

## Partie 2

### Initiation aux langages HTML et CSS



### Support Pratique

### Utilisé en TD

samir DELIMI

[samir.delimi@univ-montpellier.fr](mailto:samir.delimi@univ-montpellier.fr)

# INTRODUCTION GENERALE

## OBJECTIFS :

L'objectif général de ce cours est de comprendre le fonctionnement du web pour mieux le maîtriser.

Dans la première partie, nous avons fait une présentation générale d'internet, du web et des protocoles de communication.

L'objectif de cette deuxième partie est de vous aider à comprendre comment construire une page web simple et efficace, avec une structure de base en **HTML** (HyperText Markup Language) et comment la mettre en forme à l'aide de propriétés **CSS** simples.

Nous ne traiterons pas la partie dynamique du web (JavaScript, PHP, MySQL entre autres), ce n'est pas l'objet de cette partie du cours.

## PREPARATION DE L'ENVIRONNEMENT DETRAVAIL :

### PREMIERS OUTILS POUR COMMENCER :

Pour suivre la partie pratique de ce document, nous aurons besoin de deux outils :

1. **Un éditeur de texte** pour écrire le code, comme **Notepad++** pour Windows ou **Wrangler** pour Mac
2. Et **un Navigateur Web** (Browser) pour afficher le contenu des pages web, comme Chrome ou FireFox.

Dans la suite du TD, nous utiliserons **Notepad++** et **Mozilla Firefox ou chrome**

*Nous n'avons pas besoin d'être connecté sur Internet pour créer nos pages html (web) ni d'avoir le dernier Editeur Html, sans doute plus performant, pour faire nos premiers pas.*

### EXEMPLES D'EDITEURS DE TEXTE



Quelques libres : le Bloc-notes, Notepad ++ pour Windows,

TextWrangler pour Mac OS X

Brackets (libre Adobe), Sublime Text 2 (version démo) pour Windows, OSX et Linux

### EXEMPLES DE NAVIGATEURS WEB :



Mozilla Firefox ou Google chrome, Internet Explorer par exemple.

---

### Remarque :

Vous pouvez utiliser des versions portables de ces outils pour les porter avec vous sur clé USB par exemple.

Il est recommandé de valider votre code source pour :

- Savoir si votre code respecte les normes du consortium W3C
- Être sûr que notre page web sera affichée correctement dans un navigateur web, qui respecte les normes du W3C, dans n'importe système, n'importe quelle taille d'écran (responsive) et langue !
- Vos pages web seront lues par des personnes handicapées
- La correction des erreurs de syntaxe. Erreur dans la syntaxe de l'écriture des balises par exemple : si une balise n'est pas fermée, tout ce qui suit va être contenu dans la boîte de cette balise. Html et css ne sont pas des langages compilés (modules de corrections automatisés etc...).
- Il peut aussi y avoir d'autres soucis.
- C'est pratique, cela nous permet aussi d'apprendre ce que nous avons le droit de faire ou pas.

Pour valider vos pages web, vous avez différentes solutions :

- Utiliser des moteurs en ligne comme le propose le W3C à cette adresse : (<http://validator.w3.org>)
- Utiliser des moteurs de validation intégrés à votre logiciel de conception web.
- Utiliser des extensions de votre navigateur Mozilla Firefox (HTML Validator) par exemple.

La troisième solution est la plus pratique.

# CHAPITRE 1 : LES BASES DU LANGAGE HTML

## COMMENT ÇA FONCTIONNE

Le HTML fournit au navigateur des instructions sur la **structure** et le **contenu** des pages à afficher (texte, images...). Par exemple, vous y écrirez des titres, des paragraphes, des liens, des listes à puces ou numérotées...

Il utilise des **balises**.

## LES BALISES HTML ET LEURS ATTRIBUTS :

Les **balises** HTML structurent le contenu d'une page web, en délimitant des blocs qui seront amenés à contenir des paragraphes, des titres, différents types de médias (images, sons, vidéos), des contrôles de formulaires ou encore des liens hypertextes.

C'est une façon de hiérarchiser ces données en les entourant par une balise ouvrante et une balise fermante.

Les balises sont repérables car elles sont encadrées par les signes **<** et **>**

### BALISES DOUBLES OU PAIRES

Elles encadrent un texte et sont composées de deux parties, une ouvrante **< >** et une fermante **</ >**

```
<p> voici ma page web </p>
```

### BALISES SIMPLES OU ORPHELINES

Elles servent à insérer un élément à un endroit déterminé, elles se placent généralement après un élément et peut être écrit :

```
< br> ou <br />
```

## LES DIFFERENTS TYPES DE BALISES

- **Balises contenant des métas informations** telles que : `<head>` ou `<title>.`
- **Balises de mise en forme** du document telles que : `<b>` `<i>` ou encore `<table>` les Balises de format ne sont plus recommandées, il faudra utiliser les styles !

- **Balises de liens** telles que

```
<a>.
```

- **Balises d'insertions (multimédia)** ou comment insérer des images, du son, de la vidéo.

---

### QUELQUES REGLES SIMPLES POUR L'ECRITURE EN LANGAGE HTML :

- Les balises HTML ont une marque de début 

```
<...>
```

 et une marque de fin 

```
</...>
```
- Certaines marques de fin sont facultatives
- Les balises HTML sont écrites en minuscules de préférence. C'est une des recommandations W3C.
- L'extension d'un fichier rédigé en HTML doit être : **.html, .htm ou .php**

---

### LES ATTRIBUTS DES BALISES

Dans les balises, peuvent être ajoutés des **attributs**. Ce sont des textes qui fonctionnent comme des options.

Les attributs consistent en deux parties :

1. Un **nom** d'attribut
2. Une **valeur** d'attribut

Les valeurs d'attribut consistant en un seul mot ou nombre, peuvent être écrits tel quel, mais dès qu'il y a deux ou plusieurs chaînes de caractères comme valeur, il doivent être écrits avec des guillemets ("). Les guillemets simples (') ou les guillemets doubles (") peuvent être utilisés. De nombreux auteurs de pages web préfèrent toujours utiliser des guillemets afin de rendre le code moins ambigu, plus facile à lire et éviter les erreurs.

```
<img nom= "monimage.jpg" > </img>
```

On comprend ici que l'image à afficher est celle qui se nomme monimage.jpg.

**L'attribut** est inscrit uniquement dans la balise ouvrante, on ne le répète pas dans la fermante.

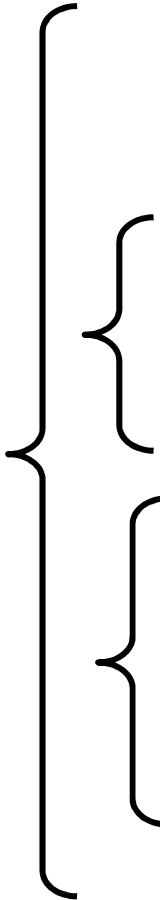
---

### LA VALIDATION D'UNE PAGE WEB

Comme indiqué ci-dessus, il est recommandé de valider vos pages web avec une extension au navigateur ou l'utilisation d'un outil en ligne comme le propose le W3C à cette URL : <http://validator.w3.org/>

## STRUCTURE GENERALE D'UNE PAGE HTML

Pour créer une page HTML, il faut commencer par créer sa [structure](#) (voir l'exemple visuel ci-dessous).



```
<!DOCTYPE html> <!--Identification du type du document
                        (note ceci est un commentaire) -->

<html>

  <head>
    <meta charset="utf-8"/>
    <title>
      Titre de la page
    </title>
  </head>
  <body>
    Ma première page Web
    <!-- ceci est un Commentaire -->
    <!-- mettre ici le texte a afficher sur la page web -->
  </body>
</html>
```

Une page web est constituée de deux parties :

**L'en-tête (head)** et **le corps (body)**.

---

### L'ENTETE OU LE HEAD <HEAD>

L'entête contient des informations générales sur la page comme le titre, les informations sur l'encodage et les balises de référencement. Ces informations qui ne sont pas affichées par le navigateur. **Ces informations sont importantes pour les navigateurs et moteurs de recherche par exemple.**

---

### MENTION DU DOCTYPE

**<!DOCTYPE html>**

Le doctype est indispensable, il mentionne qu'il s'agit d'une page web html et spécifie la version du langage html qui sera utilisée pour le site.

Si vous voyez une ligne Doctype courte du style <!DOCTYPE html>, cela signifie que la page est codé en HTML 5, soit la dernière version du langage.

---

### MENTION DU LANGAGE

## La balise `<html>` ... `</html>`

- C'est la balise principale. Elle représente tout le document et englobe tout le contenu de la page web. Elle est constituée, comme la plupart des balises HTML, d'une balise d'ouverture et de fermeture, soit `<HTML>` pour l'ouverture, et `</html>` pour la fermeture.

---

## L'EN-TETE DE LA PAGE

`<head>` ... `</head>`

---

## LE CORPS DE LA PAGE

Le corps est la partie principale de la page web et affichable par le navigateur.

`<body>` `</body>`

---

## LA BALISE `<META>`

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

Il s'agit du format de caractère utilisé dans la page web.

Normalement l'encodage "universel" est le « utf-8 » qui permet de lire tous les caractères utilisés de toutes les langues du monde.

Les balises meta sont des balises qui servent à écrire des métas données (metatags) pour la page web. Elles n'ont pas vocation à s'afficher, c'est pourquoi elles sont réservées au head de la page. Elles sont destinées aux navigateurs, aux moteurs de recherche et aux outils d'indexations.

`< meta name = " description " content = " La description de la page " >`

`< meta name = " keywords " content = " Ecrire ici les mots clés " >`

`< meta name = " author " content = " Samir DELIMI " >`

---

## LA BALISE TITRE

Donne un titre qui s'affichera dans l'onglet de votre navigateur, il est utile voir obligatoire pour le référencement

`<title>` titre la page web `</title>`

---

## LES COMMENTAIRES

Ecrire des commentaires dans le code pour laisser des indications, donner des informations, se repérer sur une page compliquée, s'utilise partout dans `<head>` et `<body>`

`<!-- Commentaires -->`

---

## LA BALISE LINK :

Utilisée dans la partie head, La balise link sert à relier la page html à sa feuille de style css ou à d'autres documents externes qui seront utilisé en général dans le rendu de la page.

Certains sont indispensables au navigateur, comme le lien sur une feuille de style, ou l'icône du site par exemple.

Lier une feuille de style :

```
<link href="/dossier/fichier.css" rel="stylesheet" type="text/css" media="screen"
```

```
/>
```

Concernant le **favicon**, si vous réalisez un site dit **responsive design** (adapté à tous les types d'écran: ordinateur de bureau, portable, tablette, smartphone, iPhone, iPad...) il faut créer un favicon pour chaque type d'appareil.

```
< link rel = " shortcut icon " href = " images / favicon.ico " >
```

```
< link rel = " icon " type = " image/x-icon " href = " images / favicon.ico " >
```

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

```
< link rel = " apple-touch-icon " href = " images / apple-touch-icon.png " >
```

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

## Les scripts :

Une page html peut faire appel à des scripts internes ou externes.

Aujourd'hui, de nombreux sites (la plupart) utilisent des bibliothèques jQuery : menus, slider, formulaires de contact...

```
< script type = " text / javascript " src = " javascript / respond.min.js " > < / script >
```

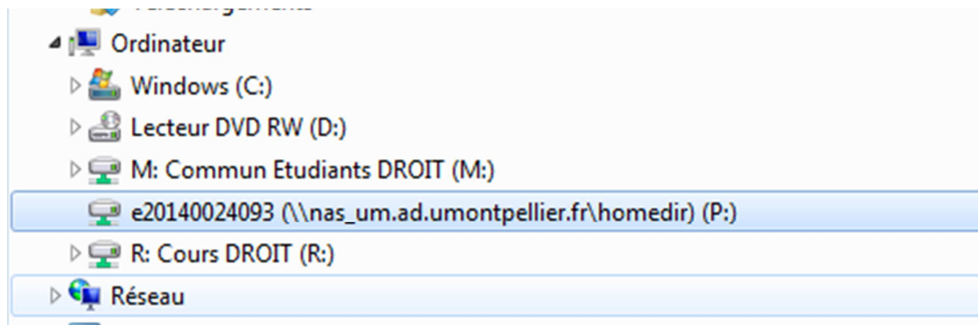
```
< script type = " text / javascript " src = " jquery / jquery-1.8.2.min.js " > < / script >
```



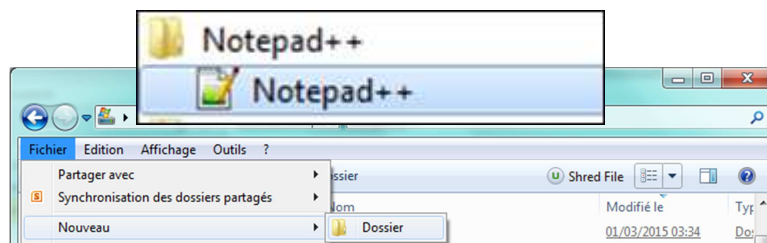
### PREPARATION DU DOSSIER DE TRAVAIL :

A l'aide de l'explorateur Windows et dans votre dossier personnel (lecteur réseau P) :

- **Créer un nouveau dossier** nommé ce nouveau dossier : **TD\_HTML** par exemple.



