



Notes de cours HTML5 & CSS3



Tic Tac Toe - Maude Maréchal

1. Table des matières

1. Introduction	3
2. Besoins et environnement	3
3. Structure du document.....	5
4. Mise en relief du texte html	7
5. Mise en relief du texte Html5	15
6. Introduction CSS	16
7. CSS, propriétés de mise en forme	18
8. Les règles d'écriture.....	21
9. Les formulaires.....	23
10. Squelette de la page, les balises génériques <div>	27
11. Les éléments de section	28
12. Nouveaux éléments sémantiques	30
13. Rich media, nouveaux éléments multimédias	33
14. Le positionnement CSS.....	36
15. Trucs et astuces CSS3.....	37



1. Introduction

Qu'est-ce qu'une page html ?

HTML = Hypertext Markup Language

C'est en 1989 que les premières pages **HTML** voient le jour. Ce langage sous forme de balisage (ci-dessous) permet de corréler les pages du site via des hyperliens. Ces fichiers peuvent être des formats de présentation (CSS), des langages de programmation (Javascript, PHP), des ressources multimédias (vidéos, images), des bases de données. Les balises contiennent des instructions sur la mise en forme du texte, lesquelles sont interprétées par les différents navigateurs. Il s'agit donc d'un langage de mise en forme du texte. C'est l'HTML qui définit qu'un mot est gras, que tel paragraphe est un titre ou un sous-titre,...

A partir des années 2000, l'utilisation du **CSS** se généralise. Ce langage est uniquement orienté vers le graphisme : couleur de fond et des bordures, police de caractère, positionnement dans la page, superposition,... Le CSS a permis une simplification du code HTML et des mises à jour, d'uniformiser les pages du site web,...

Nom	Modifié le	Type	Taille
 mini-position	14/05/2012 08:16	Chrome HTML Document	1 Ko
 style	24/09/2012 13:38	Document de feuille de style en cascade	1 Ko

- 1989 – Début de l'html
- 1995 – Spécification html 2.0 (structure en tableau, iframe)
- 1997 – Spécification html 4.0
- 2000 – Spécification xhtml (structure div et positionnement CSS)
- 2007 – Spécification html5 (le xhtml commence seulement à se généraliser)

Évolution de l'HTML5

L'html5 est l'association de trois langages : HTML (version 4) + CSS (version 3) et JavaScript.

L'évolution de HTML5 repose sur les besoins des usages contemporains du web, tel que l'intégration audio et vidéo, la géolocalisation, la fluidité des conteneurs (drag and drop), la gestion étendue des formulaires,... et surtout une simplification de la grammaire.

Qu'est-ce qu'une balise

Les balises, parfois appelées **marqueurs** ou **repères** ou **tags** en anglais, sont des éléments permettant de mettre en forme le texte.

' < 'et' > '

L'ensemble de ces balises définissent la syntaxe de votre page html.

Exemple : Ceci est un **Titre1** = <h1> Blablabla </h1>

Ceci un **paragraphe** = <p> Blabla </p>

<h2>Passez une douce nuit dans nos cocons de plumes</h2>

2. Besoins et environnement

Comment visualiser une page html (utilisateur)

La visualisation d'une page html se fait par l'utilisation d'un navigateur. Ils permettent d'envoyer des requêtes au serveur web et d'afficher son résultat de réponse.



Attention : les navigateurs (et leurs versions) ont tous une manière différente d'interpréter votre code. Il est donc important de tester vos pages dans plusieurs d'entre eux. Cette différence se marquera encore plus lorsque vous utiliserez des feuilles de style CSS.

Comment produire une page html

1. Pour coder en Html le Bloc-notes peut suffire, mais vous pouvez également utiliser un éditeur de code tel que Notepad++, Dreamweaver,... Ceux-ci vous aideront à écrire votre code grâce à l'auto-complétion et la coloration du texte.
 - a. Clic 'droit'.
 - b. Sélectionner '**Ouvrir avec ...**' et choisir le Blocnote ou un éditeur de code (**DreamWeaver**).
2. Vous devrez respecter la structure de base (html + head + body).
3. Tous vos documents doivent être enregistrés avec l'extension de fichier '**.html**'
4. Vous n'utiliserez pas pour nommer vos fichiers :
 - c. pas d'accent,
 - d. pas d'espace,
 - e. pas de caractères spéciaux,
 - f. pas de signe de ponctuation,
 - g. pas de majuscule,

DONC a-->z et 0-->9.

Comment définir mon espace de travail ?

Images = les images de la structure de votre site.

Photos = les photos (images) de vos articles.

index.html = 1ere page de votre site.

pagenbr.html = les autres pages de votre site.

style.css = la feuille de style de votre site.

Remarque : C'est votre code html qui lie tous ces fichiers (html + images + CSS).

3. Structure du document

Html5, une syntaxe permissive

- **Majuscules autorisées** pour le nom des éléments et attributs
`<p>` → `<P>`
- Guillemets non obligatoires pour les valeurs
`<p class="code">` → `<p class=code>`
- **Balises fermantes non obligatoire** pour les éléments `<p>`, `<dd>`, `<dt>`, ``, `<optgroup>`, `<option>`, `<td>`, `<th>`, `<tr>`, `<thead>` et `<tfoot>`
- **Balises ouvrantes et fermantes non obligatoires** pour les éléments `<html>`, `<head>`, `<body>`, `<thead>`, `<tfoot>` et `<tbody>`

```
<!doctype html>
<meta charset=UTF-8>
<title></title>
<p>Voici une page en HTML5
```

Remarque : cette utilisation extrême de l'HTML5 nécessite une grande précaution (erreur, oubli), il est conseillé de conserver autant que possible la rigueur de XHTML.

Comparer la structure Html et html5

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="fr">
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8"
>
<title>Titre affiché dans la barre du navigateur</title>
<link href="style.css" rel="stylesheet" type="text/css" >
</head>
<body>
<p> Contenu de la page </p>
</body>
</html>
```

<html> Début d'un document de type HTML.

<head> Début de la zone d'en-tête.

(Prologue au document, contenant des informations destinées au browser)

</head> Fin de la zone d'en-tête.

<title> Début du titre de la page. Celui-ci est affiché dans haut à gauche de la fenêtre du navigateur.

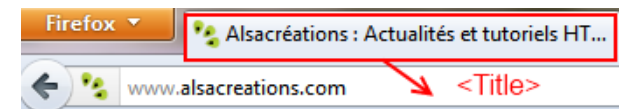
</title> Fin du titre de la page.

<body> Début du document proprement dit.

ICI VOUS ENTREREZ VOS INFORMATIONS

</body> Fin du document proprement dit.

</html>



Doctype

La doctype est la déclaration du type de document, elle indique au navigateur la version HTML utilisée. L'interprétation du code par celui-ci en dépend.

XHTML 1.0 strict

Sans utilisation de balises obsolètes.

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

XHTML 1.0 Transitional

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

HTML5

```
< !DOCTYPE html>
```

Codage des caractères

Il permet de gérer tous les caractères unicodes. Si ce codage est correctement déclaré, il sera possible d'encoder directement les lettres accentuées.

Exemple : à → à

Méta charset XHTML

```
<meta http-equiv="content-type" content="text/html; charset=utf-8" />
```

Méta charset HTML5

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

Lien vers la CSS

Le CSS offre une stylisation indépendante de la structure du document, c'est l'habillage de votre structure. L'HTML décrit l'architecture interne, tandis que le CSS décrit les aspects de la présentation (aspects graphiques). Cette séparation permet d'améliorer l'accessibilité, de changer plus facilement de structure et de présentation, et réduit la complexité.

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

Appel d'un script javascript

```
<script src="file.js"> </script>
```

4. Mise en relief du texte html

Les principales balises (éléments syntaxiques)

En xhtml et html5, l'utilisation de ces balises fait référence à la **sémantique** du texte et non à une simple apparence visuelle.

Exemple : `<i>` = italique et `` = **emphase** ; `<g>` = gras et `` = **évidence**

- Titre-----`<h1>...</h1>`
- Sous-titre----- `<h2> ...</h2>`
- Sous-sous-titre ----- `<h3>...</h3>`
- ...
- Paragraphe -----`<p>...</p>`
- Gras -----`...`
- Italique ----- `...`
- Liste à puces -----`...` (unordered list)
- Liste numérotée -----`...` (order list)
- Élément de liste-----`...`
- Citations -----`<blockquote><p>... </p></blockquote>`
- Acronyme / abréviation ---`<abbr> ...</abbr>`
- Indice H₂O ----- H`₂`O
- Exposant ab² ----- ab`²`
- Saut de ligne -----`
`
- Espace insécable -----` `
- Ligne horizontale ----- `<hr/>`
- commentaire -----`<!-- ... -->`
- Lien ----- `` Blabla visible ``
- Image----- ``

Les listes de définitions

dl = liste de définition dt = titre de la définition dd = description de la définition

