

# COURS HTML-CSS

---

```
<html>
  <link rel="stylesheet" type="text/css"
  <font color="#00CC00 size="20">
  h1 { <head> p {
    font size=12px; color: gray;
  } <body> }
  CSS background-image:
  absolu signifie que l'objet
  </keyring> </font> <br>
  <p> 10px; <3
  </head> <br> </body>
  <div id="Image"> <br> </div>
  </font> <font color="FF9900">
  <hr> </hr>
</html>
```

# Table des matières

<b>1</b>	<b>Utilisation du logiciel Quanta+</b>	<b>2</b>
<b>2</b>	<b>Les feuilles de style CSS</b>	<b>4</b>
2.1	Définition et principe . . . . .	4
2.2	Mais pourquoi donc faire du CSS? . . . . .	5
2.3	Comment ça marche? . . . . .	5
<b>3</b>	<b>Pour commencer en CSS</b>	<b>5</b>
3.1	Couleurs et arrières-plans . . . . .	6
3.2	Exercice . . . . .	7
3.3	Les polices . . . . .	7
3.4	Exercice . . . . .	9
3.5	Le texte . . . . .	9
3.6	Exercice . . . . .	10
3.7	Les liens . . . . .	10
3.8	Exercice . . . . .	11
<b>4</b>	<b>Petite parenthèse sur la présentation d'une page web</b>	<b>12</b>
<b>5</b>	<b>Pour aller plus loin</b>	<b>13</b>
5.1	Identification et regroupement d'éléments (class et id) . . . . .	13
5.1.1	Regroupement en classes (class) . . . . .	13
5.1.2	L'identification d'un élément par son identifiant (id) . . . . .	14
5.1.3	Deux balises qui ne font rien... . . . . .	15
5.2	Exercice . . . . .	17
5.3	C'est dans la boîte! . . . . .	18
5.3.1	gérer les marges d'un élément . . . . .	18
5.3.2	Gérer les espacements d'un élément . . . . .	18
5.3.3	Les bordures . . . . .	19
5.3.4	La hauteur et la largeur . . . . .	19
5.3.5	Le positionnement sur la page . . . . .	21
5.4	Exercice . . . . .	21
5.5	Les éléments flottants . . . . .	22
5.6	Exercice . . . . .	22
<b>6</b>	<b>Et pour la suite?</b>	<b>23</b>

# 1 Utilisation du logiciel Quanta+

Durant cette séquence HTML-CSS, nous allons pour nous faciliter la vie, utiliser un logiciel de développement web, *Quanta+* (figure 1). Nous avons vu dans la partie HTML la structure générale d'un fichier HTML, ainsi que les principales commandes permettant de mettre en forme une page. L'utilisation d'une plate-forme telle que *MolésosOS* permet de gagner beaucoup de temps en intégrant des boutons cliquables et une prévisualisation du résultat. Le logiciel *Quanta+* a la même vocation : il intègre une foule de raccourcis pour la création de la page et intègre des outils de prévisualisation dans les principaux navigateurs web.

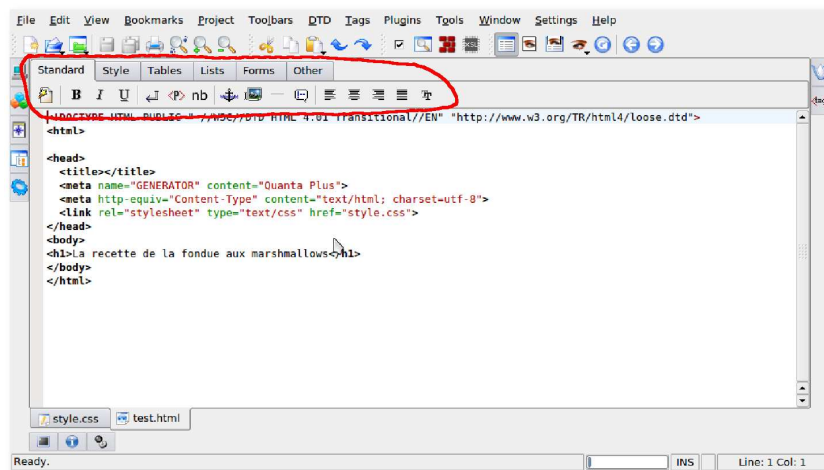



FIGURE 1 – L'interface de Quanta+

Notez, encadré en rouge, toute une série d'onglets permettant de se simplifier la vie : en un clic, on obtient automatiquement la plupart des balises dont nous avons besoin en HTML !

Étapes pour la création d'un projet :

1. Dans votre dossier personnel, créez un dossier « projetHTML » dans lequel vous placerez les fichiers \*.html et \*.css (\* sera remplacé par les noms de vos fichiers, autant que possible des noms simples et significatifs, sans espaces, par exemple projet1.html et style.css).
2. Dans ce dossier, créez un sous-dossier « Images » qui contiendra toutes les images, photos...etc
3. Ouvrez Quanta+ (Applications/Programmation/Quanta plus)
4. Cliquez sur le bouton 
5. Une fenêtre s'ouvre (figure 2), et la seule chose que vous ayez à faire, c'est de rajouter le lien vers la feuille de style (entouré en rouge).
6. L'environnement général pour le fichier HTML est créé, avec un lien vers la feuille de style associée (sur l'exemple, style.css). A présent, sauvegardez le fichier dans le dossier créé au premier point.
7. Créez un nouveau fichier sans aucun cadre général (Fichier/Nouveau) et sauvegardez-le toujours dans le même dossier, en le nommant style.css
8. Vous êtes prêt à travailler. Vous pourrez réaliser les exercices en suivant ces étapes.

