

Plan

Unix et Programmation Web

Cours 5

kn@lri.fr
<http://www.lri.fr/~kn>

- 1 Réseaux, TCP/IP ✓
- 2 Web et HTML ✓
- 3 CSS
 - 3.1 Introduction
 - 3.2 Boîtes
 - 3.3 Autres propriétés
 - 3.4 Selecteurs
 - 3.5 requête media-type

Cascading Style Sheets (CSS)

L'attribut style

CSS : Langage permettant de décrire le **style graphique** d'une page HTML

On peut appliquer un style CSS

- À un élément en utilisant **l'attribut style**
- À une page en utilisant l'élément `<style>...</style>` dans l'en-tête du document (dans la balise `<head>...</head>`).
- À un ensemble de pages en référençant un fichier de style dans chacune des pages

```
<a href="http://www.u-psud.fr" style="color:red">Un lien</a>
```

Apperçu:

[Un lien](http://www.u-psud.fr)

Inconvénients :

- il faut copier l'attribut style pour tous les liens de la page
- modification de tous les éléments difficiles

L'élément style

```
<html>
  <head>
    <title>...</title>
    <style>
      a { color: red; }
    </style>
  </head>
  <body>
    <a href="#">Lien 1</a> <a href="#">Lien 2</a>
  </body>
</html>
```

Apperçu :

[Lien 1](#) [Lien 2](#)

Inconvénient : local à une page

Fichier .css séparé

Fichier style.css:

```
a { color: red; }
```

Fichier test.html:

```
<html>
  <head>
    ...
    <link href="style.css" type="text/css" rel="stylesheet" />
  </head>
  ...
</html>
```

Modifications & déploiement aisé

Syntaxe

Une **propriété** CSS est définie en utilisant la syntaxe:

```
nom_prop : val_prop ;
```

■ Si on utilise l'attribut `style` d'un élément:

```
<a href="#" style="color:red;border-style:solid;border:1pt;">Lien 1</a>
```

■ Si on utilise un fichier `.css` ou une feuille de style:

```
a {
  color : red;
  border-style: solid;
  border: 1pt;
}
h1 { /* Le style des titres de niveau 1 */
  text-decoration: underline;
  color: green;
}
```

Plan

1 Réseaux, TCP/IP ✓

2 Web et HTML ✓

3 CSS

3.1 Introduction ✓

3.2 Boîtes

3.3 Autres propriétés

3.4 Selecteurs

3.5 requête media-type

Unités de longueur

CSS permet de spécifier des longueurs comme valeurs de certaines propriétés (position et taille des éléments, épaisseur des bordures, ...). Les longueurs **doivent** comporter une unité. Les unités reconnues sont:

- px** : pixel
- in** : pouce (2,54cm)
- cm** : centimètre
- mm** : millimètre
- pt** : point (1/72ème de pouce, 0,35mm)
- pc** : pica (12 points)
- em** : facteur de la largeur d'un caractère de la police courante
- ex** : facteur de la hauteur d'un caractère « x » de la police courante
- %** : pourcentage d'une valeur particulière (définie par propriété)
- vh** : *viewport height* (% de la hauteur de la partie visible de la page) CSS3
- vw** : *viewport width* (% de la largeur de la partie visible de la page) CSS3

Boîte

Chaque élément de la page HTML possède une **boîte rectangulaire** qui délimite le contenu de l'élément:

Lien 1

La **taille t du contenu** est calculée pour que:

$$(height|width) = padding + margin + border + t$$

Marge, bordure, ajustement

On peut spécifier jusqu'à 4 valeurs:

- 1 valeur: toutes les dimensions égales à cette valeur
- 2 valeurs: haut et bas égal à la première valeur, gauche et droite égal à la deuxième
- 3 valeurs: haut à la première valeur, gauche et droite égale à la deuxième, bas égal à la troisième
- 4 valeurs: haut, droit, bas, gauche

```
span {  
  padding:10pt 20pt 5pt 0pt;  
  margin:10pt 5pt;  
  border-width:3pt;  
  border-color:red blue green;  
  border-style:solid dotted;  
}
```

Du **texte** dans une boîte

Modes d'affichage

La propriété **display** contrôle le mode d'affichage d'un élément:

