

CSS

Outils de développement web L1

Olivier Mallet

Université de Rouen

1 Généralités

2 Sélecteurs

- Classes et identifiants
- Sélecteurs utilisant le contexte

3 Propriétés

- Propriétés du texte
- Couleurs et image de fond
- Dimensions, bordures et marges
- Positionnement

Généralités

- **Séparation** fond-forme :
 - HTML → contenu et structure
 - CSS → présentation
- **Feuille de style** : ensemble de règles qui définissent la présentation d'un document HTML
- Une feuille de style peut être définie par
 - l'auteur du site
 - l'utilisateur
 - le navigateur (feuille de style par défaut)
- Conflits possibles entre différentes feuilles de style ⇒ mécanisme de la cascade pour les résoudre (CSS = Cascading Style Sheets)
- Inventées par Håkon Wium Lie et Bert Bos (1994–95)

Généralités (2)

- **Standard** du W3C (1996)
- **Validation** : <http://jigsaw.w3.org/css-validator/>
- Avantages de CSS :
 - Facilite l'écriture et la mise à jour des pages Web
 - Permet de créer des sites moins lourds et plus accessibles
- Une page doit rester lisible même sans styles

Où placer le code CSS ?

- Dans l'attribut `style` des balises HTML (à éviter)
- Dans une balise `<style> </style>` située dans l'en-tête
- Dans un fichier externe avec l'extension `.css` auquel on fait référence depuis la page HTML (balise `<link>` dans l'en-tête)

Exemple

```
<link rel="stylesheet" href="feuille.css" />
```

Exemple : code CSS dans les balises HTML

```
<!DOCTYPE html>

<html lang="fr">
  <head><title>Ma page</title></head>

  <body>
    <h1 style="color: red">Partie 1</h1>
    <p>Texte de la partie 1</p>

    <h1 style="color: red">Partie 2</h1>
    <p>Texte de la partie 2</p>
  </body>
</html>
```

Exemple : code CSS dans l'en-tête de la page

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Ma page</title>
    <style>
      h1 {color: red;}
    </style>
  </head>

  <body>
    <h1>Partie 1</h1>
    <p>Texte de la partie 1</p>

    <h1>Partie 2</h1>
    <p>Texte de la partie 2</p>
  </body>
</html>
```

Exemple : code CSS dans un fichier séparé

Page HTML :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>Ma page</title>
    <link rel="stylesheet" href="feuille.css" />
  </head>

  <body>
    <h1>Partie 1</h1>
    <p>Texte de la partie 1</p>

    <h1>Partie 2</h1>
    <p>Texte de la partie 2</p>
  </body>
</html>
```

Fichier CSS (feuille.css) : `h1 {color: red;}`

Résultat de ces exemples

Partie 1

Texte de la partie 1

Partie 2

Texte de la partie 2

Syntaxe basique

- Feuille de style : suite de règles de la forme
sélecteur {
 propriété1: valeur1;
 propriété2: valeur2;
 ...
}
- Le **sélecteur** définit les éléments auxquels on applique le style
- Sélecteur le plus simple : nom de balise

Exemple (Tous les titres de niveau 1 en rouge)

```
h1 {color: red;}
```

- **Commentaires** : entre /* et */

Sélecteur multiple

Exemple

```
h1, h2, h3 {color: red;}
```

est équivalent à

```
h1 {color: red;}
```

```
h2 {color: red;}
```

```
h3 {color: red;}
```

Classes

- HTML : on peut attribuer une **classe** à certains éléments (attribut `class`)
- Une classe peut contenir plusieurs valeurs séparées par des espaces
- Choisir des noms de classes liés au fond et pas à la forme
- Sélecteur CSS correspondant :
 - `.truc` : tous les éléments ayant la classe `truc` (`truc` est l'une des valeurs contenues dans l'attribut `class`)
 - `machin.truc` : tous les éléments de type `machin` (nom de balise) ayant la classe `truc`
 - `.truc.bidule` : tous les éléments ayant la classe `truc` et la classe `bidule`...