- **Ouvrir l'éditeur de texte préféré de votre système pour Windows : bloc-notes, notepad++, Sublime text2 par exemple.**
  - Démarrer, Tous les programmes, accessoires, Notepad ++



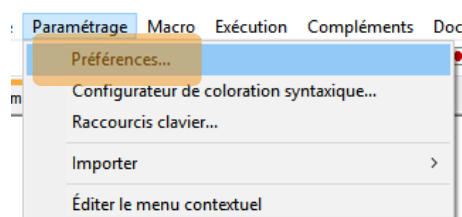
Pour la suite des exemples nous utiliserons Notepad++. Vous pouvez aussi utiliser Sublimetext 2. L'utilisation du bloc-notes est aussi possible mais pas très confortable seulement.

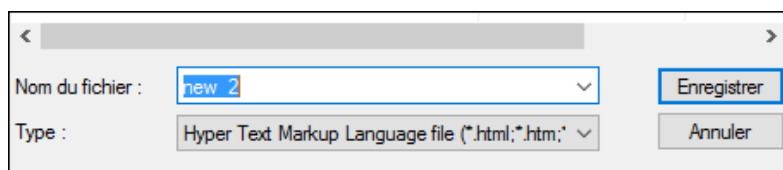
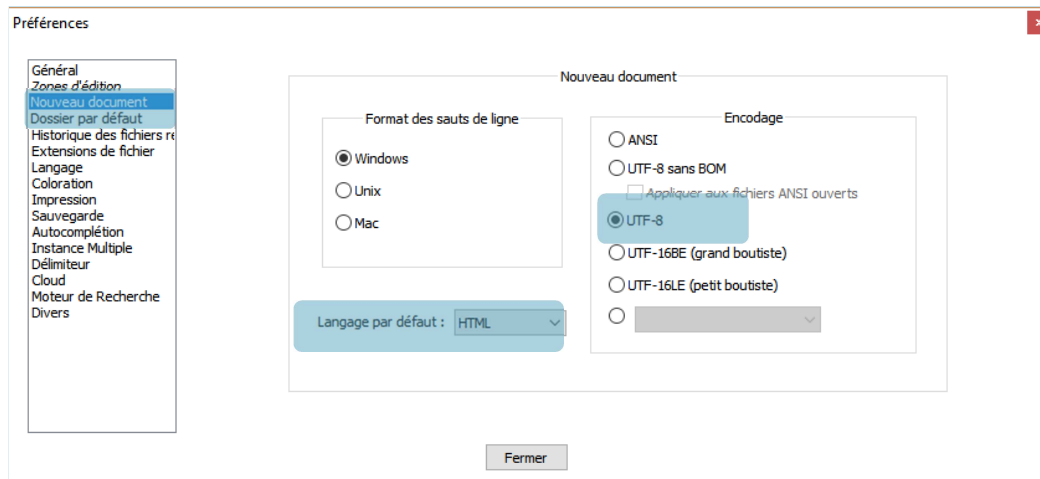
- **Paramétrage du langage : Pour le HTML, On fera la même chose pour le CSS:**

Notepad++ peut-être utiliser comme éditeur de la plupart des langages de programmation, pour l'utiliser pour le HTML (CSS), nous allons le paramétrer pour que les nouveaux fichiers soient des pages html.

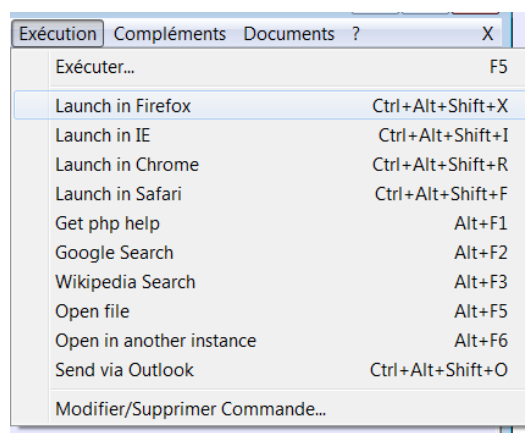
Vous pouvez aussi paramétrer votre dossier par défaut à. Mettre votre lecteur Perso P :

Cela nous donnera le type de fichier à enregistrer par défaut : .html et .css.





- Saisir le code de votre page HTML du TD (ci-dessous) demandé par votre chargé de TD
- Enregistrer le document (la page) sous le nom td1 avec l'extension **.html** ou **.htm**
- Afficher la page web sur le navigateur (Client). Préférer le navigateur Mozilla Firefox ou Chrome.
  - Menu Exécution, Launch in Firefox ou Chrome



- Mettre cote à cote la fenêtre de Notepad (Backend) et le navigateur (FrontEnd) pour basculer entre les différentes fenêtres rapidement.
- Raccourci : **CTRL + S** pour sauvegarder la page sur Notepad++
- et **F5** pour actualiser la page web sur le navigateur.
- Pour les tous débutants :
  - Pour éviter tout problème de fichiers en double etc..., laisser votre dossier (répertoire) de travail ouvert.

## EXERCICE PRATIQUE 1 :

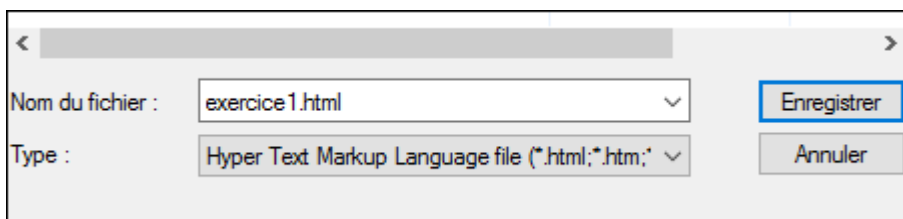
Ecrire la page web suivante, la présenter sur le navigateur et la valider.



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <title> Titre </title>
6   </head>
7   <body>
8
9 </body>
10
11 </html>
```

length : 127 lines : 11 Ln : 11 Col : 8 Sel : 0 | 0 Dos\Windows UTF-8 INS

- Enregistrer le document (la page) sous le nom **exercice1** avec l'extension **.html** ou **.htm**
  - Attention si vous utilisez le bloc-notes : mettre dans Type : **tous les fichiers**.



- Ouvrir la page dans le navigateur

**Rappel :** Pour vos éventuelles modifications, il n'est pas nécessaire de rouvrir à chaque fois le navigateur. Laissez votre navigateur ouvert.

- Retourner dans l'éditeur de texte (sans fermer le navigateur).
- Modifier les codes Html.
- Enregistrer le fichier.
- Utiliser la commande **actualisée** du menu affichage du navigateur. Raccourci clavier : Ctrl + R pour Mozilla et F5 pour Internet Explorer.
- Vous pouvez ouvrir les deux fenêtres l'éditeur et le navigateur en mosaïque (cote à cote) sur votre écran.

### VALIDATION :

Vous devrez vous assurer que le code source produit est valide. Cette étape est importante car elle vous prémunira des fautes de frappe ainsi que d'erreur dans la structure du code (mauvaise balise, erreur d'imbrication, etc...).

Vous valider votre code, allez sur le site officiel du W3C <http://validator.w3.org/>. Ensuite, uploadez votre fichier HTML ou copié/collé le code source.

### LE TEXTE EST GÉNÉRALEMENT ÉCRIT À L'INTÉRIEUR DE PARAGRAPHES.

- *Html propose la balise <p> pour délimiter les paragraphes*

#### Exemple :

```
<p> bonjour et bienvenue sur notre site </p>
```

- *La balise Retour à la ligne : <br/>*

#### Exemple :

```
<p> bonjour et bienvenue sur mon site <br/>  
Je suis heureux de vous présenter ce que je sais faire </p>
```

### LES TITRES, SOUS-TITRES, TEXTE COURANT, ETC. :

C'est là où le langage CSS est utile aussi.

Par défaut, il y a 6 types de hiérarchisation des caractères en html, nommés :

**<h1>, <h2>, ... <h6>**

Autres mises en exergue de textes : mise en gras, en italique, surlignage, etc.

**<strong> ... </strong>** Gras ou texte important.

**<em> ... </em>** Italique ou texte.

**<mark> .... </mark>** Surlignage.

### EXERCICE : RAJOUTER À LA PAGE DES TITRES H1, H2 , ... H6

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="utf-8" />  
    <title>Niveaux de titres</title>  
  </head>  
  
  <body>  
    <h1>Titre super important</h1>  
    <h2>Titre important</h2>  
    <h3>Titre un peu moins important (sous-titre)</h3>  
  
    <h4>Titre pas trop important</h4>  
    <h5>Titre pas important</h5>  
    <h6>Titre vraiment pas important du tout</h6>  
  </body>  
</html>
```

Si l'on souhaite centrer le titre, le mettre en rouge et le souligner par exemple, nous ferons tout cela plutôt avec les CSS.

### LA MISE EN VALEUR

Dans une page web et au sein de ses paragraphes, certains mots sont parfois plus importants que d'autres et nous souhaitons les faire ressortir. HTML propose différents moyens de mettre en valeur le texte de votre page.

### **Mettre un peu en valeur : `<em>` ... `</em>` Italique ou texte.**

Pour mettre un peu en valeur votre texte, vous devez utiliser la balise `<em>` `</em>`.

Utiliser la balise `<em>` a pour conséquence de mettre le texte en italique. En fait, c'est le navigateur qui choisit comment afficher les mots. On lui dit que les mots sont assez importants et, pour faire ressortir cette information, il change l'apparence du texte en utilisant l'italique.

### **Mettre bien en valeur : `<strong>` `</strong>` Gras ou texte important.**

Pour mettre un texte bien en valeur, on utilise la balise `<strong>` qui signifie « fort », ou « important » si vous préférez. Elle s'utilise exactement de la même manière que `<em>` :

### **Marquer le texte : `<mark>` . `</mark>` Surlignage.**

La balise `<mark>` permet de faire ressortir visuellement une portion de texte. Par défaut, `<mark>` a pour effet de surligner le texte.

On pourra changer l'affichage en CSS (décider de surligner dans une autre couleur, d'encadrer le texte, etc.). C'est le même principe pour les balises (`h1`, `h2`, `Strong`, `em` `mark`), ces balises indiquent le sens du texte et non pas comment ceux-ci doivent s'afficher.

Le rôle des balises est d'indiquer le sens du texte. Ainsi, `<strong>` indique à l'ordinateur « Ce texte est important ». C'est tout. Et pour montrer que le texte est important, l'ordinateur décide de le mettre en gras (mais il pourrait aussi bien l'écrire en rouge !). La plupart des navigateurs affichent les textes importants en gras, mais rien ne les y oblige.

### Comment créer des listes et les référencer avec des puces, des tirets, etc.

Les balises `<ul>` `</ul>` créent une liste non ordonnée :

Entre ces deux balises on désigne chaque élément (item) entre deux autres balises `<li>` `</li>`

### EXEMPLE :

```
<ul>
    <li> premier étage </li>
    <li> deuxième étage </li>
</ul>
```

### Comment créer des listes et les référencer avec des puces, des tirets, etc.

Les balises `<ol>` `</ol>` créent une liste ordonnée

Entre ces deux balises on désigne chaque élément entre deux autres balises `<li>` `</li>`

```
<ol>
    <li> premier étage </li>
    <li> deuxième étage </li>
</ol>
```

Résumé :

Liste numérotée :	Liste à puces (non numérotées) :	Liste de tabulations
<code>&lt;ol &gt;</code>	<code>&lt;ul&gt;</code>	<code>&lt;dl&gt;</code>
<code>&lt;li&gt;élément 1,</code>	<code>&lt; li &gt;élément 1,</code>	<code>&lt;dt&gt; élément 1,</code>
<code>&lt;li&gt;élément 2,</code>	<code>&lt; li &gt;élément 2,</code>	<code>&lt;dt&gt; élément 2,</code>
<code>&lt;li&gt;élément 3.</code>	<code>&lt; li &gt;élément 3.</code>	<code>&lt;/dl&gt;</code>
<code>&lt;/ol&gt;</code>	<code>&lt;/ul&gt;</code>	<code>&lt;dl&gt;</code>

## EXERCICE : RAJOUTER A LA PAGE DES TITRES H1, H2 , ...H6

### Rajouter une ou deux listes

## FAIRE DES LIENS

Les liens permettent de changer de page dans le même site ou vers un autres site. Ils sont, par défaut, écrits en bleu et soulignés.

