**Feuille de style CSS**

ou comment personnaliser son HTML



# A. Belaïd

Université de Nancy 2

En utilisant des feuilles de style



## Intérêt des feuilles de style :

* + Offrent des fonctionnalités puissantes de mise en page, mieux adaptées aux besoins des sites Web actuels
    - Étudiées pour aller loin dans la mise en page
  + Permettent de séparer le travail de rédaction et celui de présentation
    - Séparation du code : (+ général) et du style (+ personnel)
  + Permettent de séparer les fichiers de code et de présentation, d’où :
    - Plusieurs pages peuvent partager la même feuille de style
    - Maintenance et adaptation facilitée

## Attention :

* + Uniquement compatibles avec navigateurs Netscape Communicator  4.0 et Internet Explorer  4.0.

## Structure des règles de style

* + Une feuille de style est composée de règles qui peuvent s'appliquer soit
    - au document entier
    - à un ensemble d'éléments du document (e.g. les balises <p> du document)
    - à un élément particulier qui sera défini par l'attribut *class* ou *id*
  + Une règle de style est composée :
    - d'un sélecteur et
    - d'une déclaration
  + Exemple de règle : h1 { color: red }
* h1 est le sélecteur,
* la déclaration est comprise dans les accolades, la propriété *color* a la valeur *red*
* Structure des règles
  + Chaque règle de style peut comporter autant de couples propriété-valeur que voulus qui seront séparés par un ";"
  + Une propriété peut avoir plusieurs valeurs séparées par un espace

table { background-color : white ; border : 1px black solid }

* + Possible de grouper plusieurs sélecteurs ayant une règle de style identique, séparés par une virgule :

h1, h2, h3 { color: red }

## Appliquer des règles de style à un document

* + Plusieurs manières :

1. Grâce à une feuille de style intégrée au document
   * celle-ci sera située dans l'entête, entre les balises <head>
     + On utilisera alors la balise <style> :

<html>

<head>

<title>style interne au document</title>

<style type="text/css">

html, body { font-family: sans-serif; background-color: white } h1, h2 { color: blue }

p {text-align: justify }

</style>

</head>

<body> … </body>

</html>

* + Exemples : Exemple\_CSS\_1.html, Exemple\_CSS\_2.html, Exemple\_CSS\_7.html

1. Grâce à l'attribut style : appelé style en ligne : Exemple\_CSS\_4.html

<p style="text-indent:15px"> Un style en ligne...

</p>

1. En faisant appel à une feuille de style externe : balise <link> dans l'entête du document :

<html>

<head>

<title>appel à une feuille de style externe</title>

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

</head>

* + - La feuille de style aura comme extension .css et ne devra pas contenir de code HTML, Javascript ou autre
    - Exemple : Exemple\_CSS\_5.html

1. Une autre manière pour importer une feuille de style :
   * en utilisant @import dans la balise <style> :

<html>

<head>

<title>style importé</title>

<style type="text/css">

@import url(styleimporte.css);

html, body { font-family: sans-serif; background-color: white }

</style>

</head>

* + - Il est possible d'importer plusieurs feuilles de style

Styles en cascade : CSS



## C'est quoi ce style ?

* + Lorsqu'un style est appliqué à un élément, la plupart de ses propriétés se répercutent en cascade sur les éléments enfants contenus dans cet élément
  + La plupart seulement, car ces éléments enfants peuvent à leur tour écraser spécifiquement certaines propriétés

Style appliqué au <BODY>

Style appliqué à <P>

<B>

<P>

<H1>

<DIV>

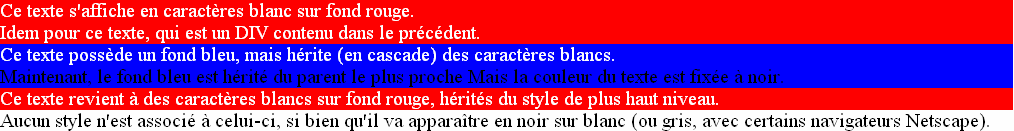
BODY

Style appliqué à <H1>

Styles en cascade



## Exemple : EffetEnCascade.html



* + Premier élément DIV :
    - Fond rouge appliqué au texte dans ce bloc ainsi qu'à son premier enfant
    - Dans le second enfant :
      * La couleur d'arrière-plan est fixée à bleu :

