

Création et Manipulation de documents

(Hélène Renard / Sylvain Schmitz)

Travaux Dirigés – Séance n°7

---

## 1 Objectifs du TD

Utiliser les modèles des boîtes ainsi que le positionnement afin d'organiser une page XHTML.

## 2 Introduction

Bien qu'une boîte soit la partie fondamentale de la disposition d'éléments dans une page XHTML, elle suit quelques règles très simples. Nous allons au cours de ce TD répondre aux questions suivantes :

Comment les dimensions d'une boîte CSS sont-elles établies ? Quelles sont les diverses propriétés d'une boîte CSS ? Comment fonctionne le modèle des boîtes CSS ? Comment les boîtes interagissent-elles ensemble ?

Après avoir répondu à ces questions, nous verrons plus en détail le positionnement des différents éléments dans une page XHTML. Le positionnement CSS nous donnera toute liberté dans la mise en page. Comparée à l'utilisation des tableaux, cette technique offre des avantages essentiels.

## 3 Le modèle des boîtes dans une CSS

La CSS suppose qu'un document est affiché dans diverses boîtes. Vous pouvez avoir des boîtes dans une autre boîte, mais le bloc de construction de base d'un document est une boîte.

### 3.1 Dissection d'une boîte

Le bloc de construction de base est le contenu de la boîte. Cette boîte peut avoir optionnellement un bord *border* et un arrière plan *background*. Un bord est une ligne décorative autour de la boîte de contenu et l'arrière plan est simplement le « remplissage de fond » de la boîte.

Le contenu d'une boîte est au centre de la boîte. Autour de ce contenu, vous avez une marge inférieure *padding* qui est l'espace entre le bord et le contenu. Le bord est autour du *padding*, et autour du bord, vous avez la marge extérieure *margin*, qui est l'espace entre la boîte et d'autres boîtes autour d'elle (cf. figure 1).

### 3.2 Affichage d'une boîte

L'arrière plan est affiché seulement derrière le contenu et le *padding*, et les marges sont toujours transparentes. Donc, une boîte avec un arrière plan vert et un bord bleu sera affichée à la manière de la figure 2 (visualisez `~hrenard/TD7.pdf` pour les couleurs).

Les marges, le bord ou le *padding* peuvent avoir une valeur nulle ; dans ce cas, ils ne sont pas visibles. De plus, vous pouvez définir une largeur différente pour les marges, le bord ou le

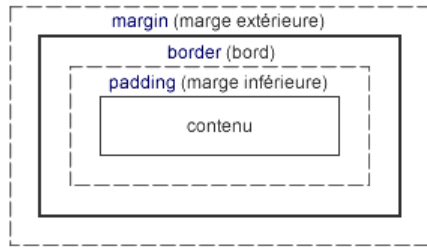


FIG. 1 – Le modèle des boîtes dans une CSS

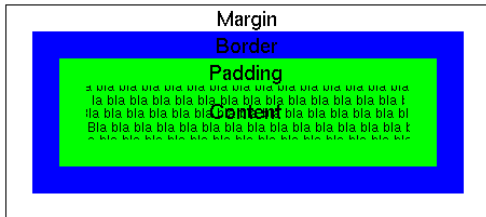


FIG. 2 – Affichage d'une boîte

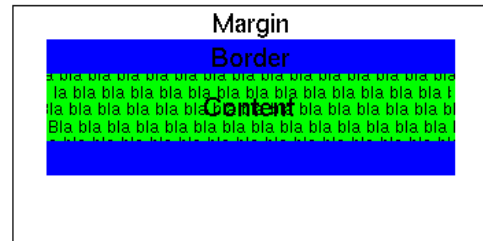


FIG. 3 – Affichage d'une autre boîte

*padding* en fonction des côtés (i.e. haut, bas, côté droit, côté gauche) de la boîte. Par exemple, la figure 3 nous montre pour :

- les bords
  - gauche / droit : largeur = 0
- le *padding*
  - tous les côtés : largeur = 0
- les marges
  - plus large en bas que sur les 3 autres côtés

Cependant dans la plupart des cas, comme le montre la figure 4, les marges et le *padding* ont beau ne pas être nuls, ils sont tout de même transparents.