### La balise de création lien `<a> ... </a>`

Pour insérer un lien, on utilise la balise `<a>` avec l'attribut `href` pour indiquer l'adresse de la page cible.

La balise `<a>` (anchor) signale un lien. Elle est utilisée sur la majorité des pages Web qui proposent des liens vers d'autres documents et d'autres sites. il ne faut pas oublier que les liens sont les éléments clés de l'ensemble du réseau Internet. Les liens sont aussi des éléments importants utilisés par Google et les autres moteurs de recherche pour trouver et indexer votre site.

### L'attribut : `href` :

L'attribut `href` s'ajoute à la balise `<a>`, il sert à saisir la destination du lien. C'est généralement une autre page de votre site, mais il peut aussi s'agir d'un autre site Web.

### Exemple :

Ainsi pour créer un lien dans une page on écrira :

```
<a href="http://www.umontpellier.fr"> Lien vers le site de l'université </a>
```

Par défaut, le lien s'affiche en bleu souligné. Si vous avez déjà ouvert la page, le lien s'affiche en violet. Nous verrons comment changer cette apparence lorsque nous étudierons le CSS.

### Quelques indications supplémentaires

- Il est possible de faire ouvrir des liens dans un autre onglet du navigateur par exemple en rajoutant une *cible* :  
**`target="blank"`**

### Exemple :

```
<a href="http://www.umontpellier.fr " target="_blank"> Visitez le site de l'UM </a>
```

- un lien pour envoyer un email

Il est également possible de faire **un lien pour envoyer un email** :

### Exemple :

```
<a href="mailto:samir.delimi@umontpellier.fr"> écrivez moi</a>
```

- Un lien pour télécharger un fichier

Il est également possible de faire télécharger un document

```
<a href="document/nomfichier.pdf"> écrivez moi</a>
```

- Il est possible de rajouter un texte qui s'affiche en surimpression lorsque la souris passe sur le lien

```
<a href="http://www.umontpellier.fr " title="Lien vers le site de l'Université de Montpellier" > site de l'UM </a>
```

### Il existe des liens absolus et des liens relatifs

Liens **absolus** = adresse complète (`href="http://www.umontpellier.fr"`)

Liens **relatifs** = adresse vers un fichier qui se trouve dans le dossier du site (`href="contact.html"`)

### Les liens en utilisant les ancres:

Une ancre est une sorte de point de repère que vous pouvez mettre dans vos pages HTML lorsqu'elles sont très longues. Il est possible également de faire des **liens vers une "ancre" = un point x de la page en cours ou vers un point y d'une autre page.**

#### Créer une ancre :

Il suffit de rajouter l'attribut **id** à une balise qui va alors servir de repère. Ce peut être n'importe quelle balise, un titre par exemple.

Utilisez l'attribut **id** pour donner un nom à l'ancre. Cela nous servira ensuite pour faire un lien vers cette ancre. Par exemple :

### Exemples de liens vers des sous-titres :

```
<h2 id="ancre1"> Les cours </h2>
<p> texte </p>
<h2 id="ancre2"> Les Horaires</h2>
<p> texte </p>
```

Ensuite, il suffit de créer un lien comme d'habitude, mais cette fois l'**attribut href** contiendra un dièse (#) suivi du nom de l'ancre. Exemple :

```
<a href="#ancre1"> revoir les cours </a>
```

L'appel au lien vers **une ancre dans une autre page** se fera :

```
<a href="presentation.html#ancre1" > revoir les cours </a>
```

### EXERCICE : RAJOUTER A LA PAGE DES TITRES H1, H2 , ... H6

**Rajouter un lien interne, un lien vers une autre pages du site et un autre lien vers une ou deux listes de votre choix.**



## La balise `<img />`:

- C'est une balise **orpheline** (comme `<br />`).
- La balise doit être accompagnée de deux attributs obligatoires:
  - o **src** (= l'emplacement de l'image en lien relatif ou absolu),
  - o et **alt** (=texte qui remplace l'image en cas de non affichage, pour accessibilité)

### Exemple d'insertion d'image :

``

*Attention, l'image est dans un dossier images.*

On peut ajouter un autre attribut (facultatif) pour générer une **infobulle** avec `title="..."`

Vous pouvez maintenant insérer des images :

1. À l'intérieur de `<p> .... </p>`
2. à l'intérieur d'une balise `<figure> ... </figure>`

Si l'image illustre le texte, préférez `<p>`

Si l'image apporte une information complémentaire au texte, préférez `<figure>`

### Attributs: A utiliser plutôt ceux des CSS:

(Texte alternatif) <code>alt="****"</code>	Pour les browser n'ayant pas l'option "image" activée
(Dimensions) <code>width=x height=y</code>	Hauteur et largeur (en pixels)
<code>border=x</code> (en pixels)	Bordure
<code>align=top align=middle</code> <code>align=bottom</code> <code>align=left align=right align=center</code>	Alignement de l'image faisant partie intégrante du texte  Alignement de l'image ne faisant pas partie intégrante du texte

**Sans préciser de longueur et de largeur dans la balise `<img>`,** qui affichera l'image dans ses dimensions originales par défaut.

Exemple d'image redimensionnée : ``

Exemple d'image avec bordure : ``

Exemple d'image faisant partie intégrante du texte

`<a href="nom_document_cible.htm"></a>`

## Quels formats d'images utiliser pour le web?

- Choisir au mieux un format afin que l'image puisse s'afficher rapidement lors de la consultation de la page.

Attention, il existe de nombreux formats, chacun est réservé à une fonction particulière.

Pour le web, préférez :

Le **jpg** (jpeg) (*Joint Photographic Expert Group*), format très utilisé pour les photos, permet de réduire leur poids et d'afficher jusqu'à 16 millions de couleurs. Les images JPEG sont enregistrées avec l'extension .jpg ou .jpeg.

Le **png** (**P**ORTABLE **N**ETWORK **G**RAPHICS) plus récent, remplace le gif pour tous les graphiques. Le PNG a deux gros avantages : il peut être rendu transparent et il n'altère pas la qualité de l'image.

Le PNG existe en deux versions, en fonction du nombre de couleurs que doit comporter l'image :

- PNG 8 bits : 256 couleurs ;
- PNG 24 bits : 16 millions de couleurs (autant qu'une image JPEG).

Le **gif** pour les graphiques, dessins ou tout fichier qui ne demande pas une grosse gestion des couleurs (limité à 256 couleurs).

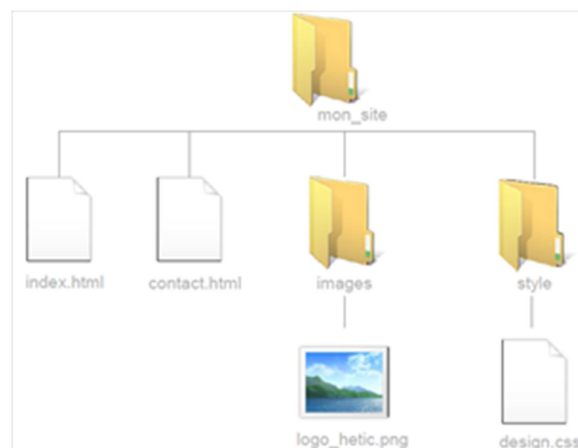
Néanmoins, le GIF conserve un certain avantage que le PNG n'a pas : il peut être animé. D'où l'explosion ces dernières années des GIF animés sur le web (aussi appelé "reaction gifs").

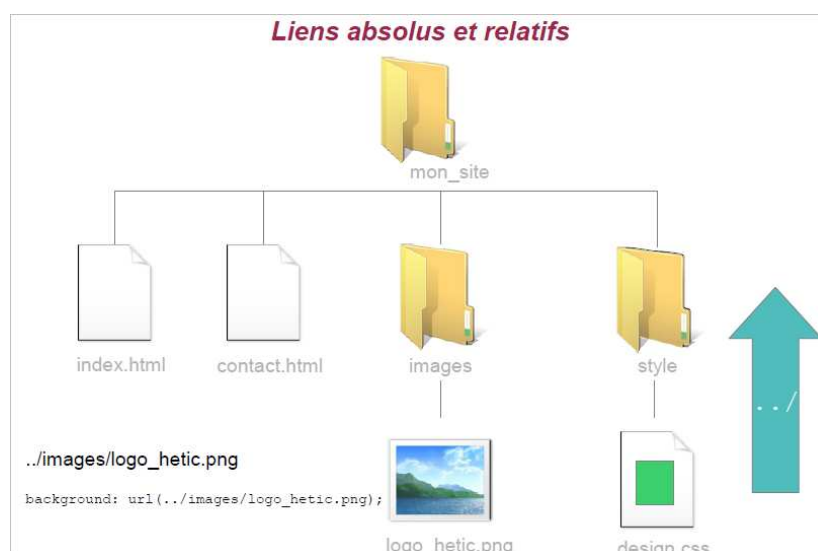
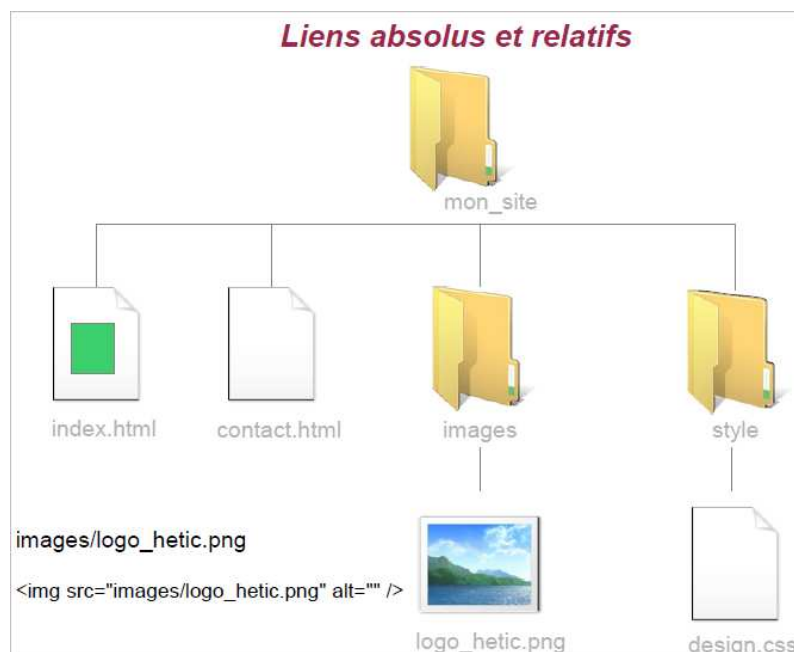
### Résumé des formats à adopter en fonction de l'image :

Une photo : utilisez un JPEG.

- N'importe quel graphique avec peu de couleurs (moins de 256) : utilisez un PNG 8 bits ou éventuellement un GIF.
- N'importe quel graphique avec beaucoup de couleurs : utilisez un PNG 24 bits.
- Une image animée : utilisez un GIF animé.

Un logiciel paint, xnview, gimp ou le leader photoshop pourront nous aider à faire nos choix.





### Insérer une Miniature cliquable

Si votre image est très grosse, il est conseillé d'en afficher la miniature sur votre site. Ajoutez ensuite un lien sur cette miniature pour que vos visiteurs puissent afficher l'image en taille originale.

