Chapitre 2

Intégration des styles CSS

Objectifs du chapitre :

- Créer et structurer une feuille de style CSS en déterminant les sélecteurs et en utilisant les attributs
- Réaliser la liaison entre les feuilles de styles et les pages web en utilisant les techniques et méthodologies les plus pertinentes
- Formater (les bases) des pages HTML à l'aide du CSS : couleurs (pleine, nommées, transparentes), typographie, taille des caractères (absolues et relatives), bordures, border-radius, ombres …
- Les classes et les identifiants, les balises universelles

1. POURQUOI LE CSS ?

Facilité au niveau des modifications …

L'apprentissage du CSS vous permettra d'accélérer la création de vos pages web tout en vous donnant un meilleur contrôle de l'apparence du contenu.

Pour les webmasters et les chefs de projet Internet, il est essentiel de connaître les CSS car en changeant le contenu d'un seul fichier texte, vous pouvez modifier la mise en page et l'apparence d'un ou de plusieurs sites de toutes tailles.

Permettre plus de créativité dans la construction des sites

Les CSS permettent de palier aux insuffisances du langage Html et augmenter la créativité du Web, ils permettent un contrôle total de la mise en forme d'un document hypertexte.

Sites web plus légers …

Dans un document d'une certaine importance, il arrive fréquemment que l'on attribue à certains éléments des caractéristiques de mise en forme identiques. Avant le CSS, il fallait répéter toutes les propriétés à chaque fois, ceci ayant pour effet de rallonger et d'alourdir le fichier.

Avec les CSS, vous donnez un nom à un format de texte très détaillé et vous invoquez ce style grâce à une propriété très brève. Tous ces styles ne sont donc rédigés qu'une seule fois dans un unique fichier.

Séparation du contenu et de la mise en forme.

Cohésion de la présentation tout au long du site lorsque les pages utilisent les mêmes feuilles de style.

Remplacer les tableaux invisibles servant à la mise en page ("Tableless")

En utilisant les CSS, il est possible de positionner au pixel près du texte et/ou des images sans passer ni par les tableaux invisibles ni par des images d'un pixel.

Permettre des mises en page proportionnelles et responsives

Les CSS permettent des mises en page fixes mais également des mises en page proportionnelles (à la taille de l'écran de l'utilisateur), et peut même permettre un mélange des deux.

Lecture Web : http://www.csszengarden.com

2. LE CSS AUJOURD'HUI

Le **Cascading Style Sheets** (CSS), ou encore les feuilles de style en cascade, sont développés par le W3C (World Wide Web Consortium) **depuis 1995**, et deviennent **disponibles et compatibles sur la plupart des navigateurs à partir de 2000**. Les feuilles de style en cascade sont **implémentées par niveau** et non par version :

Le **CSS1** équivaut au 1er niveau du CSS qui fût publié en décembre 1996 avec une quarantaine de propriétés, il incluait des propriétés simples comme les polices, les couleurs et les marges.

Suit **CSS niveau 2** avec 70 nouveautés en 1998, il intégrait des concepts avancés tels le flottement et le positionnement, ainsi que de nouveaux sélecteurs comme les sélecteurs d'enfants, les sélecteurs de frères adjacents et les sélecteurs universels.

Le **CSS3** commence son développement en 1999 (parallèlement au CSS niveau 2 révision 1, qui lui, débute en 2001). Son développement est décomposé en modules qui peuvent être rendus officiels et implémentés indépendamment, chaque module ayant son propre niveau d'avancement et d'intégration auprès des navigateurs. L'essentiel des travaux autour

des CSS3 a été de standardiser des propriétés pour réaliser des traitements graphiques qui nécessitaient auparavant des astuces et l'apport de toutes les propriétés concernant la mise en page et le positionnement.

Le dernier niveau du **CSS : le niveau 4**, commence son écriture en 2010. Il apporte des nouveautés appréciables au niveau des sélecteurs (voir le chapitre de notre cours consacré à ce sujet), la gestion des césures et des images hautes définition notamment.

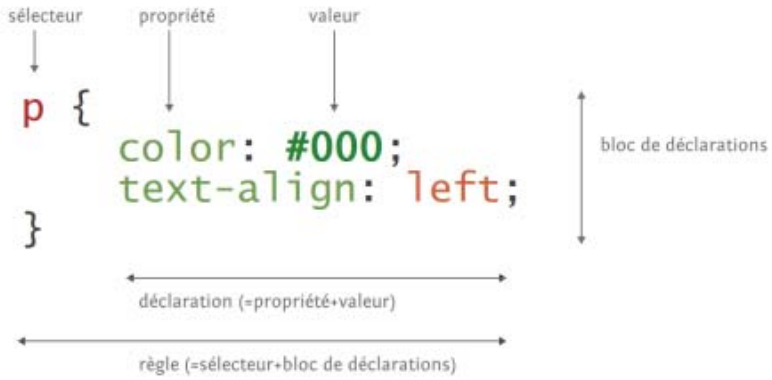
3. OUTILS NECESSAIRES POUR REDIGER DES FEUILLES DE STYLE ...

Pour rédiger des feuilles de style, vous avez besoin d'un éditeur de texte (comme pour l'HTML, le bloc-note suffit) et d'un navigateur internet classique. Il est possible d'intégrer vos CSS à des logiciels comme Dreamweaver mais il est plus facile de travailler avec un éditeur syntaxique.

4. LA SYNTAXE DE BASE DES FEUILLES DE STYLE

La plupart des langages informatiques sont structurés, c'est à dire que l'ordre dans lequel les commandes sont rédigées est vital. **Le HTML est structuré parce que chaque élément qu'il contient est traité et affiché dans l'ordre**.
Les feuilles de style ne sont pas structurées, elles **sont déclaratives** car tout le contenu des feuilles de style s'exprime par des règles, ces règles ne doivent pas être ordonnées dans un ordre particulier[13] mais sont rassemblées et quand un élément de style est invoqué, le navigateur lit la règle appropriée.
Pour créer une feuille de style, il faut donc définir ces règles et les pages web les respectent.
On parle de feuilles de style en **cascade** [*Cascading Style Sheets ou CSS*] car en cas de styles identiques, un **ordre de priorité** sera déterminé par le browser.
Précisons pour terminer que les feuilles de style ne sont pas une composante directe du langage Html mais un développement à part dans la publication de pages Web.

4.1. Déclaration d'un style



La définition de base d'un style est simple :

```
balise{propriété de style: valeur; propriété de style:valeur;}
```

Exemple :

```
h3 { font-family: arial; font-style: italic; }
```

Ici, la balise h3 sera en Arial et en italique, et dans votre document, toutes les balises <h3> auront comme style Arial et italique.

La déclaration de style sont entourées par des "{" .

[13] Sauf quelques rares exceptions.

La déclaration peut être divisée en deux parties, la propriété et la valeur. Ici, la propriété est `font-family` et la valeur `Arial`. Le couple "propriété de style/valeur" est séparé par un double-point (:) et chaque couple "propriété de style/valeur" est séparé par un point-virgule (;).
Les feuilles de style sont insensibles à la casse, il est toutefois préférable d'utiliser les minuscules.