```
<body>
  <dl>
    <dt>W3C</dt>
    <dd>World Wide Web Consortium</dd>
    <dd>L'organisme de standardisation des technologies fondatrices du Web</dd>
    <dd>But initial : éviter la fragmentation</dd>
  </dl>
</body>
```

Les listes

ul = *unordered list* ou **ol** = *order list*

li = chaque élément de la liste

Non numérotée (unordered list) <ul style="list-style-type: none"> Élément de liste Élément de liste Élément de liste 	<pre> Élément de liste Élément de liste Élément de liste </pre>
Numérotée (ordered list) <ol style="list-style-type: none"> Élément de liste Élément de liste Élément de liste 	<pre> Élément de liste Élément de liste Élément de liste </pre>
Liste imbriquée <ol style="list-style-type: none"> Élément de liste <ul style="list-style-type: none"> Sous-liste imbriquée Sous-liste imbriquée 	<pre> Élément de liste Élément de liste Élément de liste </pre>

La portée `...`

La portée est un groupe de mots ou de lettres à l'intérieur d'un paragraphe. Elle peut juste être mise en forme.

Remarque : Préférez utiliser `` et `` à `` pour une mise en relief du texte, car ces balises ont une signification sémantique.

Les tableaux

Tableau	<code><table>...</table></code>
Légende d'un tableau	<code><caption>...</caption></code> (facultatif)
Groupe de colonnes	<code><colgroup> ...</colgroup></code>
Propriété de style de colonne	<code><col>...</col></code>
Groupe de cellules d'en-tête (table header)	<code><thead>...</thead></code>
Groupe de cellules de corps (table body)	<code><tbody>...</tbody></code>
Groupe de cellules de pied (table footer)	<code><tfoot>...</tfoot></code>
Ligne (table row)	<code><tr>...</tr></code>
Titre (cellule d'en-tête)	<code><th>...</th></code>
Cellule (table data)	<code><td>...</td></code>

Les attributs NON SUPPORTÉ EN HTML5

`<table border="1px" cellspacing="0" cellpadding="2px" />`

- Bord du tableau `border=" "`
- Espace entre les cellules `cellspacing=" "`
- Marge intérieure `cellpadding=" "`
- Résumé du tableau (règle d'accessibilité) `<table summary="résumé du contenu du tableau" >`

Fusion des cellules

`<td colspan="2" rowspan="2" />`

- Fusion horizontale `colspan=" "`
- Fusion verticale `rowspan=" "` Exercices tableau

```
<table border="1px">
  <tr>
    <td>une seule cellule</td>
  </tr>
</table>
<br />
<table border="1px">
  <tr>
    <td>première ligne</td>
  </tr>
  <tr>
    <td>deuxième ligne</td>
  </tr>
</table>
<br />
<table border="1px">
  <tr>
    <td>première colonne</td>
    <td>deuxième colonne</td>
  </tr>
</table>
```

une seule cellule

première ligne
deuxième ligne

première colonne	deuxième colonne
------------------	------------------

Légende à afficher

Titre d'en-tête	1er cellule	2eme cellule
Titre de la cellule dans le body	1er cellule	2eme cellule
Titre de la cellule dans le body	1er cellule	2eme cellule
Titre du pied	1er cellule	2eme cellule

```

<table summary="descriptif du contenu du tableau" border="1px">
  <caption> Légende à afficher</caption>

  <thead>
    <tr>
      <th>Titre d'en-tête</th>
      <td>1er cellule</td>
      <td>2eme cellule</td>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th>Titre de la cellule dans le body</th>
      <td>1er cellule</td>
      <td>2eme cellule</td>
    </tr>
    <tr>
      <th>Titre de la cellule dans le body</th>
      <td>1er cellule</td>
      <td>2eme cellule</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <th>Titre du pied</th>
      <td>1er cellule</td>
      <td>2eme cellule</td>
    </tr>
  </tfoot>
</table>

```

```

<table border>
  <tr>
    <td rowspan=2>Pages web</td>
    <td>code CSS</td>
  </tr>
  <tr>
    <td>code html</td>
  </tr>
</table>
<br />
<table border="1px">
  <tr>
    <td colspan=2>Pages web</td>
  </tr>
  <tr>
    <td>CSS</td>
    <td>html</td>
  </tr>
</table>

```

Pages web	code CSS
	code html

Pages web
CSS html

o	o
o	o
o	o
o	o

```

<table border="1px">
  <tr>
    <td>1</td>
    <td>2</td>
    <td>3</td>
    <td>4</td>
  </tr>
  <tr>
    <td>5</td>
    <td>6</td>
    <td>7</td>
    <td>8</td>
  </tr>
  <tr>
    <td>9</td>
    <td>10</td>
    <td>11</td>
    <td>12</td>
  </tr>
  <tr>
    <td>13</td>
    <td>14</td>
    <td>15</td>
    <td>16</td>
  </tr>
</table>

```

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

```

<table border="1px">
  <tr>
    <td colspan="3">1</td>
    <td>4</td>
  </tr>
  <tr>
    <td rowspan="3">5</td>
    <td>6</td>
    <td>7</td>
    <td>8</td>
  </tr>
  <tr>
    <td>10</td>
    <td colspan="2" rowspan="2">11</td>
  </tr>
  <tr>
    <td>14</td>
  </tr>
</table>

```

1		4	
	6	7	8
5	10		11
	14		

L'attribut `span="nbr"` permet de définir un nombre de colonne sur lesquelles des propriétés CSS (`class="columns"`) peuvent s'appliquer.

```

<!-- Tableau utilisant colgroup -->
<table>
  <colgroup span="4" class="columns"></colgroup>
  <tr>
    <th>Pays</th>
    <th>Capitales</th>
    <th>Population</th>
    <th>Langue</th>
  </tr>
  <tr>
    <td>USA</td>
    <td>Washington D.C.</td>
    <td>309 millions</td>
    <td>Anglais</td>
  </tr>
  <tr>
    <td>Suède</td>
    <td>Stockholm</td>
    <td>9 millions</td>
    <td>Suédois</td>
  </tr>
</table>

```

Pays	Capitales	Population	Langue
USA	Washington D.C.	309 millions	Anglais
Suède	Stockholm	9 millions	Suédois

```

<!-- Tableau utilisant colgroup et col -->
<table>
  <colgroup>
    <col class="column1">
    <col class="columns2plus3" span="2">
  </colgroup>
  <tr>
    <th>Citron vert</th>
    <th>Citron</th>
    <th>Orange</th>
  </tr>
  <tr>
    <td>Vert</td>
    <td>Jaune</td>
    <td>Orange</td>
  </tr>
</table>

```

Citron vert	Citron	Orange
Vert	Jaune	Orange

Remarque : L'élément `<colgroup>` doit apparaître **après** tout élément `<caption>` et **avant** les éléments `<thead>`, `<th>`, `<tbody>`, `<tfoot>` et `<tr>`.

Les liens

Les liens sont la base même d'internet. Ils vous permettent de passer d'une page à l'autre de votre site (lien interne), d'ouvrir un e-mail vers un destinataire prédéfini (mailto), de vous rendre sur un autre site (lien externe). Ils peuvent être placés sur un mot, une phrase,... une image.

- **href** : adresse du lien
- **title** : texte utilisé dans l'infobulle

```
<body>
  |<a href="page2.html" title="Ce lien ouvre ma deuxième page"> Texte du Lien </a>
</body>
```

Texte du Lien

Ce lien ouvre ma deuxième page

Exercice 1

1. Créer deux page html ;
2. Intitulez-les 'page1.html' & 'page2.html' ;
3. Dans chacune des pages insérer 2 liens :