– l'élément hérite cependant des caractères blancs, puisque la propriété couleur d'avant-plan reste inchangée

* + - Enfin, l'enfant de cet enfant fixe la couleur d'avant-plan à noir (au lieu de blanc)
      * Remarquez que l'arrière plan n'est pas écrasé

Les mesures



## 2 types de mesures pour les tailles et dimensions

* + Mesures absolues :
    - le millimètre (mm)
    - le centimètre (cm)
    - le pouce (in) qui vaut 25,4 mm
    - le point (pt)
    - le pica (pc)
  + Mesures relatives : utiliser
    - le pourcentage (%)
    - la hauteur em (em)
    - le pixel (px)
  + L'emploi de telle ou telle unité dépendra de l'élément concerné (une marge, une police) et du média de sortie (imprimante, écran...)

## Pour l'affichage à l'écran

* + Préférez toujours une unité relative :
    - Une marge d'un cm s'affichera bien sur le vôtre mais paraîtra minuscule sur un autre et énorme sur un troisième
  + Préférez le pixel qui est relatif à la résolution de l'écran :
* sur un même écran, un pixel sera plus gros avec une résolution 800X600 par rapport à une résolution 1024X768
  + Les unités cm et le % sont calculés en fonction de l'élément parent
    - Dans cet exemple :

<body style="font-size: 12pt">

<p style="font-size:110%">Bla bla bla. </p>

</body>

* l'élément <p> est l'enfant de l'élément <body>
* Les caractères du paragraphe auront donc une taille de :

– 12pt 110% soit 13,2pt

## Exemple : mesure-relatives.html

<html>

<head>

<title>Untitled</title>

</head>

<body style="font-size:150%">

<p> Les caractères de cette page ont une taille de 150% définie dans l'élément body

</p>

<p style="font-size:80%"> ce paragraphe à une taille de 80%, sa taille sera calculée relativement à celle définie dans le body

</p>

</body>

</html>

* Autre exemple : positionnement dans la page

## Grâce au style, on va pouvoir positionner librement, de manière absolue ou relative des parties de texte

<html>

<body>

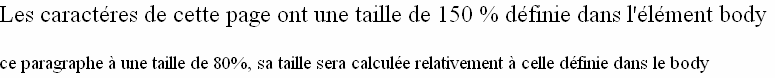
<p style="position: absolute; top: 99 px; left: 99 px; visibility : visible;z-index : 2;"> Mon texte

</p>

</body>

</html>

* Résultat : mesure-relatives.html



## Pour les couleurs

* + Quatre possibilités

1. Mots-clés (définis par le W3C)
   * 'aqua', 'black', 'blue', 'fuchsia' , 'gray', 'green' , 'lime', 'maroon', 'navy', 'olive', 'purple', 'red', 'silver', 'teal', 'white' et 'yellow'
2. Héxadécimale par paire :
   * #000000 (noir) #FF0000 (rouge), #00FF00 (vert) …
3. Héxadécimale abrégée
   * #000 (noir), #F00 (rouge), #0F0 (vert)
4. rgb :
   * rgb(o,o,o) et rgb(0%,0%,0%) donnent du noir
   * rgb(255,0,o) et rgb(100%,0%,0%) donnent du rouge pur

## Les feuilles de style de police (font)

* + font-family
    - Définit un nom de police ou une famille de police
      * Ex : h3 {font-family:Arial}
  + font-size
    - Définit la taille de la police
    - Xx-small / x-small / médium ou précise en points, inch…
      * Ex : P {font-size:72pt}
  + font-style
    - Définit le style de l'écriture
      * Ex : h3 {font-style:italic}
  + font-variant
    - Normal ou small-caps
      * Ex : P {font-variant:small-caps}