4.2. Exemples de styles css :

Essayons de comprendre syntaxe et règles de style …

h1	**Titre de niveau 1**

{
color: #ff6600;
font-size: 25px;
font-weight: bold;
}
	Couleur orange
	Taille de police de 25 pixels
	Epaisseur grasse

h2	**Titres de niveau 2**

{
color: #996633;
font-style: italic;
font-size: 1em;
}
	Couleur brune
	Style italique
	Taille de police correspondant à 100% du choix de base de l'utilisateur

body	**Corps de la page**

{
background-color: #ffffcc;
font-family:
Arial,Helvetica,sans-serif;
}
	Couleur de fond jaune pâle
	Police pour le texte
	Arial en 1er choix –
	Si l'utilisateur ne la possède pas, Helvetica sera appliquée – et le cas échéant, n'importe laquelle des polices sans-serif

p	**Zone délimitée par la balise paragraphe**

{
margin-right : 250px;
margin-left : 250px;
text-align : right;
}
	Marge de droite de 250px
	Marge de gauche de 250px
	Alignement de texte à droite

Lecture Web :
Liste imprimables de propriétés de styles : http://www.smashingmagazine.com/2010/05/13/css-2-1-and-css-3-help-cheat-sheets-pdf/

Il n'y a pas de limite pour le nombre de couples "propriétés de style/valeur".
On peut attribuer plusieurs valeurs à une même propriété. Dans ce cas, on séparera les différentes valeurs par des virgules (,).

```
h3 {font-family: arial, helvetica, sans-serif}
```

On peut attribuer un même style à plusieurs balises (séparées par des virgules).

```
h1, h2, h3 {font-family: arial; font-style: italic}
```

L'espace entre propriétés de style et valeur n'est pas obligatoire mais aide à la lisibilité du code source, écrivez également vos styles sur plusieurs lignes comme ci-après :

```
h3
{
     font-family: Arial;
     font-style: italic;
     color: green;
}
```

Conclusion : Attribution d'un style

Au départ, nous disposons d'un document HTML classique

```
<html>
      <head>
      <title>  utilisation des feuilles de style   </title>
      </head>

      <body>
      <h1> le mécanisme des feuilles de style </h1>
      <p> l'attribution d'un style, … </p>
      </body>

</html>
```

Enregistré dans un dossier racine de site sous « votrefichier.html »

Auquel est associée une feuille de style

```
h1
{
      color : blue;
      text-align : center;
}

p
{
      font-family : verdana, arial, tahoma;
      color : red;
      text-align : justify;
}
```

Enregistrée dans le dossier racine de site sous « votrefichier.css »

Les commentaires en CSS

Les commentaires en CSS sont des instructions facultatives commençant par les caractères **/*** et se terminant par ***/**.

```
h1
{
      /*color : blue;*/
      text-align : center;
}
```

On peut les placer partout entre les règles, voire au sein des règles (mais pas entre une propriété et sa valeur), leur contenu n'ayant aucune influence sur le rendu. On ne peut pas les imbriquer.
Ils sont très pratiques dans le cadre de projets en commun et dans l'optique de justifier tel ou tel choix de déclaration CSS, ou encore de délimiter les différentes parties de la feuille de styles.

5. ASSOCIATION D'UNE FEUILLE DE STYLE A UN DOCUMENT HTML

Il existe plusieurs méthodes pour associer une feuille de style à une ou plusieurs pages HTML :

- **Intraligne (styles internes)**
- **Imbriquée (styles internes)**
- **Liée (styles externes)**
- **Importée (styles externes)**

5.1. Styles internes

Correspondent à l'incorporation à l'intérieur d'une page, d'où le titre "Styles internes".

L'intraligne

```
<html>
     <head>
     <title>  utilisation des feuilles de style   </title>
     </head>

     <body>
     <h1 style="color:orange; font-size:1.5em;">
Le mécanisme des feuilles de style
     </h1>
     <p style="color:blue; font-family:impact;>
l'attribution d'un style, …
     </p>
     </body>
</html>
```

Cette méthode est à **déconseiller** car il faut **réinsérer les informations de style pour chaque balise HTML**, plusieurs avantages des CSS sont donc perdus : facilité au niveau des modifications, sites web plus légers, séparation du contenu et de la mise en forme.
Vous l'utiliserez donc uniquement pour définir une mise en forme distincte dans une balise d'une page bien précise, et ce, **à titre exceptionnel** !

L'imbriquée

Dans ce cas, tous les éléments de style entre les balises <style> et </style> sont placés dans la section <head> du code HTML.

```
<html>
<head>
<title>  utilisation des feuilles de style  </title>

<style>
h2{font-family:verdana; color:green; font-size:1em;}
b {color:red; font-size:1.2em;}
</style>
</head>
<body>
<h2> décollage</h2>
Attention, pendant le décollage, vous devez <b> accrocher le manche </b> et <b>
garder le cap </b>
</body>
</html>
```

Cette méthode facilite la gestion de votre page car **les balises de style valent pour toute la page** et vous permettent de gagner au niveau richesse de mise en page.
Elle ne vous **permet** toutefois **pas d'utiliser la même feuille de style pour un ensemble de pages d'un même site**, cette méthode ne sera donc utilisée que lorsque certaines pages doivent être différentes de l'ensemble du site.

5.2. Styles externes

Outre définir une présentation de style valable pour une page (styles internes) les CSS nous propose mieux : définir une présentation de style valable pour plusieurs pages et même pour toutes les pages d'un site.
Créer une page externe regroupe toutes les feuilles de style et en reliant chaque page à cette page de style.

Liée

De loin la plus intéressante, la liaison entre la feuille de style et le document HTML se définit dans la zone d'en-tête du document de la manière suivante :

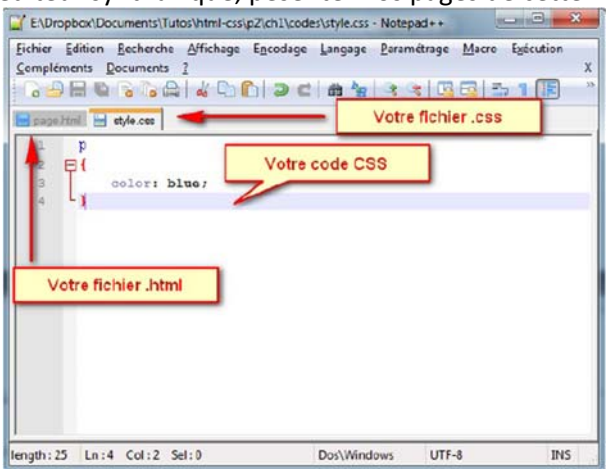
```
<html>
<head>
<link rel="stylesheet" type="text/css" href="url_du_fichier.css">
<head>
```

La balise `<link>` avertit le browser qu'il faudra réaliser un lien.
L'attribut `"rel=stylesheet"` précise qu'il y trouvera une feuille de style externe.
L'attribut `type="text/css"` précise que l'information est du texte et du genre *Cascading Style Sheets* (CSS).
L'attribut classique de lien `href=" ... "` donne le chemin d'accès (reltif !) et le nom du fichier à lier.

Dans le répertoire du site, un fichier **avec l'extension .css** contiendra tous les tags de style.