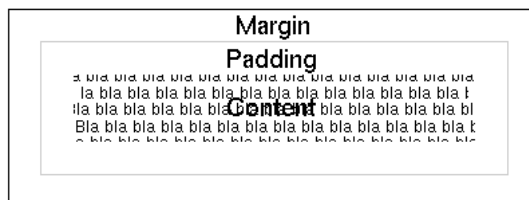


FIG. 4 – Marges et *padding* transparents

### 3.3 Exemple d'affichage d'une page XHTML

Pour afficher un document XHTML, un navigateur web va créer une boîte pour l'élément `body`. Le contenu de cette boîte est le bloc qui va contenir tous les éléments qui sont à l'intérieur de `body`. Ils sont placés les uns en dessous des autres. Si un de ces éléments a des éléments fils, ils sont alors placés dans le contenu de cet élément. Prenons un exemple pour illustrer cela en faisant quelques suppositions, histoire d'alléger un peu l'affichage de la figure 6. Supposons que :

1. les marges des deux premiers paragraphes sont égales
2. les marges de l'élément `div` et celles du second paragraphe sont elles aussi égales
3. l'élément `body` a un bord noir et un arrière plan blanc

4. l'élément `div` a un bord noir et un arrière rouge
5. les paragraphes fils de l'élément `div` ont un bord vert et un arrière jaune
6. le dernier paragraphe n'a pas de bord et il a un arrière plan transparent
7. les rayures noires représentent le contenu des paragraphes

```

<body>
<div>
  <p>
    Contenu du premier paragraphe.
  </p>
  <p>
    Contenu du second paragraphe.
  </p>
</div>
<p>
  Contenu du second paragraphe.
</p>
</body>

```

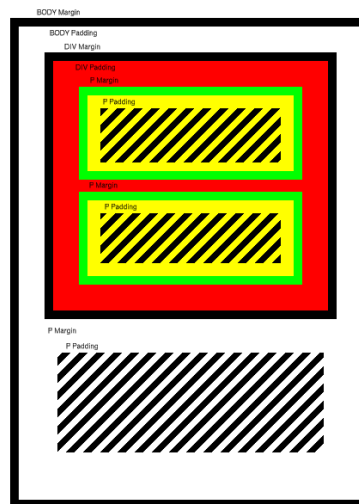


FIG. 5 – Code XHTML correspondant à la figure 6

FIG. 6 – Affichage du code XHTML de la figure 5 dans un navigateur web

### 3.4 Position d'une boîte

**margin-left, margin-right, margin-top, margin-bottom** : permet de définir un espace entre l'élément sélectionné et l'élément qui l'encadre. Les valeurs sont des nombres suivis d'une unité (par exemple `cm` pour le centimètre, `px` pour le pixel, `em` pour l'unité liée à la taille de la police de l'élément, ...). Dans l'exemple suivant, les marges de l'élément `p` sont placées à `1em` du haut de leur boîte, à 1 pouce du côté gauche, à 20% de la largeur de la boîte pour le côté droit et le bas.

```

p {
  margin-top: 1em;
  margin-left: 1in;
  margin-right: 20%;
  margin-bottom: 20%;
}

```

**margin** : est un raccourci pour toutes les propriétés listées précédemment.

**padding-top, padding-left, padding-bottom, padding-right** : permet de définir un espace entre le contenu de l'élément et sa bordure. Les valeurs acceptées sont les mêmes que pour l'ensemble **margin**.

**padding** : est un raccourci pour toutes les propriétés listées précédemment. Vous pouvez par exemple écrire :

```

h1 {
  /* 2 em sur tous les côtés */
  padding: 2em;
  /* 1 pouce en haut et en bas, 2 pouces sur les côtés */
  margin: 1in 2in;
  /* 3em en haut, 20% sur les côtés, et 2em en bas */
  padding: 3em 20% 2em;
}

```

```

/* 5 ex en haut, 2ex à droite,
   3ex bas et 3ex à gauche */
margin: 5ex 2ex 3ex 3ex;
}

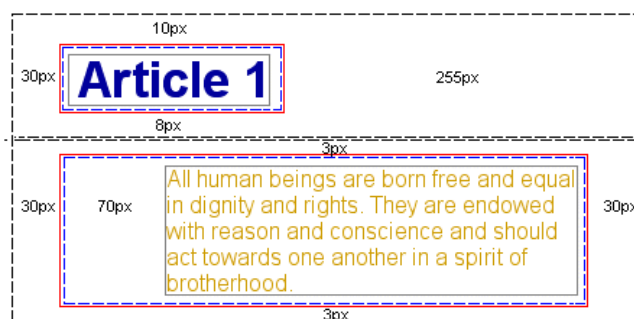
```

### Exercice n°1 :

1. Pour aérer un peu la lecture d'un document XHTML, on met en général un espace entre les bords de la fenêtre et le corps du document. Pour cela, reprenez vos fichiers `tp_css.xhtml` et `style.css` de la séance précédente, puis définissez une marge gauche, haute et droite pour le document de façon à ce qu'elle soit appliquée à tout le contenu de la page.
2. Les valeurs de type numérique peuvent être négatives. Faites en sorte que les titres de type `h2` soient un peu en retrait (vers la gauche) par rapport aux autres éléments.
3. Validez votre fichier `style.css`.