## Les feuilles de style de texte (text)

* + text-align
    - Définit l'alignement : à gauche, centré, …
      * Ex : h3 {text-align:center}
  + text-indent
    - Définit un retrait dans la première ligne
      * Ex : P {text-indent:1cm}
  + text-decoration
    - Définit une décoration du texte : souligné, barré…
      * Ex : a:visited {text-decoration:blink}
  + text-transform
    - Définit la casse du texte : minuscule, majuscule
      * Ex : P {text-transform:uppercase}
* Les feuilles de style d'arrière-plan (background)
  + background-color
    - Définit la couleur de l'arrière-plan

h3 {background-color:#000000}

* + background-image
    - Définit l'image de l'arrière plan

body {background-image:url(image.gif)}

* + background-repeat
    - Définit la façon de répéter l'image de l'arrière plan
      * repeat ou no-repeat ou repeat-x ou repeat-y

body {background-image:url(image.gif); background-repeat: repeat-4}

* + background-attachment
    - Définit si l'image d'arrière-plan reste fixe avec les déplacements de l'écran

body {background-image:url(image.gif); background-attachment: fixed}

Exemple : background-color.html

<HEAD>

<TITLE>Bordures</TITLE>

<STYLE>

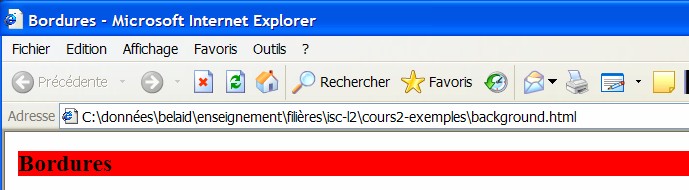
h3 {background-color:#FF0000}

</STYLE>

</HEAD>

<BODY>

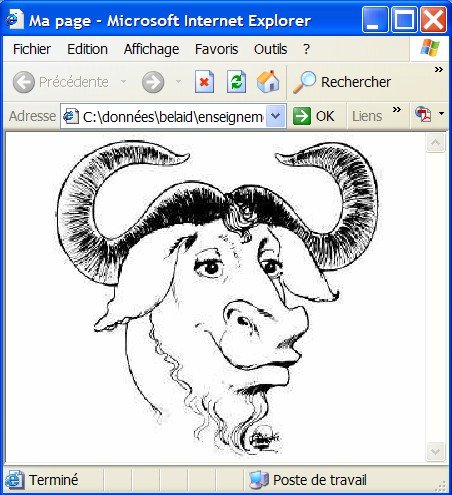
<H3>Bordures</H3>



## background-image1.html

<html>

<head>

<title>Ma page</title>

<style type="text/css"> body{background-image:url(gnu.jpg); background-repeat:no-repeat; background-position:center}>

</style>

</head>

<body>

</body>

</html>

© A. Belaïd 20

* background-image2.html

body {background-image: url(gnu.jpg); background-repeat: no-repeat; background-position: top center; background-attachment: fixed}

<body>

<p>1</p>

<p>2</p>

<p>3</p>

<p>4</p>

<p>5</p>

<p>6</p>

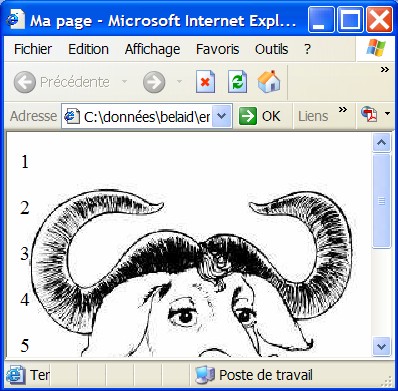
<p>7</p>

<p>8</p>

<p>9</p>

En activant l’ascenseur, les chiffres bougent et pas l’image

* background-image3.html

body {background-image: url(gnu.jpg); background-repeat: no-repeat; background-position: top center; background-attachment: scroll}

<body>

<p>1</p>

<p>2</p>

<p>3</p>

<p>4</p>

<p>5</p>

<p>6</p>

<p>7</p>

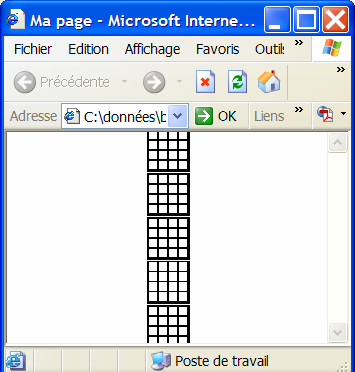
<p>8</p>

<p>9</p>

En activant l’ascenseur, les chiffres ne bougent pas, par contre l’image bouge

## background-image4.html

body {background-image: url(gnu.jpg); background-repeat: repeat; background-position: top center; background-attachment: scroll}