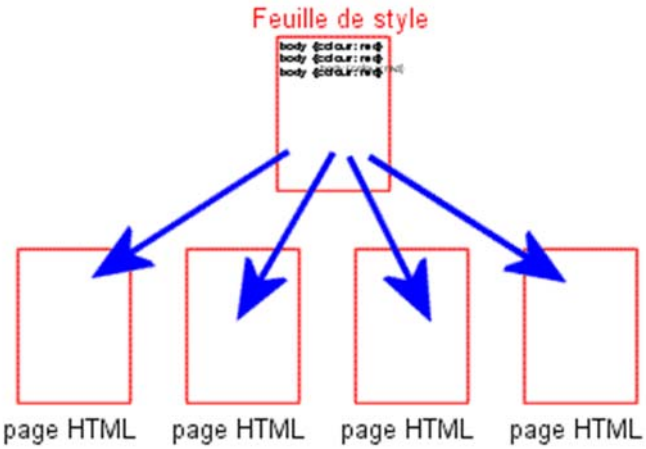
```
body
{
     color:red;
}
p
{
     font-size:0.8em;
     font-family:impact;
}
b
{
     color:green;
     font-size:1.2em;
}
```

Lorsque vous travaillez dans votre éditeur synthaxique, pésentez vos pages de cette manière :



Ouvrez ensuite votre fichier .html dans votre navigateur comme d'habitude, les mises en forme définies par la feuille de style (fichier.css) apparaissent sur la page HTML (fichier.html).

Cette même feuille de style est ensuite rattachée à plusieurs pages d'un site et permet donc de gagner au niveau cohésion de la présentation tout au long du site, facilité des modifications, légèreté et créativité dans la mise en page.



Dans la feuille de style, on précise donc la présentation des différents éléments qui sont présents dans les pages HTML.
Toutes les informations concernant la présentation sont séparées du document lui-même.

Importée

Il ne s'agit pas d'une balise HTML mais d'une règle CSS. Il est donc nécessaire de la déclarer dans l'élément <style> de l'en-tête du document :

```
<style type="text/css">
     @import url(styles.css);
</style>
```

Ou dès le début d'une feuille de style externe :

```
@import url("reset.css");
@import url("screen.css");
```

La règle @import , propriété CSS 2, sera suivie de l'URL d'un fichier contenant les styles à appliquer en plus de la feuille de styles en cours (chemin relatif !). On pourra préciser une liste de médias.

Par exemple :

```
@import url(/styles/habillage.css) screen;
ou :
@import url(impression.css) print;
```

Cette propriété permet en outre d'inclure des feuilles de styles dans d'autres feuilles de style. Cette méthode permet de créer des feuilles de styles dynamiques sans devoir recopier plusieurs fois le même code.

5.3. Ordre d'importance des feuilles de style

Si différentes méthodes d'association sont utilisées en même temps, le navigateur Web doit pouvoir donner la priorité à l'une ou l'autre des méthodes.
Les navigateurs Web qui supportent les feuilles de style disposent pour cela d'une règle concernant **l'ordre des cascades** qui leur indique quoi faire dans quelle situation.

1. intraligne
2. imbriquée
3. liée
4. importée
5. style par défaut du navigateur

Ainsi, les styles intralignes prennent le pas sur les styles imbriqués, qui eux-mêmes ont la priorité sur les styles liés ...

5.4. Pourquoi feuilles de style de cascade ?

Pourquoi parle-t'on de "feuilles de style en cascade" (Cascading Style Sheets) ? Car on peut déclarer les styles à différents endroits, et selon ces endroits ils seront plus ou moins prioritaires. On obtient donc une cascade de styles.
On parle également de "feuilles de style en cascade" car la majorité des styles que l'on applique à la balise HTML va être propagée à ses enfants (les balises incluses dans HTML), ainsi qu'à tous les descendants, ce qui forme une cascade. Et ceci, jusqu'à ce nous indiquions le contraire.

6. LA COULEUR EN CSS

Même si « les goûts et des couleurs ne se discutent pas », il existe certaines règles en matière d'association des couleurs : certaines cohabitent mal, souvent pour des raisons évidentes de lisibilité du document. De plus, les signes et influences attribués aux couleurs par les cultures et les sociétés humaines sont variés.

6.1. Les couleurs et leurs symboliques

Couleur	Symbolique	Exemples d'utilisation
Blanc	Le vide, l'espace, la pureté, la propreté et la sobriété. C'est une couleur reposante et non agressive.	Intervient rarement explicitement ; souvent relégué au second plan (une page web aérée se devant de présenter un arrière-plan allégé). Il permet aussi de faire ressortir les éléments importants de la page.
Noir	Symbole de deuil ou de tristesse, le noir est également associé à l'art, à l'apparat et au luxe.	Un fond noir donne une impression d'élégance : on retrouve ainsi cette technique sur les sites web d'art, les boutiques de luxe ou les casinos. Le noir se marie très bien avec les autres teintes.
Gris	Malgré sa connotation de mélange, le gris est également une couleur passe-partout.	Comme le bleu, cette couleur est souvent associée aux sites technologiques ou informatiques : les ordinateurs et le matériel informatique sont généralement gris.
Vert	Représente la nature et tout ce qui s'y rapporte. C'est aussi une couleur qui confère une impression de fraîcheur saine : on la retrouve ainsi dans les sirops et chewing-gums rafraîchissants.	Sites en rapport avec la nature, la chasse ou les loisirs.
Bleu	Couleur froide par excellence, le bleu inspire la rigueur et la science. Il représente également de grands espaces comme le ciel ou la mer, conférant ainsi une impression de tranquillité.	Sites web technologiques ou scientifiques.
Jaune	Couleur stimulante évoquant le dynamisme, le jaune attire l'œil… mais trop de jaune agresse.	Sites associés à des thèmes dynamiques (sport, loisir, publicité, médias).
Rouge	Le rouge (avec l'orange) est la couleur chaude par excellence. Couleur associée à plusieurs éléments très forts, tels que le feu, l'amour, le sang.	Sa force rehausse la pâleur générale et dirige le regard du visiteur. Un site à dominante rouge dégage chaleur et passion.

Lecture Web :
La couleur et les marques : https://cdn.1min30.com/wp-content/uploads/2013/02/Color_Emotion_Guide22-640x560.png

62. La notation RGB

Cette notation permet de définir les composantes de rouge, de vert et de bleu en indiquant leur proportion relative en pourcentage ou bien en notation entière absolue (chaque composante étant un entier compris entre 0 et 255).
La syntaxe CSS pour la notation des couleurs en RGB (Rouge, Vert, Bleu) est la suivante : rgb(0,0,255) (ceci correspond au bleu).

6.3. La notation hexadécimale

Plus adaptée à l'informatique et notamment à la manipulation d'octets. On s'y autorise seize chiffres (*les chiffres habituels complétés par les six premières lettres de l'alphabet – généralement écrites en minuscules*). Deux caractères peuvent alors exprimer 16 fois 16 (soit 256) valeurs différentes, de 00 (0) à ff (255).
Le symbole dièse (#) introduit une valeur RGB codée en hexadécimal. Les trois composantes sont rassemblées sur six caractères : **`#0000ff`** (bleu).