Classes : exemple

Exemple (Dialogue avec une couleur pour chaque personnage)

Code HTML :

```
<p class="personnage1">Oui !</p>  
<p class="personnage2">Non !</p>  
<p class="personnage1">Si !</p>
```

Résultat :

Oui !

Non !

Si !

Code CSS :

```
p.personnage1 {color: red;}  
p.personnage2 {color: blue;}
```

Plusieurs classes pour le même élément

Exemple

- Code CSS :

```
.difficile {color: red;}  
.amusant {font-weight: bold;}
```

- Code HTML :

```
<p class="difficile">Ceci est difficile</p>  
<p class="amusant">Ceci est amusant</p>  
<p class="difficile amusant">Ceci est difficile mais aussi  
amusant</p>
```

- Résultat :

Ceci est difficile

Ceci est amusant

Ceci est difficile mais aussi amusant

Remarque sur les conflits de règles

- Et si on avait écrit `.amusant {color: blue;}` ?
- À chaque sélecteur, on associe une priorité
- En cas de conflit, le sélecteur de plus grande priorité l'emporte
- Si plusieurs sélecteurs ont la même priorité, celui qui est déclaré en dernier l'emporte
- Dans cet exemple, `.difficile` et `.amusant` ont la même priorité : le dernier paragraphe serait donc bleu

Pseudo-classes

- Pour les liens

`:link` lien non visité

`:visited` lien visité

Exemple

```
a:link {color: red;} /* liens non visités en rouge */  
.bidule:visited {color: green;} /* liens visités de classe bidule en vert */
```


Pseudo-classes (2)

- Pour les interactions

:**hover** élément survolé

Exemple (Couleur de fond pour certains paragraphes survolés)

```
p.difficile: hover {background-color: yellow;}
```

:**focus** élément qui accepte les évènements clavier ou souris

Exemple (Fond jaune pour certains champs de formulaire sélectionnés)

```
input: focus, textarea: focus {background-color: yellow;}
```

Identifiants

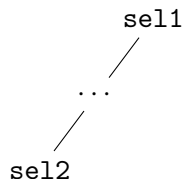
- HTML : on peut attribuer un **identifiant** à certains éléments (attribut `id`)
- Deux éléments distincts ne peuvent avoir le même identifiant
- CSS : sélecteur de la forme `#identifiant`

Exemple (Couleur de fond spéciale pour un menu)

- Code HTML : `<div id="menu"> </div>`
- Code CSS : `#menu {background-color: aqua;}`

Sélecteur de descendant

- Sélecteur de la forme `se11 se12`
- Représente tous les éléments correspondant au sélecteur `se12` contenus dans un élément correspondant au sélecteur `se11`



Exemple

Code HTML :

```
<ul id="maliste">
  <li>Item 1</li>
  <li>Item 2 avec <a href="x.html">un lien</a>
</li>
</ul>
<p><a href="y.html">Un autre lien</a></p>
```

Résultat :

- Item 1
- Item 2
avec
un lien

Un autre lien

Code CSS : `#maliste a {color: gray;}`

Sélecteur de descendant : généralisation

- Sélecteur de la forme `sel1 sel2 ...seln-1 seln`
- Représente tous les éléments correspondant à `seln` contenus dans un élément correspondant à `seln-1` contenu dans ... contenu dans un `sel2` contenu dans un `sel1`



Exemple de la syntaxe précédente

Exemple

Code HTML :

```
<ol>  
  <li>Sortir le plat de son emballage</li>  
  <li>Le faire cuire au micro-ondes  
    <ul>  
      <li>Temps de cuisson : 4 min</li>  
      <li>Puissance maximale</li>  
    </ul>  
  </li>  
  <li>Servir</li>  
</ol>
```

Code CSS :

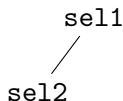
```
ol li ul li {color: red;}
```

Résultat :

1. Sortir le plat de son emballage
2. Le faire cuire au micro-ondes
 - Temps de cuisson : 4 min
 - Puissance maximale
3. Servir

Sélecteur d'enfant

- Sélecteur de la forme `sel1 > sel2`
- Représente tous les `sel2` directement contenus dans un `sel1` (pas d'élément contenant `sel2` et contenu dans `sel1`)



