

Débuter en CSS (2)

Création Site Web

Licence 2 MASS

Sébastien Verel
verel@i3s.unice.fr
www.i3s.unice.fr/~verel

Équipe ScoBi - Université de Nice Sophia-Antipolis

17 octobre 2008

Objectifs de la séance 5

- 1 Connaitre les sélecteurs avec attributs et de pseudo-classe
- 2 connaître les règles de cascade et d'héritage
- 3 connaître les propriétés de mise en forme du texte
- 4 Connaitre le modèle de boîte CSS
- 5 Savoir modifier le style d'une boîte

Plan

- 1 Rappels
- 2 Sélecteurs
- 3 Style des listes
- 4 Modèle de Boîte

Règles générales de déclaration de style

- sélecteur : détermine l'élément (et condition) sur lequel s'applique le style
- déclaration de propriétés : entre accolades

```
h1 { color : red ; background-color : blue ; }
```

Sélecteur

propriété

propriété

synthaxe concrète en rouge

éléments grammaticaux en gras italique

Associer un style

- Définir les styles de la page web correspondante :
dans l'élément `<style>`
- fichier séparé :
Utiliser l'élément `<link>`
- un peu les deux :
Utiliser l'élément `<style>` avec `@import`
- attribut `style` :
directement dans la base de l'élément

Les "vieux"

Quels sont les sélecteurs déjà rencontrée la semaine passée ?

Avez-vous fini l'exercice 2 ?

Avez-vous utilisé ces sélecteurs dans le fil rouge ?

Sélecteur d'attributs

Sélectionner un élément possédant un attribut :

```
h2[title] { color: black ; background-color: green; }
```

Sélectionner un élément possédant plusieurs attributs :

```
h2[title][id] { color: black ; background-color: green; }
```

Sélecteur d'attributs

- Sélectionner un élément possédant un attribut de valeur particulière :

```
h2[title="vetement"] { color: black ; background-color: green ;
```

sélectionne l'élément :

```
<h2 title="vetement">r&egrave;gle vestimentaire 2</h2>
```

- Sélectionner un élément possédant un attribut contenant au moins une valeur particulière :

```
h2[title ~= "vetement"] { color: black ; background-color: green ;
```

sélectionne l'élément :

```
<h2 title="vetement 2">r&egrave;gle vestimentaire 2</h2>
```

- Sélectionner un élément possédant un attribut commençant par une valeur donnée :

```
h2[title |= "vet"] { color: black ; background-color: green ;
```

sélectionne l'élément :

```
<h2 title="vetement 2">r&egrave;gle vestimentaire 2</h2>
```


Pseudo-classes et pseudo-éléments

Sélecteurs de partie de documents non identifiable par la hierarchie

par ex. : première ligne, premier caractère, déplacement de curseur, etc.

Pseudo-classes applicables aux liens

- `:link` , sélecteur de lien non encore visité
- `:visited` , sélecteur de lien déjà vu, après retour sur la page d'origine

```
a:link { color: green ; }  
a:visited { color: yellow ; }
```

Pseudo-classes dynamiques

classes existants seulement au moment de l'action du visiteur

- `:focus` , pseudo-classe lors du focus (lorsque le navigateur le veut bien)
- `:hover` , pseudo-classe lors du survol d'élément visible
- `:active` , pseudo-classe lors du click de l'utilisateur

Pseudo-éléments

s'applique à une partie du contenu

- `:first-letter` , pseudo-élément de la première lettre du contenu
- `:first-line` , pseudo-élément de la première ligne du contenu
- `:before` , pseudo-élément qui permet d'insérer un contenu de style donné avant le début du contenu
- `:after` , pseudo-élément qui permet d'insérer un contenu de style donné après le contenu

```
p:first-letter { font-size: 300% ; color: red ; }  
p:first-line { font-size: 140% ; }  
a:before { content: " <-- " ; color: blue ; }  
a:after { content: " --> " ; color: blue ; }
```

Déclaration !important

- pour marquer la priorité :

```
* { color: black !important ; background-color: yellow ; }  
div { color: blue ; background-color: white ; }
```

Cascade et héritage

Permet d'établir les règles d'application de style sur les éléments.
Plusieurs acteurs dans la réalisation d'un style :

- auteur (concepteur)
- utilisateur
- navigateur

Logicielle CSS se charge de résoudre les conflits :

- différentes opérations permettant d'attribuer un style à un élément : [règles de cascades](#)
- inventaire des styles applicables sur un élément,
- sélection en fonction de différents critères

Sélection selon le media

Seul le style du media spécifié est utilisé

media indiqué dans les balises `<link>` ou `<style>` :
screen, print, projection, aural, braille, handheld, tty, tv, all

Sélection selon le créateur du style

la déclaration !important.

