

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Selecteurs

Syntaxe
Cascade
Priorité

Propriétés

Typographie
texte
Unités
Boîtes
Gestion du flux
Positionnement

Frameworks

Langages Web 1



Feuilles de styles

Florent Nicart

Université de Rouen

2016–2017

Introduction

Principe

Liaison

Inclusion

CSS Multiples

Selecteurs

Syntaxe

Cascade

Priorité

Propriétés

Typographie

texte

Unités

Boîtes

Gestion du flux

Positionnement

Frameworks

- Dès 1970, SGML propose la séparation de la structuration des données de leur mise en forme.
- Les langages à balises (HTML/XHTML) sont des sérialisations de l'arbre d'un document.
- L'objectif est de procéder à la mise en forme des données au moment de l'affichage, à la volée (*on the fly*), à l'aide de **feuilles de styles** ...

F. Nicart

Mise en forme à la volée

Introduction

Principe

Liaison

Inclusion

CSS Multiples

Selecteurs

Syntaxe

Cascade

Priorité

Propriétés

Typographie

texte

Unités

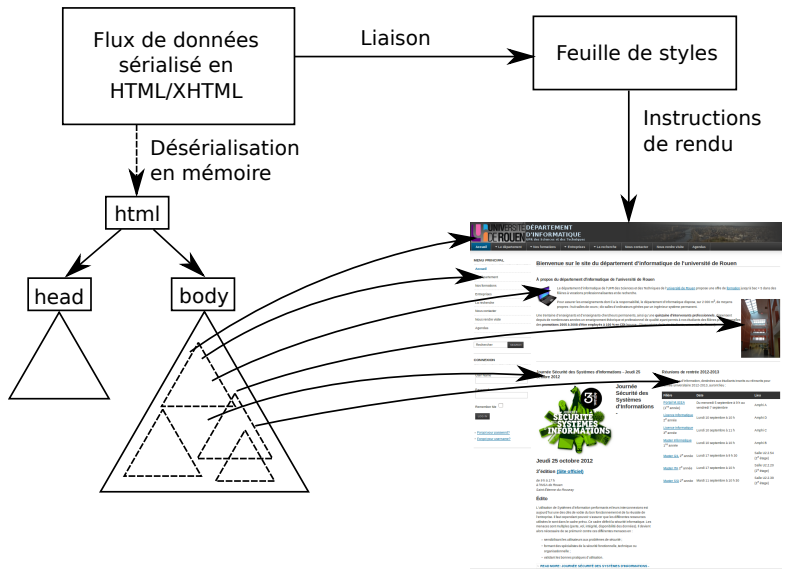
Boîtes

Gestion du flux

Positionnement

Frameworks

—



Un langage pour les feuilles de styles

Cascading Style Sheets

Introduction

Principe

Liaison

Inclusion

CSS Multiples

Selecteurs

Syntaxe

Cascade

Priorité

Propriétés

Typographie

texte

Unités

Boîtes

Gestion du flux

Positionnement

Frameworks

- Problématique des feuilles de styles : spécifier les propriétés de rendu des éléments de l'arbre.
- La description d'un style doit pouvoir être appliquée à un ensemble d'éléments de l'arbre au moyen de règles.
- Un nouveau langage est nécessaire qui met en uvre ces règles et des propriétés d'affichage :
- **CSS** (pour **Cascading Style Sheets**) est le langage de description des feuilles de style standardisé par le W3C.

Syntaxe de CSS

Introduction

Principe

Liaison

Inclusion

CSS Multiples

Selecteurs

Syntaxe

Cascade

Priorité

Propriétés

Typographie

texte

Unités

Boîtes

Gestion du flux

Positionnement

Frameworks

selecteur { [**propriété** : **valeur**;]⁺ }

- **selecteur** est une expression de complexité arbitraire dénotant un ensemble d'éléments de l'arbre d'après leur type, id, classe, ...
- les couples (**propriété,valeur**) dénotent la liste des modifications à apporter au style des éléments.
- les éléments ont un style par défaut, ou bien hérité de leur parent (cascade).

Liaison HTML–CSS

Liaison externe

Introduction

Principe

Liaison

Inclusion

CSS Multiples

Selecteurs

Syntaxe

Cascade

Priorité

Propriétés

Typographie

Texte

Unités

Boîtes

Gestion du flux

Positionnement

Frameworks

–

- La méthode de liaison la plus appropriée est l'inclusion d'un fichier de définitions externe :

```
1  ...  
2  <head>  
3    <link rel="stylesheet" type="text/css" href="url_fichier.css" />  
4  </head>  
5  ...
```

- la liaison se fait à partir de l'en-tête,
- `url_fichier.css` dénote l'url du fichier contenant le code CSS,
- cette url peut être donnée relativement par rapport au document courant.
- Cette méthode doit être privilégiée car les données de présentation sont séparées et réutilisables.

Introduction

Principe

Liaison

Inclusion

CSS Multiples

Selecteurs

Syntaxe

Cascade

Priorité

Propriétés

Typographie

texte

Unités

Boîtes

Gestion du flux

Positionnement

Frameworks

- Des styles peuvent être définis localement :

```
1  ...
2  <head>
3    <style type="text/css">
4      /* mystyles.css – Styles for the main site */
5      h1, h2, h3, h4 { color : blue ; }
6      h1 { font-size : 18pt ; }
7      h2 { font-size : 16pt ; }
8      h3 { font-size : 14pt ; }
9      h4 { font-size : 12pt ; }
10     p { font-size : 10pt ; }
11   </style>
12 </head>
13  ...
```

- la définition se fait à partir de l'en-tête.
- Les styles définis ont une portée locale au document et ne sont pas réutilisables.

