

CSS : les modèles de boîtes

Contenu

Introduction.....	2
Le modèle standard.....	3
La boîte de contenu (content).....	3
La boîte de remplissage.....	3
La bordure	4
La marge extérieure	4
Les différentes interactions.....	6
Les dépassements	6
L'arrière-plan	7
Le contour.....	8
Les types de boîte.....	9
Le modèle de boîte flexible : Flexbox.....	11
Présentation	11
Exemple	11
Exercices	12

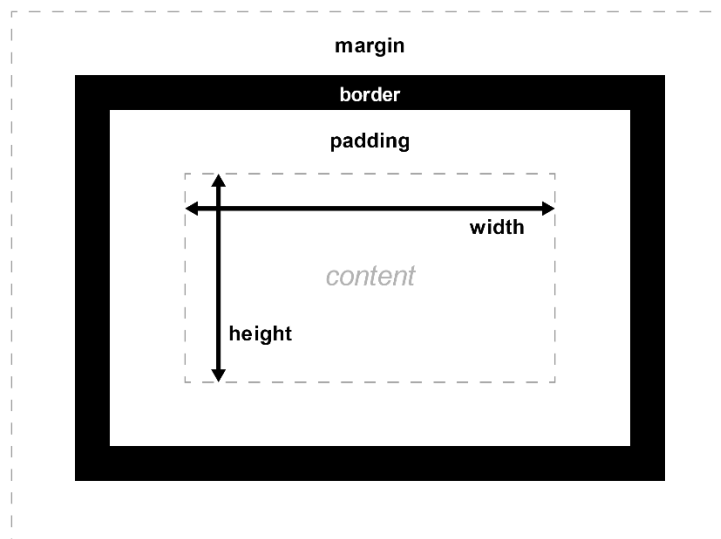
Introduction

Dans un document HTML, chaque élément est représenté par une boîte rectangulaire. Le calcul de la taille, des propriétés (couleur, arrière-plan, forme du cadre) et de la position des boîtes est le rôle du moteur de rendu. En CSS, chacune de ces boîtes est décrite à partir d'un modèle standard qui permet de décrire le contenu de l'espace occupé par un élément.

Le modèle de boîte CSS est à l'origine de la disposition des contenus d'un document. Avant d'aborder cette disposition, commençons par décrire le fonctionnement des boîtes.

Le modèle standard

Chaque élément d'un document est matérialisé par une boîte qui peut être ajustée grâce à des propriétés CSS spécifiques. Ces propriétés peuvent être représentées ainsi :



La boîte de contenu (content)

La boîte de contenu est l'endroit où le contenu de l'élément est affiché. Ce contenu est le texte de l'élément ou les autres boîtes des éléments fils. La taille de la boîte de contenu est définie avec les propriétés CSS *width* (pour la largeur) et *height* (pour la hauteur).

Il existe d'autres propriétés qui permettent de gérer la taille de la boîte de contenu en fonction de contrainte (plutôt que d'exprimer une taille absolue). On peut donc aussi utiliser les propriétés : *min-width*/*max-width* (pour limiter les valeurs de largeur) et *min-height*/*max-height* (pour limiter les valeurs de hauteur).

La boîte de remplissage

Le remplissage d'une boîte est une zone, visible ou non, qui représente la limite de la boîte. Par défaut, la taille de la bordure vaut 0, ce qui la rend invisible. La bordure est définie grâce à la propriété CSS *border*. Cette propriété est aussi une propriété raccourcie composée de *border-size* (l'épaisseur de la bordure), *border-style* (le style à appliquer pour la ligne de la bordure, généralement *solid*) et *border-color* (la couleur). Il est également possible de définir chaque côté séparément et il existe donc un ensemble de propriétés *border-** pour chacun des côtés (*top* pour le haut, *right* pour la droite, *bottom* pour le bas et *left* pour la gauche).

La bordure

La bordure d'une boîte est une zone, visible ou non, qui représente la limite de la boîte. Par défaut, la taille de la bordure vaut 0, ce qui la rend invisible. La bordure est définie grâce à la propriété CSS `border`. Cette propriété est aussi une propriété raccourcie composée de `border-size` (l'épaisseur de la bordure), `border-style` (le style à appliquer pour la ligne de la bordure, généralement `solid`) et `border-color` (la couleur). Il est également possible de définir chaque côté séparément et il existe donc un ensemble de propriétés `border-*` pour chacun des côtés (`top` pour le haut, `right` pour la droite, `bottom` pour le bas et `left` pour la gauche).

