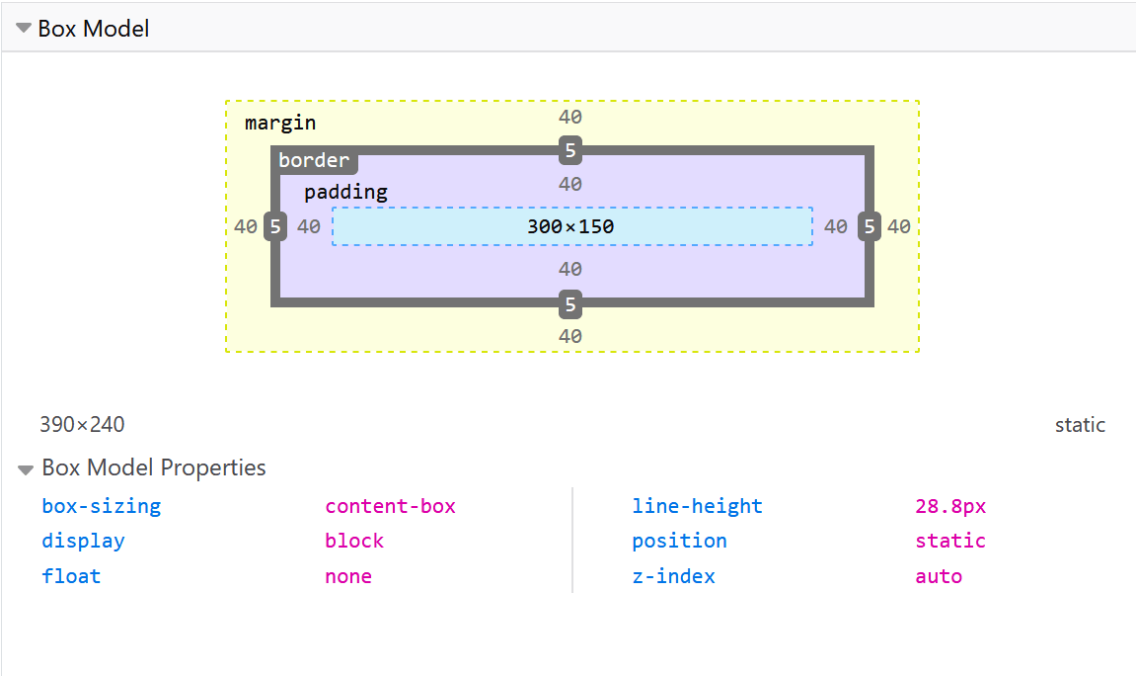
Architecture et langages Web
(NFA040)

CSS : modèle de boîte (tutoriel)

Remarque. Ce tutoriel est extrait de la section de MDN intitulée : Qu'est-ce que le modèle de boîte CSS ?

Utiliser les outils de développement pour voir le modèle de boîte

Les outils de développement de votre navigateur peuvent vous permettre d'appréhender les concepts de boîte bien plus facilement. Si vous inspectez un élément (clic droit > Examiner l'élément), vous pouvez avoir accès à toutes les propriétés des différentes couches de la boîte (contenu, *padding*, bordure et marge) dans l'interface graphique interactive montrée ci-dessous. Inspecter un élément ainsi, c'est s'assurer qu'il possède bien la taille que l'on désire !



Marges, remplissages (paddings), et bordures

Nous avons déjà rencontré ensemble les propriétés `margin`, `padding` et `border`, ainsi que leurs effets dans les exemples précédents. Mais ces propriétés sont des **raccourcis** qui nous permettent de définir ces règles pour les quatre côtés de la boîte d'un seul coup. Ces raccourcis ont donc aussi leurs propriétés équivalentes permettant de régler séparément chaque côté pour plus de personalisation.

Regardons de plus près ces nouvelles propriétés.