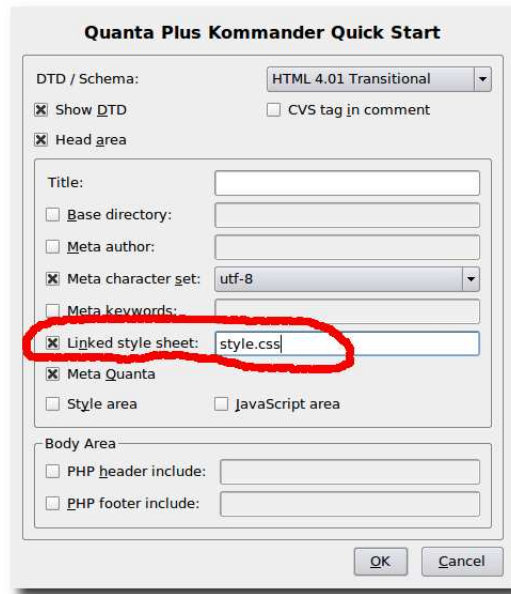


FIGURE 2 – La fenêtre Quick-start de Quanta

## 2 Les feuilles de style CSS

### 2.1 Définition et principe

Jusqu'ici, nous avons vu que HTML sert à structurer un contenu, par exemple titre, insertion d'images, liens internet. CSS est l'abréviation de Cascading Style Sheet, qui pourrait se traduire par Feuilles de Style en Cascade.

La différence entre HTML et CSS par un petit exemple : si on se rend sur le site [RPN](http://www.rpn.ch), on devrait voir ceci (figure 3) :



FIGURE 3 – Le site RPN avec CSS

Maintenant, sans le CSS, voici ce qu'on obtiendrait (figure 4) (le jeu des 10 différences) :



FIGURE 4 – Le site RPN... sans CSS !

En résumé, HTML sert à structurer le contenu, CSS sert à formater le contenu structuré.

## 2.2 Mais pourquoi donc faire du CSS ?

Bonne question ! Pour y répondre, ouvrez votre navigateur et rendez-vous sur votre site favori, puis demandez-vous pourquoi toutes les pages ont le même aspect ? Simplement parce-que toutes ces pages sont liées à une même feuille de style... Les avantages sont évidents :

- le contrôle de la présentation de plusieurs documents par une seule feuille de style.
- un contrôle plus précis de la présentation.
- des présentations différentes en fonction du média utilisé (typiquement, une feuille de style spéciale selon la consultation via un smartphone ou un écran d'ordinateur).

## 2.3 Comment ça marche ?

il existe plusieurs façons de faire du CSS :

- en *interne* (pas biennnnnn !), c'est à dire dans la balise HTML.

Un exemple : pour donner un magnifique fond rouge à notre page, nous pouvons inclure un attribut dans la balise body `<body style="background-color:red;">` . Le fond de la page sera rouge... admettons maintenant que le site possède trente pages : il faudra copier trente fois la balise. Puis si on veut changer la couleur, et biennnnn, il faut la changer sur les trente pages !

On comprend aisément que ce n'est pas une bonne solution, et de toute façon, *ce n'est pas la philosophie du CSS!*

- en feuille de style *externe* (biennnnnn !). Il faudra alors créer une feuille de style, en la nommant par exemple "style.css", puis placer dans le head la balise

`<link rel="stylesheet" type="text/css" href="style.css">` et dans ce cas, on place la feuille de style dans le même dossier que le document html. Pour notre exemple, la feuille de style contiendrait les lignes :

```
body {  
background-color: red;  
}
```

Avantage de cette méthode : une seule feuille de style pour autant de pages HTML que l'on veut !

## 3 Pour commencer en CSS

Nous allons donner ici une liste des principaux attributs, mais il faut savoir que le CSS offre énormément de possibilités ! Pour avoir une vision plus complète il vaut mieux se référer à des tutoriels sur internet, comme par exemple ceux du site [HTML.net](http://HTML.net). Il faut aussi être raisonnables, parce-que devant la profusion de possibilités, il y a un risque d'en faire trop (voir chapitre 4)...



FIGURE 5 – Du style...

### 3.1 Couleurs et arrières-plans

- **color** : détermine la couleur d'avant plan. Exemple : si on veut que les titres de niveau 1 soient rouges, on insère dans la feuille de style :

```
h1 {
color: red;
}
```

En général, les noms des couleurs sont en anglais, pour les couleurs de base. Pour une liste de couleurs standard du web plus fournie : [voir ici](#). Il suffit de copier le code en hexadécimal précédé d'un # , exemple #b0c4de.