<body>

## Mettre des marges et dessiner des bordures

l r

top

margin

padding

border

content

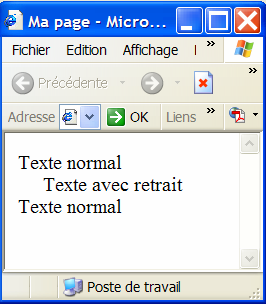
e i

f g

t h

t

bottom

* Marges et retraits (margin)
  + margin-top
    - Détermine la valeur de la marge supérieure en unité de longueur ou auto

img {margin-top:5px} margin-right, margin-left, …

* + Exemple : marge.html

<html>

<head>

<title>Ma page</title>

</head>

<body>

Texte normal<br>

<span style="margin-left:20px"> Texte avec retrait</span><br>

Texte normal<br>

</body>

</html>

© A. Belaïd

* span : balise permettant de structurer un élément de texte 25
  + Bordures :
    - Toute bordure prend en charge 3 propriétés :
      * Style, width, color
    - Style permet de modifier l'apparence visuelle de la bordure
      * Les valeurs possibles sont : solid, double, inset, outset, …
    - Exemple : border1.html

<HEAD>

<TITLE>Bordures</TITLE>

<STYLE>

largeur

couleur

P {border:solid 5px #C0C0C0;}

</STYLE>

</HEAD>

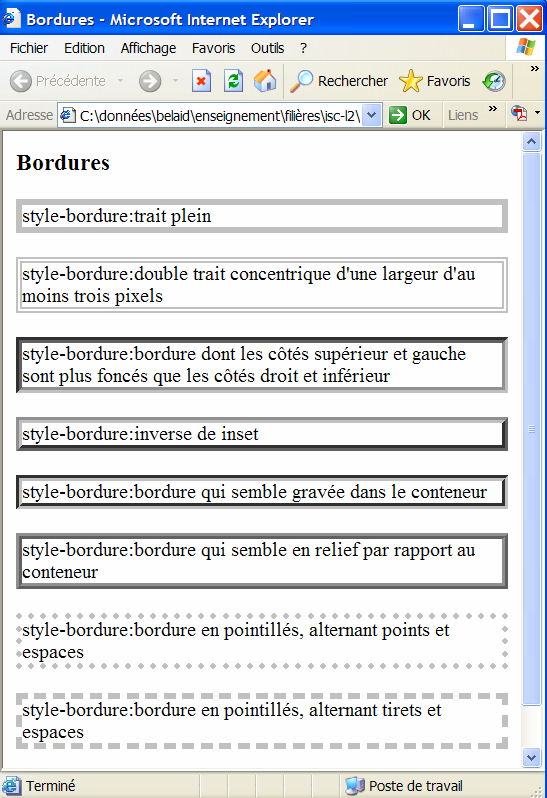
<BODY>

<H3>Bordures</H3>

style

<P style="border-style:solid">style-bordure:trait plein</P>

<P style="border-style:double">style-bordure:double trait concentrique d'une largeur d'au moins trois pixels</P>…



Les feuilles de style



## Bordures : border1.html

* + Autres feuilles de style de bordure (border)
    - border-top-width
      * Donne l'épaisseur du bord supérieur

table {border-top-width:thin}

border-right-width, border-bottom-width …

* + - Exemple : border2.html

<html>

…

<table>

<tr>

<td style="border-top-width:3px;

border-right-width:25px; border-bottom-width:3px; border-left-width:25px; border-color:red;

border-style:solid">

... votre texte ...</td></tr><table>…

* + Pour l'enrobage (padding)
    - padding-top
      * Valeur de remplissage haut entre l'élément et le bord

td {border-top:3px}

padding -right, padding -bottom…

* + - Exemple : padding.html

<html> …

<table>

<tr>

<td style="padding-top:0px;

padding-right:2px; padding-bottom:30px; padding-left:2px; padding-color:red; border-width:thin; border-style:solid">