Exemple

Code HTML :

```
<body>
  <div>
    <p>Un paragraphe</p>
  </div>
  <p>Un autre paragraphe</p>
</body>
```

Résultat :

Un paragraphe

Un autre paragraphe

Code CSS : `body > p {color: red;}`

Sélecteur d'enfant : généralisation

On peut généraliser comme le sélecteur de descendant :

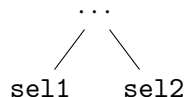
Exemple

```
<ul>
  <li>
    <ol>
      <li><a href="p1.html">Truc</a></li>
      <li><a href="p2.html">Machin</a></li>
    </ol>
  </li>
  <li>
    <a href="p3.html">Bidule</a>
    <em><a href="p4.html">chouette</a></em></li>
  </li>
</ul>
```

Le sélecteur `ul > li > a` permet d'appliquer un style au lien « Bidule »

Sélecteur de successeur immédiat

- Sélecteur de la forme `sel1 + sel2`
- Représente tous les `sel2` qui suivent immédiatement un `sel1`



Exemple

Code HTML :

```
<div class="bidon">
  Un bloc bidon
</div>
<p>Un paragraphe</p>
<p>Un autre paragraphe</p>
```

Code CSS :

```
.bidon + p {background-color: yellow;}
```

Résultat :

Un bloc bidon

Un paragraphe

Un autre paragraphe

Le sélecteur universel

- Notation : *
- C'est un joker qui représente n'importe quel élément

Exemple (On affiche tout le texte en rouge)

```
* {color: red;}
```

- Souvent utilisé en combinaison avec d'autres sélecteurs

Exemple (On affiche sur fond rouge les descendants des paragraphes)

Code HTML :

```
<p>  
  Il y a <a href="a.html">un lien</a> et <em>un point crucial</em>.  
</p>
```

Code CSS : `p * {background-color: red;}`

Résultat : Il y a **un lien** et **un point crucial**.

Unités de longueur

- Unités **absolues**

`cm` centimètre

`mm` millimètre

`in` pouce ($1 \text{ in} = 2,54 \text{ cm}$)

`pt` point ($1 \text{ pt} = \frac{1}{72} \text{ in} \simeq 0,35 \text{ mm}$)

`pc` pica ($1 \text{ pc} = 12 \text{ pt} = \frac{1}{6} \text{ in} \simeq 4,2 \text{ mm}$)

`px` pixel ($1 \text{ px} = \frac{1}{96} \text{ in} \simeq 0,26 \text{ mm}$)

- Unités **relatives**

`em` taille de la police

`ex` hauteur des minuscules

`%` pourcentage

- Pas d'espace avant l'unité ou le signe % (exemples : 50px, 75%)
- Séparateur décimal : le point (exemple : 1.5em)

Propriétés du texte

font-family police spécifique (Arial, Times...) ou famille (serif / sans-serif / monospace...)

- une police serif (ex : Times) s'affiche comme ceci
- une police sans-serif (ex : Arial) comme ceci
- et une police monospace (ex : Courier) comme ceci

Exemple

```
h1 {font-family: Arial;}  
h2 {font-family: sans-serif;}
```

On peut indiquer plusieurs valeurs séparées par ,

Exemple (On utilisera Arial si elle est disponible, sinon Helvetica et en dernier recours sans-serif)

```
h1 {font-family: Arial, Helvetica, sans-serif;}
```

Propriétés du texte (2)

font-size (taille des caractères) longueur / pourcentage
référence pour les unités relatives : taille du texte de
l'élément parent

Exemple (Titres de niveau 1 trois fois plus gros que le texte de leur parent)

```
h1 {font-size: 300%;}
```

Propriétés du texte (3)

`font-style` normal / italic / oblique

- la valeur italic donne *ceci* (avec une police serif)
- la valeur oblique donne *ceci* (idem)

Exemple

Code HTML : `<h2>Tout est en italique sauf ceci</h2>`

Code CSS :

```
h2 {font-style: italic;}  
h2 em {font-style: normal;}
```

Résultat : ***Tout est en italique* sauf ceci**

Propriétés du texte (4)

