Exemple avec un rôle de structure

```
<nav role=navigation>  
    ...  
</nav>
```

© Achref EL MOUËL

HTML

Exemple avec un rôle de structure

```
<nav role=navigation>  
    ...  
</nav>
```

Exemple avec des rôles composés

```
<nav role=navigation>  
    <ul role=menubar>  
        <li role=menuitem>Accueil</li>  
        <li role=menuitem>Formation</li>  
    </ul>  
</nav>
```

Remarque

ARIA recommande de privilégier l'utilisation d'élément HTML natif et de ne pas modifier la sémantique d'une balise...

© Achref EL MOUËL

Remarque

ARIA recommande de privilégier l'utilisation d'élément HTML natif et de ne pas modifier la sémantique d'une balise...

Ne jamais faire ça

```
<h1 role="button">titre transformé en bouton</h1>
```

Remarque

ARIA recommande de privilégier l'utilisation d'élément HTML natif et de ne pas modifier la sémantique d'une balise...

Ne jamais faire ça

```
<h1 role="button">titre transformé en bouton</h1>
```

Cependant, il est possible d'écrire

```
<h1><button>titre contenant un bouton</button></h1>
```

L'attribut `aria-*`

- permet d'ajouter une sémantique à nos balises
- * à remplacer par quelques propriétés
- accepte comme valeur un texte personnalisé

© Achref EL W.

HTML

L'attribut `aria-*`

- permet d'ajouter une sémantique à nos balises
- `*` à remplacer par quelques propriétés
- accepte comme valeur un texte personnalisé

Exemples de propriétés

- propriétés de composant graphique : `aria-label`, `aria-valuemax`, `aria-required`...
- propriétés de relation : `aria-labelledby`, `describedby`...

Exemple 1 avec une propriété de composant graphique

```
<div>  
  <div id=prenom>Prénom</div>  
  <input type=text aria-required=true>  
</div>
```

© Achref EL MOU

HTML

Exemple 1 avec une propriété de composant graphique

```
<div>  
  <div id=prenom>Prénom</div>  
  <input type=text aria-required=true>  
</div>
```

Exemple 2 avec une propriété de composant graphique

```
<div>  
  <div id=age>Age</div>  
  <input type=number aria-valuemin=0 aria-valuemax  
    =150 >  
</div>
```

Remarques

- Les propriétés de composant graphique `aria-valuemax`, `aria-required`, etc ont été introduites avant HTML5
- HTML5 a intégré quelques propriétés comme `required`, `maxlength`, `max`...

HTML

Exemple 3 avec une propriété de composant graphique

```
<div role=link>  
  <a aria-label="pour visiter la page de john wick" href="#">visitez ma  
    page</a>  
</div>
```

© Achref EL MOUELHI ©

HTML

Exemple 3 avec une propriété de composant graphique

```
<div role=link>  
  <a aria-label="pour visiter la page de john wick" href="#">visitez ma  
    page</a>  
</div>
```

Exemple avec une propriété de relation (`aria-labelledby` permet d'indiquer les identifiants des éléments qui labellisent l'objet)

```
<div id="teacher">Enseignant</div>  
  
<div>  
  <div id="nom">Nom</div>  
  <input type="text" aria-labelledby="nom teacher">  
</div>  
  
<div>  
  <div id="prenom">Prénom</div>  
  <input type="text" aria-labelledby="prenom teacher">  
</div>
```

HTML

En cas de présence de `aria-labelledby` **et** `aria-label`,
`aria-label` **sera ignoré**

```
<div id=teacher>Enseignant</div>
```

```
<div>
```

```
  <div id=nom>Nom</div>
```

```
  <input type="text" aria-label="zone de saisie"  
    aria-labelledby="nom teacher">
```

```
</div>
```

```
<div>
```

```
  <div id=prenom>Prénom</div>
```

```
  <input type=text aria-labelledby="prenom teacher">
```

```
</div>
```

Exemple avec `aria-describedby`

```
<div>
  <div id="prenom">Prénom</div>
  <input type="text" aria-required=true aria-
    describedby="description" >
</div>

<div id="description" hidden>
  Le prénom est un champ obligatoire, il faut saisir
  au moins un caractère
</div>
```

HTML

L'attribut `data-*`

- permettant d'associer des données à un élément HTML
- disponible depuis HTML5

© Achref EL MOUËL

HTML

L'attribut data-*

- permettant d'associer des données à un élément HTML
- disponible depuis HTML5

Exemple

```

```

Le code suivant n'est pas HTML5 valide

```

```

HTML

Pour tester la compatibilité d'un élément HTML5 avec les navigateurs

`https://caniuse.com/`

HTML

Conventions et bonnes pratiques

- HTML n'est pas sensible à la casse mais il est recommandé d'écrire le nom des balises et attributs en minuscule
- En HTML5, il n'est plus nécessaire de fermer les balises orphelines . Donc, ces deux écritures sont équivalentes :
`<input />` et `<input >`
- Contrairement aux langages de programmation, il est recommandé en HTML5 de ne pas mettre d'espace avant et après l'opérateur =
- Éviter d'écrire des lignes de code de plus de 80 caractères
- En HTML5, les balises `<html>` et `<body>` ne sont pas obligatoires mais il est recommandé de les utiliser

HTML

Conventions et bonnes pratiques

- En HTML5, la balise `<head>` n'est pas obligatoire (mais recommandée) et son contenu (par exemple la balise `<title>`) peut être déclaré juste avant `body`
- En HTML5, la balise `<title>` est recommandée
- En HTML5, il est recommandé d'indiquer la langue utilisée dans la page `<html lang="en-US">` pour aider les moteurs de recherche
- En HTML5, il est recommandé de donner un nom en minuscule aux fichiers.
- En HTML5, une nouvelle balise a été introduite afin d'ajuster l'affichage en fonction de l'écran
`<meta name="viewport" content="width=device-width, initial-scale=1.0">`

HTML

Pour tester la validité d'un code HTML5

<https://validator.w3.org/>

© Achref EL W...

HTML

Pour tester la validité d'un code HTML5

<https://validator.w3.org/>

Extension **VSC** pour la validité du code

W3C Validation