### EXEMPLE :

Il faut d'abord préparer les deux versions de la photo : la miniature et l'image originale, On peut le faire avec des logiciels comme easy thumbnails pour PC et thumbs Up pour Mac OSX

Vous souhaitez voir l'image dans sa taille d'origine ? Cliquez dessus !<br />

```
<p>
<a href="img/montagne.jpg"></a>
</p>
```

Il faut disposer de deux versions de la photo

### Donnez une légende à votre image

entre <figure> à l'aide de la balise <figcaption> :

```
< figure >  
  
<figcaption> Voici le logo du site </figcaption>  
</figure>
```

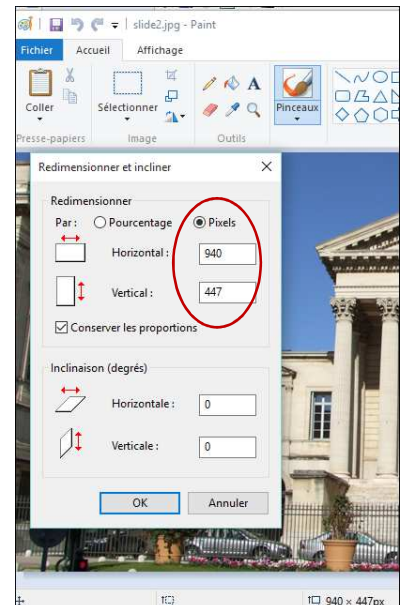
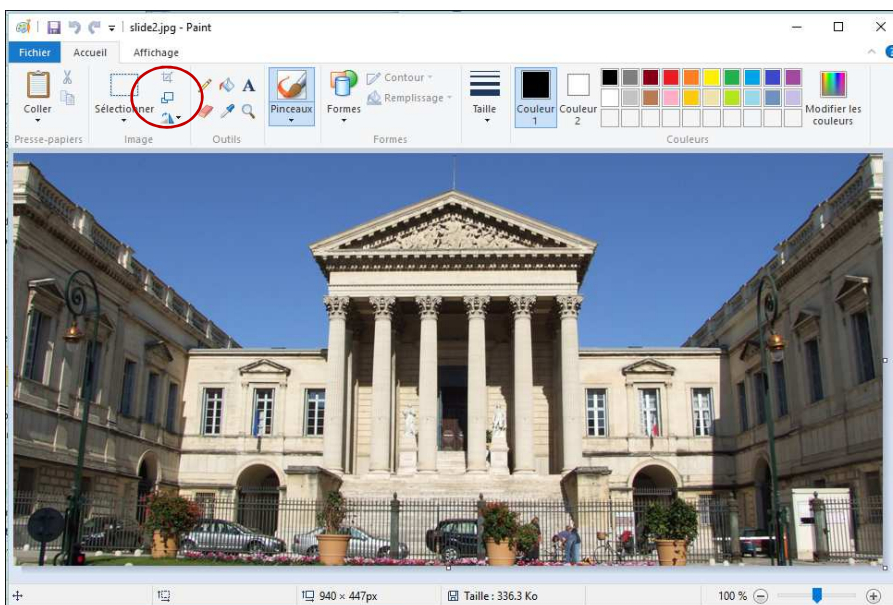
### OUTILS DE MODIFICATION DES PROPRIETES DE L'IMAGE :

Utilisez Photoshop, Gimp, Xnview pour modifier la résolution ou les dimensions d'une image.

Si vous souhaitez juste modifier les dimensions (Largeur xHauteur) Utiliser tout simplement Paint pour modifier la taille:

Exemple de modification de la largeur et hauteur d'une image :

LANCEZ PAINT, ONGLET ACCUEIL, REDIMENSIONNER ET INCLINER, PAR POURCENTAGE OU PAR PIXELS..



### EXERCICE :

Insérer une image de votre choix en bas de la page web.

L'image doit être centrée, de 250 pixels de largeur et 90 pixels de hauteur.

# CHAPITRE 2 : LE LANGAGE CSS

Le langage CSS (Cascading Style Sheets) (feuilles de styles en cascade) a été introduit, en 1996, à partir de HTML 4 pour gérer **la mise en forme des pages web de notre site**, par l'intermédiaire de **propriétés d'apparence** (couleurs, bordures, polices, etc.) **et de placement** (largeur, hauteur, côte à côte, dessus-dessous, etc.).


*On peut le comparer en quelque sorte à la barre d'outils de Microsoft Word pour la mise en forme du texte.*

Un fichier CSS (ou feuille de style) est un simple fichier texte, avec des définitions de règles de styles.

## OUTILS DE CREATION DE FEUILLES DE STYLES :

Comme les pages HTML, un simple éditeur de texte suffit. On utilisera donc notre Notepad++, Wrangler pour les Mac.

Exemple de règle de style détaillé plus loin :

Sélecteur	Déclaration 1	Déclaration 2
<b>h1</b>	{ <b>color:blue;</b>	<b>font-size:12px;</b> }
		

## OU ECRIRE LES BALISES CSS ?

On peut déclarer des styles CSS de trois façons différentes, et selon ces endroits ils seront plus au moins prioritaires, et on obtient donc une cascade de styles (**Cascading** Style Sheets):

1. directement dans les balises de la page web (fichier HTML) (dans l'élément qui va utiliser le style), on utilise l'attribut **style**, donc dans le corps de la page `<body>` (la moins recommandée). *Pas pour la partie pratique du partiel*
2. dans l'en-tête `<head>` de la page web (fichier HTML) . on utilise la balise `<style>` ; *Pas pour la partie pratique du partiel*
3. Déclaration dans un **fichier .css externe**, (méthode la plus recommandée) ; on utilise a balise `<link>` ;

1. Style Local : Déclaration des styles directement dans les balises html ("inline")

De façon ponctuelle, si l'on veut attribuer un style à un seul endroit, on peut déclarer le style à l'intérieur d'une balise html à l'aide de l'attribut **style**.

## PAR EXEMPLE :

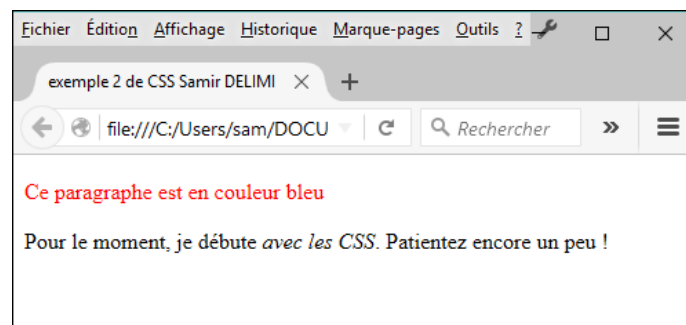
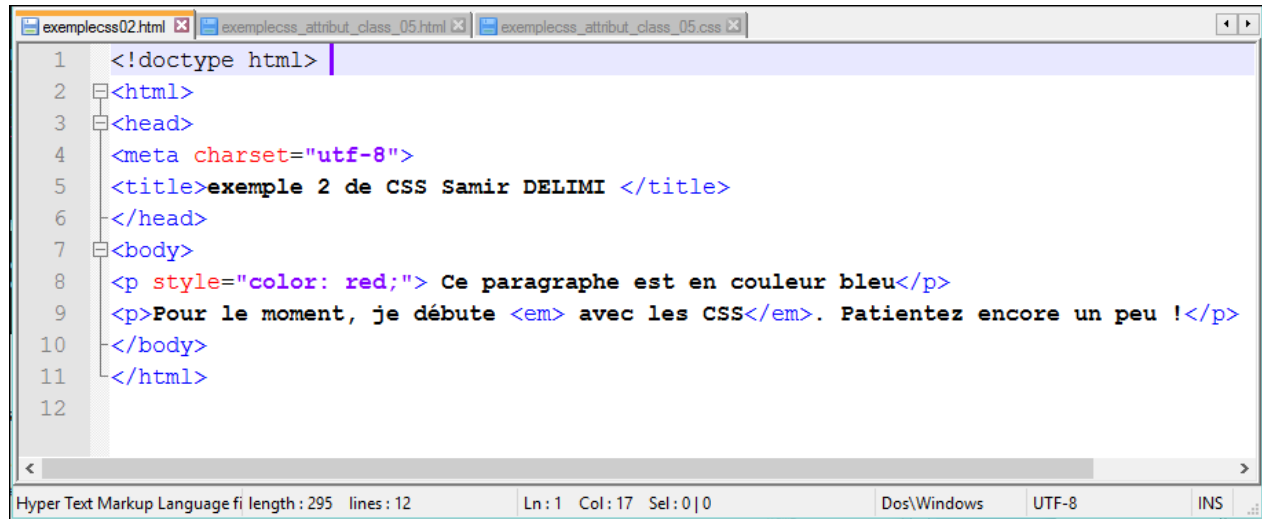
Vous pouvez donc, ajouter un attribut style à n'importe quelle balise. Vous insérerez votre code CSS directement dans cet attribut. Par exemple :

C'est la solution la moins recommandée maintenant.

## EXERCICE 1 :

```
<p style="{color : red ; font-size : 20px;}" > Ceci est un paragraphe </p>
```

On veut que le texte du document html soit de couleur rouge en utilisant le CSS dans le fichier HTML



## 2. Style Intégré : Déclaration des styles dans l'en-tête ( head) de la page web (fichier HTML) à l'aide de la balise <style>

C'est la deuxième méthode d'utilisation des styles, elle consiste à insérer le code CSS directement dans une balise **<style>** à l'intérieur de l'en-tête **<head>**. Par exemple :

```
<style> p {color : red; font-size : 20px; } </style>
```

## EXERCICE 2 :

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <!-- exemple de style intégré dans le head-->
6     <style>
7       p
8       {
9         color: blue;
10      }
11    </style>
12
13    <title> Exemple 2 CSS Samir DELIMI</title>
14  </head>
15
16  <body>
17    <h1>My Chikoula site</h1>
18
19    <p>Bonjour et bienvenue sur mon site !</p>
20    <p>Pour le moment, mon site est un peu <em>vide</em>. Patientez encore un peu !</p>
21  </body>
22 </html>
```

3. S  
t  
y  
l  
e  
E  
x  
t  
e  
r  
n  
e  
:  
.css

Finalement on peut  
.css externe, on

dans un fichier

les déclarer un fichier  
utilise une balise



```
<link href="fichiercss.css" rel="stylesheet"/>
```

<link> pour faire le  
et le fichier .CSS.

lien entre le fichier .html

Cette dernière méthode est la plus recommandée, pour plusieurs raisons :

- Séparation du format et du contenu et la structure
- Allège les pages web
- Réutilisation des feuilles de styles pour toutes les pages du site web
- Etc...

C'est la 3ème méthode que nous utiliserons dans notre TD (Fichier de style externe).

Nous allons donc détailler cette dernière solution, en créant nos règles de styles dans un fichier style externe.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="utf-8" />
5     <link rel="stylesheet" href="style.css" />
6     <title>Exercice fichierCSS</title>
7   </head>
8   <body>
9     <h1>Exercice css</h1>
10  </body>
```



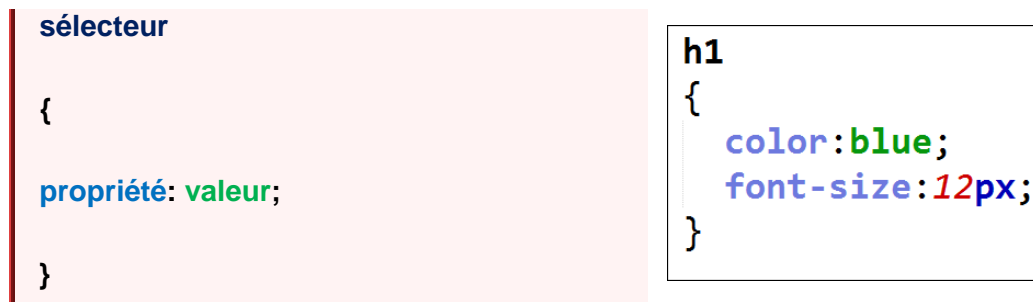


Les principes de la Syntaxe d'une règle de style CSS

Une règle (plus communément appelé un style) se compose d'un sélecteur et d'une déclaration de propriétés.

Elle s'écrit comme ceci :

**sélecteur** {**propriété**: **valeur**; }



La **structure** et la syntaxe d'une règle de style CSS est **toujours la même** et elle est assez simple :

On trouve trois éléments différents :

Le **sélecteur**, la **propriété** et la **valeur de la propriété**.

Le **sélecteur** est la cible de la modification :

On cible le **Sélecteur** (**élément** ou **une famille d'éléments** du document HTML et on lui associe une **déclaration de règles de style** ou plusieurs.

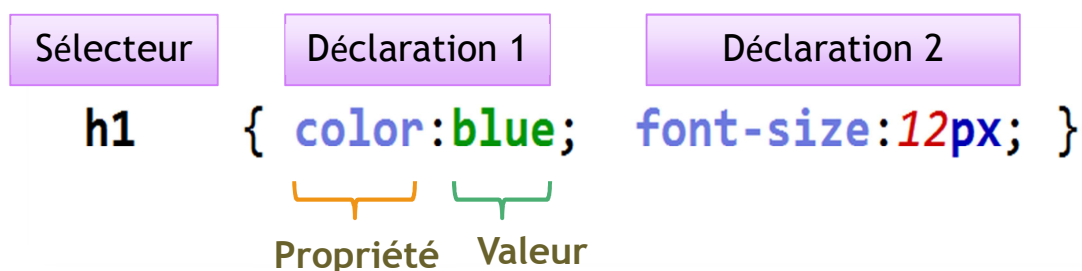
Pour chaque **déclaration de règle de style**, on met une propriété et la valeur prise par la propriété.

