

# Programmation client web

## cours 5 : CSS

# Plan

- ❖ Modèles de programmation via le service Web
- ❖ Protocole HTTP
- ❖ **HTML**
  - **BALISES**
- ❖ Cascading Style Sheet
- ❖ Javascript
- ❖ Applet java

# Langage HTML

- ❖ Langage à balises :
  - Une balise : une instruction de mise en forme
  - Une balise : contient des textes et/ou des autres balises autorisées
  - Une balise : structure le texte
  - Avant : une balise : mise en forme
- ❖ Navigateur : recherche des balises et interprétation
  - <NOM\_DE\_BALISE>....</NOM\_DE\_BALISE>
  - & (caractères spéciaux) : "&#code ascii;". Ex :
    - ◆ il y a : &amp; / & ; &lt; / < ; &gt; / >
    - ◆ &eacute; / é (note les caractères s'obtiennent par la &+lettre+accent; les accents étant acute, grave, circ, cedil, tilde et aussi les ligatures avec lig, e.g. &aelig; / æ).
- ❖ Liens hypertexte :
  - Chaque document HTML peut être représenté par un URL comme une ressource sur Internet.
  - Un document HTML peut contenir des liens vers les autres documents HTML ou les autres sources de données (multimédia, programme)
- ❖ Outil de validation : <http://validator.w3.org>

# Fichier HTML minimum

## ◆ Version html 4.01

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
    "http://www.w3.org/TR/html4/strict.dtd">

<HTML>
  <HEAD>
    <TITLE>
    </TITLE>
  </HEAD>

  <BODY>
  </BODY>

</HTML>
```

# Propriétés des Balises html

- ❖ Propriétés générales (mais pas systématiques)
- ❖ id, class (document-wide identifiers)
- ❖ lang (language information), dir (text direction)
- ❖ title (element title)
- ❖ style (inline style information )
- ❖ onclick, ondblclick, onmousedown, onmouseup, onmouseover, onmousemove, onmouseout, onkeypress, onkeydown, onkeyup (intrinsic events )

# Balises Meta (entêtes)

<b>&lt;meta name="xxx" ... &gt;</b>	<b>Description</b>	<b>Nb. Car.</b>	<b>Exemple</b>
<b>description</b>	<b>Description du site en une phrase de préférence.</b>	<b>200</b>	<b>&lt;meta name="description" content="HTML - Le Portail dédié à l'option IPW"&gt;</b>
<b>keywords</b>	<b>Liste de mots clés</b>	<b>500</b>	<b>&lt;meta name="keywords" content="html, xml, sgml, tags"&gt;</b>
<b>author</b>	<b>Auteur du site</b>	<b>-</b>	<b>&lt;meta name="author" content="imaginance"&gt;</b>
<b>identif-ur</b>	<b>URL de votre site</b>	<b>-</b>	<b>&lt;meta name="identif-ur" content="http://deptinfo.unice.fr/"&gt;</b>
<b>date-creation-yyyymmdd</b>	<b>Date de la création de votre site</b>	<b>-</b>	<b>&lt;meta name="date-creation-yyyymmdd" content="20060525"&gt;&gt;</b>
<b>date-revision-yyyymmdd</b>	<b>Date de la dernière modification</b>	<b>-</b>	<b>&lt;meta name="date-revision-yyyymmdd" content="20070717"&gt;</b>
<b>reply-to</b>	<b>Votre e-mail</b>	<b>-</b>	<b>&lt;meta name="reply-to" content="webmaster@allhtml.net"&gt;</b>
<b>publisher</b>	<b>Celui qui publie votre site</b>	<b>-</b>	<b>&lt;meta name="publisher" content="unice"&gt;</b>
<b>copyright</b>	<b>Copyright(s) de votre site</b>	<b>-</b>	<b>&lt;meta name="copyright" content="unice"&gt;</b>
<b>generator</b>	<b>Logiciels utilisés pour la création de votre site</b>	<b>-</b>	<b>&lt;meta name="generator" content="vi, gedit, openoffice"&gt;</b>

# Balises Meta (entêtes)

## ◆ Robots : outils des moteurs de recherche (référencement)

- Pour indexer toutes les pages de votre site ( vers lesquels vous avez placé des liens), utilisez la propriété *All* : `<META NAME="Robots" CONTENT="All">`
- En interdisant l'accès aux robots grâce à l'attribut *None*.
- Avec *Index*, votre page sera indexée aussitôt. Tandis que si vous mettez *Noindex*, elle ne sera pas indexée.
- Si le robot rencontre *Follow*, les destinations des liens sont indexées. Mais si vous écrivez *Nofollow*, le robot ne gardera pas mémoire de vos liens.
- Avec *Noimageindex*, les images d'une page déjà indexée ne le seront pas. Et *freesurvey* empêche de suivre les liens pointant vers les images.

<code>&lt;meta name="xxx" ... &gt;</code>	Description	Nb. Car.	Exemple
<code>revisit-after</code>	Indique la revisite du robot après n jours	-	<code>&lt;meta name="revisit-after" content="15 days"&gt;</code>
<code>category</code>	Catégorie de votre site (annuaire)	-	<code>&lt;meta name="category" content="internet"&gt;</code>
<code>robots</code>	Diriger le robot	-	<code>&lt;meta name="robots" content="index, follow"&gt;</code>

# Balises Meta (entêtes)

## ❖ **META : Le rafraîchissement et la redirection**

- Pouvoir faire recharger une page périodiquement (toutes les **n** secondes )

`<META HTTP-EQUIV="refresh" CONTENT="n">`

- Rediriger le navigateur vers un autre site :

`<META HTTP-EQUIV="refresh" CONTENT="n; URL=Mettez ici l'URL souhaitée">`

- HTTP-EQUIV : tout d'une requête http...

