HTML a ses limites. La mise en page est difficile. La taille des polices est limitée. La maintenance est difficile. Le contenu d'un document HTML n'est pas séparé des balises de mise en forme, toute modification est difficile.
CSS (Cascading Style Sheet ou Feuilles de style en cascade) est une norme et une recommandation de W3C. La première version CSS1 a été crée en 1996 et la deuxième CSS2 en 1998. CSS3 pour bientôt.
Le principe d'une feuille de style CSS est de séparer la mise en forme du contenu de document. Les instructions CSS de mise en forme sont rassemblées au début de document HTML dans la balise <head> … </head> ou carrément dans un ficher à part. Dans une feuille de style CSS nous décrivons la mise en forme des balises HTML.

Où mettre une feuille de style ?

Il y a trois possibilités :
1) CSS globale : à inclure aussi dans <head> … </head> à l'aide de
<style type="text/css"> … règles css …</style>
2) CSS importée : Une feuille de style est décrite dans un fichier (d'extension .css) à part et attaché au document (X)HTML à l'aide de
<link rel="stylesheet" type="text/css" href="fichier.css">
à placer dans <head> … </head>,
3) CSS Intra-lignes : réalisée à l'aide de l'attribut style, elle modifie la mise en forme d'une balise localement :
<balise style=" règles CSS "> … </balise>.

CSS est un langage où chaque mot de son vocabulaire est appelé **propriété** et chaque propriété à un ensemble de **valeurs** possibles prédéfinies. La syntaxe est

propriété : valeur ;

qu'on appelle **déclaration**. Une propriété est séparée de sa valeur par **:** et chaque déclaration se termine par **;** . Par exemple : color est une propriété qui prend comme valeur une couleur. On écrit alors color:red ; et pour rendre tous les titres de niveau 1 bleus, dans la feuille de style on écrit h1{color:red; }.

Une feuille de style CSS est un code inclus dans le document HTML ou dans un fichier relié au document HTML. Ce texte contient un ensemble de règles et chaque règle décrit la mise en forme associée à des balises HTML.
Sélecteur peut être une balise HTML, plusieurs balises HTML aux quelles on veut donner la même mise en forme, deux balises HTML dont la mise en forme voulue à l'une dépend de la relation de parentés avec l'autre, un style qu'on définit et qu'on voudra appliquer à plusieurs différentes balises, ou autres .(voir ci-dessous).
Structure d'une règle CSS

```
sélecteur {propriété1:valeur1 ;
           propriété2:valeur2 ;
          . . . }
```

Une feuille de style est formée d'une ou de plusieurs règles. Pour la lisibilité, il vaut mieux écrire chaque déclaration sur une ligne :
Exemple :

```
h1 { font-family: Arial;
     font-size: 12 ;
     color: red; }
```

Généralté sur le langage CSS

Casse : Le langage CSS est indépendant de la casse. En général on écrit en miniscule.
Commentaire : une ou plusieurs lignes /* ... */ comme en C, C++, Javascript :
Couleurs en CSS : peuvent être déclarées de 3 manières :
- **par nom de couleur** en anglais : aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, white, yellow; orange etc . . .
- **#rrggbb**, où rr, gg et bb sont des nombres hexadécimaux de 00 à FF, indiquant la portion de rouge, vert et le bleu respectivement.
- **rgb(r,g,b)**, où r, g et b sont soit des pourcentages soit des entiers de 0 à 255, pour le rouge, le vert et le bleu.

Exemple :

```
h1{color:#0000ff; background-color:red;}
h1{color: rgb(50%,50%,50%);
background-color:rgb(25,125,255);}
```

Unités de longueur en CSS
Les unités absolues :
in : inch (pouce, 1in=2,54cm)
pt : point (point, 1pt=1/72in, environ 0,0352 mm)
pc : pica (1pc=12pt)
mm : millimètre
cm : centimètre
La relation entre ces unités est :
1in = 2.54cm = 25.4mm = 72pt = 12pc.