- **background-color** : définit la couleur d'arrière-plan d'un élément. Exemple : si on veut que le corps de la page soit rouge (!), on insère dans la feuille de style :

```
body {
background-color: red;
}
```

- **background-image** : insère une image d'arrière-plan. Exemple : si on veut que les paragraphes aient une image image.jpg en fond, on insère dans la feuille de style :

```
p {
background-image: url("image.jpg");
}
```

Seuls les formats suivants sont gérés en HTML :

- \*.jpg, image compressée, idéal de par la taille des fichiers générés (chargement rapide de la page)
- \*.png, plus volumineuses, mais permettant plus d'options, par exemple la transparence.
- \*.gif, mêmes caractéristiques que le png, mais permettant par exemple les images animées.

**Attention ! concernant les chemins vers des éléments externes, il faut être très rigoureux : dans l'exemple précédent, l'url vers l'image indique que celle-ci est localisée dans le même dossier que le fichier html. Si, par exemple on crée un sous-dossier Images dans le dossier ou se trouve le fichier html, le chemin devient : Images/image.jpg**

- **background-repeat** : répète l'image d'arrière-plan. Plusieurs possibilités :
  - repeat-x : l'image se répète horizontalement.
  - repeat-y : l'image se répète verticalement.
  - repeat : l'image se répète horizontalement et verticalement (par défaut).
  - no-repeat : l'image ne se répète pas.

Exemple : si on veut que les paragraphes aient une image image.jpg en fond qui se répète horizontalement et verticalement, on insère dans la feuille de style :

```
p {
background-image: url("image.jpg");
background-repeat: repeat;
}
```

- **background-attachement** : définit si l'image est fixe ou défile avec la page.
  - scroll : l'image défile avec la page (non bloquée).
  - fixed : l'image est bloquée (utile pour des bannières de haut de page, par exemple).Exemple : si on veut que le body possède une image d'arrière-plan fixe, on insère dans la feuille de style :

```
body {  
background-image: url("image.jpg");  
background-attachement : fixed;  
}
```

- **background-position** : par défaut, l'image d'arrière-plan se positionnera dans le coin supérieur gauche de l'écran. Cette propriété permet de changer cette position et de placer l'image n'importe où sur l'écran.

Exemple : si on veut que le body possède une image d'arrière-plan fixe positionnée à 5cm de la gauche et 5cm du haut de la page, on insère dans la feuille de style :

```
body {  
background-image: url("image.jpg");  
background-position: 5cm 5cm;  
}
```

Dans tous ces cas ainsi que les suivants on peut bien sûr utiliser plusieurs attributs à la fois !

## 3.2 Exercice

A l'aide de Quanta+(1), créez un fichier nommé « exercice.html » dans laquelle vous insérerez via une feuille de style « style.css » l'image « lac.png » qui se trouve [ici](#). Comme style pour le body, nous voulons que :

- l'image soit dans une position fixe, sans répétition, à une position 50% de la droite et 0% du haut de la page
- le body soit de la couleur #b0c4de (nuance de bleu-gris).

## 3.3 Les polices

Attention : quand vous utilisez une police dans une page web, elle ne s'affichera que si l'internaute a la même police installée sur son ordinateur.

- Vous n'avez aucun moyen de forcer l'internaute à télécharger et installer une police, pas même automatiquement.
- Vous n'avez non plus aucune garantie sur les polices présentes sur l'ordinateur de l'internaute.
- Vous n'avez donc aucune garantie sur la manière dont votre page sera rendue dans les navigateurs.

Il faut donc faire « au mieux », c'est à dire spécifier des polices (en espérant qu'elles seront présentes), et spécifier d'autres noms de polices alternative (si possible proches de celle que vous souhaiteriez).



- **font-family** : sert à indiquer la liste des polices à utiliser pour l’affichage d’un élément donné ou de toute une page Web. Il n’est bien entendu pas obligatoire de changer la police, encore moins de mélanger 5 types de polices sous prétexte qu’on peu le faire...  
Exemple : si on veut que les titres de niveau 1 soient dans la police arial, et si le navigateur ne peut l’afficher, il essaye verdana, sans serif<sup>1</sup> , on insère dans la feuille de style :

```
h1 {  
font-family: arial, verdana, sans-serif;  
}
```

- **font-style** : définit le style de la police utilisée, à savoir normal ou *italique* (italic). Si on veut que dans l’exemple précédent, les titres de niveau 1 soient italiques, on insère dans la feuille de style :

```
h1 {  
font-family: arial, verdana, sans-serif; font-style: italic;  
}
```

- **font-variant** : indique la variante de la police, à savoir **normal** (comme son nom l’indique...) ou **small-caps** (majuscules dont la taille est réduite à la tailles des minuscules). Toujours dans le même exemple, on veut les titres en petites majuscules, donc on insère dans la feuille de style :

```
h1 {  
font-family: arial, verdana, sans-serif; font-variant: small-caps;  
}
```

- **font-size** : sert à indiquer la taille de la police, avec 3 écritures possibles. Pour des titres h1 en taille 30px, des h2 de 12 points et h3 de 120% de la normale , on insère dans la feuille de style :

```
h1 {  
font-size: 30px;  
}  
h2 {  
font-size: 12pt;  
}  
h3 {  
font-size: 120%;  
}
```