6.4. Les mots-clés étendus (CSS3)

La spécification CSS2.1 ne fournit que 17 mots-clés, raccourcis désignant les couleurs primaires RVB et leurs dérivés. Le W3C propose aujourd'hui (dans sa spécification CSS3) **147 couleurs nommées**. La couleur `transparent` doit également s'ajouter à la liste. `currentColor` correspond à la couleur de la propriété color ou celle héritée par la cascade.

Lecture Web :
couleurs nommées : http://htmlcolorcodes.com/fr/noms-de-couleur/

6.5. Utiliser RGBA et HSLA (CSS3)

Le module CSS3 (`color`) définit 4 fonctions de couleur: **rgb()**, **hsl()** et leurs pendants respectifs avec transparence **rgba()** et **hsla()**.

rgba()

La fonction rgba() acceptent comme paramètres les valeurs *Rouge* (R), *Vert* (G), *Bleu* (B) allant de 0 (noir) à 255 (blanc)et *Alpha* (A) allant de 0 (transparent) à 1 (visible).

Lecture Web :
convertir hexadécimale en RGBa : http://hex2rgba.devoth.com/

HSLA()

Les fonctions `hsl()` ou `hsla()` reposent sur un mode de représentation des couleurs différent, basé sur une *Teinte* (H), la *Saturation* (S) et la *Lumière* (L).
Cette façon de représenter la couleur offre plusieurs avantages, notamment la possibilité de définir les nuances (couleurs plus claires) et ombres (couleurs plus foncées) d'une teinte. Il suffit pour cela de modifier la luminosité :

```
/* Teinte de base */
hsl(14, 76%, 55%);
/* Plus foncée */
hsl(14, 76%, 75%);
/* Plus claire */
hsl(14, 76%, 35%);
```

Deux contraintes :

- Ces fonctions ne sont supportées qu'à partir d'IE9
- Les logiciels graphiques (tel Photoshop) n'utilisent pas le même modèle HSL que celui de CSS. Il n'est donc pas possible de récupérer les valeurs fournies pour les utiliser directement dans votre CSS, il faut les convertir.

Lecture Web :
www.colorzilla.com, https://kuler.adobe.com
http://www.editions-eyrolles.com/Chapitres/9782212136890/annexe_A_Draillard.pdf
www.pourpre.com qui aborde exhaustivement les couleurs, de façons variées.
http://www.colorschemer.com/online.html et http://colorschemedesigner.com/ permettant d'effectuer des tests dans les accords de couleurs.
www.joliespages.com et http://www.cssmozaic.com propose une galerie de sites au design classé par couleur.

7. LA TYPOGRAPHIE ET LA MISE EN FORME DE CARACTERES

7.1. Déclarer une police de caractère en CSS

`font-family` permet de spécifier la famille de police que nous souhaitons utiliser pour le rendu de nos textes. Une liste de polices séparées par une virgule doit être précisée, la première de la liste étant le choix principal.

```
body {
font-family: 'Trebuchet MS', verdana, sans-serif;
}
```

Cette propriété se transmettant hiérarchiquement par héritage, elle vaudra également pour tous les descendants de l'élément <body>, c'est-à-dire à toute la page web. Il sera donc inutile de préciser une police pour chacun d'entre eux, sauf évidemment pour en changer.

Si l'utilisateur ne possède pas l'une des trois polices choisies, une police système par défaut est utilisée pour mettre en forme votre texte.

7.2. Les familles de polices génériques

Toutes les polices sont classées par familles génériques, qui les regroupent selon leur empattement, leur chasse, ou leur style. On retrouve ainsi les polices dites :

`serif` : polices à empattements
`sans-serif` : polices sans empattements
`monospace` : polices à chass fixe
`cursive` : polices manuscrites
`fantasy` : polices décoratives

Les polices sans serifs, dénuées d'empattements, sont d'apparence plus simple et plus sobre. Leur confort de lecture sur écran les fait souvent privilégier dans les chartes graphiques pour le Web. Les plus fréquentes sont Verdana et Arial.
Les polices à chasse fixe, dites monospace, attribuent à chaque caractère une largeur constante, comme sur les machines à écrire (les « m » des textes imprimés étant au contraire généralement plus larges que leurs «l»). Dans cette famille, on retient souvent Courier New pour l'affichage de code, de listings et de toutes les portions de texte à reproduire.
Pour éviter que le système client opte pour une solution par défaut faute de disposer de la police demandée, il est d'usage d'en préciser trois types : une police pour PC, une autre pour Mac, en concluant par une famille générique disponible partout :

```
body {
font-family : arial, helvetica, sans-serif;
}
```

Pour des raisons syntaxiques, il est obligatoire en CSS de protéger les noms composés par des apostrophes simples (') ou doubles ("). Ainsi, on peut écrire arial mais on devra écrire `'Trebuchet MS'` ou `"Trebuchet MS"`.

Le site Font Stack

Le site http://cssfontstack.com/ donne une estimation du support des polices de caractère en fonction des différents OS et permet d'obtenir la règle `font-family` optimale pour une police choisie.

Les polices standards

On peut établir une liste de 11 polices de caractères dites standards :

Nom	Caractéristiques	Usage
Arial	Sans sérif	Imprimé, Web
Arial Black	Sans sérif, forte graisse	Imprimé, Web
Comic Sans MS	Police fantaisie, sans sérif	Web
Courier New	Chasse fixe	Listing, Code
Georgia	Empattements simples (sérif)	Web
Impact Monotype	Sans sérif	Imprimé, Web
Symbol Monotype	Alphabet grec	Imprimé, Web
Times New Roman	Empattements (sérif)	Imprimé, Web
Trebuchet MS	Sans sérif	Web
Verdana	Sans sérif	Web
Webdings	Police fantaisie	Web

7.3. Affichage des polices exotiques en utilisant les images

Qui consiste à transformer les textes exotiques en images, dans un format bitmap (GIF, JPEG, PNG) ou vectoriel.
Inconvénients :

- ✓ Une image occupe dix à cent fois plus d'espace que son équivalent texte.
- ✓ L'efficacité de référencement est meilleure sur du texte
- ✓ Insérer des images de texte dans un code HTML remet en cause la séparation entre structure et mise en forme. Cela complique aussi les maintenances et mises à jour.
- ✓ Même si sa propriété alt est renseignée (à contre-emploi de son rôle théorique de texte alternatif), une image reste moins accessible aux non-voyants et sera mal exploitée par les moteurs de recherche et les navigateurs en mode texte.

Cette technique est donc à éviter à tous prix !!

7.4. Importer des polices de caractères (CSS3)

Il est à présent possible grâce à la propriété @font-face d'embarquer des polices sur son site, et que celles-ci s'affichent correctement même si l'utilisateur n'a pas la police installée.
Le CSS 3 propose **le téléchargement automatique par le navigateur de la police utilisée** dans les pages. Toutefois, il faut posséder la police mise en oeuvre car elles tombent dans les **règles des droits d'auteur**.

Les différents formats de polices

La police de caractère, ou plutôt les différents formats de la police de caractère doiventt être placés sur le serveur.

