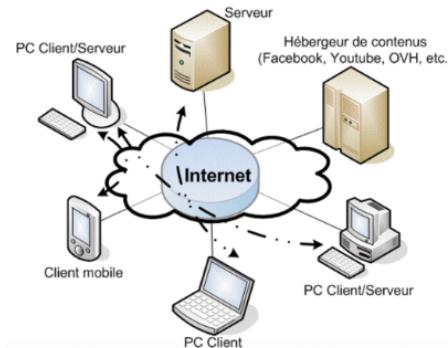


## Chapitre 2 : Initiation au Web (html et css)

### I. Web et internet

Un **réseau informatique** est un ensemble d'équipements reliés entre eux pour échanger des informations. Ces échanges se font par le biais d'un **protocole**, programme qui permet de transmettre les informations d'un élément A vers un élément B du réseau.

Le plus grand réseau informatique mondial est Internet. Il rend accessible à ses utilisateurs un certain nombre de services comme la messagerie, la publication (le Web), les réseaux sociaux ou les transferts de fichiers. Il repose sur une grande variété d'infrastructures physiques (câbles, antennes et relais, satellites, fibres).



Le **World Wide Web**, ou "la toile d'araignée mondiale", aussi appelé le **Web**, est un système fonctionnant sur internet par des **liens hypertextes**. Le Web a été mis au point dans les années **1990**, notamment par **Tim Berners-Lee** (1955, Londres). Mais son utilisation par le grand public ne commence réellement qu'en 1994, lors de la fondation du W3C (WWW Consortium) par le CERN et le MIT qui s'occupe de la normalisation et des développements du web.

Un lien hypertexte est un texte dans une page source qui permet d'atteindre une page cible lorsque l'on clique dessus. Ces liens peuvent être :

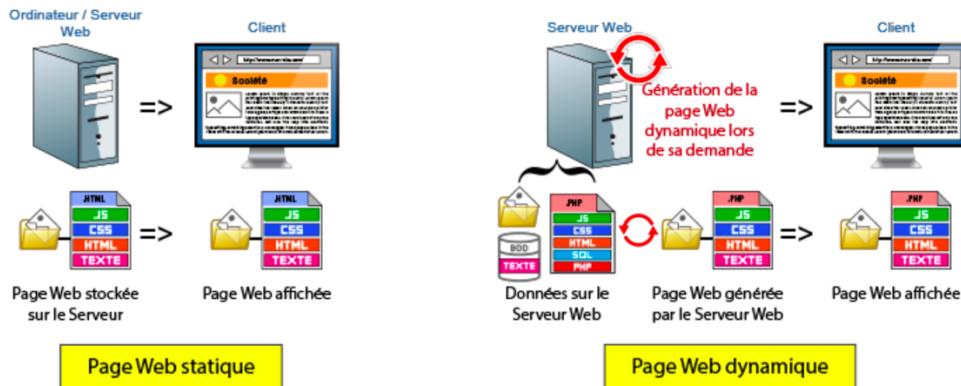
- **externes** lorsqu'ils nous dirigent vers un autre serveur
- **internes** lorsqu'ils nous dirigent vers une autre page du même site

Pour gérer et organiser le contenu des pages sur le Web, on utilise le **langage HTML**, *HyperText Markup Language*, (aujourd'hui HTML5) est un **langage de description**, il ne décrit pas le contenu sémantique du document mais il décrit sa **mise en forme**. Son code est public. Un document HTML a une extension **.html**.

**Il y a deux types de pages Web :**

- **Pages statiques** : si on visualise la page *n* fois sur le même navigateur, on obtient le même affichage
- **Pages dynamiques** : la page visualisée est générée dynamiquement selon les actions de l'utilisateur dans la page et dépend également de nombreux éléments de contexte (date, lieu, identité du visiteur, cookies, ...).

C'est pourquoi de nos jours, on utilise le langage HTML5 couplé à du CSS et JavaScript et du côté serveur du PHP et Python.



Le contenu d'une page HTML est interprété par le **navigateur (Browser)** qui se charge de son affichage. Le résultat obtenu dépend donc du navigateur !

Le principal problème, c'est que les navigateurs n'affichent pas tous les sites exactement de la même façon ! Cela est dû au fait que les navigateurs ne connaissent pas toujours les dernières fonctionnalités du HTML5.

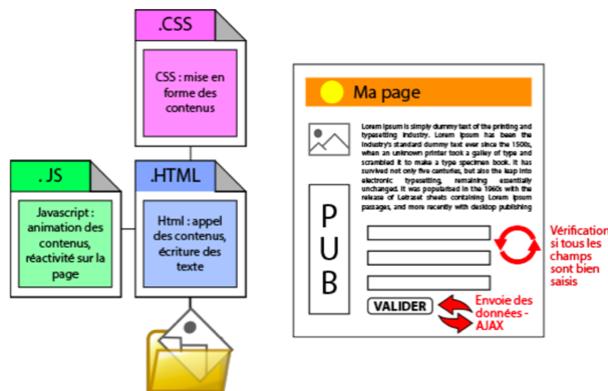


Pendant longtemps, IE fut le plus utilisé, mais aujourd'hui c'est Chrome, suivi de Firefox qui sont en tête des statistiques d'utilisation, en particulier avec les téléphones mobiles.

