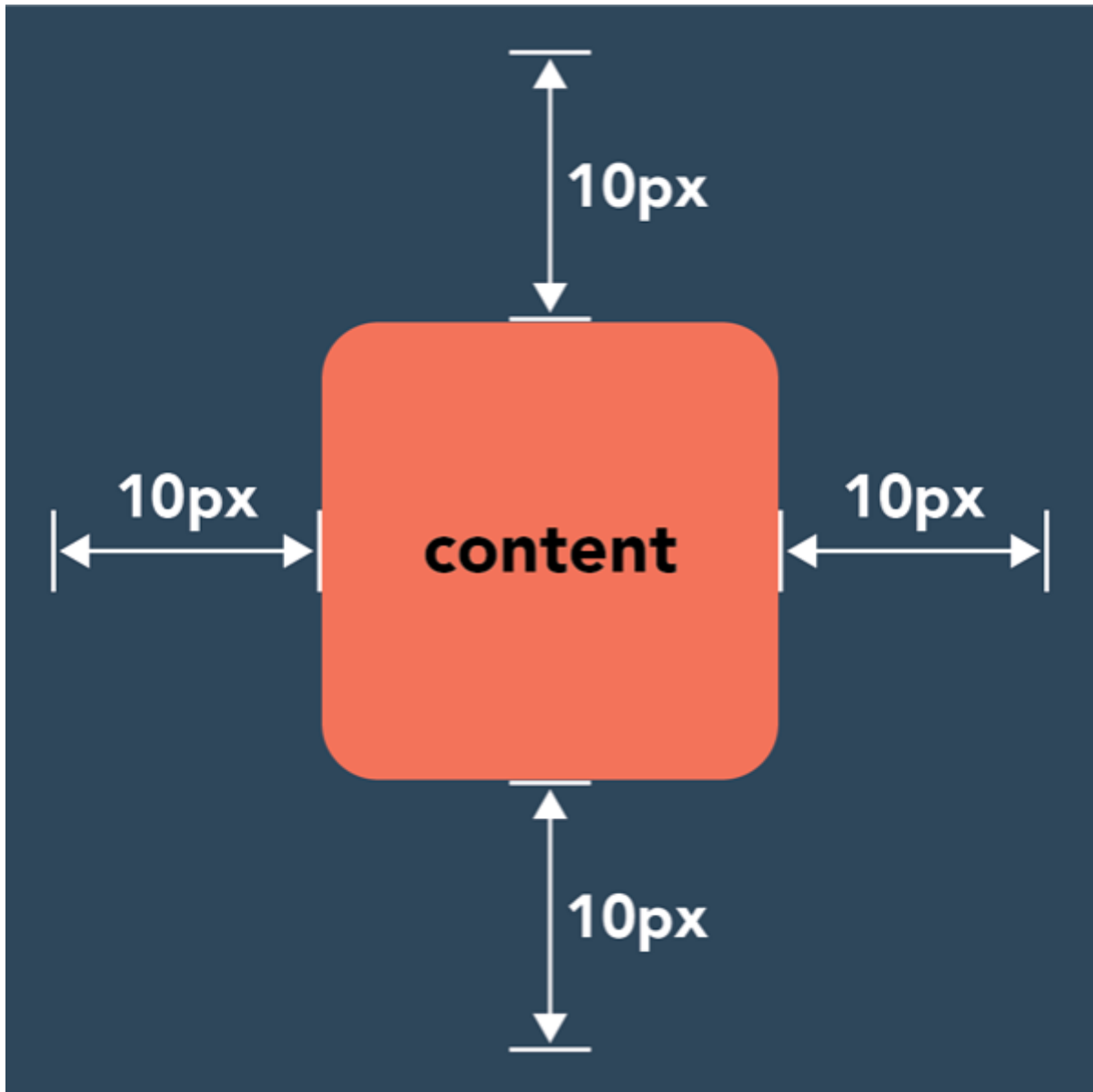


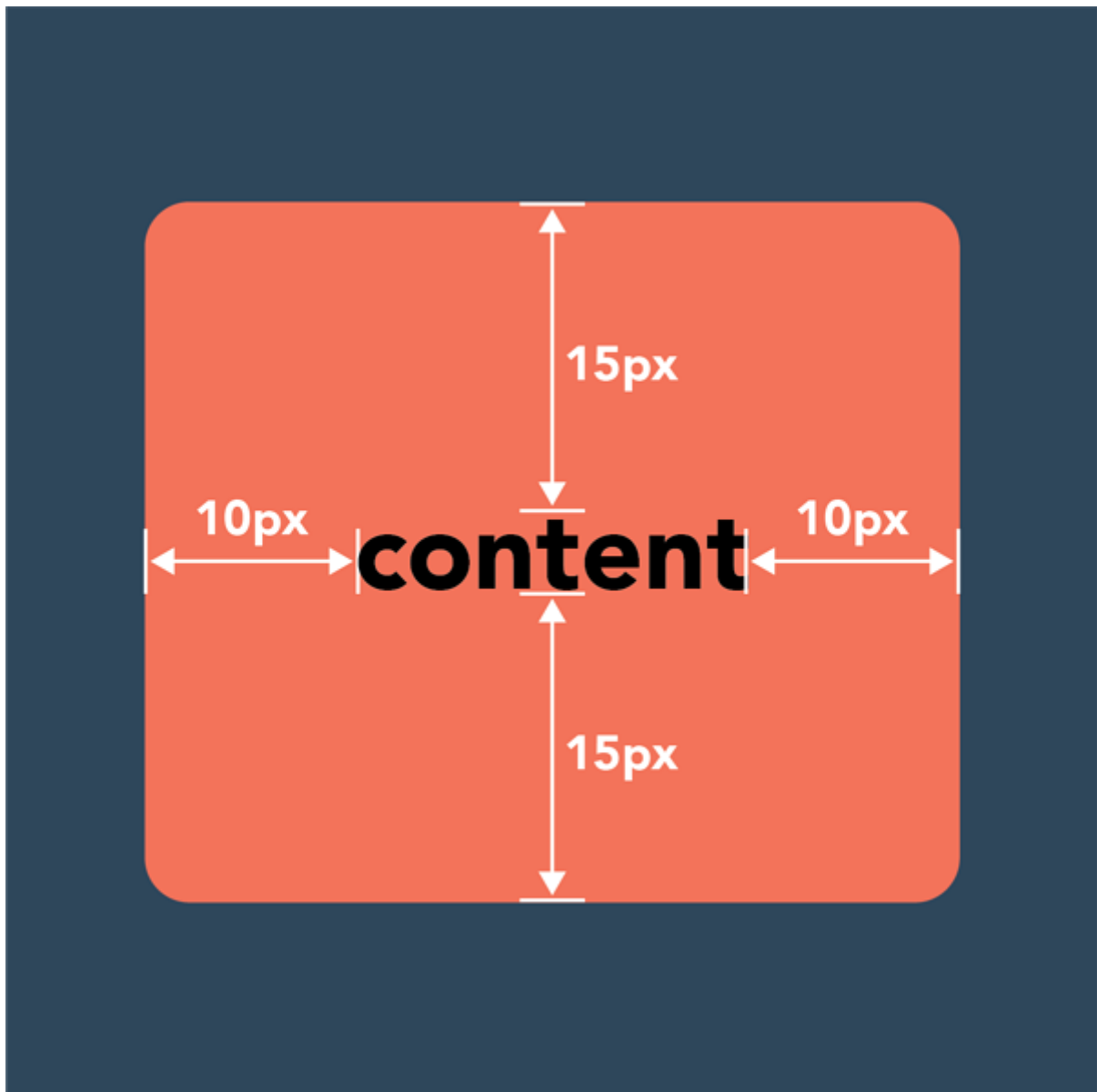
Quelle est la différence entre MARGIN et PADDING ?

Les propriétés *margin* et *padding* se confondent sur un point, la modification de marges, mais diffèrent sur le type de marges. En effet, *margin* façonne les marges extérieures à un élément. En d'autres termes, cette propriété définit l'espace existant entre cet élément et tout ce qui l'entoure.



Elle s'applique donc uniquement par rapport aux composants extérieurs, en les éloignant ou les rapprochant. Ainsi, elle a la capacité de centrer l'élément sur la page, en modifiant l'espace existant jusqu'à la bordure de la feuille. L'ajout de la valeur auto répartira automatiquement les marges hautes et basses. Enfin, l'utilisation d'une valeur négative, telle que `margin: -3px`, donne la possibilité de superposer plusieurs éléments.