Les unités relatives :
em : relative à la taille de police héritée. Exemple : `font-size: 1.2em` produit une taille 20% plus grande que la taille de la police héritée.
ex : hauteur du caractère "x" minuscule d'une police.
px : pixel, dépend de la résolution du périphérique d'affichage.
% : relative soit la taille de la police héritée pour la propriété CSS font-size, soit la longueur ou la hauteur

de l'élément parent pour les propriétés width ou height , de même pour les marges intérieures et extérieures.
Le séparateur décimal est le point et non pas la virgule.
Important : toute longueur en CSS **doit** avoir une unité.

div et span

Les balises, et n'ont aucune présentation particulière, et servent à s'attacher un style CSS.
div permet de définir un bloc de texte particulier.
span sert à définir un style intraligne.

```
<p style="color: blue">Ce texte est en bleu avec une
partie en <span style="color: red"> rouge </span></p>.
```

Selecteur class :
On peut nommer et définir un style indépendamment de toute balise HTML et qu'on pourra appliquer à différentes balises ou donner plusieurs mise en forme à une même balise.

```
.ma_classe { propriété : valeur; ... }
```

Un style peut aussi être défi spécifiquement à une balise :

```
balise.ma_classe { propriété : valeur; ... }
```

Pour utiliser la classe ma_classe :

```
<balise class="ma_classe"> .... </balise>
```

Exemple :

```
.style1{color : red;}
h1.style2{color : blue;}
<h1 class="style1"> … </h1>
<p class="style1"> … </p>
<h1 class="style2"> … </h1>
```

Selecteur id
on peut définir un style pour une balise unique : un document HMTL valide ne peut contenir deux balises avec le même id. id s'utilise surtout avec Javascript. On définit un id par # :

```
#un_style {propriété : valeur;}
```

et pour l'utiliser <balise id="un_style"> … </balise>
Exemple :

```
#bleu {color: blue;}
<p id="bleu"> Ce paragraphe va être écrit en bleu </p>
```

On peut avoir à la fois class et id dans une même balise.

Propriétés de styles de texte

text-align : définit l'alignement du texte, valeurs : left, right, center ou justify. Exemple : h1 {text-align: right ;}.
text-indent : définit un retrait dans la première ligne d'un bloc de texte spécifié en longueur ou en pourcentage (de la largeur de l'élément). Exemple : p {text-indent: 1.5cm ;}
text-decoration : définit une décoration du texte. Valeurs : blink (clignoter), underline (souligné), line-through (barré), overline (surligner) ou none . Exemple : h1 {text-decoration: blink ;}
text-transform : change la casse (majuscule, minuscule) du texte , valeurs : uppercase, lowercase, capitalize (met le premier caractère en majuscule) . Exemple : h1 {text-transform: capitalize ;}
color : change la couleur du texte.
Exemple : h1 {color: #000080 ;}
word-spacing : espace entre les mots. Valeurs : longueur ou normal. Exemple : p{word-spacing: 7pt ;}
letter-spacing : espace entre les lettres. Valeurs : unité de longueur ou normal.
Exemple : p {letter-spacing: 3pt ;}
line-height : interligne : espace entre les lignes du texte. Valeurs : unité de longueur, normal ou un nombre qui sera multiplié par la taille de la police pour donner l'interligne. Exemple : p{line-height: 3 ;}
white-space : gestion des espaces blancs. Valeurs : normal, pre (espace blancs préservés) ou nowrap (plusieurs espace blancs sont réduits à un seul) pre-line, pre-wrap. Exemple : p{white-space: pre ;}
vertical-align : alignement vertical d'un élément. Valeurs : longueur (valeur négatives possibles), baseline (valeur par défaut), middle : centré verticalement, bottom, sub (indice), super (exposant), **text-top** : alignement partie supérieure de l'élément, **text-bottom** : alignement partie inférieure de l'élément. Pourcentage. Exemple : img {vertical-align: bottom;}
direction : sens de l'écriture.Valeurs possibles : rtl ou ltr

Propriétés de polices de caractères

CSS fait complètement disparaître l'usage de …
font-family : pour indiquer une ou plusieurs polices, séparées par des virgules, police précise (Arial, Times, Helvetica, "verdana", 'courier'...) ou famille génériques (serif, sans-serif, cursive, fantasy, monospace), il est conseillé de toujours mettre une famille générique en dernier recours. les noms comportant des espaces doivent être mis entre guillemets.
Exemple : h3 {font-family: Georgia, "Times New Roman" ;}
font-size : changer la taille de police. Valeurs : xx-small, x-small, small, médium, large, x-large, xx-large, ou une valeur relative à la taille actuelle larger, smaller ou taille précise en pt, in, cm, px ou pourcentage de la taille actuelle (%). Exemple : p {font-size: 18pt}

