

# Introduction au WEB

David Hébert  
hebert.iut@gmail.com

2020



# Table des matières

---

<b>Table des matières</b>	<b>1</b>
<b>1 Débuter</b>	<b>2</b>
<b>2 Premières balises</b>	<b>3</b>
<b>3 Gestion des caractères : Code HTML</b>	<b>4</b>
<b>4 Attributs</b>	<b>5</b>
<b>5 Liste et tableaux</b>	<b>8</b>
<b>6 Liens</b>	<b>10</b>
<b>7 Images</b>	<b>13</b>
<b>8 Audio et vidéo</b>	<b>14</b>
<b>9 Mise en forme avec le CSS</b>	<b>16</b>
<b>10 Formatage du texte</b>	<b>18</b>
<b>11 Boites</b>	<b>21</b>
<b>12 Formulaires</b>	<b>24</b>
<b>13 Hacking</b>	<b>27</b>
<b>14 Pseudo-classe et pseudo-éléments en CSS</b>	<b>28</b>
<b>15 Annexe : création de VTT</b>	<b>30</b>

# 1. Débuter

---

## Balises

Le langage HTML est un langage à balise<sup>1</sup> encadré par des chevrons ("`<`" et "`>`"). On distingue deux types de balises :

- Les balises (standards) par *paires*. Elles sont les plus répandus et marche par paire : une balise ouvrante et une balise fermante précédée par un "`/`".

```
<balise>...</balise>
```

Dans ce cours, on dira que ... est imbriquée dans la balise **balise**.

- Les balises *orphelines*. Elles sont moins communes et sont simplement notées entre chevrons.

```
<balise>
```

On dénombre environ 140 balises HTML.

## Au minimum

La structure minimale d'une page HTML est

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

La balise (orpheline) `<!DOCTYPE html>` est la balise du type du document. Elle indiquera au navigateur<sup>2</sup> qu'il va lire du HTML.

La balise `html` englobe toutes les informations du code html.

La balise `head` va permettre d'ajouter des informations à la page (encodage, titre, mots clefs, feuille de style, javascript ...). Le code contenu dans cette balise ne sera pas affichée dans la page web (dans la pratique, mais une mauvaise pratique engendre des réactions involontaires).

La balise `body` englobe tout le code html qui sera affiché sur la page.

## Debuggage

En général, les navigateurs gèrent bien les erreurs de code HTML. Par exemple, l'absence de balise fermante n'empêchera pas l'affichage de la page. Dans le pire des cas, elle s'affichera mal ; dans le meilleur, on ne remarquera même pas qu'il existe une erreur.

Néanmoins, pour obtenir un code propre, le W3C a développé un validateur de code HTML disponible à l'adresse

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

Il est possible de :

- Indiquer l'URL d'une page web.
- De charger un fichier
- D'écrire du code HTML directement dans un encadré

Quelque soit le mode, le programme de validation indiquera la présence ou non d'erreur (ou de *warning*, point du code qui mérite une attention particulière).

## Un dernier commentaire

Il peut être parfois utilise de commenter<sup>3</sup> son code pour permettre un relecture plus confortable.

En HTML les commentaires se font de la manière suivantes :

```
<!-- Voici un commentaire -->
```

---

1. Il en existe d'autre comme le (déjà mort) XHTML ou XML

2. Un navigateur est un interpréteur de HTML.

3. Un commentaire est une information essentiellement à but pédagogique, qui s'affiche dans l'éditeur de code mais qui est ignoré lors de son interprétation.

## 2. Premières balises

Commençons par quelques balise permettant de structurer le texte.

### Dans le head

Balise	Description	Exemple
<code>title</code>	Permet de modifier le titre de la page web	<code>&lt;title&gt;Ceci est le titre de la page&lt;/title&gt;</code>

### Dans le body

Balise	Description	Exemple
<code>p</code>	Définit un paragraphe	<code>&lt;p&gt;Ceci est un paragraphe&lt;/p&gt;</code>
<code>br</code>	Cette balise <u>orpheline</u> permet de sauter une ligne	Ligne 1 <code>&lt;br&gt;</code> Ligne 2
<code>hr</code>	Cette balise <u>orpheline</u> permet de sauter à la ligne et de tracer un trait horizontal entre les deux lignes	Ligne 1 <code>&lt;hr&gt;</code> Ligne 2
<code>h1</code>	Définit un titre principal	<code>&lt;h1&gt;Ceci est titre principal&lt;/h1&gt;</code>
<code>h2</code>	Définit un titre secondaire	<code>&lt;h2&gt;Ceci est titre secondaire&lt;/h2&gt;</code>
<code>h3, h4, h5, h6</code>	Définit d'autre niveau de titre	<code>&lt;h6&gt;Ceci est un tout petit titre&lt;/h6&gt;</code>
<code>em</code>	Permet d'emphaser du texte (italique)	<code>&lt;em&gt;Ceci est un texte emphasé&lt;/em&gt;</code>
<code>strong</code>	Permet d'apporter de l'importance au texte (mise en gras)	<code>&lt;strong&gt;Ceci est un texte important&lt;/strong&gt;</code>
<code>mark</code>	Met en valeur le texte (surlignage)	<code>&lt;strong&gt;Ceci est un texte mis en valeur&lt;/strong&gt;</code>
<code>abbr</code>	Définit une abréviation	<code>&lt;abbr&gt;ceci est une abréviation&lt;/abbr&gt;</code>
<code>sup</code>	Permet la mise en exposant	Ceci est en <code>&lt;sup&gt;</code> exposant <code>&lt;/sup&gt;</code>
<code>sub</code>	Permet la mise en indice	Ceci est en <code>&lt;sub&gt;</code> indice <code>&lt;/sub&gt;</code>
<code>blockquote</code>	Permet de faire une citation longue	<code>&lt;blockquote&gt;Ceci est une petite citation&lt;/blockquote&gt;</code>
<code>q</code>	Permet de faire une citation courte dans du texte	<code>&lt;q&gt;Ceci est une petite citation&lt;/q&gt;</code>
<code>del</code>	Texte supprimé (rayé)	<code>&lt;del&gt;Ce texte est supprimé&lt;/del&gt;</code>
<code>ins</code>	Texte inséré (souligné)	<code>&lt;ins&gt;Ceci est un texte inséré&lt;/ins&gt;</code>
<code>kbd</code>	Texte de saisie au clavier (police courrier)	<code>&lt;kbd&gt;Ceci est une saisie au clavier&lt;/kbd&gt;</code>
<code>progress</code>	Permet d'afficher une barre de progression	<code>&lt;progress&gt;&lt;/progress&gt;</code>

### 3. Gestion des caractères : Code HTML

Certain caractères spéciaux ont un **code HTML** pouvant être compris par le navigateur. Il existe plusieurs moyen de préciser le caractère :

**Code décimal** : se spécifie par

`&#...;`

où ... est un nombre entier. Par exemple le symbole \$ a pour code HTML décimal `&#36;`;

**Code hexadécimal** : se spécifie par

`&#x...;`

où ... est un nombre entier. Par exemple le symbole \$ a pour code HTML hexadécimal `&#x24;`;

**Code mnémonique** : se spécifie par

`&...;`

où ... est une chaîne de caractère. Par exemple le symbole \$ a pour code HTML mnémonique `&dollar;`;

ATTENTION : Certain caractère n'ont pas d'écriture mnémonique.

Il y a de nombreux code HTML permettant de gérer les caractères. A titre d'exemple, voici une brève liste. On pourra compléter cette liste ici<sup>4</sup>.

[https://fr.wikipedia.org/wiki/Aide:Liste\\_de\\_caractères\\_spéciaux](https://fr.wikipedia.org/wiki/Aide:Liste_de_caractères_spéciaux)

On peut passer du code décimal au code hexadécimal par une *simple* conversion numérique (dans l'exemple, on remarque que l'entier 36 s'écrit bien 24 en hexadécimal).

Mnémo	Hexa	Rendu
<code>&amp;plusmn;</code>	<code>&amp;#x00B1</code>	±
<code>&amp;times;</code>	<code>#xD7;</code>	×
<code>&amp;divide;</code>	<code>&amp;#xF7;</code>	÷
<code>&amp;bull;</code>	<code>&amp;#x2022;</code>	•
<code>&amp;part;</code>	<code>&amp;#x2202;</code>	∂
<code>&amp;prod;</code>	<code>&amp;#x220F;</code>	∏
<code>&amp;sum;</code>	<code>&amp;#x2211;</code>	∑
<code>&amp;minus;</code>	<code>&amp;#x2212;</code>	−
<code>&amp;lowast;</code>	<code>&amp;#x2217;</code>	*
<code>&amp;radic;</code>	<code>&amp;#x221A;</code>	√
<code>&amp;and;</code>	<code>&amp;#x2227;</code>	∧
<code>&amp;or;</code>	<code>&amp;#x2228;</code>	∨
<code>&amp;cap;</code>	<code>&amp;#x2229</code>	∩
<code>&amp;cup;</code>	<code>&amp;#x222A;</code>	∪
<code>&amp;int;</code>	<code>&amp;#x222B;</code>	∫
	<code>&amp;#x222C;</code>	∬
	<code>&amp;#x222D;</code>	∭
	<code>&amp;#x222E;</code>	∫
<code>&amp;oplus;</code>	<code>&amp;#x2295;</code>	⊕
<code>&amp;otimes;</code>	<code>&amp;#x2297;</code>	⊗
<code>&amp;sdot;</code>	<code>&amp;#x22C5;</code>	·

4. Un exemple parmi tant d'autre

## 4. Attributs

Un *attribut* est une information complémentaire permettant d'enrichir une balise ou de préciser son fonctionnement. Certaines balises ont des attributs obligatoires et certaines optionnelles.