- **font-weight** : décrit avec quel degré de gras présenter la police. Elle peut être normale (normal) ou **grasse** (bold). Certains navigateurs permettent même d’utiliser des nombres entre 100 et 900 (les centaines) pour décrire la graisse de la police.  
Exemple : un paragraphe avec la police en gras,

```
p {  
font-weight: bold;  
}
```

---

1. serif : voir la définition [ici](#)

### 3.4 Exercice

Poursuivons l'exercice précédent, en ajoutant comme titre de niveau 1 "Le lac de Neuchâtel" et comme texte (à copier-coller et placer dans une balise `<p></p>`) :

*Neuchâtel est une commune de Suisse, chef-lieu du canton de Neuchâtel et du district de Neuchâtel. Située au bord du lac de Neuchâtel, sur le flanc sud du massif du Jura, elle fait face à la chaîne des Alpes. Neuchâtel est citée pour la première fois en 1011. Elle est gérée par des comtes de Neuchâtel jusqu'en 1458, puis passe sous l'autorité de la famille d'Orléans-Longueville jusqu'en 1707. La ville est ensuite sous l'autorité des rois de Prusse jusqu'en 1848, tout en ayant dès 1814 adhéré à la Suisse en formant le canton de Neuchâtel. L'économie de la ville a longtemps été liée à l'administration (ville la plus importante de la région) et à l'agriculture (vigne). Elle a été dominée par le commerce international au XVIIIe siècle, puis par les secteurs secondaire (horlogerie) et tertiaire. Au tournant des XIXe et XXe siècles plusieurs hôtels sont construits pour soutenir le tourisme qui accompagne le développement du chemin de fer. La région est fortement industrielle avec la chocolaterie Suchard, les forges et scieries Martenet, l'horlogerie et la fabrication d'indiennes qui se développent durant le XIXe siècle et s'effondrent à la fin du XXe siècle. Elle se renouvelle aujourd'hui grâce aux technologies de pointe. La ville est également connue pour son université, son club de football, ses activités de hautes technologies et, sur le plan touristique, par son bourg médiéval et sa situation. Elle a en outre été l'une des quatre villes organisatrices d'Expo.02.*

Vous constaterez que le résultat n'est pas très joli... en effet, il vous faut insérer des sauts de lignes avant le titre pour qu'il n'empiète pas sur l'image ainsi que dans le texte pour l'aérer. Dans un prochain chapitre, nous verrons une autre manière de procéder, en plaçant les éléments sur la page.

Ensuite, nous appliquerons les éléments de style suivants :

- titre de niveau 1 , sera de la famille arial de taille 60 pixels et italique.
- le texte du paragraphe sera de la famille « Times New Roman » (il faut mettre des guillemets à cause des espaces dans le nom) et de taille 120%.

### 3.5 Le texte

- **text-indent** : permet d'ajouter une indentation<sup>2</sup> en début de paragraphe. Exemple :

```
p {
  text-indent: 30px;
}
```

- **text-align** : Comme son nom l'indique, gère l'alignement du texte, gauche (**left**), droite (**right**) ou centré (**center**). De plus, on peut justifier (**justify**), c'est à dire que la ligne sera étirée de sorte que les marges de gauche et de droite soient identiques.

Exemple :

```
p {
  text-align: center;
}
```

Il existe d'autres options, pas vraiment utiles ou esthétiques que nous ne mentionnerons pas ici.

---

2. indentation : petit décalage de la première ligne du paragraphe

### 3.6 Exercice

Toujours sur notre exercice précédent :

- indentez le texte du paragraphe de 20 pixels et justifiez-le.
- centrez le texte du titre de niveau 1.

### 3.7 Les liens

On peut appliquer aux liens tout ce qu'on sait au sujet des polices et du texte. la nouveauté, c'est que le CSS permet des effets sympathiques. La norme sur le net veut qu'un lien soit bleu et souligné s'il n'a jamais été visité, et rouge souligné s'il a déjà été visité. Pour bien comprendre, nous devons définir la notion de *pseudo classe* qui permet de tenir compte de certains événements.

- Pseudo-classes liens visités ou non (**link** et **visited**) : nous savons que la balise <a> définit des liens internet. Nous voulons donc que les liens non visités bleus, visités en rouges.

```
a:link {
color: blue;
}
a:visited {
color: red;
}
```

Résultat :

liens non visités      [Voici un lien vers HTML.net](#)  
liens visités          [Voici un lien vers HTML.net](#)

- Pseudo-classes d'affichage : **active** permet d'agir sur des liens activés.  
Exemple : les liens auront un arrière-plan jaune s'ils sont activés (cliqués).

```
a:active {
background-color: #FFFF00;
}
```

- Pseudo classe d'effets au survol par le curseur... très joli pour donner un petit côté dynamique aux liens (**hover**). Exemple : au survol, les liens se colorent en orange, deviennent gras et les lettres s'espacent de 5px.

```
a:hover{
color: orange;
font-weight: bold;
letter-spacing: 5px;
}
a:visited {
color: red;
}
```

A doser évidemment, il ne faut pas exagérer...

## Comment supprimer les traits sous les liens ?

Une question souvent posée... par défaut, les liens sont soulignés, mais si vraiment on désire procéder autrement, voici une solution :

```
a {  
text-decoration :none;  
}
```

Attention à bien évaluer si vous voulez *vraiment* enlever le soulignement, sachant que c'est un standard du web et que les gens sont en général habitués...