font-style : définit le style de police. Valeurs : italic, oblique ou normal
font-variant : la variante de polices valeurs possibles : small-caps (petites capitales) ou normal (police normale).
font-weight : change l'épaisseur de police. Valeurs pour une valeur descriptive: normal, bold , bolder, lighter, ou une valeur numérique : (100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900). Très fin (100) jusqu'à très gras (900).
font-stretch : (pour fixer l'étirement de la police. Les valeurs possibles :
- narrower : assez étroit
- semi-condensed : un peu condensé
- condensed : condensé
- extra-condensed : très condensé
- ultra-condensed : extrêmement condensé
- normal : étirement habituel
- wider : étirement plus large
- semi-expanded : un peu étendu
- expanded : étendu
- extra-expanded : très étendu
- ultra-expanded : extrêmement étendu

font : raccourci pour les différentes propriétés de police facultatives suivantes : font-family , font-style, font-variant , font-size, font-weight. Dans l'ordre : font-style

font-variant font-weight font-size/line-height font-family. Exemple : p{ font:italic small-caps bold 30px Georgia, serif; }

Propriétés d'arrière-plans

background-color : définit la couleur de l'arrière-plan . Exemple : H1 {background-color: #000000} .
background-image : définit l'image de l'arrière-plan . Exemple : body {backgroung-image: url('fond1.gif')}
background-repeat : définit la façon de répéter l'image d'arrière-plan repeat (Répétition sur tout l'écran, valeur par défaut) ou no-repeat ou repeat-x (Répétition selon l'axe des X) ou repeat-y (Répétition selon l'axe des Y)
body {backgroud-image: url('fond.jpg'); background-repeat: reapeat-y}
background-attachment : spécifie si l'image d'arrière-plan reste fixe avec les déplacements de l'écran scroll ou fixed body {background-image: url('fond1.gif'); background-attachement: fixed|scroll}.
background-position : spécifie la position de l'image d'arrière-plan par rapport au coin supérieur gauche de la fenêtre {top ou center ou bottom , left ou center ou right} valeurs : longueur ou %. Exemple : body {background-image: url('img.gif'); background-position: right top}
background : raccourci pour les différentes propriétés d'arrière-plan p{background: image.gif fixed repeat}
Exemple :body{background : #00ff00 url('smiley.gif') no-repeat fixed center;}

Propriétés de tableau

caption-side : Position du titre de tableau. Valeurs :
top : en haut du tableau.
bottom : sous le tableau.
left : sous le tableau aligné à gauche.
right : sous le tableau aligné à droite.
Exemple :caption{caption-side:bottom;}
table-layout largeurs fixe/variable. Valeurs :
fixed : Les mentions de largeur priment sur le contenu des cellules.
auto : Le contenu des cellules prime sur les mentions de largeur (par défaut).
border-collapse : modèle de bordure de cellules mitoyennes
separate : les bordures de cellules de tableau ne coïncident pas (comme en HTML).
collapse : les bordures de cellules de tableau coïncident.
border-spacing : espace entre les bordures de cellules. Valeur : longueur.
empty-cells : Affichage ou non-affichage de cellules vides. Valeurs :
show : les bordures de cellules vides sont affichées.
collapse : les bordures de cellules vides ne sont pas affichées (par défaut).
Exemple : table{border-collapse:separate;
border-spacing:10px 50px;}

Propriétés de liste

list-style-type : type de puce ou de numérotation. Valeurs pour listes ol : decimal (1, 2, etc ...), lower-roman (i, ii, etc ...), upper-roman (I, II, etc ...), lower-alpha (a, b, etc ...), upper-alpha (A, B, etc ...), lower-latin, upper-latin, armenian, decimal-leading-zero (01, 02 etc …), georgian (an, ban, gan, etc ...), lower-greek (alpha, beta, etc ...)
Valeurs pour listes ul : disc, circle, square ou none
list-style-position: retrait des éléments de liste. Valeurs : inside = puces et numérotation dans le corps de la liste ;
outside = retrait à gauche des puces et numérotation (par défaut).
list-style-image:url("image") image de type .gif ou .jpg comme puce de liste.
list-style : regroupe toutes les propriétés de liste
Exemple : ul{list-style:square inside
url("/images/blueball.gif");}

Propriétés du Curseur (cursor)