Liaison HTML–CSS

Sur un unique élément

Introduction

Principe

Liaison

Inclusion

CSS Multiples

Sélecteurs

Syntaxe

Cascade

Priorité

Propriétés

Typographie

Texte

Unités

Boîtes

Gestion du flux

Positionnement

Frameworks

–

- La mise en forme *en dur* des éléments utilise l'attribut **style** et la syntaxe CSS :

```
1  ...
2  <body>
3    <h1 style="color :red;">Titre en rouge</h1>
4    blabla ...
5    <h1>Titre en couleur normale</h1>
6    blabla ...
7  </body>
8  ...
```

- La définition porte uniquement sur l'instance courante de l'élément,
- d'où l'absence de sélecteur.

Inclusion de source CSS

Introduction

Principe

Liaison

Inclusion

CSS Multiples

Selecteurs

Syntaxe

Cascade

Priorité

Propriétés

Typographie

texte

Unités

Boîtes

Gestion du flux

Positionnement

Frameworks

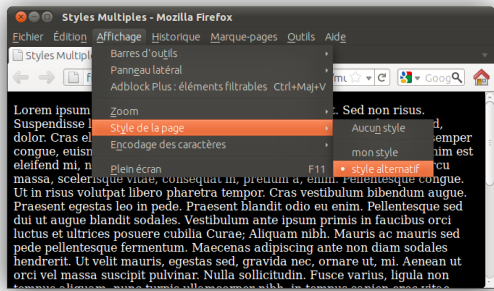
- L'instruction `@import` permet d'importer du contenu CSS à partir d'un fichier externe depuis une autre feuille de styles : `@import url(/url_fichier.css);`
- Les instructions `@import` doivent précéder toute autre instruction (à part `@charset`).
- Ex :

```
1 @import url(/menu.css);  
2 @import url(/main.css);  
3 h1, h2, h3, h4 { color: blue; }  
4 ...
```

Styles multiples

- Il est possible d'inclure plusieurs ensembles de styles :

```
1 <head>
2   ...
3   <style title="mon_style" media="screen">
4     body { background-color : lightgray ; }
5   </style>
6   <style title="style_alternatif" media="screen">
7     body { background-color : black ; color :white }
8   </style>
9 </head>
```



Style par média

- Une feuille de style peut être sélectionnée automatiquement en fonction du média cible (ex : écran ou impression).

- Dans le code CSS :

```
1 @media screen { body { background-color : lightgray ; } }  
2 @media print { body { font-size : 12pt ; } }
```

- Lors de l'import :

```
1 @import(/impression.css) print ;  
2 @import(/ecran.css) screen ;
```

- À la liaison :

```
1 <head>  
2 ...  
3 <link rel="stylesheet" type="text/css" href="ecran.css" media="screen">  
4 <link rel="stylesheet" type="text/css" href="impression.css" media="print">  
5 </head>
```

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Selecteurs

Syntaxe
Cascade
Priorité

Propriétés

Typographie
texte
Unités
Boîtes
Gestion du flux
Positionnement

Frameworks

—

Les différents médias possibles sont :

- **all** : toutes les cibles,
- **screen** : écrans d'ordinateur,
- **print** : impression sur papier,
- **handheld** : terminal mobile de petite taille,
- **projection** : projecteurs video,
- **speech** : liseuse à synthèse vocale,
- **braille/embossed** : terminal/imprimante braille,
- **tty** : terminal texte,
- **tv** : navigateur sur télévision (WebTV).

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Selecteurs

Syntaxe
Cascade
Priorité

Propriétés

Typographie
texte
Unités
Boîtes
Gestion du flux
Positionnement

Frameworks

Les sélecteurs

selecteur { [propriété : valeur;]⁺ }

Recherche de motifs ¹ appliquée à un arbre

selecteur { [**propriété : valeur;]⁺ }**

- Les sélecteurs sont des expressions qui permettent de sélectionner un ensemble d'éléments (nuds) de l'arbre du document.
- Les nuds sélectionnés se voient appliquer le bloc d'affectations de propriété associé au sélecteur.
- Ces expressions permettent de décrire un chemin ou un ensemble de chemins dans un arbre.
- L'élément sélectionné est toujours en bas d'un tel chemin.

Recherche de motifs appliquée à un arbre

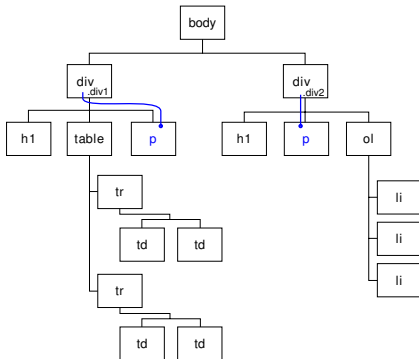
Exemple

selecteur { [propriété : valeur;]⁺ }

Exemple de motifs :

1 `div > p { color : blue ; }`

Tout paragraphe
descendant d'une
div.



Syntaxe des sélecteurs

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Selecteurs

Syntaxe
Cascade
Priorité

Propriétés

Typographie
texte
Unités
Boîtes
Gestion du flux
Positionnement

Frameworks

selecteur { [**propriété : valeur;]⁺ }**

Comme toute expression, Les sélecteurs se définissent à partir :

- de symboles terminaux : (éléments, propriétés, classes, identifiants)
- et d'opérateurs permettant de les combiner.

