

# ***Web Academy***



## **Structuration de page web en *HTML***

*Alexis EL MRINI (@alexis-elmrini)*  
*Tristan LE GODAIS (@TrisTOON)*

# Sommaire

I. Syntaxe d'un nœud .....	4
1. Nœuds .....	4
2. Attributs .....	6
3. Échappement de caractères .....	8
II. Contenu d'un document <i>HTML</i> .....	9
1. Corps de texte .....	9
2. Formatage de texte (à la <i>HTML4</i> ) .....	10

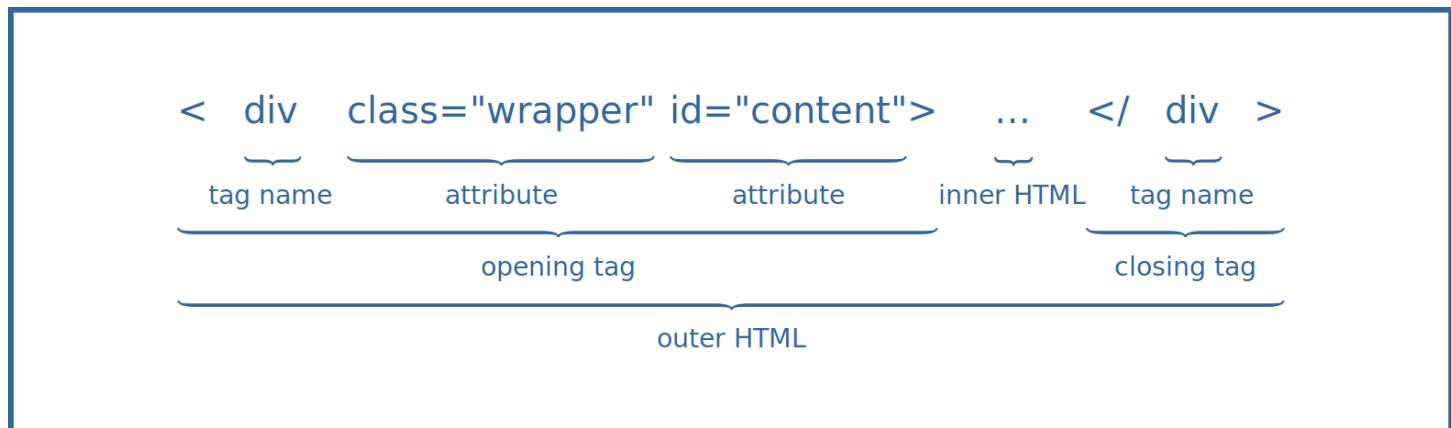
---

3. Balisage sémantique de texte .....	12
4. Références à d'autres documents .....	13
5. Structures composites plus complexes .....	15
6. Formulaire .....	19
7. Sections .....	22
III. Métadonnées d'un document <i>HTML</i> .....	25
IV. Structure générale d'un document <i>HTML</i> valide .....	27

---

# I. Syntaxe d'un nœud

## 1. Nœuds



*Anatomie d'un élément*

- Deux écritures selon le type d'élément :
  - Balises par paire : `<p class="nice">Bonjour</p>`
  - Balises auto-fermantes (éléments sans contenu généralement) : ``
- Commentaires : `<!-- ceci est un commentaire -->`

## 2. Attributs

- Attributs universels apposables à n'importe quel élément :
  - `class="..."` et `id="..."` pour la sélection et l'identification (en *CSS* et *JS*)
  - `style="..."` pour du style *CSS*
  - `onclick="..."`, `onerror="..."`... pour des gestionnaires d'événements *JS*
  - `dir="..."`, `lang="..."`, `title="..."`, `tabindex="..."`...

- Attributs spécifiques selon le type d'élément :
  - `width="..."`, `height="..."`, `src="..."` et `alt="..."` pour les éléments `<img/>`...
  - `href="..."` et `download="..."`... pour les éléments `<a>...</a>`  
...
  - `type="..."`, `name="..."`, `value="..."`... pour les éléments `<input/>` et `<button/>`...
  - `for="..."` pour les éléments `<label>...</label>`...
  - `colspan="..."` et `rowspan="..."` pour les éléments `<th>...</th>` et `<td>...</td>`...