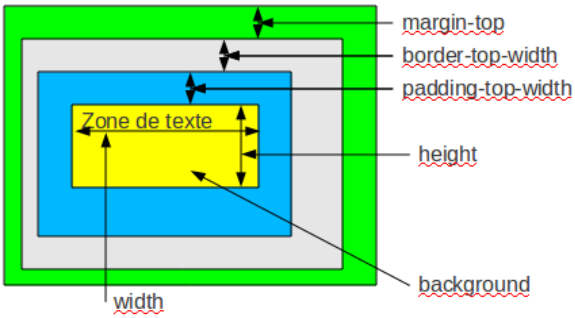
Définit un curseur pour un élément (X)HTML. le curseur prend la forme mentionnée lorsque la souris survole cet élément., Exemple : h1 {cursor : crosshair;}
default : curseur standard indépendant de la plate-forme.
crosshair : curseur de la forme d'une simple croix.
pointer : curseur de la forme d'une flèche.
move : curseur de la forme d'une croix signalant la possibilité de déplacer l'élément.
n-resize : curseur de la forme d'une flèche pointant vers le haut (n = nord).
ne-resize : curseur de la forme d'une flèche pointant vers le haut à droite (ne = nord-est).
e-resize : curseur de la forme d'une flèche pointant vers la droite (e = est).
se-resize : curseur de la forme d'une flèche pointant vers le bas à droite (se = sud-est).
s-resize : curseur de la forme d'une flèche pointant vers le bas (s = sud).
sw-resize : curseur de la forme d'une flèche pointant vers le bas à gauche (sw = sud-ouest).
w-resize : curseur de la forme d'une flèche pointant vers la gauche (w = ouest).
nw-resize : curseur de la forme d'une flèche pointant vers le haut à gauche (nw = nord-ouest).
text : curseur sous une forme qui symbolise du texte normal.
wait : curseur sous la forme d'un symbole signalant l'attente.
help : curseur sous forme d'un symbole qui signale de l'aide pour l'élément.
url(image), text; : curseur au choix, image doit être de type gif ou jpg.
Exemple : p{cursor : url("img1.gif"), url("second.jpg"), text; } on peut indiquer plusieurs, l'images doit être de petite taille.

Propriétés de dimensionnement

height : définit la hauteur d'un élément (titre, Image, Cellule de tableau etc). Valeurs : Une hauteur (nombre + unité), Un pourcentage, auto (valeur par défaut). Exemple : p1{height: 50px}, p2{height: 50%}.
width : Définit la largeur d'un élément : Image, Cellule de tableau, Élément de bloc. Valeurs : Une hauteur (nombre + unité), Un pourcentage, auto (valeur par défaut). Exemple : p1{height: 50px}.
min-height : définit la hauteur minimale d'un élément. img {height: 50%; min-height: 50px}
min-width : définit la largeur minimale d'un élément. img {width: 50%; width: 50px}

max-height et max-width : Comme min-height et min-width, mais concerne la valeur maximale.

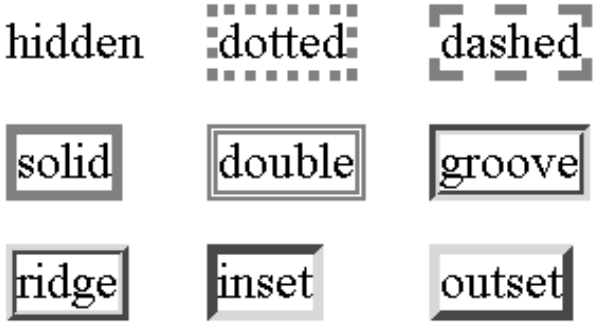
Les boites : bordures, paddings, marges



Tous les éléments HTML (div, span, h1, p, td, img, form, etc.) s'inscrivent dans une boîte rectangulaire dont on peut définir, pour chaque côté, l'épaisseur, la couleur et le style des bordures ainsi que l'épaisseur, le padding et la marge (gauche, droite, haut, bas), la couleur de fond… comme le montre l'image qui suit : des marges,

Style de bordures

les valeurs suivantes déterminent le style du trait de la bordure : none, solid, dotted, dashed, double, groove, ridge, inset ou outset. (voir figure)



border-top-style : style de bordure supérieur. Exemple : h3 {border-top-style: solid ;}.
border-right-style : style de bordure droit.
border-bottom-style : style de bordure inférieur.
border-left-style : style de bordure gauche.
border-style : regroupe les différentes propriétés de style de bordures. h1{border-style: solid dashed ;}

