



Remerciements

- Le matériel des diapositives est pris de différentes sources comprenant :



Feuille de style

- Qu'est-ce ?
 - = Ensemble de mises en forme génériques (position des éléments, aspect des textes à afficher, ...) associés à des balises.
 - Norme actuelle : CSS (Cascading Style Sheets)
 - Peut être enregistrée dans un fichier séparé de la page Web, et ainsi être utilisée par plusieurs pages.
 - Déclaration de la feuille de style ou de son utilisation dans l'entête (`< HEAD > ... </ HEAD >`)



Feuille de style

- Syntaxe

- Balise {propriété: valeur [; propriété: valeur ...]}*

- Exemples

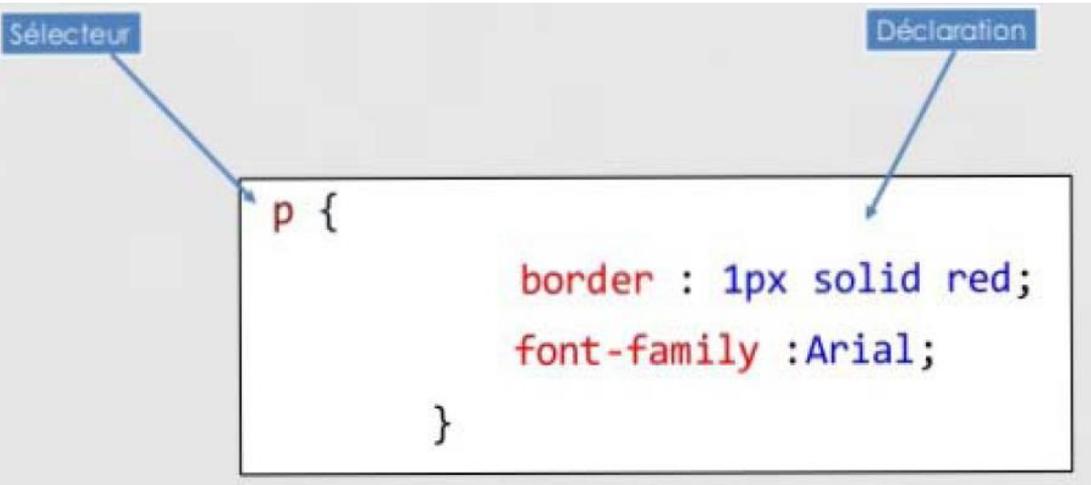
- H1 {font-size: 20pt; font-weight: bold ; color: red}
- H2 {font-size: 16pt; font-weight: bold ; color: #080000}
- P {margin-left: -20px; margin-right: -20px; margin-top: 30px }
- BODY {background: URL(http://my.server.com/pictures/back.gif); text-indent: 2cm }

- Trois possibilités pour les définir :

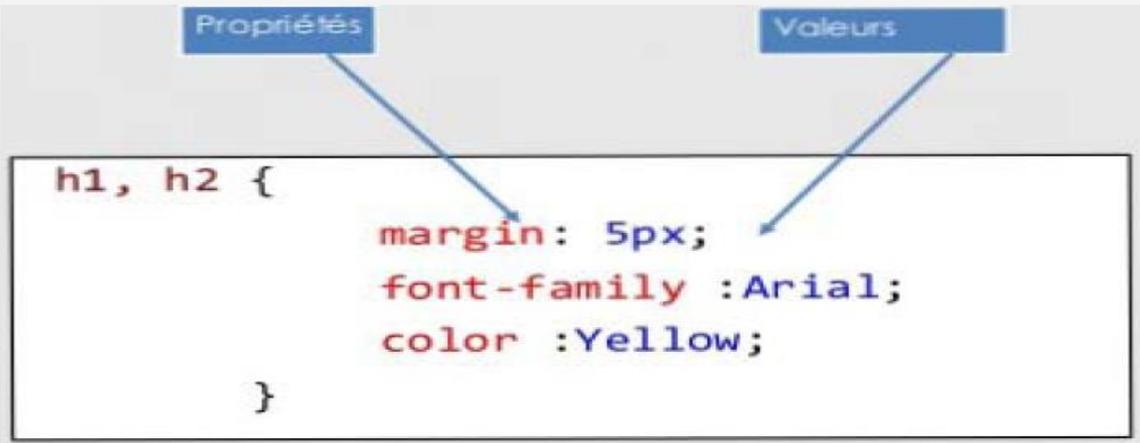
- Dans la page elle-même : - localement dans un élément
- globalement pour la page
- Dans un fichier séparé.