### Exercice n°2 :

1. Vous allez maintenant créer un fichier `droits.xhtml` ainsi qu'un fichier `css_droits.css` afin de reproduire le morceau de page suivante :



Attention! Pensez bien à mettre le squelette minimal d'une page XHTML.

2. Vous allez maintenant trouver à l'aide de votre moteur de recherche préféré, le texte complet dont est extrait le paragraphe que vous venez de mettre en forme.
3. Reprenez quelques paragraphes de ce texte dans `droits.xhtml`. Balisez votre texte et modifiez votre fichier `css_droits.css` afin de mettre en forme votre page Web (titres, arrière plan, paragraphe, police, texte, etc.)
4. Validez votre fichier `droits.xhtml` ainsi que votre fichier `css_droits.css`, puis insérez le(s) logo(s).

## 4 Le positionnement dans une CSS

Pour représenter le positionnement, on peut imaginer le navigateur parcourant (logiquement) la page de code XHTML du début à la fin et retranscrivant son contenu au fur et à mesure des balises rencontrées. Comme dans la lecture d'un texte, il procède verticalement, commençant en « haut » de l'écran pour aller jusqu'en « bas », et horizontalement de gauche à droite, sur la totalité de l'espace disponible et nécessaire en largeur comme en hauteur.

### 4.1 Deux types de boîtes

#### 1. Boîte de type « block »

Le texte contenu dans la boîte forme un paragraphe, et deux boîtes successives de ce type seront, par défaut, disposées l'une en dessous l'autre par les navigateurs. En d'autres termes, et en l'absence d'une indication de flottement, une boîte de type *block* commence

– et se termine – par un retour à la ligne. On peut changer la donne en modifiant le style grâce à la propriété `display` et une valeur `inline`.

```
<p class="jaune">Une boîte jaune</p>
<p class="verte">Une boîte verte</p>
```

FIG. 7 – Code dans le fichier XHTML

```
.jaune {
  background-color: #ffff00;
}
.verte {
  background-color: #00ff00;
}
```

FIG. 8 – Code dans le fichier CSS

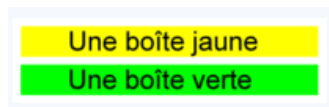


FIG. 9 – Le résultat pour le type de boîtes *block*

Les principaux éléments créant des boîtes *block* sont :

- l'élément `div`;
- les titres `h1`, `h2`, `h3`, `h4`, `h5`, `h6`;
- le paragraphe `p`;
- les listes et éléments de liste `ul`, `ol`, `li`, `dl`, `dd`;
- le bloc de citation `blockquote`;
- le texte pré-formaté `pre`;

## 2. Boîte de type « inline »

Le texte contenu dans la boîte s'affiche à la suite du texte précédent, et deux boîtes successives de ce type sont placées l'une à côté de l'autre par les navigateurs (s'ils disposent de suffisamment de place pour le faire). On peut aussi changer la donne en modifiant le style. Si vous souhaitez les afficher verticalement (les unes en dessous des autres), il faut utiliser la propriété `display` et la valeur `block` (`display :block`).

```
<p>
  <span class="jaune">Une boîte jaune</span>
  <span class="verte">Une boîte verte</span>
</p>
```

FIG. 10 – Code dans le fichier XHTML

```
.jaune {
  background-color: #ffff00;
}
.verte {
  background-color: #00ff00;
}
```

FIG. 11 – Code dans le fichier CSS

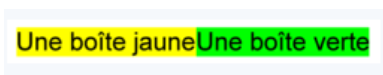


FIG. 12 – Le résultat pour le type de boîtes en ligne

Les principaux éléments créant des boîtes en ligne sont :

- l'élément `span`;
- le lien `a`;
- l'image `img`;
- les emphases `em` et `strong`;
- la citation `q`;
- la citation `cite`;
- l'élément `code`;

- l'exemple `samp` ;
- la variable `var` ;
- le texte inséré `ins` ;
- le texte supprimé `del`.

## 4.2 La position relative

La boîte en position relative est placée en flux normal (à la suite du contenu, comme un humain lit un texte) puis décalée horizontalement ou verticalement en fonction des propriétés de la CSS.

Le code est légèrement différent puisque la feuille de style s'occupe du positionnement grâce à la propriété `position`. Les propriétés `left` et `top` permettent respectivement d'indiquer la position par rapport à la gauche puis par rapport au haut de la boîte à positionner.

```
span{                                <p>
position:relative;                   <span>test 2</span>
top:15px;                             </p>
}
```

FIG. 13 – Code dans le fichier CSS

FIG. 14 – Code dans le fichier XHTML

Le contenu de la balise `<span>` sera rabaissé de 15 pixels.