Quant à la propriété padding, elle permet de modifier les marges intérieures, qui correspondent à l'espace contenu entre l'élément et son contour.

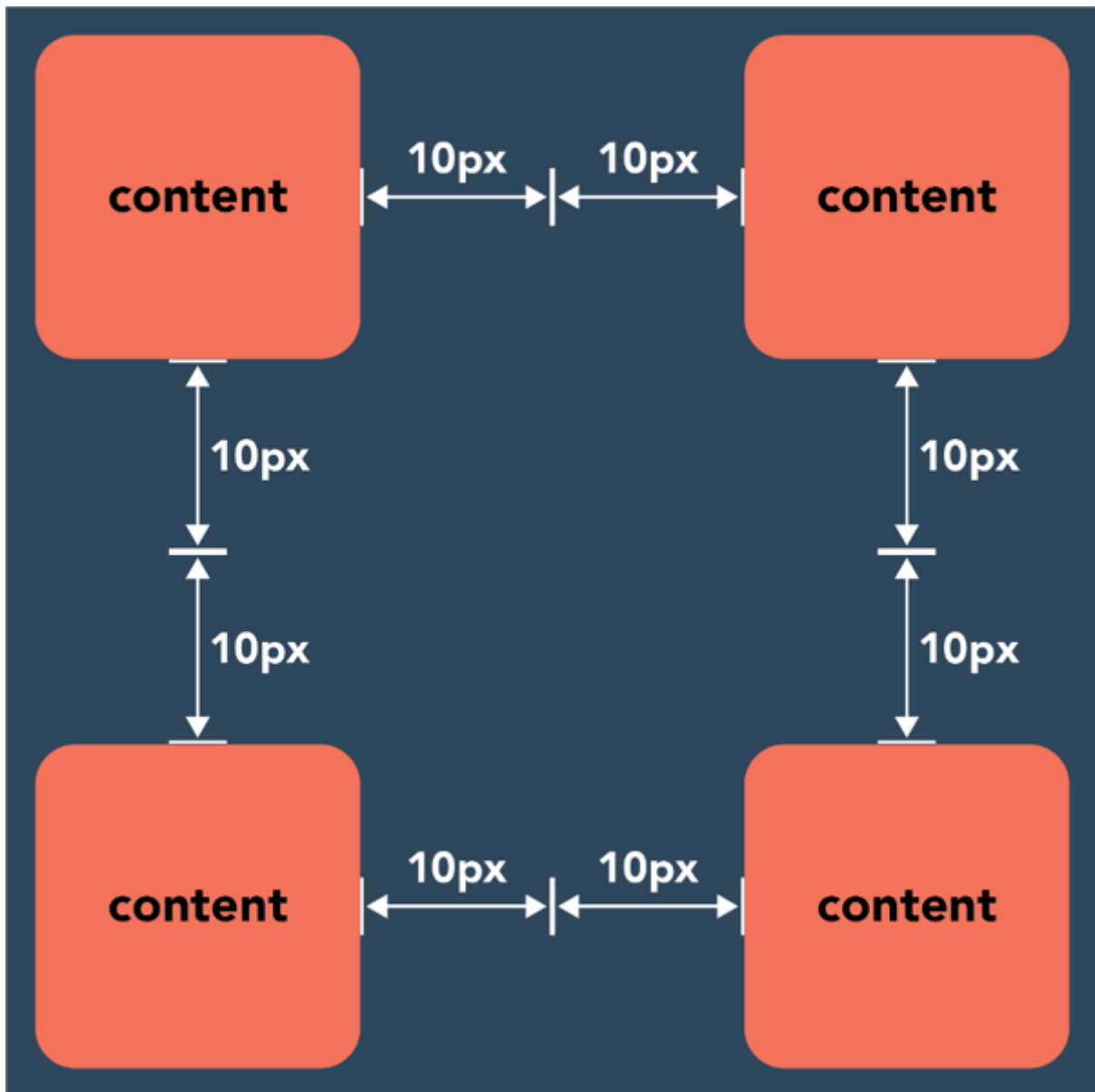


Contrairement à margin, padding n'impacte que l'élément en lui-même, en l'éloignant ou l'approchant du bord, sans toucher à la position de ce qui l'entoure. Cette propriété peut aider à étendre le fond de l'élément si nécessaire, puisqu'elle reproduit la couleur de fond.