Syntaxe (suite)



```
<style type="text/css">  
  p {  
    font-family: 'Trebuchet MS';  
    margin-bottom: 5px;  
  }  
  h1, h2 {  
    color: blue;  
  }  
</style>
```





Syntaxe (suite)

- La feuille de style peut être dans un fichier externe dont l'extension est « css »
- Le lien entre le fichier HTML se fait à travers la balise link
- L'attribut « href » spécifie l'URL du fichier CSS
- L'attribut « type » doit être à « text/css »
- L'attribut « rel » doit être à « stylesheet »
- Un fichier HTML peut référencer plusieurs fichiers CSS
- Il vaut mieux mettre les feuilles de style dans des fichiers externes pour faciliter la maintenance

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



Définition de styles

- Définition locale
 - `<P STYLE="margin-left: 1in; margin-right: 1in; color: #0000FF">`
- Définition globale (dans l'entête `< HEAD > ... </ HEAD >`)
 - `<STYLE type="text/css">`
 - `P {font-size: 14pt; color: red ; text-align: center}`
 - `H1.red {color: #FF0000; font-size: 20pt}`
 - `H1.blue {color: #0000FF; font-size:20pt}`
 - `</STYLE>`
- Utilisation
 - `<H1 CLASS=red>This will be red< /H1>`
 - `<P> ... </P>`
 - `<H1 CLASS=blue>This will be blue< /H1>`



Définition de styles : positionnement

- Absolu, par rapport à la marge
 - Exemple : `{position:absolute; left:40px; top:75px;}` positionnera le texte :
 - à 40 pixels du bord gauche, et
 - à 75 pixels du haut de la fenêtre.
 - Ce type d'instruction permet de superposer des textes
- Relatif, par rapport au dernier élément affiché
 - Exemple : `{margin-top:20px; margin-left:300px;}` positionnera le texte :
 - 20 pixels plus bas et
 - 300 pixels plus à gauche
 - que le dernier texte affiché.



Définition de styles : polices

- font-size : taille de la police.
 - Valeurs possibles :
 - xx-small, x-small, small, medium, large, x-large, xx-large, larger, smaller
 - ou une taille exprimée en pixels (px), en points (pt)
- font-weight : graisse
 - Valeurs possibles :
 - normal, bold, bolder, lighter,
 - ou une valeur numérique comprise entre 100 et 900
- font-style : style d'écriture
 - Valeurs possibles :
 - normal, italic, oblique



Définition de styles : polices

- font-family : police
 - Valeurs possibles :
 - serif, sans-serif, cursive, fantasy, monospace
 - ou police précise
- color : couleur
 - Valeurs possibles :
 - #rrrvvbb (codage classique)
 - valeur symbolique
- background-color : couleur du fond
 - Valeurs possibles :
 - #rrrvvbb (codage classique)
 - valeur symbolique
- background-image : URL | none
 - URL de l'image de fond
- etc...



Sélecteurs

- Les sélecteurs permettent de sélectionner des éléments ayant un critère particulier : par exemple tous les éléments, appartenant à la même balise, même classe
- Les sélecteurs CSS sont sensibles à la casse

Sélecteur	Description	Exemple
Sélecteur universel	Sélectionne tous les éléments. Utilise le symbole « * »	* {} : tous les éléments
Sélecteur de type	Concerne tous les éléments appartenant à la même balise	h2 {} : tous les « h2 » p, span {} tous les paragraphes et les spans
Sélecteurs de classe	Concerne tous les éléments appartenant à une certaine classe (identifiés par l'attribut « classe ». Un élément HTML peut appartenir à plusieurs classes.	span.nuance : tous les span dont la classe est « nuance » .couleurVerte {} : n'importe quel élément ayant la classe « couleurVerte »
Sélecteur d'id	Concerne un seul élément identifié par son attribut « id »	span#spanId {} : le span qui a l'attribut id à « spanId » #unElement {} : n'importe quel élément ayant comme id « unElement »