La marge extérieure

La marge (extérieure) représente la distance qui doit séparer deux boîtes. La marge similaire au remplissage mais se situe à l'extérieur de la bordure. La propriété raccourcie est `margin` et les propriétés individuelles sont `margin-top`, `margin-right`, `margin-bottom` et `margin-left`.

La plupart du temps, les marges s'encastrent (`margin collapsing`). Autrement dit, quand deux boîtes se suivent, la distance entre elles n'est pas la somme des deux marges adjacentes mais la marge la plus grande.

Prenons un exemple pour illustrer ces différentes quantités :

HTML (c'est ce code HTML qui sera utilisé pour tous les exemples) :

```
<p>Je suis un paragraphe</p>
<p>Je suis un autre paragraphe</p>
```

La feuille de style CSS :

```
/* Pour tous les paragraphes */
p {
  /* Ils auront exactement une largeur de 100px */
  width: 100px;

  /* Leur hauteur est définie automatiquement
   en fonction de la taille du contenu */
  height: auto;

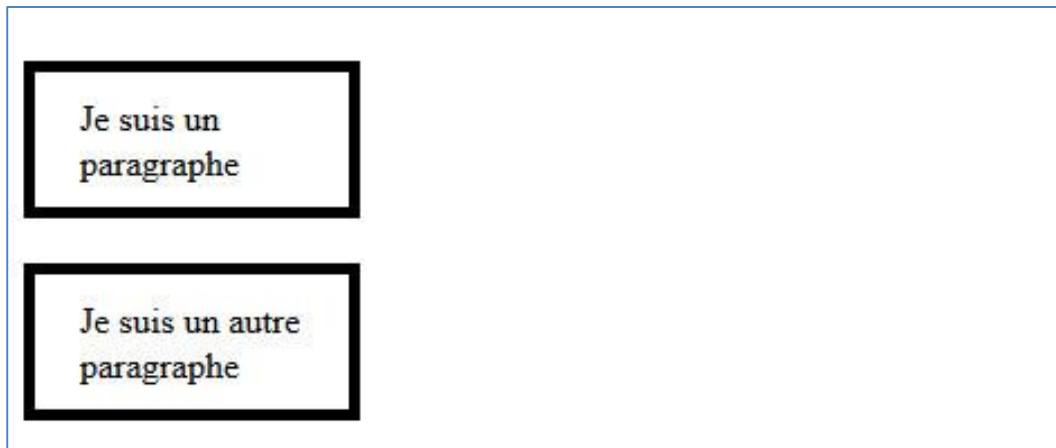
  /* La distance entre le contenu et la bordure
   est 20px à droite et à gauche et est
   10px en haut et en bas */
  padding: 10px 20px 10px 20px;

  /* On définit une bordure noire, large de 5px */
  border: 5px solid black;

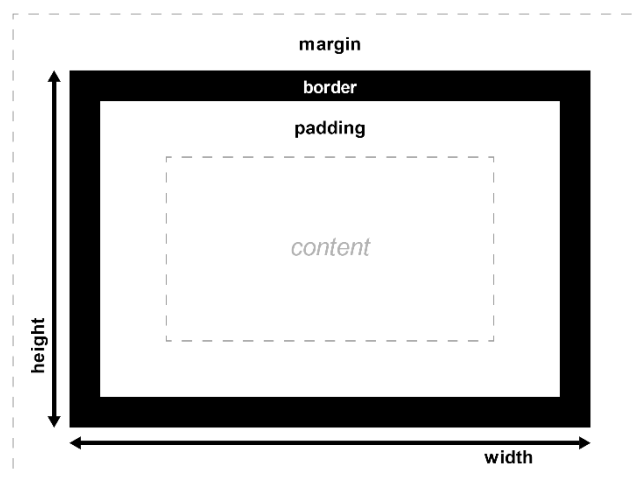
  /* La distance minimale entre un paragraphe
```

```
et une autre boîte au-dessus ou en-dessous  
sera de 20px */  
margin: 20px 0 20px 0;  
}
```

Ce document HTML, associé à cette feuille de style donnera le résultat suivant :



Vous avez pu voir que la taille réelle d'une boîte était en fait la somme des propriétés `width`, `padding-right`, `padding-left`, `border-right` et `border-left`. Dans certains cas, cela peut gêner (par exemple, comment faire si on souhaite avoir une boîte dont la largeur totale vaut 50% et dont les bordures et le remplissage sont exprimés en pixels). Pour éviter de tels problèmes, il est possible d'ajuster le modèle de boîte grâce à la propriété `box-sizing` et à la valeur `border-box`. Cela permettra d'utiliser ce nouveau modèle de boîte :



Les différentes interactions

En CSS, gérer des boîtes ne se limite pas qu'à définir leur taille. Il s'agit également qu'elles soient réactives.