Propriétés d'épaisseur de bordures

border-top-width : épaisseur du bord supérieur. Valeurs : thin, medium, thick ou valeur numérique. Exemple : h3 {border-top-width: thin ;}.
border-right-width : épaisseur du bord droit. Valeurs : thin, medium, thick ou valeur numérique. Exemple : h3{border-right-width: medium ;} .
border-bottom-width : épaisseur du bord inférieur. Valeurs : thin, medium ou valeur numérique. Exemple : h3{border-bottom-width: thick;} .
border-left-width : épaisseur du bord gauche. Valeurs : thin, medium, thick ou valeur numérique. Exemple : h3 {border-left-width: 0.5cm;} .
border-width : regroupe les différentes propriétés de bordure.

Couleur de la bordure

border-top-color : couleur du bord supérieur.
border-right- color : couleur du bord droit.
border-bottom-color : couleur du bord inférieur.
border-left-color : couleur du bord gauche.
border-color : regroupe les différentes propriétés de couleur de la bordure.
Exemple :p{border-style:solid;
border-right-color:#ff0000 ;}
Border :regroupe toutes les propriétés de la bordure.
Exemple :p{border:5px groove pink;}

outline est le trait autour d'un élément pour marquer le "stand out".
outline-color : couleur .
outline-style : style.
outline-width : largeur.
Outline : regroupe les différentes propriétés de outline

Exemple :
p{outline-style:dotted; outline-color :dotted;}

Propriétés des marges

margin-top : marge supérieure. Valeur : longueur, pourcentage ou auto. Exemple : p1{ margin-top: 5px ;} .
margin-right : marge droite. Valeur : longueur, pourcentage ou auto.
Exemple : p1{margin-right: 5px ; } .
margin-bottom : marge inférieure. Valeur : longueur, pourcentage ou auto.
Exemple : img{ margin-bottom: 5px ; } .
margin-left : marge gauche. Valeur : longueur, pourcentage ou auto. Exemple : p1{margin-left: 5px ; } .
margin : regroupe toutes les propriétés de marges ci-dessus. Exemple : p{margin:2cm 4cm 3cm 4cm;} dans l'ordre haut, droit, bas et gauche. p{margin:2cm 4cm;} indique haut, droit. p{margin:2cm} indique que toutes les marges sont à 2cm.

Propriétés des Paddings

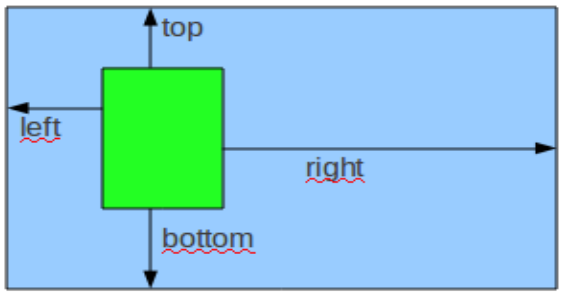
Le padding est la distance entre le contenu et la bordure d'un élément.
padding-top : padding haut. Valeur : longueur ou % h3 {padding-top: 3px}
padding-right : padding droite Valeur : longueur ou % h3 {padding-right: 3px}
padding-bottom : padding bas Valeur : longueur ou % h3 {padding-bottom: 3px}
padding-left : padding gauche Valeur : longueur ou % h3 {padding-left: 3px}
Padding : regroupe les différentes propriétés de remplissage

Positionnement des éléments

Il est possible grâce aux CSS de positionner au pixel près des images, des titres, etc et des blocs de textes grâce aux balises et .
Le positionnement absolu {position: absolute} se fait par rapport au coin supérieur gauche de la fenêtre du navigateur.

Exemple : texte

les propriétés de positionnements sont **left, top, bottom, right.**
left : distance enter le bord gauche de la zone à placer et le bord gauche de la fenêtre. Exemple : img{left:2cm;}



Propriété : position:static|absolute|relative|fixed.
position:absolute, le bloc est placé à l'endroit spécifié par les propriétés de positionnement.
position:relative, déplace l'élément relativement à position où ils devraient normalement se trouver.
position:relative + position:absolute
tous les éléments contenus dans un élément positionné relativement seront positionnés relativement à cet élément s'ils sont déclarés avec la valeur absolute de propriété.