- none** : l'élément n'est pas dessiné et n'occupe pas d'espace
- inline** : l'élément est placé sur la ligne courante, dans le flot de texte. La taille du contenu (avec les marges, ajustements et bordures) dicte la taille de la boîte, **height** et **width** sont ignorés (`<i>`, ``, ``, ``, ... sont **inline** par défaut).
- block** : l'élément est placé seul sur sa ligne. La taille est calculée automatiquement mais peut être modifiée par **width** et **height** (`<div>`, `<h1>`, `<p>`, ... sont **block** par défaut)
- inline-block** : positionné comme **inline** mais la taille peut être modifiée comme pour **block**

Modes d'affichage (exemples)

```
a { display: inline; ... }
a { display: block; ... }
```

Le [lien 1](#), le [lien 2](#) et le [lien 3](#).

Le [lien 1](#)
, le [lien 2](#)
et le [lien 3](#)

```
a { display: inline-block;
width: 4em;
height: 2em;
... }
```

Le [lien 1](#) , le [lien 2](#) et le [lien 3](#) .

Positionnement

Le type de positionnement est donné par la propriété **position**

- static** : positionnement « automatique »
- fixed** : positionnement par rapport à la fenêtre du navigateur (la boîte est supprimée du flot)
- relative** : positionnement « relatif » par rapport à la position normale
- absolute** : positionnement « absolu » par rapport à l'ancêtre le plus proche qui n'est pas *static*

Pour **fixed**, **relative** et **absolute**, les propriétés **top**, **bottom**, **left** et **right** dénotent les décalages respectifs.

Positionnement (exemple)

fixed (right:10pt,top:10pt)

```
a { position: static;
... }
a { position: fixed;
right:10pt;
top: 10pt;
}
a { position: relative;
left: 10pt;
bottom: -5pt;
... }
a { position: absolute;
right:0pt;
bottom: 10pt;
}
```

```
<ul style="position:relative;"><li>...</li> ...</ul>
```

- Positionnement **static**
- Positionnement
- Positionnement **relative (left:10pt,bottom:-5pt)**
- Positionnement **absolute (right:10pt,bottom:10pt)**

Plan

- 1 Réseaux, TCP/IP ✓
- 2 Web et HTML ✓
- 3 CSS
 - 3.1 Introduction ✓
 - 3.2 Boîtes ✓
 - 3.3 Autres propriétés
 - 3.4 Selecteurs
 - 3.5 requête media-type

Couleurs

Les couleurs peuvent être données:

- par nom symbolique: **red, blue, purple, ...**
- en hexadécimal: #**xyyzz**, avec $00 \leq xx, yy, zz \leq ff$
- en décimal: **rgb(x, y, z)**, avec $0 \leq x, y, z \leq 255$
- en décimal avec transparence: **rgba(x, y, z, a)**, avec $0 \leq x, y, z \leq 255$ et $0 \leq a \leq 1$ CSS3

Propriétés du texte

Certaines propriétés permettent d'altérer le rendu du texte d'un élément

direction :	ltr ou rtl (orientation du texte)
text-transform :	capitalize, uppercase, lowercase
text-decoration :	underline, overline, line-through
text-align :	left, right, center, justify
text-indent :	longueur du retrait de paragraphe

Propriétés de la police

- font-family :** liste de nom de polices séparées par des virgules (Helvetica, sans, "Times New Roman")
- font-style :** normal, italic
- font-weight :** normal, lighter, bold, bolder
- font-size :** soit une longueur soit xx-small, x-small, small, medium, large, x-large, xx-large

On peut aussi spécifier un descripteur de police CSS3

```
@font-face {  
  font-family: Toto;  
  src: url(toto.ttf);  
}  
a { font-family: Toto; }
```

Plan

- 1 Réseaux, TCP/IP ✓
- 2 Web et HTML ✓
- 3 CSS
 - 3.1 Introduction ✓
 - 3.2 Boîtes ✓
 - 3.3 Autres propriétés ✓
 - 3.4 **Selecteurs**
 - 3.5 requête media-type