Dans la pratique, on précise le **nom** et la **valeur** d'un attribut entre guillemets de la manière suivante

```
<balise nom="val">...</balise>
```

De la même manière pour les balises orphelines.

Il est également possible de donner plusieurs attributs à une balise en les séparant par un espace.

```
<balise nom1="val1" nom2="val2" ... >...</balise>
```

Certains attributs sont dit *universels* c'est à dire qu'ils s'appliquent à toutes les balises<sup>5</sup>. Les autres sont *spécifiques*, c'est à dire qu'ils s'appliquent uniquement à certaines balises et sont parfois même indispensables.

### Attributs universelles

Attribut	Description
<b>accesskey</b>	Cet attribut fournit une indication permettant de générer un raccourci clavier pour l'élément courant. Cet attribut se compose d'une liste de caractères séparés par des espaces. Le navigateur doit utiliser le premier caractère qui existe selon la disposition du clavier utilisée.
<b>class</b>	Cet attribut permet de définir la ou les classes auxquelles appartient un élément afin de le manipuler en script ou de le mettre en forme avec CSS.
<b>contenteditable</b>	Un attribut à valeur contrainte qui indique si l'élément peut être édité par l'utilisateur. Si c'est le cas, le navigateur modifie l'interface utilisateur afin de permettre l'édition. Les valeurs autorisées pour cet attribut sont soit <b>true</b> (ou la chaîne vide) : ces valeurs indiquent que l'élément doit pouvoir être édité ; soit <b>false</b> qui indique que l'élément ne doit pas pouvoir être édité.
<b>contextmenu</b>	La valeur de cet attribut correspond à la valeur de l'attribut <b>id</b> d'un élément <code>&lt;menu&gt;</code> qui doit être utilisé comme menu contextuel par cet élément.
<b>data-*</b>	Cette classe d'attribut, appelée <i>attributs de données personnalisés</i> , permet d'échanger des informations dans un format propriétaire entre le HTML et le DOM <sup>6</sup> afin de pouvoir les manipuler via des langages de scripts. La propriété <code>HTMLElement.dataset</code> permet d'accéder à l'ensemble des attributs définis de cette façon.
<b>dir</b>	Un attribut à valeur contrainte qui indique la directionnalité du texte de l'élément. Les valeurs autorisées pour cet attribut sont : <b>ltr</b> <sup>7</sup> <b>rtl</b> <sup>8</sup> ou <b>auto</b> (dans ce dernier cas, c'est l'agent utilisateur qui décide)
<b>draggable</b>	Un attribut à valeur contrainte qui indique qu'un élément peut être glissé/déposé grâce à l'API <sup>9</sup> <i>Drag &amp; Drop</i> . Les valeurs autorisées pour cet attribut sont <b>true</b> (l'élément peut être glissé/déposé) ou <b>false</b> .
<b>hidden</b>	Un attribut booléen dont la valeur indique que l'élément n'est pas encore, ou n'est plus pertinent. Cet attribut peut ainsi être utilisé pour masquer les éléments d'une page qui ne peuvent pas être utilisés tant que la procédure de connexion n'est pas terminée. Le navigateur n'affichera pas les éléments pour lesquels cet attribut est utilisé. Cet attribut ne doit pas être utilisé afin de masquer du contenu qui aurait pu être affiché de façon pertinente <sup>10</sup> .

5. Dans certains cas, l'attribut n'a pas de sens pour la balise et il n'y aura aucun impact sur la page.

6. Document Object Model : permet à des scripts d'interagir (consultation et modification) avec le HTML

7. Left To Right ; comme la langue française.

8. Right To Left ; comme la langue arabe.

9. Application Programming Interface : c'est une sorte de bibliothèque permettant l'ajout de fonctionnalités au programme

10. le code HTML est à tout moment accessible via *afficher le code source*

<b>id</b>	Cet attribut définit un identifiant, unique au sein de tout le document, pour un élément. Il doit permettre d'identifier un élément lorsqu'on crée un lien vers lui et/ou lorsque le manipule avec des scripts.
<b>title</b>	Cet attribut contient une représentation textuelle de l'information auquel il est lié. Une telle information est généralement, mais pas nécessairement, affichée sous la forme d'une bulle d'informations.

Il en existe certains qui sont à l'état de développement et non totalement fonctionnel avec tous les navigateurs. Pour plus de détails :

[https://developer.mozilla.org/fr/docs/Web/HTML/Attributs\\_universels](https://developer.mozilla.org/fr/docs/Web/HTML/Attributs_universels)

## Attributs spécifiques

### Balises du body

Les balises de `body` que nous avons introduits dans les précédents chapitres font parties des balises les plus communes et bénéficient donc de très peu d'attributs spécifiques<sup>11</sup>. Voici quelques cas particuliers :

Attribut	Balise(s)	Description	Exemple
<b>cite</b>	<code>&lt;blockquote&gt;</code> , <code>&lt;q&gt;</code> , <code>&lt;ins&gt;</code> , <code>&lt;del&gt;</code>	Cet attribut est une URI <sup>12</sup> qui pointe vers la source de la citation ou de la modification	<pre>&lt;ins cite="www.ataraxy.info"&gt;   NOUVEAU &lt;/ins&gt; &lt;del cite="Game of thrones"&gt;   C'est un bon livre &lt;/del&gt;</pre>
<b>max</b>	<b>progress</b>	Cet attribut indique la valeur maximale autorisée	
<b>value</b>	<b>progress</b>	Cet attribut définit la valeur par défaut qui sera affichée dans l'élément au chargement de la page	<pre>&lt;progress max="100" value="33"&gt; &lt;/progress&gt;</pre> Affichera une barre de progression à 33/100.

### Balises du head

Une balise *orpheline* `meta`, impérativement placée dans la balise `head`<sup>13</sup>, permet de définir des métadonnées<sup>14</sup>. Cette balise est uniquement définie par les attributs qui lui sont spécifiés. Détaillons quelqu'un de ses attributs.

**charset.** Cet attribut permet de spécifier l'encodage de la page. En général, on utilise le `utf-8`<sup>15</sup> qui est le plus large et le plus répandu.

```
<meta charset="utf-8">
```

**content.** Permet de spécifier le contenu d'une métadonnée. S'utilise avec l'attribut `name` et `http-equiv`.

**name.** Permet de nommer la métadonnée que l'on cherche à définir. S'utilise avec l'attribut `content`. Les valeurs possibles de l'attribut `name` sont :

11. Les attributs universelle suffisent.

12. URI : Uniform Resource Identifier. Les URL (L=Locator) et URN (N= Name) sont des cas particulier d'URI.

13. Les balises `meta` répondent à une contrainte assez particulière : être placée dans les 512 premiers octets de la page. Ce, afin que le navigateur sache rapidement (lors de la lecture du code HTML), quelles sont les métadonnées qu'il doit appliquer à la page (comme l'encodage).

14. Une métadonnée est une donnée permettant de décrire une autre donnée. Dans notre cas (WEB), les métadonnées permettent d'enrichir la page par des informations supplémentaires

15. On pourra consulter [https://www.w3schools.com/html/html\\_charset.asp](https://www.w3schools.com/html/html_charset.asp)

- **author** qui précise l'auteur de la page.

```
<meta name="author" content="David Hébert">
```

- **description** permet de spécifier un résumé de la page. C'est en général<sup>16</sup> ce texte qui apparaît dans les moteurs de recherche.

```
<meta name="description" content="Des maths, tabernacle!">
```

- **keywords** permet de préciser un ensemble de mots-clefs (séparé par des virgules). Cela permet une meilleure indexation de la page.

```
<meta name="keywords" content="Gauss, Euler, Hilbert, Noether">
```

- **robots** permet d'influencer les robots d'indexation<sup>17</sup>. En choisissant de mettre **robots** à l'attribut **name** les valeurs de **content** possible sont alors :

<b>index</b>	Permet au robot d'indexer la page
<b>noindex</b>	Interdit au robot d'indexer la page
<b>follow</b>	Permet au robot de suivre les liens contenus dans la page
<b>nofollow</b>	Interdit au robot de suivre les liens contenus dans la page
<b>noodp</b>	Empêche que la description si celle-ci est présente, soit affichée dans les résultats du moteur de recherche
<b>noarchive</b>	Empêche le moteur de recherche de mettre en cache le contenu de la page
<b>nosnippet</b>	Empêche l'affichage de la description de la page dans les résultats du moteur de recherche

A noter que l'on peut remplacer **name="robots"** par **name="googlebot"** pour n'influencer que le robot de Google, **name="slurp"** pour n'influencer que le robot de Yahoo.

**http-equiv** prend des valeurs prédéfinies permettant d'interférer (dans une faible mesure) avec le protocole HTTP. Nous n'en aurons pas besoin pour ce cours d'introduction au WEB. Notons tout de même la redirection.

- **refresh** permet de rediriger vers une nouvelle page au bout d'un certain nombre de seconde.

```
<meta http-equiv="refresh" content="10; url=www.ataraxy.info">
```

La présence de cette métadonnée dans le **head** redirigera l'utilisateur vers **ataraxy.info** au bout de 10 secondes.