Comment fonctionne le margin ?

Le margin s'exprime en pixel, en pourcentage, en point ou en « em », qui correspond à la taille de la police. Il s'applique indépendamment ou spécifiquement sur chacun des côtés de l'élément selon la précision de la marge. Si une seule valeur est précisée, alors elle s'applique aux

quatre *margins*, *left*, *right*, *top*, *bottom* (gauche, droit, haut, bas) de manière uniforme.



En revanche, si 4 valeurs se succèdent, comme 10px, 5px, 1px, 5px, alors la première valeur correspond à la marge du haut, la seconde de droite, la troisième du bas et la quatrième de gauche. La fluctuation de cet espace est notamment utilisée pour modifier la position de l'élément sur la page, superposer ou séparer deux éléments et plus globalement pour aérer la présentation d'une page afin de faciliter la lecture.

```
HTML
1 - <h1>Objectif :
  Augmentation des
  ventes</h1>
2
3 - <p class="p1">Définir
  une cible adaptée</p>
4 - <p
  class="p2">Utiliser
  un CRM pour améliorer
  la relation
  client</p>
5 - <p
  class="p3">Réaliser
  un argumentaire de
  vente efficace</p>
6

CSS
1 - .p1{
2   background-color:
  lightblue;|
3   margin: 10px 0px
  10px 10px
4 }
5
6 - .p2{
7   background-color:
  lightpink;
8   margin: -20px 0px
  30px 200px
9 }
10
11 - .p3{
12  background-color:
  lightyellow;
13  margin: -40px 0px
  40px 550px
14 }
```

Objectif : Augmentation des ventes

Définir une cible adaptée Utiliser un CRM pour améliorer la relation client Réaliser un argumentaire de vente efficace

Dans l'exemple ci-dessus, le `margin` permet à chaque paragraphe de se décaler par rapport au précédent pour que chacun se chevauche et soit également lisible. Ainsi le premier paragraphe respecte une marge extérieure de 10 pixels en haut, 0 pixel à droite, 10 pixels en bas et 0 pixel à gauche. Le second paragraphe observe une marge de -20 pixels en haut, 0 pixel à droite, 30 pixels en bas et 200 pixels à gauche. Enfin, le dernier paragraphe exécute une marge de -40 pixels en haut, 0 pixel à droite, 40 pixels en bas et 200 pixels à gauche.

Comment fonctionne le `padding` ?

Le `padding` s'exprime avec les mêmes unités que le `margin` et se comporte de manière identique lorsqu'une ou plusieurs valeurs sont spécifiées. Il est généralement utilisé pour ajuster l'espace présent entre le contenu et le bord,