Toutefois, nous verrons que la récurrence ne produit pas de multiple sous-expressions.

Syntaxe des sélecteurs

Les terminaux

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Sélecteurs

Syntaxe
Cascade
Priorité

Propriétés

Typographie
texte
Unités
Boîtes
Gestion du flux
Positionnement

Frameworks

Sélecteurs simples :

- `*` : universel, tous les éléments,
- `p` : nom d'élément,
- `.maiclass` ou `*.maiclass` : classe d'élément :

```
1 <div class="maiclass">blabla ... </div>
```

- `P.maiclass` : élément et classe :

```
1 <p class="maiclass">selected ... </p>  
2 <div class="maiclass">not selected ... </div>
```

- `#monid` : par identifiant (unique dans le document) :

```
1 <p id="monid">selected ... </p>
```

Syntaxe des sélecteurs

Les terminaux

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Sélecteurs

Syntaxe
Cascade
Priorité

Propriétés

Typographie
texte
Unités
Boîtes
Gestion du flux
Positionnement

Frameworks

–

Pseudo classes (liens) :

- `:link` : liens non visités,
- `:visited` : liens visités,
- `:active` : liens en train d'être cliqué,
- `:hover` : lien survolé,
- `:focus` : lien ayant le focus,

Autres pseudo classes :

- `:first-child` : le nud est sélectionné s'il est le premier fils de son parent, ex `p:first-child`,
- `:lang` : sélectionne un élément d'après son langage, ex `p:lang(fr)`,

Syntaxe des sélecteurs

Les terminaux

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Selecteurs

Syntaxe
Cascade
Priorité

Propriétés

Typographie
texte
Unités
Boîtes
Gestion du flux
Positionnement

Frameworks

Pseudo éléments :

- `:first-line` : dénote la première ligne d'un flux texte,
- `:first-letter` : dénote la première lettre d'un flux texte,
- `:before` : dénote un élément virtuel avant l'élément sélectionné,
- `:after` : dénote un élément virtuel après l'élément sélectionné, ex :

```
1  blockquote :before { content : open-quote ; }  
2  blockquote :after { content : close-quote ; }
```

Syntaxe des sélecteurs

Les terminaux

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Sélecteurs

Syntaxe
Cascade
Priorité

Propriétés

Typographie
texte
Unités
Boîtes
Gestion du flux
Positionnement

Frameworks

—

Les sélecteurs d'attributs :

- `[att]` : sélectionne tout élément portant l'attribut `att`, quelque-soit sa valeur,
- `[att=val]` : sélectionne tout élément dont l'attribut `att` vaut `val`,
- `[att~=val]` : idem mais `att` prend une valeur dans `val` qui est une liste séparée par des espaces.
- ...

Syntaxe des sélecteurs

Les terminaux

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Sélecteurs

Syntaxe
Cascade
Priorité

Propriétés

Typographie
texte
Unités
Boîtes
Gestion du flux
Positionnement

Frameworks

- Les sélecteurs simples se combinent de manière précise : nom d'élément ou * (implicite) suivi d'un nombre arbitraire de classe, identifiant, pseudo-classe, sélecteur d'attribut.
- Exemples :

```
1 *.maclasse { ... }  
2 div#monid { ... }  
3 .maclasse[lang="fr"] :hover
```

Syntaxe des sélecteurs

Les opérateurs

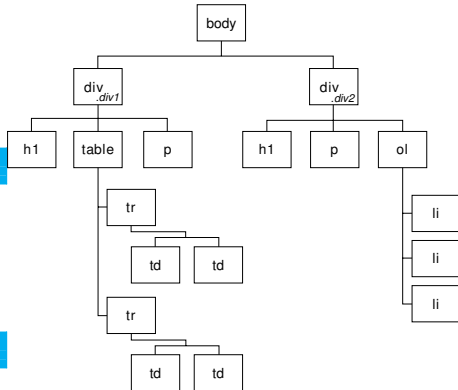
Sélecteurs par lien de parenté :

- **e1 e2** : tous les e2 descendants de e1 ;
ex :

1 `div .div2 li { ... }`

- **e1 > e2** : tous les éléments e2 fils directs de e1 ; ex :

1 `div .div1 > table { ... }`



Syntaxe des sélecteurs

Les opérateurs

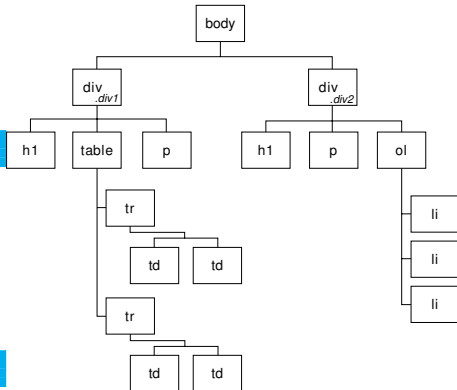
Sélecteurs par lien de parenté (suite) :

- $e1 + e2$: tous les $e2$ successeurs de $e1$; ex :

1 `table + p { ... }`

- $e1 \sim e2$: tous les $e2$ suivant indirectement un $e1$; ex :