... votre texte ...</td></tr><table>…

## Pour les listes

* + - list-style-type
      * Détermine le type de puce ou de numérotation

disc ou circle ou square ul {list-style-type :square}

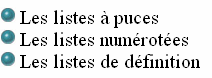
* + - list-style-image
      * Permet de remplacer les puces par une image

ul {list-style-image:url(image.gif)}

* + - list-style-position
      * Spécifie si les puces sont à l'intérieur ou à l'extérieur du texte

ul {list-style-position:inside}

* + Les listes (suite)
    - Exemple : liste.html

<html>

<head>

<title>Ma page</title>

</head>

<body>

<ul style="list-style-image:url(Pucebleue.JPG)">

<li> Les listes à puces</li>

<li> Les listes numérotées</li>

<li> Les listes numérotées</li>

</ul>

</body>

</html>

**Retour sur les sélecteurs**



Plusieurs formes

## Sélecteurs d'éléments ou de type

* + s'appliquent à la balise HTML à laquelle ils font référence :

<style type="text/css">

body {font-family: arial, helvetica, sans-serif; color: black; background-color:white;}

p {text-align: justify }

</style>

* + Autre exemple : h1 b {color: blue }
    - la couleur bleue sera appliquée à tout texte gras (balise <b>) contenu dans un titre de niveau 1 (balise <h1>)

## Sélecteurs de classe

* + Permettent de regrouper des éléments HTML spécifiques, relatifs à un même sujet
  + Syntaxe :

.nom-de-la-clase (le point avant la classe est important)

* + Exemple :
* Le site Web d'un magasin veut par exemple créer une classe pour la description de ses produits, une autre pour le mode d'emploi, une autre encore pour les commentaires des utilisateurs

<style type="text/css">

.description {color: red}

.mode { color:blue; font-size : 90%}

.comments { font-style : italic; color:green}

</style>

## Sélecteurs de classe

* + On pourrait l’utiliser de la façon suivante : select-class1.html

<p>Le site Web d'un magasin veut par exemple créer <span class=description> une classe pour la description de ses produits</span>, <span class=mode> une autre pour le mode d'emploi </span>, <span class=comments> une autre encore pour les commentaires des utilisateurs</span>

</p>

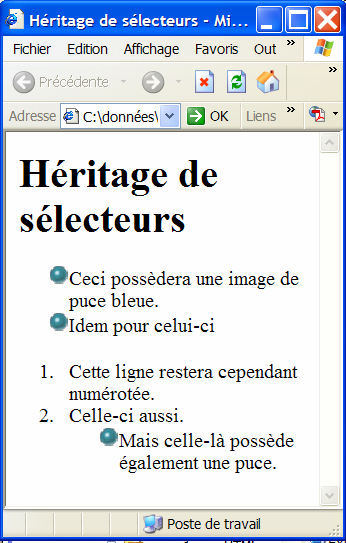
## Sélecteurs de classe (suite)

* + Cette notion de classe peut également servir à distinguer les mêmes éléments mais différemment
  + Exemple
    - Imaginons que l’on veuille différencier deux types de paragraphe: un en gras pour les paragraphes importants et un autre en italique pour les exemples

 les sélecteurs peuvent être déclinés en classes comme suit :

p.important {font-weight:bold;color :blue} p.exemple {font-style:italique;color :red}

* + - Exemple d’utilisation : Exemple\_CSS\_1.html

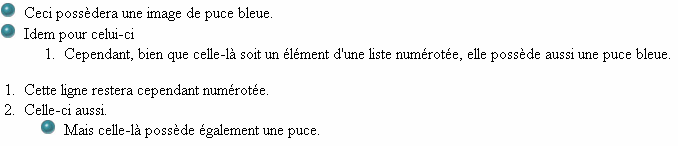
1. Sélecteurs de classe en cascade dans les listes
   * HTML prend en charge deux types de balises déterminant la nature des listes :
   * OL :
     + Balise de liste ordonnée, numérotant automatiquement les membres LI de la liste
   * UL :
     + Balise de liste à puces, faisant précéder chaque membre LI d'une puce
   * Comment faire la différence entre les deux LI ?
     + Créer une relation entre ancêtre descendant : > sur FireFox et blanc sur IE
     + OL>LI :
       - sélectionnera que les éléments des listes numérotées
     + UL>LI :
       - n'identifiera que les éléments d'une liste non ordonnée, ou liste à puces
   * Exemple :
     + On veut remplacer une simple puce par une image de puce plus sophistiquée nommée : Pucebleue.jpg
     + Si on définit la balise LI elle-même, on court le risque de voir des éléments de liste ordonnée afficher également le graphisme
     + En limitant la sélection aux listes à puces, on est certain que les membres des listes numérotées resteront inchangés :
   * Fichier : HeritageDeSelecteurs.html
   * Il est également possible d'indiquer qu'un élément est inclus dans un autre élément, à n'importe quelle profondeur :
     + Il suffit de séparer dans le sélecteur les deux éléments par un espace :