Les dépassements

Lorsqu'on définit la taille d'une boîte en utilisant des valeurs absolues, il peut arriver que le contenu ne tienne pas dans l'espace alloué. Dans ces cas, le contenu dépasse de la boîte. Pour décider ce qui se produit alors, on peut utiliser la propriété *overflow*. Celle-ci peut prendre différentes valeurs mais les valeurs qu'on rencontre le plus souvent sont :

auto	s'il y a trop de contenu, celui-ci est rogné et des ascenseurs (horizontaux et/ou verticaux) sont affichés pour que l'utilisateur puisse parcourir le contenu.
hidden	s'il y a trop de contenu, la partie qui dépasse est rognée et n'est pas affichée.
visible	s'il y a trop de contenu, la partie qui dépasse est affichée en dehors de la boîte (c'est généralement le comportement par défaut).

Voici un exemple simple pour illustrer ce concept.

Ci-dessous le code HTML qu'on utilisera :

```
<p class="autoscroll">
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  Mauris tempus turpis id ante mollis dignissim. Nam sed
  dolor non tortor lacinia lobortis id dapibus nunc. Praesent
  iaculis tincidunt augue. Integer efficitur sem eget risus
  cursus, ornare venenatis augue hendrerit. Praesent non elit
  metus. Morbi vel sodales ligula.
</p>

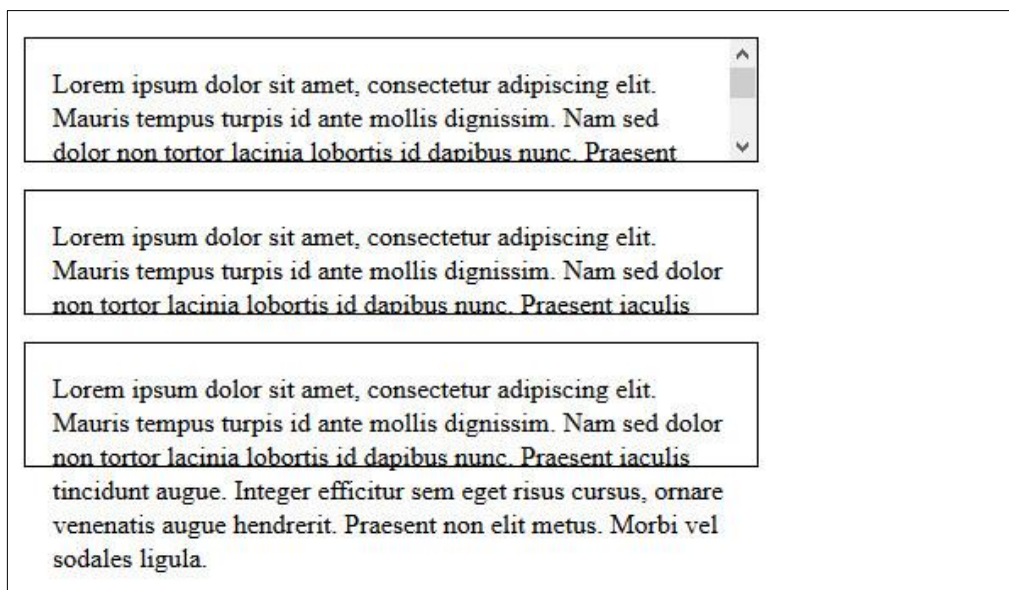
<p class="clipped">
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  Mauris tempus turpis id ante mollis dignissim. Nam sed
  dolor non tortor lacinia lobortis id dapibus nunc. Praesent
  iaculis tincidunt augue. Integer efficitur sem eget risus
  cursus, ornare venenatis augue hendrerit. Praesent non elit
  metus. Morbi vel sodales ligula.
</p>

<p class="default">
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  Mauris tempus turpis id ante mollis dignissim. Nam sed
  dolor non tortor lacinia lobortis id dapibus nunc. Praesent
  iaculis tincidunt augue. Integer efficitur sem eget risus
  cursus, ornare venenatis augue hendrerit. Praesent non elit
  metus. Morbi vel sodales ligula.
</p>
```