Un exemple connu, si on se rend sur le site de [wikipedia](#), les liens non visités sont bleus et non soulignés, mais au survol de la souris, ils se soulignent.

### 3.8 Exercice

A présent, pour poursuivre l'exercice :

- insérez un texte « Source : Wikipedia » dans une balise faisant le lien vers le site de wikipedia (sous le texte, dans le fichier HTML)
- donnez le style liens non visités en bleu, sans soulignement et visités en #b22222
- au survol du lien par la souris, les lettres s'espacent de 2 pixels.

## 4 Petite parenthèse sur la présentation d'une page web

A ce stade, nous savons faire beaucoup de choses permettant des présentations de pages complexes.

Au chapitre suivant, nous aborderons des notions importantes qui permettent encore plus de puissance dans la création, mais attention !

Ce n'est pas parce-que nous savons faire plein de choses qu'il faut les faire !

Avant la création d'un site ou même d'une page seule, il faut peut-être se poser devant une feuille de papier, prendre un crayon et esquisser le plan de cette page.

En effet, trop de sites amateurs offrent une présentation de mauvais goût, peu pratiques et d'un degré de navigation douteux (figure 6) !

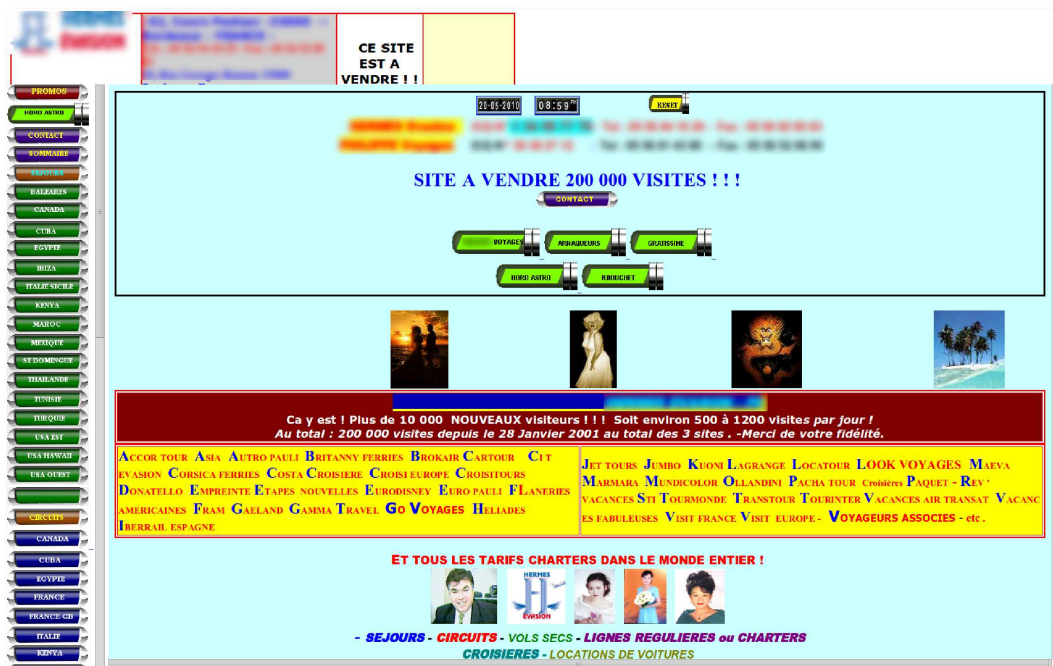


FIGURE 6 – Oh le beau site...

## 5 Pour aller plus loin

### 5.1 Identification et regroupement d'éléments (class et id)

Attention, partie un peu technique, mais très puissante !

Il arrive que l'on veuille appliquer un style spécial à un seul élément de la page, par exemple le menu, ou que l'on désire appliquer un même style à plusieurs éléments différents.

#### 5.1.1 Regroupement en classes (class)

Prenons un exemple : dans un groupe d'élèves, on a 3 filles et 3 garçons. On désire donner une liste avec les noms des filles en... rose et les noms des garçons en... bleu (oh, les clichés ! ). Pour le résultat, voir figure 7.

Voici le code sans style :

```
<ul>
<li>Alain</li>
<li>Julie</li>
<li>Luc</li>
<li>Véronique</li>
<li>Estelle</li>
<li>Marc</li>
</ul>
```

Maintenant, on attribue une classe « boy » aux garçons et une classe « girl » aux filles. C'est comme si on collait une étiquette à chacun en fonction de son sexe. Pour cela, on va ajouter les classes aux balises contenant les prénoms :

```
<ul>
<li class="boy">Alain</li>
<li class="girl">Julie</li>
<li class="boy">Luc</li>
<li class="girl">Véronique</li>
<li class="girl">Estelle</li>
<li class="boy">Marc</li>
</ul>
```

Puis on ajoute dans la feuille de style associée au fichier HTML :

```
li.boy {
color: pink;
}
li.girl {
color: blue;
}
```

- Alain
- Julie
- Luc
- Véronique
- Estelle
- Marc

FIGURE 7 – Résultat

Voilà, vous savez faire du regroupement ! Notez que les exemples peuvent paraître tirés par les cheveux, ils ne sont là que pour apprendre la notion. Ce n'est qu'avec du recul que vous comprendrez l'importance de chaque élément acquis et saurez les utiliser à bon escient.