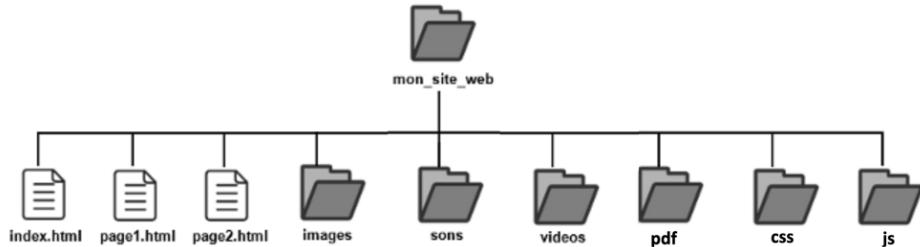
Ouvrir une page Web et faire un clic droit sur la page, et sélectionner "Afficher le code source". Le code qui apparaît est du HTML ! On peut aussi obtenir le code source en faisant Ctrl+U (ouverture d'une page avec le code) ou en faisant F12 (ouverture de la page « outils de développements »).

## II. Langage HTML

- On utilisera un éditeur de texte et les fichiers seront enregistrés avec l'extension *.html*.
- Ensuite on ouvre ce fichier *.html* avec un navigateur à côté et on commence le cycle édition/correction :
  - o Éditer la page
  - o Afficher la page
  - o Modifier si nécessaire et recommencer jusqu'à satisfaction.
- Attention il nous faut un minimum d'organisation.
- Pour créer une page dynamique, on a besoin de plusieurs langages : HTML CSS et JS (et parfois PHP). Chaque langage a son rôle :



- Lors de la création d'un site web contenant plusieurs pages web (dont la page d'accueil est index.html) et regroupant plusieurs langages HTML, CSS et JS, il est conseillé de structurer les fichiers et les dossiers selon l'exemple suivant.



- ATTENTION : il faut impérativement écrire les noms de tous les fichiers et de tous les dossiers en minuscules, sans espace et sans accentuation.

### III. Syntaxe du langage HTML

Le langage HTML est un langage qui utilise des **balises**.

Il existe deux types de balises :

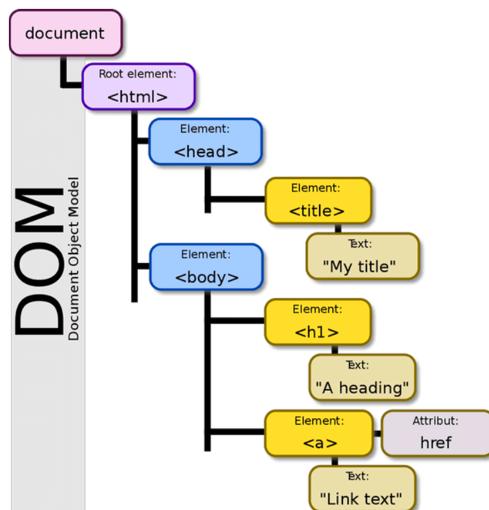
- **les balises d'encadrement :**  
<balise attribut 1="valeur1" attribut2="valeur2" ...> ... </balise>
- **les balises orphelines appelées aussi marqueurs**  
<balise attribut 1="valeur1" attribut2="valeur2" ... />

Les **attributs** servent à modifier les propriétés des balises. Ils sont optionnels.

On peut aussi mettre des commentaires dans le code dont voici la syntaxe :

```
<!-- Voici mes commentaires -->
```

#### 1. Structure d'une page web en HTML



**Exemple:**

```

1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta http-equiv="Content-type" content="text/html"; charset="utf-8">
5          <title>Ma première page web</title>
6      </head>
7      <body>
8          <!--Ici le contenu de ma page-->
9          <h1>Une page très simple</h1>
10         <p>Bonjour!</p>
11     </body>
12 </html>
```

**Ligne 1, 2 et 12 :** indique qu'on utilise le langage HTML.

```
<!DOCTYPE html>
<html>
</html>
```

**Ligne 3, 5 et 6 :** L'en-tête correspond au message apparaissant sur l'onglet de la fenêtre du navigateur de la page.

```
<head>
    <meta charset="utf-8">
    <title>Titre de l'onglet</title>
</head>
```

**Ligne 4 :** `<meta charset="utf-8">` indique la façon dont le fichier est enregistré ( en déterminant comment les caractères spéciaux vont s'afficher).

**Ligne 7 et 8 :** Le corps du document correspond au contenu de la page web elle-même. Il est contenu entre les balises `<body>` et `</body>`.

