

# HTML 5

**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

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

## 9 Images

## 10 Favicon

## 11 Liens

- `<a>`
- download
- target
- **Mail et Téléphone**
- `<base>`

## 12 Tableaux

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

## 13 Formulaires

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

## 14 Encore un peu de multimédia

- `<audio>`
- `<video>`

## 15 `<iframe>`

## 16 Objets

## 17 Classification des balises

## 18 Restructuration d'une page web avec HTML 5

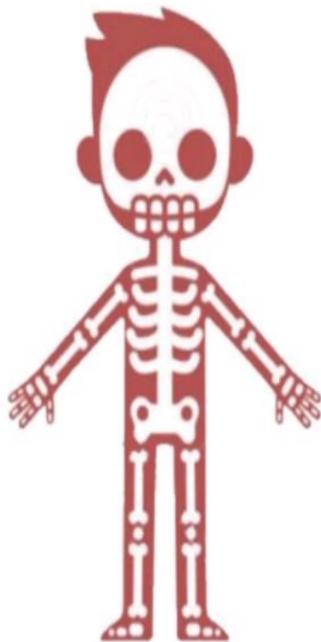
## 19 Caractères spéciaux

## 20 Émojis

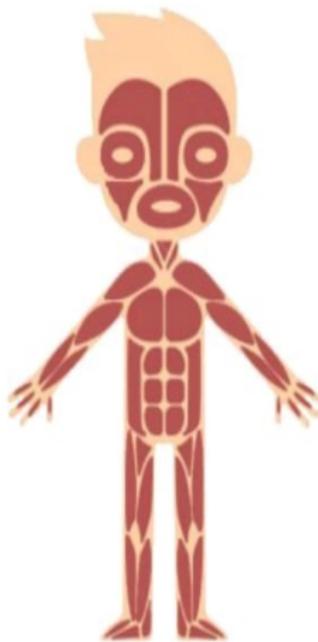
# Plan

- 21 Attributs globaux
- 22 ARIA
- 23 Données
- 24 HTML 5 et compatibilité des navigateurs
- 25 Conventions et bonnes pratiques
- 26 Pour un meilleur référencement
- 27 Code HTML 5 valide
- 28 Liens utiles

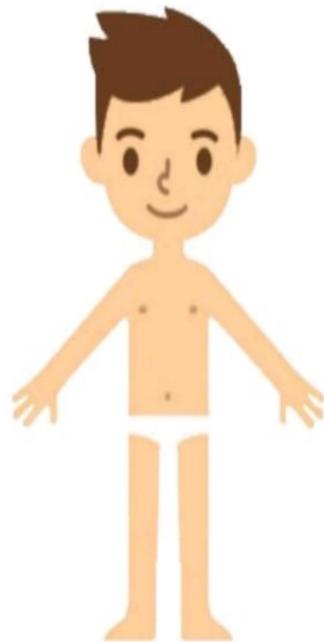
# HTML



**HTML**



**JavaScript**



**CSS**

## HTML : Hyper Text Markup Language

- Ce n'est pas un langage de programmation
- C'est 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)** :
  - Groupe de travail formé en 2004 : collaboration non officielle des différents développeurs de navigateurs Web (**Mozilla Foundation, Opera Software, Apple...**)
  - Initialement créé pour travailler sur le développement du **HTML** et du **DOM (Document Object Model)**
  - Activités actuellement étendues aux **API WEB**

## Remarque

**HTML 5** : travail collaboratif entre **W3C** et **WHATWG**.

© Achref EL MOUETRI

## Remarque

**HTML 5** : travail collaboratif entre **W3C** et **WHATWG**.

Le 28 mai 2019

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

## Quelques projets créés par WHATWG

- **Fetch API**
- **IndexedDB**
- **Service Workers**
- ...

## HTML : évolution

- **HTML 1** : première version créée par Tim Berners-Lee en 1991.
- **HTML 2** : deuxième version, apparue en 1994. On commence à parler de **W3C**.
- **HTML 3** : apparue en 1996 avec plusieurs nouveautés comme les tableaux, les scripts, le positionnement du texte autour des images, etc.
- **HTML 4** : 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**).
- **HTML 5** : finalisée en octobre 2014 et permet :
  - d'inclure facilement des vidéo/audio,
  - d'ajouter plus de précisions sur les champs d'un formulaire,
  - d'ajouter les balises sémantiques.