Sélecteurs CSS (suite)

Sélecteur	Description	Exemple
Descendant direct	Les éléments qui sont descendants directs d'un autre élément. Le symbole « > » est utilisé. La fonction nth-child spécifie quel élément choisir.	div > p : paragraphes descendants directs d'une div div->a:nth-child(2) : le deuxième lien descendant direct d'un div div->a:nth-child(2n+1) : tous les 2ème + 1 descendants directs : 1 ^{er} , 3 ^{ème} , 5 ^{ème} ,...etc
Descendants indirects	Tous les éléments qui sont des enfants (pas forcément directs) d'autres éléments	div span.spanInterne : les span avec la classe « spanInterne » qui sont des descendants d'un div
Frère direct	Un éléments adjacent à un autre élément et qui sont au même niveau (symbole +)	h2+p : le paragraphe qui vient directement après un h2
Frère indirect	Les éléments se trouvant au même niveau qu'un autre élément (symbole ~)	li~li : un élément adjacent à un autre élément de liste. Exclut le premier.



Sélecteurs avancés

Sélecteur	Description	Exemple
:link	Liens pas encore visités	
:visited	Lien visité	
:hover	Élément qui est survolé par la souris	
:focus	Élément qui détient le focus	input:focus : contrôle qui reçoit le focus
[attribut=valeur]	Sélectionne des élément ayant des attributs avec une certaine valeur	a[target=_blank] {} : sélectionne les liens qui ouvrent dans une nouvelle fenêtre
[attribut^=valeur]	Sélectionne les éléments ayant un attribut commençant par une valeur	a[href^=http] : sélectionne les liens ayant une URL commençant par HTTP
[attribut*=valeur]	Sélectionne les éléments ayant un attribut contenant la valeur	a[href*=ynt] : sélectionne des liens qui contiennent « ynt » dans l'URL
:only-child	Sélectionne l'élément qui est l'unique descendant d'un autre élément	div p:only-child : sélectionne le paragraphe unique à l'intérieur d'un div



Sélecteurs avancés

Sélecteur	Description	Exemple
<code>nth-child(n)</code>	Nème descendant	<code>n-child(2)</code> : deuxième descendant <code>n-child(2n+1)</code> , tous les descendants impairs
<code>:first-child</code>	Premier enfant	
<code>:first-line</code>	Première ligne d'un paragraphe	
<code>:first-letter</code>	Première lettre d'un paragraphe	
<code>:before</code>	Ajoute du contenu avant l'élément	
<code>:after</code>	Ajoute du contenu après l'élément	
<code>not</code>	Sélectionne les éléments qui ne sont pas un élément donné	<code>p > not(a)</code> tous les éléments sauf les liens



[attr^="string Value"]

- Exemple:

- le code CSS

```
p.example{
  margin:0;
  padding:10px;
  color:#000;
}
p.example[title^="ess"]{
  color:#fff;
  background:#333;
}
```

- le code HTML

```
<p class="example"> je n'ai pas d'attribut title</p>
<p class="example" title="comment"> j'ai un attribut title mais il ne
commence pas par "ess"</p>
<p class="example" title="essai"> j'ai un attribut title commençant par
"ess"</p>
<p class="example" title="esson"> j'ai un attribut title commençant par
"ess" également</p>
```

- Résultat obtenu

je n'ai pas d'attribut title

j'ai un attribut title mais il ne commence pas par "ess"

j'ai un attribut title commençant par "ess"

j'ai un attribut title commençant par "ess" également



[attr\$="string Value"]

- Exemple:
 - le code CSS

```
p.example2{
  margin:0;
  padding:10px;
  color:#000;
}
p.example2[title$="sai"]{
  color:#fff;
  background:#045FB4;
}
```

- le code HTML

```
<p class="example2"> je n'ai pas d'attribut title</p>
<p class="example2" title="comment"> j'ai un attribut title mais il ne
finit pas par "sai"</p>
<p class="example2" title="essai"> j'ai un attribut title finissant par
"sai"</p>
<p class="example2" title="esson"> j'ai un attribut title finissant par
"sai" également</p>
```