**Les balises du corps de la page web**

Balise	Rôle
<code>&lt;h1&gt; ... &lt;/h1&gt;</code>	Titre 1
<code>&lt;h2&gt; ... &lt;/h2&gt;</code>	Titre 2
<code>&lt;p&gt; ... &lt;/p&gt;</code>	Paragraphe
<code>&lt;strong&gt; ... &lt;/strong&gt;</code>	Mon texte en gras
<code>&lt;b&gt; ... &lt;/b&gt;</code>	Mon texte en gras
<code>&lt;em&gt; ... &lt;/em&gt;</code>	Mon texte en italique
<code>&lt;i&gt; ... &lt;/i&gt;</code>	Mon texte en italique
<code>&lt;sub&gt;...&lt;/sub&gt;</code>	Mon texte en indice
<code>&lt;sup&gt;...&lt;/sup&gt;</code>	Mon texte en exposant
<code>&lt;br/&gt;</code>	Saut à la ligne

On peut modifier la taille ou la police d'un texte ou d'un titre en utilisant les balises `<font>` :

`<font size=3 face="arial">texte de taille 3 et écrit en arial</font>`

On peut également modifier la couleur du texte ou son alignement :

```
<p style="color:blue">paragraphe en bleu</p>
<p style="text-align:center">paragraphe centré</p>
```

## 2. Quelle est la balise permettant de créer un lien hypertexte ? Quelle est sa structure ?

Pour faire un lien vers une autre page internet, on utilise la syntaxe suivante (avec l'adresse internet du lien entre guillemets) :

```
<a href="adresse_internet.html"> Nom qui apparaitra sur votre page </a>
```

## 3. Quelle est la balise permettant d'insérer une image ?

Pour mettre une image, on utilise la syntaxe suivante :

```

```

Cette balise est particulière car il n'a pas de fin (on ne met pas </img>)

On peut préciser la largeur (« width ») et la hauteur (« height ») de l'image en pixel :

```

```

ou la redimensionner à l'aide des pourcentages :

```

```

Pour d'autres balises, se référer à : <http://www.w3schools.com/html/default.asp>.

## IV. Structurer sa page web

Une première méthode pour structurer une page web consiste à utiliser des **tableaux**, dont les bords sont apparents ou non.

Pour cela, on utilise les balises <table> ... </table>, <tr>...</tr> et <td>...</td>.

Les balises <tr>...</tr> correspondent aux lignes du tableau et les balises <td>...</td> aux colonnes.

Les tableaux se définissent lignes par ligne.

```
<table>
  <tr>
    <td>Case 1</td>
    <td>Case 2</td>
    <td>Case 3</td>
  </tr>
  <tr>
    <td>Case 4</td>
    <td>Case 5</td>
    <td>Case 6</td>
  </tr>
</table>
```

Le code donne l'affichage suivant : Case 1 Case 2 Case 3  
Case 4 Case 5 Case 6

Les tableaux peuvent être utilisés tel quel, ou bien pour faciliter le formatage de la page internet (pour placer des images et du texte côte-à-côte par exemple).

On peut changer les attributs de style du tableau, des lignes et des colonnes en utilisant la syntaxe : <table style = "type: format" >

- Pour des bordures : `<table style="border:3px solid black">`
- Pour changer la largeur des colonnes : `<table style="width:50%">`

Il est aussi possible de fusionner des cellules ensemble pour qu'elles couvrent plusieurs colonnes

`<td colspan="2">` ou plusieurs lignes `<td rowspan="2">`:

```
<table style="border:3px solid black" border=1>
  <tr>
    <td>Case 1</td>
    <td colspan="2">Case 2</td>
  </tr>
  <tr>
    <td rowspan="2">Case 3</td>
    <td>Case 4</td>
    <td>Case 5</td>
  </tr>
  <tr>
    <td>Case 6</td>
    <td>Case 7</td>
  </tr>
</table>
```

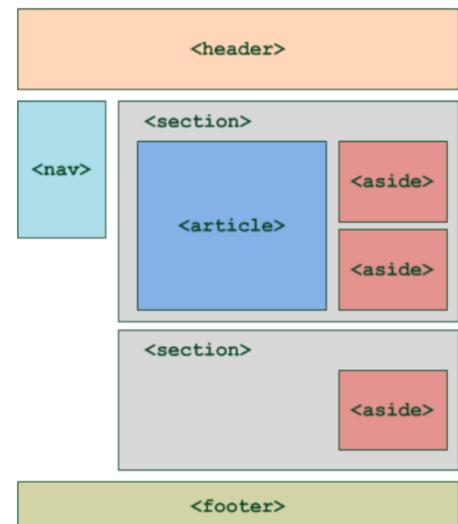
Case 1	Case 2
Case 3	Case 4 Case 5
	Case 6 Case 7

On peut aussi utiliser des listes, il existe 2 types de listes : les liste à point `<ul>` et les listes ordonnées, avec des nombres `<ol>`. Pour toutes les listes, chaque élément de la liste est défini par `<li>` élément de la liste `</li>`.

<code>&lt;ul&gt;</code> <code>&lt;li&gt; élément 1 &lt;/li&gt;</code> <code>&lt;li&gt; élément 2 &lt;/li&gt;</code> <code>&lt;/ul&gt;</code>	• élément 1 • élément 2	<code>&lt;ol&gt;</code> <code>&lt;li&gt; élément 1 &lt;/li&gt;</code> <code>&lt;li&gt; élément 2 &lt;/li&gt;</code> <code>&lt;/ol&gt;</code>	1. élément 1 2. élément 2
---	----------------------------	---	------------------------------

Pour aller plus loin, en général, une page web est constituée d'une **en-tête**, d'un **menu de navigation** et d'un **pied de page** et d'autre encore, appelés conteneurs.