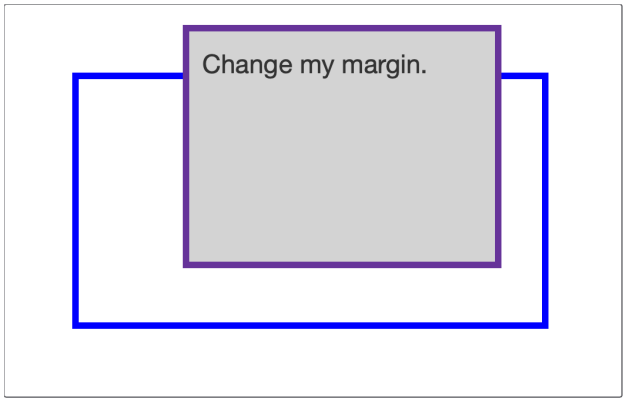
1 Les marges

La marge est une zone d'espacement invisible qui encadre votre boîte (une marge extérieure). La marge repousse les éléments alentours de la boîte. On peut de plus lui donner une valeur numérique positive ou bien même négative ! Lorsque cette valeur est négative, cela peut cependant engendrer des superpositions entre votre boîte et d'autres éléments. Que vous utilisiez le modèle alternatif ou standard, la marge est toujours décomptée en surplus de la taille totale de la boîte et est ajoutée après que celle-ci a été calculée.

On peut fixer les quatre marges d'une boîte d'un seul coup à l'aide de la propriété `margin`, ou bien régler chaque côté individuellement avec les propriétés équivalentes suivantes :

- `margin-top`
- `margin-right`
- `margin-bottom`
- `margin-left`

Exercice. Dans l'exemple ci-dessous, tentez donc de modifier les valeurs de `margin` pour voir comment la boîte est repoussée et évolue à cause des espaces créés ou supprimés (si la marge est négative) par vos soins.



Fichier CSS et body HTML

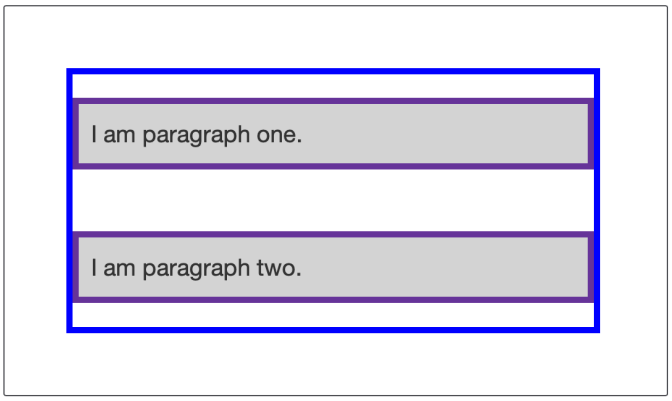
```
.box {
  border: 5px solid rebeccapurple;
  background-color: lightgray;
  margin-top: -40px;
  margin-right: 30px;
  margin-bottom: 40px;
  margin-left: 4em;
  height: 100px;
}

.container {
  border: 5px solid blue;
  margin: 100px;
}

<div class="container">
  <div class="box">Change my margin.</div>
</div>
```

2 La fusion des marges

Le concept de fusion entre les marges est important à maîtriser pour la mise en page. Si deux éléments de votre page ont des marges qui se touchent, alors ces marges fusionnent pour ne faire plus qu'une seule marge qui aura pour taille la plus grande des deux tailles des marges initiaux.

Dans l'exemple ci-dessous, nous avons deux paragraphes. Le paragraphe du haut a un `margin-bottom` de 50 pixels. Le second paragraphe a un `margin-top` de 30 pixels. Puisque ces deux marges se touchent, elles fusionnent ensemble, et ainsi la marge finale entre les deux paragraphes est de 50 pixels et non 80, la somme des deux marges.

Exercice. Vous pouvez tester cette propriété par vous-même en modifiant la propriété `margin-top` du deuxième paragraphe à 0 dans l'exemple ci-dessous. La marge visible entre les deux paragraphes demeure inchangée — elle conserve sa taille de 50 pixels qui provient de la propriété `margin-bottom` du premier paragraphe.