```
<p>
  <a href="page1.html"> Accueil </a> | <a href="page2.html"> Page 2 </a>
</p>
```

4. Dans la **page1**, insérer un titre <h1>Page d'accueil</h1>
Dans la **page2**, insérer un titre <h1>Page 2</h1>
5. Visualisez le résultat dans un navigateur

Exercice 2

1. Reproduisez le schéma suivant :

2. Quelles conclusions ?

Quand je remonte dans l'arborescence
(vers un dossier parent) :

.....
.....
.....

Quand je descends dans l'arborescence
(vers un dossier enfant) :

.....
.....
.....

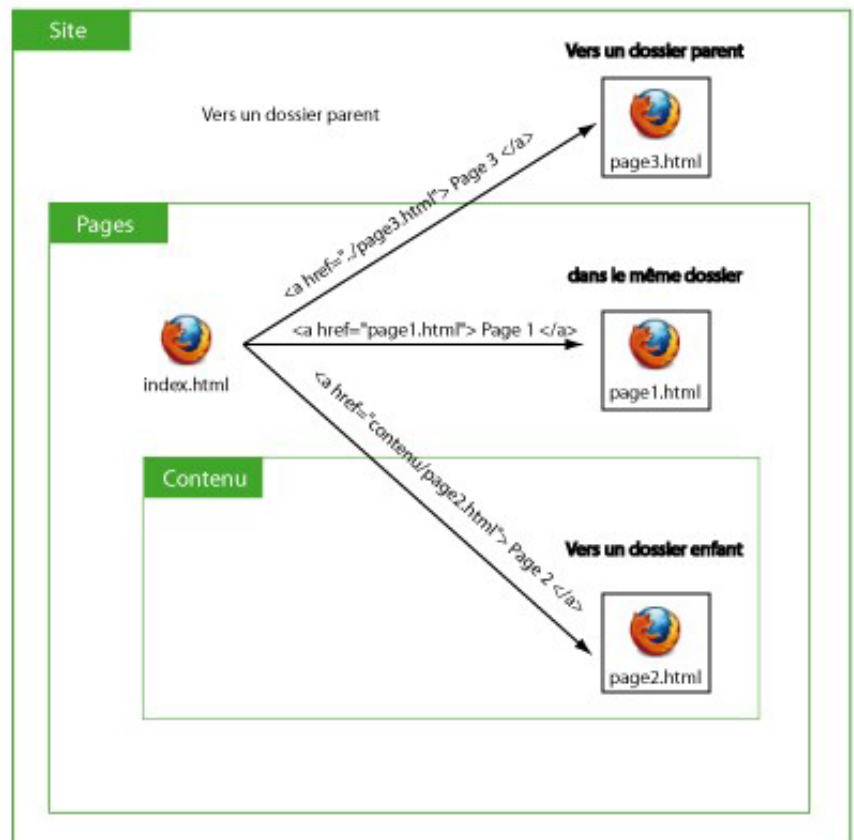
Quand je reste au même niveau de
l'arborescence (pages côte à côte)

.....
.....

3. Traduisez les chemins suivants :

"../images/animaux/chien.gif"

"template/pages/header.php"



Les adresses absolues et relatives

Une adresse absolue est l'adresse complète du site avec l'indication du chemin du document cible.

Ex : <http://www.lesite/template/pages/contact.html>

Une adresse relative est le chemin du document cible à partir du dossier ouvert.

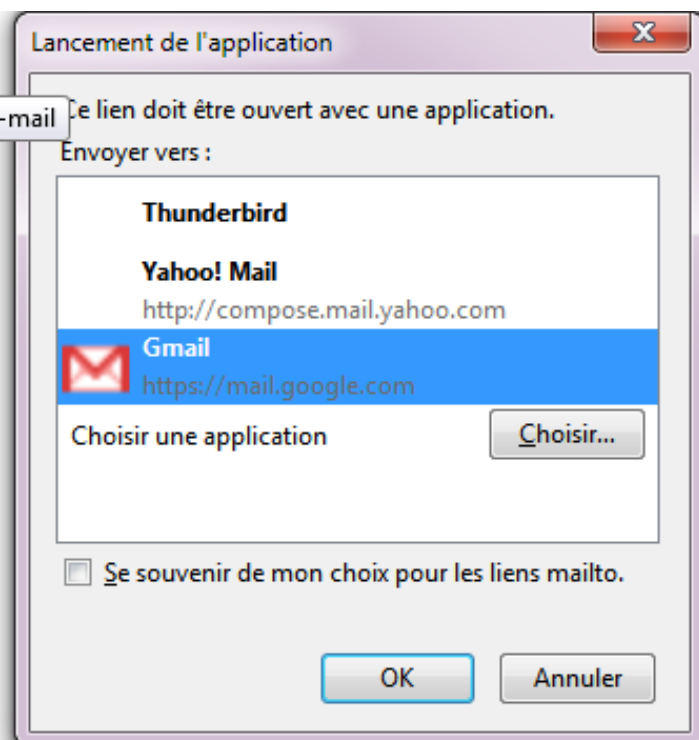
Ex : </images/bkgfooter.png>

Les liens vers une adresse e-mail

```
<body>
  <a href="mailto:maudemarechal@gmail.com" title="Envoyez moi un e-mail">
    Ce lien ouvre un e-mail
  </a>
</body>
```

[Ce lien ouvre un e-mail](mailto:maudemarechal@gmail.com)

Envoyez moi un e-mail



Remarque : il est intéressant de faire figurer le contact dans une balise `<address>...</address>`

```
<address>
  <a href="mailto :... ">...</a>
</address>
```

5. Mise en relief du texte Html5

Tout est possible en HTML5 : majuscules, minuscules, guillemets ou pas, balise auto-fermante ou non,...

Balises obsolètes

<acronym>

<big>
<center>
<strike>
<font-size>
<text-align>
<bgcolor>
<cellspacing>
<cellpadding>
<valign>
<hgroup> (balise html5 déjà obsolète)

Comeback

 → simple indication stylistique sans exprimer une importance supplémentaire.

<i> → signifie une humeur alternative, un changement de ton, mais n'est pas une emphase.

Remarque : = important et = emphase¹.

Lifting

<cite> → titre d'une œuvre, mais vous ne pouvez plus l'utiliser pour désigner le nom de l'auteur.

<small> → "voici des petits caractères", utiliser pour les mentions spécifiques ou légales (jargon juridique, conditions d'utilisation).

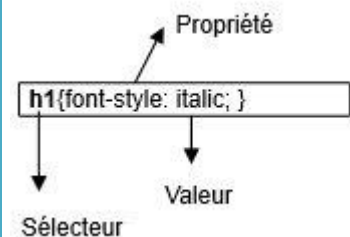
<a> → Les éléments de lien peuvent maintenant être des conteneurs.