Il existe beaucoup d'autres métadonnées. On trouvera une liste exhaustive ici :

<https://wiki.whatwg.org/wiki/MetaExtensions>

---

16. Par exemple pour Firefox, Opera

17. Googlebot, Slurp etc qui sont les robots d'indexation répandus respectent cette métadonnée. D'autres robots, en général à but malveillant et/ou commercial, vont l'ignorer.



## 5. Liste et tableaux

---

### Balise de liste

Pour définir des listes en HTML, on dispose de 3 balises.

`ol` pour *ordered list* permet de définir des listes ordonnées<sup>18</sup>.

`ul` pour *unordered list* permet de définir des listes non ordonnées<sup>19</sup>.

Ces deux balises définissent une liste. Pour définir un élément de la liste, on utilise la balise `li`.

```
<ol>                                <ul>
  <li>Ouvrir un éditeur de texte</li>    <li>Manettes</li>
  <li>Saisir les lignes de code HTML</li> <li>Console PS1</li>
  <li>Sauvegarder</li>                  <li>Prise péritelte</li>
  <li>Ouvrir avec un navigateur</li>     <li>Carte mémoire</li>
</ol>                                </ul>
```

### Attributs sur les balises de liste

Il n'y a pas d'attribut spécifique à la balise `ul`<sup>20</sup>.

Pour la balise `ol` :

Attribut	Description	Exemple
<code>start</code>	Permet de commencer la liste à une autre valeur que 1	<pre>&lt;ol start="7"&gt;   &lt;li&gt;Manger&lt;/li&gt;   &lt;li&gt;Dormir&lt;/li&gt; &lt;/ol&gt;</pre>
<code>reversed</code>	Numérote les puces dans l'ordre décroissant	<pre>&lt;ol reversed&gt;   &lt;li&gt;Manger&lt;/li&gt;   &lt;li&gt;Dormir&lt;/li&gt; &lt;/ol&gt;</pre>
<code>type</code>	Prend des valeurs parmi <code>a</code> , <code>A</code> , <code>i</code> et <code>I</code> pour numéroter respectivement par des lettres minuscules, majuscules, chiffres romains minuscules et chiffres romains majuscules	<pre>&lt;ol type="i"&gt;   &lt;li&gt;Manger&lt;/li&gt;   &lt;li&gt;Dormir&lt;/li&gt; &lt;/ol&gt;</pre>

Enfin, dans le cas de liste ordonnée, il est possible de spécifier la valeur d'un élément (`li`) à l'aide de l'attribut `value` qui est un entier.

---

18. C'est à dire que les éléments des listes seront par exemple numérotés

19. C'est le cas des listes à puce

20. Pour une liste non ordonnée, l'action principale consiste à changer la forme de la puce, ce qui se fait confortablement avec CSS.

```
<ol>
  <li>Jouer</li>
  <li value="1">Manger</li>
  <li>Dormir</li>
</ol>
```

Cet exemple numérotera les éléments en 1, 1, 2.

```
<ol reversed type="A">
  <li>Jouer</li>
  <li value="10">Manger</li>
  <li>Dormir</li>
</ol>
```

Cet exemple numérotera les éléments en C, J, I.

## Tableaux

On utilise la balise `table` pour définir un tableau.

La balise `tr` définit une ligne du tableau et la balise `td` définit une case dans la ligne. Il existe également une balise `th` définissant aussi une case dans une ligne ayant pour but de spécifier le titre de la colonne concerné.

Ainsi dans un tableau on aura (devra) toujours avoir une balise `td` ou `th` dans une balise `tr` elle même dans une balise `table`, comme dans l'exemple ci-contre.

```
<table>
  <tr>
    <td></td>
    <th>Colonne 1</th>
    <th>Colonne 2</th>
  </tr>
  <tr>
    <th>Ligne 1</th>
    <td>L1 C1</td>
    <td>L1 C2</td>
  </tr>
  <tr>
    <th>Ligne 2</th>
    <td>L2 C1</td>
    <td>L2 C2</td>
  </tr>
</table>
```

### Attributs des tableaux

La balise `table` n'as pas d'attributs spécifique. Il existe néanmoins une *balise* permettant de donner un titre au tableau c'est `caption`.

```
<table>
  <caption>Liste des pokémons de type foudre</caption>
  ...
</table>
```

**colspan** permet d'indiquer le nombre de colonne devant être fusionné. Il s'agit d'un attribut des balises `td` et `th`.

```
<table>
  <tr>
    <th colspan="2">DATA</th>
    <th>Le prix</th>
  </tr>
  <tr>
    <td>DATA 1</td>
    <td>DATA 2</td>
    <td>Prix</td>
  </tr>
</table>
```

**rowspan** permet d'indiquer le nombre de colonne devant être fusionné. Il s'agit d'un attribut des balises `td` et `th`.

```
<table>
  <tr>
    <th rowspan="2">Ligne 1</th>
    <th>Le prix</th>
  </tr>
  <tr>
    <td>DATA 1</td>
    <td>DATA 2</td>
    <td>Prix</td>
  </tr>
</table>
```

## 6. Liens

Le mot *WEB* signifie littéralement *toile* et fait référence au fait que les pages web sont "connectées" les unes aux autres et que l'on peut atteindre une page depuis une autre. C'est la notion de **liens**, ou plus exactement les **liens hypertexte**. Il s'agit d'un texte qui possède une information supplémentaire : l'URL d'une page web.

La balise permettant de faire un lien hypertexte est la balise `a`.

```
<a>Voici un lien</a>
```

Par construction, aucun attribut n'est obligatoire pour cette balise. Cependant, il est évident que si on ne précise pas l'URL de redirection, ce lien hypertexte est inutile.

L'attribut permettant de spécifier l'URL est `href`.

```
<a href="l'URL_de_redirection">Voici un lien</a>
```

Voyons les différents moyen de naviguer de pages en pages.

### Liens vers le WEB (intérieur)

Lorsque l'on souhaite diriger l'utilisateur vers une autre page (du même site), il suffit simplement d'indiquer le nom de la page.

Par exemple, mon site dispose de 2 pages : `Page1.html` et `Page2.html`. Alors pour diriger de la page 1 vers la page 2, j'indique dans la page 1 :

```
<a href="Page2.html">Se rendre à la page 2</a>
```

Autre exemple, si je dispose de 2 répertoires : `Rep1` et `Rep2` et que chacun de ces répertoires contient deux pages HTML : `Page1.html` et `Page2.html`. Alors :

– pour créer un lien de la page 1 du répertoire 1 vers la page 2 du répertoire 2 :

```
<a href=" ../Rep2/Page2.html">Pikachu</a>
```

– pour créer un lien de la page 2 du répertoire 2 vers la page 1 du répertoire 2 :

```
<a href="Page1.html">Raïchu</a>
```

### Ancre

Il est possible dans un lien hypertexte de préciser l'endroit de la page web sur lequel on souhaite pointer. On parle d'**ancres**. Le navigateur placera le haut de la page au niveau de l'ancre.

Cela se fait en deux temps.

1. Créer l'ancre à l'aide de l'attribut universel<sup>21</sup> `id`.
2. Diriger vers l'ancre en précédant le nom de l'`id` choisi par un `#`.

Exemple : je dispose de 2 répertoires : `Rep1` et `Rep2` et que chacun de ces répertoires contient deux pages HTML : `Page1.html` et `Page2.html`. La page 1 du répertoire 1 possède un titre `<h1 id="Foudre">Pokémon foudre</h1>` et une barre de progression `<progress id="EvolutionPikachu"></progress>` Alors :

– pour créer un lien de la page 1 du répertoire 1 vers la barre de progression de la même page :

```
<a href="#EvolutionPikachu">Voir l'évolution de Pikachu</a>
```

– pour créer un lien de la page 2 du répertoire 2 vers le titre de la page 1 du répertoire 1 :

```
<a href=" ../Rep1/Page1.html#Foudre">Raïchu</a>
```

### Le téléchargement

Il est également possible d'utiliser la balise `a` pour le téléchargement (d'à peu près n'importe quoi).

Par exemple je dispose, dans répertoire de la `page.html` d'un répertoire PDF avec d'une `image.png`. Alors, le code suivant permettra, au moment du clique, de télécharger l'image :

```
<a href="PDF/image.png">Télécharger l'image</a>
```

---

21. Cela signifie que l'on peut placer une ancre a peu près n'importe où.

## Liens vers le WEB (extérieur)

On peut spécifier l'URL d'une autre page WEB<sup>22</sup> dans le href.

Il faut cependant prendre garde à spécifier le protocole (`http` ou `https` pour consulter une page).

Ce code comporte une erreur (à moins qu'une page porte le nom `www.ataraxy.info`)

```
<a href="www.ataraxy.info">Ataraxy</a>
```

Le code exacte est

```
<a href="http://www.ataraxy.info">Ataraxy</a>
```

Il est possible d'utiliser d'autre protocole avec le href.