### 5.1.2 L'identification d'un élément par son identifiant (id)

Encore une illustration : vous ! Votre patrimoine génétique est votre css propre et fait que votre style est unique, un test adn donnerait en quelque sorte votre id...

En CSS, les utilisations ne sont pas très parlantes, mais sur une page web, c'est une notion capitale, qui permet de récupérer le contenu d'une balise pour l'utiliser ultérieurement, particulièrement en Javascript<sup>3</sup>.

Exemple sur une page web : un tableau contient des noms de fruits, avec un légume. On désire le singulariser en donnant à sa police la taille 30 pixels (voir figure 8) :

```
<table border="1px">
  <tr>
    <td>pomme</td>
    <td>poire</td>
  </tr>
  <tr>
    <td>raisin</td>
    <td>haricot</td>
  </tr>
</table>
```

On peut par exemple attribuer un identifiant à sa case comme ceci :

```
<table border="1px">
  <tr>
    <td>pomme</td>
    <td>poire</td>
  </tr>
  <tr>
    <td>raisin</td>
    <td id="legume">haricot</td>
  </tr>
</table>
```

---

3. Javascript est un langage de programmation permettant par exemple de créer une page web interactive

Lié à une feuille de style contenant le code (le # signifie que l'on utilise l'id) :

```
#legume {
font-size: 30px;
}
```

pomme	poire
raisin	haricot

FIGURE 8 – Résultat

### 5.1.3 Deux balises qui ne font rien...

...mais qui sont indispensables!!

Dans le code HTML, on rencontre ces deux balises : `<div>` et `<span>` qui ne font rien en soi, mais qui liées aux notions de class et d'id offrent d'énormes possibilités! On pourrait donner comme image celle-ci : ce sont des conteneurs dans lesquels on place ce qu'on veut... et dont on fait ce que l'on veut.



FIGURE 9 – Une div...

**La Balise span :** Si dans ce paragraphe :

```
<p>Il se faut entr'aider, c'est la loi de Nature.<br>
L'âne un jour pourtant s'en moqua :<br>
Et ne sais comme il y manqua;<br>
Car il est bonne créature</p>
```

on désire mettre dans un style particulier certains mots (figure 10), on peut les regrouper par la balise span :

```
<p>Il se faut <span class="evidence">entr'aider</span>,
c'est la <span class="evidence">loi</span> de Nature.<br>
<span class="evidence">L'âne</span> un jour
pourtant <span class="evidence">s'en moqua</span> :<br>
Et ne sais comme il y manqua;<br>
Car il est bonne <span class="evidence">créature</span></p>
```

L'utilisation de la classe « evidence » permet de joindre à la feuille de style une mise en évidence des morceaux de texte. (ici, couleur rouge et taille des caractères 120%)

```
span.evidence {
color: red;
font-size: 120%;
}
```

Il se faut **entr'aider**, c'est la **loi** de Nature.  
**L'âne** un jour pourtant **s'en moqua** :  
Et ne sais comme il y manqua;  
Car il est bonne **créature**

FIGURE 10 – Résultat



## La balise div :

Nous avons vu que la balise span sert à regrouper des éléments d'une même classe, mais on pourrait vouloir attribuer un style à un élément unique. C'est la seule différence entre ces deux balises qui, rappelons-le, ne font rien.

Dans la liste suivante :

```
garçons :  
<ul>  
<li>Alain</li>  
<li>Marc</li>  
<li>Luc</li>  
</ul>  
filles :  
<ul>  
<li>Véronique</li>  
<li>Estelle</li>  
<li>Julie</li>  
</ul>
```

On veut attribuer un style différent au groupes garçons et filles (figure 11). On peut donc les regrouper dans une balise div :

```
garçons :  
<div id="boys">  
<ul>  
<li>Alain</li>  
<li>Marc</li>  
<li>Luc</li>  
</ul>  
</div>  
filles :  
<div id="girls">  
<ul>  
<li>Véronique</li>  
<li>Estelle</li>  
<li>Julie</li>  
</ul>  
</div>
```

Et le style donné dans la feuille de style (notez la syntaxe) :

```
#boys {  
background: blue;  
}  
#girls {  
background: pink;  
}
```



FIGURE 11 – Résultat

## 5.2 Exercice

Dans le texte, nous allons à l'aide de balises `span` créer une classe « `neuch` » afin de mettre en gras et en couleur `#b22222` tous les mots « Neuchâtel » du texte.

Ensuite, à l'intérieur d'une `<div>` à laquelle nous attribueront l'id « `menu` » et d'une liste à puces, nous allons créer le menu suivant :

```
<ul>  
<li><a href="http://fr.wikipedia.org/wiki/Canton_de_Neuch%C3%A2tel"><li> Région</a></li>  
<li><a href="http://fr.wikipedia.org/wiki/Neuch%C3%A2tel"><li> Histoire</a></li>  
<li><a href="http://fr.wikipedia.org/wiki/Fichier:Neuchatel2009-sm.jpg"><li> Photos</a></li>  
<li><a href="http://www.google.com/">Liens</a></li>  
</ul>
```

Notez que le menu est cliquable grâce aux liens vers des pages externes.