## 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/>
- **Brackets** : <http://brackets.io/>
- ...

© Acti

# HTML

## Quelques éditeurs de texte

- **Sublime text** : <https://www.sublimetext.com/3>
- **Atom** : <https://atom.io/>
- **Notepad++** : <https://notepad-plus-plus.org/>
- **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

## Quel IDE pour HTML ?

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

© Achref EL MOUËLMI

# HTML

## Quel IDE pour HTML ?

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

## Visual Studio Code (ou VSC) , pourquoi ?

- Gratuit
- Multi-langage
- Multi-système d'exploitation
- Extensible via l'installation de quelques centaines d'extensions

## Quelques raccourcis VSC

- Pour activer la sauvegarde automatique : aller dans `File > AutoSave`
- Pour indenter son code : `Alt` `Shift` `f`
- Pour commenter/décommenter : `Ctrl` `:`
- Pour sélectionner toutes les occurrences : `Ctrl` `f2`
- Pour sélectionner l'occurrence suivante : `Ctrl` `d`
- Pour placer le curseur dans plusieurs endroits différents : `Alt`

## 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` ou `!` puis cliquez sur `Entrée`

## 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>
```

## Quelques extensions utiles

- **Live Server** : pour la mise à jour de la page dans le navigateur après chaque modification
- **Live Share** : pour avoir accès à une instance de **Visual Studio Code** en temps réel
- **Indent Rainbow** : pour une meilleure visualisation de l'indentation

## 3 solutions pour ouvrir un fichier **HTML** dans le navigateur

- Ouvrir le fichier directement dans le navigateur
- Utiliser le terminal intégré
- Installer et utiliser l'extension **Live Server**

## Première solution : ouvrir le fichier directement dans le navigateur

- Ouvrir l'explorateur de fichiers de votre système d'exploitation.
- Naviguer jusqu'au répertoire contenant votre fichier **HTML**.
- Double-cliquer sur le fichier **HTML** pour l'ouvrir dans votre navigateur par défaut.

## Deuxième solution : utiliser le terminal intégré

- Accédez à la vue du terminal intégré en sélectionnant `Terminal > New Terminal` dans le menu ou en utilisant le raccourci `Ctrl` `+` (ou `Cmd` `+`) sur **macOS**.
- Dans le terminal, utilisez une des commandes suivantes pour ouvrir le fichier **HTML** dans votre navigateur par défaut :
  - `open index.html` sur **macOS**
  - `start index.html` sur **Windows**
  - `xdg-open index.html` sur **Linux**

### Troisième solution : installer et utiliser l'extension **Live Server**

- Accéder à l'onglet des extensions en cliquant sur l'icône des extensions dans la barre latérale gauche ou en appuyant sur `Ctrl` `Shift` `X`.
- Rechercher une extension comme **Live Server**
- Installer l'extension **Live Server**
- Faire un clic droit sur `index.html`
- Cliquer 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/>
```

© Achref EL MUELHI ©

# HTML

## Syntaxe

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

## ou bien (les balises orphelines ou auto-fermantes)

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

## Avec HTML 5, 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 HTML 5, plus besoin de / pour les balises orphelines

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

Commentaire : balise particulière

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

## 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 M

# 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>
```

## DOCTYPE : Document Type Declaration

- 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

© Achre

# HTML

## DOCTYPE : Document Type Declaration

- 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

### Pour HTML 5

```
<!-- HTML 5 -->  
<!DOCTYPE html>
```

# HTML

Pour HTML 4, on doit ajouter DTD (Document Type Definition) : une spécification formelle qui définit la structure, la syntaxe et les règles de validation

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

<!-- HTML 4.01 strict -->
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">

<!-- XHTML 1.0 strict -->
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

## 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

## 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>
```

## Exemple d'attribut acceptant plusieurs valeurs

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

© Achref EL MOUËZ

## Exemple d'attribut acceptant plusieurs valeurs

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

### Remarques

Avec **HTML 5**, la valeur d'un attribut peut être entourée par des " ", des ' ' 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
- ...

