

CSS



QU'EST-CE QUE CSS ?

Qu'est-ce que CSS ?

CSS (Cascading Style Sheets) est un langage de feuilles de style utilisé pour styliser les documents écrit en HTML. Il décrit comment les éléments doivent être affichés à l'écran, sur du papier, en parole ou sur d'autres médias.

CSS offre un contrôle incroyable sur la mise en page, la couleur, la typographie et le design en général de la page.

Qu'est-ce que CSS ?

Il existe plusieurs manières d'écrire du CSS

La balise HTML <style>:



```
1 <style>
2      /* Utilisation d'une balise HTML <style> */
3 </style>
```

L'attribut HTML style="":



```
1 <p style="color: red;">Lorem ipsum dolor sit amet.</p>
```

Qu'est-ce que CSS ?

Créer un fichier externe (Ex. style.css) puis l'inclure au fichier HTML:



```
1 <link rel="stylesheet" href="assets/css/style.css">
```

LA SYNTAXE DE CSS

La syntaxe de CSS

Pour comprendre la syntaxe de CSS, il faut retenir 3 choses :

1. **selecteur** - Permet de sélectionner le contenu HTML sur lequel le CSS doit s'appliquer.
2. **propriete** - Correspond à ce que l'on souhaite définir sur l'élément (couleur/typo/alignement/etc..).
3. **valeur** - La valeur est la mise en forme à appliquer pour une propriété donnée.



```
1 selecteur {  
2     propriete: valeur;  
3 }
```



```
1 p {  
2     color: red;  
3 }
```

LES SÉLECTEURS

Les sélecteurs

Voici une explication des sélecteurs les plus courants :

Sélecteur d'élément

Ils ciblent tous les éléments HTML du même type.



```
1  p {  
2      color: red;  
3  }
```

Les sélecteurs

Sélecteur de classe

Ils ciblent tous les éléments possédant un certain attribut **class**. Ils sont précédés par un (.)

HTML



```
1 <p class="alert">Ceci est un message d'alerte</p>
```

CSS



```
1 .alert {  
2     background-color: red;  
3     color: white;  
4 }
```

Les sélecteurs

Sélecteur d'ID

Ils ciblent un élément spécifique basé sur son attribut ID. Ils sont précédés par un dièse (#).

HTML

```
● ● ●  
1 <p id="success">Félicitations!</p>
```

CSS

```
● ● ●  
1 #success {  
2     background-color: green;  
3     color: black;  
4 }
```

Les sélecteurs

Sélecteur descendant

Ils ciblent les éléments qui sont descendants d'un élément particulier.

HTML

```
● ● ●  
1 <section>  
2   <h2>Titre de la section</h2>  
3 </section>
```

CSS

```
● ● ●  
1 section h2 {  
2   color: aquamarine;  
3 }
```

Les sélecteurs

Sélecteur de groupe

Permet de cibler plusieurs sélecteurs pour leur appliquer le même style.

HTML



```
1 <h1>Titre principal</h1>
2 <h2>Sous-titre</h2>
3 <h3>Sous-titre</h3>
```

CSS



```
1 h1, h2, h3 {
2   font-family: Arial, sans-serif;
3 }
```

Les sélecteurs

Sélecteur d'enfant direct

Ils ciblent les éléments qui sont des enfants directs d'un élément parent.

HTML

```
1 <div>
2   <p>Paragraphe 1</p>
3   <section>
4     <p>Paragraphe 2</p>
5   </section>
6 </div>
```

CSS

```
1 <div> > p {
2   color: red;
3 }
```

<div> &

<div> &

Les balises <div> et sont deux balises génériques et polyvalentes en HTML qui sont souvent utilisés en association avec CSS pour structurer et styliser le contenu.

DIV

<div> est une balise de type bloc servant à contenir d'autres balises HTML et y appliquer des styles



```
1 <div style="background-color: lightblue;">
2   <h2>Ceci est un titre dans une div</h2>
3   <p>Ceci est un paragraphe dans une div.</p>
4   <a href="">Ceci est un lien dans une div</a>
5 </div>
```

<div> &

Les balises <div> et sont deux balises génériques et polyvalentes en HTML qui sont souvent utilisés en association avec CSS pour structurer et styliser le contenu.

SPAN

 est une balise de type inline servant à appliquer des styles sur une portion de texte à l'intérieur d'un élément de type block



```
1 <div style="background-color: lightblue;">
2   <h2>Ceci est <span style="color: red">un titre dans une div</span></h2>
3   <p>Ceci est un paragraphe dans une div.</p>
4   <a href="">Ceci est un lien dans une div</a>
5 </div>
```

LES PROPRIÉTÉS CSS

Les couleurs

En CSS, vous pouvez spécifier les couleurs de différentes manières pour styliser les éléments HTML.

