

Web (Persistence)



Andrea G. B. Tettamanzi

Université de Nice Sophia Antipolis

Département Informatique

andrea.tettamanzi@unice.fr

CM - Séance 2

HTML et CSS

Plan

- HTML
 - Quelques balises importantes
 - Tables
 - Formulaires
 - Images cliquables
- CSS
 - Principe de fonctionnement
 - Syntaxe
 - Valeurs
 - Sélecteurs
 - Assignation, cascade, héritage

Balises HTML

- Balises structurelles
- Balises contenant du texte (phrase)
- Balise hypertexte
- Balises de titres
- Balises de paragraphes et citations
- Balises de listes
- Balises de table (tableaux)
- Balises de formulaire
- les images cliquables, objets, applet
- Balises de frame
- Balises meta (entêtes)
- Balises de mise en forme (déconseillées !)

Balises Structurelles

- html : définition du document
 - un head, un body
- head : partie comprenant les entêtes et meta-données du document
 - Un `<title> ... </title>` obligatoire
 - Les balises « meta », « script », « style », etc. optionnelles
- title : titre du document
 - Titre de la page (en haut du navigateur)
 - Contient du texte
- body : corps du document
 - Partie centrale du document

Balises de bloc

- Regroupement générique : div | fieldset
- Paragraphes : p | pre
- Titres : h1 | h2 | h3 | h4 | h5 | h6
- Listes : ol | ul | dl
- Tableau : table
- Formulaire : form
- Divers : noscript | blockquote | hr | address

Balises en-ligne

- texte (PCDATA)
- phrase : em | strong | dfn | code | samp | kbd | var | cite | abbr | acronym
- special : a | img | object | br | script | map | q | sub | sup | span | bdo
- formulaire : input | select | textarea | label | button
- fontstyle : tt | i | b | big | small (déconseillées)

Balise <a> (ancre)

- Référencer une page par son URI
 - `....`
- référencer à l'intérieur d'une page
 - référence : `href= "URI#nom de l'ancre"` (lien local si URI = "")
 - "nom de l'ancre" = la valeur d'un attribut `id` d'une balise (n'importe laquelle)
 - `http://www.i3s.unice.fr/~tettaman/Classes/IntroWeb/html/exo/index.html#exo2`
- Contient des éléments html « inline » excepté `<a>`

Tables

- `<table>...</table>`
- Table row : `<tr>...</tr>`
- Table data : `<td>...</td>`
- (caption?, (col*|colgroup*), thead?, tfoot?, tbody+)
 - Le tableau peut avoir une légende `<caption>`
 - On peut regrouper les colonnes (`<col>` et `<colgroup>`)
 - Il peut y avoir une entête (répété pour impression) : `<thead>`
 - Il peut y avoir un pied de tableau (répété pour impression) : `<tfoot>`
 - Il doit y avoir au moins un corps de tableau `<tbody>`

Colspan et Rowspan

- L'attribut `colspan= n` permet de définir une cellule qui mesure n colonnes de large
- L'attribut `rowspan= n` permet de définir une cellule qui mesure n ligne de haut

Formulaires

```
<FORM action="http://somesite.com/prog/adduser" method="post">
  <P>
    <LABEL for="firstname">First name: </LABEL>
      <INPUT type="text" id="firstname"><BR>
    <LABEL for="lastname">Last name: </LABEL>
      <INPUT type="text" id="lastname"><BR>
    <LABEL for="email">email: </LABEL>
      <INPUT type="text" id="email"><BR>
    <INPUT type="radio" name="sex" value="Male"> Male<BR>
    <INPUT type="radio" name="sex" value="Female"> Female<BR>
    <INPUT type="submit" value="Send"> <INPUT type="reset">
  </P>
</FORM>
```

Images cliquables

- des images contenant des zones servant de liens
- Pour la mise en place d'une image cliquable, on doit composer 2 balises :
 - une balise Image `` pour charger l'image
 - une balise `<map>` pour définir les zones et les liens concernés
 - la référence à la balise `<map>` est faite par l'attribut « usemap »
- Deux types d'images cliquables
- Côté client (le plus fréquent et préférable) : traitement par le navigateur
- Côté serveur : traitement sur le serveur