```
<a href= "http://www.lelien.be">  
  <h2>Mon titre</h2>  
  <p>Texte du paragraphe.</p>  
</a>
```

¹ Emphase : désigne tout procédé d'insistance ou de mise en relief.

6. Introduction CSS

Le sélecteur CSS	La balise HTML
div { ... }	<div> </div>
h1 { ... }	<h1>... </h1>
p { ... }	<p>... </p>



1. Sélecteur simple

body{ ... } div{ ... } h1{ ... } p{ ... }

2. Class

La class définit une catégorie de balise. Elle peut être **utilisée plusieurs fois** dans la même page.

CSS	HTML	Remarque
p.tomate { font-color: red; }	<p class= "tomate"> ...<p>	La class tomate s'applique uniquement à des balises <p>.
.tomate { font-color: red; }	<p class= "tomate"> ...<p> <h1 class= "tomate"> ...<h1>	Moins spécifique, tomate s'applique à plusieurs types de balises.
p.tomate { font-color: red; } p.cerise { font-size: 60%; }	<p class= "tomate cerise"> ...<p>	Addition des propriétés : rouge et 60%. Les classes tomate et cerise s'appliquent uniquement à des balises.

3. Identifiant

L'identifiant définit **un élément unique** (il ne peut avoir qu'un seul élément identifiant par page)

CSS	HTML	Remarque
div#menu { ... } p#chapeau { ... }	<div id= "menu"> ... </div> <p id= "chapeau"> ... </p>	
#news { ... }	<... id= "news"> ... </p>	L'identifiant sans nom de balise est lui aussi unique dans la page et s'applique à plusieurs types de balises.
p.tomate#menu { ... }	<p id= "menu" class="tomate"> ... </p>	Il est possible d'utiliser une class et un identifiant en même temps

Classe

(Plusieurs fois par page)

CSS

.tomate

Xhtml

(élément en ligne)

<div **class**="tomate"> </div>
(élément type bloc)

Identifiant

(1 x par page)

CSS

#tomate

Xhtml

<div **id**="tomate"> </div>
(élément type bloc)

4. Pseudo-classes

Accolées à une balise, elles apportent des précisions sur le comportement des sélecteurs.

:link → lien hypertexte qui n'a pas encore été visité

:visited → lien visité (dans l'historique du navigateur)

:hover → élément survolé par le curseur de la souris

:active → élément activé, le bouton de la souris est enfoncé sur ce lien

Exemple :

`a:hover {color: green; }` au survol du curseur le texte devient vert

`a.menu:hover {color: green; }` au survol du curseur de la balise `...`, le texte devient vert.

Remarque, certains navigateurs sont sensibles à l'ordre dans lequel les sélecteurs ont été encodés :

`a:link`, `a:visited`, `a:hover` et `a:active`.

5. Pseudo-élément

Apportent d'autres types de précisions.

Première **lettre** du bloc `p:first-letter { font-size: 150%; }`

Première **ligne** du bloc `p:first-line { color: #FF0000; }`

7. CSS, propriétés de mise en forme

Certaines des propriétés définies dans les feuilles de style sont transmises aux éléments imbriqués
Exemple : Hérité → font-family Non hérité → border

Propriété	Écriture	Remarque
font-family (héritée)	p {font-family: Arial, Verdana, sans-serif; } p { font-family: "Times New Roman", serif; }	Les polices de caractère Sans-serif = famille de caractère sans empattements
font-size (héritée)	p { font-size: 80%; } h2 { font-size: 1.3em; }	Utilisez les unités relatives (% , em, ex)
Color (héritée)	p { color: #6e05c3; } h1 {color : red ;}	La couleur des caractères
font-weight (héritée)	.chapeau { font-weight: bold; }	normal - lighter - bolder - (un nombre)
font-style (héritée)	p { font-style: italic; }	Italic – Normal – Oblique - Inherit
text-decoration (héritée)	a:hover { text-decoration: none; } h1 { text-decoration: underline overline; }	<u>underline</u> – overline – line-through – none
text-transform	h1 {text-transform: uppercase; }	lowercase : tout en minuscule uppercase : tout en majuscule
font-variant	h2 {font-variant: small-caps; }	petite majuscule
vertical-align (non héritée)	.exposant { vertical-align: super; } .indice { vertical-align: -50%; }	baseline – super – sub – middle – top – bottom – valeur - %
font Propriété raccourcie (héritée)	h3{font: bold 140% verdana sans-serif ; }	font-style - font-variant – font-weight - font-size, line-height – font-family (seule la propriété font-family est obligatoire)

text-align (héritée)	p { text-align: justify; }	left - right - center – justify
text-indent (héritée)	p { text-indent: 2%; } p { text-indent: -2%; padding-left: 2%; }	Retrait de la première ligne d'un paragraphe Retrait négatif de la première ligne
line-height (héritée)	p.news { line-height: 2em; }	Interligne
letter-spacing (héritée)	.pluslarge { letter-spacing: 0.5em; }	Interlettrage
word-spacing (héritée)	.grandespaces { word-spacing: 0.5em; }	Intermot Ne pas utiliser les %
content (non héritée)	p.remarque:before {content:url(image.gif);} .citation:before {content: open-quote; } .citation:after {content:close-quote;} p.note:before { content: "Nota bene : ";	Affiche automatiquement un texte, une image,... avec les pseudo-éléments :before et :after
border-style (non héritée)	h2 { border-style: solid; }	none - hidden (aucune) - solid (plein) - double - dotted (pointillés) - dashed (tirets) - groove(creux) - ridge(relief) - inset(creux ombré)- outset(relief ombré)
border-width (non héritée)	p.news { border-width: 2px; }	valeur numérique en em, px, mais pas en %
border-color (non héritée)	div.remarque { border-color: gray; }	
border Propriété raccourcie (non héritée)	h2.chapitre { border: 5px solid gray; }	border-width - border-style - border-color le style de bordure est obligatoire
Remarque : L'ensemble des propriétés des bordures peuvent être définies distinctement sur chacun des quatre côtés border-top-color{ } border-right-color{ } border-bottom-color{ } border-left-color{ }		
background-color (non héritée)	p.news { background-color: yellow; }	
background-image (non héritée)	body {background-image: url(image.png);}	

background-repeat (non héritée)	body { background-repeat: repeat-y; }	repeat (horizontale et vert) repeat-x (horizontale) repeat-y (verticale) no-repeat (pas de répétition)
background-position (non héritée)	body { background-position: center top; }	left - center – right et bottom – top nombre (0% = left – top 100% = right - bottom).
background-attachment (non héritée)	.news { background-attachment: fixed; }	scroll - fixed
Background Propriété raccourcie (non héritée)	h1 { background: blue url(logo.png) center right repeat-x fixed; }	background-color, background-image, background-repeat, background-attachment, background-position (dans un ordre quelconque)
list-style-type (héritée)	ul { list-style-type: square; } ol { list-style-type: upper-roman; }	none - disc (●) - circle (○) - square (■) decimal (1, 2...) - decimal-leading-zero (01, 02...) – lower-roman (i,ii...) - upper-roman (I,II...) - lower-alpha (a, b, c...) - upper-alpha (A, B, C...)
list-style-image (héritée)	ul { list-style-image: url(puce.gif); }	Personnaliser les listes à puces avec des images
list-style-position (héritée)	ul { list-style-position: outside; } ol { list-style-position: inside; }	outside : la puce est dans la marge (valeur par défaut) inside : elle fait partie de la première ligne du paragraphe
list-style Propriété raccourcie (héritée)	li { list-style: circle inside; }	
table-layout (héritée)	table { table-layout: fixed; }	Largeur fixe ou variable des colonnes / du tableau fixed (largeur fixe) - auto (largeur automatique)
border-collapse (héritée)	table { border-collapse: collapse; }	collapse : fusion des bordures separate : séparation des bordures
border-spacing	table { border-collapse: separate; border-spacing: 2px 5px; }	1 valeur : espacement pour toutes les bordures 2 valeurs : espacement horizontal et vertical
vertical-align (héritée)	.titre { vertical-align: middle; }	définit l'alignement vertical d'un élément en ligne ou d'une cellule de tableau Baseline - top - middle – bottom

8. Les règles d'écriture

Priorité des sélecteurs

Plus une règle de style est spécifique et plus elle l'emporte sur les autres. Par contre, si deux règles ont la même spécificité la deuxième l'emporte sur la première.

Exemple : **div p** { color : red; } → **forte** = **plus spécifique**

p { color : blue; } → **faible**

```
* {...}
p {...}
div p {...}
.class {...}
p.class {...}
div p.class {...}
#id {...}
p#id {...}
div p#id {...}
.class #id {...}
.class p#id {...}
div.class p#id {...}
```

Remarque :

- Le style à l'intérieur d'une page html, non recommandé <p style="...">
- Il est conseillé d'aller de la règle la plus générale à la plus spécifique.

Style prioritaire

Utilisez **important** en cas de conflit entre deux styles ou d'incompatibilité du code avec un navigateur.

- Il permet qu'un style ne soit pas modifié par un autre : **body {background-color: white !important; }**
- L'image de fond PNG est affichée sauf pour IE6 qui ne prend pas en compte la transparence :

```
div {
    background-image : url (image.png) !important;
    background-image : url (image.gif) ;
}
```

Règles associées à plusieurs sélecteurs

div.news, p { text-align: center; } <div class="news"> <p>... </p> </div>

CSS	HTML
h1, h2, h3.contentright { margin-left: 10px; }	<h1class="contentright"> ... </h1> <h2 class="contentright"> ... </h2> <h3 class="contentright"> ... </h3>

Raccourcis

h1 { border: 1px solid red; } = border-width: 1px; border-style: solid; border-color: red;

Hiérarchisation des sélecteurs

- **Imbrication directe** **div > h1 { margin:25px 0; }**
Uniquement les balises <h1> qui sont dans le premier niveau d'imbrication à l'intérieur d'une balise <div> (enfant direct)
- **Juxtaposition** **h1 + h2 { margin-top: 10px; }**
À chaque balise <h2> qui suit une balise de fermeture </h1>. Entre ces deux balises peut se trouver du texte, mais pas une autre balise.

Les commentaires

CSS	HTML
<code>/* ... */</code>	<code><!-- ... --></code>

Unités de taille relative

Les unités de tailles relatives sont fonction de la taille du texte environnant ou du nombre de points sur l'écran. Cela permet à l'internaute d'adapter l'apparence de la taille du texte.

- **em** (largeur d'une majuscule comme M) ;
1em signifie que la police a la même taille que celle réglée par défaut dans le navigateur. 1.5em l'augmente de moitié, 0.5em la divise par deux.
- **% (100 % = 1em pour font-size);**
L'élément définit prendra un pourcentage donné de la taille de son élément parent.
- **ex** (hauteur d'une minuscule comme x, souvent arrondie à 0.5em) ;
- **px** (remarque : cette unité est relative à l'écran, mais elle ne permet pas de changer la taille.

Points	Pixels	Em	%
6pt	8px	0.5em	50%
12pt	16px	1em	100%
22pt	29px	1.8em	180%

9. Les formulaires

Les formulaires autorisent la saisie de données et permettent les échanges entre le client et le serveur, mais aussi d'exécuter un script coté client dans le cas de cookies par exemple.

Les formulaires sont déclarés par la balise `<form>` et obligatoirement terminés par `</form>`.

```
<form name="Nom" action="programme ou action" method="get|post"> ... </form>
```

Name est le nom du formulaire

Action est le nom du programme auquel les données doivent être transmises ou l'action à accomplir.

Method est la méthode de transmission des données du formulaire :

- **Get** envoi des données dans l'URL. La taille des données est limitée à 256 octets.
- **Post** envoi des données dans le corps de la requête. Pour une question de sécurité, cette méthode est préférable car elle n'affiche pas en clair les données saisies. De plus, la taille des données n'est pas limitée.
- **Exemple**

```
<form name="Enregistrement" action="traitement.php" method="post">
```

Fieldset

L'élément `<fieldset>` permet de regrouper différents champs sous une catégorie thématique plus compréhensible `<legend>`, ce qui améliore l'accessibilité du formulaire.

<code><fieldset>texte</fieldset></code>	Rassembler plusieurs champs par thème.
<code><legend>texte</legend></code>	Légende affichée dans l'élément <code><fieldset></code> .
<code><input type="text" name="Nom" id="Identifiant" value="valeur"></code>	Zone de texte. « name » est le nom du contrôle qui est envoyé avec les données du formulaire.
<code><label for="identifiant"></code> Texte : <code><input type="text" id="identifiant" ></code> <code></label></code>	La balise <code><label></code> définit une étiquette pour un élément <code><input></code> . L'attribut <code>for</code> permet de lier <label> à un élément en référence à l'attribut id de celui-ci.