<STYLE>

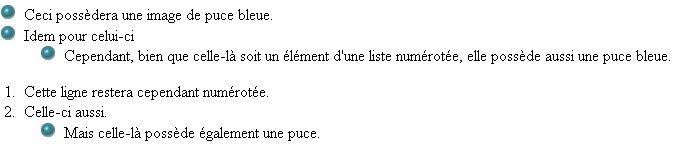
UL> LI {list-style-image:url(Pucebleue.jpg);}

</STYLE>

* + - Cela va s'appliquer à tout élément LI contenu dans une liste à puces, quelle que soit la profondeur de cet élément
    - Cela signifie que si la liste contient une liste ordonnée, tout élément de cette dernière possèdera également la puce bleue au lieu d'être numérotée
  + Exemple : sans :



* + Exemple : avec : HeritageDeSelecteurs2.html



* + On souhaite maintenant affecter toutes les balises possibles à l'aide d'une feuille de style
    - Exemple : assurer que tous les éléments de texte possèderont à l'affichage les mêmes tailles et type de police (courier 12 points)

 Recourir à l'opérateur étoile : "\*"

* + - Exemple

<STYLE>

UL LI {list-style-image:url(Pucebleue.jpg);}

\*{font-family:Courier;font-size:12pt;font-style:plain;} H1 {text-decoration:underline;}

</STYLE>

<H1>Annonce</H1>

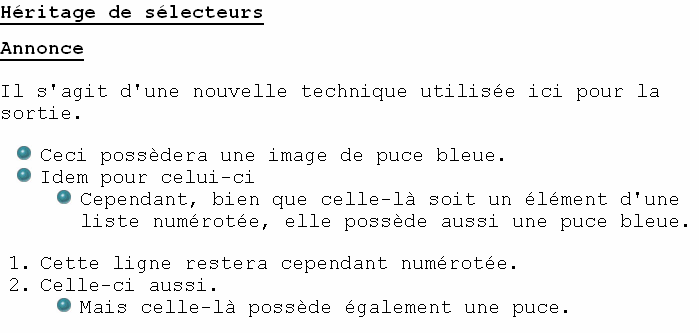
<P>Il s'agit d'une nouvelle technique utilis&eacute;e ici pour la sortie.</P>

<UL>

<LI>Ceci poss&egrave;dera une image de puce bleue.</LI>

<LI>Idem pour celui-ci</LI>

* + Résultat : HeritageDeSelecteurs3.html



* + Sélecteurs d'identificateurs (id)
    - Permettent d'appliquer une règle de style à un élément unique du document HTML
      * La syntaxe d'un sélecteur d'ID est la suivante : #nom\_ID { *style* }
      * Un tel style s'appelle de la manière suivante :

<BALISE ID="nom\_ID" > ... </BALISE>

* + - Exemple

<style type="text/css"> #special { font-size: 120% }

</style>

...

<p id="special">Paragraphe avec l'identificateur "spécial"</p>

1. Pseudo-classes
   * Proposées en plus des sélecteurs pour des éléments particuliers
     + Des pseudo-classes pour les liens :
       - Sans indication spéciale, les navigateurs affichent différemment les liens visités, ou non et cliqués
       - Si on veut modifier cet affichage, on utilise les sélecteurs : :link, :hover et :visited appliqués à la balise a (pour les liens)
     + a:link
       - permet d'attribuer des feuilles de style au lien (non-visité) a:link {text-decoration:none}
     + a:hover
       - permet d'attribuer des feuilles de style lors du survol du lien a:hover {font-weight:bold; color:purple}
     + a:visited
       - permet d'attribuer des feuilles de style au lien visité a:visited {color:blue}
   * Une pseudo-classe pour le premier élément (premier caractère, première puce…)
     + :first-child
       - S'applique sur le premier enfant du type donné d'un nœud
       - Exemple :