Syntaxe	Description
<code>href="http://"</code>	Protocole de transfert de page d'hypertexte.
<code>href="mailto:"</code>	Protocole SMTP pour l'envoi de mail  <pre>&lt;a href="mailto:hebert.iut@gmail.com"&gt; Cliquer ici pour m'envoyer un email. &lt;/a&gt;</pre> Il est également possible de rajouter de l'information au mail : <ul style="list-style-type: none"><li>– <code>subject</code> pour préremplir le champ objet du mail.</li><li>– <code>cc</code> pour ajouter des destinataires au mail.</li><li>– <code>bcc</code> pour ajouter des destinataires cachés.</li><li>– <code>body</code> pour préremplir le message.</li></ul> Il n'est pas nécessaire de les rajouter tous. Voici un exemple : <pre>&lt;a href="mailto:hebert.iut@gmail.com?       Subject=Demande informations&amp;       bcc=E.Macron@gouv.fr&amp;       body=Bonjour. Des infos svp."&gt; Cliquer ici pour une demande d'informations. &lt;/a&gt;</pre>
<code>href="ftp://"</code>	Protocole de transfert de fichier. Par exemple :  <pre>&lt;a href="ftp://127.0.0.1/Rep/fichier.txt"&gt;   Liens &lt;/a&gt;</pre> Si le serveur est verrouillé avec login (Log) et mot de passe mDp : <pre>&lt;a href="ftp://Log:mDp@127.0.0.1/Rep/fichier.txt"&gt;   Liens &lt;/a&gt;</pre> On utilise rarement le protocole FTP pour permettre le téléchargement de fichier <sup>23</sup> .
<code>href="pnm://"</code>	Protocole PNM pour le streaming
<code>href="news:"</code>	Protocole NNTP pour les forums de discussion
<code>href="telnet://"</code>	Protocole TelNet pour le contrôle d'ordinateur distant.
<code>href="wais://"</code>	Protocole Wais pour les base de données

Dans la pratique on utilise les protocoles `http`, `https` et `mailto`.

22. Mais aussi de la page web actuel, ça fonctionnera très bien. Mais dans ce cas, un simple lien intérieur est plus propre

23. On procède simplement en pointant vers le fichier comme nous l'avons vu ?

## L'attribut `target`

La balise `a` possède un attribut spécifiant la manière dont le navigateur doit gérer le lien. Il s'agit de l'attribut `target` qui prend des valeurs spécifiques :

`_self` spécifie que la page s'ouvrira dans la fenêtre courante (c'est la valeur par défaut<sup>24</sup>).

`_blank` spécifie que la page s'ouvrira dans une nouvelle fenêtre (nouvel onglet)<sup>25</sup>.

Par exemple, le code suivant ouvrira la page dans un nouvel onglet.

```
<a href="http://www.ataraxy.info" target="_blank">Aller à XY</a>
```

---

24. C'est à dire que ça équivaut à ne pas spécifier l'attribut `target`.

25. Comme un pop-up.

## 7. Images

Il existe différents formats d'image identifiés par leur extension : JPEG<sup>26</sup>, PNG, GIF, SVG, ICO...

La majorité des navigateurs supportent ces extensions d'images.

On utilise la balise orpheline `img`. Cette balise possède un attribut spécifique obligatoire : `src` indiquant la source de l'image.

```

```

Il existe également un attribut spécifique facultatif : `alt` affichant une information alternative lorsque l'image ne s'affiche pas<sup>27</sup>.

```

```

### Les favicons et la balise `link`

Un favicon<sup>28</sup> est l'icône qui apparaîtra lorsque la page sera enregistrée dans les favoris ou lorsqu'un raccourci de la page sera créé.

On précise le favicon dans le `head` par la balise orpheline `link`. Cette balise permet de définir une relation entre le document courant et une ressource externe. L'attribut `href` permet de spécifier la ressource externe et l'attribut `rel` la nature de la relation.

<code>rel="..."</code>	Description
<code>icon</code>	Indique que le lien est l'icône de la page. <pre>&lt;link rel="icon" href="favicon.ico"&gt;</pre>
<code>help</code>	Indique que le lien est une page d'aide sur la page <pre>&lt;link rel="help" href="Aide\_sur\_la\_page.html"&gt;</pre>
<code>author</code>	Indique que le lien concerne l'auteur de la page. <pre>&lt;link rel="author" href="mailto:hebert.iut@gmail.com"&gt;</pre>
<code>license</code>	Indique que le lien porte sur la licence de la page.
<code>preload</code>	Indique que le lien doit être entièrement chargé avant d'afficher la page <pre>&lt;link rel="preload" href="Cours.pdf"&gt;</pre>

Pour plus de détail on pourra consulter :

[https://developer.mozilla.org/fr/docs/Web/HTML/Types\\_de\\_lien](https://developer.mozilla.org/fr/docs/Web/HTML/Types_de_lien)

### Les favicons et l'attribut `type`

A noter que l'on peut spécifier un favicon qui n'est pas d'extension ICO mais pour cela il faut préciser le type, par le biais de l'attribut `type` de la balise `link` :

```
<link rel="icon" type="image/png" href="img.png">
```

On trouvera ici la liste des différents types qui sont reconnus :

[https://fr.wikipedia.org/wiki/Type\\_MIME](https://fr.wikipedia.org/wiki/Type_MIME)

26. Certains OS, comme Windows 98, ont eu du mal à gérer les extensions à plus de 3 caractères. Ainsi l'extension de JPEG est parfois notée JPG.

27. Il y a plusieurs raisons pour qu'une image ne s'affiche pas. Si la source n'est pas bonne, l'extension n'est pas acceptée par le navigateur, ou plus communément parce que l'utilisateur a désactivé le chargement des images (pour des raisons de débit ou de déficience visuelle).

28. Favicon est féminin. Il s'agit de la contraction d'icône favorite.

## 8. Audio et vidéo

### Les formats

Que ça soit pour l'audio ou pour les vidéos, il est important de prendre soin de s'assurer que le format du support est accepté par le navigateur.

Les formats audio classique (MP3, AAC, WAV, etc) comme les formats vidéo (AVI, MP4, MKV, etc) sont en général supportés par les navigateurs. Pour faire le point pour chaque navigateur des formats supportés on pourra par exemple consulter [www.caniuse.com](http://www.caniuse.com).

### Les balises

On utilise les balises `audio` pour ajouter un son et `video` pour une vidéo. Ces balises fonctionnent un peu comme les liens hypertextes (balise `a`). Il faudra en effet préciser la source (`src`) du média.

```
<audio src="Mucic.mp3"></audio>
<video src="Video.mp4"></video>
```

Le HTML5 va permettre de préciser plusieurs source et le navigateur *choisira* au mieux. Pour ce faire on utilise la balise orpheline `source` avec l'attribut obligatoire `src`. Cette balise doit obligatoirement être placée dans la balise `audio` ou `video`.

Si plusieurs balises `source` sont présente alors le navigateur devrait <sup>29</sup> choisir le premier qu'il supporte.

```
<video>
  <source src="Video.mp4">
  <source src="Video.avi">
  <source src="Video.mkv">
</video>
```

Il est également possible de préciser le `type` <sup>30</sup> à la balise `source`.

```
<video>
  <source src="Video.mp4" type="video/mp4">
  <source src="Video.avi" type="video/x-msvideo">
  <source src="Video.mkv" type="video/webm">
</video>
```

### Attributs

Voici un liste d'attributs valide pour les deux balises `audio` et `video`.

Attribut	Description
<code>autoplay</code>	Lance la lecture à l'ouverture de la page (et dès que le média est chargé). Cet attribut ne prend pas de valeur.
<code>controls</code>	Fait apparaître les boutons de contrôle du média. Cet attribut ne prend pas de valeur.
<code>loop</code>	La lecture du média sera en boucle. Cet attribut ne prend pas de valeur.
<code>muted</code>	Le son sera initialement coupé. Cet attribut ne prend pas de valeur.
<code>preload</code>	Spécifie le préchargement du média. Cet attribut est à valeur restreinte : <ul style="list-style-type: none"><li>– <code>none</code> : le média sera chargé au moment de la lecture.</li><li>– <code>metadata</code> : seul les métadonnées <sup>31</sup> seront chargées.</li><li>– <code>auto</code> : le média sera téléchargé au moment de l'affichage de la page (au bon vouloir du navigateur).</li></ul> Le préchargement d'un média peut aussi être forcé par la balise <code>link</code> . Théoriquement la ligne <code>&lt;video preload="auto" src="Video.mp4"&gt;</code> dans le <code>body</code> est équivalente à la ligne <code>&lt;link rel="preload" href="Video.mp4"&gt;</code> dans le <code>head</code> .

La balise `video` possède un attribut qui lui est spécifique : `poster`. Cet attribut permet de spécifier l'image qui sera affichée avant la lecture de la vidéo. La valeur de cet attribut est le lien (relatif ou absolu) de l'image.

29. Les navigateurs font un peu ce qu'ils veulent avec le contenu multimédia... Dans la pratique ils *choisissent* le meilleur.

30. Il s'agit d'un attribut facultatif à valeur restreinte. Voir le chapitre sur les images dans lequel nous avons introduit la balise `link` qui possède aussi l'attribut `type`.

31. Comme le temps du média, la taille d'une vidéo etc...

## Balise track

La balise orpheline facultative `track` obligatoirement placée dans une balise `video` (ou, dans une moindre mesure, une balise `audio`), permet de fournir une piste de texte pour le média<sup>32</sup>.

Le format des fichiers texte est VTT<sup>33</sup>. Ces VTT sont spécifiés dans la balise `track` par l'attribut obligatoire `src`.

```
<video controls autoplay muted>
  <source src="Video.mp4">
  <track src="Text.vtt">
</video>
```

On trouvera en annexe quelques éléments relatifs à la rédaction de VTT.