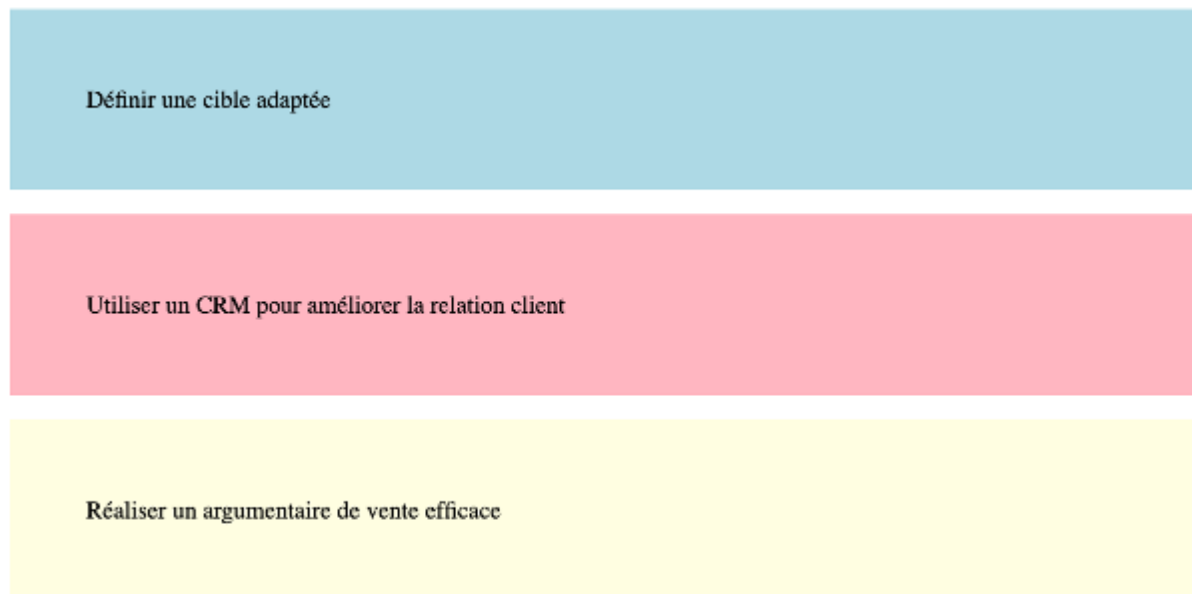
pour étendre ou réduire la dimension du contenu sans impacter la mise en page globale, ou encore pour prolonger le fond si celui-ci est coloré.



```
HTML
1 <h1>Objectif :
  Augmentation des
  ventes</h1>
2
3 <p class="p1">Définir
  une cible adaptée</p>
4 <p class="p2">Utiliser
  un CRM pour améliorer
  la relation client</p>
5 <p class="p3">Réaliser
  un argumentaire de
  vente efficace</p>
6

CSS
1 .p1{
2   background-color: lightblue;
3   padding: 50px 0px 50px 50px
4 }
5
6 .p2{
7   background-color: lightpink;
8   padding: 50px 0px 50px 50px
9 }
10
11 .p3{
12  background-color: lightyellow;
13  padding: 50px 0px 50px 50px
14 }
```

Objectif : Augmentation des ventes



Dans l'exemple ci-dessus, les padding ont été appliqués à chaque paragraphe de manière à étendre le fond de couleur, tout en laissant plus d'espace entre chaque texte. Ainsi, chaque paragraphe applique une marge interne de 50 pixels en haut, 0 pixel à droite, 50 pixels en bas et 50 pixels à gauche.

Qu'est-ce que le modèle de boîte CSS ?

Le langage CSS fonctionne sur le principe de boîtes, c'est-à-dire que chaque élément qui constitue une page web est une boîte. Cette dernière est divisée en 4 zones : *margin* qui correspond à la marge extérieure, *border* qui est le contour de l'élément, *padding*, la marge intérieure et enfin *content*, le contenu de l'élément.

Deux types de boîtes existent : les boîtes en bloc dites « block boxes » et les boîtes en ligne dites « inline boxes ». La boîte en bloc occupe par défaut toute la largeur disponible sur la page et place les éléments suivants à la ligne. Les titres et les paragraphes sont automatiquement définis comme des boîtes en bloc, mais ils peuvent être modifiés en boîte en ligne en le spécifiant. Pour sa part, la boîte en ligne ordonne par défaut aux autres éléments de se succéder sur la même ligne. Les propriétés *width* et *height* ne fonctionnent pas et seuls les *padding* s'appliquant à droite et à gauche provoquent le déplacement des éléments environnants.

Le modèle de boîte CSS configure la manière dont tous les paramètres s'harmonisent pour aboutir à l'aspect final de l'élément. Il s'applique complètement aux boîtes en bloc et seulement partiellement aux boîtes en ligne. Deux modèles de boîte CSS existent : le standard et l'alternatif. Concernant le modèle standard, les propriétés *width* et *height* définissent la taille du contenu de l'élément. Pour obtenir la hauteur totale de l'élément, il faut additionner la valeur de *height*, deux fois la valeur de *border* et deux fois la valeur de *padding*. De même, pour obtenir la largeur totale, il faut additionner la valeur de *width*, deux fois la valeur de *border* et deux fois la valeur de *padding*. Le calcul n'est pas compliqué, mais il peut impliquer plusieurs ajustements pour obtenir le résultat souhaité. Tandis que le modèle alternatif inclut toutes ces dimensions dans *width* et dans *height*. Autrement dit, *width* définit la largeur totale de l'élément et *height*, la hauteur totale. La gestion des dimensions est donc plus simple et rapide avec ce modèle. Pourtant, le modèle standard est celui utilisé par défaut. La propriété *box-sizing : border-box;* permet d'utiliser le modèle alternatif.