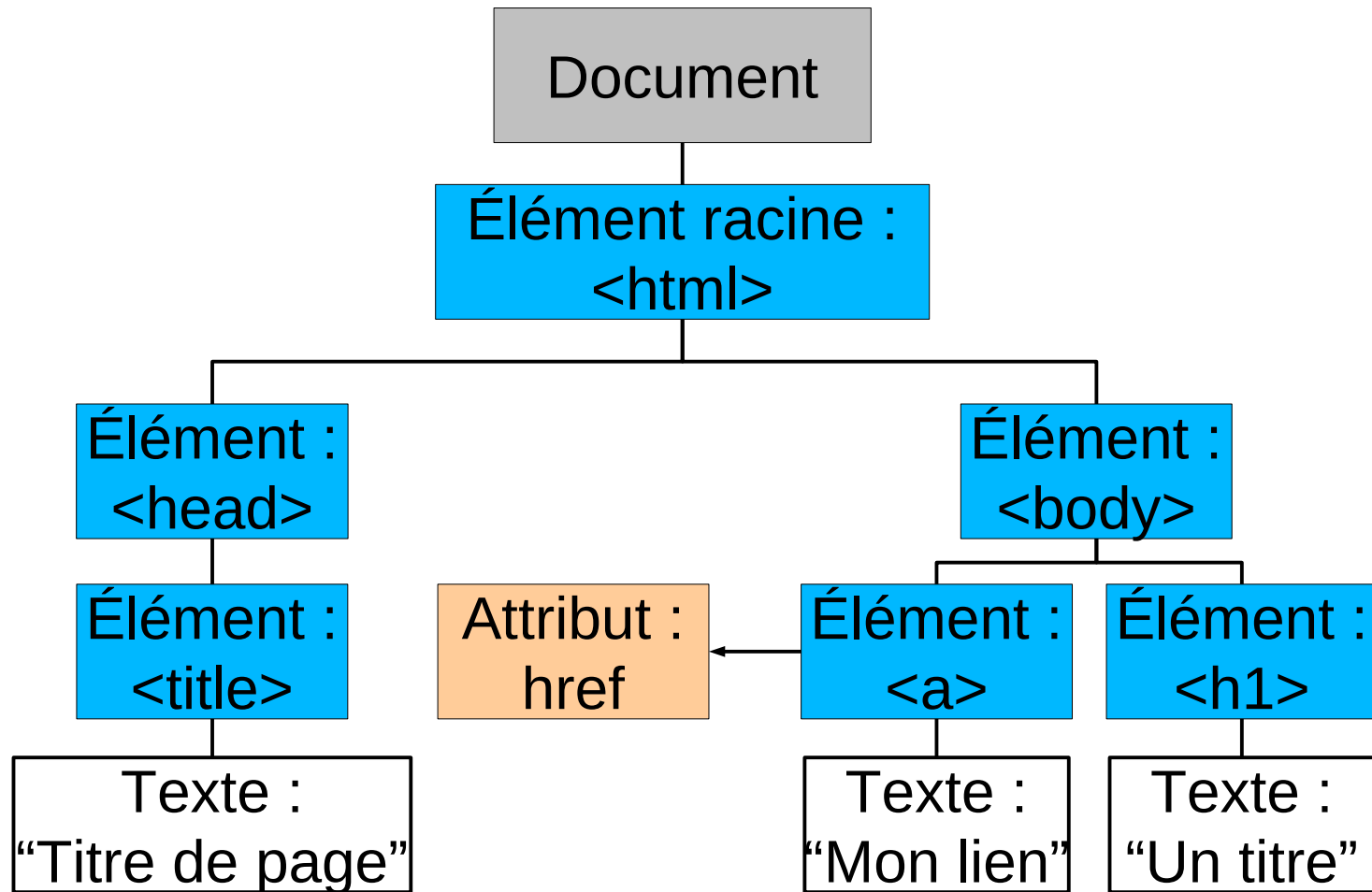
CSS

- D'un point de vue "conception" (ergonomique et programmation), HTML non satisfaisant.
- 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 facilement l'aspect d'un page ou d'un site sans en modifier le contenu.
 - Cohésion de la présentation tout au long du site (feuilles de style externes).
 - Réduction du temps de chargement des pages.
- Correction de certains écueils d'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, maintenance plus facile