- ◆ **expires** : date limite d'expiration de vos pages (cache)

`<meta http-equiv="expires" content="wed, 06 sept 2006 12:00:00 gmt">`

- ❖ `<BASE HREF= ...>` donne la base de l'adresse URL qui sera placée devant les références relatives dans le document, de façon à ce que hors contexte les fichiers soient cherchés à l'adresse : `BASE adresse relative`

# Balises Meta (LINK)

- ❖ LINK : lien (uniquement dans l'entête)
  - HREF=adresse
  - rel et rev = alternate | appendix | bookmark | chapter | contents | copyright | glossary | help | home | index | next | prev | section | start | stylesheet | subsection
    - ◆ rel : relation entre ce document et le document cible
    - ◆ rev : relation entre le document cible et ce document
  - type = MIME\_TYPE (text/css ou image/gif ou ...)
  - target = \_blank | \_self | \_top | \_parent (frameset)
  - media = cf transparent suivant
  - charset : encodage de la cible (par défaut "ISO-8859-1")
  - hreflang : définit la langue du document cible
- ❖ non standard ? `<link rel="icon" href="ipw.png" type="image/png">`

# Type de Media

- ◆ **all**
  - convient pour tous les appareils ;
- ◆ **aural**
  - destiné aux synthétiseurs de parole. Voir les détails fournis dans le chapitre sur les feuilles de style auditives
- ◆ **braille**
  - destiné aux appareils braille à retour tactile ;
- ◆ **embossed**
  - destiné aux appareils à impression braille ;
- ◆ **handheld**
  - destiné aux appareils portatifs (typiquement ceux avec petits écrans, monochromes et à bande passante limitée) ;
- ◆ **print**
  - destiné à un support paginé opaque et aux documents vus sur écran en mode de prévue avant impression. Consulter le chapitre sur les médias paginés pour des informations sur les questions de mise en forme spécifiques à ceux-ci ;
- ◆ **projection**
  - destiné aux présentations en projection, par exemple avec des projecteurs ou des impressions pour des transparents. Consulter le chapitre sur les médias paginés pour des informations sur les questions spécifiques à ceux-ci ;
- ◆ **screen**
  - destiné principalement aux moniteurs couleurs ;
- ◆ **tty**
  - destiné aux médias utilisant une grille de caractères fixe, tels les télétypes, les terminaux ou les appareils portatifs aux capacités d'affichage réduites. Les auteurs ne devraient pas utiliser de valeurs exprimées en pixel avec ce type de média ;
- ◆ **tv**
  - destiné aux appareils du type télévision (avec ces caractéristiques : basse résolution, couleur, défilement des pages limité, sonorisé).

# Plan

- ❖ Modèles de programmation via le service Web
- ❖ Protocole HTTP
- ❖ HTML
  - BALISES
- ❖ **Cascading Style Sheet**
- ❖ Javascript
- ❖ Applet java

# CSS : introduction

- ❖ D'un point de vue "conception" (ergonomique et programmation), HTML non satisfaisant.
  - la maintenabilité non aisée, les modifier pour en changer l'aspect complique encore plus
  - d'autant plus à cause des répercussions
- ❖ Les feuilles de style (CSS = *Cascading Style Sheet*) permettent :
  - Principe en IHM : séparation du contenu de la présentation :
    - ◆ Séparation du contenu et de la mise en forme.
    - ◆ Modifier l'aspect d'un page ou d'un site sans en modifier le contenu.
    - ◆ Modifier facilement l'aspect d'une page.
  - Cohésion de la présentation tout au long du site (feuilles de style externes).
  - Réduire le temps de chargement des pages.
  - Correction de certains écueils d'HTML :
    - ◆ compréhensible, simple et logique par rapport au HTML.
    - ◆ contrôle des polices, de la distance entre les lignes, des marges et des indentations (sans devoir utiliser de tableaux...)
    - ◆ moins de code, mieux structuré, et donc des sites plus faciles à maintenir et à faire.

# CSS : introduction

- ❖ Utilisation de plusieurs feuilles de style (héritage)
- ❖ Syntaxe :
  - BALISE { Propriété : valeur }
  - Exemple : `H2 { color : green ; }` indique que tous les titres de niveau 2 sont en vert. Ici la propriété est color mais il en existe une bien plus !
- ❖ Une feuille de style CSS peut être définie de différentes façons :
  - Dans un document, il est possible de donner l'URI (url) d'un fichier externe. La syntaxe est la suivante :  
`<link rel="stylesheet" type="text/css" media="screen" href="fichier.css" />`.  
La balise LINK doit être utilisée entre les balises `<head>` et `</head>`
  - `@import "fichier.css"`, entre les balises `<style>...</style>` et avant la règle suivante
  - A l'intérieur du document lui-même (toujours dans la partie entête) par l'utilisation de balises STYLE : `<style type="text/css"> h1 { color: green ; }</style>`.
  - Par définition du style dans une balise qui supporte un ajout de style, comme par exemple `<li style="color:red;"> ...</li>`

# CSS : autres éléments de syntaxes

- ❖ Les commentaires
  - Entre /\* \*/
  - <!-- et --> acceptés (commentaire HTML) mais par de commentaire permet de cacher le CSS pour un navigateur qui ne les comprend pas
- ❖ Blocs : définis par { }
- ❖ Déclaration de propriétés :
  - Plusieurs lignes par bloc possible : séparation par ;
  - Dernière apparition d'une propriété valide qui compte
  - exemple-cours5/declaration1.html et exemple-cours5/declaration2.html
- ❖ Non prise en compte de ce qui est non valide
  - **Propriétés inconnues**
  - **Valeurs invalides** (unité !)
  - exemple-cours5/declaration3.html

# CSS : valeurs

## ❖ Les entiers et les nombres

### ❖ unités

ABBR.	UNIT	EXAMPLE	NOTES
cm	Centimeters	2.5cm	
em	Ems	3em	1 em equals the font point size.
in	Inches	1in	
mm	Millimeters	25mm	
pc	Picas	6pc	1 inch = 6 picas
px	Pixels	96px	
pt	Points	72pt	1 inch = 72 points
ex	X-heights	2ex	1 x-height usually equals the height of the letter x.