Comme pour les balises `source`, on peut spécifier plusieurs `track`<sup>34</sup>, le navigateur fera le choix le mieux adapté suivant le cas d'utilisation.

Voici certains attributs de la balise `track`.

Attribut	Description
<code>default</code>	Cet attribut sans valeur spécifie qu'il s'agit de la piste textuelle par défaut.
<code>kind</code>	Indique la nature de la piste textuelle. Cet attribut peut prendre les valeurs suivantes <sup>35</sup> : <ul style="list-style-type: none"><li>– <code>subtitles</code> : pour des sous-titres. C'est la valeur par défaut (c'est à dire ne pas mettre de balise <code>kind</code> équivaut à <code>kind="subtitles"</code>).</li><li>– <code>descriptions</code> : pour l'audiodescription.</li></ul>
<code>srclang</code>	Spécifie la langue du VTT. Cet attribut est obligatoire lorsque <code>kind="subtitles"</code> . Cet attribut prend une valeur du code <b>ISO 639-1</b> des langues. Il s'agit d'une chaîne de deux caractères. Par exemple <code>en</code> pour l'anglais, <code>fr</code> pour le français, <code>cr</code> pour le cri, etc <sup>36</sup>

---

32. Pour des sous-titres ou des commentaires d'une vidéo par exemple

33. VTT : Vidéo Text Track

34. Un fichier VTT de sous-titre par exemple.

35. Voir <https://developer.mozilla.org/fr/docs/Web/HTML/Element/track> pour une liste plus exhaustive

36. On trouvera par exemple la liste des codes de langue ici : [https://fr.wikipedia.org/wiki/Liste\\_des\\_codes\\_ISO\\_639-1](https://fr.wikipedia.org/wiki/Liste_des_codes_ISO_639-1)



## 9. Mise en forme avec le CSS

Nous avons vu que la balise `em` permet d'emphaser du texte. Cet emphasement passe par une mise en italique. Le CSS va permettre de modifier cette association `emphasage=italique`.

Il va être possible de styliser chacune des balises par l'intermédiaire du CSS.

### Utiliser du CSS

Il existe trois manières d'intégrer du CSS à une page. Pour nous permettre d'illustrer la stylisation, voici une première propriété CSS de coloration du texte en rouge :

```
color : red;
```

#### L'attribut `style`

L'attribut universel `style` permet de styliser la balise concernée.

```
<p style="color:red;">
  Ce texte est en rouge.
</p>
```

#### Le document CSS

Il est possible de créer une feuille de style, dont l'extension est CSS<sup>37</sup>, et contenant du code exactement comme dans la balise `style`. On indique alors la page de style à l'aide de la balise `link` en précisant que la relation est une feuille de style.

```
<link rel="stylesheet" href="style.css">
```

#### La balise `style`

La balise `style` placée dans le `head` va permettre de définir le style de la page, balise par balise.

```
<head>
  <title>Page WEB</title>
  <style>
    p{
      color : red;
    }
    h1{
      color : red;
    }
  </style>
</head>
<body>
  <h1>Ce titre est rouge</h1>
  <a>Cet hypertexte n'a pas de style personnel
    (ca sera par défaut souligné en bleu)
  </a>
  <p>Ce paragraphe est rouge</p>
</body>
```

### Sélecteurs

Dans le dernier exemple que nous avons traité, les titres `h1` et les paragraphes `p` étaient colorés en rouge. Pour arriver à cette fin nous avons dû préciser la propriété `color:red` pour `p` et `h1`. Il est possible de sélectionner les deux balises en même temps en écrivant

```
p, h1{
  color : red
}
```

Les **sélecteurs** permettent de préciser les balises devant être concerné par le style indiqué.

Selecteur	Description
*	S'applique à toutes les balises.
#nom	S'applique à toutes les balises qui ont l'attribut <code>id</code> de valeur <code>nom</code>
.nom	S'applique à toutes les balises qui ont l'attribut <code>class</code> de valeur <code>nom</code>
X	S'applique à toutes les balises X
X, Y	S'applique à tous les éléments <sup>38</sup> X et Y
X Y	S'applique aux éléments Y contenu dans une balise X.
X>Y	S'applique aux éléments Y contenu dans une balise X sans aucune autre balise intermédiaire.

37. CSS : Cascading Style Sheet

38. Un **élément** est une balise, un #nom (attribut `id`), un .nom (attribut `class`) ou l'élément universel \*

X+Y	S'applique aux balises Y directement placée après une balise X
X~Y	S'applique à toutes les balises Y placée après la première apparition de la balise X.
[att]	S'applique à tous les éléments avec l'attribut att  <pre>[href]{   color : red; }</pre> Affichera en rouge toutes les balises avec l'attribut href.
[att="val"]	S'applique à tous les attributs valant val.  <pre>[title="oUi"]{   color : red; }</pre> Affichera en rouge toutes les balises avec l'attribut title="oUi".
[att^="val"]	S'applique à tous les attributs dont la valeur commence par val.  <pre>[href^="https"]{   color : red; }</pre> Affichera en rouge toutes les balises avec href qui commencent par https.
[att\$="val"]	S'applique à tous les attributs dont la valeur fini par val.  <pre>[href\$=".pdf"]{   color : red; }</pre> Affichera en rouge toutes les balises avec href qui finissent par .pdf.
[att*="val"]	S'applique à tous les attributs dont la valeur contient val.  <pre>[title*="Ataraxy"]{   color : red; }</pre> Affichera en rouge toutes les balises avec title qui contient Ataraxy.

\* 39

Une fois la sélection effectuée, il faut préciser les propriétés et les valeurs à attribuer à ces propriétés.

```
selecteur{
  propriete1 : valeur;
  propriété2 : valeur;
  .
  .
  .
  propriétén : valeur;
}
```

Il est possible de commenter le code CSS en plaçant le commentaire entre /\*...\*/.

---

39. Les éléments [att=...] sont des éléments, c'est à dire qu'il peuvent se combiner. Par exemple le sélecteur #nom[href] s'appliquera aux balises avec l'attribut id="nom" et l'attribut href

## 10. Formatage du texte

Dans ce tableau récapitulatif, on donne les propriétés permettant le formatage du texte. On donne des éléments complémentaires (lorsque c'est nécessaire) dans les paragraphes suivants.

Propriété	Valeur(s)	Description
<code>font-size</code>	19px, 2.5em, x-small, smaller, 80%	Modifier la taille du texte. Des détails sont donnés après ce tableau.
<code>font-family</code>	Arial, 'MaPolice'	Modifie la police. Des détails sont donnés après ce tableau.
<code>font-style</code>	normal, italic, oblique	Modifie le style du texte.
<code>font-weight</code>	normal, bold, 400	Modifie le <i>poids</i> du texte. La ligne <code>font-weight:bold</code> permet de mettre en gras tandis que <code>font-weight:600</code> permet de mettre en gras et de <i>contrôler</i> le niveau de gras.
<code>font-kerning</code>	auto, normal, none	Permet de modifier le crénage <sup>40</sup> de la police (lorsque la police le permet). La valeur <code>auto</code> est la valeur par défaut : c'est le navigateur qui choisit l'application ou non d'un crénage. La valeur <code>normal</code> force l'application d'un crénage. Pour finir la valeur <code>none</code> empêche l'application d'un crénage.
<code>font-stretch</code>	ultra-condensed, extra-condensed, condensed, semi-condensed, normal, semi-expanded, expanded, extra-expanded, ultra-expanded	Permet de choisir la forme <code>normal</code> , <code>condensée</code> ou <code>étendue</code> de la police. Si la police ne le permet pas cette propriété n'aura aucun effet.
<code>color</code>	maroon, rgba(200, 199,0, 0.75), rgb(125, 77,31), #A1D359	Permet de modifier la couleur du texte. Des détails sont donnés après ce tableau.
<code>opacity</code>	0.3	Permet d'afficher avec de la transparence. La valeur est un nombre réel entre 0 et 1.
<code>text-align</code>	left, right, center, justify	Modifie l'alignement du texte.
<code>letter-spacing</code>	1.2px, 0.2em	Permet de préciser l'espace entre les lettres.
<code>word-spacing</code>	normal, 3px, 1.3em	Définit l'espace entre les mots.
<code>text-indent</code>	40px, 3mm	Définit l'indentation (espace horizontal au début de la première ligne).

40. Il s'agit du retrait de certains caractères par-dessus un autre pour compresser les mots. Par exemple, la lettre L suivie de la lettre T. Dans ce cas, avec un crénage, le T va *rentrer* un peu dans le L. Cela ajoute un confort de lecture.