- Résultat obtenu

je n'ai pas d'attribut title

j'ai un attribut title mais il ne finit pas par "sai"

j'ai un attribut title finissant par "sai"

j'ai un attribut title finissant par "sai" également



attr*="string Value"

- Exemple:

- le code CSS

```
p.example3{
  margin:0;
  padding:10px;
  color:#000;
}
p.example3[title*="val"]{
  color:#fff;
  background:#990000;
}
```

- le code HTML

```
<p class="example3"> je n'ai pas d'attribut title</p>
<p class="example3" title="comment"> j'ai un attribut title mais il ne
contient pas "val"</p>
<p class="example3" title="val"> j'ai un attribut title contenant au
moins "val"</p>
<p class="example3" title="evaluer"> j'ai un attribut title contenant au
moins "val" également</p>
<p class="example3" title="eval"> j'ai un attribut title contenant au
moins "val" également</p>
```

- Résultat obtenu

je n'ai pas d'attribut title

j'ai un attribut title mais il ne contient pas "val"

j'ai un attribut title contenant au moins "val"

j'ai un attribut title contenant au moins "val" également

j'ai un attribut title contenant au moins "val" également



Le combinateur d'adjacence directe

- Exemple:

- le code CSS

```
.example4 div{
  margin:0;
  padding:10px;
  color:#000;
}
.example4 div~p{
  color:#fff;
  margin:20px;
  width:200px;
  padding:5px;
  border:1px solid #333;
  background:#006644;
}
```

- le code HTML

```
<div class="example4">
  <div>je suis l'élément particulier div</div>
  <p> je suis un p qui suit le div (l'élément particulier)</p>
  <p>je suis un p qui suit le div (l'élément particulier)</p>
  <span>je suis un span</span>
  <p>je suis un p qui ne suit pas le div (l'élément
particulier)</p>
</div>
```

- Résultat obtenu

je suis l'élément particulier div

je suis un p qui suit le div
(l'élément particulier)

je suis un p qui suit le div
(l'élément particulier)

je suis un span

je suis un p qui ne suit pas le div (l'élément particulier)



:nth-child(expression)

- Exemple:
 - le code CSS
 - le code HTML

```
.exampleTable{
  width:100%;
  border:1px solid #444;
}
.exampleTable tr:nth-child(even){ /*tous les enfants aux numéros pairs*/
  background:#999999;
  text-shadow: 2px 2px 5px #111;
  color:#fff;
}
.exampleTable tr:nth-child(odd){ /*tous les enfants aux numéros impairs*/
  background:#990000;
  color:#fff;
}
.exampleTable tr:nth-child(3n){ /*tous les 3 enfants*/
  background:#045FB4;
  color:#fff;
}
.exampleTable tr:nth-child(7){ /*L'enfant numéro 7*/
  background:#006400;
  text-shadow: 2px 2px 2px #fff;
  color:#000;
}
```

```
<table class="exampleTable">
  <tr>
    <td>1ere ligne</td>
  </tr>
  <tr>
    <td>2eme ligne</td>
  </tr>
  <tr>
    <td>3eme ligne</td>
  </tr>
  <tr>
    <td>4eme ligne</td>
  </tr>
  <tr>
    <td>5ere ligne</td>
  </tr>
  <tr>
    <td>6eme ligne</td>
  </tr>
  <tr>
    <td>7eme ligne</td>
  </tr>
  <tr>
    <td>8eme ligne</td>
  </tr>
</table>
```

1ere ligne
2eme ligne
3eme ligne
4eme ligne
5ere ligne
6eme ligne
7eme ligne
8eme ligne



:nth-last-child(expression)

- Exemple:

- le code CSS

```
.exampleTable2{
  width:100%;
  border:1px solid #444;
}
.exampleTable2 tr:nth-last-child(odd){ /*tous les enfants aux numéros
impairs depuis la fin*/.
  background:#990000;
  color:#fff;
}
.exampleTable2 tr:nth-last-child(-n+2){ /*les 2 derniers enfants*/
  background:#045FB4;
  color:#fff;
}
.exampleTable2 tr:nth-last-child(7){ /*l'enfant numero 7 en partant de la
fin donc la 2eme ligne du tableau*/
  background:#006400;
  text-shadow: 2px 2px 2px #fff;
  color:#000;
}
```