# CSS : valeurs

- ❖ **Les pourcentages** (par rapport à quelque chose)
- ❖ **URI** (url) : `url("http://deptinfo.unice.fr/~renevier/L2/ipw.png")`
- ❖ Les compteurs : `exemple-cours5/unite.html`
- ❖ Les chaînes
  - entre " "
  - Caractères spéciaux : `\codeascii`
  - `\22` ou `\"` ; `\27` ou `\'`
  - `"bla bla bl a 'bla' bla bla"` : valide
- ❖ Angles : `deg`, `rad` ou `grad` (feuilles de style auditives)
- ❖ Temps : `s` ou `ms` (feuilles de style auditives)
- ❖ Fréquence : `Hz` ou `kHz` (feuilles de style auditives)

# CSS : valeurs (couleurs)

KEYWORD	RGBHEX	RGB INTEGER	RGB PERCENTAGE
aqua	#00ffff	rgb(0,255,255)	rgb(0%,100%,100%)
black	#000000	rgb(0,0,0)	rgb(0%,0%,0%)
blue	#0000ff	rgb(0,0,255)	rgb(0%,0%,100%)
fuchsia	#ff00ff	rgb(255,0,255)	rgb(100%,0%,100%)
gray	#808080	rgb(128,128,128)	rgb(50%,50%,50%)
green	#008000	rgb(0,128,0)	rgb(0%,50%,0%)
lime	#00ff00	rgb(0,255,0)	rgb(0%,100%,0%)
maroon	#800000	rgb(128,0,0)	rgb(50%,0%,0%)
navy	#000080	rgb(0,0,128)	rgb(0%,0%,50%)
olive	#808000	rgb(128,128,0)	rgb(50%,50%,0%)
purple	#800080	rgb(128,0,128)	rgb(50%,0%,50%)
red	#ff0000	rgb(255,0,0)	rgb(100%,0%,0%)
silver	#c0c0c0	rgb(192,192,192)	rgb(75%,75%,75%)
teal	#008080	rgb(0,128,128)	rgb(0%,50%,50%)
white	#ffffff	rgb(255,255,255)	rgb(100%,100%,100%)
yellow	#ffff00	rgb(255,255,0)	rgb(100%,100%,0%)

Et plus encore : [http://www.w3schools.com/css/css\\_colornames.asp](http://www.w3schools.com/css/css_colornames.asp)

# CSS : Sélecteurs

- ❖ **Pattern : élément** (sélecteur de type)
  - tous les éléments « désignés » apparaissent avec le style défini entre { }
  - Ex :
    - ◆ `p {color: white ; background-color: black;}`
    - ◆ Écrit le paragraphe en blanc sur fond noir
- ❖ **Pattern : ,** (regroupement)
  - Regroupe les déclarations
  - Ex :
    - ◆ `H1, H2, H3 { color: blue;}`
    - ◆ Les titres de niveaux 1, 2 et 3 sont écrits en bleu
- ❖ **Pattern : \*** (sélecteur universel)
  - tous les éléments (\* veut dire n'importe quel élément) apparaissent avec le style défini entre { }
  - Ex :
    - ◆ `* {background-color: rgb(230,255,230);}`
    - ◆ Effet : mets le fond en vert
- ❖ [exemple-cours5/selecteurs1.html](#)

# CSS : Sélecteurs

- ❖ **Pattern : élément élément** (sélecteur descendant)
  - tous les éléments (en 2ième) contenus dans des éléments (en 1er) apparaissent avec le style défini entre { }
  - Ex:
    - ◆ `p span {color: red ; background-color: orange;}`
    - ◆ Tous les span dans des p sont écrit en rouge sur fond orange
- ❖ **Pattern : élément.classname** (sélecteur de classe)
  - tous les éléments « éléments » dont la class (attribut) est spécifiée à « classname »" apparaissent avec le style défini entre { }
  - Ex :
    - ◆ `p.special {color: white ; background-color: black;}`
    - ◆ Les paragraphes de class « special » sont écrits en blanc sur fond noir
- ❖ **Pattern .classname** (sélecteur de classe)
  - tous les éléments dont la class est spécifiée à « classname » apparaissent avec le style défini entre { }
  - Ex :
    - ◆ `.special2 {color: white ; background-color: black;}`
    - ◆ Les éléments de class « special2 » sont écrits en blanc sur fond noir
- ❖ **exemple-cours5/selecteurs2.html**

# CSS : Sélecteurs

- ❖ Pattern : #id (sélecteur d'id)
  - L'élément dont l'id (attribut, unique) est spécifiée à "id" apparaît avec le style défini entre { }
  - Ex :
    - ◆ #special {background-color:#ffffef; border:blue 2px solid;}
    - ◆ L'élément défini avec id="id" apparaît sur un fond légèrement jaune et encadré en bleu
- ❖ Pattern élément > élément (sélecteur d'enfant)
  - tous les éléments (en 2ième) contenus dans des éléments (en 1er) apparaissent avec le style défini entre { }
  - Ex :
    - ◆ p > span {background-color:#ffefff; border: green 2px solid;}
    - ◆ Tout span défini dans un p apparaît sur un fond légèrement violet et encadré en vert
- ❖ Pattern élément + élément (sélecteur d'enfant adjacent)
  - Chaque élément (en 2ième) immédiatement après l'élément (en 1er) apparaît avec le style défini entre { } : les deux éléments ont le même parent
  - Ex :
    - ◆ p + ul {background-color:#efffff; border-color:red; border-width:2; border-style: solid;}
    - ◆ Tout ul défini juste après un p apparaît sur un fond légèrement cyan et encadré en rouge
- ❖ exemple-cours5/selecteurs3.html

# CSS : Sélecteurs

- ❖ Pattern : élément[attribut] (sélecteurs d'attribut)
  - Chaque élément p qui a un attribut id (quelque soit la valeur de l'attribut apparait avec le style défini entre {}
  - Ex :
    - ◆ `p[id]{color: white ; background-color: black;}`
    - ◆ Chaque p qui a un id est écrit en blanc sur fond noir
- ❖ Notez qu'il est possible de restreindre le style :
- ❖ Pattern : élément [att="value"]
  - le style ne vaut que quand l'attribut att a la valeur précise "value"
- ❖ Pattern : élément [att~="value"] :
  - le style ne vaut que quand la valeur de l'attribut att contient "value"
- ❖ [exemple-cours5/selecteurs4.html](#)

# CSS : pseudo-classes

- ❖ **élément:first-child**
  - correspond au premier élément (celui-ci) enfant d'un autre élément.