© Acti

# 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](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
- `<br>` : pour aller à la ligne
- `<b>...</b>` : 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

## Les paragraphes

- `<p>...</p>` : pour délimiter un paragraphe
- `<br>` : pour aller à la ligne
- `<b>...</b>` : 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

- `<strong>...</strong>` : pour mettre un texte bien en valeur
- `<em>...</em>` : 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
- `<del>` : 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 MOUADJIB ©

# 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

- `<ul>...</ul>` : une liste non-ordonnée (non-numérotée)
- `<ol>...</ol>` : une liste ordonnée
- `<li>...</li>` : 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

## 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

# HTML

## Terminologie

- dl : **D**escription **L**ist
- dt : **D**escription **T**erm
- dd : **D**escription **D**etails

## Les images

- La balise `<img >` 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

## 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

## Exemple

```

```

© Achref EL MOUELHI ©

# HTML

## Exemple

```

```

On peut aussi définir les dimensions avec l'attribut `style`

```

```

© Achref EL... ELHI ©

# HTML

## Exemple

```

```

On peut aussi définir les dimensions avec l'attribut `style`

```

```

## Remarques

- Les attributs `width`, `height` et `style` sont tous valides en **HTML**.
- Cependant, il est recommandé d'utiliser l'attribut `style` si vous souhaitez empêcher les feuilles de styles de modifier la taille des images.

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

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

# HTML

Avec `map` et `area`, on peut définir des zones cliquables

```
  
  
<map name="mymap">  
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="https://consumer.huawei.com/">  
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="https://www.oppo.com">  
  <area shape="circle" coords="337,300,44" alt="Coffee" href="https://www.nespresso.com">  
</map>
```

© Achref EL MOUËL

# HTML

Avec `map` et `area`, on peut définir des zones cliquables

```
  
  
<map name="mymap">  
  <area shape="rect" coords="34,44,270,350" alt="Computer" href="https://consumer.huawei.com/">  
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="https://www.oppo.com">  
  <area shape="circle" coords="337,300,44" alt="Coffee" href="https://www.nespresso.com">  
</map>
```

## Valeurs de l'attribut `shape`

- `poly`
- `rect`
- `circle`
- `default` (toute l'image)

# HTML

On peut aussi modifier la source de l'image en fonction de la taille de l'écran

```
<picture>
  <source media="(min-width: 700px)" srcset="http://elmouelhia.free.fr/courses/web/luigi.png">
  <source media="(max-width: 500px)" srcset="http://elmouelhia.free.fr/courses/web/mario.png">
  
</picture>
```

© Achref EL MOUËL

# HTML

On peut aussi modifier la source de l'image en fonction de la taille de l'écran

```
<picture>  
  <source media="(min-width: 700px)" srcset="http://elmouelhia.free.fr/courses/web/luigi.png">  
  <source media="(max-width: 500px)" srcset="http://elmouelhia.free.fr/courses/web/mario.png">  
    
</picture>
```

## Image affichée

- écran avec largeur minimale = 700px ⇒ luigi.png
- écran avec largeur maximale = 500 ⇒ mario.png
- sinon : yoshi.png

## Favicon

- Image de petite taille
- Affiché à coté du titre de la page
- Pouvant avoir comme extension : **PNG, ICO, GIF, JPEG, SVG**
- Générateur de Favicon : <https://www.favicon.cc/>

© ADP

# HTML

## Favicon

- Image de petite taille
- Affiché à coté du titre de la page
- Pouvant avoir comme extension : **PNG, ICO, GIF, JPEG, SVG**
- Générateur de Favicon : <https://www.favicon.cc/>

## Exemple

```
<link rel="icon" type="image/x-icon" href="/favicon.ico">
```

## Les liens

- `<a>...</a>` : insérer un lien (interne ou externe)
- pour créer un lien vers un autre document local ou distant  
`<a href="http://monsite.fr"> monsite </a>` ou  
`<a href="page2.html"> Page 2 </a>`
- pour pointer vers un signet (' name' ou ' id' ) dans un document  
`<a href="#top"> vers le haut de page </a>`
- `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://elmouelhia.free.fr/courses/web/coursCSS.pdf"
  download="cours.pdf">
  Télécharger mon cours CSS
</a>
```

© Achref EL MOUELHI

# HTML

## Lien pour télécharger un fichier

```
<a href="http://elmouelhia.free.fr/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://elmouelhia.free.fr/" target="_blank">  
  Ma page  
</a>
```

© Achref EL MOUËLTI

# HTML

## Lien pour ouvrir une nouvelle fenêtre