- le code HTML

```
<table class="exampleTable2">
  <tr>
    <td>1ere ligne</td>
  </tr>
  <tr>
    <td>2eme ligne</td>
  </tr>
  <tr>
    <td>3eme ligne</td>
  </tr>
  <tr>
    <td>4eme ligne</td>
  </tr>
  <tr>
    <td>5ere ligne</td>
  </tr>
  <tr>
    <td>6eme ligne</td>
  </tr>
  <tr>
    <td>7eme ligne</td>
  </tr>
  <tr>
    <td>8eme ligne</td>
  </tr>
</table>
```

1ere ligne
2eme ligne
3eme ligne
4eme ligne
5ere ligne
6eme ligne
7eme ligne
8eme ligne



::first-line

- Applique la règle de style à la première ligne du texte de l'élément.
- Exemple:

```
p::
```

- La 1^{ère} ligne de ou des éléments "p" est mise en majuscule.



::first-letter

- Applique la règle de style à la première lettre du texte de l'élément.
- Exemple

```
p::first-letter { font-size: 2em }
```

- La 1^{ère} lettre de ou des éléments "p" a une taille de police de 2em.



::selection

- Applique la règle de style à la sélection du texte de l'élément faite par l'utilisateur.
- Exemple

```
p::selection { background:#006644 }
```

- A la sélection, le texte sélectionné aura une couleur d'arrière plan de valeur '#006644'.



Cascade

- Si plusieurs règles s'appliquent à un élément, les mécanismes suivants sont appliqués:
 1. Si deux règles sont identiques la dernière est prise
 2. Si une règle est plus spécifique que l'autre, elle est prise sur l'autre.
 - h1 est spécifique sur *, span#id est spécifique à span, Span.maClasse est spécifique à .maClasse.
- Si on veut prioriser une règle. on la marque avec le mot clé « !important »



Couleur

- Un mix entre trois valeurs: rouge, vert et bleu
- La propriété « **color** » définit la couleur du texte
- La propriété « **background-color** » définit la couleur de fond
- La propriété « **border-color** » définit la couleur de bordure
- Pour choisir une couleur:

http://www.w3schools.com/tags/ref_colorpicker.asp

- On peut affecter une couleur par son nom, par exemple « red » ou « darkcyan ». CSS prend en charge 147 noms de couleurs.
- Par valeurs hexadécimales de RGB. Par exemple #FFFF00.
- Par fonction rgb qui demande trois paramètres: rouge. vert et bleu. Cette fonction demande trois paramètres allant de 0 à 255



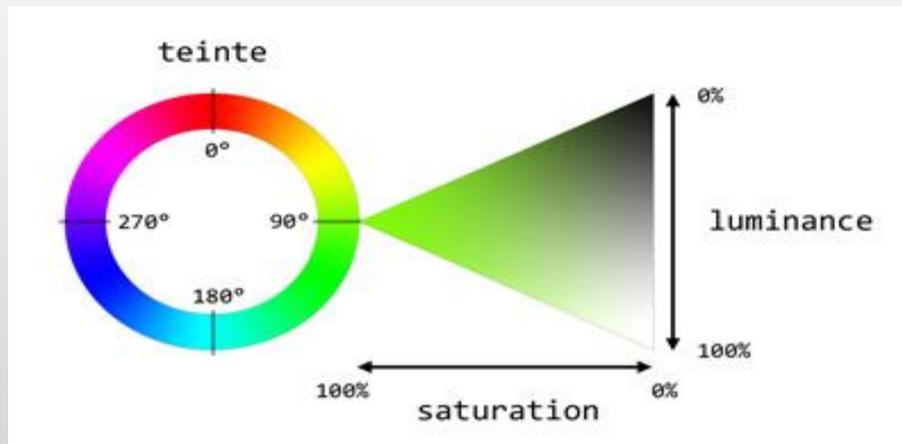
Couleurs standards

```
div#byName > span:nth-child(1) {  
    border-color: red;    }  
div#byHex > span:nth-child(1) {  
    background-color: #982222;}  
div#byShortHex > span:nth-child(1) {  
    border-color: #bb1;    }  
div#byRGB > span:nth-child(1) {  
    color: rgb(11,11,11);  
}
```