`font-variant` normal / small-caps (PETITES MAJUSCULES)

Exemple

```
ul li {font-variant: small-caps;}
```

`font-weight` (graisse) normal / bold (gras)

Exemple

```
#menu a {font-weight: bold;}
```

Propriétés du texte (5)

`line-height` (interligne) normal / longueur / pourcentage / nombre

Exemple (Interligne double pour les portions de texte préformaté)

```
pre {line-height: 200%;} ou pre {line-height: 2em;} ou  
pre {line-height: 2;}
```

`text-decoration` none / underline (souligné) / overline (surligné) /
line-through (barré) / plusieurs valeurs

Exemple (Liens surlignés et soulignés quand on les survole)





```
a:hover {text-decoration: underline overline;}
```

Propriétés du texte (6)

text-transform none / capitalize (1^{re} lettre de chaque mot en majuscule) / lowercase (tout en minuscules) / uppercase (tout en majuscules)

Exemple

```
strong {text-transform: uppercase;}
```

text-align (alignement) left  / right  / center  / justify 

Exemple

```
h1 {text-align: center;}
```

text-indent (décalage du début de la 1^{re} ligne) longueur / pourcentage









Exemple (Retrait de première ligne d'1 cm pour tous les paragraphes)

```
p {text-indent: 1cm;}
```










Couleurs

- 16 **mots-clés** en anglais pour les couleurs de base (voir page suivante avec les équivalents RGB)
- Pour utiliser d'autres couleurs, il faut se servir de la notation **RGB** :
 - Notation RGB « classique » : `rgb(R,G,B)` ($0 \leq R, G, B \leq 255$)
 - Notation RGB en hexadécimal : `#RrGgBb` ($00 \leq Rr, Gg, Bb \leq FF$)
 - Notation RGB en pourcentage : `rgb(R%,G%,B%)` ($0 \leq R, G, B \leq 100$)

Couleurs de base (1/2)

	aqua	<code>rgb(0,255,255)</code>	<code>#00FFFF</code>	<code>rgb(0%,100%,100%)</code>
	black	<code>rgb(0,0,0)</code>	<code>#000000</code>	<code>rgb(0%,0%,0%)</code>
	blue	<code>rgb(0,0,255)</code>	<code>#0000FF</code>	<code>rgb(0%,0%,100%)</code>
	fuchsia	<code>rgb(255,0,255)</code>	<code>#FF00FF</code>	<code>rgb(100%,0%,100%)</code>
	gray	<code>rgb(128,128,128)</code>	<code>#808080</code>	<code>rgb(50%,50%,50%)</code>
	green	<code>rgb(0,128,0)</code>	<code>#008000</code>	<code>rgb(0%,50%,0 %)</code>
	lime	<code>rgb(0,255,0)</code>	<code>#00FF00</code>	<code>rgb(0%,100%,0%)</code>
	maroon	<code>rgb(128,0,0)</code>	<code>#800000</code>	<code>rgb(50%,0%,0%)</code>

Couleurs de base (2/2)

	navy	<code>rgb(0,0,128)</code>	<code>#000080</code>	<code>rgb(0%,0%,50%)</code>
	olive	<code>rgb(128,128,0)</code>	<code>#808000</code>	<code>rgb(50%,50%,0%)</code>
	purple	<code>rgb(128,0,128)</code>	<code>#800080</code>	<code>rgb(50%,0%,50%)</code>
	red	<code>rgb(255,0,0)</code>	<code>#FF0000</code>	<code>rgb(100%,0%,0%)</code>
	silver	<code>rgb(192,192,192)</code>	<code>#C0C0C0</code>	<code>rgb(75%,75%,75%)</code>
	teal	<code>rgb(0,128,128)</code>	<code>#008080</code>	<code>rgb(0%,50%,50%)</code>
	white	<code>rgb(255,255,255)</code>	<code>#FFFFFF</code>	<code>rgb(100%,100%,100%)</code>
	yellow	<code>rgb(255,255,0)</code>	<code>#FFFF00</code>	<code>rgb(100%,100%,0%)</code>