Fichier CSS et body HTML

```
.box {
  border: 5px solid rebeccapurple;
  background-color: lightgray;
}

.container {
  border: 5px solid blue;
  margin: 200px;
}

.one {
  margin-bottom: 50px;
}

.two {
  margin-top: 30px;
}
```

```
<div class="container">
  <p class="box one">I am paragraph one.</p>
  <p class="box two">I am paragraph two.</p>
</div>
```

Il existe quelques règles qui contrôlent la fusion ou non des marges. Pour plus d'informations, référez vous à la page détaillée Maîtriser la fusion des marges. Si vous ne devez retenir qu'une chose, c'est que les marges peuvent fusionner, et que si vos marges ne correspondent pas à vos attentes, c'est certainement ce phénomène qui est derrière.

3 Les bordures

La bordure se situe entre la marge et le remplissage (*padding*) d'une boîte. Si vous utilisez le modèle standard de boîte, la taille de la bordure s'ajoute à la largeur (`width`) et la hauteur (`height`) de la boîte. Si vous utilisez le modèle de boîte alternatif, alors la taille de la bordure rend la taille disponible pour le contenu plus petite puisqu'elle utilise une partie de la largeur et de la hauteur disponible.

Pour agir sur le style d'une bordure, il existe de nombreuses propriétés qui permettent de régler le style, la taille et la couleur pour chacun des quatre côtés de la bordure.

Vous pouvez naturellement fixer la forme taille et couleur des quatre côtés en une seule fois, par le biais de la propriété `border`.

Pour régler ces propriétés individuellement pour chacun des côtés, vous pouvez utiliser :

- `border-top`
- `border-right`
- `border-bottom`
- `border-left`

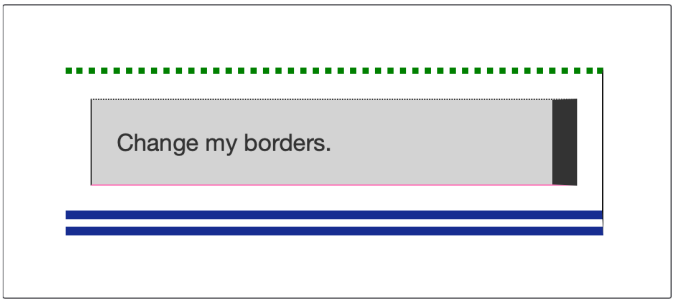
Pour modifier la taille, le style ou la couleur de tous les côtés en même temps, utilisez les propriétés suivantes :

- `border-width`
- `border-style`
- `border-color`

Pour modifier la taille, le style ou la couleur d'un seul coté à la fois, vous pouvez faire l'usage de ces propriétés :

- `border-top-width`
- `border-top-style`
- `border-top-color`
- `border-right-width`
- `border-right-style`
- `border-right-color`
- `border-bottom-width`
- `border-bottom-style`
- `border-bottom-color`
- `border-left-width`
- `border-left-style`
- `border-left-color`

Exercice. Dans l'exemple ci-dessous, nous avons utilisé différentes propriétés, qu'elles soient des raccourcis ou bien les propriétés précises, pour créer une bordure. Amusez-vous à modifier les valeurs de ces différentes propriétés pour vérifier que vous comprenez bien comment elles s'organisent. Les pages MDN pour les propriétés des bordures (données ci-dessus) documentent les différents styles que vous pouvez appliquer à vos pages. N'hésitez pas à les consulter.



Fichier CSS et body HTML

```
.container {
  border-top: 5px dotted green;
  border-right: 1px solid black;
  border-bottom: 20px double rgb(23,45,145);
}

.box {
  border: 1px solid #333333;
  border-top-style: dotted;
  border-right-width: 20px;
  border-bottom-color: hotpink;
  margin: 20px;
  background-color: lightgray;
}
```

```
<div class="container">
  <div class="box">Change my borders.</div>
</div>
```