Couleurs HSL

- La fonction hsl (hue+saturation+lightness) définit une couleur à partir de la teinte, saturation et brillance
- La teinte est une valeur de 0 à 360
- La saturation est une valeur de 0 à 100%. 0% correspond à un niveau de gris. 100% correspond à la couleur d'origine.
- La luminosité est une valeur de 0 à 100%. 0% correspond au noir et 100% correspond au blanc.





Couleurs HSL- Exemple

```
div#byHSL > span:nth-child(1) {  
    background-color: hsl(120,30%,50%);  
    color: hsl(0,100%,50%);  
    border-color: hsl(0,100%,25%);  
}
```



Couleurs CSS3-transparence

- En plus des fonctions rgb et hsl. CSS3 définit deux nouvelles fonctions prenant un quatrième paramètre (alpha) qui sont rgba et hsla
- Alpha peut être entre 0 et 1. 0 pour complètement transparent et 1 pour complètement opaque.
- L'opacité peut être aussi définie en utilisant propriété CSS3 « opacity »
- Transparence : Exemples

```
div#byRGBA > span:nth-child(2) {  
    background-color: rgba(0,0,0,0.5);  
}  
div#byHSLA > span:nth-child(2) {  
    background-color: hsla(250,75%,50%,0.5);  
}  
div#byOpacity > span:nth-child(2) {  
    background-color: darkolivegreen;  
    opacity : 0.5;  
}
```



TEXTE: Polices de caractère-typologie

- Une police de caractère **serif** est une police avec des traits additionnels sur les caractères. Ces polices sont appréciées en lectures car ils sont très lisible. Les petits traits supplémentaires sont appelés « serifs ».
- les polices **sans-serif** sont des polices sans les traits supplémentaires. Elles sont très claires sur un dispositif comme l'écran.
- Les polices **monospace** sont des polices où tous les caractères ont la même largeur. Une des utilisations des polices monospace est le code source
- Une police courbée (**cursive**) est une police qui ressemble à l'écriture de main
- Une police fantaisiste (**fantasy**) est une police ayant un caractère décoratif



Exemples

Police serif cambria

Police sans serif Arial

Police mono-space consolas

Police cursive Monotype Corsiva

Police fantaisiste Haettenschweile



Attention

- Avant de choisir les polices de caractères pour la page web, il faut bien connaître certaines contraintes :
- Toutes les polices de caractères n'existent pas sur tous les ordinateurs.
 - **Conséquence** : si l'ordinateur de votre visiteur ne possède pas la police que vous avez choisie, une autre s'affichera ruinant probablement le design que vous espériez.
- Mac et PC ne possèdent pas toujours les mêmes noms de polices, ni même la même résolution.
 - **Conséquence** : bien vérifier les équivalences entre Mac et PC
- Tous les navigateurs ne gèrent pas le rendu des polices de la même façon.
 - **Conséquence** : là encore, n'espérez pas avoir le même rendu partout.



Famille Serif

- Les fontes de caractères de la famille serif assez courantes sont :
 - Times new roman (PC), Times (Mac)
 - Georgia (Mac/PC)
 - Palatino Linotype (PC), Palatino (Mac)

```
body {  
  font-family:"times new roman", times, serif;  
}
```

```
body {  
  font-family:georgia, serif;  
}
```

```
body {  
  font-family:"palatino linotype", palatino, serif;  
}
```



Famille sans-serif

- Verdana (Mac/PC)
- Arial (Mac/PC)
- Trebuchet (PC), Helvetica (Mac)
- Tahoma (PC), Geneva (Mac)

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

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

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

```
body {  
  font-family:tahoma, geneva, sans-serif;  
}
```