1. Couleurs Nommées



```
1 p {  
2   color: red;  
3   background-color: yellow;  
4 }
```

2. Valeurs Hexadécimales



```
1 div {  
2   color: #FFFFFF;  
3   background-color: #000000;  
4 }
```

3. Valeurs RGB



```
1 body {  
2   color: rgb(255, 255, 255); /* blanc */  
3   background-color: rgb(0, 0, 0); /* noir */  
4 }  
5
```

4. Valeurs RGBA



```
1 section {  
2   background-color: rgba(255, 0, 0, 0.5);  
3 }
```

Modifier les polices d'écriture

En CSS, on utilise la propriété “font-family” pour définir la police de caractères à utiliser pour le texte d'un élément. C'est un aspect fondamental du stylisme typographique sur le web.

```
● ● ●  
1 selector {  
2   font-family: "NomDeLaPolice", "NomDeLaPoliceDeSecours", TypeDePolice;  
3 }
```



```
● ● ●  
1 body {  
2   font-family: "Arial", "Helvetica", sans-serif;  
3 }
```

Modifier les polices d'écriture

Avec l'avènement de CSS3, vous pouvez intégrer des polices web à l'aide de @font-face, permettant d'utiliser des polices qui ne sont pas installées sur les machines des utilisateurs.



```
1 @font-face {  
2     font-family: 'Roboto';  
3     src: url('assets/fonts/Roboto.woff2') format('woff2'),  
4          url('assets/fonts/Roboto.woff') format('woff');  
5 }  
6  
7 body {  
8     font-family: 'Roboto', Arial, Helvetica, sans-serif;  
9 }
```

Les unités

En CSS, plusieurs unités et méthodes peuvent être utilisées pour spécifier la taille des éléments. Chacune a son propre usage et contexte d'application optimal.

1. Les unités absolues

px (Pixels)

- Surtout pour définir les tailles de bordures, ombres, et autres détails de conception.
- border: **3px solid black**

pt (Points)

- Utilisé principalement pour les documents imprimés.
- font-size: **12pt**

cm, mm, in (Centimètres, Millimètres, Pouces)

- Préférable pour les styles destinés à l'impression.
- margin: **1cm**

Les unités

En CSS, plusieurs unités et méthodes peuvent être utilisées pour spécifier la taille des éléments. Chacune a son propre usage et contexte d'application optimal.

2. Les unités relatives

em

- Pour les tailles de police et les espacements relatifs à la taille de police de l'élément parent.
- font-size: **2em**

%

- Similaire à em
- font-size: **50%**

rem

- Pour les tailles de police et les espacements relatifs à la taille de la police de l'élément racine
- font-size: **1.5rem**

Modifier les polices d'écriture // part. 2

font-size:

- Définit la taille de la police.
- Valeurs possible :
 - Absolues : px, pt.
 - Relatives : em, rem, %.
 - Précises : small, medium, large, x-large.

font-weight:

- Définit l'épaisseur des caractères d'un texte.
- Valeurs :
 - Numériques : de 100 à 900 (normal est 400, bold est 700).
 - Précises : normal, bold, bolder, lighter.

font-style:

- Définit le style de la police
- Valeurs : normal, italic, oblique.



```
1  h1 {  
2      font-size: 3rem;  
3      font-weight: bold;  
4      font-style: italic;  
5      text-decoration: underline;  
6      text-transform: capitalize;  
7      line-height: 1.6;  
8      letter-spacing: 1px;  
9      word-spacing: 2px;  
10 }
```

Modifier les polices d'écriture // part. 2

text-decoration:

- Définit la décoration appliquée au texte
- Valeurs :
 - none : supprime les décorations.
 - underline : souligne le texte.
 - overline : ligne au-dessus du texte.
 - line-through : ligne à travers le texte.

text-transform:

- Contrôle la capitalisation du texte.
- Valeurs :
 - none : aucun changement.
 - capitalize : met en majuscule la première lettre de chaque mot.
 - uppercase : met en majuscule
 - lowercase : met en minuscule



```
1  h1 {  
2      font-size: 3rem;  
3      font-weight: bold;  
4      font-style: italic;  
5      text-decoration: underline;  
6      text-transform: capitalize;  
7      line-height: 1.6;  
8      letter-spacing: 1px;  
9      word-spacing: 2px;  
10 }
```

Modifier les polices d'écriture // part. 2

line-height:

- Définit la hauteur de ligne utilisée pour le texte.
- Valeurs :
 - Nombre **sans unité** : multiplicateur de la taille de la police (1.5, 2, etc.).
 - **Avec unité** : px, em, rem, %, etc.

letter-spacing:

