Les feuilles de styles CSS (Cascading Style Sheets)

Les feuilles de styles permettent de décrire la présentation des éléments HTML, de façon séparée du contenu et de sa structure, décrits par le HTML. Cette séparation présente l'immense avantage de pouvoir facilement modifier le style d'un site en éditant un seul fichier au lieu de parcourir les pages HTML et changer le style de chacune des balises.

1 Cascade de styles

Il existe quatre niveaux de définition des styles appelé cascade :
1-Le style par défaut défini par chaque navigateur pour toutes les balises HTML
2-Le style défini dans une feuille de style séparée et inclut dans une page HTML grâce à la balise <link rel='stylesheet' href='URL d un fichier de styles .css' />
3-Le style peut être défini au niveau d'une page du site en introduisant la balise <style> </style> au niveau de la balise <head>
4-L'attribut global style permet de définir pour toutes balise HTML un style.

Ces quatre niveaux de définitions de style définissent ce qui nommé la cascade CSS. Un style défini au niveau feuille séparée remplace un style défini au niveau par défaut ; un style défini au niveau de la balise « style » remplace le style défini au niveau feuille séparée, et un style défini au niveau attribut remplace le style défini aux autres niveaux.

2 Format d'une règle de style

La feuille de style css ou la balise style est constituée d'un ensemble de règles CSS. Chaque règle CSS est constituée de deux partie : un sélecteur et des déclarations (les déclarations sont entre deux accolades). Enfin chaque déclaration est constituée d'une propriété CSS et d'une valeur séparée par deux point ' : ' et terminée par un point-virgule ' ;'. Comme pour le HTML/XHTML, les spécifications du CSS sont écrites par le W3C, la version 2.1 contient plus de 100 propriétés. Et la version CSS3 en rajoute des dizaines.

Format d'une reèle
sélecteur {
Nom-propriété-css-1 : valeur1 ;
Nom-propriété-css-1 : valeur2 ;
.
.
.
}

Exemple :
p {
font-size : 1em ;
color : black ;
}

2.1 Types de sélecteurs

A- sélecteur de type élément HTML : C'est le type le plus simple, car il est un ou plusieurs noms de balises séparée par une virgule. Dans ce cas le style s'appliquera à toutes les balises indiquées dans le sélecteur ainsi qu'aux balises qu'elles contiendraient éventuellement.

B- Sélecteur de type classe : Toute balise HTML possède un attribut class qui peut prendre comme valeur un ou plusieurs nom de classes séparés par un espace comme class= 'classe1 classe2, classe3'.
Dans ce cas le sélecteur prend la forme d'un point suivi du nom de classe. Par exemple .classe1 {…}
Si une propriété CSS est définie dans classe1, puis classe2 et classe3, c'est la dernière valeur dans classe3 qui sera appliquée en vertu des règles de cascades.
Une classe peut être générique (ie s'appliquer à toutes les balises si elle est déclarée avec juste le point '.' ou être spécifique à un élément HTML comme ceci : balise.classe1. Par exemple p.classe1{} s'appliquera seulement aux paragraphes qui ont classe1 dans leur attribut class. En fait .classe1 est équivalent à la notation *.classe1. L'étoile veut dire toutes les balises HTML. Par exemple si on veut afficher toutes les bordures des éléments il suffit de rajouter une règle sous la forme : * {border : 1px solid black; }, par exemple.

C-Sélecteur de type id où le format de la règle est : #nom_d_un_identificateur {....}
Dans ce cas la règle s'applique à une balise ayant l'attribut id='nom_d_un_identificateur'. Le id doit être unique dans une page donnée.

Exemple
#resume {
font-style : italic ;
}
s'appliquera à la balise
 bla bla bla

D- Sélecteur de type pseudo-classe. Ils sont utilisés pour ajouter des effets à certaines balises. Elles sont souvent utilisées pour ajouter des effets à l'élément ancre <a>.
Il existe quatre effets pour les éléments de type <a> qui sont dans l'ordre

a:link { color: blue; text-decoration: underlined; } -->lien non encore visité
a:visited { color: green; text-decoration: underlined; } -->lien visité
a:hover { color: red; text-decoration: none; } --> effet sur le lien quand la sourit le survole
a:active { color: purple; text-decoration: none; } -->effet sur le lien quand le bouton de la souris est pressé sur le lien

Remarque : Si on veut modifier ces styles, il faut les déclarer dans cet ordre.
Ces pseudo-classes peuvent être déclarées autrement comme ceci :
:link { color: blue; text-decoration: underlined; }
:visited { color: green; text-decoration: underlined; }
:hover { color: red; text-decoration: none; }
:active { color: purple; text-decoration: none; }

Et elles peuvent aussi être combinées avec des classes CSS qui contiennent le lien comme ceci :
a.main:link { color: blue; }
a.sidebar:link { color: grey; }
a.footer:link { color: white; }
Dans ce cas, le lien qui est contenu dans un élément de classe sidebar sera affiché en gris et celui qui dans la classe footer sera en blanc.