Taille du texte

- La taille du texte est spécifiée en utilisant **font-size**. La taille est faite en pixels.
- L'affichage du texte dépend de la résolution. Moins la résolution est grande, plus les caractères sont grands.
- La taille du texte peut être spécifiée en pourcentage (%) par rapport au parent. La taille par défaut dans les navigateurs est 100%.
- La taille des caractères peut être spécifiée en points pour de meilleurs résultats sur des supports imprimés.
- La taille peut être spécifiée en (em) où «em» correspond à la taille en cours de la police sur le navigateur. Par défaut 1em équivaut à 16px.
- 1 em est équivalent à la largeur de la lettre « m »
- La taille peut être spécifiée en utilisant les alias: *xx-small* : minuscule ; *x-small* : très petit ; *small* : petit ; *medium* : moyen ; *large* : grand ; *x-large* : très grand ; *xx-large* : gigantesque.
- L'alias *inherit* permet d'hériter du parent



Exemple d'héritage

```
body {  
  font-family:arial, sans-serif;  
  font-size:90%;  
}  
p {  
  font-size:90%;  
}
```



Exemple d'héritage

```
body {  
  font-family:arial, sans-serif;  
  font-size:100%;  
}  
h1 {  
  font-size:200%;  
}  
#footer p {  
  font-size:90%;  
}
```



Autres propriétés du texte

Règle	Description
font-weight	Permet de créer un texte en gras. (bold, bolder ou normal) sont des valeurs possibles.
font-style	Style de la police, par exemple italique, oblique ou normal
text-transform	Transforme en majuscule (uppercase), minuscule (lowercase) ou première lettre majuscule (capitalize)
text-decoration	Souligné (underline), surligné (overline) ou avec tiret (line-through) ou supprimer la ligne (none)
line-height	Hauteur de la ligne. Idéal pour centrer verticalement le texte.
letter-spacing	Espacement entre les lettres
word-spacing	Espacement entre les mots



Autres propriétés du texte- suite

Règle	Description
text-align	Alignement du texte. Left, right, center ou justify
vertical-align	Alignement vertical des éléments en ligne
text-indent	Indentation du texte
:first-letter	Première lettre
:first-line	Première ligne



Text-shadow

- Permet un effet d'ombre sur le texte
- Syntaxe : **texte-shadow dh dv couleur;**
 - dh : décalage horizontal.
 - dv : décalage vertical,
 - couleur : couleur de l'ombre
- Par défaut, un conteneur change sa taille pour s'adopter à son contenu
- Pour fixer la taille d'un éléments, il faut affecter les propriétés width (largeur) et (height) (hauteur)
- Les tailles peuvent être fixées en pixel, em ou pourcentage
- Pour définir des limites de redimensionnement: min-height, min-width, max-height, max-width



Contrôler le dépassement

```
<!DOCTYPE html>
<html>
<head>
<style>
div.scroll {
  background-color: #00FFFF;
  width: 100px;
  height: 100px;
  overflow: scroll;
}
```

```
div.hidden {
  background-color: #00FF00;
  width: 100px;
  height: 100px;
  overflow: hidden;
}
</style>
</head>
<body>
```

<p>The overflow property specifies what to do if the content of an element exceeds the size of the element's box.</p>

```
<p>overflow:scroll</p>
<div class="scroll">You can use the overflow property when you want to have better control of the layout. The default value is visible.</div>
```

```
<p>overflow:hidden</p>
<div class="hidden">You can use the overflow property when you want to have better control of the layout. The default value is visible.</div>
```

```
</body>
</html>
```

- lorsque les tailles sont définies, il y a un risque que le contenu d'un élément dépasse sa taille
- La propriété «**overflow**» contrôle ce dépassement
 - La valeur «*hidden*» permet de masquer les dépassements
 - La valeur «*scroll*» affiche des barres de défilements



Bordure et marges



- Un élément (boite) a la largeur et l'emplacement contrôlés par les propriétés «**border**», «**margin**», et «**padding**» .