- L'en-tête : `<header>...</header>`,
- Le pied de page : `<footer>...</footer>`,
- Le menu de navigation du site : `<nav> ...</nav>`.
- `<section>` Les sections regroupent des contenus en fonction de leur thématique. Elles sont au centre de la page et comportent généralement les articles et asides.
- `< article >` C'est pour une portion généralement autonome de la page, par exemple d'articles de journaux ou blogs.
- `< aside >` C'est une zone annexe. Elle peut contenir des informations complémentaires au document qu'on visualise, généralement placées sur le côté.



## V. Mettre en forme une page web

Le langage **CSS**, Cascading StyleSheet (feuilles de style en cascade), vient compléter le langage HTML.

La page web créée comme précédemment manque de mise en forme : taille du texte, couleur du fond, style de police, etc. Plusieurs solutions s'offrent à nous, mais une seule est efficace.



- La première consiste à ajouter à chaque balise des attributs de mise en forme. Le problème est que si on décide de ne plus mettre les titres de niveaux 1 en rouge mais en bleu, il nous faut reprendre toute la page pour modifier la valeur de l'attribut dans chaque balise `<h1>...<h1>`. Quel travail si en plus notre site comporte plusieurs pages...
- La deuxième consiste à créer une **feuille de style**, en CSS, qui vient compléter nos fichiers HTML. Ainsi, lorsqu'on souhaite modifier un style, on corrige seulement notre feuille de style.

Au début, le CSS n'existe pas, il a fait son apparition en **1996** alors que le HTML est né en 1991. La complexité des pages HTML à lancer la création du CSS.

### Prendre un bon départ

On indique dans l'en-tête de nos fichiers HTML la feuille de style à considérer en faisant le lien suivant :

```
<link rel="stylesheet" href="ma_feuille_style.css" type="text/css"/>
```

Dans le fichier CSS, la mise en forme d'une balise se fait de la façon suivante :

Par exemple voici les propriétés données au titre de premier niveau <h1>:

```
nom_balise
{
    attribut1 : valeur1 ;
    attribut2 : valeur2 ;
    ...
}
```

```
mon_style.css
1 h1
2 {
3     color:blue;
4     text-align:center;
5     font-size:15px;
6 }
```

On peut faire des commentaires dans un fichier css en utilisant : /\* COMMENTAIRE \*/

Pour d'autres propriétés se référer au site <http://www.w3schools.com/css/default.asp>.

On peut aussi créer des **classes** en introduisant le paramètre class dans la balise de la manière suivant :

```
<balise class="nom_de_la_classe"> ... </balise>
```

Pour un format qui n'est pas spécifique à une balise donnée mais qui peut être utilisé quelle que soit la balise, on utilise une classe de format. On désigne cette classe dans le fichier css en débutant son nom par un point. (attention, le nom ne doit pas commencer par un chiffre !) On pourra alors spécifier lorsqu'on veut utiliser cet attribut de style en l'appelant à l'intérieur d'une balise avec class="nom\_classe"

Dans la feuille de style, on définit cette classe par .nom\_de\_la\_classe.

Exemple ligne 3 du tableau ci-dessous

Une classe peut être définie dans une division à part avec la balise :

```
<div class="nom_de_la_classe"> ... </div>
```

Quelques exemples de mises en forme

CSS	HTML	Affichage
p {         color: green;         text-align: center;     }	<p> Tout ce qui se trouve entre les balises sera en vert et centré </p>	Tout ce qui se trouve entre les balises sera en vert et centré
#paraV {         color: green;         text-align: center;     }	<p> Si je ne précise rien, le format reste basique <p> <p id="paraV"> Mais si j'appelle l'attribut, le format change </p>	Si je ne précise rien, le format reste basique Mais si j'appelle l'attribut, le format change
.EnVert {         color: green;         text-align: center;     }	<h1 class="EnVert"> Grand titre centré en vert </h1> <p class="EnVert"> Paragraphe centré en vert </p>	Grand titre centré en vert Paragraphe centré en vert
p, h1, h2 {         color: green;         text-align: center;     }	<h1> Les grands titres, </h1> <h2> les moyens titres </h2> <p> et les paragraphes auront un format similaire </p>	Les grands titres, les moyens titres et les paragraphes auront un format similaire

Exemples :