UL>LI {list-style-type:disc}

UL>LI:first-child {list-style-image:url(Pucebleue.jpg)}

* + - * Impose à tous les éléments de listes non ordonnées de débuter par une puce en forme de disque, sauf la première qui possède une image de puce bleue (Pucebleue.jpg)

– *Attention, cela n'est pas pris en charge par Internet Explorer 5.0.*

* + Des pseudo-classes pour gérer le curseur
    - :hover
      * S'applique quand le curseur se déplace sur la zone de l'élément pointé, avant que celui-ci n'ait été cliqué
    - :active
      * S'applique quand l'élément est activé
    - :focus:
      * Un élément possédant le focus peut recevoir la saisie, par ex. un rectangle

## Exemple : liens.html

<STYLE>

UL LI:hover {background-color:lightBlue;color:black;} UL LI:active {background-color:navy;color:white;} Input[type=text]:focus {border:solid 1px red}

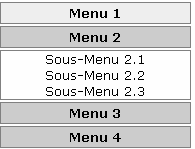
</STYLE>

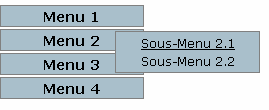
* + - En se déplaçant dans une liste d'éléments, chacun d'entre eux s'illumine à son tour, affichant un texte noir sur un fond bleu clair
    - Lorsque l'utilisateur clique sur l'un d'entre eux, celui-ci s'affiche en blanc sur fond bleu marine
    - Remarquez :
      * que si on se déplace, il peut rapidement repasser au stade

:hover avant de revenir à :active

* + - * si on clique sur une zone de texte, sa bordure devient rouge pour signaler qu'elle possède désormais le focus
  + Créer des menus en utilisant les styles CSS
    - menu-horizontal.html
      * En se mettant au-dessus du menu, il est grisé, du texte est affiché



* + - menu-vertical.html :
      * En passant sur un menu, il passe au gris clair
      * En cliquant sur Menu 2, des sous-menus apparaissent. Il sont cliquables



variante : menu-vertical2.html

* + - :first-letter et :first-line

<STYLE>

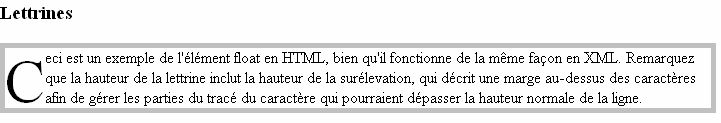
P:first-letter {float:left;font-size:36pt;font-family:Times Roman;}

</STYLE>

<BODY>

<P>Ceci est un exemple de l'&eacute;l&eacute;ment float en HTML, bien qu'il fonctionne de la m&ecirc;me façon en XML. Remarquez que la hauteur de la lettrine inclut la hauteur de la sur&eacute;levation, qui d&eacute;crit une marge au-dessus des caract&egrave;res afin de g&eacute;rer les parties du trac&eacute; du caract&egrave;re qui pourraient d&eacute;passer la hauteur normale de la ligne.</P>

</BODY>

* + - Permet de créer une lettrine : Lettrines.html
    - :before et :after
      * Servaient à ajouter un contenu au début ou à la fin de ce qui figurait déjà dans la balise

<STYLE>

P.Page {font-style:italic;}

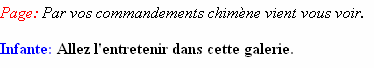
P.Page:before {content: "Page: ";color:red;} P.Infante {font-weight:bold;}

P.Infante:before {content: "Infante: ";color:blue;}

</STYLE>

<P class="Page"> Par vos commandements chim&egrave;ne vient vous voir.</P>

<P class="Infante"> Allez l'entretenir dans cette galerie.</P>



## Gestion des conflits

* + - En cas de conflit entre la feuille de style de l'auteur et celle du lecteur celle de l'auteur aura la priorité, sauf si le lecteur déclare sa règle de style !important :