Couleurs de premier plan et d'arrière-plan

`color` couleur du texte

Exemple

```
a:visited {color: gray;}
```

`background-color` couleur de fond

Exemple

```
#menu {background-color: #CCFFFF;}
```

Autres propriétés de l'arrière-plan

background-image url("chemin/image") (chemin absolu ou relatif par rapport au fichier CSS)

background-attachment fixed (l'image de fond reste fixe par rapport à la page) / scroll (par défaut : elle défile)

background-repeat no-repeat (image de fond non répétée) / repeat-x (répétée horizontalement) / repeat-y (répétée verticalement) / repeat (par défaut : répétée dans les 2 directions)

background-position position horizontale (left / center / right / longueur / pourcentage) et position verticale (top / center / bottom / longueur / pourcentage) séparées par un espace

background raccourci pour toutes ces propriétés (y compris background-color)

Propriétés de l'arrière-plan : exemple

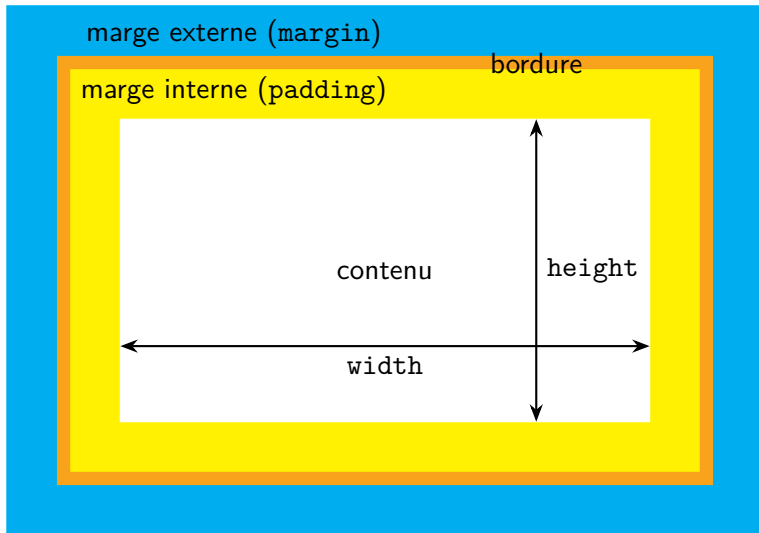
Exemple

```
body {  
  background: url("mer.jpg") no-repeat left bottom;  
}
```

est équivalent à

```
body {  
  background-image: url("mer.jpg");  
  background-repeat: no-repeat;  
  background-position: left bottom;  
}
```

Modèle de boîte



Les dimensions du contenu d'une boîte

`width` largeur du contenu

`height` hauteur du contenu

- Pas pour les éléments en ligne
- Valeurs possibles de ces 2 propriétés : longueur / pourcentage / auto (valeur par défaut)

Exemple (Tous les paragraphes occuperont 50 % de la largeur disponible)

```
p {  
  width: 50%;  
}
```


Bordures

border-style none / hidden / solid / dashed / dotted / double /
groove / ridge / inset / outset

border-width longueur / thin / medium / thick

border-color couleur

border raccourci pour spécifier ces 3 propriétés

Exemple

div {border: 2px dashed red;} est équivalent à

```
div {  
  border-width: 2px;  
  border-style: dashed;  
  border-color: red;  
}
```

Valeurs par défaut des propriétés de bordure

- La valeur par défaut de `border-style` est `none` \Rightarrow toujours **donner une valeur** à `border-style` si on veut que la bordure soit visible !
- La valeur par défaut de `border-width` est `medium`
- La valeur par défaut de `border-color` est la couleur du texte

Exemple

```
/* pas de bordure (border-style=none) */
```

```
p {border: 1px black}
```

```
/* le texte et la bordure sont en rouge
```

```
border-width a la valeur medium */
```

```
div {  
  color:red;  
  border-style: solid;  
}
```

Exemples pour la propriété border-style

solid

dashed

dotted

double

groove

ridge

inset

outset

Différents styles pour les 4 bordures

- Propriétés spécifiques : `border-left-width`, `border-top-color`...