Les règles de style sont écrites dans un fichier css, qui aura comme extension.css ou en interne, dans la page web elle-même.

La première solution est la plus efficace. Ce fichier CSS ne contient que les règles CSS et rien d'autre.

Une fois ce fichier créé et renseigné, vous devez le lier aux pages HTML voulues, celles dont les éléments HTML doivent utiliser les styles définis.

La syntaxe en CSS est très précise, il faut toujours la respecter. En cas d'erreur les styles ne seront pas pris en compte. Attention à la ponctuation.

EXEMPLE DE REGLE CONTENU DANS UN FICHIER CSS :

Le sélecteur peut donc être une balise html, comme dans notre exemple la balise h1, La valeur correspond à la valeur que nous souhaitons modifier. Par exemple, la propriété la couleur d'affichage ou la taille de la police.

■ **Le sélecteur, c'est :**

- **Le nom de la balise html à modifier, L'identifiant (id) ou la classe (class):**

On écrit les noms des balises html dont on veut modifier l'apparence (body, p ; h1 ; etc.).

Nous verrons les identifiants et les classes un peu plus bas dans le document.

Par exemple :

Si je souhaite *modifier l'apparence de tous les paragraphes <p>*, je dois écrire **p**.

■ **La propriété CSS :**

Tous les « effets de style » de la page sont rangés dans des propriétés.

Il y a par exemple :

- la propriété *color* qui permet d'indiquer la couleur du texte,
- *font-size* qui permet d'indiquer la taille du texte,
- *background, margin, etc...*

Il y a beaucoup de propriétés CSS et, bien évidemment, nous ne sommes obligés à les connaître toutes par cœur. Je vous présente les plus utilisées dans un document annexe.

- **Les valeurs :** pour chaque propriété CSS, on doit indiquer une valeur, c'est la caractéristique de la propriété.

## **EXEMPLE :**

Modifier la couleur de la page web en bleu.

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

```
h1
{
  color:blue;
  font-size:12px;
}
```

## DES COMMENTAIRES DANS DU CSS

Comme en HTML, les commentaires ne seront pas affichés par les navigateurs, ils servent simplement à indiquer des informations pour vous, par exemple pour vous y retrouver dans un long fichier CSS.

Format d'un commentaire, Tapez

***/\* suivi de votre commentaire, puis \*/***

Vos commentaires peuvent être écrits sur une ou plusieurs lignes. Par exemple :

### CSS: STYLE PAR DEFAUT

C'est Comme Word avec ?

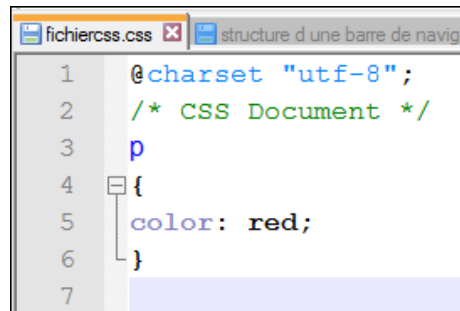
- **Titre1, Titre2 etc..**
- Les CSS ont leurs styles par défaut: **h1, h2, P, em**, etc...



## PREMIER FICHIER OU FEUILLE DE STYLE DE BASE :

La feuille de style est un fichier .css externe,

on utilise la balise `<link>`, présentée ci-dessus, pour faire le lien entre le fichier .html et le fichier .CSS:



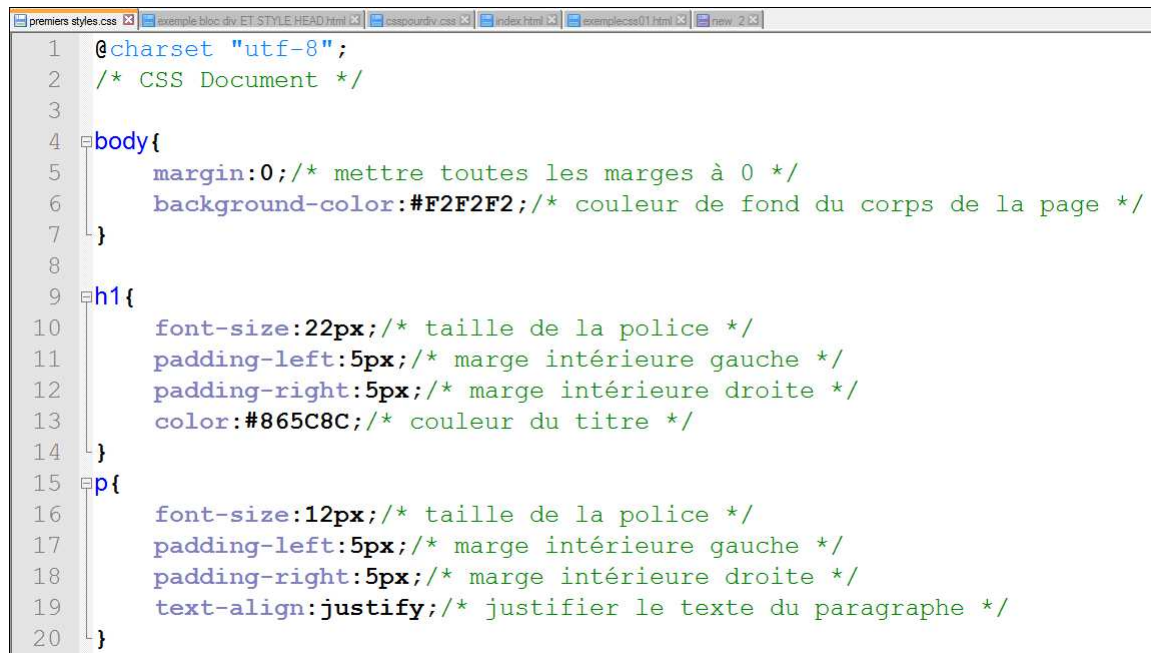
### @charset "UTF-8";

La première ligne appelée la "at-rule" @charset, est la déclaration du codage de caractère utilisé dans le fichier de feuille de style CSS, ici on déclarera toujours que votre fichier CSS est codé en UTF-8.

### P {color : red }

Indique qu'on souhaite mettre la couleur rouge à tous les paragraphes.

## EXEMPLE :



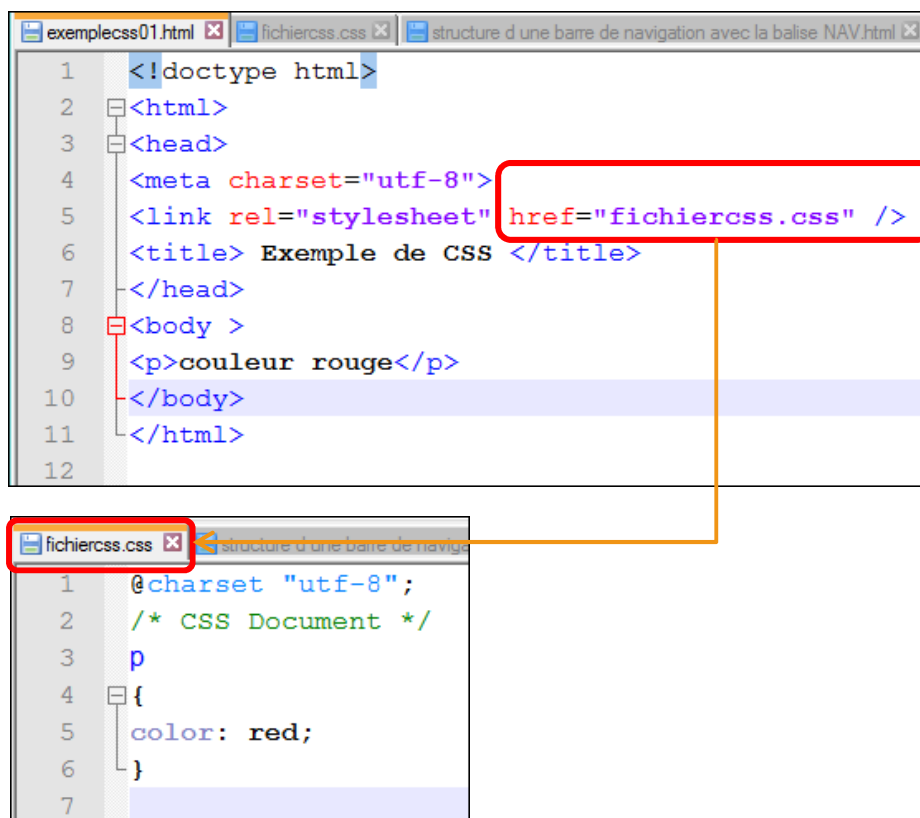
## Les styles en cascade:

Les propriétés CSS bénéficient du système d'héritage de la programmation objet.

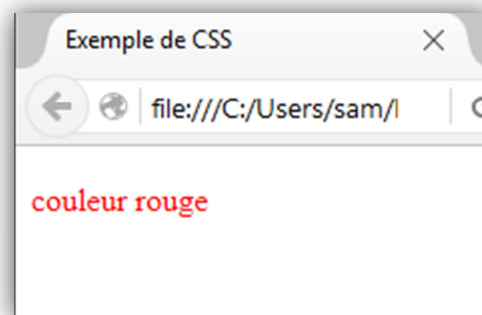
Par exemple si vous modifiez la couleur du texte pour dans la balise body, tous les textes qu'elle contient seront modifiés.

## EXERCICE 2 :

On veut que le texte des paragraphes <p> du document html soit de couleur rouge



Fichiercss.css



On obtient le Résultat suivant:

Modifier l'exemple en rajoutant les propriétés suivantes :

```
font-family: "trebuchet ms"; /* famille de police "trebuchet ms" */
font-size: 22px; /* taille de police de 22 pixels */
color: #252525; /* couleur de police noir (#252525=noir) */
text-align: center; /* aligner le texte à gauche */
margin-bottom: 20px; /* ajouter une marge sous le texte de 20 pixels
```

### EXERCICE 3:

on veut écrire le paragraphe 1 en bleu et le paragraphe 2 en rouge

On obtient :

```
exemplecss03.html x fichiercss.css x structure d une barre de navigation avec la balise NAV.html x
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <link rel="stylesheet" href="exemplecss03.css" />
6 <title> Exemple 3 de CSS </title>
7 </head>
8 <body>
9 <h1>Couleur bleu</h1>
10 <h2>couleur rouge</h2>
11 </body>
12 </html>
13
```

```
exemplecss03.html x exemplecss03.css x
1 @charset "utf-8";
2 /* CSS Document */
3 h1
4 {
5 color: blue;
6 }
7 h2
8 {
9 color: red;
10 }
11
```



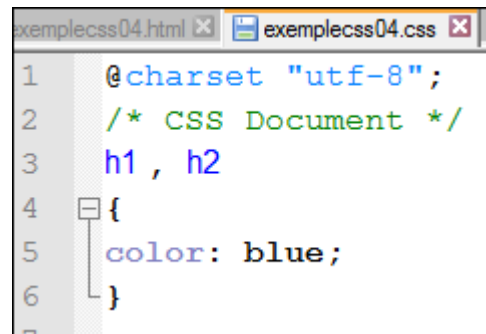
## APPLIQUER UN STYLE A PLUSIEURS BALISES

Dans le cas où plusieurs balises ont la même propriété. Il suffit de séparer les balises par une virgule.