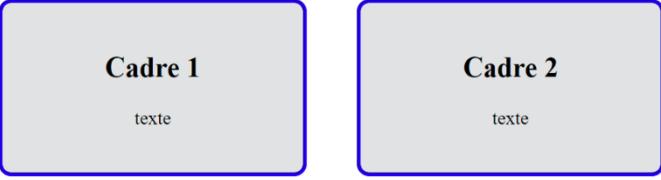
Fichier gestion\_blocs.html

```
<!DOCTYPE HTML>
<html>
<head>
    <meta http-equiv="Content-type" content="text/html"; charset="utf-8"> ,
    <title>Gestion cadres</title>
    <link href="gestion_cadre.css" rel="stylesheet" type="text/css" />
</head>
<body>
    <div class="cadreThemes">
        <h2>Cadre 1</h2>
        <p>texte</p>
    </div>
    <div class="cadreThemes">
        <h2>Cadre 2</h2>
        <p>texte</p>
    </div>
</body>
</html>
```

Fichier gestion\_cadre.css

```
.cadreThemes{
    /*dimensions du cadre*/
    width: 200px;           /*longeur*/
    /*height:200px;*/        /*largeur sinon auto*/
    /*placement cadre*/
    float: left;            /*à gauche*/
    margin-top: 20px;        /*haut*/
    margin-bottom: 20px;     /*bas*/
    margin-left: 20px;       /*gauche*/
    margin-right: 20px;      /*droite*/
    /*mise en forme texte*/
    background: #E0E2E4;    /*couleur du fond*/
    font: 15px Aladin;      /*taille et police*/
    text-align: center;      /*alignement texte*/
    padding: 20px;           /*marge avec le texte*/
    /*bordures*/
    border: groove;          /*style*/
    border-color: blue;       /*couleur*/
    border-width: 3px;         /*largeur*/
    border-radius: 10px;       /*bordure arrondie*/}
```

Affichage dans le navigateur



### Exercice 1 A vous de jouer !

Créer une page web qui vous présente de façon attractive (étude, passion, métier futur, vacances, ...). Il faut obligatoirement :

- Une balise title
- Deux titres de niveau de niveaux différents
- Au moins deux paragraphes
- Une mise en forme avancée (police, alignement, couleurs)
- Au moins une image
- Au moins un lien vers une page web
- Une liste numérotée ou non numérotée

Les productions originales seront valorisées à condition qu'elles soient personnelles !

### Exercice 2 Avec du CSS

Reprendre la page web de l'exercice 1 et faire la mise en forme dans un fichier CSS.

### Exercice 3 Un mini site

Créer un mini-site respectant les caractéristiques suivantes.

- Il doit contenir 3 pages minimum : accueil – sujet 1 – sujet 2.
- Il faut un menu avec des boutons cliquables pour naviguer entre les différentes pages.
- La page d'accueil doit contenir la présentation de votre site.
- La page « sujet 1 » doit présenter une personnalité qui a marqué un tour dans le monde du numérique. Il faut obligatoirement des photos, des tableaux, des liens vers vos sources d'information.
- La page « sujet 2 » doit présenter une invention dans le monde du numérique. Il faut obligatoirement des photos, des tableaux, des liens vers vos sources d'information.

### **Des vidéos à regarder**

Les 4 vidéos ci-dessous abordent les notions de base nécessaires à la réalisation d'un site web avec HTML et CSS. Elles sont réalisées par Graven - Développement sur Youtube.

- [CREER UN SITE ? HTML/CSS #1 - LES BASES](#)
- [LES BASES DU CSS - CREER UN SITE ? HTML/CSS #2](#)
- [LE MENU DE NAVIGATION - CREER UN SITE ? HTML/CSS #3](#)
- [COMMENT STRUCTURER LA PAGE ? CREER UN SITE ? HTML/CSS #4](#)

## AIDE-MÉMOIRE HTML 5

BALISE	ATTRIBUT / DESCRIPTION		
<!-- Commentaire -->	Commentaire		
<!DOCTYPE html>	Déclaration du Doctype		
<b>BALISE DE PREMIER NIVEAU</b> – Code minimal d'une page web			
<html>	Indiquez au navigateur que le document est en HTML		
<head>	En-tête de la page	<body>	Corps de la page
<b>BALISE D'EN-TÈTE</b> – Entre les balises <head>			
<title>	Titre de la page, très important pour le référencement		
<script>	Insérer script	<style>	Insérer CSS
<noscript>	Message à afficher si le script n'est pas toléré		
<base />	URL par défaut	<base href="http://41mag.fr/" target="_blank" />	
<link />	CSS PAGE RSS FAVICON	<link rel="stylesheet" type="text/css" href="#" /> <link rel="start" href="index.html" /> <link rel="alternate" type="application/rss+xml" href="#" /> <link rel="shortcut icon" type="image/x-icon" href="#" />	
<meta />	TITLE DESCRIPTION KEYWORDS ROBOTS SYNDICATION-SOURCE ORIGINAL-SOURCE NOTTRANSLATE HTTP-EQUIV CHARSET HTTP-EQUIV REFRESH HTTP-EQUIV PRAGMA	Titre de la page Description de la page Mots-clés de la page Restrictions pour les robots Indique l'URL d'origine Indique que c'est l'original Ne sera pas traduit Jeux de caractères Rafraîchir la page Définit le cache du navig.	
<b>BALISE D'ARCHITECTURE</b>			
<header>	Définit l'en-tête d'une section ou d'une page		
<footer>	Définit le bas d'une section ou d'une page		
<hgroup>	Définit les informations d'en-tête d'une section ou d'une page		
<details>	Définit les détails d'un élément		
<summary>	Définit l'en-tête des détails d'un élément		
<menu>	Définit un menu en forme de liste		
<section>	Définit une section		
<article>	Définit un article		
<aside>	Définit un élément latéral		
<nav>	Définit un groupe de liens de navigations		
<iframe>	Introduit une page HTML dans une frame		
<div>	Calque ou section		
<span>	Section de type inline		
<b>BALISE DE STRUCTURATION DE DE TEXTE</b>			
<h1> à <h6>	Créer un titre		
<p>	Paragraphe	<a>	Lien
<strong>	Mise en exergue	<b>	Texte en gras
<em>	Italique en exergue	<i>	italique
<mark>	Marqueur de texte	<small>	Rétrécis le texte
<sub>	Mise en indice	<sup>	Mise en Exposant
<address>	Définit une adresse	<cite>	Citation
<abbr>	Abréviation	<dfn>	Définition
<del>	Texte supprimé	<ins>	Texte ajouté
<time>	Date / Horaire	<meter>	Mesure
<code>	Portion de code	<pre>	Texte préformaté
 	Saut de ligne	<wbr>	Empl pr saut 2 ligne
<blockquote>	Longue citation	<q>	Courte citation
<samp>	Echantillon	<var>	Variable
<kbd>	Raccourci clavier	<bdo>	Sens du texte