  - Si l'élément est le premier fils d'un autre
- ❖ **Pour une ancre (a) :link et :visited**
  - généralement représentation différente des liens visités de ceux qui l'ont déjà été.
  - Equivalent CSS (déconseillé)
    - ◆ :link s'applique aux liens qui n'ont pas été visités ;
    - ◆ :visited s'applique aux liens déjà visités
- ❖ **pseudo-classes dynamiques :active , :hover et :focus**
  - :hover pour l'élément désigné par un pointeur (souris)
  - :active pour l'élément actif (entre le pressed et le released)
  - :focus pour l'élément qui reçoit l'attention (celui-ci acceptant les événements du clavier ou d'autres formes d'entrées de texte)
- ❖ **:lang**
  - Pour préciser des éléments (en association avec l'attribut lang)

# CSS : pseudo-éléments

- ❖ `:first-line` : première ligne
- ❖ `:first-letter` : première lettre
- ❖ `:before` et `:after`
  - `:before` et `:after` pour à insérer un contenu généré avant ou après celui d'un élément.
    - ◆ `H1:before {content: counter(chapno, upper-roman) ". "}`
  - En combinaison avec les pseudo-éléments `:first-line` et `:first-letter`, les pseudo-éléments `:before` et `:after` s'appliquent à la première lettre ou la première ligne, y compris le texte inséré.
    - ◆ Exemple(s) :
      - `P.special:before {content: "Spécial ! "}`
      - `P.special:first-letter {color: gold}`
      - Ceci donnera une couleur or au "S" de "Spécial !".
- ❖ [exemple-cours5/pseudoElt.html](#)

# Assignment, cascade et héritage

- ❖ Une fois qu'un agent utilisateur (navigateur web) a parcouru un document et en a construit une arborescence, celui-ci, pour chacun des éléments de l'arborescence, doit distribuer une valeur pour chacune des propriétés concernées du type de média visé.
- ❖ La valeur finale d'une propriété est le résultat d'un calcul en trois étapes : celle-ci est déterminée par la spécification (la valeur "spécifiée"), puis résolue dans une valeur absolue si nécessaire (la valeur "calculée") et finalement transformée en accord avec les contraintes de son contexte (la valeur "réelle").

# Valeurs spécifiées

- ❖ Les agents utilisateurs (navigateur web) doivent d'abord appliquer une valeur spécifiée à une propriété selon le mécanisme suivant (dans l'ordre de priorité) :
  - 1. Si la cascade donne une valeur, utiliser celle-ci ;
  - 2. Autrement, si la propriété est héritée, utiliser la valeur de l'élément parent, qui est en général une valeur calculée ;
  - 3. Autrement, utiliser la valeur initiale de la propriété. Cette valeur initiale est définie pour chaque propriété.
- ❖ La racine de l'arborescence du document n'ayant pas de parent, et ne pouvant donc pas hériter de valeurs, c'est la valeur initiale qui est utilisée si nécessaire.

# Valeurs calculées

- ❖ Les valeurs spécifiées peuvent être absolues (ex. les valeurs 'red' et '2mm' ne sont pas relatives à une autre valeur) ou relatives (ex. les valeurs 'auto', '2em' et '12%' se rapportent à une autre valeur). Une valeur absolue ne requiert aucune opération supplémentaire pour établir sa valeur calculée.
- ❖ Par contre, une valeur relative doit subir une transformation qui aboutit à sa valeur calculée : une valeur en pourcentage doit être multipliée par une valeur de référence (celle-ci est précisée pour chaque propriété), une valeur avec une unité relative (em, ex, px) doit être multipliée par les tailles de police ou de pixel appropriées, les valeurs 'auto' doivent être calculées selon les formules données pour chaque propriété, certains mots-clés ('smaller', 'bolder', 'inherit') doivent être remplacés en accord avec leurs définitions.
- ❖ Dans la plupart des cas, les éléments héritent de valeurs calculées. Cependant, certaines propriétés ont des valeurs spécifiées qui sont héritées (ex. la valeur numérique de la propriété 'line-height'). Dans les cas où les éléments enfants n'héritent pas de valeurs calculées, ceci est décrit dans les définitions des propriétés.

# Héritage

- ❖ Les éléments enfants héritent de certaines valeurs de leurs éléments parents dans l'arborescence du document. Chacune des propriétés définit si elle est héritée, ou non. Supposons un élément accentué (ici EM) dans un élément H1 :
  - `<H1 style="color:blue">Le titre <EM>est</EM> important !</H1>`
  - Si aucune couleur n'est précisée pour l'élément EM, le mot accentué "est" héritera de la couleur de l'élément parent, ainsi l'élément H1 ayant une couleur bleu, EM le sera également.
- ❖ Pour appliquer une propriété de style "par défaut" à un document, un auteur peut l'appliquer à la racine de l'arborescence du document. On peut utiliser les éléments HTML ou BODY pour cet usage.
- ❖ `exemple-cours5/heritage.html`

# inherit et !important

- ❖ La valeur 'inherit' provoque l'héritage des valeurs par les propriétés. Ceci s'applique également aux propriétés dont la valeur n'est normalement pas héritée.
- ❖ !important
  - CSS essaye de préserver un équilibre entre les prérogatives de l'auteur et celles de l'utilisateur. Par défaut, les règles d'une feuille de style de l'auteur surclassent celles de l'utilisateur (voir la règle de cascade numéro 3).