Flottement

float:left|right|none : permet de faire flotter (faire couler du texte autour des images, bloc, tables etc) un élément à droite ou à gauche. Exemple : img {float: left}

La visibilité ou non du contenu d'un bloc est gouvernée par les propriétés suivantes :
visibility : visible|hidden| . . .affiche ou non l'élément .
clear: none ¦ left ¦ right ¦ both ; force un élément à couler sous un autre élément défini à l'aide de la propriété float.
Valeurs : left , le texte se poursuit en dessous de l'élément dé fini à l'aide de float (float: left)
right : de même que précédemment avec float: right
both : impose dans chaque cas la poursuite de l'élément.
None : pas de règle spécifique
clip: rect (<long.>, <long.>, <long.>, <long.>);

Blocs superposés

z-index : entier mise en avant ou en arrière plan de blocs partiellement ou totalement superposés.

display : block ¦ inline ¦ list-item ¦ none ¦ . . .

Débordement dans une boite

overflow: visible ¦ hidden ¦ scroll ¦ auto
spécifie ce qui se passe quand le contenu déborde des dimensions allouées au bloc.
h1{position:absolute;
width:6cm;
height:1cm;
top:50px;
right:400px;
background-color:red;
visibility: visible;}

Classes : .class

Une classe en CSS sert à définir un style dans le but de l'appliquer à plusieurs balises.
.ma_classe {propriété1:valeur1; propriété2:valeur2; ...}
Pour appliquer ce style à une balise :
<balise class="ma_classe"> .... </balise>
.classe1 { background : yellow; font-color: #000080 }
 ...
… <td class=" classe1"> texte </TD></TD> …

Identifiants : #mon_id

De même que class sauf qu'on ne peut l'utiliser qu'une seule fois dans la page. Les id fonctionnent exactement comme les classes. La syntaxe est :
#mon_id {propriété1:valeur1; propriété2:valeur2; ... }
Et pour l'appliquer :
<balise id="mon_id"> ... </balise>

Pseudo-classes

En CSS les pseudo-classes permettent d'accéder à certains éléments. Il constituent des classes prédéfinies contrairement aux classes. Il existe plusieurs types de pseudo-classes :

pseudo-classes de liens

Pour chacun des états de lien on peut définir une mise en forme particulière, à l'aide des pseudo classes
a:link : lien
a:visited : lien visité
a:active : lien sous-clique
a:hover : curseur survole le lien
Exemple : a:hover{color:red; text-decoration:none;}

Pseudo-classes pour toutes les balises

:hover : Définit un style particulier pour tout élément survolé.
:active : Définit un style particulier pour un lien activé.
:focus : Définit un style particulier pour un lien ou un élément de formulaire ayant le focus.
input:focus:hover {font-size: 12pt; color: red}
table:hover{background:red;}

Pseudo-éléments

:first-letter : définit un style particulier pour la première lettre d'un texte.
Exemple : P:first-letter { font-size: 200%; font-weight: bold; }
:first-line : finit un style particulier pour la première ligne d'un texte.
p:first-line { text-transform: uppercase }
:before Insère un contenu avant celui d'un élément.
p:before{content:"Paragraphe : ";}
:after Insère un contenu après celui d'un élément.
Exemple : p:after { content:"- Remember this";
background-color:yellow;
color:red;
font-weight:bold; }
Exemple : h1:before {content: counter(chapno, upper-roman) ". "}

Priorité

La concurrence entre plusieurs éléments de style peut provenir des différentes possibilités de localisation de feuilles de style :
- dans un fichier externe avec l'extension .css.
- dans la balise head du document.
- dans le body du document.

La règle de priorité est d'appliquer la feuille de style la plus proche de l'élément par le navigateur .
Il y a cependant moyen de contourner cette règle de priorité par la déclaration **!important**; par exemple :
body { background-color : #0000FF ! important;}

Héritage en CSS

Certaines propriétés CSS sont transmises d'éléments ancêtres à des éléments descendants. On appelle cela l'héritage (inheritance).
ol {color:#FF0000;}
li hérite de ol
body {font-family: Arial, sans-serif; font-size: 12px;}.
Ces propriétés s'appliquent à tous les descendants de body.
Toutes les propriétés ne sont pas héritées par exemple : border, margin etc.