1 `h1 ~\textasciitilde ol { ... }`



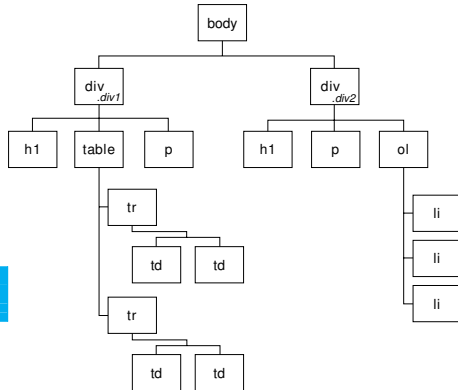
Syntaxe des sélecteurs

Les opérateurs

Sélecteurs multiples :

- e_1, e_2, e_3, \dots : tout nud satisfaisant au moins une des expressions e_1, e_2, e_3, \dots :

1 `table , div.div1 , .div2 > p
{...}`



Syntaxe des sélecteurs

Grammaire simplifiée (tentative)

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Sélecteurs

Syntaxe
Cascade
Priorité

Propriétés

Typographie
texte
Unités
Boîtes
Gestion du flux
Positionnement

Frameworks

—

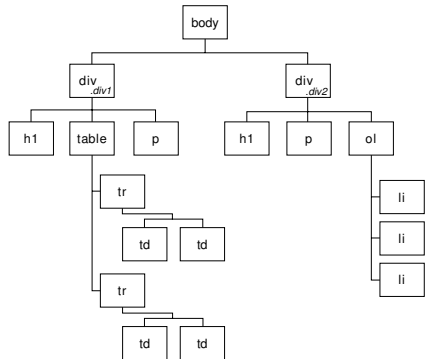
```
liste : liste ',' selector
      | selector;
selector : [suite operateur]* suite [pselement]?;
operateur : /* vide */
          | '+' | '>' | '~';
suite : [ ELEMENT | '*' ]? [ '#' | class | attrib | psclass ]+
      | ELEMENT | '*';
class : '.' IDENT;
attrib : '[' IDENT '=' VAL ']';
psclass : ':' IDENT;
pselement : ':' IDENT;
```

- Une expression est une suite de sélecteurs (précédence = ordre de lecture),
- les pseudos-éléments n'interviennent qu'en sélection finale
- ex :

```
.menu:firstchild > div:hover + p::first-line {  
... }
```

Cascade de styles

La *cascade* est *similaire* à la notion d'héritage² : le style d'un élément est reporté sur ses descendants (sous-arbre) jusqu'à ce qu'une règle CSS l'altère.



- 1 `.div1 { color : blue ; }`
- 2 `tr + tr { color : yellow ; }`

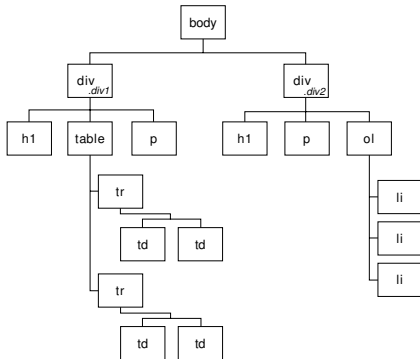
2. Excepté qu'elle intervient sur un arbre d'objets (et non de classe).

Cascade de styles

La *cascade* est *similaire* à la notion d'héritage² : le style d'un élément est reporté sur ses descendants (sous-arbre) jusqu'à ce qu'une règle CSS l'altère.

```
1 .div1 { color : blue ; }  
2 tr + tr { color : yellow ; }
```

Le texte du sous-arbre *div1* est en bleu à l'exception de la seconde ligne du tableau qui est en jaune. Le reste de l'arbre est inchangé.



2. Excepté qu'elle intervient sur un arbre d'objets (et non de classe).

Cascade de styles

Pas systématique !

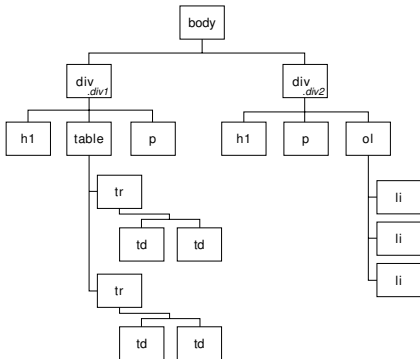


Certains attributs ne sont pas cascadés !

1

```
table { border-color : blue ; }
```

Seule la table aura une bordure, pas les éléments internes (tr, td, etc.).



W3School (`boder-color`) :

Default value:

The current color of the element

Inherited:

no

Animatable:

yes. [Read about animatable](#)

Version:

CSS1

JavaScript syntax:

`object.style.borderColor="#FF0000 blue"`

Try it

Try it

Conflits de règles

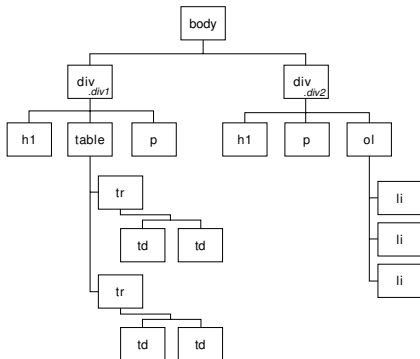
Le mécanisme de règle CSS peut conduire à des conflits, i.e. le moteur CSS peut trouver plusieurs règles qui s'appliquent à un élément pour les mêmes attributs.