```
<a href="http://elmouelhia.free.fr/" target="_blank">  
  Ma page  
</a>
```

### Autres valeurs de l'attribut `target`

- `_self` (par défaut)
- `_top` et `_parent` à utiliser avec `frameset` et `frame` qui sont dépréciés (<https://developer.mozilla.org/fr/docs/Web/HTML/Element/frame>)

# HTML

**Pour créer un lien vers une adresse email, utiliser `mailto` dans `href`**

```
<a href="mailto:elmouelhi@gmail.com">mail</a>
```

© Achref EL MOUELHI ©

# HTML

**Pour créer un lien vers une adresse email, utiliser `mailto` dans `href`**

```
<a href="mailto:elmouelhi@gmail.com">mail</a>
```

**Pour créer un lien vers plusieurs adresses email**

```
<a href="mailto:achref@gmail.com, elmouelhi@gmail.com">mail</a>
```

© Achref MOUËLHI ©

# HTML

**Pour créer un lien vers une adresse email, utiliser `mailto` dans `href`**

```
<a href="mailto:elmouelhi@gmail.com">mail</a>
```

**Pour créer un lien vers plusieurs adresses email**

```
<a href="mailto:achref@gmail.com, elmouelhi@gmail.com">mail</a>
```

**Pour créer un lien vers un numéro de téléphone, utiliser `tel` dans `href`**

```
<a href="tel:+33612345678">Appeler</a>
```

# HTML

**Pour définir une base pour tous les liens de la page**

```
<base href="http://elmouelhia.free.fr/">
```

© Achref EL MOUELHI ©

# HTML

**Pour définir une base pour tous les liens de la page**

```
<base href="http://elmouelhia.free.fr/">
```

**Ainsi, on peut simplifier la création de liens**

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

# HTML

**Pour définir une base pour tous les liens de la page**

```
<base href="http://elmouelhia.free.fr/">
```

**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://elmouelhia.free.fr/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>
```

## 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>
```

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

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

## 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
- ...

## 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 M. ELHI ©

# 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

## Exemple

```
<select name="sport" id="sport">  
  <option value="foot">Football</option>  
  <option value="tennis">Tennis </option>  
  <option value="basket">Basket</option>  
</select>
```

# HTML

## Exemple avec une option qu'on ne peut sélectionner

```
<select name="sport" id="sport">
  <option selected hidden>Choisissez votre sport</option>
  <option value="foot">Football</option>
  <option value="tennis">Tennis </option>
  <option value="basket">Basket</option>
</select>
```

© Achref EL MOU

# HTML

## Exemple avec une option qu'on ne peut sélectionner

```
<select name="sport" id="sport">
  <option selected hidden>Choisissez votre sport</option>
  <option value="foot">Football</option>
  <option value="tennis">Tennis </option>
  <option value="basket">Basket</option>
</select>
```

Ou

```
<select name="sport" id="sport">
  <option selected disabled>Choisissez votre sport</option>
  <option value="foot">Football</option>
  <option value="tennis">Tennis </option>
  <option value="basket">Basket</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>
```

## 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**

## Remarque

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

© Achref EL MOUELHANI

# 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">
```

© Achref EL MOUADJIB

# 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>
```

## 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 MOUËZ

# 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 autres attributs/propriétés

- `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 l'auto-complétion
- `multiple` : pour indiquer qu'il est possible de choisir ou d'insérer plusieurs éléments
- `placeholder` : pour afficher un message indicatif (grisé) dans un champ
- `readonly` : pour rendre un champ en lecture seule
- `disabled` : pour désactiver un bouton

# HTML

**Pour désactiver l'autocomplete (activé par défaut)**

```
<input type="text" name="nom" autocomplete="off" autofocus>
```

© Achref EL MOUËL

# HTML

**Pour désactiver l'`autocomplete` (activé par défaut)**

```
<input type="text" name="nom" autocomplete="off" autofocus>
```

**Exemple avec `disabled` et `readonly`**

```
<input type="text" name="nom" value="doe" readonly>  
<button disabled>cliquer</button>
```

# 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>Genre</legend> <!-- partie genre -->
    <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>
```

## Remarque

L'attribut `autocapitalize` est utilisé pour contrôler la capitalisation automatique du texte et accepte les valeurs suivantes :

- `none` : Aucune capitalisation automatique n'est appliquée. Le texte reste tel que l'utilisateur l'a entré.
- `sentences` : La première lettre de chaque phrase est automatiquement mise en majuscule. Les phrases sont généralement définies par un point suivi d'un espace.
- `words` : La première lettre de chaque mot est automatiquement mise en majuscule.
- `characters` : Toutes les lettres sont automatiquement mises en majuscule.