```
<fieldset>
<legend> Informations personnelles </legend>
  <label for="Nom">
    Nom : <input type="text" id="Nom" name="nomUtilisateur" required>
  </label>
  <label for="Prenom">
    Prénom : <input id="Prenom" name="prenomUtilisateur" type="text" required>
  </label>
<br />
  <label for="Adresse">
    Adresse : <input type="text" id="Adresse" name="adresseUtilisateur" size=40>
  </label>
</fieldset>
```

Informations personnelles

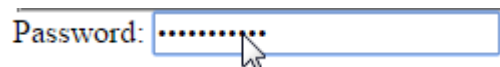
Nom : Prénom :

Adresse :

Les boutons

<code><input type="button" name="Nom" value="Légende du bouton" id="Identifiant"></code>	Simple bouton.
<code><input type="image" src="image.png" id="Identifiant"></code>	Bouton contenant une image.
<code><input type="submit" name="Nom" id="Identifiant" value=" Légende du bouton"></code>	Bouton d'envoi des données du formulaire au serveur.
<code><input type="hidden" name="Nom" id="Identifiant" value="valeur"></code>	Champ caché dont le contenu est envoyé au serveur lorsque l'utilisateur clique sur le bouton submit.
<code><input type="reset" name="Nom" id="Identifiant" value=" Légende du bouton"></code>	Bouton de réinitialisation du formulaire.

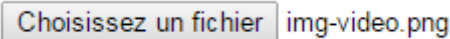


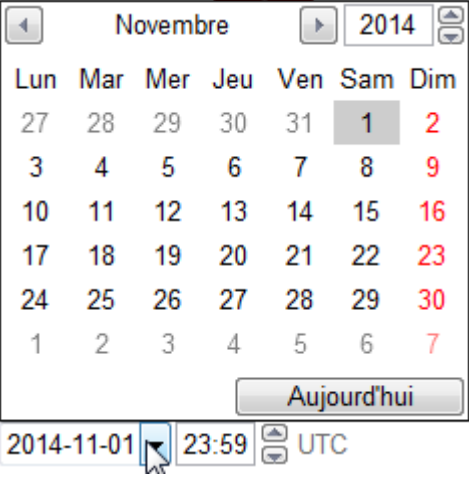
Mot de passe

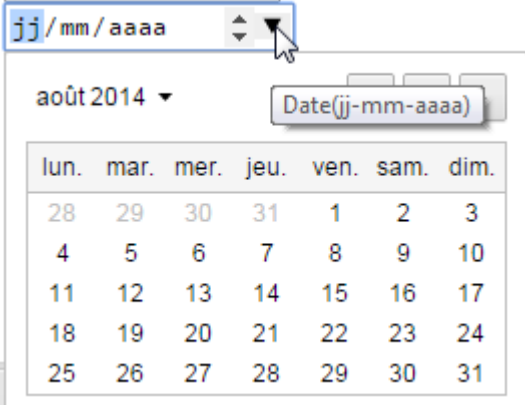
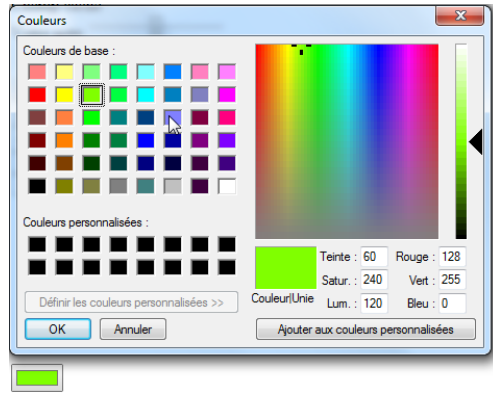
<code><input type="password" name="pwd"></code>	<p>Définit un champ de mot de passe.</p> <p>Les caractères dans un champ de mot de passe sont masqués.</p> 
---	---

Les choix

<code><input type="checkbox" name="Nom" id="Identifiant" value="valeur"></code>	Case à cocher (choix multiples).
<code><input type="radio" name="Nom" id="Identifiant" value="valeur"></code>	<p>Bouton radio (choix unique).</p> <p>Ajouter 'checked' pour présélectionner.</p>
<code><select name="Nom" size="1" id="Identifiant"></code> <code> <option value="valeur 1">Valeur affichée</option></code> <code> <option value="valeur 2" selected>Valeur affichée</option></code> <code></select></code>	<p>Liste déroulante.</p> <p>L'élément <selected> permet de présélectionner une valeur.</p>
<code><label for="choix">Quel est votre choix ?</label></code> <code><input type="text" id="choix" name="Nom" list="Identifiant" placeholder="Texte affiché" /></code> <code><datalist id="Identifiant"></code> <code> <select id="Identifiant" name="Nom"></code> <code> <option value="valeur 1">Valeur</option></code> <code> <option value="valeur 2">Valeur</option></code> <code> </select></code> <code></datalist></code>	<p>Liste d'options prédéfinies.</p> <p>L'attribut list permet de lier un bloc avec l'id d'un élément <datalist>.</p> <p>L'élément <code><select></code> permet une alternative d'affichage pour les navigateurs ne supportant pas <code><datalist></code>.</p>
<code><input type="range" value="15" max="50" min="0" step="5"></code>	Echelle de valeur

Les champs de saisie

<code><input type="email" name="Nom" id="Identifiant"></code>	Saisie de l'e-mail.
<code><input type="url" name="Nom" id="Identifiant"></code>	Saisie d'une URL.
<code><textarea cols="Nombre-colonnes" rows="Nombre-lignes" id="Identifiant">Texte</textarea></code>	Zone de saisie multi-ligne.
<code><input type="text" name="Nom" id="Identifiant" value="valeur"></code>	Zone de texte.
<code><input type="file" name="Nom" id="Identifiant"></code>	Saisie d'un fichier.  Choisissez un fichier  img-video.png
<code><input type="search" name="Nom" id="Identifiant" list="datalist" value="Recherche"></code>	Champ de recherche.
<code><input type="time" name="Nom" id="Identifiant" value="une heure 10:00" title="Heure(hh-mm)"></code>	Saisie d'une heure avec une aide au remplissage (type datepicker) présente uniquement sur certains navigateurs. 
<code><input type="datetime" name="Nom" id="Identifiant"></code>	

<pre><input type="date" name="Nom" id="Identifiant" value="Une date" title="Date(jj-mm-aaaa)"></pre>	<p>Saisie d'une date avec une aide au remplissage (type datepicker) présente uniquement sur certains navigateurs.</p> 
<pre><input type="number" name="Nom" id="Identifiant" value="Un nombre" step="5" min="0" max="50"></pre>	<p>Saisie d'un numéro.</p>
<pre><input type="color" name="Nom" id="Identifiant" value="red"></pre>	<p>Saisie d'une couleur.</p> 

Résultat

<pre><output for="Identifiant" name="nom"> 15 </output></pre>	<p>Utilisé pour stocker le résultat d'un calcul.</p>
---	--

https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Comment_structurer_un_formulaire_HTML#.C3.A9l.C3.A9ment_<output>

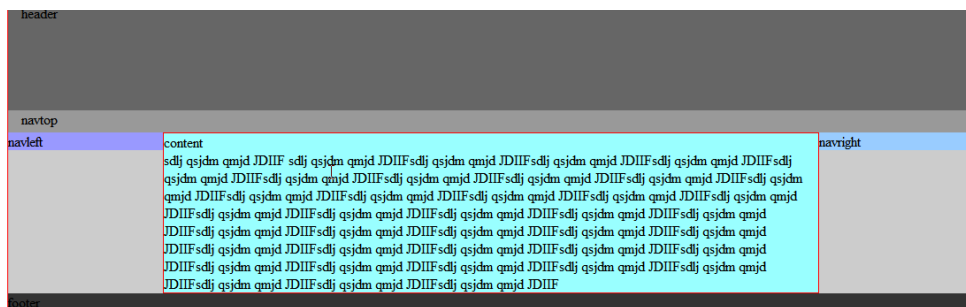
10. Squelette de la page, les balises génériques <div>

La division `<div>...</div>`

Une division est un conteneur de blocs de texte, d'images, d'autres balises div,... dont la position est définie dans la page. Il peut-être encadré, mis en forme, imbriqué, positionné dans la page.

Ces balises servent à construire l'ossature des pages.

CSS	HTML
#content{}	<div id="content ">...</div>
. content {}	<div class=" content ">...</div>



Telle une *Bentobox*, les balises `div` sont des compartiments qui divisent et structurent la page.



11. Les éléments de section

Ces nouveaux **éléments de section** sont une alternative aux blocs génériques `<div>` et ``.

`<section>`

Les éléments génériques `<section>` **regroupent du contenu autour d'une même thématique**, en segmentant des portions de la page ou de l'application web. Ils peuvent être comparés avec l'élément `<div>`, qui lui n'a aucune valeur sémantique. → Même sujet, même fonctionnalité.

Exemple : News regroupées sous un même thème.

`<header>`

Les éléments `<header>` représentent **l'en-tête d'une section d'introduction** d'un article, d'une autre section ou du document entier (en-tête de page).

`<nav>`

Les éléments de section `<nav>` contiennent des informations de la **navigation principale** (au sein du document ou vers d'autres pages).

`<aside>`

L'élément de section `<aside>` déclare une **section annexe liée au reste du contenu**, une information supplémentaire relative à un article ou une page. Le contenu de `<aside>` doit pouvoir exister par lui-même et ne pas être essentiel à l'article ou la page qui y est associée. En dehors d'un article, elle remplace généralement les barres latérales `<div id="sidebar">` et contient des informations supplémentaires relatives au contenu principal.

Utilisation dans un article : glossaire, des citations ou des liens relatifs à l'article.

Utilisation en **dehors** d'une balise `<article>` : liste de liens préférés du site, navigation additionnelle, publicité si elle est reliée à la page.

`<article>`

Les éléments de section `<article>` sont utilisés pour le **contenu indépendant**, ceux-ci pouvant être extrait individuellement du document ou syndiqués (flux RSS ou équivalent), sans pénaliser la compréhension.

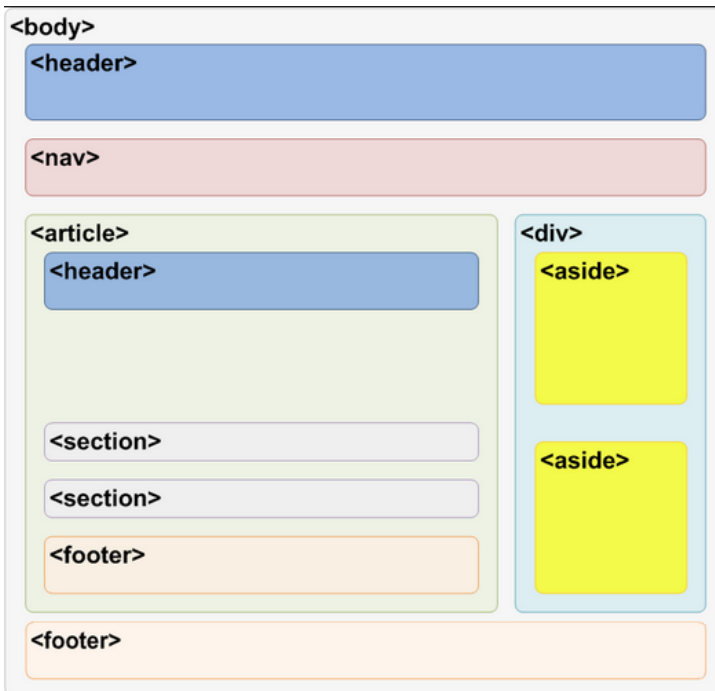
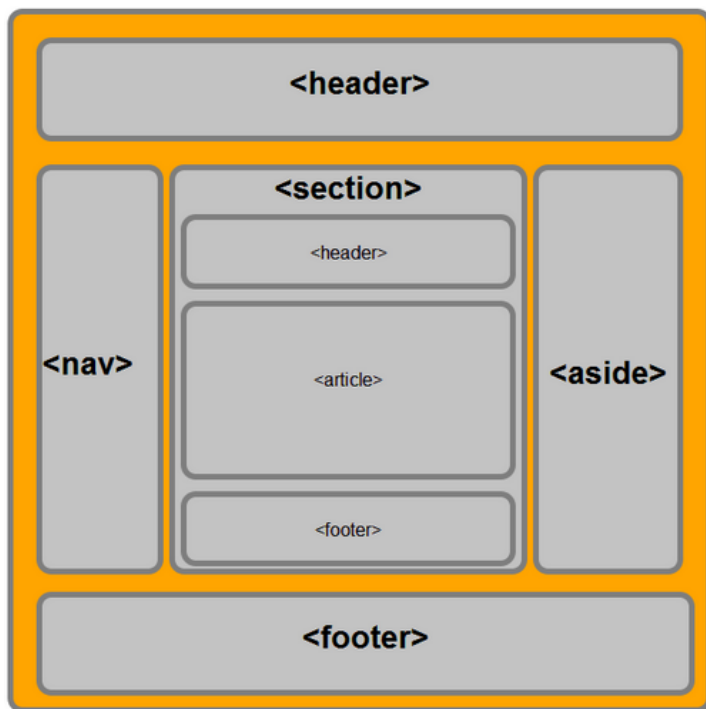
Remarque : Des éléments `<article>` dans un parent `<article>` sont censés représenter des blocs de commentaires relatifs à cet élément parent.

`<main>`

L'élément `<main>` spécifie le contenu principal d'un document. Il ne peut y avoir plus d'un élément `<main>` dans le document et le contenu de celui-ci doit être unique dans la page ou le site. Il ne peut pas être un enfant d'un élément `<article>`, `<aside>`, `<footer>`, `<header>`, `<nav>`.

`<footer>`

L'élément de section `<footer>` **représente le pied de page**, il contient des informations au sujet de l'élément qui le contient : auteur, liens vers contenus apparentés, les mentions légales, les coordonnées, les dates de publication, crédits ou copyright.



12. Nouveaux éléments sémantiques

Les microformats sont un ensemble de conventions fixées, cherchant à rationaliser et standardiser le contenu existant.

Microdonnées, <div itemscope>

L'attribut `itemscope` sert à indiquer aux robots que le contenu de la balise qui porte cet attribut aura des données sémantiques.

<http://schema.org/docs/schemas.html>