- .**ttf** : TrueType Font
- **.otf** : OpenType Font
- **.eot** : Embedded OpenType (propriétaire Microsoft, pour IExplorer),
- **.svg**, **.svgz** : SVG Font (Iphone, smartphone)
- **.woff** : Web Open Font Format

Lecture Web :
Le site www.fontsquirrel.com crée pour vous des kits@font-face « prêt à l'emploi » contenant tous ces formats de police téléchargeables ainsi que deux fichiers permettant de copier-coller rapidement la syntaxe de @font-face dans nos pages.

Déclaration de base de@font-face

Elle sera chargée au moment de l'appel de la feuille de style dans laquelle est déclarée **`@font-face`**.
Cette déclaration est définie par une **@règle** et est composée de deux propriétés minimum : `font-family` permettant de nommer la police (nom ensuite employé pour l'appliquer aux textes) et `src` permettant de spécifier l'adresse de la font à télécharger.

```
@font-face {
    font-family: 'anudawitalic';
    src: url('ANUDI-webfont.eot');
    src: url('ANUDI-webfont.eot?#iefix') format('embedded-opentype'),
         url('ANUDI-webfont.woff') format('woff'),
         url('ANUDI-webfont.ttf') format('truetype'),
         url('ANUDI-webfont.svg#anudawitalic') format('svg');
    font-weight: normal;
    font-style: normal;
}
```

Utiliser la police déclarée

Il suffit ensuite de déclarer le nom de cette police dans la liste des valeurs de la propriété **font-family** d'un élément :

```
h1 {
      font-family: anudawitalic, Arial, Helvetica, sans-serif;
}
```

Optimiser le téléchargement

Attention : il est essentiel de *préciser le format de la police* afin d'optimiser le chargement. En effet, un navigateur ne téléchargera pas un format de police qu'il ne supporte pas.
La succession des déclarations s'ordonne en fonction du poids des polices : *de la plus légère vers la plus lourde (*pour des raisons de performance).
Le format WOFF, spécialement pensé et conçu pour le web, est à présent supporté par tous les navigateurs récents.

Lecture Web :
FontSquirrel fournit les différents formats, mais il fournit aussi le code CSS qui va bien. Après avoir reclassé les formats par poids croissant.
http://typographisme.net/post/Bonnes-pratiques-pour-les-déclarations-@font-face
http://typographisme.net/post/Les-formats-de-polices-typographiques-pour-le-Web

Exploiter les variantes

Il est possible de forcer le navigateur à utiliser les polices de substitution adéquate pour les différentes variantes de police (gras, italique). Si rien n'est précisé, les navigateurs vont simuler ces variantes et provoquer parfois des rendus disgracieux.
Il est possible dans certains cas de télécharger plusieurs types d'une même police et de préciser pour celle-ci les règles `font-weight`, `font-style` ou `font-variant`.

```
@font-face {
      font-family:'quadranta';
      src: url('quadranta.woff') format('woff'),
           url('quadranta.otf') format('truetype'),
           url('quadranta.eot?#iefix') format('embedded-opentype'),
           url('quadranta.svg#QuadrantaBold') format('svg');
           font-weight:normal;
           font-style:normal;
}
@font-face {
      font-family:'quadranta';
      src: url('quadranta_bold.woff') format('woff'),
```

```
            url('quadranta_bold.otf') format('truetype'),
            url('quadranta_bold.eot?#iefix') format('embedded-opentype'),
             url('quadranta_bold.svg#QuadrantaBold') format('svg');
             font-weight:bold;
             font-style:normal;
}
```

L'appel à cette police se fera ensuite en précisant **`font-weight:bold;`** ce qui forcera le navigateur à récupérer la version gras de la police au lieu de la simuler.

Le « cas » Internet Explorer

Internet Explorer interprète exclusivement le format EOT (d'ailleurs reconnu par aucun autre navigateur) **jusqu'à sa version 8**.
De plus, dans sa version de 6 à 8, il ne comprend pas la syntaxe CSS3 à base de déclarations multiples séparées par des virgules (parsage), utilisée pour la propriété src et renvoit un message d'erreur. La solution de contournement[14] pour corriger le problème d'Internet Explorer consiste à placer un point d'interrogation (?) juste après le nom du fichier EOT et ainsi lui faire croire que ce qui suit est une *query string*[15] et télécharger le fichier sans soucis.

Et pour Internet Explorer 9.0 et suite …
Afin de profiter du support de la police au format WOFF, il est possible de « sauter » l'appel au format EOT. Pour ce faire, modifiez le nom du format de la police reconnue par les autres versions d'IE (EOT), de manière à ce qu'il ne le reconnaisse pas et teste alors le format suivant (WOFF) et le télécharge.
Un dernier problème subsiste lorsqu'IExplorer 9.0 fonctionne en mode compatibilité – pour pallier à cela, une balise meta empêche le navigateur de basculer en ce mode là :

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
```

@font-face generator

http://www.fontsquirrel.com/tools/webfont-generator
Propose de créer nos propres kits à partir d'une police de notre création ou d'une police libre de droit.

Les sites de fonts

Il existe des services qui **fournissent des polices à intégrer directement dans son CSS depuis une URL distante**. Les polices proposées sont en général adaptées à l'affichage dans un navigateur, s'afficheront correctement sur la majorité des navigateurs, et enfin le service se chargera de fournir la police à l'utilisateur, ce qui peut s'avérer non négligeable étant donné que la plupart des polices font entre 50ko et 300ko !
Google Web Fonts (www.google.com/webfonts) propose d'héberger sur leurs serveurs les polices employées. Lors de l'appel du code généré par Google et intégré dans la CSS de notre site, celui-ci se charge de vérifier quel format de fonte est adapté au navigateur et ainsi ne faire télécharger que ce fichier par l'utilisateur.

Le choix d'une ou plusieurs polices pour notre site se fait ainsi en 3 étapes :

1. Choisir la police de caractère
2. Comparer et vérifier le contenu
3. Obtenir le code (>> *Quick use*)

La méthode d'importation peut être :

1. Via une balise `link`
2. Via la règle `@import`
3. Via Javascript

[14] Astuce mise au point par Ethan Dunham
[15] Une chaîne de caractère qui permet de passer des données de type clé/valeur dans l'URL.

Ici, il n'y a donc pas de @font-face à écrire. Tout est géré dans la feuille de style hébergée chez Google, qui va effectuer tous les traitements de son côté afin d'assurer l'affichage correct de la police.
L'utilisation de l'API se résume donc à :

- Inclure la police qui nous intéresse
- L'appliquer à une balise ou classe CSS avec font-family

Lecture Web :
http://typekit.com, http://new.myfonts.com

8. LA TAILLE DES CARACTERES

Deux systèmes permettent d'indiquer les dimensions des éléments en CSS : **les unités de taille fixes ou absolues** (px, pt, cm, …) et **les unités de taille relatives** (%, em, …). C'est la propriété `font-size` qui détermine la taille de la police d'un élément.