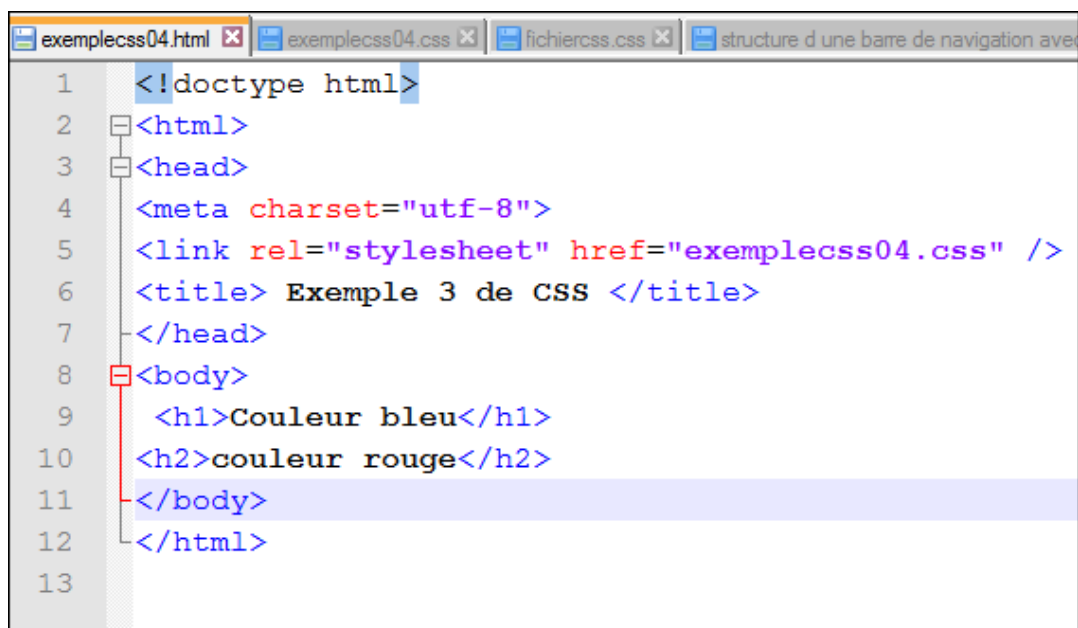
### EXERCICE 4 :

```
@charset "utf-8";  
/* CSS Document */
```

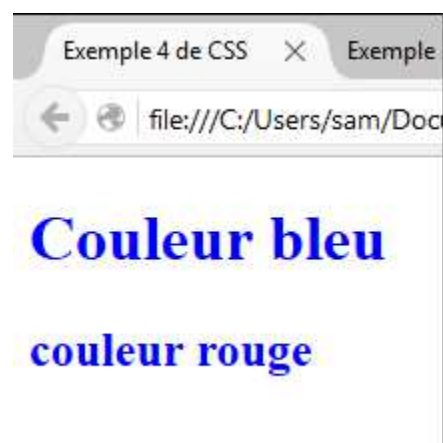
```
h1,h2  
{  
color: blue;  
}
```



```
exemplecss04.html x exemplecss04.css x  
1 @charset "utf-8";  
2 /* CSS Document */  
3 h1 , h2  
4 {  
5 color: blue;  
6 }
```



```
exemplecss04.html x exemplecss04.css x fichiercss.css x structure d une barre de navigation avec  
1 <!doctype html>  
2 <html>  
3 <head>  
4 <meta charset="utf-8">  
5 <link rel="stylesheet" href="exemplecss04.css" />  
6 <title> Exemple 3 de CSS </title>  
7 </head>  
8 <body>  
9 <h1>Couleur bleu</h1>  
10 <h2>couleur rouge</h2>  
11 </body>  
12 </html>  
13
```



## LES BLOCS "BOITES" OU CONTENEURS :

La notion de boîte est importante pour comprendre les balises Html et les styles CSS à leurs appliquer.

La balise <Body> c'est une "boîte" qui contient toutes les autres.

Un paragraphe <p> est une boîte, et elle est elle-même placée dans body.

Dans les paragraphes il y a des liens (placés dans des boîtes), des sous-listes, ...

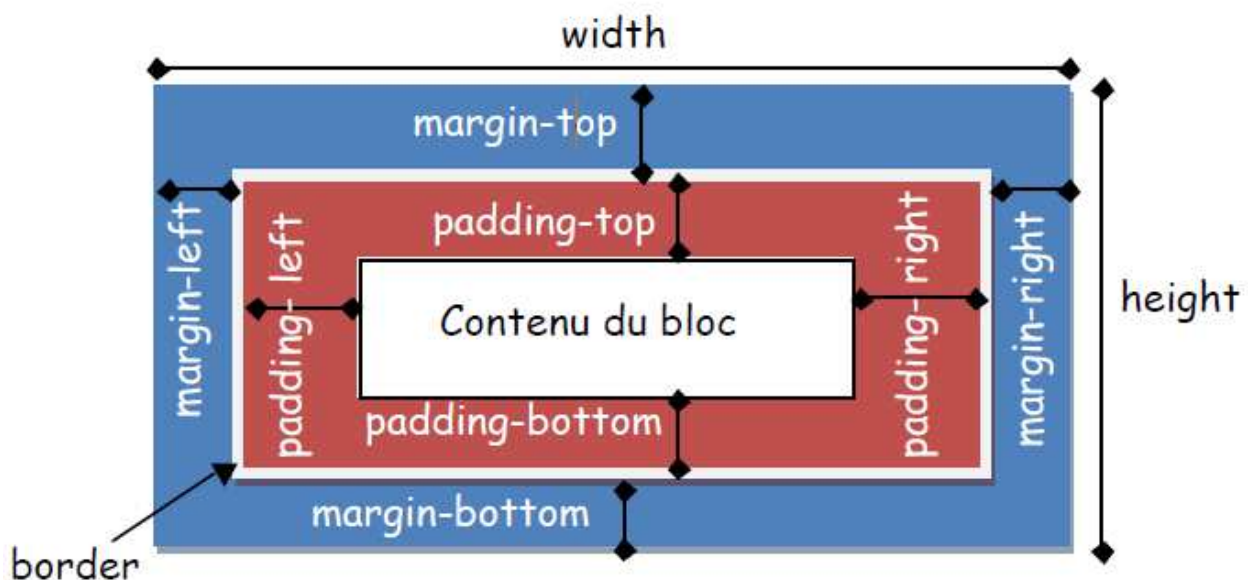
Tout ça dans des boîtes contenues dans des boîtes ... Il y a aussi des tableaux, des images, ...

Bref tout ce contenu est géré par votre navigateur avec des boîtes, comme des poupées russes mais c'est plus facile à gérer.

C'est pour ça qu'il y a les propriétés margin-left, margin-top, margin-bottom et margin-right et idem avec padding-top, ... : les margin indiquent les 4 marges extérieures à votre boîte, par rapport aux autres boîtes environnantes, tandis que les padding indiquent la distance séparant le contenu d'une boîte au contour de la boîte.

Les border définissent l'apparence de la bordure de la boîte. L'idée de tout mettre dans des boîtes est pratique puisque chaque boîte a ses petites propriétés (background-color, ...) et c'est pour ça que les CSS sont les initiales de Cascading Style Sheets : les propriétés d'une grosse boîte sont données à toutes les boîtes contenues dans cette grosse boîte, sauf si les propriétés des petites boîtes sont plus précises.

### Propriétés des Blocs



### APPLIQUER UN STYLE AVEC LES SELECTEURS SPECIFIQUES: CLASS ET ID :

Tout ce que l'on a fait jusqu'à maintenant implique que si on modifie le style d'un paragraphe par exemple, tous les paragraphes de toute la page web seront modifiés et auront la même présentation.

Comment faire pour que certains paragraphes seulement soient écrits de manières différentes ?

C'est-à-dire, comment faire pour ne formater qu'une balise particulière parmi d'autres, dans une même page web ?

### Solution :



Pour ne formater qu'une balise, nous utiliserons un des attributs suivants :

- l'attribut **class** ;
- l'attribut **id**.

Ces attributs sont utilisés pour n'importe quelle balise, aussi bien titre que paragraphe, image, etc.

Les sélecteurs class et id sont incontournables et indispensables. Voyons leurs similitudes et leurs différences par deux exemples.

### 1. L'attribut class :

On doit écrire un nom qui sert à identifier la balise. Ce que vous voulez, du moment que le nom commence par une lettre.

Si vous utilisez des id, lorsque vous définirez leurs propriétés dans le fichier CSS, il faudra faire précéder le nom de l'id par un **point** •

## EXERCICE 5 : (ATTRIBUT CLASS) :

Sur la page html, on écrit :

```
!doctype html>
<html>
<head>
<meta charset="utf-8">
<link rel="stylesheet" href="fichiercss.css" />
<title>Document sans nom</title>
</head>
<body>
<h1> explication des classes et identifiants</h1>
<h1 class="bleu"> Couleur bleu</h1>
<h1 class="rouge">couleur rouge</h1>
</body>
</html>
```

Sur la page CSS on écrit :

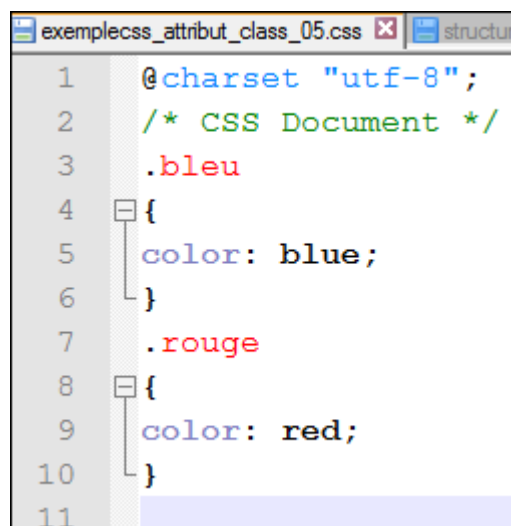
```
@charset "utf-8";
/* CSS Document */

.bleu
{
color: blue;
}

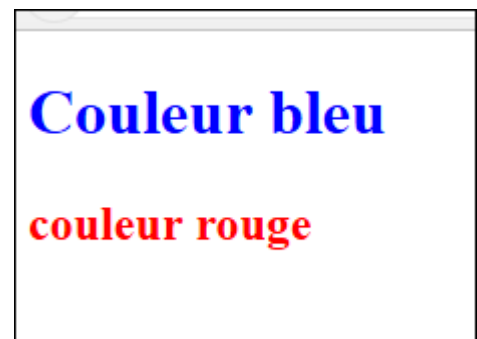
.rouge
{
color: red;
}
```



```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <link rel="stylesheet" href="exemplecss_attribut_class_05.css" />
6 <title>Document sans nom</title>
7 </head>
8 <body>
9 <h1 class="bleu"> Couleur bleu</h1>
10 <h2 class="rouge">couleur rouge</h2>
11 </body>
12 </html>
13
```



```
1 @charset "utf-8";
2 /* CSS Document */
3 .bleu
4 {
5 color: blue;
6 }
7 .rouge
8 {
9 color: red;
10 }
11
```



## 2. L'attribut ID :

Il fonctionne exactement de la même manière que class, à un détail près : il ne peut être utilisé qu'une seule fois dans le code. Si vous utilisez des id, lorsque vous définirez leurs propriétés dans le fichier CSS, il faudra faire précéder le nom de l'id par un dièse (#) :

### EXERCICE 6 :

```
#rouge
{
color: red;
}
```

## LES BALISES (ELEMENTS) DIV ET SPAN

Il arrivera parfois que vous ayez besoin d'appliquer une class (ou un id) à certains mots qui, à l'origine, ne sont pas entourés par des balises.

En effet, le problème de class, c'est qu'il s'agit d'un attribut. Vous ne pouvez donc en mettre que sur une balise. Si, par exemple, je veux modifier uniquement « bienvenue » dans le paragraphe suivant :

**1<p>Bonjour et bienvenue sur le site Licence 2 Droit !</p>**

Cela serait facile à faire s'il y avait une balise autour de « bienvenue » mais, malheureusement il n'y en a pas. En fait, on a inventé deux balises dites universelles, qui n'ont aucune signification particulière (elles n'indiquent pas que le mot est important, par exemple). Il y a une différence minime (mais significative !) entre ces deux balises :

- **<span> </span>** : c'est une balise de type inline, c'est-à-dire une balise que l'on place au sein d'un paragraphe de texte, pour sélectionner certains mots uniquement. Les balises <strong> et <em> sont de la même famille. Cette balise s'utilise donc au milieu d'un paragraphe et c'est celle dont nous allons nous servir pour colorer « bienvenue ».
- **<div> </div>** : c'est une balise de type block, comme une boîte, qui entoure un bloc de texte. Les balises <p>, <h1>, etc. sont de la même famille. Ces balises ont quelque chose en commun : elles créent un nouveau « bloc » dans la page et provoquent donc obligatoirement un retour à la ligne. <div> est une balise fréquemment utilisée dans la construction d'un design (tableau).