```
<div itemscope itemtype="http://schema.org/Person">
  <p>Je suis l'auteur de ce blog. Mon nom est
    <span itemprop="name"> Mary Poppins </span>
  </p>
</div>
```

`itemtype` est un attribut qui, grâce au site schema.org, permet de définir le type de contenu.

- **name** : pour renseigner le nom de la personne ;
- **address** : pour renseigner l'adresse de la personne ;
 - **streetAddress** (rue)
 - **addressLocality** (ville),
 - **addressRegion** (région)
 - **postalCode** (code postal)
- **jobTitle, telephone, email, url, colleague,...**
- **affiliation** : pour dire si la personne fait partie d'une école, ou d'une équipe ;
- **email** : l'email de la personne ;
- ...

<mark>

Surbrillance sur le résultat d'un terme recherché, pour attirer l'attention.

```
<h1>Résultat de recherche pour "Ce mot"</h1>
<ol>
  <li><a href="#">Voici le texte où apparaît <mark>Ce mot</mark></a></li>
</ol>
```

1. [Voici le texte où apparaît](#) **Ce mot**

<progress>

Permet de baliser une valeur en changement. Il est possible d'utiliser les attributs `min`, `max` et `value`.

```
<progress min="0" max="100" value="60"></progress>
```

Microformats

<hCalendar> & <time>

Marquer des **événements** pour que les utilisateurs puissent les ajouter à leur calendrier.

```
<abbr class="dtstart" title="2014-06-18" >  
18 juin 2014  
</abbr>
```

Attribut datetime ou pubdate → Lier la date à un élément article.

```
<time date="2014-06-18">18 juin 2014</time>  
<time date="16:26:00">16 heures 26</time>  
<time month="2014-06"> juin 2014</time>  
<time datetime="2014-06-18T16:26:00+1">Il est 16 h 26</time>
```

hCard

Les hCard permettent de définir sémantiquement des personnes, des sociétés,...

```
fn → nom de la personne  
adr → adresses postales  
geo → coordonnées géographiques (latitude, longitude)  
hListing → petites annonces  
hMedia → images, video et audio  
hNews → actualités  
hProduct → produits  
hRecipe → recettes de cuisine  
hReview → critiques  
hReview-aggregate → critiques et notes  
hResume → résumés ou CV  
url → lien vers le site web
```

Exemple :

```
<div id="contact" class="vcard">  
...  
<div class="adr">  
  <div class="street-address">Rue truc-much</div>  
  <div class="locality">4800</div>  
  <div class="region">Verviers</div>  
</div>  
</div>
```

Informations supplémentaires <details>

Cet élément est utilisé pour donner des éléments supplémentaires.

```
<details>
  <summary>Licence Creative Commons</summary>
  <p> Informations supplémentaires </p>
  <p> Informations supplémentaires </p>
</details>
```

► Licence Creative Commons

▼ Licence Creative Commons

Pas d'utilisation commerciale [NC] (Noncommercial) : le titulaire des utilisations non commerciales

Pas de modification [ND] (NoDerivs) : le titulaire de droits peut contraire autoriser à l'avance les modifications, traductions.

Boîte de dialogue <dialog>

Cette balise permet de créer des boîtes de dialogue contextuelles ou des composants interactifs.

L'élément `<form>` peut-être insérer dans une boîte de dialogue, il faut alors spécifier l'attribut `method="dialog"`.