BALISE	ATTRIBUT / DESCRIPTION		
<b>BALISE DE LISTE</b>			
<ul>	Liste non-ordonné	<ol>	Liste ordonné
<li>	Elément de liste	<dl>	Liste de définition
<dt>	Terme à définir	<dd>	Définition du terme
<b>BALISE DE TABLEAU</b>			
<table>	Tableau	<caption>	Titre du tableau
<thead>	En-tête du tableau	<tbody>	Corps du tableau
<tr>	Ligne du tableau	<th>	Cellule d'en-tête
<td>	Cellule du tableau	<tfoot>	Bas du tableau
<col>	Colonne du tableau	<colgroup>	Groupe de colonne
<b>BALISE DE FORMULAIRE</b>			
<form>	Formulaire	<fieldset>	Regroupe plusieurs éléments du formulaire
<legend>	Titre d'un groupe	<label>	Titre d'un élément
<datalist>	Liste déroulante	<select>	Liste sélectionnable
<option>	Elément d'une liste	<optgroup>	Grp d'elts d'une list
<textarea>	Zone de texte	<keygen>	Génération d'une clé
<button>	Bouton cliquable	<command>	Bouton de commande
<output>	Définit un type de sortie		
<input />	button checkbox color date datetime	file hidden image month number	radio range reset search submit
	email	password	tel
<b>BALISE MULTIMEDIA</b>			
<area>	Zone cliquable à l'intérieur d'une image		
<audio>	Contenu audio	<canvas>	Graphique
<img />	Image ou photo	<progress>	Progression
<figure>	Groupe d'élément multimédia	<figcaption>	Légende du groupe d'élément
<video>	Vidéo	<source>	Source du media
<map>	Carte / image	<param>	Paramètre d'objet
<embed />	Contenu extérieur	<object>	Objet du cont. ext
<b>AUTRE</b>			
<rp>	Annotation pr le nav	<rt>	Explication ruby
<ruby>	Annotation ruby	<hr />	Barre horizontale
<b>ATTRIBUT STANDAR</b>			
Accesskey	Raccourci clavier	itemprop	Utilisé pr un gp d'élémt
class	Attribut une classe	context-menu	Menu contextuel d'elemnt
lang	Langage d'un élément	spellcheck	Correction automatique
data-	Définit un attribut	style	Applique un style
dir	Direction du texte	subject	Définit l'elmt correspondt
draggable	Element déplaçable	tabindex	Définit l'ordre d'un tabl0
hidden	Element caché	title	Titre de l'élément
id	Nomme un élément	contenteditable	Element editable
item	Utilisé pour un groupe d'élément		
<b>ÉVÉNEMENT</b>			
Pour la balise <body>	Pour formulaire	Pour la souris	Pour les medias
Onafterprint, onbeforeprint, onafterload, onblur, onerror, onfocus, onhaschange, onload, onmessage, onoffline, online, onpagehide, onpageshow, onpopstate, onredo, onresize, onstorage, onundo, onunload	Onblur, onchange, ondrag, ondragend, oncontextmenu, onfocus, onformchange, onforminput, oninput, oninvalid, onsubmit, onselect	Onclick, ondblclick, ondrag, ondragend, ondragenter, ondragleave, ondragover, ondragstart, ondrop, onmousedown, onmouseup, onmousemove, onmouseout, onmouseover, onwheel, onselect	Onabort, oncancelplay, oncancelplayrought, ondurationchange, onemptied, onended, onerror, onloaddata, onloadstart, onpause, onplay, onplaying, onprogress, onseeked, onsuspend, onwaiting, onvolumechange