E-Sélecteur pour enfant : C'est un type de sélecteur qui s'applique à une balise enfant seulement si elle se trouve dans la balise parent.
Par exemple p em { font-weight: bold; color: red; } appliquera le style gras et la couleur rouge seulement aux éléments « em » contenus dans un éléments <p>. Si 'em' est dans <h1> par xemple, ce style ne s'appliquera pas.
Autre exemple avec #mainNav ul { font-family: Arial, sans-serif }, la police deviendra Arial seulement pour les éléments de type 'ul' qui se trouvent dans la balise qui porte le id='mainNav'.

3 Arrangement CSS (CSS Layout)

Les éléments HTML sont positionnés dans la fenêtre du navigateur suivant un modèle bien précis. D'abord, les éléments ayant la propriété display:inline comme et <a> ne provoquent pas un retour à la ligne ni avant ni après, tandis-ce que ceux en mode bloc (display:block) le font (par ex. <h1>, <p>, et <li>). Un élément en mode bloc prend toute la largeur disponible et provoque un retour à la ligne avant et après. Si le bloc a comme propriété display:none, alors le bloc ne sera pas affiché et l'espace prévu pour lui sera consommé par d'autres blocs. La propriété Visibility:hidden, par contre, permet de ne pas afficher un bloc mais l'espace prévu pour l'élément restera occupé, ie les autres blocs ne le prennent pas.

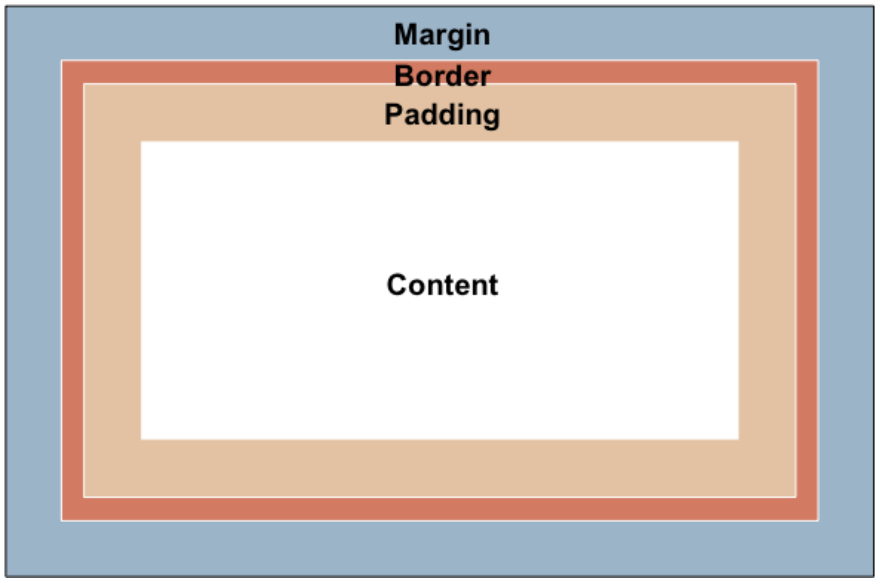
De plus, les éléments en mode bloc suivent le modèle en boîte, sont affichés dans un 'flux normal' qui est de gauche à droite et de haut en bas (sauf exception) et peuvent être positionnés soit par rapport à leur place dans le flux normal soit à la position du bloc les contenant.

3.1 Le modèle en boîte

D'autres part, les blocs suivent ce qu'on appelle le modèle en boite, i.e. ils possèdent une bordure, un espacement interieur entre leur contenu et la bordure (padding) et un espacement exterieur entre la bordure et les autres blocs (margin). Il existe une marge supérieure (top), droite (right), inférieure (bottom) et gauche (left). Et de même pour l'espacement intérieur (padding).

Tous ces paramètres du bloc peuvent être spécifié comme ceci :

#unbloc {
margin : 12px 3px 6px 9px;
padding : 12px 3px 6px 9px;
}
Ils peuvent aussi être spécifiés séparément comme



ceci :
margin-top : 12px ;
margin-right: 3px ;
margin-bottom:6px ;
margin-left : 3px ;
padding-top : 12px ;
padding-right: 3px ;
padding-bottom:6px ;
padding-left : 3px ;

De même si on a une seule valeur pour toutes les marges ou toutes les padding,
margin : 10px ; et padding:2px ; appliquera la même marge pour tous les côtés et le même espacement interne de tous les côtés.
Si on utilise deux valeurs pour margin come par exemple margin : 10px 5px ; le 10px sera valable pour margin-top et margin-bottom et 5px sera pour margin-right et margin-left. Le même résisonnemnt est valable pour padding. La valeur auto est une valeur spéciale, qui prend l'espace restant dans le bloc contenant.
La bordure elle suit la syntaxe : largeur trait couleur où largeur peut être une largeur exprimée en px, em, rem, in ou une constante telle dans parmis 'thin,medium, thick', le trait peut être dashed, dotted, solid, etc... et la couleur soit une valeur en héxadécile comme #CCFF00 ou bien une couleur prédéfinie comme blue, red, grey, lightgrey,etc...