  - Par souci d'équilibre, les déclarations avec "!important" (les mots-clés "!" et "important" suivent la déclaration) établissent ainsi leur préséance sur les déclarations normales. Aussi bien les feuilles de style de l'auteur que celles de l'utilisateur peuvent contenir des déclarations avec "!important", celles de l'utilisateur ayant priorité. Cette fonction de CSS améliore l'accessibilité des documents, offrant à ceux des utilisateurs qui ont des besoins particuliers (une grande taille de police, d'autres combinaisons de couleur, etc.), une certaine maîtrise de la présentation.

# Cascade

- ❖ Les feuilles de style ont trois origines différentes : l'auteur, l'utilisateur et l'agent utilisateur.
  - **L'auteur** : produit des feuilles de style pour un document source selon les conventions du langage de ce document. Celles-ci peuvent être incorporées dans le document ou reliées à celui-ci.
  - **L'utilisateur** : peut être capable d'indiquer une information de style pour un document particulier. Par exemple, celui-ci peut désigner une feuille de style contenue dans un fichier ou un agent utilisateur peut fournir l'interface pour produire une feuille de style personnelle.
  - **L'agent utilisateur** : l'agent utilisateur conforme doit appliquer sa feuille de style par défaut avant toutes les autres feuilles de style d'un document. Cette feuille de style devrait présenter les éléments du document de façon à satisfaire au rendu généralement admis pour ceux-ci. Par exemple, pour des navigateurs visuels, l'élément EM en HTML est rendu avec une police en italique. Noter que cette feuille de style par défaut peut varier si l'utilisateur change les réglages de son système (ex. les couleurs du système). Cependant, il peut être impossible de modifier les valeurs de celle-ci, en raison d'une implémentation réduite de l'agent utilisateur.

# Cascade

- ❖ 1. Trouver toutes les déclarations qui concernent l'élément et la propriété en question, pour le type de média visé. Celles-ci s'appliquent si le sélecteur correspond à cet élément ;
- ❖ 2. Un tri primaire est effectué sur les déclarations selon leur poids et leur origine : pour une déclaration normale, les feuilles de style de l'auteur surclassent celles de l'utilisateur, ces dernières surclassant la feuille de style par défaut. Pour une déclaration avec "!important", celles-ci surclassent les déclarations normales. Une feuille de style importée a la même origine que celle qui l'a importée ;
- ❖ 3. Un tri secondaire est effectué selon la spécificité des sélecteurs : les plus spécifiques surclasseront ceux plus généraux. Les pseudo-éléments et les pseudo-classes sont considérés respectivement comme des éléments et des classes normaux ;
- ❖ 4. Un tri final selon l'ordre de spécification : si deux règles ont les mêmes poids, origines et spécificités, c'est la dernière survenue qui l'emporte. Les règles issues de feuilles de style importées sont considérées comme étant survenues avant chacune de celles de la feuille de style elle-même.

# Spécificité d'un sélecteur

- ◆ La spécificité d'un sélecteur est déterminée comme suit :
  - dans le sélecteur, compter le nombre d'attributs Id (= a) ;
  - puis celui des autres attributs et des pseudo-classes (= b) ;
  - et ensuite le nombre de noms des éléments (= c) ;
  - ignorer les pseudo-éléments.
- ◆ Assembler les trois nombres a-b-c (dans un système de nombre avec une base étendue) pour obtenir la spécificité.
  - \* {} /\* a=0 b=0 c=0 -> spécificité = 0 \*/
  - LI {} /\* a=0 b=0 c=1 -> spécificité = 1 \*/
  - UL LI {} /\* a=0 b=0 c=2 -> spécificité = 2 \*/
  - UL OL+LI {} /\* a=0 b=0 c=3 -> spécificité = 3 \*/
  - H1 + \*[REL=up] {} /\* a=0 b=1 c=1 -> spécificité = 11 \*/
  - UL OL LI.red {} /\* a=0 b=1 c=3 -> spécificité = 13 \*/
  - LI.red[title] {} /\* a=0 b=2 c=1 -> spécificité = 21 \*/
  - #x34y {} /\* a=1 b=0 c=0 -> spécificité = 100 \*/
- ◆ Pour HTML, les valeurs de l'attribut "style" sont des règles de feuille de style. Ces règles n'ont pas de sélecteurs, mais dans l'optique du point 3 de l'algorithme de cascade, on considère qu'elles ont un sélecteur d'ID (spécificité : a=1, b=0, c=0). Et dans l'optique du point 4, on considère qu'elles surviennent après toutes les autres règles.

# Style hors CSS

- ❖ L'agent utilisateur peut tenir compte des indications de présentation provenant d'autres sources que les feuilles de style, par exemple l'élément FONT ou l'attribut "align".
- ❖ Dans ce cas, ces indications doivent être traduites dans leurs règles équivalentes de CSS avec une spécificité égale à zéro. Ces règles sont sensées se trouver au début de la feuille de style de l'auteur, permettant leur surclassement par les règles suivantes de cette feuille de style.

# Media

- ❖ Une des fonctions primordiales des feuilles de style repose sur le fait de pouvoir adapter la représentation d'un document pour différents médias : un écran, une feuille de papier, un synthétiseur de parole, un appareil braille, etc.
- ❖ Certaines propriétés spécifiques au média
- ❖ Spécification du média :
  - `@import url("loudvoice.css") aural;`
  - `@media print { /* la feuille de style pour l'impression vient ici */ }`
  - `@media screen, print { BODY { line-height: 1.2 } }`
  - `<LINK rel="stylesheet" type="text/css" media="print, handheld" href="foo.css">`

# Catégorie de média

- ❖ **continu** ou **paginé** (continuous/paged) ;
- ❖ **visuel, auditif** ou **tactile** (visual/aural/tactile ) ;
- ❖ **grille** (pour les appareils avec grille de caractères) ou **bitmap** (grid/bitmap);
- ❖ **interactif** (pour les appareils qui interagissent avec l'utilisateur) ou **statique** (à l'inverse, pour ceux qui n'interagissent pas avec celui-ci) (interactive/static) ;
- ❖ **all** (comprend tous les types de médias).