### 3. Échappement de caractères

Caractère	"	&	'	<	>
Entité	&quot;;	&amp;;	&apos;;	&lt;;	&gt;;

*Séquences d'échappement héritées de XML*

- Quelques centaines d'autres entités prédéfinies existent en *HTML* (&nbsp;;, &copy;;, &pi;;...)
- Possibilité de donner directement le point de code *Unicode* en décimal (&#dddd;) ou hexadécimal (&#xhhhh;)



## II. Contenu d'un document *HTML*

### 1. Corps de texte

*HTML* sert à structurer un document (un livre, un cours, un CV, une page *web*...), notamment à l'aide de :

- Balises de titres : `<h1>...</h1>` de haut niveau à de `<h6>...</h6>` de bas niveau
- Balises de paragraphe : `<p>...</p>`
- Balise de saut de ligne : `<br/>` (au cours d'un paragraphe)
- Balise de règle horizontale : `<hr/>` (changement de sujet)

## 2. Formatage de texte (à la *HTML4*)

`<i>...</i>`

- Italique

`<b>...</b>`

- Gras

`<u>...</u>`

- Soulignage

`<s>...</s>`

- Biffage (raturage)

```
<sub>...</sub>
```

- Texte en indice

```
<sup>...</sup>
```

- Texte en exposant

```
<div>...</div>
```

```
<span>...</span>
```

- Conteneurs de dernier recours, sans aucune valeur sémantique, à réserver notamment pour la mise en page avec CSS

### 3. Balisage sémantique de texte

```
<em>...</em>
```

- Emphase (insistance)

```
<strong>...</strong>
```

- Forte emphase (importance)

```
<ins>...</ins>
```

- Insertion de texte

```
<del>...</del>
```

- Suppression de texte

## 4. Références à d'autres documents

Un document *HTML* peut faire référence à du contenu externe (une autre page web, une image, une vidéo...) via une *URL* notamment sous la forme de :

- Lien hypertexte : `<a href="..." target="...">...</a>` avec les attributs :
  - `href="..."` qui donne l'adresse (l'*URL*) d'un document ou d'une partie d'un document (un identifiant d'élément *HTML*)
  - `target="..."` qui donne le contexte dans lequel ouvrir le lien : l'onglet courant (`_top`), un nouvel onglet (`_blank`), le contexte courant (`_self`) ou le contexte parent (`_parent`)

- Image intégrée : `` avec les attributs :
  - `src="..."` qui donne l'adresse (l'*URL*) de l'image
  - `alt="..."` qui donne le texte alternatif équivalent pour les utilisateurs n'ayant pas accès aux images ou en cas d'erreur de chargement

## 5. Structures composites plus complexes

- Liste non ordonnée (avec puces) :

---

```
<ul>
```

```
  <li>...</li>
```

```
  <li>...</li>
```

```
  <li>...</li>
```

```
  <li>...</li>
```

```
</ul>
```

---

Chaque item est délimité par un élément `<li>...</li>`

- Liste ordonnée (avec numérotation) :

---

```
<ol>
```

```
<li>...</li>
```

```
<li>...</li>
```

```
<li>...</li>
```

```
</ol>
```

---

Chaque item est délimité par un élément `<li>...</li>`

L'attribut `start="..."` d'une liste ordonnée permet de spécifier le numéro du premier item

L'attribut `reversed="..."` d'une liste ordonnée permet de la numéroter de manière descendante



- Tableau de données :

```
<table>
  <tr>
    <th>...</th>
    <th>...</th>
    <th>...</th>
  </tr>
  <tr>
    <th>...</th>
    <td>...</td>
    <td>...</td>
  </tr>
</table>
```

Chaque ligne est délimitée par un élément `<tr>...</tr>`

Chaque cellule est délimitée par un élément `<td>...</td>` pour les cellules de données ou `<th>...</th>` pour les cellules d'en-tête

L'attribut `rowspan="..."` d'une cellule permet d'indiquer le nombre de lignes qu'elle occupe

L'attribut `colspan="..."` d'une cellule permet d'indiquer le nombre de colonnes qu'elle occupe

## 6. Formulaire

```
<textarea name="...">...</textarea>
```

- Champ de texte multiligne

```
<input name="..." value="..." />
```

- Champ de texte sur une ligne par défaut...
- ...ou alors un bouton radio, une boîte à cocher, un champ de date, un nuancier...

```
<button name="..." value="...">...</button>
```

- Bouton de soumission de formulaire par défaut

```
<label>...</label>
```

- Légende d'un contrôle (champ de texte, bouton...)