4 Le padding (remplissage)

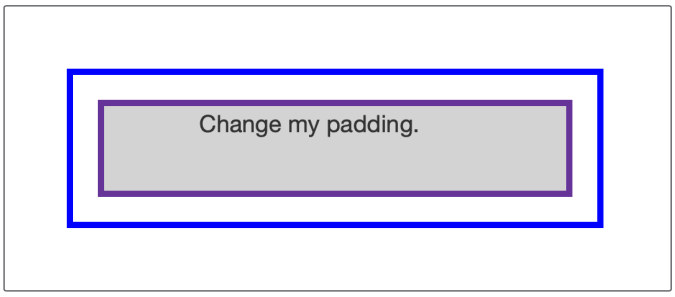
Le *padding* (ou remplissage) se situe entre la bordure et le contenu. Contrairement aux marges, on ne peut attribuer une valeur numérique négative à un *padding*, la valeur ne peut être que 0 ou bien une valeur positive. Si vous avez défini un arrière-plan à votre élément, celui-ci continuera de s'afficher dans la *padding*, et c'est pourquoi cette propriété est souvent utilisée pour repousser le contenu de la bordure.

On peut une fois de plus configurer le *padding* pour tous les côtés à la fois à l'aide de la propriété `padding`, ou bien chaque côté indépendamment des autres en utilisant les variantes plus précises suivantes :

- `padding-top`
- `padding-right`
- `padding-bottom`
- `padding-left`

Exercice. Si vous modifiez les valeurs du *padding* sur la classe `.box` de l'exemple ci-dessous, vous pouvez observer comment l'emplacement du texte est impacté par les marges intérieures.

Tentez aussi de modifier la valeur du *padding* dans la classe `.container`, cela aura pour effet d'espacer le conteneur et la boîte. Le *padding* peut être modifié sur tout élément pour permettre d'espacer le contenu de la bordure.



Fichier CSS et body HTML

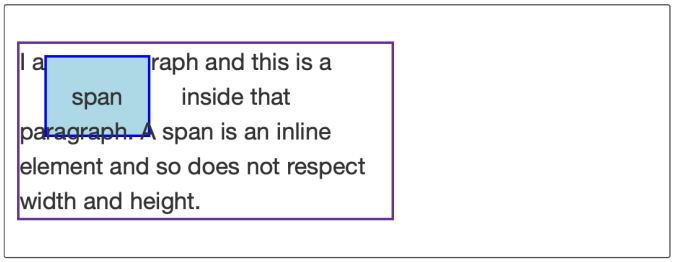
```
.box {
  border: 5px solid rebeccapurple;
  background-color: lightgray;
  margin: 20px;
  padding-top: 0;
  padding-right: 30px;
  padding-bottom: 40px;
  padding-left: 4em;
}

.container {
  border: 5px solid blue;
  margin: 200px;
}
```

```
<div class="container">
  <div class="box">Change my padding.</div>
</div>
```

5 Le modèle de boîte et la disposition en ligne

Toutes les règles énoncées plus haut s'appliquent totalement aux boîtes positionnées en bloc. Mais qu'en est-il des boîtes positionnées en ligne, comme l'élément `<span>` par exemple ?

Dans l'exemple ci-après, nous avons un élément `<span>` inclus dans un paragraphe auquel on a défini les propriétés `width`, `height`, `margin`, `border` et `padding`. Vous pouvez alors observer que les paramètres `width` et `height` sont totalement ignorés. Les propriétés de `margin`, `padding` et `border` sont quant à elles appliquées, mais n'ont pas modifié l'espacement avec les autres éléments de la page, se superposant ainsi avec les mots environnants dans le paragraphe.



Fichier CSS et body HTML

```
.container {
  border: 1px solid rebeccapurple;
  margin: 100px;
  width: 15em;
}

span {
  margin: 20px;
  padding: 20px;
  width: 80px;
  height: 50px;
  background-color: lightblue;
  border: 2px solid blue;
}
```

```
<div class="container">
  <p>
    I am a paragraph and this is a <span>span</span> inside that paragraph.
    A span is an inline element and so does not respect width and height.
  </p>
</div>
```