Conflits :

```
1 .div1 { color : blue ; }  
2 div { color : yellow ; }  
3 body > div :first-child {  
4   ^^!color : red ; }
```

Pas de conflits :

```
1 .div1 { color : blue ; }  
2 div { background-color : yellow ; }  
3  
4 body > div :first-child {  
   ^^!border-color : red ; }
```



Conflits de règles

Un ordre de précedence est appliqué sur les sélecteurs :

- 1 les définition de style en ligne,
 - 2 les *ids*,
 - 3 les classes, pseudo-classe et les attributs,
 - 4 éléments et pseudo-éléments,
 - 5 le sélecteur universel (*),
 - 6 les sélecteurs par défaut.
- Si deux règles ont la même priorité, c'est la dernière définie qui s'applique.
 - Pour protéger un style contre la surcharge, on peut utiliser l'attribut important :

```
1  ^^body { background-color : white !important ; }
```

Priorité de règles

Le degré de priorité d'une règle est un nombre à quatre
« chiffres » :

$$X_4 X_3 X_2 X_1$$

où

- x_4 vaut 1 si le style est prioritaire (en ligne ou !important), 0 sinon ;
- x_3 est le nombre d'identifiants (ids) dans le sélecteur ;
- x_2 est le nombre de classes (ids) dans le sélecteur ;
- x_1 est le nombre d'éléments séparés par des espaces dans le sélecteur ;

Priorité de règles

Exemples

Style	Style local ou !important	Nombre d'identifiants	Nombre de classes	Nombre d'éléments	Priorité
* {...}	0	0	0	0	0000
p {...}	0	0	0	1	0001
div p {...}	0	0	0	2	0002
.class {...}	0	0	1	0	0010
p.class {...}	0	0	1	1	0011
div p.class {...}	0	0	1	2	0012
#id {...}	0	1	0	0	0100
p#id {...}	0	1	0	1	0101
div p#id {...}	0	1	0	2	0102
.class #id {...}	0	1	1	0	0110
.class p#id {...}	0	1	1	1	0111
div.class p#id {...}	0	1	1	2	0112
<p style="...">	1	0	0	0	1000
..{...!important;}	1	0	0	0	1000

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Selecteurs

Syntaxe
Cascade

Priorité

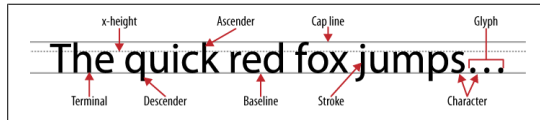
Propriétés

Typographie
texte
Unités
Boîtes
Gestion du flux
Positionnement

Frameworks

—

Les polices



- `font` : raccourci pour les propriétés de polices,
- `font-family` : nom de la police utilisée, plusieurs noms sont donnés et évalués dans l'ordre,
- `font-size` : taille de la police utilisée : unités habituelles ou liées aux polices `xx-small`, `x-small`, `small`, `medium`, `large`, `x-large`, `xx-large`, ou unités relatives `larger`, `smaller`,
- `font-style` : style de la police par défaut : `normal`, `italic` ou `oblique`,
- `font-variant` : variant de la police par défaut : `normal` ou `small-caps`,
- `font-weight` : épaisseur de la police par défaut : `normal`, `bold`, `bolder`, `lighter`, `100` (`lighter`), `200`, `300`, `400`

Introduction

Principe

Liaison

Inclusion

CSS Multiples

Selecteurs

Syntaxe

Cascade

Priorité

Propriétés

Typographie

texte

Unités

Boîtes

Gestion du flux

Positionnement

Frameworks

—

Ajout de nouvelles polices

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Selecteurs

Syntaxe
Cascade
Priorité

Propriétés

Typographie
texte
Unités
Boîtes
Gestion du flux
Positionnement

Frameworks

- Format de la police à ajouter :
PFR (Portable Font Resource) ou TrueDoc.

- Forme générique de l'ajout d'une police :

```
@font-face {  
    font-family: nouvelle_police;  
    src:         url("Polices/fic_police.pfr");  
    font-weight: normal;  
}
```

Propriétés liées au texte

Introduction

Principe

Liaison

Inclusion

CSS Multiples

Selecteurs

Syntaxe

Cascade

Priorité

Propriétés

Typographie

texte

Unités

Boîtes

Gestion du flux

Positionnement

Frameworks

—

- `letter-spacing`, `word-spacing` : espace entre deux lettres (`letter`) ou entre deux mots (`word`) ;
- `line-height` : hauteur entre deux lignes successives ;
- `text-align` : alignement horizontal du texte dans le container : `left`, `right`, `center`, `justify` ;
- `text-decoration` : décoration de la police par défaut : `none`, `underline`, `overline`, `line-through`, `blink` ;
- `text-indent` : décalage de la première ligne ;
- `text-transform` : transformation du texte : `capitalize`, `uppercase`, `lowercase`, `none` ;
- `vertical-align` : alignement vertical du texte sur la ligne : `baseline`, `middle`, `sub`, `super`, `text-top`, `text-bottom` ;

Unités de mesure

Introduction

Principe

Liaison

Inclusion

CSS Multiples

Selecteurs

Syntaxe

Cascade

Priorité

Propriétés

Typographie

Texte

Unités

Boîtes

Gestion du flux

Positionnement

Frameworks

- unités absolues :

inch (**in**) un pouce (2,54 centimètres) ;
millimètre (**mm**) un millimètre (0,0394 pouce) ;
centimètre (**cm**) un centimètre (0.394 pouce) ;
point (**pt**) unité liée aux fontes (18pt=0.25in) ;
pica (**pc**) unité liée aux fontes (12pc=18pt).

- unités relatives :

pixel (**px**) un pixel (unité physique à l'écran)
em (**em**) la valeur de **font-size** pour le container englobant (largeur d'un M majuscule) ;
ex (**ex**) la hauteur d'un x minuscule pour la police utilisée pour le container englobant ;

- autres unités :

pourcentage (%) pourcentage des dimensions du container englobant ;
auto (**auto**) calcul automatique par le navigateur (marges essentiellement).