Bordures

Règle	Description
<code>border-width</code>	Spécifie la largeur de la bordure. Cette largeur peut être en pixels ou en alias (<code>thin</code> , <code>medium</code> ou <code>thick</code>). CSS permet aussi de spécifier des largeurs différentes aux quatre bordures en utilisant <code>border-top-width</code> , <code>border-right-width</code> , <code>border-bottom-width</code> et <code>border-leftwidth</code> . Des raccourcis peuvent être utilisés : <code>border-width</code> : haut droite bas gauche.
<code>border-style</code>	Contrôle l'apparence de la bordure. Plusieurs styles sont possibles : <code>solid</code> (continu), <code>dotted</code> (pointillés), <code>dashed</code> , <code>insert</code> , <code>ridge</code> , <code>none</code> .
<code>border-color</code>	Spécifie la couleur de la bordure. Des couleurs différentes peuvent être spécifiées pour les quatre côtés.



Example

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.one {
```

```
    border-style: solid;
```

```
    border-width: 5px;
```

```
}
```

```
p.two {
```

```
    border-style: solid;
```

```
    border-width: medium;
```

```
}
```

```
p.three {
```

```
    border-style: solid;
```

```
    border-width: 1px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p class="one">Some text.</p>
```

```
<p class="two">Some text.</p>
```

```
<p class="three">Some text.</p>
```

```
<p><b>Note:</b> The "border-width"  
property does not work if it is used alone.  
Use the "border-style" property to set the  
borders first.</p>
```

```
</body>
```

```
</html>
```



Marges

- Les marges sont exprimées en pixel ou en pourcentage
- Les marges intérieures sont définies avec la propriété «padding»
- Les marges extérieures sont définies avec la propriété «margin»
- Les cotés peuvent avoir des marges différentes
- Une marge «auto » permet de centrer horizontalement un élément à condition qu' il ait une largeur.



Affichage des éléments

- La propriété « **visibility** » permet d'afficher ou de cacher un élément. Elle peut avoir les valeurs « *visible* » ou « *hidden* ».
- La valeur « *collapse* » de « **visibility** » s'applique aux tableaux et aux colonnes et permet de cacher une ligne ou une colonne sans affecter la disposition du tableau
- Un élément peut aussi être caché en utilisant la propriété « **display** » avec la valeur « **none** ». La différence avec **visibility** est que la valeur « **hidden** » laisse l'espace de l'élément réservé tandis que la valeur « *none* » le cache et ne réserve pas sans espace,



Propriété display

Valeur	Description
none	L'élément ne sera pas affiché
block	L'élément sera affiché en mode bloc
inline	L'élément sera affiché en mode en ligne
List-item	Permet d'afficher les éléments sous forme de liste.
table,table-column, table-row et table-cell	Affiche l'élément sous forme de tableau, ligne, colonne ou cellule



Bordures de type image

- CSS3 permet de définir de nouvelles bordures
- L'image spécifiée est découpée en neuf (09) portions qui sont utilisées pour les lignes et les coins.
- La propriété « **border-image-source** » spécifie l'URL de l'image à utiliser comme bordure.
- Les propriétés «**border-image-slice** » et « **border-image-width**» déterminent comment découper cette image.
- La propriété « **border-image-repeat**» détermine comment dessiner une bordure. « **stretch** » ajuste l'image à la taille, « **repeat** » répète l'image et « **round** » répète avec ajustement.



Exemple

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    background-color: lightgrey;
    border: 30px solid transparent;
    border-image-source: url('border.png');
    border-image-slice: 30;
}
</style>
</head>
<body>
<div>
This DIV element uses an image as a border.
</div>
<p>Here is the image used:</p>

<p><b>Note: </b>Internet Explorer 10, Opera 12, and Safari 5 do not support the border-image-source property.</p>
</body>
</html>
```



border-image-slice

```
div {  
  border: double orange 1px;  
  border-image-source: url(small-border.png);  
  border-image-slice: 27;  
}
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec eget ligula orci, sed hendrerit odio. Morbi bibendum volutpat porta. Ut nibh nisl, laoreet sed aliquam at, ornare vitae libero. Nam mattis nibh a turpis tristique pellentesque. Nam eu tincidunt urna. Nullam a justo at turpis egestas luctus. Etiam elementum pellentesque faucibus. Donec ac luctus justo. Integer ut arcu leo. Aliquam adipiscing tellus eu nisl aliquam pulvinar. Aenean tincidunt rhoncus velit id eleifend. Aliquam erat volutpat. Suspendisse pulvinar fermentum porta.