### EXERCICE 7 : UTILISATION DE LA BALISE SPAN :

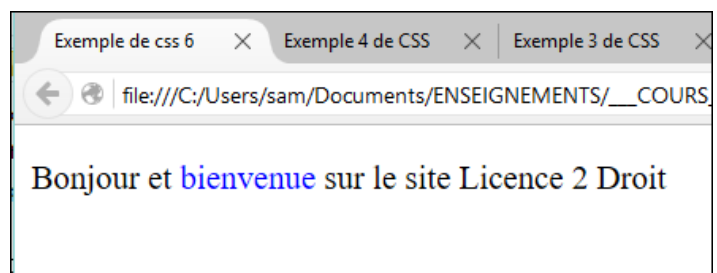
Nous utilisons dans l'exemple qui suit la balise <span>. On la met autour de « bienvenue », on lui ajoute une classe (du nom qu'on veut), on crée le CSS et c'est gagné !

```
<!doctype html>
<html>
<head>
<meta charset="utf-8">
<link rel="stylesheet"
href="fichiercss.css" />
<title>Document sans nom</title>
</head>
<body>
<p>Bonjour et <span
class="bleu">bienvenue</span> sur
le site Licence 2 Droit</p>
</body>
</html>
```

```
@charset "utf-8";
/* CSS Document */
.bleu
{
color: blue;
}
.rouge
{
color: red;
}
```

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <link rel="stylesheet" href="exemplecss_balise_span_07.css" />
6 <title> Exemple de css 6 </title>
7 </head>
8 <body>
9 <p>Bonjour et <span class="bleu">bienvenue</span> sur le site Licence 2 Droit</p>
10 </body>
11 </html>
12
```

```
1 @charset "utf-8";
2 /* CSS Document */
3 .bleu
4 {
5     color: blue;
6 }
7 .rouge
8 {
9     color: red;
10 }
```



## LA BALISE <DIV> :

Nous utilisons la balise **<div>** pour diviser notre page web en blocs ou zones spécifiques à des portions particulières de notre page (en-tête ou pied de page par exemple),

On peut ainsi définir une mise en forme sur une portion de page sans qu'elle ait une fonction particulière.

## EXEMPLE :

Nous allons à présent structurer notre page en la décomposant à l'aide de plusieurs balises **<div></div>** . Celles-ci vont nous permettre de diviser notre document HTML. Pour ce faire, nous avons besoin de trois grands blocs dans la partie **<body>** :

1. l'en-tête,
2. le contenu
3. et le pied de page.

Nous allons leur donner chacun un identifiant unique afin de les différencier.

**<body>**

```
<div id="entete"> </div>
<div id="contenu"> </div>
<div id="pied de page"> </div>
</body>
```

### Structurer un page avec la balise DIV :

#### EXEMPLE :

Nous continuons ce travail en rentrant plus en détail dans l'arborescence de notre structure. L'en-tête sera composé de deux parties :

Le logo et le menu. La partie centrale du contenu se divise également en deux blocs : le texte et les images. Par contre, le pied de page restera tel quel car il n'aura pas besoin d'autre division.

```
<body>
  <div id="entete" >
    <div id="logo"></div>
    <div id="menu"></div>
  </div>
  <div id="contenu">
    <div id="texte"></div>
    <div id="images"></div>
  </div>
  <div id="pied de page"></div>
</body>
```

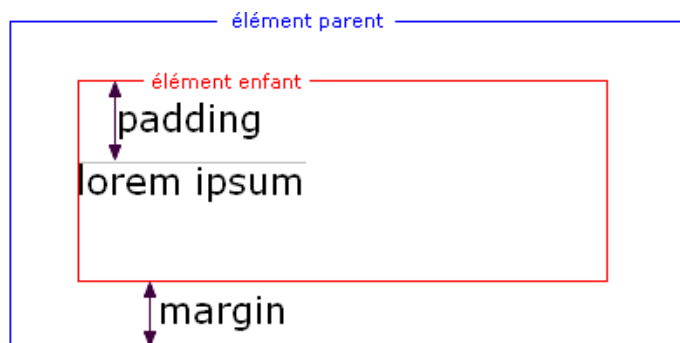
**Un exemple complet d'utilisation de la balise DIV est présenté dans le cas pratique résolu à la fin du document.**

*Nous pouvons évidemment le faire avec les nouvelles balises html5, ce qui nous facilitera le travail, mais ce n'est pas l'objet de cette présentation!!!*

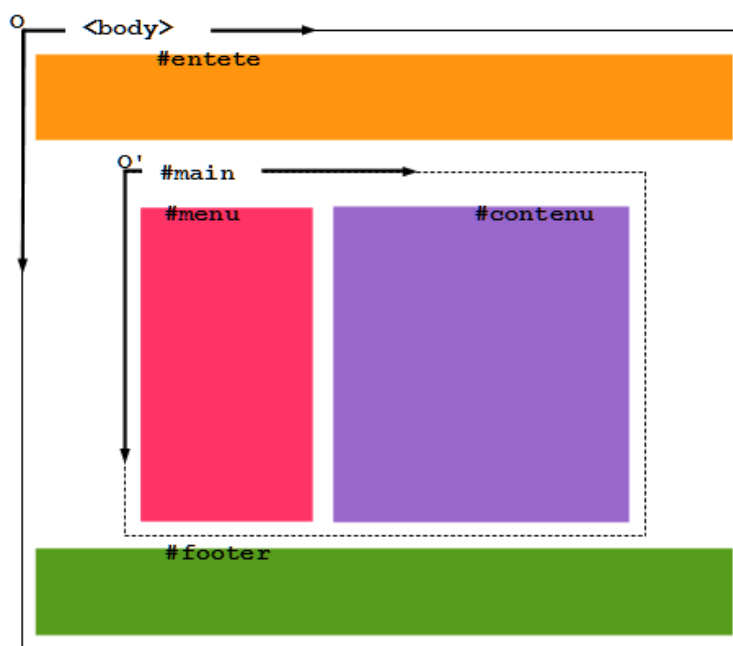
## GERER LES MARGES EN CSS

Définitions des marges intérieures et extérieures

Pour chaque élément html on peut donc définir l'espace qui le séparera des autres éléments (**margin**) et les espacements intérieurs dont il peut bénéficier (**padding**).



## MISE EN PAGE FULL CSS



## BORDURES EN CSS

La propriété permettant d'obtenir une bordure en CSS est tout simplement **<border>**. A cet attribut, on pourra associer une épaisseur de bordure (border-width), un style (border-style) et une couleur (border-color).

Propriétés utilisées :

- border-width
- border-style

- border-color
- padding

## IMAGE DE FOND EN CSS :

Mettre une image de fond dans une page html est un exercice très répandu. Et il est loin et révolu le temps où on utilisait le code du type : `<body background="images/image_de_fond.png">` si peu adaptable aux besoins.

Avec les feuilles de style on peut obtenir beaucoup plus, surtout depuis l'arrivée des CSS3 qui permettent de mettre plusieurs images de fond d'un seul coup.

### Attributs utilisés :

- background-image
- background-repeat
- background-position

## Code CSS de base

L'image de fond se déclare en général pour la balise **body** (corps de page), mais on peut aussi l'appliquer à un bloc (`<div></div>`), ou un titre (`<h1></h1>`), une citation (`<blockquote></blockquote>`), etc.

Pour le corps de page, la syntaxe CSS de base sera celle-ci :

```
body {  
  color:black;  
  background-color:white;  
  background-image:url(images/image_de_fond.png);  
}
```

- **color** renseigne la couleur de la police de caractère
- **background-color** est de la couleur de fond de page ;
- **background-image:url()** est bien sûr l'adresse de l'image choisie pour le fond de page.

## Fond de page fixe

Pour que le fond de page devienne fixe, on rajoutera dans le code :

```
background-attachment:fixed;
```

## POSITIONNEMENT AU CENTRE

Pour une position au centre de la page, cela donnera :

```
background-position:center center;
```

ou :

```
background-position:50% 50%;
```



ALIGNER UNE IMAGE ET DU TEXTE À L'AIDE DE LA PROPRIÉTÉ : FLOAT

EFFETS ROLLOVER À L'AIDE DE :HOVER

PUCES EN IMAGES

CITATIONS ET RETRAITS

## LES OUTILS CSS :

---

### GENERATEUR DE STYLES :

Regroupe divers générateurs de code CSS pour faire des coins arrondis, des ombres ou des lueurs.

Convertisseur d'unités de longueurs pour vous permettre de trouver les équivalences entre PX, EM, % et PT.

Des cours et générateurs de code CSS pour les transitions, transformations et animations.

Ces outils ne rentrent pas dans le cadre de notre initiation.

Si vous désirez générer simplement une feuille de style un peu plus fournie, un [générateur de feuille de style est à disposition](#).

## Partie 3

### Initiation aux langages HTML et CSS

#### EXEMPLE PRATIQUE COMPLET



### Support Pratique



## EXERCICE :

- **Travail à Faire :** Reproduire la page html suivante :
- Objectif : créer un modèle de page Web en HTML et CSS qui s'adapte automatiquement à la taille du navigateur et de l'écran, tout cela sans utiliser de tableaux.

### Création de pages Web (documents HTML) avec les CSS:

### Exercice Corrigé HTML, CSS et la page structurée avec les DIV

RESULTAT ATTENDU :

DU CMSI

### D.U. Création et Maintenance de Sites Internet (CMSI)

**Objectifs de la formation:**

Former des créateurs de sites Web dotés d'un savoir-faire dans les domaines de l'utilisation des outils de développement web, de la gestion de projet, du Webmarketing et du Webdesign pour la stratégie numérique des entreprises.

**Conditions d'accès:**

Cette formation est ouverte aux étudiants en formation initiale et aux professionnels en formation continue. L'admission des candidats se fait sur dossier. Le projet professionnel est un critère prépondérant.

**Enseignements**

La formation s'organise en séminaires de 1 jour/semaine, le lundi ou le jeudi après-midi, sur une période de 6 mois, pour un volume horaire total de 131 heures. Elle s'articule autour de trois grandes parties de modules, chacune correspond à une Unité d'Enseignement:

**UE1 – Modules de Gestion:**

La partie gestion comprend les modules liés à la gestion de projet dont l'objectif est de permettre aux étudiants de maîtriser les fonctions de support liées à la création d'un site internet

**UE2 – Modules Pratiques**

La partie pratique est composée de modules techniques donnant aux étudiants toutes les ressources nécessaires pour le développement de sites modernes.

**UE3 – Mémoire : Création de site web dynamique de A à Z avec suivi personnalisé.**

Responsable de la formation: Samir DELIMI

lien vers le site: Cliquez [ici](#)

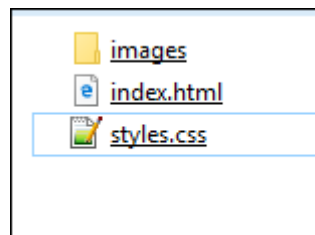
Envoyez un mail au Responsable: Cliquez [ici](#)

© Diplôme d'Université Création et Maintenance de Sites Internet

La page web : index.html

Les fichiers des styles : style.css

Un dossier image qui contient les images



## FICHER HTML : INDEX.HTML

```
<!DOCTYPE html>
<html>

  <!-- En-tête de la page -->
  <head>

    <meta charset=utf-8" />

    <!--Titre affiché dans la barre de titre du navigateur-->
    <title>Ma première page</title>

    <!--Appel du fichier css rattaché à la page html-->

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

  </head>
  <!-- Fin de l'en-tête de la page -->

  <!-- Corps de la page -->
  <body>

    <!-- le header contient en général le logo et le nom du site -->
    <div id="header">
      <div id="logo"></div>
      <div id="title_header"> DU CMSI</div>
    </div>
    <!-- fin du header -->
```

```

<!-- Content -->
<div id="content">
    <h1>D.U. Création et Maintenance de Sites Internet (CMSI)</h1>

    <h2>Objectifs de la formation: </h2>

    <p> Former des créateurs de sites Web dotés d'un savoir-faire dans les
domaines de
    l'utilisation des outils de développement web, de la gestion de projet,
    du Webmarketing et du Webdesign pour la stratégie numérique des
entreprises.
    </p>

    <h2>Conditions d'accès:</h2>

    <p> Cette formation est ouverte aux étudiants en formation initiale et aux
professionnels en formation continue. L'admission des candidats se fait sur
dossier.
    Le projet professionnel est un critère prépondérant.
    </p>

    <h2>Enseignements</h2>

    <p>
        La formation s'organise en séminaires de 1 jour/semaine, le lundi ou le
jeudi après-midi, sur une période de 6 mois, pour un volume horaire total de 131
heures.