Dimensions d'une boîte

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Selecteurs

Syntaxe
Cascade
Priorité

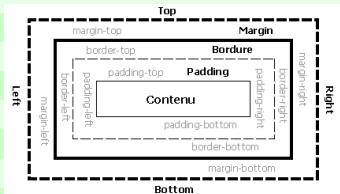
Propriétés

Typographie
texte
Unités
Boîtes
Gestion du flux
Positionnement

Frameworks

—

<code>height</code>	hauteur ;
<code>min-height</code>	hauteur minimale ;
<code>max-height</code>	hauteur maximale ;
<code>width</code>	largeur ;
<code>min-width</code>	largeur minimale ;
<code>max-width</code>	largeur maximale ;
<code>margin</code>	espace entre la bordure et le container englobant ;
<code>padding</code>	espace de remplissage entre la bordure et le texte ;



Individualisation des bordures :

- `margin-left: 4px;` ;
- `margin: 10px 20px 30px 40px;` haut, droite, bas et gauche.

Attention au *margin collapse*.

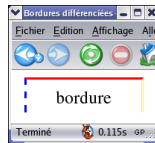
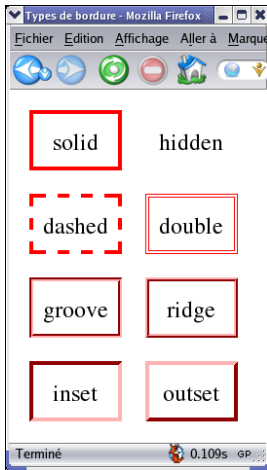
Types de bordures

`border: 1px solid black;`

- `border-width: 1px;`
- `border-style: solid;`
- `border-color: black;`

Individualisation des bordures

- `border-left-style: dashed;`
- `border-right-style: double;`
- `border-left-color: blue;`

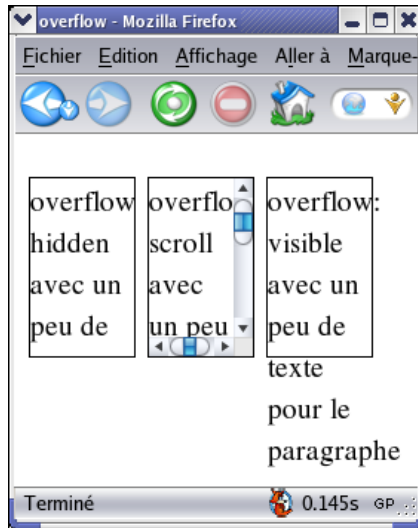


Contenu d'une boîte

La propriété **overflow** permet d'indiquer la conduite à tenir lorsque le contenu déborde de la boîte.

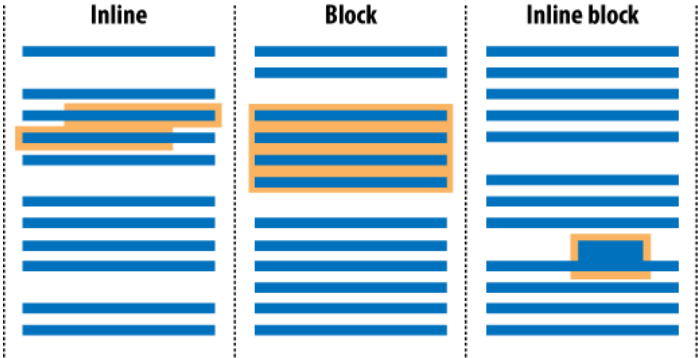
Valeurs possibles :

hidden, scroll,
visible



Flottaison des éléments

Les trois principales catégories de flottaison :



Introduction

Principe
Liaison
Inclusion
CSS Multiples

Selecteurs

Syntaxe
Cascade
Priorité

Propriétés

Typographie
texte
Unités

Boîtes

Gestion du flux

Positionnement

Frameworks

—

Flottaison des éléments

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Selecteurs

Syntaxe
Cascade
Priorité

Propriétés

Typographie
texte
Unités
Boîtes
Gestion du flux
Positionnement

Frameworks

–

- la propriété **display** permet de modifier la relation d'un élément avec le flux de texte.
- Valeurs possibles : **inline**, **block**, `list-item`, `run-in`, **inline-block**, `table`, `inline-table`, `table-row-group`, `table-header-group`, `table-footer-group`, `table-row`, `table-column-group`, `table-column`, `table-cell`, `table-caption`, **none** ;
- Attention, `none` rend l'élément invisible.

Positionnement flottant des containers

`float : left | right | none ;`

- le container ayant la propriété `float` est placé lorsqu'il est rencontré dans le document HTML ;
- il est placé relativement au container englobant ;
- la suite du document se place à l'autour du container flottant.

Code CSS

Code HTML

```
<div id="i1">Division 1</div>
<div id="i2">Division 2</div>
<p>Le texte est placé après
les balises de divisions, il
va mettre en évidence les
boîtes flottantes.</p>
```

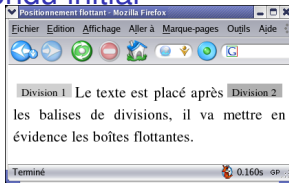
```
#i1 { background-color: #d0d0d0;
      float: left; }

#i2 { background-color: #b0b0b0;
      float: right; }

#i1, #i2 { width: 80px;
            height: 20px;
            margin: 5px;
            text-align: center; }
```

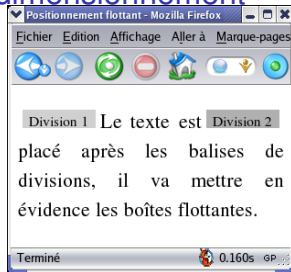
Modifications du positionnement flottant

Rendu initial



- la division `i1` est flottante à gauche ;
- la division `i2` est flottante à droite ;
- le paragraphe s'organise en fonction des containers flottants déjà rencontrés (placés).