<code>text-transform</code>	<code>capitalize, uppercase, lowercase, none</code>	.Permet de modifier le texte et d'utiliser des lettres capitale. La valeur par défaut est <code>none</code> , la valeur <code>capitalize</code> la première lettre de chaque mot sera en majuscule, la valeur <code>uppercase</code> met tous les caractères en majuscule tandis que <code>lowercase</code> les mets en minuscules.
<code>text-decoration-line</code>	<code>underline, line-through, overline, none</code>	Ajoute une décoration au texte. Respectivement : une ligne en dessous (souligné), une ligne dessus (barré) ou une ligne au dessus, il est possible de déclarer plusieurs position simultanément. Par exemple <code>text-decoration-line : underline, line-through</code> soulignera un texte barré.
<code>text-decoration-color</code>	<code>currentColor, red, rgb(1,2,3)</code>	Permet de définir la couleur utilisé pour les lignes
<code>text-decoration-style</code>	<code>solid, double, dotted, dashed, wavy</code>	Permet de préciser le type de la ligne. Respectivement normal, double, en pointillé, en pointé ou en vague.
<code>text-shadow</code>	<code>Apx Bpx Cpx</code> Couleur	Permet d'ajouter un effet d'ombre au texte. La valeur A est le décalage de l'ombre suivant les abscisses, le B suivant les ordonnées, le C l'angle de diffusion et la <code>Couleur</code> peut être donnée d'une des 4 manières que le CSS le permet (voir le paragraphe ci-dessous sur les couleurs).
<code>background-color</code>	<code>maroon, rgb(125, 77, 31), rgba(200, 199, 0, 0.75), #A1D359</code>	Permet de modifier la couleur de font. Des détails sont donnés après ce tableau.

## Tailles

La propriété spécifiant la taille de la police est `font-size`. Il existe deux manières de spécifier la taille de la police :

**Absolue.** On précise la taille en pixel noté `px`, comme par exemple `19px`. Le CSS possède des taille prédéfinie :

- `xx-small` : très très petit
- `x-small` : très petit
- `small` : petit
- `medium` : moyen
- `large` : grand
- `x-large` : très grand

**Relative.** On précise la taille en pourcentage de taille contextuelle. On note cette unité `em`. Ainsi `100%` correspond `1em`. On peut aussi préciser la taille relative avec les pourcentages, comme par exemple `99%`.

Le CSS propose des tailles relative prédéfinie.

- `smaller` : plus petit
- `larger` : plus grand

## Polices

La propriété `font-family` peut prendre plusieurs valeur, séparées par des virgules.

`font-family` : `police1, police2, ...`

Si la `police1` n'est pas disponible sur le navigateur de l'utilisateur, alors il tentera un affichage avec la `police2` etc...

Il existe de nombreuses polices prédéfinies sur la plupart des navigateurs, comme *Arial*, *Verdana*, *fantasy* etc... Celle par défaut est **sans-serif**. Il existe un moyen de personnaliser sa police<sup>41</sup>. Pour cela il faut télécharger la police souhaitée avec un champ de format le plus large pour couvrir tous les supports utilisateurs (téléphones, tablettes etc). Le plus répandu étant le **ttf** (et **svg**). Une fois que vous avez téléchargé votre police il faut saisir les lignes suivantes pour l'utiliser :

```
@font-face{
  font-family : 'LeNomQueJeVeux';
  src : url('LEndroitOuCestEnregistré.ttf') format('truetype');
}
```

On peut préciser plusieurs types différents (en général le **truetype** et **svg**), le navigateur *choisira* celle qu'il comprendra le mieux. Il est également possible de préciser, lors de la déclaration de la police des propriétés de fonte.

Par exemple :

```
@font-face{
  font-family : 'MaPolice';
  src : url('LEndroitOuCestEnregistré.ttf') format('truetype'),
       url('LEndroitOuCestEnregistré.svg') format('svg');
  font-weight : bold;
  font-style : oblique;
}
p{
  font-family : 'MaPolice', Arial, Courier, serif;
}
```

## Couleurs

En CSS, il est possible de préciser les couleurs de 4 manières différentes :

- De manière mnémonique à l'aide de mot clef interprété par le navigateur (**yellow**, **red**, etc).
- Par le code hexadécimale de la couleur (**#AEDC01**, ...).
- En précisant les valeurs RGB<sup>42</sup> qui sont des entiers entre 0 et  $2^8 - 1 = 255$  (**rgb(123, 213, 132)**, ...). Il est également possible de compléter cette information par le niveau de transparence qui est un nombre réel entre 0 et 1 à l'aide de la valeur **rgba** (**rgba(123, 213, 132, 0.75)**, ...)
- En précisant les valeurs HSL<sup>43</sup> qui correspondent à un entier entre 0 et 255 pour la teinte (0 c'est le rouge et 255 c'est le bleu), un pourcentage pour la saturation (0% pour du gris, 100% pour la couleur vive) et un pourcentage pour la luminosité (**hsl(40, 75%, 30%)** pour du marron foncé, ...). On peut, comme pour le **rgb** préciser la transparence (**hsla(40, 75%, 30%, 0.79)**, ...)

Tous ces modes d'utilisation de couleurs s'équivalent (sauf pour les valeurs mnémoniques - toutes les valeurs **rgb** n'ont pas leur pendant textuelle). On pourra trouver de nombreuses pages de conversion comme par exemple :

<http://htmlcolocodes.com/fr/>

---

41. Les deux inconvénients sont d'un point de vue fonctionnel (sans s'en rendre compte, l'utilisateur devra télécharger la police, ce qui peut prendre un peu de temps et ralentir l'expérience utilisateur) et d'un point de vue législatif (les polices sont soumises au droit d'auteur)

42. Red, Green, Blue pour rouge, vert et bleu.

43. Hue Saturation Lightness pour teinte, saturation, luminosité.

## 11. Boîtes

Dans ce tableau récapitulatif, on donne les propriétés permettant le formatage des boîtes. On donne des éléments complémentaires (lorsque c'est nécessaire) dans les paragraphes suivants.

Propriété	Valeur(s)	Description
<code>width</code>	<code>19px</code> , <code>2.5em</code> , <code>80%</code>	Modifier la largeur de la boîte.
<code>max-width</code>	<code>19px</code> , <code>2.5em</code> , <code>80%</code>	Spécifie la largeur maximale de la boîte.
<code>min-width</code>	<code>19px</code> , <code>2.5em</code> , <code>80%</code>	Spécifie la largeur minimale de la boîte.
<code>height</code>	<code>19px</code> , <code>2.5em</code> , <code>80%</code>	Modifier la hauteur de la boîte.
<code>max-height</code>	<code>19px</code> , <code>2.5em</code> , <code>80%</code>	Spécifie la hauteur maximale de la boîte.
<code>min-height</code>	<code>19px</code> , <code>2.5em</code> , <code>80%</code>	Spécifie la hauteur minimale de la boîte.
<code>margin</code>	<code>19px</code> , <code>2.5em</code> , <code>80%</code> , <code>auto</code>	Indique la marge de la boîte. Le mot clef <code>auto</code> permet un alignement vertical.
<code>padding</code>	<code>19px</code> , <code>2.5em</code> , <code>80%</code> , <code>auto</code>	Indique la marge à l'intérieur de la boîte. On peut faire les même précision qu'avec la propriété <code>margin</code> .
<code>visibility</code>	<code>visible</code> , <code>hidden</code>	Indique si la boîte est invisible ou cachée.
<code>overflow</code>	<code>visible</code> , <code>hidden</code> , <code>scroll</code> , <code>auto</code>	Permet de gérer les barres de défilement de la boîte. <ul style="list-style-type: none"><li>– <code>visible</code> est la valeur par défaut.</li><li>– <code>hidden</code> le contenu est rogné.</li><li>– <code>scroll</code> fait apparaitre la barre de défilement dans la boîte</li><li>– <code>auto</code> laisse le navigateur "choisir".</li></ul>
<code>background-image</code>	<code>url('chemin')</code>	Affiche une image de fond. On peut préciser plusieurs <code>url</code> séparé par des virgules.
<code>background-position</code>	<code>top</code> , <code>bottom</code> , <code>left</code> , <code>right</code> , <code>center</code>	Indique la position de l'image de fond dans la boîte.
<code>background-repeat</code>	<code>repeat-x</code> , <code>repeat-y</code> , <code>repeat</code> , <code>space</code> , <code>round</code> , <code>no-repeat</code>	Indique la manière dont l'image d'arrière plan se répète dans la boîte. <ul style="list-style-type: none"><li>– <code>repeat</code> est la valeur par défaut - l'image de fond se répète</li><li>– <code>no-repeat</code> l'image ne se répète pas.</li><li>– <code>repeat-x</code> l'image se répète suivant l'axe des abscisses.</li><li>– <code>repeat-y</code> l'image se répète suivant l'axe des ordonnées.</li><li>– <code>round</code> l'image se répète avec étirement pour remplir la boîte.</li><li>– <code>space</code> l'image se répète sans rognage.</li></ul>
<code>background-size</code>	<code>auto</code> , <code>300px</code> , <code>19%</code>	Précise la taille de l'image.
<code>border</code>	<code>...</code>	Précise la bordure de la boîte. Voir le paragraphe dédié.

## Margin & Padding

On présente les spécificité de la propriété `margin`, c'est la même chose avec `padding`.

Il est possible de préciser les bords concernés par la marge. Pour ce faire,

- `margin-left` pour la marge à gauche,
- `margin-right` pour la marge à droite,
- `margin-top` pour la marge en haut,
- `margin-bottom` pour la marge en bas.

- Si on utilise `margin : 3px`; alors la valeur s'applique aux quatre cotés.
- Si on utilise `margin : 3px 5em`; alors la valeur `3px` d'applique en haut et en bas et `5em` s'applique à gauche et à droite.
- Si on utilise `margin : 3px 5em 10% 0px`; alors la valeur `3px` d'applique en haut, `5em` s'applique à droite, `10%` s'applique en bas et le dernier à gauche.