Compteurs

Counter-increment : incrémente un compteur.
counter-reset : crée ou remet a zéro un compteur.
Exemple :
h1:before{ counter-increment:section;
content:"Section " counter(section) ". ";}
body{counter-reset:section;}
h1{counter-reset:subsection;}
h2:before{ counter-increment:subsection;
content:counter(section) "." counter(subsection) " ";}

Arborescence d'un fichier (X)HTML

Un document HTML est un arbre, avec des ancêtres, des descendants, des parents et des enfants.
descendant : un élément qui est l'enfant, le petit enfant ou plus éloigné encore d'un élément dans l'arbre du document.
ancêtre : un élément qui est le parent, le grand parent ou plus éloigné encore d'un élément dans l'arbre du document.
enfant : Le descendant direct d'un élément. Aucun autre élément ne peut s'intercaler entre les deux dans l'arbre du document.
parent : l'ancêtre direct d'un élément. Aucun autre élément ne peut s'intercaler entre les deux dans l'arbre du document.

Sélection simultanée

S1, S2, …, Sn {propriété1:valeur1; … } déclaration commune à plusieurs sélecteurs (qui sont, séparés par des virgules) . Exemple : h1, h2, h3, h4 {color:red ;}.

Sélecteur de descendance : S1 S2

S1 S2 {propriété1:valeur1; … } (sélecteurs séparés par un espace) la déclaration s'applique à S2 uniquement quand celui-ci est un descendants de S1.
Exemple : h1 code {...} la déclaration s'applique uniquement aux balises <code> ... </code> qui sont dans les titres </h1>.
ol li {...} les éléments li dans les listes

Sélecteur d'enfant : S1 > S2

S1 > S2 {propriété1:valeur1; … } la déclaration s'applique à S2 uniquement quand celui-ci est un enfant de S1.
Exemple : td > ul{...}
Déclaration s'applique uniquement à <ul> ... contenue dans une cellule <td> ... </td> de table.

Sélecteur de succession : S1 + S2

S1 + S2 {propriété1:valeur1; … }
la déclaration s'applique à S2 uniquement quand celui-ci est placé juste après S1.
Exemple : h1+p {color:red}
<h1> ... </h1>
 ...
 ...
Seul le premier paragraphe sera en rouge.

Sélecteur Universel : *

applique les règles CSS à tous les éléments de la page, même les propriétés de bloc
* {border-width : 0; } : supprimer toutes les bordures par défaut.
.ma_classe est équivalent à *.ma_classe
* {margin:0; padding:0;} remet à zéro les propriétés margin et padding de tous les éléments :

Sélecteur d'attribut

balise[attribut]{propriété1:valeur1; … } la déclaration modifie le style de <Balise> uniquement si elle contient l'attribut attribut. (La balise cible doit contenir l'attribut cité).
Exemple : table[border] {border-style: double }
<table border="...">

balise[Attribut = "Valeur"] la déclaration modifie uniquement le style des balises <balise> contenant l'attribut Attribut de valeur exactement "Valeur"**.**
Exemple :table[border="0"] {background-color:rgb(0, 120, 255)}
<table border="0">

balise[Attribut ~= "Valeur"] la déclaration modifie uniquement le style des balises <balise> contenant le mot "Valeur"**.**
Exemple : a[title~="site"] {outline-style: outset}
<a href="page.html" title="... site ...">

balise[Attribut |= "Valeur"] la déclaration modifie uniquement le style des balises <balise> commençant impérativement par le mot "Valeur"**.**
Exemple : a[title="Allez"] {font-color: aqua }
<a href="page.html" title="Allez ...">

Pseudo-classes de page

Le sélecteur @**page** permet de modifier les définitions de mise en page d'une page HTML (taille, marge, etc.) à l'impression, telles que les marges (margin-left, margin-top, margin-right, margin-bottom), la taille (size). Il faut alors imaginer la page web comme un ensemble de pages constituant un ouvrage papier.

Les pseudo-classes de page permettent ainsi de sélectionner les pages de gauche, de droite ou bien la première page d'un document.
Il existe différentes pseudo-classes de page :
@**page:left** : permet de définir les propriétés des pages de gauche.
@**page:left** { size: landscape; margin-left: 2cm; }
@**page:right** : permet de définir les propriétés des pages de droite.
@**page:right** { size:landscape; margin-left: 2.5cm; }

@**page:first** : permet de définir les propriétés de la première page d'un document.
@**page:first** { size: portrait; }

*** * * * * * * * * * * ***
*** * * * * ***
*** * ***
*****