Priorité la plus importante à la moins importante :

- styles utilisateur la déclaratn !important
- styles concepteur déclarés !important
- styles concepteur non déclarés !important
- styles visiteur non déclarés !important
- styles du navigateur

Sélection par spécificité

élaboration de la spécificité : nombre de 4 chiffres $abcd$

- $a = 1$ lorsque le style est défini localement dans l'attribut, $a = 0$ sinon. (si $a = 1$ les autres chiffres ne sont pas calculés)
- b : nombre de sélecteur d'attribut id présent dans le sélecteur
- c : nombre de classes, de pseudo-classes et de sélecteurs d'attributs présent dans le sélecteur
- d : nombre d'éléments XHTML présent dans le sélecteur

le style de plus grande spécificité s'applique

sélection par ordre d'apparition

dans le cas où aucun critère ne s'applique.

- le dernier apparu dans le code s'applique
- si plusieurs fichiers sont importés, le dernier s'applique en cas de conflit.

Héritage

Regle d'héritage

élément enfant possède les mêmes styles que l'élément parent qui le contient.

Il est possible d'explicitier l'héritage par la déclaration de *inherit* bien qu'il est appliquer à un grand nombre de propriété.

Numérotation des listes

Numérotation

`list-style-type` : *type* | none | inherit

- decimal : 1, 2, 3,...
- decimal-leading-zero : 01, 02, 03,...
- upper-latin ou upper-alpha : A, B, C,...
- lower-latin ou lower-alpha : a, b, c, ...
- upper-roman : I, II, III, IV,...
- lower-roman : i, ii, iii, ...
- lower-greek : α , β , ...

Position de numérotation

Position

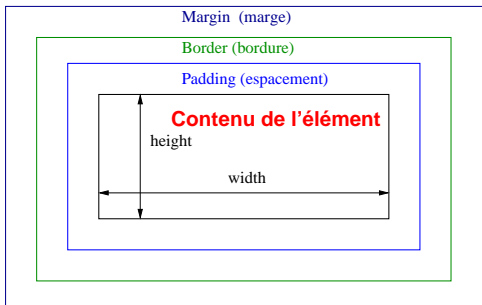
`list-style-position` : `inside` | `outside` | `inherit`

- `inside` : dans la marge de retrait de l'item
- `outside` (default) : intégrer au texte de l'item

Modèle de boîte

- chaque élément de type bloc ou ligne possède sa boîte
- conditionne la surface de l'élément
- boîte : zone rectangulaire
- Information du w3c :
<http://www.w3.org/TR/CSS21/box.html>

La boîte



Bordure

Style de bordure

`border-style` : inherit | none | hidden | *style* (1, 2, 3 ou 4)

le style est copié sur la bordure opposée

style : dotted, dashed, solid, inset, double, groove, ridge, outset.

Détaillé en : `border-top-style`, `border-right-style`,
`border-bottom-style`, `border-left-style`

cf. `ex05/bordure.html`

Bordure

Largeur

`border-width` : `inherit` | `thin` | `medium` | `thick` | *-longueur-* (1, 2, 3 ou 4)

copié sur la bordure opposée

cf. `ex05/mystere1.html`

Couleur

`border-color` : `inherit` | `transparent` | *-couleur-* (1, 2, 3 ou 4)

copié sur la bordure opposée

cf. `ex05/mystere2.html`

Définition globale

`border` : largeur style couleur | `inherit`

Bordure : globalement

Définition globale

`border : largeur style couleur | inherit`

Marges

margin : % | *-longueur-* | auto | inherit (de 1 à 4)

- % : pourcentage de l'élément parent
- auto : de manière automatique en fonction des autres informations

Naturellement peut-être détaillé en : margin-top, margin-right, margin-bottom, margin-left

Espacement

padding : % | *-longueur-* | inherit (de 1 à 4)

- % : pourcentage de l'élément parent

Naturellement peut-être détaillé en : padding-top, padding-right, padding-bottom, padding-left