## Border

Pour chaque bordure, `top`, `bottom`, `left`, `right` on peut préciser

**la taille** en `px` ou `em` ou alors avec les mots clefs `thin`, `medium` ou `thick` à l'aide de la propriété `width`,

**la couleur** avec les même spécificité que nous avons introduit lors de l'introduction du texte, à l'aide de la propriété `color`

**le style** avec la propriété `style`. Cette propriété peu prendre les valeurs :

- `none` sans bordure
- `hidden` sans bordure
- `dotted` en pointé
- `dashed` en pointillé
- `solid` ligne droite
- `double` deux lignes
- `groove` ligne incrusté
- `ridge` ligne qui ressort
- `inset` effet boîte enfoncé
- `outset` effet boîte surélevé.

**les angles de courbure sur les bord** que l'on peut indiquer en `px`, `em` ou pourcentage. On peut préciser l'angle gauche avec la propriété `left-radius` et l'angle droit avec la propriété `right-radius`.

Par exemple

```
p{
border-left-color : rgba(42, 42, 42, 0.5);
border-top-style : inset;
border-bottom-left-radius : 3px;
}
```

Il est possible de préciser plusieurs propriété sur une bordure comme dans l'exemple :

```
p{
border-left : thick dotted red;
}
```

Il est aussi possible de préciser pour toutes les bordures :

```
p{
border : 3px solid black; /*Toutes les bordures*/
border-style : double; /*Style des bordures*/
border-radius : 5px /*angle des 4 coins*/
}
```

## Position des boîtes

On peut modifier la position d'une boîte à l'aide des propriétés **top**, **bottom**, **left** et **right** dont on donne la valeur de manière relative ou absolue. On peut aussi préciser le comportement de la boîte au moment du scroll à l'aide de la propriété **position** qui prend les valeurs :

- **static** c'est la valeur par défaut,
- **relative** - en précisant cette valeur les propriétés **top** etc... sont pris en compte,
- **fixed** pour que la boîte ne bouge pas lors d'un scroll.

## Nature d'une boîte

La propriété **display** permet de modifier la nature d'une boîte. Principalement les boîtes bloc ou les boîtes de lignes. Les valeurs restreintes sont nombreuses mais on retiendra :

- **bloc** : pour faire de la sélection une boîte de type bloc,
- **inline** : pour faire de la sélection une boîte de ligne,
- **inline-bloc** : pour faire de la sélection une boîte de ligne avec les propriétés des boîtes de bloc (ou inversement),
- **none** : pour ne pas afficher la boîte,

Ainsi `strong{display:bloc;}` modifie la balise **strong** (qui est par défaut une balise de ligne) en une balise bloc.

## Propriété CSS des tableaux

Il existe une poignée de propriétés CSS propre aux tableaux. Citons-en deux (et demi) :

**border-collapse** qui peut prendre deux valeurs : **collapse**, **separate**. La valeur **collapse** permet de fusionner les cases d'un tableau sur leur bordure. La valeur **separate** qui est la valeur par défaut peut être augmentée de la propriété **border-spacing** dont la valeur est une distance indiquant la distance séparant les cases.

**vertical-align** permet de préciser la position du contenu dans la case. Cette propriété peut prendre les valeurs restreintes suivantes :

- **baseline** pour un alignement sur la ligne de base,
- **sub** pour un alignement sur la ligne de base inférieure (comme les indices résultant de l'utilisation de la balise homonyme),
- **super** pour un alignement sur la ligne de base inférieure (comme les indices résultant de l'utilisation de la balise **sup**),
- **middle** pour un alignement au milieu,
- **top** pour un alignement en haut,
- **bottom** pour un alignement en bas.

il est également possible d'indiquer une valeur de longueur (absolue ou relative) pour déterminer la position par rapport à la ligne de base.

Il est possible d'utiliser ces propriétés pour gérer aussi les boîtes, qui peuvent être vues comme des tableaux avec une seule et unique case.



## 12. Formulaires

Pour envoyer des données ou faire des sélections, on utilise le PHP <sup>44</sup> qui permet de manipuler et traiter les informations transmises.

Ici on ne traite que de la partie de l'affichage des données, vous apprendrez à traiter ces données lors du cours de PHP.

### Balise form

La balise `form` permet de créer un formulaire. Cette balise possède deux attributs obligatoires :

- **action** qui précise **quoi faire** lors de l'envoi du formulaire,
- **method** qui précise **comment** envoyer les données du formulaire.

**get**. Cette méthode laisse apparaître les données dans la barre d'URL.

**post**. Cette méthode ne laisse pas apparaître les informations qui sont en quelque sorte *invisible* <sup>45</sup>.

Par exemple :

```
<form action="Traitement.php" method="post">
```

...

```
</form>
```

Cette balise possède aussi des attributs spécifiques optionnels :

Attribut	Description
<b>autocomplete</b>	Cet attribut prend deux valeurs : <ul style="list-style-type: none"><li>– <b>on</b> qui est la valeur par défaut. Le navigateur va, lorsque c'est possible (par de précédente saisie de l'utilisateur) compléter les champs.</li><li>– <b>off</b> empêche l'autocomplétion. Il faut aussi rajouter cet attribut aux balises <code>input</code> dans la balise <code>form</code>.</li></ul>
<b>enctype</b>	Spécifie l'encodage des données transmises. Cet attribut n'a d'effet que lorsque la <b>method</b> est <b>post</b> . Cet attribut est utilisé lorsque l'on souhaite transmettre des fichiers. Dans ce cas l'attribut prend la valeur <code>multipart/form-data</code> .
<b>target</b>	C'est le même attribut que pour la balise <code>a</code> qui prend la valeur <code>_blank</code> lorsque l'on souhaite que la nouvelle page s'ouvre dans un nouvel onglet.

### Balise input

La balise orpheline `input` permet de créer un contrôle interactif de formulaire. Cette balise a en générale deux attributs obligatoires :

- Le **type** de donnée. Spécifier le type permet d'afficher des éléments de contrôle adaptés.
- Le nom (**name**) pour que le serveur (en PHP) puisse identifier la donnée et la traiter en conséquence.

### Différents type

Type	Description
<b>button</b>	Crée un bouton sans comportement particulier.
<b>checkbox</b>	Case à cocher.

44. PHP : PHP Hypertext Preprocessor

45. En réalité, ces informations ne sont pas du tout invisibles ! Il ne faut pas préférer cette méthode pour des raisons de sécurité, c'est une erreur.

<code>color</code>	Préciser une couleur (code hexadécimal)
<code>date</code>	Envoyer une date
<code>datetime-local</code>	Envoyer date et heure (heure et minutes)
<code>email</code>	Envoyer un courriel
<code>file</code>	<p>Pour charger un fichier à envoyer. On peut préciser le type de fichier devant être accepté à l'aide de l'attribut optionnel <code>accept</code> comme dans l'exemple :</p> <pre>&lt;input type="file" accept=".xls, .xlsx, .mp4, .jpg"&gt;</pre> <p>On peut aussi préciser les types MIME.</p>
<code>hidden</code>	Permet d'envoyer des données sans les afficher.
<code>number</code>	Envoie d'un nombre
<code>password</code>	Envoie d'un mot de passe. On peut enrichir, dans ce cas, de l'attribut <code>maxlength</code> pour empêcher l'utilisateur de saisir un mot de plus de <code>x</code> caractère.
<code>radio</code>	Pour sélectionner une donnée parmi un ensemble. Il est important d'ajouter un attribut <code>id</code> permettant d'identifier le nom des radios.
<code>range</code>	Permet d'envoyer un ordre de grandeur (nombre entier). On peut enrichir cette balise avec les attributs <code>max</code> et <code>min</code> spécifiant les valeurs maximale et minimale.
<code>reset</code>	Réinitialise toutes les valeurs d'une formulaire.
<code>search</code>	Barre de recherche
<code>submit</code>	Bouton d'envoi du formulaire
<code>text</code>	Une ligne de texte.
<code>time</code>	Permet d'indiquer une heure (heure et minute)
<code>url</code>	Permet de saisir un URL
<code>week</code>	Permet d'envoyer un numéro de semaine.

### Attributs spécifiques

Attribut	Description
<code>autocomplete</code>	Même attribut que pour la balise <code>form</code> .
<code>autofocus</code>	Ne peut être présent qu'une fois sur une page. Placera le curseur sur l' <code>input</code> concerné.
<code>checked</code>	Coche une case (pour les type <code>checkbox</code> et <code>radio</code> )
<code>disabled</code>	Pour empêcher la saisie.
<code>form</code>	Prend la valeur de l' <code>id</code> attribué à la balise <code>form</code> . Cela permet de mettre des éléments <code>input</code> autrement que fils d'une balise <code>form</code> .
<code>multiple</code>	Cet attribut indique que l'utilisateur peut saisir plusieurs valeur. Ne fonctionne que pour les <code>email</code> et <code>file</code> .
<code>name</code>	Indique le nom de la variable créée (pour le PHP).
<code>placeholder</code>	Indique un texte à titre indicatif dans le champs de saisie (lorsqu'il y en a un).

<b>required</b>	Indique que la champs est obligatoire empechant l'envoi tant qu'il n'est pas renseigné.
<b>size</b>	Indique la taille, en pixel, de l'élément. La valeur par défaut est 20.
<b>value</b>	Préremplie le champ avec une valeur.