Media Types	Media Groups			
	continuous/paged	visual/aural/tactile	grid/bitmap	interactive/static
aural	continuous	aural	N/A	both
braille	continuous	tactile	grid	both
emboss	paged	tactile	grid	both
handheld	both	visual	both	both
print	paged	visual	bitmap	static
projection	paged	visual	bitmap	static
screen	continuous	visual	bitmap	both
tty	continuous	visual	grid	both
tv	both	visual, aural	bitmap	both

# Color

## ❖ **color**

- *Valeur* : <couleur> | inherit
- *Initiale* : selon l'agent utilisateur
- *S'applique à* : tous les éléments
- *Héritée* : oui
- *Pourcentage* : sans objet
- *Médias* : visuel

❖ Cette propriété décrit la couleur d'avant-plan du contenu de texte d'un élément. Voici plusieurs façons d'indiquer la couleur rouge

- EM { color: red } /\* nom de couleur prédéfini \*/
- EM { color: rgb(255,0,0) } /\* couleurs en RGB allant de 0 à 255 \*/

# background

- ❖ **Les propriétés d'arrière-plan : 'background-color', 'background-image', 'background-repeat', 'background-attachment', 'background-position' et 'background'**
- ❖ **background-color**
  - *Valeur* : <couleur> | transparent | inherit
  - *Initiale*: transparent
  - *S'applique à* : tous les éléments
  - *Héritée* : non
  - *Pourcentage* : sans objet
  - *Médias* : visuel
- ❖ Cette propriété donne la couleur d'arrière-plan d'un élément, avec une valeur de <couleur>, ou bien avec le mot-clé 'transparent', celui-ci laissant éventuellement voir les couleurs situées plus en-dessous.

# background

## ❖ **background-image**

- *Valeur* : <uri> | none | inherit
- *Initiale* : none
- *S'applique à* : tous les éléments
- *Héritée* : non
- *Pourcentage* : sans objet
- *Médias* : visuel

- ❖ Cette propriété spécifie l'image d'arrière-plan d'un élément. En même temps qu'une image, les auteurs devraient aussi spécifier une couleur d'arrière-plan, cette couleur étant employée en remplacement d'une image indisponible. Celle-ci, une fois disponible, vient se superposer sur le fond coloré. La couleur du fond étant ainsi visible au travers des zones transparentes de l'image.
- ❖ La propriété admet les valeurs d'<uri>, pointant vers une image, ou 'none', prohibant l'emploi d'image.

# background

## ❖ **background-repeat**

- *Valeur* : repeat | repeat-x | repeat-y | no-repeat | inherit
- *Initiale* : repeat
- *S'applique à* : tous les éléments
- *Héritée* : non
- *Pourcentage* : sans objet
- *Médias* : visuel

❖ Quand on spécifie une image d'arrière-plan, cette propriété indique si l'image est répétée et la manière de la répétition. La mosaïque de fond résultante correspond aux aires de contenu, d'espacement et de bordure de la boîte de l'élément en question. Les significations des valeurs sont :

- **repeat** L'image se répète à la fois horizontalement et verticalement ;
- **repeat-x** L'image ne se répète qu'horizontalement ;
- **repeat-y** L'image ne se répète que verticalement ;
- **no-repeat** L'image ne se répète pas : un seul exemplaire de celle-ci est dessiné.

# background

## ❖ background-attachment

- *Valeur* : scroll | fixed | inherit
- *Initiale* : scroll
- *S'applique à* : tous les éléments
- *Héritée* : non
- *Pourcentage* : sans objet
- *Médias* : visuel

- ❖ Quand on spécifie une image d'arrière-plan, cette propriété indique si l'image est fixe par rapport à l'espace de visualisation (pour la valeur 'fixed'), ou si celle-ci défile en même temps que le document (pour la valeur 'scroll').
- ❖ Noter qu'il n'existe *qu'un seul* espace de visualisation par document ; c.à.d., même si l'élément est doté d'un mécanisme de défilement (voir la propriété 'overflow'), un arrière-plan avec la valeur 'fixed' ne se déplace pas avec cet élément.
- ❖ Si l'image est fixe, celle-ci n'est seulement visible que quand elle se trouve dans les aires d'arrière-plan, d'espacement et de bordure de l'élément. À moins que l'image ne se répète en mosaïque ('background-repeat: repeat'), celle-ci peut ainsi ne pas apparaître.

# background-position

- Valeur : [ [<pourcentage> | <longueur> ]{1,2} | [ [top | center | bottom] || [left | center | right] ] ] |inherit
- Initiale : 0% 0%
- S'applique à: ceux des éléments de type bloc et ceux remplacés
- Héritée : non
- Pourcentage : se rapporte à la taille de la boîte elle-même
- Médias : visuel

◆ Pour une image d'arrière-plan, cette propriété indique la position initiale de celle-ci.

- <pourcentage> <pourcentage> : pour la paire de valeurs x% y%, le point de l'image correspondant est aligné avec le point de la surface d'affichage (même proportion)
- <longueur> <longueur> : pour la paire de valeur '3cm 2cm', le coin en haut et à gauche de l'image se place sur le point, situé à 3cm vers la droite et 2cm vers le bas en partant du coin en haut et à gauche de l'aire d'affichage ;
- top left et left top : Identique à '0% 0%' ;
- top, top center et center top : Identique à '50% 0%';
- right top et top right : Identique à '100% 0%' ;
- left, left center et center left : Identique à '0% 50%' ;
- center et center center : Identique à '50% 50%' ;
- right, right center et center right : Identique à '100% 50%' ;
- bottom left et left bottom : Identique à '0% 100%' ;
- bottom, bottom center et center bottom : Identique à '50% 100%' ;
- bottom right et right bottom : Identique à '100% 100%'.