```
p {
font-size: 1.4em;
}
```

Elle est héritée, sa valeur sera donc retransmise à tous les descendants de l'élément considéré.

8.1. Les unités de taille fixes

Le W3C conseille de limiter leur usage à des médias de sortie connus, aux propriétés déterminées … on évitera donc ces mesures pour un écran d'ordinateur chaque moniteur étant particulier (de par sa taille de diagonale, sa résolution, son nombre de couleurs, etc.). De telles unités sont toutefois parfaitement adéquates à une sortie sur papier.

8.2. Les unités de taille relatives et pourcentage

Les unités relatives sont **les em** (cadratins), **le ex** (hauteur d'X), **le pourcentage** (%).
1 em représente la taille (largeur) d'un caractère – m - (ainsi que l'espace pour ses jambages) dans la police de référence, tandis qu'*1 ex correspond à la hauteur d'un caractère minuscule (x) dans la police de référence*. Le % n'est pas spécifique aux textes et polices, il se définit relativement à la taille de référence dans le conteneur de l'élément.
Ces unités se basent donc sur les préférences de l'utilisateur. Si le navigateur est configuré à 14px, 1em correspondra alors à 14px.
Pour ces 3 unités, la taille de la police de référence se transmet par héritage : dans le cas d'éléments imbriqués, la police change à chaque nouveau conteneur.

Ainsi :

```
p, span{
font-size:2em;
}
```

Fera apparaître le texte des paragraphes doublement plus grand que la taille dans la police de référence de l'utilisateur.
Et dans un code de type :

```
<p> cette phrase contient un <span> grand </span> mot
```

La phrase aura une taille de 2em mais "grand" sera en 4em (le double).

Les mots-clés de taille sont aussi possibles mais sont des mentions approximatives car elles sont laissées à l'appréciation de chaque navigateur.
`xx-small` =minuscule.
`x-small` = très petit.
`small` = petit.
`medium` = moyen.
`large` = grand.

`x-large` = très grand.
`xx-large` = géant.
`smaller` = visiblement plus petit que normal.
`larger` = visiblement plus grand que normal.

8.3. Permettre l'agrandissement des polices

De nombreux utilisateurs souhaitent agrandir la police, pour certains c'est une nécessité. Le rendu d'un site devant être conforme aux attentes et aux souhaits des utilisateurs, il convient d'autoriser une certaine souplesse dans les pages. Ainsi, un affichage correct n'implique d'imposer **aucune dimension**, **seules les images et les illustrations doivent être figées pour s'afficher dans les meilleures conditions**. Partout ailleurs, on préférera des dimensions relatives.

8.4. Zoom des navigateurs récents

Alors que les anciennes versions d'IE par le menu *Affichage / Taille du texte* refusait d'agrandir tout texte dont la taille est spécifiée en pixels, les dernières versions des principaux navigateurs (Safari, Firefox, Google Chrome, Opera, Internet Explorer) proposent maintenant par défaut un zoom de page plutôt qu'une mise à l'échelle du texte avec les commandes Ctrl++/– et Cmd++/–. Le zoom de page est un agrandissement qui affecte la totalité de la page (mise en page, formatage et taille de texte). Le design demeure ainsi harmonieux et contrôlé.

8.5. Le REM

Le CSS3 définit une nouvelle unité de mesure pour les textes, le `rem`. Cette unité relative permet de baser la taille d'un texte sur la taille de l'élément racine (*root*), dans notre cas :`html`, au lieu de se baser sur l'élément parent comme le fait le `em`.

Ainsi :

```
p, span{
font-size:2rem;
}
```

Fera apparaître le texte des paragraphes doublement plus grand que la taille dans la police de référence de l'utilisateur.
Et dans un code de type :

```
<p> cette phrase contient un <span> grand </span> mot
```

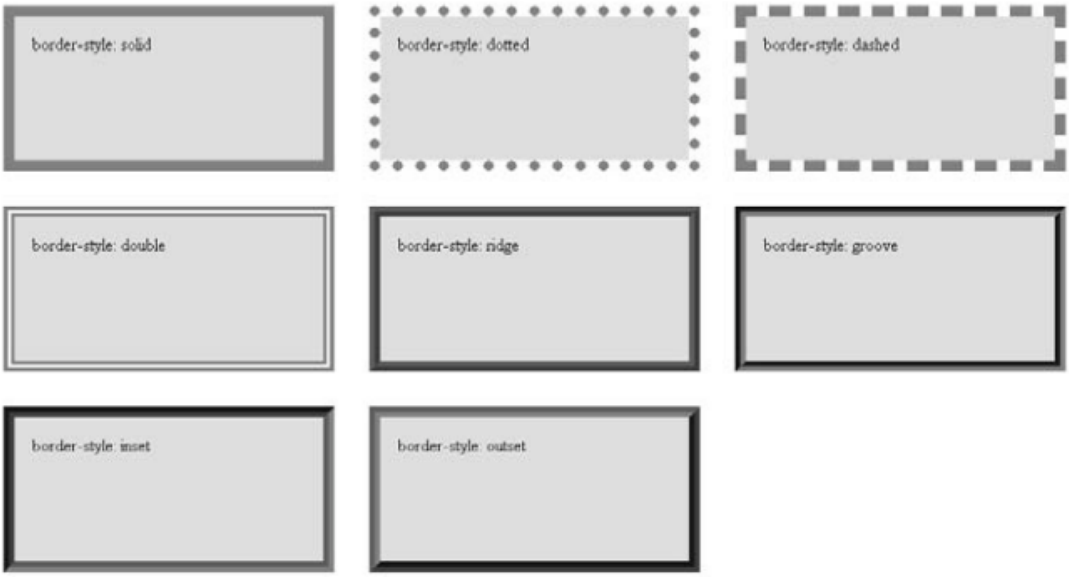
La phrase et "grand" auront une taille de 2em car ils se baseront tous deux sur la racine (HTML).

Lecture Web :
A consulter pour l'unité « em » : http://css.alsacreations.com/Tutoriels-et-articles-divers/gerer-la-taille-du-texte-avec-les-em
Pour convertir les pixels en em : http://pxtoem.com/

9. LES BORDURES, LES ARRONDIS ET LES OMBRES

9.1. La propriété border

La propriété **border-style** met en place le type des bordures, 10 styles de bordures sont possibles :



La propriété **border-width** définit l'épaisseur des bordures, et n'a de sens qu'en accompagnement d'un style (**border-style**) ou d'une couleur de bordure (**border-color**). Certains (anciens) navigateurs n'interprètent d'ailleurs que les bordures ayant renseigné ces deux propriétés, il est donc préférable de les déclarer en premier.

`border-style`	`solid, dotted, dashed, double, groove, ridge, inset, outset`	*Type de bordure*
`border-color`	Exemple : `#CF1A20`	*Couleur de bordure*
`border-width`	`3px`	*Épaisseur de bordure*
(version raccourcie) `border`	`3px solid black`	*Combine border-width, border-color, border-style. Existe aussi en version border-top, border-right, border-bottom, border-left.*
`border-radius` (CSS3)	`5px`	*Bordure arrondie*

Astuce : pour supprimer les bordures autour des images contenant un lien hypertexte