## Autres balises

### textarea

Pour envoyer un texte plus grand (avec notamment des saut de lignes) on utilisera la balise `textarea`. On peut préciser la taille du champs à l'aides des attributs `cols` et `rows` permettant d'indiquer respectivement le nombre de colonne et le nombre de ligne.

```
<textarea name="TXT" cols="100" rows="10">
  Saisir le texte ici !
</textarea>
```

### select

Pour afficher un menu déroulant on utilise la balise `select` accompagnée de balise `option` indiquant les différentes valeurs du menu.

```
<select name="menu">
  <option value="F"> Frites </option>
  <option value="R"> Riz </option>
  <option value="P"> Pâtes </option>
</select>
```

On peut préciser qu'il est possible de sélectionner plusieurs éléments en précisant l'attribut `multiple` dans la balise `select`.

Enfin il est possible de présélectionner un élément dans un menu déroulant à l'aide de l'attribut `selected`.

```
<select name="menu" multiple>
  <option value="F"> Frites </option>
  <option value="R" selected> Riz </option>
  <option value="P"> Pâtes </option>
</select>
```

## 13. Hacking

---

Oui on va en parler, on fera même des TP, mais je crois que ça pourrait m'attirer des problèmes si je rédigeais un cours dessus...

## 14. Pseudo-classe et pseudo-éléments en CSS

On peut rajouter à des sélecteurs des *pseudo-classe* (en les précédant du caractère `:`) ou des *pseudo-éléments* (en les précédant des caractères `::`) précisant dans quel état appliquer des conditions CSS.

### Pseudo-classes

Selecteur	Description
<code>:active</code>	S'applique au moment du clique. <pre>p:active{   background-color: red; }</pre> Ce sélecteur va surligner les paragraphes en rouge au moment du clique.
<code>:hover</code>	S'applique au passage de la souris. <code>p:hover background-color : yellow;</code> Surlignera en jaune lorsque la souris passera sur un paragraphe.
<code>:empty</code>	Permet de cibler des balises qui n'ont pas d'enfant (pas de sous-balise).
<code>:first-child</code>	Permet de cibler le premier élément dans une balise. Par exemple <code>div:first-child</code> permet de cibler la première balise dans les balises <code>div</code>
<code>:last-child</code>	Permet de cibler le dernier élément dans une balise.
<code>:focus</code>	Permet de cibler les éléments aux focus (souris ou clavier).
<code>:link</code>	Permet de cibler les liens vers des pages non visitées.
<code>:not</code>	Permet de sélectionner les éléments qui ne sont pas concerné par le sélecteur. ATTENTION! S'utilise comme une fonction. Pour sélectionner tout sauf les titre <code>h1</code> , on écrira <code>:not(h1){...}</code>
<code>:visited</code>	Permet de cibler les liens visités.

### Pseudo-classe des formulaires

Selecteur	Description
<code>:checked</code>	Accompagne la balise <code>input</code> dont l'attribut <code>type</code> vaut <code>radio</code> ou <code>checkbox</code> et mettra en style les boutons sélectionnés. <pre>input:checked{   height: 50px;   width: 50px; }</pre>
<code>:disabled</code>	Permet de cibler les éléments désactivés
<code>:enabled</code>	Permet de cibler les éléments qui ne sont pas désactivés
<code>:invalid</code>	Permet de cibler les éléments qui ont empêcher l'envoi d'un formulaire (email etc)
<code>:required</code>	Permet de cibler les éléments avec l'attribut <code>required</code>

## Pseudo-éléments

Sélecteur	Description
<code>::after</code>	<p>S'appliquera à la fin de la balise. On l'utilise avec la propriété <code>content</code> qui permet de rajouter du contenu. Par exemple</p> <pre>h1::after{   content : "Et voila le titre";   color : red; }</pre> <p>ajoutera la phrase de <code>content</code> à la fin de la balise <code>h1</code>.</p>
<code>::before</code>	S'appliquera au début de la balise.
<code>::first-letter</code>	S'appliquera à la première lettre.
<code>::first-line</code>	S'appliquera à la première ligne.

## 15. Annexe : création de VTT

---

Dans ce chapitre, on survole la création de fichier d'extension **VTT** pour *Video Text Track*, permettant de faire apparaître du texte lors d'une vidéo. La grande force de cette pratique c'est qu'elle en nécessite aucun logiciel (autre que ceux du web : éditeur de texte et navigateur). On renvoie au chapitre sur l'audio et la vidéo en HTML pour comprendre comment utiliser un VTT.

Il est possible de vérifier votre code VTT, par exemple, ici :

<https://quuz.org/webvtt/>

### Avant-propos

**Un saut de ligne** correspond à un saut de ligne<sup>46</sup> ; c'est à dire à la saisie de la touche entrée.

**Une ligne blanche** correspond à deux sauts de ligne.

**L'encodage** du fichier créé doit être de l'**utf-8**.

Le fichier VTT est composée de 3 entités :

**L'entête** qui correspond au code placé au début du document.

**Des commentaires** placées entre des lignes blanches et débutant par le mot clef **NOTE**.

**Des répliques** qui correspondent aux éléments devant apparaître sur la vidéo.

### L'entête

L'entête se place au début du VTT<sup>47</sup> et commence par le mot clef **WEBVTT**. Ce mot clef est obligatoire.

**WEBVTT**

On peut rajouter un descriptif optionnel sur la même ligne séparé par au moins un espace.

**WEBVTT - Audio description de la vidéo.**

On peut placer également dans l'entête le code CSS du style du VTT à l'aide du mot clef **STYLE**. Voir le paragraphe *Style* de ce chapitre.

### Les commentaires

- Un commentaire permet d'ajouter une information au code VTT sans que celui-ci ne soit interprété par le navigateur.
- Les commentaires peuvent être placées dans l'entête ou entre les répliques.
- On ne place pas de commentaire dans une réplique.
- Un commentaire commence obligatoirement par le mot clef **NOTE**.
- Un commentaire prend fin à la première ligne blanche.
- Un commentaire est précédé (et succédé) par une ligne blanche
- Un commentaire peut contenir des sauts de ligne mais pas de ligne blanche.
- La chaîne de caractère `->`, le caractère `&` et le caractère `<` sont interdits dans un commentaire ; on peut contourner ce problème en remplaçant ces caractères par le code HTML mnémorique correspondant, ce qui donne respectivement `-&gt;`, `&amp;` et `&lt;`;

NOTE Cet exemple contient  
une erreur

NOTE Cet exemple est  
sans erreur

NOTE Cet exemple est  
NOTE sans erreur

---

46. #Evidence!

47. #SmartGuy!

## Les répliques

Une réplique est constituée de :

- Un identifiant (facultatif)
- L'intervalle de temps (obligatoire)
- Des options d'affichage (facultatif)
- Du texte (obligatoire)

### L'identifiant d'une réplique

Il n'est pas obligatoire d'identifier la réplique. Il peut s'agir d'un nombre correspondant au numéro de la réplique, d'un nom correspondant au nom du personnage prenant la parole etc...

La chaîne de caractère -> est interdite dans l'identifiant.

L'identifiant d'une réplique se termine par un saut de ligne.

### L'intervalle de temps

Le temps en VTT est représenté par un **timestamp** qui prend la forme **hh:mm:ss.ttt** où

**hh** correspond aux heures. C'est un nombre entier d'au moins 2 chiffres<sup>48</sup> entre 00 et 9999.

**mm** correspond aux minutes. C'est un nombre entier à deux chiffres entre 00 et 59.

**ss** correspond aux secondes. C'est un nombre entier à deux chiffres entre 00 et 59.

**ttt** correspond aux millisecondes. C'est un nombre entier à trois chiffres entre 000 et 999.

Les heures sont facultatives.

Les timestamp suivants sont faux.	Les timestamp suivants sont justes.
00:01:01.01	00:01:01.001
45:67.890	45:57.890
98:76:54.321	98:56:54.321

L'intervalle de temps en VTT est composé de trois morceaux :

- Le timestamp du début suit d'au moins un espace
- La chaîne de caractère -> suit d'au moins un espace
- Le timestamp de fin nécessairement plus grand que le timestamp de début.

Ces intervalles de temps sont faux.	Ces intervalles de temps sont justes.
01:01:01.001-->01:01:02.001	01:01:01.001 --> 01:01:02.001
45:57.890 -->01:01:01.001	45:57.890 --> 01:01:01.001
98:56:54.321 --> 98:56:54.123	98:56:54.321 --> 98:56:54.322

S'il n'y a pas d'option d'affichage l'intervalle de temps doit finir par un saut de ligne.

### Les options d'affichage

Les options facultatives d'affichage de la réplique se place entre l'intervalle de temps et le saut de ligne qui marque cette intervalle.

On peut en mettre plusieurs séparés par des espaces. La valeur d'une option est précisée en succédant le nom de l'option du caractère : et la valeur (sans espace).

98:56:54.321 -> 98:56:54.322 opt1:val1 opt2:val2 opt3:val3

---

48. Ainsi pour 1 heure on écrira 01



Voici la liste des options<sup>49</sup>.

Option	Description
<b>vertical</b>	Précise le sens de lecture. Cette option prend donc deux valeurs : <ul style="list-style-type: none"><li>- <b>rl</b> : pour right to left = de gauche à droite (comme dans les pays francophones).</li><li>- <b>lr</b> : pour left to right = de droite à gauche (comme dans les pays arabophones).</li></ul>

EN CONSTRUCTION

**Le texte**

**Style**

---

49. On pourra compléter ici : [https://developer.mozilla.org/en-US/docs/Web/API/WebVTT\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebVTT_API)