Après redimensionnement



- aucune modification des containers flottants ;
- réorganisation automatique du paragraphe.

Positionnement flottant des containers

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Selecteurs

Syntaxe
Cascade
Priorité

Propriétés

Typographie
texte
Unités
Boîtes

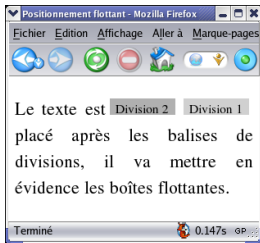
Gestion du flux
Positionnement

Frameworks

Nouveau code CSS

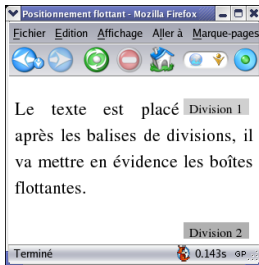
```
#i1{background-color: #d0d0d0;}  
#i2{background-color: #b0b0b0;}
```

```
#i1, #i2 { width:      80px;  
            height:    20px;  
            margin:    5px;  
            float:      right;  
            text-align: center; }
```



Nouveau code HTML

```
<div id="i1">Division 1</div>  
<p>Le texte est placé après les  
balises de divisions, il va  
mettre en évidence les boîtes  
flottantes.</p>  
<div id="i2">Division 2</div>
```



Modifications du positionnement flottant

Rendu et explications

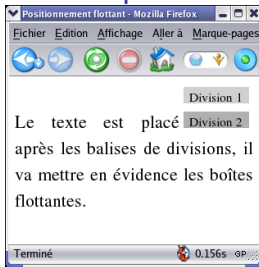
Nouveau code CSS

```
#i1 { background-color: #d0d0d0; }  
#i2 { background-color: #b0b0b0; }  
.clear { clear: both; }
```

```
#i1, #i2 { width:      80px;  
            height:    20px;  
            margin:     5px;  
            float:      right;  
            text-align: center; }
```

Nouveau code HTML

```
<div id="i1">Division 1</div>  
<div classe="clear"/>  
<div id="i2">Division 2</div>  
<p>Le texte est placé après les  
    balises de divisions, il va  
    mettre en évidence les boîtes  
    flottantes.</p>
```



- 1 placement de `i1` ;
- 2 réinitialisation du placement ;
- 3 placement de `i2` ;
- 4 placement du paragraphe, relativement à `i2` (sans tenir compte de `i1`).

Modifications du positionnement flottant

Introduction

Principe

Liaison

Inclusion

CSS Multiples

Selecteurs

Syntaxe

Cascade

Priorité

Propriétés

Typographie

texte

Unités

Boîtes

Gestion du flux

Positionnement

Frameworks

—

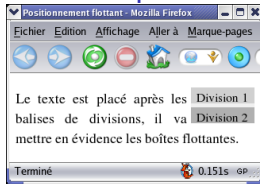
Nouveau code CSS

```
#i0 { background-color: #f0f0f0;
      width: 80px;
      height: 45px; /* 20+5+20 */
      float: right;
      text-align: center;
      margin-left: 5px;
      margin-bottom: 5px;}
#i1, #i2 { width: 80px;
           height: 20px;}
#i1 { margin-bottom: 5px;}
```

Nouveau code HTML

```
<div id="i0">
  <div id="i1">Division 1</div>
  <div id="i2">Division 2</div>
</div>
<p>Le texte est placé après les
  balises de divisions, il va
  mettre en évidence les boîtes
  flottantes.</p>
```

Rendu et explications



- 1 placement de `i0` :
- 1 placement de `i1` ;
2 placement de `i2` ;
- 2 placement du paragraphe, relativement à `i0`.

Propriétés de positions

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Selecteurs

Syntaxe
Cascade
Priorité

Propriétés

Typographie
texte
Unités
Boîtes
Gestion du flux
Positionnement

Frameworks

–

- La position d'un élément peut être donnée par rapport à ses quatre bords : `top`, `right`, `bottom`, `left`.

L'effet de ces attributs dépend de la valeur de `position` :

- **`static`** : par défaut, position attendue le flux,
- `fixed` : l'élément reste à sa position,
- `absolute` : position par rapport au bord supérieur gauche du canevas du navigateur,
- `relative` : position relative à la position attendue.

Propriétés de positions

Un exemple

Supposons le code suivant :

```
1 <style type="text/css">
2 #someDiv { left : 160px ; top : 96px ; }
3 </style>
4 <body><div id="main"> ... <div id="someDiv">The quick
5 red fox jumps over the lazy brown dog.</div> ... </div></body>
```

Effets de `position` sur `#someDiv` :

- `static` : *left* et *top* ne sont pas appliqués, l'élément est à la place attendue dans le flux ;
- `absolute` : 160 pixels à gauche et 96 pixels en bas du coins supérieur gauche de la page,
- `fixed` : comme *absolute*, mais l'élément garde sa position en cas de défilement,
- `relative` : 160 pixels à gauche et 96 pixels en bas à partir de la position attendue.