## 4.3 La position absolue

Cette position permet de s'affranchir du flux normal et de positionner une boîte en fonction de propriétés de positionnement telles que `left`, `top`, `right` ou `bottom`.

Ceci s'avère très utile en particulier pour :

- placer les menus de navigation en fin de page, pour améliorer l'accessibilité de votre site en donnant un accès immédiat à son contenu dans les navigateurs textes, tout en les faisant apparaître en haut de page ou encore dans une colonne pour les navigateurs graphiques ;
- créer plusieurs colonnes au positionnement indépendant de l'ordre dans lequel elles se trouvent en XHTML.

```
p{                                    <p>
position:absolute;                   test de positionnement
top:15px;                             </p>
left:20px;
}
```

FIG. 15 – Code dans le fichier CSS

FIG. 16 – Code dans le fichier XHTML

Ce paragraphe sera placé à 15 pixels du haut et 20 pixels à gauche de votre document.

## 4.4 La position float

En fonction de la valeur de la propriété `float` (`left` ou `right`) on positionne à gauche ou à droite une boîte par rapport au reste de la page. Cette boîte n'est plus dans le flux normal, le contenu de la boîte suivante viendra se placer juste à droite (ou à gauche en fonction de l'inverse de la valeur de la propriété `float`).

Le navigateur place tout d'abord la boîte jaune, alignée sur le bord droit de cette partie de notre page, puis reprend le cours normal du flux dans l'espace laissé libre à sa gauche pour afficher la boîte verte. Le flux occupe tout l'espace disponible : la boîte verte reprend donc toute la largeur de la page sitôt passée la limite inférieure de la boîte flottante jaune.

```

.jaune {
  background-color: #ffff00;
  float: right;
  width: 100px;
  margin: 0;
}
.verte {
  background-color: #00ff00;
}

```

FIG. 17 – Code dans le fichier CSS

```

<p class="jaune">
  Une boîte jaune flottant
</p>
<p class="verte">
  Une boîte verte doté d'un contenu
  plus long ...
</p>

```

FIG. 18 – Code dans le fichier XHTML

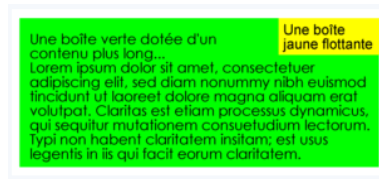


FIG. 19 – Placement de deux boîtes dont une en position float

Dans les navigateurs n'appliquant pas la feuille de style (navigateurs texte ou non conformes aux standards ...), la boîte flottante sera simplement inscrite en flux normal avant la boîte qui la suit.

### Exercice n°3 :

1. La combinaison de la position flottante et du flux permet de réaliser aisément une mise en page à deux colonnes (par exemple un contenu et un menu). Vous allez donc créer une page ayant comme modèle la figure 20 et ayant comme liens « A. Rimbaud », « P. Verlaine » et « C. Baudelaire ». Votre page XHTML aura le nom `intro_poesie.xhtml` et la feuille de style le nom `style_poesie.css`. Il ne vous est pas demandé de faire une page pour chaque auteur, juste la page d'accueil basique d'un site autour des 3 auteurs.
2. Vous pouvez tout aussi bien multiplier les flottants pour créer autant de colonnes que voulu. Vous allez donc maintenant créer une nouvelle page d'accueil basique sous le nom `poesie.xhtml` avec sa feuille de style sous le modèle de la figure 21. Vous mettrez dans chaque colonne une courte biographie de chaque auteur vu à la question 1.
3. Recommencez les questions 1 et 2 en utilisant cette fois la position absolue (vous êtes donc amenés à placer vous mêmes les boîtes où vous le souhaitez).

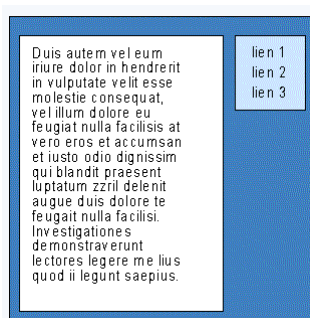


FIG. 20 – Un modèle de page XHTML à 2 colonnes

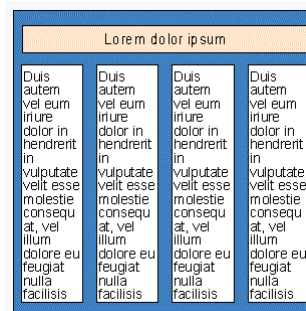


FIG. 21 – Un modèle de page XHTML à 4 colonnes

Source Modèles des boîtes : <http://www.webreference.com/html/tutorial9/1.html>

Source Positionnement : [http://openweb.eu.org/articles/initiation\\_flux/](http://openweb.eu.org/articles/initiation_flux/)