Et la feuille de style :

```
p {  
  width : 400px;  
  height : 2.5em;  
  padding: 1em 1em 1em 1em;  
  border : 1px solid black;  
}  
  
.autoscroll { overflow: auto; }  
.clipped { overflow: hidden; }  
.default { overflow: visible; }
```

On obtient alors :



L'arrière-plan

En modifiant l'arrière-plan on peut améliorer l'apparence des boîtes. La gestion des arrière-plans est un sujet à part entière et nous n'aborderons ici que l'interaction entre les boîtes et les arrière-plans. Les arrière-plans peuvent être des couleurs ou des images qui seront appliquées et dessinées dans chaque boîte. Par défaut, l'arrière-plan est dessiné en dessous du contenu, du remplissage et des bordures. Cela sera peu important lorsqu'on utilise une couleur opaque mais pourra être utile à gérer lorsqu'on aura une image (pour éviter certains effets visuels indésirables).

Parmi les différentes propriétés qui permettent de gérer l'arrière-plan, une permet d'indiquer l'endroit à partir duquel dessiner l'arrière-plan : *background-clip*.

Un exemple vaut mieux qu'un long discours.

Le document HTML :

```
<div class="default"></div>
<div class="padding-box"></div>
<div class="content-box"></div>
```

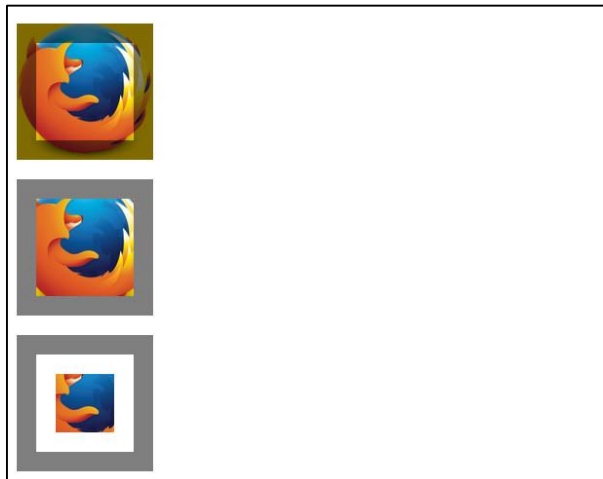
La feuille de style CSS :

```
div {
    width : 60px;
    height : 60px;
    border : 20px solid rgba(0, 0, 0, 0.5);
    padding: 20px;
    margin : 20px 0;

    background-size      : 140px;
    background-position: center;
    background-image     : url('https://mdn.mozillademos.org/files/11947/ff-
logo.png');
    background-color     : gold;
}

.default      { background-clip: border-box; }
.padding-box { background-clip: padding-box; }
.content-box  { background-clip: content-box; }
```

Résultat :



Le contour

Enfin, *outline* représente le contour d'une boîte. Le contour se rapproche en quelque sorte d'une bordure mais ne fait pas partie du modèle de boîte. Le contour est dessiné autour de la boîte dans la zone dédiée à la marge (après la bordure) mais n'est pas pris en compte dans le calcul de la taille des boîtes.

Attention à l'utilisation de la propriété *outline*, en effet le contour est parfois utilisé pour l'accessibilité afin d'identifier la zone active de la page pour l'utilisateur (par exemple le lien qui est actif lorsqu'il a été sélectionné au clavier).

Les types de boîte

Tout ce que nous avons vu jusqu'à présent s'applique aux boîtes qui sont des blocs. Cependant, il existe d'autres types de boîte en CSS et ceux-ci se comportent différemment. Le type de boîte qui s'applique pour un élément est déterminé par la propriété *display*. CSS fournit de nombreuses valeurs pour cette propriété. Dans cet article, nous nous concentrerons avant tout sur les trois valeurs basiques que sont *block*, *inline*, et *inline-block*.

block	Une boîte block (ou bloc) est définie comme une boîte qui va s'empiler sur d'autres boîtes (autrement dit, on crée une cassure de la ligne visuelle avant et après la boîte). Le modèle de boîte vu juste avant s'applique.
inline	Une boîte inline (ou en ligne) est l'opposée d'une boîte bloc : elle flotte dans le texte du document (autrement dit, elle ne crée pas de cassure de ligne visuelle avant et après la boîte et le contenu de cette boîte passera à la ligne selon le contenu qui l'entoure). Les propriétés du modèle de boîte s'appliquent mais n'impactent pas les boîtes environnantes.
inline-block	Une boîte inline-block (en bloc et en ligne) est quelque chose d'intermédiaire : le contenu flotte dans le document sans créer de cassure de ligne avant et après. Le contenu reste à l'intérieur de la boîte et la boîte ne suit pas les règles de saut à la ligne dicté par le contenu en dehors de la boîte. Enfin les propriétés du modèle de boîte s'appliquent à ces boîtes et cela impacte les boîtes environnantes.