```
a img {
border: 0;
     }
```

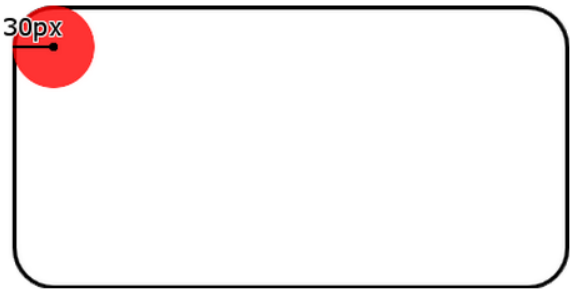
9.2. La propriété border-radius (CSS3)

Des coins en forme de cercles

La propriété CSS3 `border-radius`[16] permet de créer des coins arrondis et de définir le rayon de l'arrondi désiré, le navigateur gère le reste :

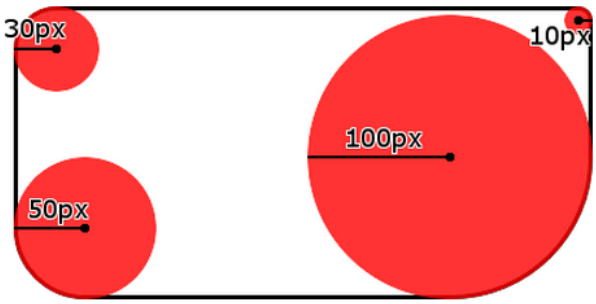
```
box {
     border-radius: 30px;
}
```

[16] Alors que les techniques Css 2.1. imposaient la création d'images pour créer des coins arrondis (très lourd à gérer)



Pour définir l'aspect des différents coins séparément :

```
box {
      border-top-left-radius: 30px;
      border-top-right-radius: 10px;
      border-bottom-right-radius: 100px;
      border-bottom-left-radius: 50px;
}
```



Des coins en forme d'ellipses

Afin de personnaliser au maximum les coins de nos boîtes, il est possible d'utiliser des ellipses en plus des cercles. Les arrondis se formeront selon le périmètre des ellipses que nous définirons.

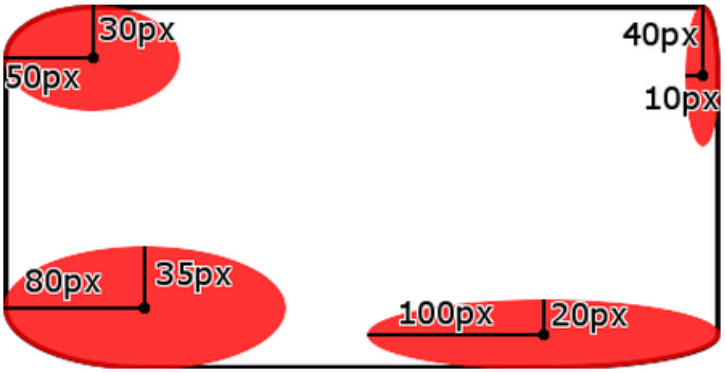
```
box {
      border-radius: 50px / 30px;
}
```

La **première valeur** définit ainsi la **largeur** du rayon horizontal de l'ellipse tandis que la **seconde** indique la **hauteur** de son rayon vertical.

Tout comme les cercles, il est bien entendu possible avec les ellipses de spécifier différents arrondis pour chaque coin. La manière de faire est similaire à ce qu'on a déjà vu :

```
box {
      border-radius: 50px 10px 100px 80px / 30px 40px 20px 35px;
}
```

Les quatre premières valeurs définissent les rayons horizontaux de chacune des quatre ellipses générées (tout d'abord celle située en haut à gauche, puis celle en haut à droite, celle en bas à droite et enfin celle en bas à gauche), après le « / » les quatre dernières valeurs décrivant les hauteurs des rayons des ellipses.



Compatibilité : border-radius est supporté par tous les navigateurs maintenant, et depuis plusieurs versions. Pour Opera et IE, ces préfixes n'ont jamais existé (lorsque la propriété n'est pas supportée, la boîte apparaît avec des coins carrés).
Source des schémas : https://openclassrooms.com/courses/stylisez-votre-site-avec-css3/repoussons-les-limites-des-bordures

Lecture Web :
http://border-radius.com/

9.3. La propriété box-shadow (CSS3)

Les ombres portées donnent de la profondeur et rehaussent l'apparence des pages trop plates. Les CSS3, avec la propriété `box-shadow`, permettent d'appliquer des effets d'ombre portée sans altérer les images d'origine (gain de temps lors d'éventuelles modifications).
Cette propriété prend **six paramètres** : les *décalages vertical* et *horizontal*, *la valeur du flou* (0=net et une valeur positive provoque un flou gaussien), la *largeur de l'ombre,* sa *couleur* et un *mot-clé* (inset ou outset, optionnel)

```
img {
  box-shadow: 3px 3px 0 6px #666 inset;
}
```

Les valeurs de décalage sont obligatoires, les valeurs de flou, de couleur et le mot-clé sont optionnelles.

Exemple :

```
img {
  box-shadow: 3px 3px 6px #666;
}
```

Compatibilité : Safari, versions récentes de Chrome, Opéra et Firefox ainsi que par Internet Explorer 9
Lorsque la propriété n'est pas supportée, la boîte apparaît sans effet d'ombre.

Des valeurs multiples peuvent être créées pour générer des ombres différentes pour chaque bord, exemple :

```
img {
  box-shadow:
      0px 0px 20px black,
      20px 15px 30px yellow,
      -20px -15px 30px lime,
      -20px -15px 30px blue,
      20px -15px 30px red;
      }
```

Lecture Web :
http://www.alsacreations.com/tuto/lire/910-creer-des-ombrages-ombres-css-box-shadow-text-shadow.html
http://www.css3generator.com/

10. SIMPLIFIER AVEC LE REGROUPEMENT DES PROPRIETES

Certaines propriétés génériques prévoient une version raccourcie permettant l'application de plusieurs valeurs en 1 seule déclaration.

Les propriétés liées à la police (font)

```
p{
      font-family:arial;
      font-style:italic;
      font-weight:bold;
      font-size:120%;
      line-height:140%;
}
```

Sera identique à :

```
p{
      font : italic bold 120%/140% arial
}
```

Les propriétés liées aux bordures (border)

```
border-width: 3px;
border-style: dotted;
border-color: gray;
```

Sera identique à :

```
border: 3px dotted gray;
```

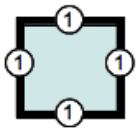
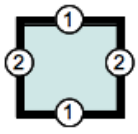
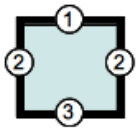
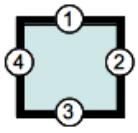
Les propriétés liées à la marge et au remplissage (padding)

```
margin-top: 10px;
margin-right: 5px;
margin-bottom: 10px;
margin-left: 5px;
```

Sera identique à :