# 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

© Achref EL

# 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

© Achref EL

# 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>
```

## 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

## 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>
```

© Achref EL MOU

# HTML

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

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

Le code précédent peut être obtenu en suivant les étapes suivantes

- 1 cliquer sur le bouton `Partager` en dessous de la vidéo **YouTube**
- 2 cliquer ensuite sur `Intégrer`
- 3 copier le code généré pour `iframe`

iframe peut aussi être utilisé pour afficher un document PDF

```
<iframe
  src="http://elmouelhia.free.fr/courses/web/coursHTML.pdf"
  width="600"
  height="800">
</iframe>
```

<iframe> peut aussi être utilisé pour intégrer des

- pages **HTML**
- documents **Word**
- feuilles de calcul **Excel**
- présentations **PowerPoint**
- ...

# 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 W.

# 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.

- `<span>...</span>` : 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

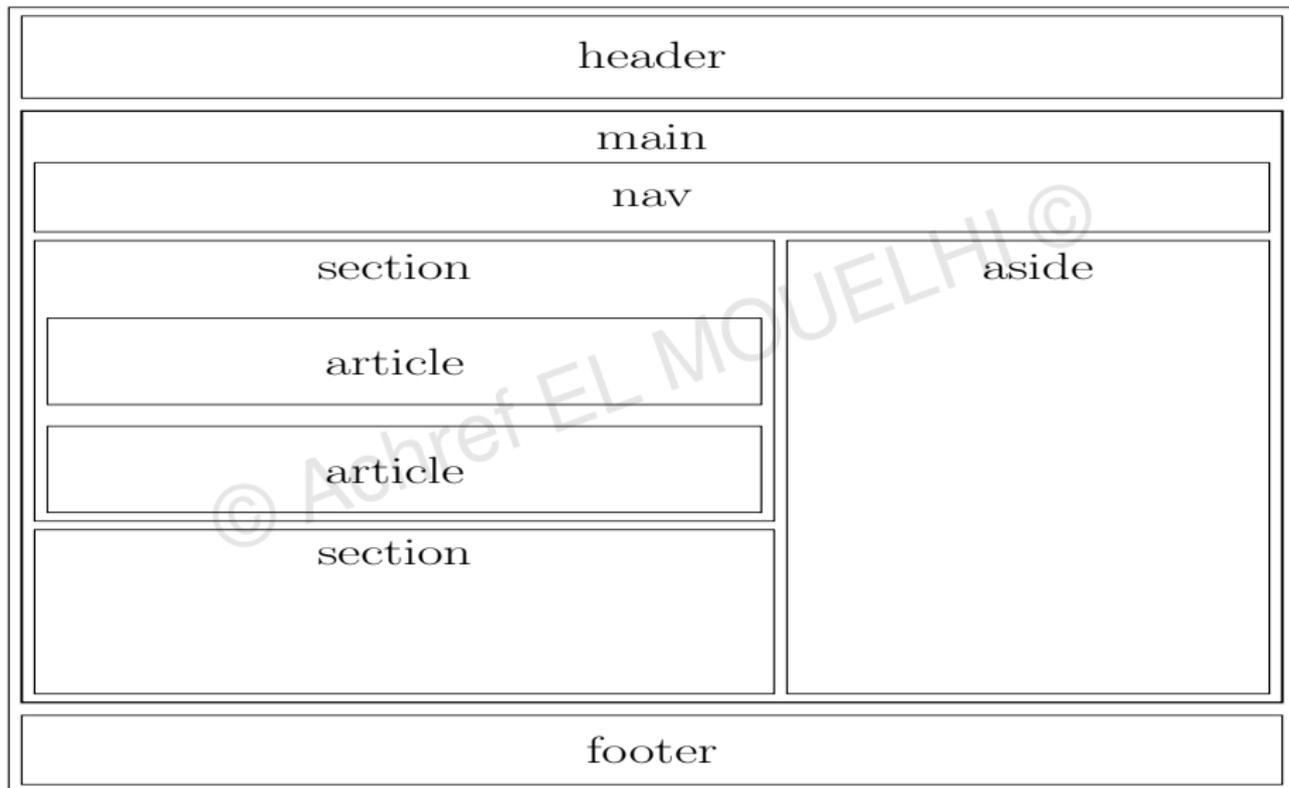
## Les balises structurales

- `<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

- `&acute;` : é
- `&grave;` : è
- `&ecirc;` : ê
- `&ccedil;` : ç
- `&amp;` : &
- `&aelig;` : æ
- `&times;` : ×
- `&quot;` : ”
- `&lt;` : <
- `&gt;` : >
- ...

# HTML

## Le codage de caractères spéciaux

- `&acute;` : é
- `&grave;` : è
- `&ecirc;` : ê
- `&ccedil;` : ç
- `&amp;` : &
- `&aelig;` : æ
- `&times;` : ×
- `&quot;` : ”
- `&lt;` : <
- `&gt;` : >
- ...

## Autres caractères spéciaux

[https://www.w3schools.com/charsets/ref\\_utf\\_latin1\\_supplement.asp](https://www.w3schools.com/charsets/ref_utf_latin1_supplement.asp)

## Émojis

- Caractères spéciaux appartenant à **UTF-8** (**U**niversal **C**haracter **S**et **T**ransformation **F**ormat - 8 bits)
- Ce sont ni images ni icônes
- Considérées comme des lettres