<style>

p { font-family: arial !important;}

</style>

* + - Dans le cas où l'auteur et le lecteur déclarent leur style

!important, l'auteur prend de nouveau le dessus

* + - Dans les styles définis par l'auteur, le navigateur va d'abord calculer l'importance de chaque sélecteur selon ce principe :
      * sélecteur d'éléments (balises) 0 - 0 - 1
      * sélecteur de classe 0 - 1 - 0
      * sélecteur d'id 1 - 0 - 0

## Gestion des conflits

* + - Le poids du sélecteur se calcule de gauche à droite et la valeur de chaque sélecteur s'accumule, voici quelques exemples :
      * p {...} /\* 0-0-1 \*/
        + a un poids moins important que
      * p.classe1 {...} /\* 0-1-1 \*/
        + qui a lui même moins d'importance que
      * p#id1 {...} /\* 1-0-1 \*/
      * div p {...} /\* 0-0-2 \*/
        + l'emporte sur
      * p {...} /\* 0-0-1 \*/
      * p.class1#id1 {...} /\* 1-1-1 \*/
        + l'emporte sur
      * div p.class1 {...} /\* 0-1-2 \*/

## Gestion des conflits

* + - Ensuite, le navigateur va trier les styles selon leur ordre d'apparition
    - le principe à retenir est :
      * *Plus un style est déclaré tardivement, plus il aura d'importance*
    - Les styles importés (@import) auront le poids le plus faible
    - Viennent ensuite :
      * la feuille de style externe
      * La feuille de style intégrée
      * et enfin les styles en ligne
  + Gestion des conflits : exemple : conflit.html

<html>

<head> <title>Untitled</title>

<style type="text/css">

p {color: red; font-family: "Comic Sans MS"} p.petitvert {color: green; font-size: 13px; } p#times { font-size:24px; font-family: times }

</style>

</head>

<body>

<p>texte rouge avec police "comic sans"</p>

<p style="color:blue">texte bleu avec police "comic sans"</p>

<p class="petitvert">texte vert avec petite police</p>

<p class="petitvert" style="font-size:24px"> texte vert avec grande police</p>

<p id="times">texte rouge avec grande police "times"</p>

<p class="petitvert" id="times">

texte vert avec grande police "times"</p></body></html>

## Gestion des conflits : résultat



* + Commentaires
    - Les quatre premiers paragraphes sont affichés avec la police "Comic Sans" bien qu'elle ne soit spécifiée que pour le premier
    - Les autres paragraphes héritent en fait des propriétés définies pour le sélecteur <p> bien qu'elles aient chacune un style spécifique tant que ces propriétés n'entrent pas en conflit avec les leurs
    - La propriété de couleur "rouge" est ainsi écrasée par le style intégré du deuxième paragraphe
    - Le dernier paragraphe hérite ainsi des propriétés du sélecteur d'élément (<p>), du sélecteur de classe (.petitvert) et du sélecteur d'id (#times)
    - Les propriétés les plus spécifiques écrasant les autres :
      * La couleur rouge du sélecteur <p> est écrasée par le vert de la classe .petitvert
      * La taille de 13px de la classe est écrasée par celle de 24px de l'id #times
      * Tout comme la police "Comic Sans" est écrasée par la police "times"
  + Concept d'héritage
    - Le concept d'héritage s'applique également entre éléments parents et enfants
    - Exemple : conflit2.html

<html>

<head> <title>Héritage</title>

<style type="text/css">

body {font-family: arial, sans-serif} div {font-size:1.2em}

.rouge {color:red}

</style>

</head>

<body>

© A. Belaïd

<p> Ce paragraphe hérite des propriétés de l'élément body : police "arial"</p>

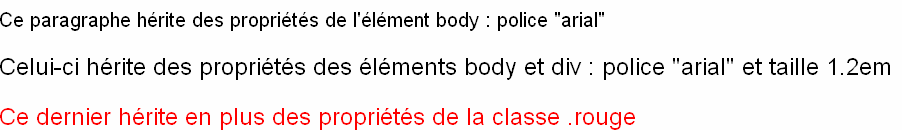
<div> <p> Celui-ci hérite des propriétés des éléments body et div : police "arial" et taille 1.2em </p>

<p class="rouge"> Ce dernier hérite en plus des propriétés de la classe .rouge</p> </div> </body> </html>

57

## Concept d'héritage

* + - Résultat :



[© A. Belaïd 58](#_TOC_250000)