Superpositions avec z-index

Code HTML

```
<body>
  <div id="c1">Division c1</div>
  <div id="c2">Division c2</div>
</body>
```

Code CSS

```
#c1 { background-color: #d0d0d0;
      position:         absolute;
      top:              10px;
      left:             30px;
      width:            200px;
      text-align:       center;
    }

#c2 { background-color: #b0b0b0;
      position:         absolute;
      top:              20px;
      left:             20px;
      width:            200px;
      text-align:       center;
    }
```

Explication du positionnement



- 1 `<div id="c1">` est rencontrée et placée en (10,30) ;
- 2 `<div id="c2">` est rencontrée et placée en (20,20) ;
- 3 c2, placée en dernier, masque c1.

Superpositions avec z-index

Code HTML

```
<body>
  <div id="c1">Division c1</div>
  <div id="c2">Division c2</div>
</body>
```

Code CSS

```
#c1 { background-color: #d0d0d0;
      position:         absolute;
      z-index:          20;
      top:              10px;
      left:             30px;
      width:            200px;
      text-align:       center;
    }

#c2 { background:        #b0b0b0;
      position:         absolute;
      z-index:          10;
      top:              20px;
      left:             20px;
      width:            200px;
      text-align:       center;
    }
```

Rendu et explications



- 1 `<div id="c1">` est rencontrée et placée en (10,30) ;
- 2 `<div id="c2">` est rencontrée et placée en (20,20) ;
- 3 les valeurs de la propriété `z-index` de `c1` et de `c2` sont comparées, on place les divisions par ordre croissant de valeur de `z-index` ;
- 4 `c1`, placée en dernier, masque `c2`.

Superpositions avec z-index

Rendu et explications

Code CSS

```
#c1 { background-color: #d0d0d0;
      position:         absolute;
      z-index:           20;
      top:               10px;
      left:              30px;
      width:             200px;
      text-align:        center;
    }

#c1:hover { z-index: 100; }

#c2 { background:         #b0b0b0;
      position:         absolute;
      z-index:           10;
      top:               20px;
      left:              20px;
      width:             200px;
      text-align:        center;
    }

#c2:hover { z-index: 100; }
```



- 1 lorsque le curseur passe sur `c2`, la propriété `z-index` prend pour valeur 100 ;
- 2 `c2`, alors placée en dernier, masque `c1` ;
- 3 lorsque le curseur passe sur `c1`, la propriété `z-index` prend pour valeur 100 ;
- 4 `c1`, alors placée en dernier, masque `c2` ;

Frameworks CSS

Des frameworks (*Cadriciels*) pour CSS ?

- Les feuilles de styles peuvent être laborieuses à créer pour des projets d'envergure,
- On fini par reprendre des feuilles existantes,
- Pouvoir d'expression limité pour exprimer les propriétés d'une interface :

```
.i1, .i2 { width: 80px; height: 20px; margin: 5px;  
          float: right; text-align: center; }
```

```
.i0 {  
    background-color: #f0f0f0; /* $boxcolor */  
    width:           80px;  
    height:          45px; /* 20+5+20 */  
    float:           right;
```

- Exemple : *Twitter bootstrap, Less, SASS, ...*

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Selecteurs

Syntaxe
Cascade
Priorité

Propriétés

Typographie
texte
Unités
Boîtes
Gestion du flux
Positionnement

Frameworks



(Twitter) Bootstrap

- Principe : une collection de feuilles de styles combinables et personnalisables,
- Les feuilles modulaires sont assemblées par une feuille principale,
- Le langage *LESS* et son préprocesseur permettent une rédaction plus concise et plus cohérente.



Préprocesseur CSS

- Principe : augmenter le langage CSS pour générer ce dernier,
- Supporte les variables :

```
@color: #4D926F;  
.header { color: @color; ... }  
h2 { color: @color; ... }
```

- Supporte les mixins (paramétriques) :

```
.rounded-corners (@radius: 5px) {  
  -webkit-border-radius: @radius;  
  -moz-border-radius: @radius;  
  border-radius: @radius;  
}  
#header { .rounded-corners; }  
#footer { .rounded-corners(10px); }
```

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Selecteurs

Syntaxe
Cascade
Priorité

Propriétés

Typographie
texte
Unités
Boîtes
Gestion du flux
Positionnement

Frameworks

—

Frameworks CSS



Préprocesseur CSS

- LESS est écrit en JavaScript,
- il est utilisable en développement dans un terminal ou comme script dans un navigateur,
- Dans un terminal, il utilise `node.js` :

```
$ sudo apt-get install npm  
$ npm install -g less  
$ lessc styles.less styles.css
```

Introduction

Principe
Liaison
Inclusion
CSS Multiples

Selecteurs

Syntaxe
Cascade
Priorité

Propriétés

Typographie
texte
Unités
Boîtes
Gestion du flux
Positionnement

Frameworks

Quelques références

- Sélecteurs CSS3 
- Spécification CSS 2.1 :
[http ://www.w3.org/TR/CSS21/](http://www.w3.org/TR/CSS21/) 
- Valideur CSS :
[http ://jigsaw.w3.org/css-validator/](http://jigsaw.w3.org/css-validator/) 
- Référence rapide :
[http ://www.w3schools.com/cssref/](http://www.w3schools.com/cssref/) 
- Langage et préprocesseur LESS :
[http ://lesscss.org/](http://lesscss.org/) 
- Bootstrap :
[http ://getbootstrap.com](http://getbootstrap.com) 