- Liste non-complète : [https://www.w3schools.com/charsets/ref\\_emoji.asp](https://www.w3schools.com/charsets/ref_emoji.asp)

© Achille

# HTML

## Émojis

- Caractères spéciaux appartenant à **UTF-8** (**U**niversal **C**haracter **S**et **T**ransformation **F**ormat - 8 bits)
- Ce sont ni images ni icônes
- Considérées comme des lettres
- Liste non-complète : [https://www.w3schools.com/charsets/ref\\_emoji.asp](https://www.w3schools.comCharsets/ref_emoji.asp)

## Exemple

```
<p>  
I &#128151; HTML &#128525;  
</p>
```

## Le codage de caractères spéciaux

- `&acute;` : é
- `&grave;` : è
- `&ecirc;` : ê
- `&ccedil;` : ç
- `&amp;` : &
- `&aelig;` : æ
- `&times;` : ×
- `&quot;` : ”
- `&lt;` : <
- `&gt;` : >
- ...

# HTML

## Le codage de caractères spéciaux

- `&acute;` : é
- `&grave;` : è
- `&ecirc;` : ê
- `&ccedil;` : ç
- `&amp;` : &
- `&aelig;` : æ
- `&times;` : ×
- `&quot;` : ”
- `&lt;` : <
- `&gt;` : >
- ...

## Autres caractères spéciaux

[https://www.w3schools.com/Charsets/ref\\_utf\\_latin1\\_supplement.asp](https://www.w3schools.comCharsets/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 placer le curseur 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://elmouelhia.free.fr" accesskey="e" title="alt shift e">
  Ma page
</a>
<br>
<a href="http://www.franceinfo.fr" accesskey="f" title="alt shift f">
  France Info
</a>
```

© Achrel L

# 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://elmouelhia.free.fr" accesskey="e" title="alt shift e">
  Ma page
</a>
<br>
<a href="http://www.franceinfo.fr" accesskey="f" title="alt shift f">
  France Info
</a>
```

Raccourci sous **Mac** pour tester le code précédent

`Ctrl` + `Alt` + `e` et `Ctrl` + `Alt` + `f`

# HTML

**Pour rendre un contenu éditable, on ajoute l'attribut `contenteditable` (cliquez sur le paragraphe et vérifiez que le contenu est éditable)**

```
<p contenteditable>  
    Les meilleures années, les années 80...  
</p>
```

© Achref EL

# HTML

**Pour rendre un contenu éditable, on ajoute l'attribut `contenteditable` (cliquez sur le paragraphe et vérifiez que le contenu est éditable)**

```
<p contenteditable>  
    Les meilleures années, les années 80...  
</p>
```