Principe de fonctionnement

- Le rendu d'une page Web est le résultat de la combinaison de :
 - Un document HTML (structure, contenu)
 - Une ou plusieurs feuilles de style (mise en forme)
- Le document est analysé et transformé en un arbre
- Une feuille de style consiste en un nombre de règles
 - Le sélecteur détermine à quels éléments la règle s'applique
 - Le corps de la règle affecte des valeurs à des propriétés
- L'arbre-document est parcouru
 - Pour chaque node (élément), les règles dont le sélecteur correspond sont appliquées
 - Le résultat est un arbre « décoré »

Arbre-document (DOM)



Syntaxe

- Deux types d'instructions :
 - Directives (*at-rules*), « @ » + <identificateur>
 - @import, @charset, @media, ...
 - Terminées par « ; » ou par un bloc de déclarations
 - Règles (*rule sets*), <sélecteur> + <bloc de déclarations>
 - Bloc de déclaration : « { » + <déclarations> + « } »
 - Déclaration : <propriété> : <valeur> ;
 - Sélecteur : une expression à base de balises et pseudo-balises
- Commentaires : /* ... */
- Tout ce qui est non valide n'est pas pris en compte

Examples

```
h2 { color: green }
```

```
@import subs.css
```

```
h3:before {  
  content: counter(sectno, upper-roman) " - ";  
  float: left;  
  margin-left: -2cm;  
}
```

Insertion dans un doc HTML

- 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>`
- 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
 - `<li style="color:red;"> ...`

Valeurs (1/3)

- Les entiers et les nombres
- unités

ABBR.	UNIT	EXAMPLE	NOTES
– cm	Centimeters	2.5cm	
– em	Ems	3em	1 em equals the font's M
– 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 = height of « x »

Valeurs (2/3)

- Les pourcentages (par rapport à... en fonction de la propriété)
- URI (url) : `url("http://deptinfo.unice.fr/~renevier/L2/ipw.png")`
- Les compteurs
- Les chaînes de caractères :
 - entre " "
 - Caractères spéciaux : `\codeascii`
 - `\22` ou `\"` ; `\27` ou `\'`
 - "bla bla bla a 'bla' bla bla" : valide
- Styles non graphiques :
 - 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)

Valeurs (3/3)

KEYWORD	RGB HEX	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

Sélecteurs (1/4)

- 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

Sélecteurs (2/4)

- Pattern **élément élément** (sélecteur descendant)
 - tous les éléments (en 2ème) 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

Sélecteurs (3/4)

- 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 2ème) contenus **directement** 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 2ème) 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

CSS2 : Sélecteurs (4/4)

- 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"

CSS2 : 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.
 - Déconseillé (modification des habitudes des utilisateurs)
- pseudo-classes dynamiques **`:active`** , **`:hover`** et **`:focus`**
 - **`:hover` pour l'élément survolé 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)

CSS2 : pseudo-éléments

- **:first-line**
 - première ligne du texte
 - dépend de la taille de la fenêtre
- **:first-letter** : première lettre du texte
- **:before** et **:after**
 - pour à insérer un contenu généré avant ou après
 - H1:before {content: counter(chapno, upper-roman) ". "}
 - Ajoute une numérotation (automatique) en chiffre romain : « I. », « II. », etc.
- En combinaison
 - Exemple :
 - P.special:before {content: "Spécial ! "}
 - P.special:first-letter {color: gold}
 - Ceci donnera une couleur or au "S" de "Spécial !".
 - Bugs possibles (selon navigateur)

Merci de votre attention