◆ Une seule valeur (% ou longueur) ne concerne que la position horizontale, la position verticale sera 50%. Deux valeurs (% ou longueur) la première concerne la position horizontale. Les positions négatives le sont également.

# Background

- ❖ La propriété 'background' est une propriété raccourcie qui sert à regrouper les propriétés individuelles 'background-color', 'background-image', 'background-repeat', 'background-attachment' et 'background-position' dans la feuille de style.
- ❖ La propriété 'background' distribue d'abord à toutes les propriétés individuelles d'arrière-plan à leur valeur initiale, puis leur applique les valeurs explicites de la déclaration.
- ❖ Ex :
  - BODY { background: red }
  - P { background: gray url("ele.jpg") 50% repeat fixed }
  - <exemple-cours5/background.html>

# Font

## ◆ font

- *Valeur* : [ [ <'font-style'> || <'font-variant'> || <'font-weight'> ]? <'font-size'> [ / <'line-height'> ]? <'font-family'> ] | **caption** | **icon** | **menu** | **message-box** | **small-caption** | **status-bar** | **inherit**
- *Initiale* : voir les propriétés individuelles
- *S'applique à* : tous les éléments
- *Héritée* : oui
- *Pourcentages*: permis sur 'font-size' et 'line-height'
- *Médias* : visuel

## ◆ Les valeurs suivantes concernent les polices de système :

- **caption** : La police employée pour légender les contrôles (ex. boutons, menus déroulants, etc.) ;
- **icon** : La police employée pour légender les icones ;
- **menu** : La police employée dans les menus (ex. les menus déroulants et les listes de menus) ;
- **message-box** : La police employée dans les boîtes de dialogue ;
- **small-caption** : La police employée pour étiqueter les contrôles de petite taille ;
- **status-bar** : La police employée dans les barres de statut de la fenêtre.

# font-style

- ❖ Le style de police précise l'aspect dans lequel doit être rendu le texte, normal, italique ou oblique.
- ❖ **font-style**
  - *Valeur* : normal | italic | oblique | inherit
  - *Initiale* : normal
  - *S'applique à* : tous les éléments
  - *Héritée* : oui
  - *Pourcentage* : sans objet
  - *Médias* : visuel

# font-variant

- ❖ La variante de police indique si le texte doit être rendu avec des glyphes normaux ou avec des glyphes en petites capitales, pour les caractères en minuscule.
- ❖ **font-variant**
  - *Valeur* : normal | small-caps | inherit
  - *Initiale* : normal
  - *S'applique à* : tous les éléments
  - *Héritée* : oui
  - *Pourcentage* : sans objet
  - *Médias* : visuel

# font-weight

- ◆ Le poids de police se réfère à la graisse, plus ou moins grasse, des glyphes utilisés pour rendre le texte, en fonction des autres membres de la même famille de polices.
- ◆ **font-weight**
  - *Valeur* : normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit
  - *Initiale* : normal
  - *S'applique à* : tous les éléments
  - *Héritée* : oui
  - *Pourcentage*
  - : sans objet
  - *Médias* : visuel
- ◆ Les significations des valeurs sont :
  - **100 à 900** : Ces valeurs forment une séquence ordonnée, où chacun des nombres indique, pour la police, une graisse au moins aussi grasse que celle du nombre précédent ;
  - **normal** Équivaut à '400' ;
  - **bold** Équivaut à '700' ;
  - **bolder** : Spécifie la graisse supérieure à celle assignée à une police, cette graisse étant plus grasse que celle héritée par la police. S'il n'y en a pas, la valeur de graisse de la propriété prend simplement la valeur numérique supérieure (l'aspect de la police ne changeant pas), à moins que la valeur héritée ne soit déjà égale à '900', auquel cas la valeur résultante devient également '900' ;
  - **lighter** : Spécifie la graisse inférieure à celle assignée à une police, cette graisse étant moins grasse que celle héritée par la police. S'il n'y en a pas, la valeur de graisse de la propriété prend simplement la valeur numérique inférieure (l'aspect de la police ne changeant pas), à moins que la valeur héritée ne soit déjà égale à '100', auquel cas la valeur résultante devient également '100'.

# font-size

## ◆ font-size

- *Valeur* : <taille-absolue> | <taille-relative> | <longueur> | <pourcentage> | inherit
- *Initiale* : medium
- *S'applique à* : tous les éléments
- *Héritée* : oui, la valeur calculée également
- *Percentages*: se rapporte à la taille de la police du parent de l'élément
- *Media*: visuel

◆ Cette propriété décrit la taille d'une police spécifiée explicitement. Celle-ci correspond au carré em, un concept issu de la typographie. Noter que certains glyphes peuvent déborder de leur carré em. Les significations des valeurs sont :

- **<taille-absolue>** Un mot-clé <taille-absolue> se réfère à une entrée de la table des tailles de police, celle-ci étant dressée et mise en oeuvre par l'agent utilisateur. Les valeurs possibles :
  - ◆ [ xx-small | x-small | small | medium | large | x-large | xx-large ]
  - ◆ facteur d'échelle de 1.2 entre les valeurs consécutives de la table ; si la valeur 'medium' correspond à 12pt, la valeur 'large' devrait correspondre à 14.4pt.
  - ◆ Les facteurs d'échelle peuvent différer selon les médias considérés.
- **<taille-relative>** Un mot-clé <taille-relative> s'entend par rapport à la table des tailles de police et par rapport à la taille de la police de l'élément parent. Les valeurs possibles :
  - ◆ [ larger | smaller ]
- **<longueur>** Une valeur de longueur spécifie une taille de police absolue (indépendante de la table des tailles de police de l'agent utilisateur). Les valeurs négatives ne sont pas admises ;
- **<pourcentage>** Une valeur en pourcentage spécifie une taille de police absolue par rapport à celle de l'élément parent. Leur emploi, de même pour les valeurs exprimées en 'em', conduit à des feuilles de style plus fiables, en plein accord avec le principe de la cascade.