Selecteurs

exemple : menu dépliant

On peut sélectionner finement les éléments auxquels un style s'applique

- x :** tous les éléments dont la balise est x
 - .foo :** tous les éléments dont l'attribut class vaut foo
 - #foo :** l'élément dont l'attribut id vaut foo (les id doivent être uniques)
 - X Y :** tous les éléments sélectionnés par Y qui sont des descendants d'éléments sélectionnés par X
 - X > Y :** tous les éléments dont sélectionné par Y qui sont des fils d'éléments sélectionnés par X
 - a:visited :** les liens déjà visités
 - a:link :** les liens non visités
 - X:hover :** élément sélectionné par X et survolé par la souris
- ```
div.foo ul li a:visited { color: red; }
```

```
<ul class="menu">
 Entrée 1
 <ul class="sous-menu"> Sous-entrée 1.1
 Sous-entrée 1.2
 Sous-entrée 1.3

 Entrée 2
 <ul class="sous-menu"> Sous-entrée 2.1
 Sous-entrée 2.2
 Sous-entrée 2.3


```

## Analyse

## exemple : menu dépliant (démonstration)

Pour que le menu soit « dépliant » lors du survol de la souris, on souhaite que :

- Par défaut, les éléments de sous-menu soient cachés (`display : none`)
- Les éléments se trouvant sous un élément survolé (`hover`) soient visibles (`display : block`)

Entrée 1

Entrée 2

## Style CSS du menu

```
li { padding: 10pt 0pt 10pt 0pt;
 display:block;
 background:orange;
 color:blue;
 }
.sous-menu { display : none; }

ul.menu > li:hover ul.sous-menu { display: block; }
ul.menu > li:hover ul.sous-menu > li {
 background:blue;
 color:orange;
 }
}
```

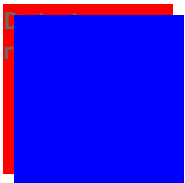
## Priorité d'application des règles (cascade)

Les règles sont appliquées dans l'ordre suivant :

- On applique toutes les règles des fichiers CSS référencés, dans l'ordre. Si plusieurs règles sont valides, elles sont appliquées dans l'ordre (en écrasant les comportements précédents, s'ils portent sur les mêmes attributs)
- Puis on applique les règles se trouvant dans l'élément style du fichier HTML courant
- Puis on applique enfin les règles se trouvant dans l'attribut style

## z-index

Il arrive que certaines boîtes se recouvrent :



On peut utiliser la propriété `z-index` pour définir l'ordre d'empilement (plus elle est élevée, plus la boîte est en avant plan)



## Plan

- 1 Réseaux, TCP/IP ✓
- 2 Web et HTML ✓
- 3 CSS
  - 3.1 Introduction ✓
  - 3.2 Boîtes ✓
  - 3.3 Autres propriétés ✓
  - 3.4 Selecteurs ✓
  - 3.5 requête media-type

## Différents styles pour différents affichages

On peut charger un style CSS de manière conditionnelle grâce à l'attribut `media` de la balise `link`. La valeur de l'attribut est une **formule logique** où l'on peut tester le type de support d'affichage ainsi que ces caractéristiques physiques :

```
<link rel="stylesheet" type="text/css" href="style1.css" media="all" />
<link rel="stylesheet" type="text/css" href="style2.css" media="print" />
<link rel="stylesheet" type="text/css" href="style3.css" media="screen and landscape" />
<link rel="stylesheet" type="text/css" href="style4.css" media="screen and min-width:480px" />
<link rel="stylesheet" type="text/css" href="style5.css" media="screen and max-width:479px" />
```

Cela permet d'appliquer des styles spécifiques lors de l'impression d'une page ou pour des terminaux mobiles (ayant une petite taille d'écran) ou de changer de style si l'orientation de l'écran est modifiée.

## Gestion du débordement

L'attribut `overflow` permet de gérer le débordement. Il peut prendre les valeurs `visible`, `hidden` et `auto` :

Lorem ipsum dolor sit  
amet, consectetur  
adipiscing elit, sed do  
eiusmod tempor

Lorem ipsum dolor sit  
amet, consectetur  
adipiscing elit, sed do  
eiusmod tempor

Lorem ipsum dolor sit  
amet, consectetur  
adipiscing elit, sed do  
eiusmod tempor

incidunt ut labore et  
dolore magna aliqua. Ut  
enim ad minim veniam,  
quis nostrud exercitation  
consequat. Duis aute irure  
dolor in reprehenderit in  
voluptate velit esse  
cillum dolore eu fugiat  
nulla pariatur. Excepteur  
sint occaecat cupidatat  
non proident, sunt in  
culpa qui officia deserunt  
mollit anim id est laboris  
nisi ut aliquip ex ea  
commodo consequat. Duis  
aute irure