Ensuite, nous lui attribuerons les éléments de style :

- un fond de couleur `#4682b4` .
- une police en gras.

### 5.3 C'est dans la boîte !

A peu près tous les éléments de la page HTML peuvent être considérés comme des boîtes avec les caractéristiques suivantes (figure 12) :

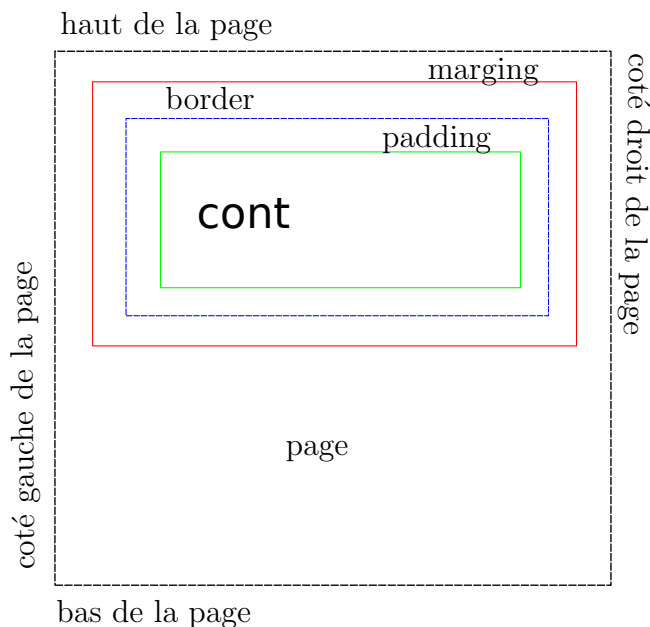


FIGURE 12 – Le modèle des boîtes

Si tout ceci ne vous dit rien, imaginez bien l'ampleur de l'utilisation concernant la présentation d'une page web, menus, images, paragraphes...etc. Cela signifie que vous pouvez voir tous les éléments de la page comme des boîtes que vous pouvez placer où vous le voulez, leur attribuer un style et toute sortes d'autres choses.

La maîtrise de cette notion vous permettra donc de créer de magnifiques pages sur lesquelles chaque élément possède le style que vous voulez et à la place que vous voulez.

#### 5.3.1 gérer les marges d'un élément

Une boîte possède 4 côtés : **haut** (top), **droit** (right), **bas** (bottom), et **gauche** (left). Donc, si par exemple, nous voulions fixer les marges du body, nous pourrions faire :

```
body {  
margin-top: 100px;  
margin-right: 40px;  
margin-bottom: 10px;  
margin-left: 70px;  
}
```

#### 5.3.2 Gérer les espacements d'un élément

Le contenu peut « bouger » dans sa boîte. Par exemple, pour que les titres de niveau 1 soient décalés de 10 px :

```
h1 {
```

```
padding-left: 10px;
}
```

### 5.3.3 Les bordures

Ici encore, méfiance : ce n'est pas parce-que plein de possibilités s'offrent à nous qu'il faut en faire des tonnes. On pourra donc faire varier le style, l'épaisseur et les couleurs des bordures, reste à voir si c'est utile ou... joli.

- Épaisseur des bordures : **border-width** avec un ordre de variation de 1 à 10 pixels, ou plus simplement **thin**, **medium** ou **thick**.
- Couleur des bordures : **border-color**
- Style des bordures : **border style**. Attention, l'intérêt esthétique est quand même très limité... Pour de beaux résultats, les pages internet utilisent souvent des images de fond plutôt que ce genre de merveilles (voir figure 13).

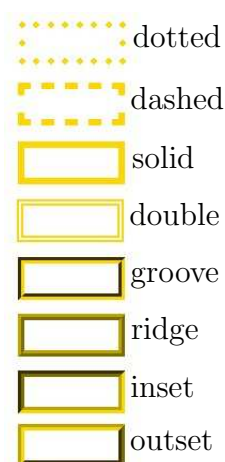


FIGURE 13 – Les différentes bordures possibles

Exemple : un titre de niveau 1 encadré avec un trait jaune en pointillé, épaisseur 3 pixels.

```
h1 {
border-width: 3px;
border-color: yellow;
border-style: dotted;
}
```

Résultat :

**La recette de la fondue aux marshmallows**

### 5.3.4 La hauteur et la largeur

Jusqu'ici, nous ne nous sommes pas posé la question de la taille des éléments, mais le CSS peut gérer ces paramètres. Un exemple utile : nous voulons créer un menu dans une div de 150 pixels sur 160, encadré de noir avec un fond gris (figure 14). pour cela, nous créons une liste à l'intérieur d'une div de classe "menu" dans le fichier html

```
<div class="menu">
<ul>
<li>Afrique</li>
<li>Allemagne</li>
<li>Portugal</li>
<li>Suisse</li>
</ul>
</div>
```

et nous lui joignons le css

```
div.menu {
width: 150px;
height: 160px;
border: 1px solid black;
background: gray;
}
```



FIGURE 14 – Résultat

### 5.3.5 Le positionnement sur la page

Nous pouvons également placer les boîtes où nous le voulons sur la page ! Il existe deux types de positionnements :

- le positionnement **absolu** (absolute) signifie que l'objet est déplacé à la position choisie et **ne laisse pas d'espace vide** à sa position originale. C'est en général ce que l'on cherche à obtenir.
- le positionnement **relatif** (relative) signifie que l'objet est déplacé à la position choisie et **laisse un espace vide** à sa position originale.