# line-height

## ❖ **line-height**

- *Valeur* : normal | <nombre> | <longueur> | <pourcentage> | inherit
- *Initiale* : normal
- *S'applique à* : tous les éléments
- *Héritée* : oui
- *Percentages*: se rapporte à la taille de la police de l'élément lui-même
- *Médias* : visuel

❖ Hauteur minimale pour les in-line inclus ou hauteur exacte pour un in-line

## ❖ **Les valeurs signifient :**

- ***normal*** : Indique aux agents utilisateurs une valeur calculée "raisonnable", basée sur la taille de la police de l'élément. Cette valeur revêt la même signification que <number>. On recommande pour 'normal' une valeur calculée entre 1.0 et 1.2 ;
- **<longueur>** : Cette longueur est appliquée à la hauteur de la boîte. Les valeurs négatives ne sont pas admises ;
- **<nombre>** : La valeur calculée de la propriété est obtenue en multipliant ce nombre par la taille de la police héritée de l'élément. Les valeurs négatives ne sont pas admises.
- **<pourcentage>** : La valeur calculée de la propriété correspond à ce pourcentage multipliée par la valeur calculée pour la taille de la police. Les valeurs négatives ne sont pas admises.

# font-family

## ◆ font-family

- *Valeur* : [[ <famille-nom> | <famille-générique> ],]\* [<famille-nom> | <famille-générique>] | inherit
- *Initiale* : selon l'agent utilisateur
- *S'applique à* : tous les éléments
- *Héritée* : oui
- *Pourcentage* : sans objet
- *Médias* : visuel

◆ Cette propriété donne une liste, par ordre de priorité, de noms de familles de polices et/ou de noms de familles génériques.

◆ Ex :

- BODY { font-family: Baskerville, "Heisi Mincho W3", Symbol, serif }
- La police "Baskerville" va fournir les glyphes pour les caractères latins, la police "Heisi Mincho W3", ceux pour les caractères japonais, la police "Symbol", ceux pour les symboles mathématiques, et la famille de polices générique 'serif', ceux éventuels d'autres caractères.
- La famille de polices générique sera utilisée si une, ou plusieurs, des polices d'un jeu de polices n'étaient pas disponibles. Bien que certaines polices fournissent un glyphe de substitution pour indiquer un "caractère manquant", celui-ci prenant typiquement l'aspect d'un carré, on ne devrait pas considérer ce glyphe comme une correspondance valide.

# Exemple de police

- ❖ **serif**
  - des terminaisons au bout de leurs traits, des bouts évasés ou en pointes, ou plus simplement des terminaisons avec des empattements
  - Polices latines : Times New Roman, Polices grecques : Bitstream Cyberbit, Polices cyrilliques : Adobe Minion Cyrillic, Polices hébraïques : New Peninim, Polices japonaises : Ryumin Light-KL, Polices arabes : Bitstream Cyberbit
- ❖ **sans-serif**
  - des traits pleins, sans évasement, ni recoupements ou autres ornements
  - Polices latines MS Arial, MS Verdana, Helvetica, Polices grecques : Helvetica Greek, Polices cyrilliques Helvetica Cyrillic, Polices hébraïques : Arial Hebrew, Polices japonaises Shin Go
- ❖ **cursive**
  - des traits qui se recoupent ou bien un aspect manuscrit plus affirmé
  - Polices latines : Caflisch Script, Polices cyrilliques : ER Architekt, Polices hébraïques : Corsiva, Polices arabes : DecoType Naskh, Monotype Urdu 507
- ❖ **fantasy**
  - Critter
- ❖ **monospace**
  - Courier

# font-size-adjust

- *Valeur* : <nombre> | none | inherit
- *Initiale* : none
- *S'applique à* : tous les éléments
- *Héritée* : oui
- *Pourcentage* : sans objet
- *Médias* : visuel

◆ Cette propriété permet de spécifier une valeur d'aspect pour un élément, ce qui préserve la valeur hauteur-x de la première police à choisir dans la liste des polices de substitution. Les significations des valeurs sont :

- **none** : ne préserve pas la valeur hauteur-x de la police ;
- **<nombre>** : Spécifie la valeur d'aspect. Ce nombre concerne la valeur d'aspect de la première police à choisir. Le facteur d'échelle pour les polices disponibles se détermine selon cette formule :
  - ◆  $y(a/a') = c$
- ce qui correspond à :
  - ◆  $y$  = la valeur de 'font-size' de la première police du choix ;
  - ◆  $a$  = la valeur d'aspect de la première police du choix ;
  - ◆  $a'$  = la valeur d'aspect de la police disponible ;
  - ◆  $c$  = la valeur de 'font-size' appliquée à la police disponible.
- Exemple :
- Par exemple, la police Verdana (valeur d'aspect de 0.58) avec une taille de police 14px spécifiée n'étant pas disponible, une police de substitution avec une valeur d'aspect de 0.46 étant disponible, la taille de police de celle-ci serait :  $14 * (0.58/0.46) = 17.65\text{px}$ .

# font-stretch

## ❖ font-stretch

- *Valeur* : normal | wider | narrower | ultra-condensed | extra-condensed | condensed | semi-condensed | semi-expanded | expanded | extra-expanded | ultra-expanded | inherit
- *Initiale* : normal
- *S'applique à* : tous les éléments
- *Héritée* : oui
- *Pourcentage* : sans objet
- *Médias* : visuel

❖ La propriété 'font-stretch' sélectionne les dessins normal, compacté ou élargi dans une famille de polices.

❖ [exemple-cours5/font.html](#)

# Text

Property	Description	Values
color	Sets the color of a text	<i>color</i>
direction	Sets the text direction	ltr rtl
letter-spacing	Increase or decrease the space between characters	normal <i>length</i>
text-align	Aligns the text in an element	left right center justify
text-decoration	Adds decoration to text	none underline overline line-through blink
text-indent	Indents the first line of text in an element	<i>length</i> %
text-transform	Controls the letters in an element	none capitalize uppercase lowercase

**Nb : en suspend  
text-shadow**