3.2 Les flottants CSS

Les blocs HTML sont affichés dans un flux dit normal de gauche à droite et de haut en bas tant qu'il reste un espace suffisant pour dessiner un bloc. Si on souhaite forcer l'affichage d'un bloc à droite ou à gauche sur une ligne, on dispose de la propriété CSS 'float' qui peut être soit left soit right soit none (par défaut).
On peut faire flotter un élément horizontalement seulement, et le dans ce cas, les blocs qui viennent avant l'élément flotté ne sont pas affectés. Par contre l'élément suivant va flotter autour de l'élément flotté.
Dans cet exemple, le texte apparaitra à droite de l'image et en bas de l'image

```
<div style='width=100px ;'>
<h1> Le titre de la section qui ne sera pas affecté par le float</h1>
<img src= 'image.svg' style ='float:left;' alt='Image flottée à gauche' width=50 heigth=100 > <p> Ce texte est un bloc qui apparaitra au tour de l'image de son coté droit d'abord et bas ensuite si le texte dépasse les 100px de hauteur de l'image</p>
</div>
```

Si on veut faire apparaître le texte sur une ligne séparée on devra lui rajouter comme style 'clear:both'. La propriété clear appliquée à élément signifie : De quel côté, les autres éléments ne doivent pas flotter. Ainsi, clear:right veut dire que les autres éléments (qui suivent) ne doivent pas flotter à droite de l'élément qui a le style clear:right. Clear peut aussi prendre la valeur, left (pas de flottants à gauche), ou both (pas de flottants ni à gauche ni à droite), none et enfin inherited.

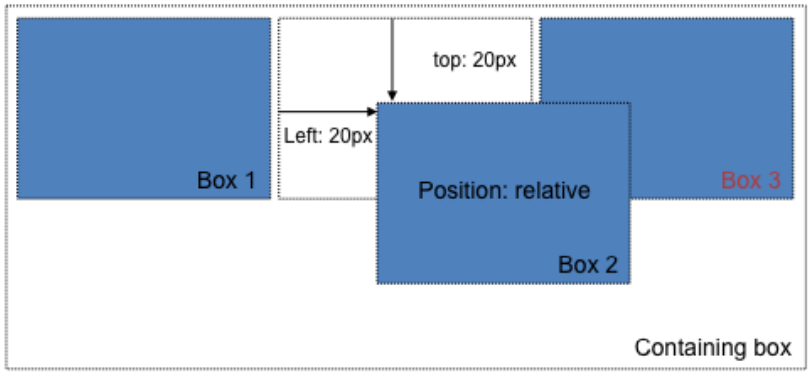
3.3 Le positionnement CSS

On peut positionner les éléments en mode bloc par rapport à deux origines (point de coordonnées (0 ,0)).
Lorsque le positionnement se fait par rapport au point d'origine du bloc dans le flux normal, on dit que le positionnement est relatif.
Lorsque le positionnement se fait pas rapport au point d'origine de son bloc parent (le bloc contenant), on dit que le positionnement est absolu.

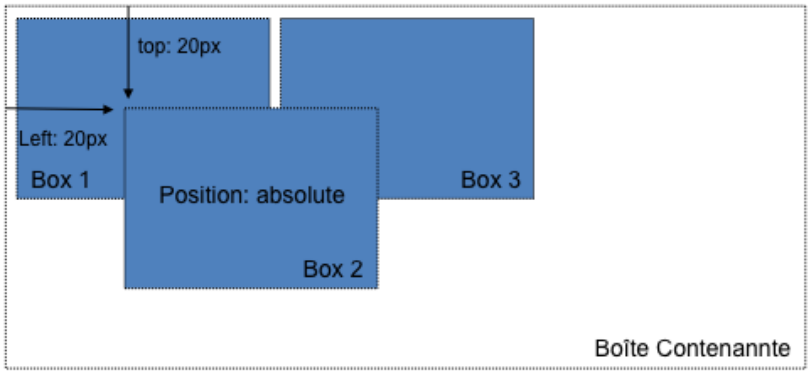
Pour les deux cas, on peut utiliser position:relative ou position:absolute, respectivement.
Les propriétés top (coordonnée en hauteur) et left (coordonnée par rapport au côté gauche) sont interprétées par rapport à l'origine indiquée par la propriété position.

Exemples :

```
#myBox {
    position: relative;
    left: 20px;
    top: 20px;
}
```




```
#myBox {
    position: absolute;
    left: 20px;
    top: 20px;
}
```



4 Conclusion

Les feuilles de styles permettent de décrire la mise en forme et le positionnement des blocs HTML dans la fenêtre du navigateur. Les styles permettent de modifier aisément la présentation de tout site web. Il existe plusieurs manières de définir des styles et de décider sur quels éléments ils s'appliqueront. Les styles CSS3 permettent en plus de créer des dégradés de couleur de dessiner des objets de toutes sortes (carré, disque, etc...) et des arrondis, choses très difficile à réaliser avec les CSS2.1, raisons pour lesquelles les développeurs faisaient appel à des images découpées.