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

```
<p spellcheck="true" contenteditable>  
    Les meilleures années, les années 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 MOULI

## 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 MOUËLTAH

# 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 MOUËLHANI

## 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 MO

## 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 MOUL

# 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 **HTML 5**
- **HTML 5** 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 labellent 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 HTML 5

© Achref EL MOUËLHI

# HTML

## L'attribut `data-*`

- permettant d'associer des données à un élément HTML
- disponible depuis HTML 5

## Exemple

```

```

## Le code suivant n'est pas HTML 5 valide

```

```

## Pour tester la compatibilité d'un élément **HTML 5** avec les navigateurs

- <https://caniuse.com/> : pour déterminer la compatibilité des navigateurs avec une balise, attribut...
- <https://html5test.co/> : pour déterminer la compatibilité d'un navigateur avec les balises et attributs **HTML 5**

## 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 **HTML 5**, 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 **HTML 5** de ne pas mettre d'espace avant et après l'opérateur =
- Éviter d'écrire des lignes de code de plus de 80 caractères
- En **HTML 5**, les balises `<html>` et `<body>` ne sont pas obligatoires mais il est recommandé de les utiliser

## Conventions et bonnes pratiques

- En **HTML 5**, la balise `<head>` n'est pas obligatoire (mais recommandée) et son contenu (par exemple la balise `<title>`) peut être déclaré juste avant `body`
- En **HTML 5**, la balise `<title>` est recommandée
- En **HTML 5**, il est recommandé d'indiquer la langue utilisée dans la page `<html lang="en-US">` pour aider les moteurs de recherche
- En **HTML 5**, il est recommandé de donner un nom en minuscule aux fichiers.
- En **HTML 5**, 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">
```

## Référencement **SEO**

- **Search Engine Optimization** : ce qui se traduit littéralement par "optimisation pour les moteurs de recherche"
- Ensemble des pratiques et des techniques
- Permettant d'améliorer la visibilité et le classement d'un site web dans les résultats des moteurs de recherche, tels que **Google, Bing, Yahoo**
- Englobant des techniques d'optimisation du contenu, architecture du site, gestion des liens (backlinks), amélioration de la vitesse de chargement...

## Structure de la page

- Utilisez une structure sémantique en utilisant des balises **HTML** appropriées comme `<header>`, `<nav>`, `<main>`, `<article>`, `<section>`, `<footer>`...
- Organisez votre contenu de manière logique avec un titre principal (`<h1>`) suivi de sous-titres (`<h2>`, `<h3>`...) pour hiérarchiser l'information.

© Achref EL

## Structure de la page

- Utilisez une structure sémantique en utilisant des balises **HTML** appropriées comme `<header>`, `<nav>`, `<main>`, `<article>`, `<section>`, `<footer>`...
- Organisez votre contenu de manière logique avec un titre principal (`<h1>`) suivi de sous-titres (`<h2>`, `<h3>`...) pour hiérarchiser l'information.

## Balise de titre (`<title>`)

- Chaque page doit avoir une balise de titre unique et descriptive.
- Incluez des mots-clés pertinents dans le titre.

## Images

- Utilisez des attributs `alt` pour toutes les images, en fournissant une description pertinente pour chaque image.
- Comprimez les images pour des temps de chargement rapides.

© Achref EL MOU

# HTML

## Images

- Utilisez des attributs `alt` pour toutes les images, en fournissant une description pertinente pour chaque image.
- Comprimez les images pour des temps de chargement rapides.

## Liens internes et externes

- Utilisez des liens internes pour relier vos pages entre elles, en veillant à ce que les liens aient un texte d'ancrage descriptif.
- Veillez à ce que les liens externes s'ouvrent dans une nouvelle fenêtre/onglet.

## Contenu de qualité

- Écrivez un contenu de haute qualité, informatif et utile pour les utilisateurs.
- Évitez le contenu dupliqué, et assurez-vous que chaque page a un contenu unique.
- Utilisez des mots-clés pertinents dans votre contenu, mais évitez le bourrage de mots-clés.
- ...

# HTML

Pour tester la validité d'un code **HTML 5**

`https://validator.w3.org/`

© Achref EL MOUADIB

# HTML

Pour tester la validité d'un code **HTML 5**

`https://validator.w3.org/`

Extension **VSC** pour la validité du code

W3C Validation

## Pour continuer l'apprentissage

- **W3Schools** : <https://www.w3schools.com/html/>
- **Mozilla** : <https://developer.mozilla.org/fr/docs/Learn/HTML>
- ...