## AIDE-MÉMOIRE CSS 3



### CSS Syntax

```
selector #id .class :pseudoclass ::pseudoelement [attr] {
    property : value ;
}
```

### Backgrounds

```
background-color: [color];           color: [color];
background-image: url(image.jpg);   none
background-repeat: repeat | repeat-x | repeat-y | no-repeat
background-attachment: scroll | fixed
background-position: [pos-x] [pos-y];
background: [color | image | repeat | attachment | position]
```

### Fonts

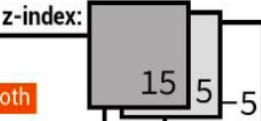
```
font-family: [font1], [font2], [font3], ... ;
    serif | sans-serif | cursive | fantasy | monospace
font-size: [size] xx-small | x-small | small | medium
    large | x-large | xx-large | smaller | larger
font-style: normal | italic | oblique
font-weight: [100-900] normal | bold | lighter | bolder
font: [style | variant | weight | size/height | family]
```

### Elements (types)

```
display: inline | block | inline-block | none | list-item
    table | table-cell | table-row
visibility: visible | hidden | collapse
```

### Displacement

```
z-index: [int]
float: none | left | right
clear: none | left | right | both
```



### Margins and paddings

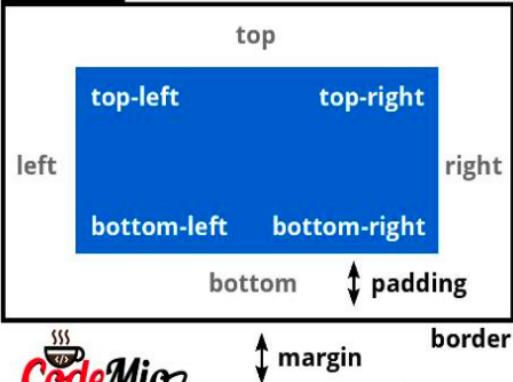
```
margin/padding: top | right | bottom | left
margin/padding: top | right | left | bottom
margin/padding: top | bottom | left | right
margin/padding: top | right | bottom | left
```

```
*-top
*-left
*-right
*-bottom
```

### Border

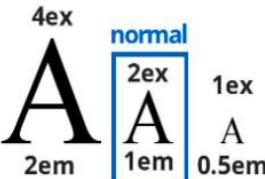
```
border-color: [color];
border-width: [size] thin | medium | thick
border-style: [style];
border: [width | style | color]
border-top-* | border-right-* | border-bottom-* | border-left-*
```

### Locations



### Colors

	Keywords: RoyalBlue; Hexadecimal: #4169E1; → #46E; RGB model: RGB(65,105,225); HSL model: HSL(225,71%,88%);		with alpha channel RGB(65,105,225, 0.5); HSL(225,71%,88%, 0.5); transparent
	currentColor		



### Tables

```
border-collapse: separate | collapse
border-spacing: [size];
caption-side: top | bottom
empty-cells: show | hide
table-layout: auto | fixed
```

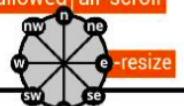
### Font (alignments and spacing)

```
letter-spacing: [size]; normal
line-height: [size]; normal
text-indent: [size];
word-spacing: [size]; normal
white-space: normal | nowrap | pre
    pre-line | pre-wrap
```

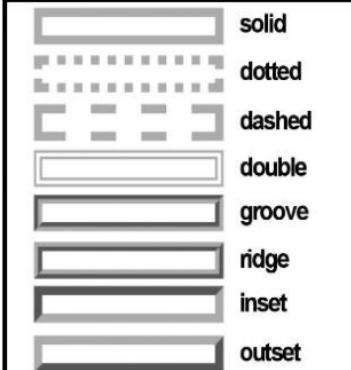
```
tab-size: [size];
text-align: left | center | right | justify
vertical-align: [size] baseline
    sub | super | top | middle | bottom
    text-top | text-bottom
```

### Mouse cursors

```
cursor: url(image.png)
    default | crosshair | help | move
    pointer | progress | text | wait
    none | context-menu | cell
    vertical-text | alias | copy
    no-drop | not-allowed | all-scroll
    col-resize | row-resize
```



### Styles



### Dimensions

```
max-width: [size]; none
min-width: [size]; none
width: [size] auto
*height
```

### Fonts (variants)

```
font-variant: normal | small-caps
text-decoration: none | underline | overline
    line-through
text-transform: none | capitalize
    lowercase | uppercase
```

### Outlines

```
outline-color: [color]; invert
outline-style: [style];
outline-width: [size]; thin | medium | thick
outline: [color | style | width]
```

### Lists

```
list-style-image: url(image.png); none
list-style-position: inside | outside
list-style-type: disc | circle | square | none
    1. | 2. | 01. | 02. | decimal | decimal-leading-zero
    a. | b. | A. | B. | lower-alpha | upper-alpha
    i. | ii. | I. | II. | lower-roman | upper-roman
list-style: type | position | image
```

### Position types

```
position: static | absolute | relative | fixed
top/right/bottom/left: [size] auto
clip-path: url(shape.svg) shape | auto
overflow: visible | hidden | scroll | auto
```