6 Le positionnement display: inline-block

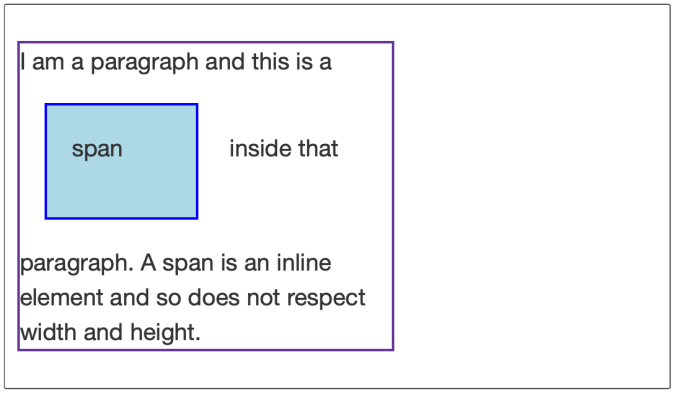
Il existe une valeur spéciale pour la propriété `display`, qui constitue un compromis entre la disposition en ligne et la disposition en bloc, une sorte d'entre-deux qui combine ces deux dispositions. Cet état peut-être utile dans les situations où l'on désire utiliser les propriétés `width` et `height`, et éviter les superpositions (voir l'exemple précédent), tout en conservant la disposition dans une même ligne (i.e. sans créer de nouvelle ligne, comme le ferait une disposition en bloc).

C'est la solution apportée par la disposition `display: inline-block;` qui emprunte des règles des deux dispositions pour satisfaire ces motivations :

- La hauteur (`height`) et la largeur (`width`) seront appliqués sur l'élément (et non ignorés).
- Les propriétés `padding`, `margin` et `border` repousseront bien les éléments alentours.

Cette disposition suit alors ces règles, tout en conservant un positionnement sur la même ligne, sans retour à la ligne, ni affichage sur sa propre nouvelle ligne. L'élément peut même devenir plus grand que son conteneur si les propriétés `width` et `height` le définissent ainsi.

Exercice. Dans cet exemple, nous avons ajouté la propriété `display: inline-block;` à notre élément `<span>`. Changez donc la valeur en `display: block;` ou bien tentez même de supprimer cette ligne pour observer l'utilité de cette nouvelle disposition.



Fichier CSS et body HTML

```
.container {
  border: 1px solid rebeccapurple;
  margin: 100px;
  width: 15em;
}

span {
  margin: 20px;
  padding: 20px;
  width: 80px;
  height: 50px;
  background-color: lightblue;
  border: 2px solid blue;
  display: inline-block;
}

<p>
    I am a paragraph and this is a <span>span</span> inside that paragraph. A
span is an inline element and so does not respect width and height.
</p>
```

Ceci peut-être très utile dans certains cas comme lorsque l'on veut élargir la zone cliquable d'un lien en aggrandissant le `padding`. l'élément `<a>` est par défaut « en ligne », comme un `<span>`, mais vous pouvez alors utiliser `display: inline-block;` pour permettre au *padding* d'être ajouté correctement sur la page, améliorant l'accessibilité du lien pour l'utilisateur.

7 Application : barre de navigation

Vous pouvez rencontrer cette astuce sur bon nombre de menus de navigation dans les sites web. Par exemple, la barre de navigation ci-dessous est affichée en une seule ligne en utilisant une table et nous avons ajouté un *padding* aux liens `<a>`pour pouvoir modifier la couleur de fond (`background-color`) au survol du curseur. Le *padding* semble se superposer sur la bordure de l'élément `<td>`. Ceci est dû au fait que `<a>` est un élément en ligne.

Exercice. Ajoutez la propriété `display: inline-block;` en utilisant le sélecteur `.links-list a` pour voir le respect du *padding* régler ce problème.

Fichier CSS et body HTML

```
.links-list a {
  background-color: rgb(179, 57, 81);
  color: #fff;
  text-decoration: none;
  padding: 1em 2em;
}

.links-list a:hover {
  background-color: rgb(66, 28, 40);
  color: #fff;
}
```

```
table {
  border: 1px solid rebeccapurple;
}

<nav>
  <table>
    <tr class="links-list">
      <td><a href="">Link one</a></td>
      <td><a href="">Link two</a></td>
      <td><a href="">Link three</a></td>
    </tr>
  </table>
</nav>
```