Exemple (Bordure rouge double de 10 pixels pour tous les paragraphes, uniquement à gauche)

```
p {  
  border-left-width: 10px;  
  border-left-style: double;  
  border-left-color: red;  
}
```

- Propriétés `border-top`, `border-right`, `border-bottom`, `border-left` (s'utilisent comme `border`)

Exemple (Même définition de style en plus concis)

```
p {border-left : 10px double red;}
```

Différents styles pour les 4 bordures (2)

On peut donner plusieurs valeurs aux propriétés `border-style`, `border-width` ou `border-color` :

- 2 valeurs : haut_et_bas droite_et_gauche

Exemple

```
border-style: dashed dotted;
```



- 3 valeurs : haut droite_et_gauche bas

Exemple

```
border-color: blue green red;
```



- 4 valeurs : haut droite bas gauche

Exemple

```
border-width: 0px 1px 2px 3px;
```



Marges

- Marges **extérieures** :

- La propriété `margin` définit les 4 marges extérieures
- Valeurs possibles : longueur / pourcentage / `auto` (permet de centrer un bloc horizontalement)

Exemple (Tous les titres de niveau 1 ou 2 sont centrés)

```
h1, h2 {  
  width: 50%;  
  margin: auto;  
}
```

- Pour avoir des marges différentes sur les 4 bords :
 - On peut donner 2, 3 ou 4 valeurs à `margin` (comme pour les bordures)
 - propriétés spécifiques : `margin-top`, `margin-right`, `margin-bottom`, `margin-left`
- Marges **intérieures** : idem en remplaçant `margin` par `padding` (la valeur `auto` n'est pas autorisée)

Positionnement dans le flux courant (ou flux normal)

- Un bloc est placé sous son frère précédent et occupe toute la largeur disponible dans son conteneur
- Un élément en ligne se place à la suite du précédent sur la même ligne (on passe à la ligne s'il n'y a plus de place)
- Positionnement par défaut qu'on peut changer avec la propriété `position`
- Un élément est dit *positionné* si on a défini une valeur pour sa propriété `position`

Positionnement dans le flux courant : exemple

Exemple

Code HTML :

```
<p>Un paragraphe</p>  
<p id="truc"> Un autre</p>  
<p>Encore un autre</p>
```

Code CSS :

```
p {border: solid;}
```

Résultat :

Un paragraphe

Un autre

Encore un autre

Positionnement relatif

- `position: relative;`
- Variante du positionnement dans le flux courant
- On peut décaler l'élément par rapport à sa position normale avec les propriétés `top`, `right`, `bottom` et `left`

Attention !

`left: 20px;` décale de 20 pixels à droite (par rapport au bord gauche) !

- Les autres éléments ne sont pas affectés par ce décalage
- Permet de superposer des éléments

Positionnement relatif : exemple

Exemple (Positionnement relatif sans décalage)

Code HTML :

```
<p>Un paragraphe</p>  
<p id="truc"> Un autre</p>  
<p>Encore un autre</p>
```

Code CSS :

```
p {border: solid;}  
#truc {  
  position: relative;  
}
```

Résultat :

Un paragraphe

Un autre

Encore un autre

Rien ne change par rapport au positionnement dans le flux courant !

Positionnement relatif : exemple (2)

Exemple (Positionnement relatif avec des décalages)

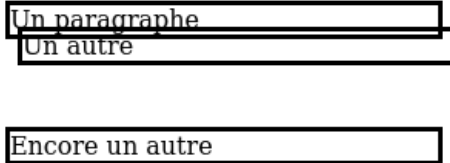
Code HTML :

```
<p>Un paragraphe</p>  
<p id="truc"> Un autre</p>  
<p>Encore un autre</p>
```

Code CSS :

```
p {border: solid;}  
#truc {  
  position: relative;  
  bottom : 1.5em;  
  left : 8px;  
}
```

Résultat :



Positionnement absolu

- `position: absolute;`
- L'élément sort du flux normal
- Les autres éléments se placent comme s'il n'existait pas
- Décalage avec `top`, `right`, `bottom`, `left`
- Position de référence : premier ancêtre positionné (en relatif ou absolu) ou la fenêtre du navigateur