### Columns

```
column-width: [size];
column-count: [number]; auto
columns: width | count
```

### Columns separator

```
column-rule-width: [size];
column-rule-style: [style];
column-rule-color: [color];
column-rule: width | style | color
```

```
column-gap: [size]; normal
column-span: [number]; all
column-fill: balance | auto
```



## AIDE-MÉMOIRE CSS 3 (SUITE)

### Gradients

#### background-image:

linear-gradient([dir], [col1], [col2]...);  
radial-gradient([shape] [size] at [pos], [col1], [col2], ...);  
repeating-linear-gradient(...);  
repeating-radial-gradient(...);

### Shadows

text-shadow: [pos-x] [pos-y] [blur] [color]; none  
box-shadow: [pos-x] [pos-y] [blur] [size] [color]; none inset

### Shadows or backgrounds

background-image: url(back1.png), url(back2.png), ...;  
background-repeat: no-repeat, repeat-x, ...;

### Backs

background-clip: border-box | padding-box | content-box  
background-origin: padding-box | border-box | content-box  
background-size: [size-w] [size-h]; cover | contain auto  
background: color | position | size | repeat | origin | clip | att | img

### Round borders CSS3

border-radius: top right bottom left border-top-left-\*  
border-radius: top bottom left right border-top-right-\*  
border-radius: top right bottom left border-bottom-left-\*  
border-radius: top right bottom left border-bottom-right-\*

### Borders with images

border-image-outset: [size]  
border-image-repeat: stretch | repeat | round | space  
border-image-slice: top right bottom left  
border-image-source: url(image.png)  
border-image-width: [size]  
border-image: source | slice | width | outset | repeat

### Transitions

transition-property: [css-property]; none | all  
transition-duration: [time];  
transition-timing-function: [timing-function]  
transition-delay: [time];  
transition: property | duration | t-function | delay

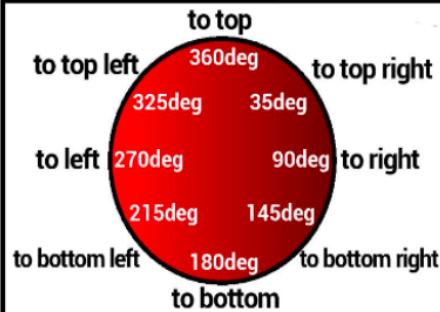
### Transformaciones

transform-origin: [pos-x] [pos-y] [pos-z];  
transform-style: flat | preserve-3d

### Animations

animation-name: [name]; none  
animation-duration: [time];  
animation-timing-function: →  
animation-delay: [time];  
animation-iteration-count: [number]; infinite  
animation-direction: normal | reverse | alternate | alternate-reverse  
animation-fill-mode: none | forwards | backwards | both  
animation-play-state: running | paused  
animation: name | duration | timing-func | delay | i-c | dir | f-m | p-s

### Lineal gradient (Direction)



### Media

```
@media print { propiedad : valor; }
@media screen { propiedad : valor; }
@media screen and (max-width: 640px) {
    propiedad : valor;
}
```

### Radial gradient (Options)

shape: ellipse | circle  
size: [size] farthest-corner | closest-corner  
farthest-side | closest-side  
pos: center | top | left | right | bottom  
top left | top right | bottom left | bottom right

### Typograph CSS3

```
@font-face {
    font-family: 'Open Sans';
    font-weight: 300;
    src: local('Open Sans'),
        url(file.ttf) format('truetype'),
        url(file.woff) format('woff'); }
```

<http://fonts.googleapis.com/css?family=Open+Sans:300,400|Roboto:400>

### Pagination

```
@page {
    size: [width] [height];
    landscape | portrait | auto
    margin: [...]
    orphans: [number];
    widows: [number]; }
```



transform: rotateX(deg\_x);  
transform: rotateY(deg\_y);  
transform: rotate(deg);



transform: scaleX(x);  
transform: scaleY(y);  
transform: scale(x, y);



transform: translateX(x);  
transform: translateY(y);  
transform: translate(x, y);



transform: skewX(deg\_x);  
transform: skewY(deg\_y);  
transform: skew(deg, deg);

### Fonts CSS3

font-stretch: ultra-condensed  
extra-condensed | condensed | semi-condensed  
normal | semi-expanded | expanded  
extra-expanded | ultra-expanded  
text-overflow: [text]; clip | ellipsis  
text-justify: auto | inter-word | distribute | none  
font-size-adjust: [number] none



### CSS Filters

filter: [filter](n)  
filter-func (n)  
grayscale: [0...1]  
blur: [size]  
sepia: [0...1]  
saturate: [0...1]  
opacity: [0...1]  
brightness: [0...1]  
contrast: [0...1]  
hue-rotate: [deg]  
invert: [0...1]  
filter: f1(n) f2(n) ...

### Keyframes

```
@keyframes nameanimation {
    0% { propiedad: valor }
    ...
    100% { propiedad: valor }
}
```



### @-vendor-keyframes

```
0% = from  
100% = to
```