Deux manières de l'y associer :

- En plaçant le contrôle à l'intérieur de la légende :

```
<label>Prénom : <input/></label>
```

- En précisant son identifiant dans l'attribut `for="..."`

```
<label for="firstname">Prénom :</label>  
<input id="firstname"/>
```

```
<form action="..." method="...">...</form>
```

- Formulaire :

```
<form action="https://ddg.co/" method="get">  
  <label>Termes : <input name="q"/></label>  
  <button>Rechercher</button>  
</form>
```

L'attribut `action="..."` d'un formulaire permet de spécifier l'adresse à laquelle envoyer les données du formulaire

L'attribut `method="..."` d'un formulaire permet de spécifier la méthode *HTTP* utilisée pour soumettre le formulaire (`get` ou `post`)

## 7. Sections

Un document *HTML* peut être hiérarchisé via l'utilisation des différents niveaux de titres (de `<h1>...</h1>` à `<h6>...</h6>`), mais aussi de sections :

- Article : `<article>...</article>` représentant un contenu autonome pouvant être extrait du document à des fins de *syndication* (un article, un commentaire d'utilisateur...)
- Aparté : `<aside>...</aside>` représentant un contenu tangentiel pouvant être considéré séparément du reste du document (un encart publicitaire, une anecdote, une note...)

- Section de navigation : `<nav>...</nav>` représentant un groupement de liens de navigation (un menu, une table des matières...)
- Section générique : `<section>...</section>` représentant tout autre groupement thématique (un chapitre, une sous-partie, un onglet...)

Les différentes portions du contenu d'une section peuvent être éventuellement regroupées à l'aide de :

- En-tête de section : `<header>...</header>` pour des titres, un logo, un formulaire de recherche...
- Pied de section : `<footer>...</footer>` pour des informations sur l'auteur, les sources, la license...
- Corps de section : `<main>...</main>` pour le contenu principal du document (cet élément **ne doit pas apparaître plus d'une fois** dans le document)...



## III. Métadonnées d'un document *HTML*

Un document *HTML* peut être décrit et enrichi par un ensemble d'informations générales (métadonnées) renseignées à l'aide de :

- Titre de document : `<title>...</title>` apparaissant dans l'onglet de la page
- Feuille de style : `<style>...</style>` décrivant la mise en forme en via CSS
- Programme : `<script>...</script>` permettant d'exécuter du *JS*

- Lien vers une ressource : `<link/>` permettant d'afficher une *favicon*, d'indiquer un moteur de recherche... :

```
<link rel="favicon" href="./favicon.png"/>  
<link rel="search" href="./opensearch.xml"/>  
...
```

- Métadonnée générique : `<meta/>` définissant tout autre type de métadonnée comme l'encodage, le *viewport*... :

```
<meta charset="utf-8"/>  
<meta name="viewport" content="width=480"/>  
...
```

## IV. Structure générale d'un document *HTML* valide

Tout document *HTML* doit commencer avec la déclaration `<!DOCTYPE html>` afin de signifier au navigateur qu'il est en codé en *HTML*, dans sa dernière version.

La déclaration doit être suivie de l'élément racine `<html>...</html>`.

L'attribut `dir="..."` permet de spécifier le sens de lecture (`ltr` ou `rtl`).

L'attribut `lang="..."` permet de spécifier la langue (`fr`, `en`...).

L'élément racine `<html>...</html>` contient dans cet ordre :

- L'en-tête du document : `<head>...</head>` regroupant les métadonnées du document (`<title>...</title>`, `<style>...</style>`, `<script>...</script>`, `<link/>`, `<meta/>`...)
- Le corps du document : `<body>...</body>` représentant le contenu du document (sections, titres, paragraphes, listes, tableaux de données...)

```
<!DOCTYPE html>
<html dir="ltr" lang="fr">
  <head>
    <title>Page d'exemple</title>
    <meta charset="utf-8"/>
    <meta name="viewport" content="width=device-width"/>
  </head>
  <body>
    <h1>Page d'exemple</h1>
    <p>Bienvenue sur cette page d'exemple.</p>
  </body>
</html>
```

*Exemple de document HTML valide*