        Elle s'articule autour de trois grandes parties de modules, chacune
correspond à une Unité d'Enseignement:
    </p>

    <h3>UE1 - Modules de Gestion: </h3>
    <p>
        La partie gestion comprend les modules liés à la gestion de projet dont
l'objectif
        est de permettre aux étudiants de maîtriser les fonctions de support liées
à la
        création d'un site internet
    </p>
    <h3>UE2 - Modules Pratiques </h3>
    <p>
        La partie pratique est composée de modules techniques donnant aux étudiants
toutes les ressources nécessaires pour le développement de sites modernes.
    </p>
    <h3> UE3 - Mémoire : Création de site web dynamique de A à Z avec suivi
personnalisé.</h3>
    <p>
        Responsable de la formation: Samir DELIMI
    </p>
    <p>
        lien vers le site: Cliquez <a href="http://www.google.fr"
target="_blank">ici</a><br />
        Envoyez un mail au Responsable: Cliquez <a
href="mailto:samir.delimi@umontpellier.fr">ici</a>
    </p>
</div>
<!-- Fin du content -->

```

```
<!-- Pied de page -->
<div id="footer">
  <div id="footer_text">&copy; Diplôme d'Université Création et Maintenance de
Sites Internet</div>
</div>
<!-- Fin de pied de page -->
</body>
<!-- Fin du corps de la page -->

</html>
```



# FICHIER CSS : STYLE.CSS

```
@charset "utf-8";
/* CSS Document */

body{
    margin:0;/* mettre toutes les marges à 0 */
    background-color:#F2F2F2;/* couleur de fond du corps de la page */
}

h1{
    font-size:22px;/* taille de la police */
    padding-left:5px;/* marge intérieure gauche */
    padding-right:5px;/* marge intérieure droite */
    color:#865C8C;/* couleur du titre */
}

h2{
    font-size:18px;/* taille de la police */
    padding-left:5px;/* marge intérieure gauche */
    padding-right:5px;/* marge intérieure droite */
    color:#865C8C;/* couleur du titre */
}

h3{
    font-size:14px;/* taille de la police */
    padding-left:5px;/* marge intérieure gauche */
    padding-right:5px;/* marge intérieure droite */
    color:#865C8C;/* couleur du titre */
}

#header{
    width:100%;/* largeur du header qui prend tout l'écran */
    height:130px;/* hauteur du header */
    background-image: url("images/header.jpg");/* lien vers l'image de fond */
    background-repeat: repeat-x;/* l'image de fond se répète au niveau horizontal */
    padding-top:5px;/* marge intérieure du haut */
}

#logo{
    width: 110px;/* largeur du logo du site */
    height:110px;/* hauteur du logo du site */
    background-image: url("images/logo.png");/* lien vers l'image de fond */
    background-repeat:no-repeat;/* pas de répétition de l'image */
    margin-top:7px;/* marge extérieure du haut */
    margin-left:22%;/* marge extérieure gauche */
}
```

```

#title_header{
    font-family:"trebuchet ms", sans-serif;/* police du titre dans le header */
    font-size:30px;/* taille de la police */
    letter-spacing:10px;/* espacement entre les caracteres */
    font-weight:bold;/* graisse de police */
    text-shadow: 1px 1px 1px #000000;/* ombre */
    color:#743393;
    margin-top:-75px;/* marge extérieure du haut */
    margin-left:40%;/* marge extérieure gauche */
}

#content{
    width: 800px;/* largeur du content */
    height:auto;/* hauteur du content */
    min-height:200px;/* la hauteur du content doit être au minimum de 200px */
    font-family:Verdana, sans-serif;/* police */
    background-color:#FBFBFB;/* couleur de fond du content */
    margin-left:auto;/* marge extérieure gauche */
    margin-right:auto;/* marge extérieure droite */
    padding-top:5px;/* marge intérieure du haut */
    padding-bottom:5px;/* marge intérieure du bas */
}

p{
    font-size:12px;/* taille de la police */
    padding-left:5px;/* marge intérieure gauche */
    padding-right:5px;/* marge intérieure droite */
    text-align:justify;/* justifier le texte du paragraphe */
}

#footer{
    width: 100%;/* largeur du footer qui prend tout l'écran */
    height:100px;/* hauteur du footer */
    background-image: url("images/footer.jpg");/* lien vers l'image de fond */
    background-repeat: repeat-x;/* répétition de l'image au niveau horizontal*/
}

#footer_text{
    width:auto;/* largeur de la zone de texte dans le footer */
    font-family:Verdana, sans-serif;/* police */
    font-size:14px; /* taille du texte */
    margin-left:45%;/* marge extérieure gauche */
    padding-top:40px;/* marge intérieure du haut */
}

```

---

# ANNEXES :Résumé de la Syntaxe, Sélecteurs et styles des CSS

## INSERER SON OU SES STYLE(S)

```
<!DOCTYPE html>

<html>

  <head>

    <meta charset="utf-8" />

    <title>Bonjour...</title>

    <link rel="stylesheet" href="url" type="text/css">

    <style type="text/css">

      ...

    </style>

  </head>

  ...

</html>
```

## Syntaxe

Générale	sélecteur { propriété: valeur; }
Multiples sélecteurs	Sélecteur1, ..., sélecteurN { propriété: valeur; }
Propriétés "combinées"	sélecteur { propriété: valeur valeurN; }

## Sélecteurs simples

<b>*</b>	Tous les éléments
<b>élément</b>	Tous les éléments “élément” dans le HTML
<b>#id</b>	L’élément avec id=”monid”
<b>élément.maclasse</b>	Le ou les éléments “élément” avec class=”maclasse”
<b>.maclasse</b>	Idem, peu importe le type d’élément HTML
<b>élément1 élément2</b>	Tous les “élément2” contenu dans un élément “élément1”
<b>élément1 &gt; élément2</b>	Tous les “élément2” ayant pour parent “élément1”
<b>élément1 + élément2</b>	Tous les “élément2” placé juste après “élément1”
<b>élément1 ~ élément2</b>	Tous les “élément2” précédé par “élément1”

## Sélecteurs d’attributs

<b>[attribut]</b>	Tous les éléments avec l’attribut “attribut”
<b>[attribut=mavaleur]</b>	Tous les éléments dont la valeur de “attribut” vaut “mavaleur”
<b>[attribut*=mavaleur]</b>	Tous les éléments dont la valeur de “attribut” contient “mavaleur”
<b>[attribut^=mavaleur]</b>	Tous les éléments dont la valeur de “attribut” commence par “mavaleur”
<b>[attribut\$mavaleur]</b>	Tous les éléments dont la valeur de “attribut” finit par “mavaleur”

## Pseudo-classes structurelles

<b>élément:first-child</b>	Premier enfant de “élément”
<b>élément:last-child</b>	Dernier enfant de “élément”
<b>élément:nth-child(n)</b>	Tous les “élément” qui sont n-ieme enfant de son parent
<b>élément:nth-last-child(n)</b>	Tous les “élément” qui sont n-ieme enfant de son parent, en comptant à partir de la fin
<b>élément:first-of-type</b>	Tous les “éléments” premier enfant de type “élément” de son parent
<b>élément:last-of-type</b>	Tous les “éléments” dernier enfant de type “élément” de son parent
<b>élément:only-child</b>	Tous les “éléments” seul enfant de leur parent
<b>élément:only-of-type</b>	Tous les “éléments” seul enfant du type “élément” de leur parent
<b>élément:not(sélecteur)</b>	Tous les “éléments” qui ne correspondent pas au “sélecteur”

## Autres pseudo-classes

<b>élément:active, élément:focus, élément:hover</b>  <b>élément:link</b>  <b>élément:visited</b>  <b>élément:checked</b>	Tous les “élément” durant les actions de l'utilisateur  Tous les “élément” cible d'un hyperlien pas encore visité  Tous les “élément” cible d'un hyperlien déjà visité  Tous les “éléments” cochés (action de l'utilisateur)
--	--

### Pseudo-éléments

<b>élément:before</b>	Ajoute du contenu avant “élément”
<b>élément:after</b>	Ajoute du contenu après “élément”
<b>élément::first-line</b>	Première ligne formatée de chaque “élément”
<b>élément::first-letter</b>	Première lettre formatée de chaque “élément”

## STYLES

### Notion de préfixe

```

élément {
  -webkit-propriété: valeur; // Chrome, Safari, Android...
  -moz-propriété: valeur;   // Mozilla
  -ms-propriété: valeur;    // Microsoft (Internet Explorer)
  -o-propriété: valeur;     // Opéra
  propriété: valeur;       // Toujours finir par la déclaration
}

```

### Formatage de texte

<b>font-size</b>	Taille de la police
<b>font-family</b>	Famille de police
<b>font-style</b>	Style de police (italique...)
<b>font-weight</b>	Graisse de la police
<b>text-decoration</b>	Décoration du texte
<b>text-align</b>	Alignement du texte
<b>@font-face</b>	Définition d'une police personnalisée

### Positionnement

<b>display</b>	Affichage des éléments en ligne, en bloc ou flexible
<b>float</b>	Flottement des boîtes
<b>position: relative</b>	Position relative des éléments les uns par rapport aux autres
<b>position: absolute</b>	Positionnement absolue
<b>position: fixed</b>	Positionnement fixe
<b>width, height</b>	Largeur et hauteur des éléments
<b>margin, padding</b>	Marges extérieures et marges intérieures

## Couleurs

Nom	<b>white, black, green...</b>	Pour des couleurs simples, le nom explicite
Valeur hexadécimale	<b>#FFFFFF, #000000, #008000</b>	Valeur hexadécimale de la couleur
Valeur RGB	<b>rgb(255,255,255), rgb(0,0,0)...</b>	Valeur RGB
Valeur RGBA	<b>rgba(0,0,0,1), rgba(0,0,0,0.5)...</b>	Valeur RGB avec gestion de la couche alpha, pour modifier l'opacité

<b>color</b>	Couleur du texte
<b>border-color</b>	Couleur de la bordure d'un bloc
<b>background-color</b>	Couleur de fond d'un bloc

## Background

<b>background-color</b>	Couleur de fond d'un bloc
<b>background-image</b>	Lien d'une image à mettre en fond
<b>background-repeat</b>	Répétition du fond (X, Y ou none)
<b>background-position</b>	Position d'origine du fond (valeurs numériques ou littérales)
<b>background-size</b>	Taille de l'image de fond : valeurs numériques (X et Y) ou littérale (cover ou contain)
<b>background-image: gradient(...)</b>	Dégradé de couleurs, "gradient" définit le style de dégradé (linéaire, radial...)

## Transitions

<b>transition-property</b>	Précise les propriétés CSS à transformer
<b>transition-duration</b>	Précise la durée de la transition
<b>transition-timing-function</b>	Précise la fonction de transition à utiliser, le modèle d'interpolation (accélération, décélération...)
<b>transition-delay</b>	Précise le retard (ou l'avance) du départ de la transition

## Transformations 2D

<b>transform: fonction(valeur);</b>	Les fonctions de transformation principales sont : <ul style="list-style-type: none"><li>- Translation (translate)</li><li>- Mise à l'échelle (scale)</li><li>- Rotation (rotate)</li><li>- Inclinaison (skewX et skewY)</li><li>- Matrice (matrix)</li></ul>
<b>transform-origin: x y;</b>	Point d'origine de la transformation

## Animations : définition

<b>@keyframe monAnim</b>	Définit les règles de l'animation qui a pour nom "monAnim"
<b>@-webkit-keyframes monAnim</b>	Webkit nécessaire pour Chrome, Safari, Opera
<b>@keyframes monAnim {</b>  <b>from {propriété: valeur1; propriétéN: valeur1;}</b>  <b>to {propriété: valeur2; propriétéN: valeur2;}</b>  }	Deux possibilités pour définir les règles d'une animation : <ul style="list-style-type: none"><li>- from et to qui équivalent à 0% et 100%</li><li>- Des pourcentages afin de définir plusieurs étapes à l'animation</li></ul>
<b>@keyframes monAnim {</b>  <b>0% {propriété: valeur1; propriétéN:</b>	

```
valeur1;}
```

```
25% {propriété: valeur2; propriétéN:  
valeur2;}
```

```
50% {propriété: valeur3; propriétéN:  
valeur3;}
```

```
100% {propriété: valeur4; propriétéN:  
valeur4;}
```

```
}
```

## NOTIONS DE RESPONSIVE DESIGN : VIEWPORT

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Différence entre surface réelle et surface en “pixels CSS”

Définition de “width” et “initial-scale” dans la balise meta pour plus de compatibilité

Pour plus d’explications et de précisions

<http://www.alsacreations.com/article/lire/1490-comprendre-le-viewport-dans-le-web-mobile.html>

---

## REFERENCE DES ELEMENTS HTML COMPLETE : AU9 MARS 2016

<https://developer.mozilla.org/fr/docs/Web/HTML/Element>



# BIBLIOGRAPHIE :

HTML5 Une référence pour le développeur web ; Rodolphe Rimelé

[http://fr.wikipedia.org/wiki/Hypertext\\_Markup\\_Language](http://fr.wikipedia.org/wiki/Hypertext_Markup_Language)

<http://www.xul.fr/css/table.php>

Open Classroom

Alsacréations

[http://www.w3schools.com/tags/att\\_a\\_target.asp](http://www.w3schools.com/tags/att_a_target.asp)

<http://css.mammothland.net/feuille-de-style-css-debutant.php#feuille>

[https://developer.mozilla.org/fr/Apprendre/Comprendre\\_les\\_URL](https://developer.mozilla.org/fr/Apprendre/Comprendre_les_URL)

<https://openclassrooms.com/>

<https://developer.mozilla.org/fr/docs/Web/Guide/HTML>

<https://fr.wikibooks.org/wiki/>