Positionnement absolu : exemple

Exemple

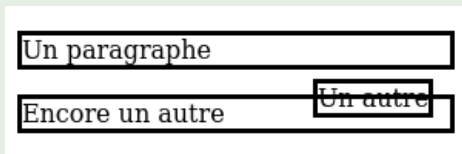
Code HTML :

```
<body>  
  <p>Un paragraphe</p>  
  <p id="truc"> Un autre</p>  
  <p>Encore un autre</p>  
</body>
```

Code CSS :

```
p {border: solid;}  
#truc {  
  position: absolute;  
  left : 200px;  
  bottom : 10px;  
}
```

Résultat :



Ordre d'empilement

- Les positionnements relatif et absolu permettent de superposer des éléments
- Cas par défaut : l'élément positionné déclaré en dernier dans le code HTML passe devant
- Pour modifier ce comportement : propriété `z-index` (ne marche qu'avec les éléments positionnés)
- Si 2 éléments sont superposés, celui qui a le `z-index` le plus grand passe devant

z-index : exemple

Exemple

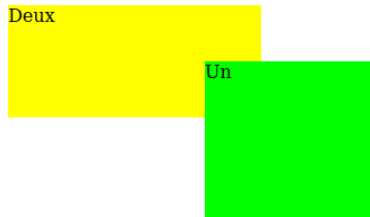
Code HTML :

```
<p id="p1">Un</p>  
<p id="p2">Deux</p>
```

Code CSS :

```
#p1 {position: absolute; z-index: 2;  
width: 150px; height: 140px;  
top: 100px; left: 200px;  
background-color: lime;}  
#p2 {position: absolute; z-index: 1;  
width: 225px; height: 100px;  
top: 50px; left: 25px;  
background-color: yellow;}
```

Résultat :



Positionnement flottant

- Propriété `float`
- L'élément se place à gauche (`float: left;`) ou à droite (`float: right;`) de son conteneur
- Les éléments qui le suivent se placent dans l'espace libre, en épousant sa forme
- Permet de placer des blocs côte à côte
- Applications : illustration dans un texte, lettrine
- Interdire le voisinage d'un élément flottant : propriété `clear` (pour les blocs)

`left` pas d'élément flottant à gauche

`right` pas d'élément flottant à droite

`both` aucun élément flottant au même niveau

Positionnement flottant : exemple

Code HTML :

```
<p class="rem">Attention, ceci est  
important !</p>  
<p>Voici la suite du texte qui s'enroule  
autour du premier paragraphe. C'est  
tout !</p>
```

Code CSS :

```
.rem {  
  background-color: red;  
  float: right;  
  width: 50%;  
}
```

Résultat :

Voici la suite
du texte qui
s'enroule
autour du
premier
paragraphe. C'est tout !

Attention, ceci
est important
!

La propriété clear : exemple

Code HTML :

```
<p class="rem">Attention, ceci est  
important !</p>  
<p>Para 1</p>  
<p id="p2">Para 2</p>
```

Code CSS :

```
.rem {  
    background-color: red;  
    float: right;  
    width: 50%;  
}  
#p2 {  
    background-color: blue;  
    clear: both;  
}
```

Résultat :

Para 1

Attention, ceci
est important
!

Para 2

Le positionnement inline-block

- **display: inline-block;**
- Les éléments ainsi positionnés se placent côte à côte (comme les éléments en ligne) et peuvent être redimensionnés (comme les blocs) avec les propriétés `width` et `height`
- Propriété `vertical-align` pour modifier l'alignement vertical des éléments :
 - `baseline`
 - `bottom`
 - `middle`
 - `top`
 - longueur (par rapport à la ligne de base)
 - pourcentage (par rapport à la hauteur de ligne; 0% = baseline)

Le positionnement inline-block : exemple

Exemple

Code HTML :

```
<div id="ib">  
  <p>Paragraphe</p>  
  <p>Un second paragraphe</p>  
</div>
```

Résultat :

Code CSS :

```
#ib p {  
  border: 1px solid black;  
  display: inline-block;  
  width: 100px;  
  vertical-align: top;  
}
```

Paragraphe

Un second
paragraphe