Un exemple : le menu du chapitre précédent doit être positionné à 100 pixels par rapport à la droite de la page et à 50 pixels à partir du haut de la page. On ajouterait au CSS les options de positionnement :

```
div.menu {  
width: 150px;  
height: 160px;  
border: 1px solid black;  
background: gray;  
position: absolute;  
right: 100px;  
top: 50px;  
}
```

Bien entendu, les options de position sont **top**, **right**, **bottom** et **left**.

## 5.4 Exercice

Nous allons améliorer le menu créé précédemment ainsi que le positionnement des éléments sur la page.

Pour commencer, le paragraphe du texte :

- positionnement absolu.
- à 330 pixels de la gauche, 400 pixels du haut.
- de largeur 950 pixels.
- profitons-en pour supprimer les passages à la ligne que nous avons insérés pour descendre le titre : nous pouvons à présent placer le titre où nous le voulons ! (faites des tests)

Puis le menu :

- positionnement absolu.
- largeur et hauteur 20 pixels.
- à 70 pixels de la gauche, 430 pixels du haut.
- complétez la pseudo-classe des liens pour que les liens du menu prennent au survol de la souris la couleur #485c6d .

Admirez comme tout cela prend forme !

## 5.5 Les éléments flottants

Nous pouvons obtenir de beaux résultats sur une page en laissant « flotter » une boîte à gauche ou à droite du texte (figure 15) :



FIGURE 15 – Un élément flottant à droite

Pour cela, nous aurions dans le HTML :

```
<div id="image"></div>
<p>La pizza est un plat d'origine napolitaine...
</p>
```

associé au CSS

```
#image {
float: right;
width: 400px;
}
```

## 5.6 Exercice

A présent, insérez [cette image](#) en flottante à droite du texte. Vous pouvez visualiser le résultat de l'exercice sur la figure 17.

## 6 Et pour la suite ?

Si vous vous rendez sur un site que vous aimez à l'aide du navigateur Firefox, vous pouvez faire un clic droit et sélectionner « code source de la page », puis essayez de lire le code (particulièrement l'intérieur du body. Vous constaterez vite que les présentations sont obtenues par différents moyens! Vous trouvez aussi dans le code source des liens vers les feuilles de style (dans le head) que vous pourrez aussi consulter. C'est ici qu'on s'aperçoit de la complexité pour obtenir une page plus évoluées! Cela dit, vous y apprendrez beaucoup d'autre subtilités que vous pourrez intégrer à vos pages.

Pour clore l'exercice que vous avez fait vous pouvez librement adapter le CSS à votre goût et, partie intéressante, créer 2-3 autre pages (le contenu n'est pas hyper important), changer les adresse du menu pour qu'elles pointent sur ces pages, et attribuer la même feuille de style à toutes ces pages. Ce sera le moment le plus intéressant puisque vous verrez de vos yeux le style s'appliquer à toutes les pages!



FIGURE 16 – Encore du style...



### Le lac de *Neuchâtel*

- [Region](#)
- [Histoire](#)
- [Photos](#)
- [Liens](#)

Neuchâtel est une commune de Suisse, chef-lieu du canton de **Neuchâtel** et du district de Neuchâtel. Située au bord du lac de Neuchâtel, sur le flanc sud du massif du Jura, elle fait face à la chaîne des Alpes.

**Neuchâtel** est citée pour la première fois en 1011. Elle est gérée par des comtes de **Neuchâtel** jusqu'en 1458, puis passe sous l'autorité de la famille d'Orléans-Longueville jusqu'en 1707. La ville est ensuite sous l'autorité des rois de Prusse jusqu'en 1848, tout en ayant dès 1814 adhéré à la Suisse en formant le canton de **Neuchâtel**.

L'économie de la ville a longtemps été liée à l'administration (ville la plus importante de la région) et à l'agriculture (vigne). Elle a été dominée par le commerce international au XVIIIe siècle, puis par les secteurs secondaire (horlogerie) et tertiaire. Au tournant des XIXe et XXe siècles plusieurs hôtels sont construits pour soutenir le tourisme qui accompagne le développement du chemin de fer. La région est fortement industrielle avec la chocolaterie Suchard, les forges et scieries Martenet, l'horlogerie et la fabrication d'indiennes qui se développent durant le XIXe siècle et s'effondrent à la fin du XXe siècle. Elle se renouvelle aujourd'hui grâce aux technologies de pointe. La ville est également connue pour son université, son club de football, ses activités de hautes technologies et, sur le plan touristique, par son bourg médiéval et sa situation. Elle a en outre été l'une des quatre villes organisatrices d'Expo.02.

Source : [Wikipedia](#)

FIGURE 17 – Résultat de l'exercice HTML