```
margin: 10px 5px 10px 5px;
```

Les propriétés raccourcies qui gèrent les bords d'une boîte telles que `border-style`, `margin` ou `padding` utilisent une méthode constante selon qu'elles reçoivent 1 à 4 valeurs :

	*1 valeur* : `border-width: 1em` — La valeur unique s'adresse à tous les côtés.
	*2 valeurs* : `border-width: 1em 2em` — La première valeur représente les côtés horizontaux en haut et en bas. La seconde valeur représente les côtés verticaux, à gauche et à droite.
	*3 valeurs* : `border-width: 1em 2em 3em` — La première valeur représente le côté haut, la deuxième les côtés gauche et droit et la troisième représente le côté bas.
	*4 valeurs* : `border-width: 1em 2em 3em 4em` — Les quatre valeurs représentent respectivement le côté haut, le côté droit, le côté bas et le côté haut, toujours dans cet ordre (le sens horaire).

Ces notations raccourcies permettent d'éviter l'accumulation de déclarations de styles mais attention ! **L'ordre est important sous peine de ne pas voir s'appliquer les styles.**

Certains sites comme http://www.styleneat.com/ ou http://www.cleancss.com vous proposent d'organiser automatiquement votre feuille de style en classant hiérarchiquement vos sélecteurs et propriétés.

11. LES CLASSES ET LES ID

On désire parfois affecter des styles différents à une même balise, les feuilles de style vous proposent pour cela **l'attribut class et l'attribut id,** ceux-ci peuvent utilisés sur toutes les balises. Les attributs **class** et **id** sont quasiment identiques.

11.1. Les classes

Une classe est un nom que l'on choisit librement (en se limitant aux caractères alphanumériques classiques) et dont on baptise les éléments concernés. **Une classe permet d'attribuer un comportement différent à certains éléments.**

La définition d'un style était :
```
balise {propriété de style: valeur;}
```
Elle devient :
```
.nom_de_classe {propriété de style: valeur;}
```
Pour appeler l'effet de style dans le document, on ajoute le nom de la classe à la balise.
```
<balise class="nom_de_classe"> .... </balise>
```

Exemple
Je souhaite mettre ce qui est important dans le texte en gras et en bleu. Je crée la classe **`.essentiel`** :
```
.essentiel {font-weight: bold; color: #000080;}
```
Et dans le document HTML, il suffit d'appeler la classe **`.essentiel`** quand cela se révèle nécessaire :
```
<p class="essentiel"> cette partie de texte est très importante </p>
<h1 class="essentiel">titre 1</h1>
<table><tr><td class="essentiel">cellule</td></td>...
```

Le nom de classes utilisé est libre (tant qu'il débute par une lettre) et est précédé d'un point. **On peut appliquer la même classe à autant d'éléments qu'on le souhaite et un même élément peut recevoir plusieurs classes.**
```
<h2 class="sommaire titre2"> Titre de niveau 2 en rouge et centré </h2>
```

11.2. Les identifiants

Les ID fonctionnent exactement comme les classes, excepté **qu'il ne peut être utilisé qu'une fois dans le code**. Le nom est choisi librement mais est précédé d'un #. En pratique, les id sont utilisés sur des éléments qui sont uniques sur votre page (comme par exemple le logo).

La syntaxe est :
```
#nom_de_ID {propriété de style: valeur;}
```
Et pour l'appeler :
```
<balise id="nom_de_ID"> .... </balise>
```
Pour #essentiel { ... }
```
<P id="essentiel"> est correct.
```
Mais si on rencontre dans la même page
```
<h1 id="essentiel"> ce n'est plus correct !
```

11.3. Nommer les éléments

Lorsque vous nommez les ID et les classes, choisissez des noms indépendants de la présentation, nommez les éléments en fonction de ce qu'ils sont (sémantique) plutôt que de l'apparence qu'ils possèdent. Le code est ainsi plus compréhensible et ne risque jamais de se désynchroniser avec la mise en page.
Lorsque vous écrivez des noms de classe et d'ID, soyez attentif à la casse, car les navigateurs la respectent. Vous pouvez employer des "-" ou des "_".

Conclusions

On ne peut appliquer qu'un seul nom d'ID à un élément dans une page mais un même nom de classe à n'importe quel nombre d'éléments… c'est tout l'intérêt des classes. Elles sont donc utiles pour identifier des types de contenu ou des éléments similaires. **Il peut donc être intéressant d'attribuer automatiquement des id aux objets uniques du code (en-têtes, bloc publicité, bloc rubriques, annexes, …) et des classes aux éléments qui se répètent (paragraphes, listes de menus, …)**
Lorsque vous intégrez du code javascript et CSS dans une même page, les identifiants peuvent être utiles pour reconnaître certaines balises.

12. LES BALISES UNIVERSELLES (SPAN ET DIV)

Il arrive parfois que vous ayez besoin d'appliquer une class (ou un id) à certains mots qui ne sont pas à l'origine entourés par des balises. <DIV> et <SPAN> sont deux balises HTML dites « **universelles** » car elles n'ont aucune signification particulière. **C'est associées aux attributs CLASS et ID qu'elles permettent d'utiliser des mécanismes de mises en page complexes.**

12.1. SPAN

La balise <SPAN> ... </SPAN> permet d'appliquer des styles à un morceau de paragraphe, il rend possible la délimitation d'un groupe de texte. **C'est une balise de type inline, elle ne provoque pas de retour à la ligne.**

Exemple

```
<html>
    <head>
    <style type="text/css">
    .element{font-size: x-large; color: navy}
    </style>
    </head>
  <body>
  <p> un monde de <span class="element"> géants </span> en papier</p>
  </body>
</html>
```

12.2. DIV

La balise <DIV> ... </DIV> permet de délimiter un bloc de contenu (texte, liste, tableau, image). **C'est une balise de type block., elles créent un nouveau "bloc" dans la page, et provoquent donc obligatoirement un retour à la ligne.**

Exemple

```
<html>
<head>
<style type="text/css">
#header{font-size: x-small;background-color:silver;}
</style>
</head>
<body>
la balise div
<div id="zone">
<p> commentaire : </p>
<p> permettant de délimiter un bloc de contenu … </p>
<table> <tr> <td> l'attribut div </td> </tr> </table>
</div>
</body>
</html>
```

Pour limiter le plus possible le balisage, utilisez d'élément div lorsqu'il n'existe pas déjà un élément distinctif (exemple: si vous vous servez d'une liste pour votre navigation principale, il n'est pas nécessaire de l'envelopper dans une div, utilisez l'ul déjà disponible).

```
<div> /* pas nécessaire
  <ul id="nav">
    <li><a href="/home/">Home</a></li>
    <li><a href="/about/">About Us</a></li>
    <li><a href="/contact/">Contact</a></li>
  </ul>
</div> /* pas nécessaire
```

Veillez également à utiliser des div pour regrouper des éléments liés par leur signification ou leur fonction plutôt que par leur présentation ou leur disposition (sémantique).

Classite ou Divinite aiguë

La classite ou divinite aigüe consistent à utiliser un sélecteur "class" ou ajouter un "div" quand ce n'est pas nécessaire alors que les CSS peuvent appliquer plusieurs règles à un élément et cibler directement les sous-éléments d'une classe ou d'un div.
Le ciblage des éléments[17] permet de simplifier votre balisage (et le rendre plus logique aux navigateurs et machines de traitement comme le Googlebot), **condenser le code et ajouter sens et structure aux pages Web**. En supprimant les classes superflues, vous simplifierez le code et réduirez le poids de vos pages. **Votre contenu est plus facile à traiter et à mettre à jour.**

[17] Consultez le chapitre sur les sélecteurs et l'arborescence – le ciblage avancé