```
<dialog open>
  <p>Ouvre une boîte de dialogue</p>
</dialog>
```


13. Rich media, nouveaux éléments multimédias

Un média interactif effectue un rendu dans le navigateur (sons, vidéos, photos, métadonnées), ce qui vous évite d'avoir à utiliser des plug-ins tiers tels que Flash pour diffuser les annonces.

<canvas>

L'élément `<canvas>` définit un espace, une surface de dessin bitmap, permettant de **créer des images dynamiques**, en **combinaison avec JavaScript** pour l'interaction. Il peut être modifié et redessiné à tout moment selon l'action de l'utilisateur, sans avoir recours à un plug-in.

Le contenu des balises `<canvas></canvas>` sera visible uniquement pas les navigateurs ne supportant pas cet élément.

```
<canvas id="mon-canvas" width="250px" height="150px">
```

Texte alternatif pour les navigateurs ne supportant pas Canvas.

```
</canvas>
```

```
<script type="text/javascript">
```

```
var c = document.getElementById("mon_canvas");
```

```
var ctx = c.getContext("2d");
```

```
// Le reste du script ici...
```

```
</script>
```

`getElementById()` ; cherche et cible l'élément `<canvas>` identifié par son attribut `id` unique.

`getContext()` ; définit le contexte de dessin (2D ou 3D).

`beginPath()` ; début du chemin

`moveTo(x,y)` ; un tracé part du point (x,y) / sauter jusqu'à (x,y)

lineTo(x,y) ; ajouter un segment vers (x,y)

`ctx.closePath()` ; fermeture du chemin (facultative)

`fillStyle = "red"` ; définit une couleur de fond

`stroke()` ; contour

`fill()` ; remplir le fond

`lineWidth = 3` ; épaisseur de la ligne

`ctx.lineJoin = "round"` ; style des coins round (arrondi), bevel (biseauté)

`strokeStyle = "green"` ; définit une couleur de contour

`ctx.strokeRect(gauche, haut, largeur, hauteur)` ; dessiner un rectangle

`arc(x, y, radius, startAngle, endAngle, sensAntiHoraire)` ; dessiner un arc de cercle.

➔ `ctx.arc(250,280,100,0,Math.PI*2,true)` ;

Sachant que l'on est dans une configuration trigonométrique les angles sont définis en radians avec `Math.PI` (un tour complet de cercle = `2*Math.PI`) et le sens de rotation est contraire aux aiguilles d'une montre lorsqu'il vaut `true`.

<http://diveintohtml5.info/canvas.html>

<audio>

Cette balise permet de diffuser des fichiers audio sans recourir à un plugin.

Le contenu des balises `<audio></audio>` sera visible uniquement pas les navigateurs ne supportant pas cet élément.

```
<audio src="Kalimba.mp3" controls>sons</audio>
```

L'attribut `controls` permet d'afficher des contrôles natifs pour lire, mettre en pause, ajuster le volume du son.

L'attribut `muted` permet de désactiver le son.

L'attribut `loop` permet de répéter en boucle le fichier audio.



```
<audio src="Kalimba.mp3" autoplay>sons</audio>
```

L'attribut `autoplay` lance le fichier sonore automatiquement, l'utilisateur n'a aucun moyen de l'arrêter via l'interface. A NE JAMAIS UTILISER !

```
<audio src="Kalimba.mp3" id="player">sons</audio>
```

```
<div>
```

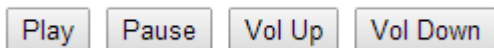
```
  <button onclick="document.getElementById('player').play()"> Play</button>
```

```
  <button onclick="document.getElementById('player').pause()"> Pause</button>
```

```
  <button onclick="document.getElementById('player').volume+=0.1">Vol Up</button>
```

```
  <button onclick="document.getElementById('player').volume-=0.1">Vol Down</button>
```

```
</div>
```



```
<audio src="Kalimba.mp3" controls poster="sintel.jpg" width="600">sons</audio>
```

<video>

Donner l'accès aux contrôles de lecture (boutons de navigation, volume) → **controls**.

`<video src="video.mp4" controls>` Description alternative `</video>`

Afficher une image représentative de la vidéo → **Poster**.

```
<video src="video.mp4" controls width="360" height="240" poster="img.jpg"> </video>
```

Proposer différents formats de la vidéo en fonction du navigateur → **source**.

```
<video width="400" height="222" controls="controls">
  <source src="video.mp4" type="video/mp4" />
  <source src="video.webm" type="video/webm" />
  <source src="video.ogv" type="video/ogg" />
  Ici l'alternative à la vidéo : un lien de téléchargement, un message,...
</video>
```

Débuter le téléchargement de la vidéo tout de suite → **preload="auto"**

(autobuffer pour Firefox 3.5 et 3.6).

Lancer la lecture automatiquement → **autoplay="true"**

<figure> et <figcaption>

`<figure>` et `<figcaption>` sont utilisées pour structurer des illustrations, des photos, des diagrammes...

`<figcaption>` désigne la légende de la figure

```
<figure>
  
  <figcaption>Première sortie des kangourous</figcaption>
</figure>
```

14. Le positionnement CSS

Le positionnement permet de placer les éléments conteneurs dans la page, les uns par rapport aux autres ou de réaliser des superpositions des éléments.

Il permet également de réaliser un habillage des images, à la manière d'un traitement de texte.

Relative

Le positionnement relatif permet d'inscrire un contenu dans le **flux normal** (c'est à dire dans son positionnement par défaut bloc ou en ligne), puis de le **décaler horizontalement ou verticalement**. Le **contenu suivant n'est pas affecté** par ce déplacement, ce qui peut donc entraîner des **chevauchements** comme sur l'exemple présenté.

```
#content{position:relative; top:10%; right:5%; } → auto, em, ex, px, %
```

Absolute

Le positionnement absolu **retire le contenu du flux normal** (c'est à dire de son positionnement par défaut bloc ou en ligne) : sa position est déterminée par référence aux limites de l'objet qui le contient. Celui-ci peut-être : une autre boîte elle-même positionnée (position relative ou absolue) par défaut le bloc conteneur initial c'est-à-dire la fenêtre du navigateur.

L'élément suivant n'est pas affecté par ce positionnement

```
article{position:absolute; top:5em; right:2em; } → auto, em, ex, px, %
```

Float

Le positionnement flottant **retire une boîte du flux normal** (c'est à dire de son positionnement par défaut bloc ou en ligne) pour la placer le plus à droite possible (float: right) ou le plus à gauche possible (float: left) possible à l'intérieur de son conteneur. L'élément suivant **est affecté** par ce positionnement

```
aside{float:right; top:5em; right:2em; } → auto, em, ex, px, %
```

Clear

Aucun élément flottant ne se place ni à gauche ni à droite.