Cela paraît compliqué ? Prenons quelques exemples.

Nous utiliserons ce document HTML :

```
<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  <span class="inline">Mauris tempus turpis id ante mollis dignis-
sim.</span> Nam sed dolor non tortor lacinia lobortis id dapibus nunc.
</p>

<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  <span class="block">Mauris tempus turpis id ante mollis dignissim.</span>
  Nam sed dolor non tortor lacinia lobortis id dapibus nunc.
</p>

<p>
  Lorem ipsum dolor sit amet, consectetur adipiscing elit.
  <span class="inline-block">Mauris tempus turpis id ante mollis dignis-
sim.</span>
  Nam sed dolor non tortor lacinia lobortis id dapibus nunc.
```

</p>

Et cette feuille de style CSS :

```
p {
  padding : 1em;
  border : 1px solid black;
}

span {
  padding : 0.5em;
  border : 1px solid green;

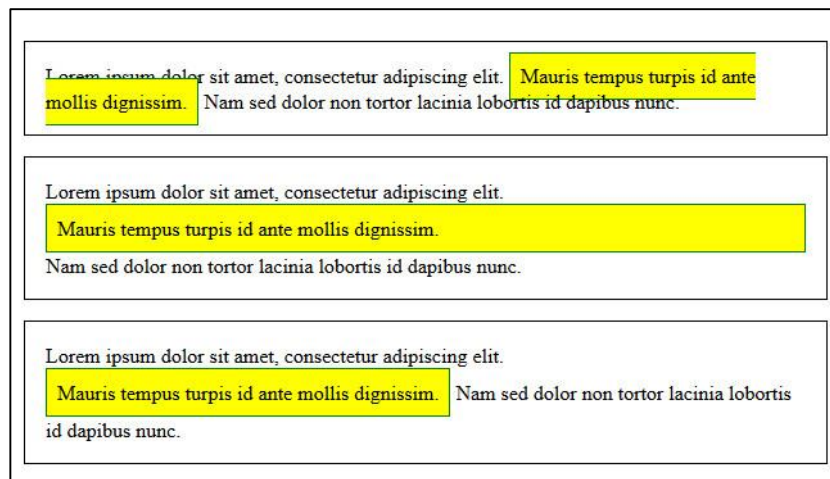
  /* On rend la boîte visible, quel que soit son type */
  background-color: yellow;
}

.inline {
  display: inline;
}

.block {
  display: block;
}

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

On obtiendra alors le résultat suivant :



Le modèle de boîte flexible : Flexbox

Présentation

CSS 3 a introduit un nouveau modèle de boîte, FlexBox (pour Flexible Box) afin de répondre aux besoins du design responsive.

Ce modèle est différent du modèle de boîte standard en ce qu'il fournit une façon efficace de disposer, aligner et distribuer l'espace entre les items d'un container, même lorsque leurs dimensions sont inconnues et/ou dynamiques (redimensionnables).

Les éléments enfants d'une flexbox peuvent être placés dans n'importe quelle direction et peuvent avoir des dimensions pour s'adapter à la place disponible. Positionner les éléments enfants est ainsi plus simple et les agencements complexes peuvent être mis en place plus simplement et avec un code plus propre, étant donné que l'ordre d'affichage des éléments est indépendant de leur ordre dans le code source.

Dans le modèle de boîte Flexbox, il n'y a pas de notions de fusion de marges entre les éléments ni de positionnement flottant. Les boîtes sont positionnées selon les axes horizontaux et verticaux et les points de jonction des boîtes (start-end, s

Exemple

Pour déclarer un élément en modèle flex, on lui attribue la valeur `flex` ou `flex-inline`.

```
#div-flex {  
  display: flex;  
}
```

Les éléments enfants seront alors directement considérés comme des boîtes flexbox (ils deviennent des « flex-items »).

👉 [En savoir plus sur Flexbox](#)

Exercices

Dans une page web, créer plusieurs éléments en boîte standard et en modifier les différentes propriétés : dimensions, marge, padding, bordure.