```
div #footer{ clear:both; } → clear:left; clear:right;
```

La propriété clear:both permet d'éviter :

- que le bloc flottant sorte du flux du conteneur. Une autre solution est de donner une hauteur au conteneur.
- que le contenu du conteneur habille l'élément flottant (passer à la ligne)

Display

<code>span.bloc{display:block;}</code>	display:block permet de transformer un élément en ligne en élément de type bloc.
<code>p.cache{display:none;}</code>	display:none permet de rendre invisible l'élément et sa place n'est plus réservée sur la page (aucune boîte n'est générée).
<code>li{display:inline;}</code>	display:inline permet de transformer un élément de type bloc en élément de type ligne.

Superposition Z-index

<code>#logo{position:absolute;top:0;z-index:-5;}</code>	Modifie l'ordre de superposition des blocs.
---	---

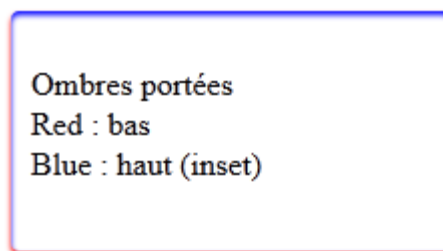
La propriété z-index s'applique uniquement aux blocs positionnés. Si ce n'est pas le cas, appliquez une position relative sans décalage. (Remarque : certains navigateur acceptent d'appliquer z-index a un élément non positionné).

15. Trucs et astuces CSS3

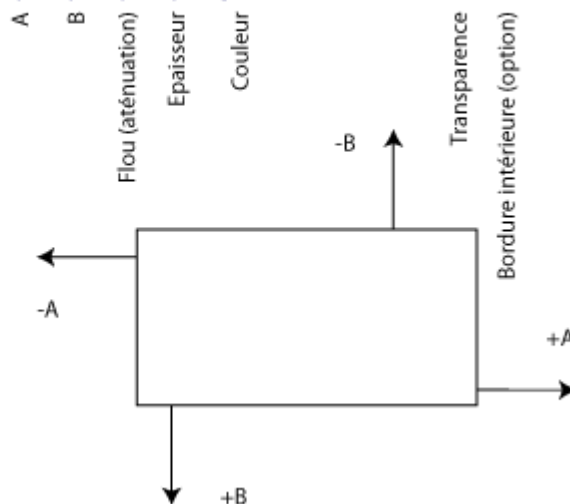
Ombre portée (non héritée)

-o-box-shadow:0px 0px 10px #1B1464 ;
 -ms-box-shadow:0px 0px 10px #1B1464 ;
 -moz-box-shadow:0px 0px 10px #1B1464 ; /*ombre firefox*/
 -webkit-box-shadow:0px 0px 10px #1B1464 ; /*ombre portée pour Chrome*/

box-shadow: 0px 0px 10px #1B1464 ; /*ombre portée pour mozilla, Opera, Chrome*/



#1B1464
 box-shadow:0px 2px 2px 0px rgba(255, 255, 255, 0.5) inset;



Dégradé (non héritée)

background-image:-webkit-linear-gradient(top, #ccc, #666);/*Chrome 10 et suivants + Safari 5.1 et suivants*/

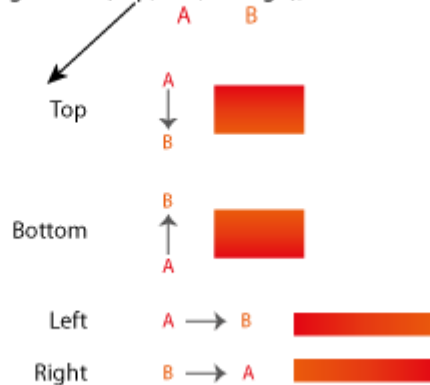
background-image:-moz-linear-gradient(top, #ccc, #666);/*Firefox 3.6 et suivants*/

background-image:-ms-linear-gradient(top, #ccc, #666);/*IE 10 et suivants*/

background-image:-o-linear-gradient(top, #ccc, #666);/*Opera 11.10 et suivants*/

background-image:linear-gradient(top, #ccc, #666);

background-image:linear-gradient(top, red, orange);



Bords arrondis (non héritée)

-webkit-border-radius:4px;

-moz-border-radius:4px;

-ms-border-radius:4px;

-o-border-radius:4px;

border-radius:4px;

border-radius:4px 0px 2px 2px 0px;

Transition (non héritée)

Cette propriété permet d'ajouter une transition à l'effet précédent.

-webkit-transition: all .5s cubic-bezier(.6, 2, .4, 1);

oz-transition: all .5s cubic-bezier(.6, 2, .4, 1);

-ms-transition: all .5s cubic-bezier(.6, 2, .4, 1);

-o-transition: all .5s cubic-bezier(.6, 2, .4, 1);

transition: all .5s cubic-bezier(.6, 2, .4, 1);

Courbe d'accélération à la souris :

<http://cubic-bezier.com/#.17,.67,1,.17>

Rotation (non héritée)

-moz-transform:rotate(15deg);

-webkit-transform:rotate(15deg);

-ms-transform:rotate(15deg);

-o-transform:rotate(15deg);

transform:rotate(15deg) ;



Les filtres

Cette propriété modifie la couleur, la netteté,... des images.

<http://www.inserthtml.com/2012/06/css-filters/>

Exemple :

```
-webkit-filter: grayscale(1) blur(5px);
-moz-filter: grayscale(1) blur(5px);
-ms-filter: grayscale(1) blur(5px);
-o-filter: grayscale(1) blur(5px);
```

```
filter: grayscale(1) blur(5px);
```

filter: blur(5px);	Flou
filter: brightness(0.2);	Luminosité
filter: saturate(50%);	Saturation
filter: hue-rotate(20deg);	Modifier la teinte
filter: contrast(150%);	Contraste
filter: invert(100%);	Négatif
filter: grayscale(100%);	Niveaux de gris
filter: sepia(100%);	Sépia

Colonnes (non héritée)

```
-moz-column-count:2;
-webkit-column-count:2;
-ms-column-count:2;
-o-column-count:2;
```

```
p{column-count:2; text-align:justify}
```

Culture citoyenne

Le portail Culture.be vise une plus grande interactivité avec les citoyens et le monde de la culture avec une information actualisée au quotidien (agenda, manifestations, offres d'emploi sectorielles, aides et financements) en présentant également une cartographie des organismes, salles et espaces culturels.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nulla lacus tortor, dictum nec dapibus non, fermentum eu felis. Praesent consequat est leo. Quisque convallis ullamcorper erat. Vivamus lacinia lobortis massa nec rhoncus. Nulla justo purus, mattis sit amet faucibus sit amet, luctus sed purus. Nunc porttitor accumsan mauris. Donec ac diam turpis. Nullam et tellus eros. Phasellus sit amet massa lorem. Curabitur ultrices, nisi non sodales feugiat, urna sapien pharetra libero, et feugiat ligula sem quis ipsum. Nam viverra commodo bibendum. Nullam pulvinar elit ornare urna mattis ullamcorper. Etiam commodo viverra neque sed aliquam. Quisque pellentesque nisi sit amet lorem egestas euismod. Aenean metus eros, lobortis vel blandit ut, lacinia at lorem. Nam arcu nulla, commodo in posuere at, vehicula at risus.

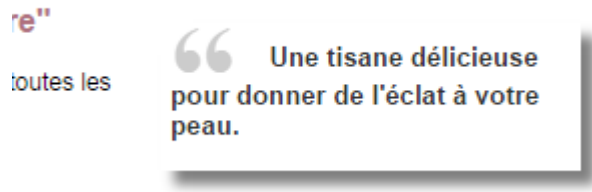
Insertion de caractères

(non héritée)

Dans cet exemple, vous découvrirez comment insérer automatiquement des guillemets dans la balise `<blockquote>`.

```
blockquote:before{content:open-quote;}
```

```
blockquote:after{content:close-quote;}
```



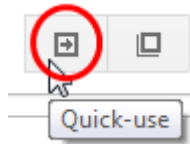
```
blockquote:before{content:open-quote;
quotes:"\201C""\201D";
font-family:"Times New Roman", Times, serif;
font-size:500%;
color:#CCC;
line-height: 0.1em;
margin-right: 0.25em;
vertical-align: -0.4em;}
```

La propriété `quotes` spécifie les guillemets à utiliser pour le contenu généré → **quotes:** `"\201C""\201D""\2018""\2019";`

Personnaliser la police avec Google Font

Google Web Fonts propose une collection de polices de caractères facilement intégrables à un site web.

- <http://www.google.com/fonts/>
- Sélectionnez la police que vous souhaitez utiliser.
- ATTENTION que chaque police utilisée pour votre site alourdira le chargement de vos pages.
- Cliquez sur le bouton 'Quick-use' :

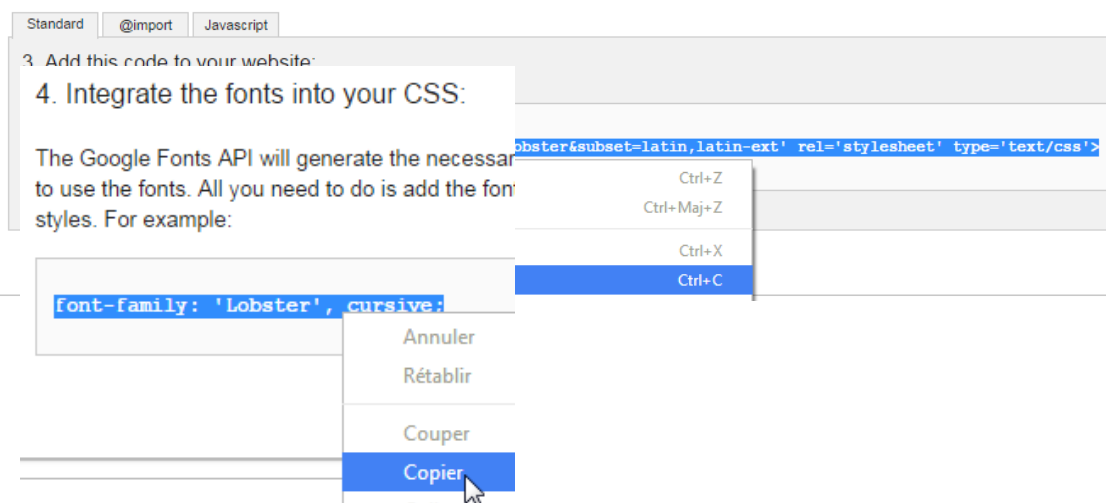


- Choisissez le type de caractères utilisés :

2. Choose the character sets you want:

☒ Latin (latin) ☒ Latin Extended (latin-ext) ☐ Cyrillic (cyrillic)

- Copiez le code html d'intégration de la police **entre** les balises <head> </head> :



- Copiez le code CSS et intégrez-le à votre feuille de style :