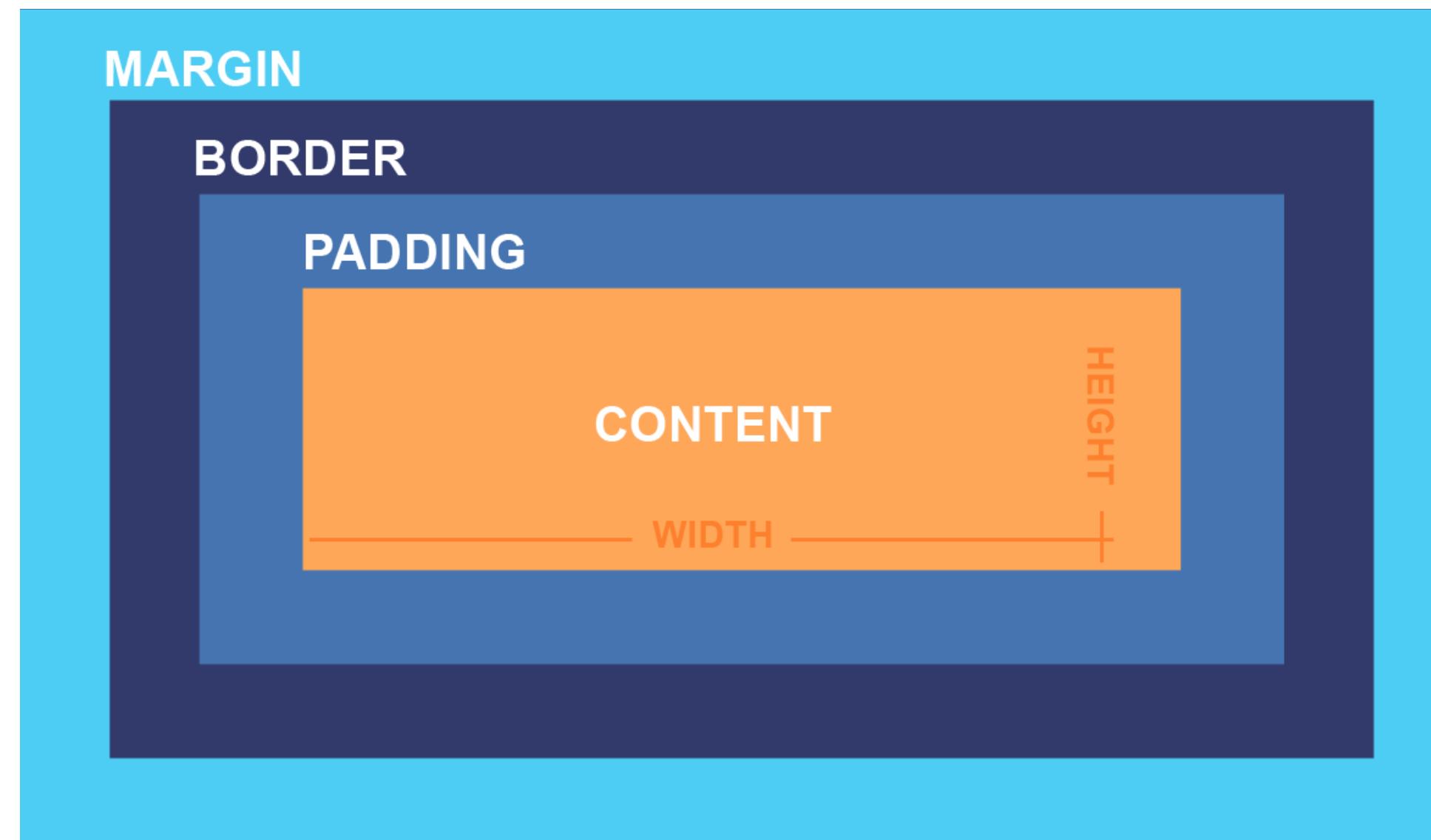
- Défini l'espacement entre les caractères du texte.
- **Valeurs :**
 - Valeurs avec unité (**px, em, rem**, etc.)
 - **normal**.



```
1  h1 {  
2      font-size: 3rem;  
3      font-weight: bold;  
4      font-style: italic;  
5      text-decoration: underline;  
6      text-transform: capitalize;  
7      line-height: 1.6;  
8      letter-spacing: 1px;  
9      word-spacing: 2px;  
10 }
```

Le Modèle de Boite

Le **modèle de boîte** est un concept fondamental en CSS, il décrit la façon dont sont structurés et dimensionnés les éléments HTML. Chaque élément est considéré comme une boîte rectangulaire, composée de plusieurs propriétés : le **contenu**, le **padding**, la **bordure** et la **marge**.



Le Modèle de Boîte

1. Le contenu

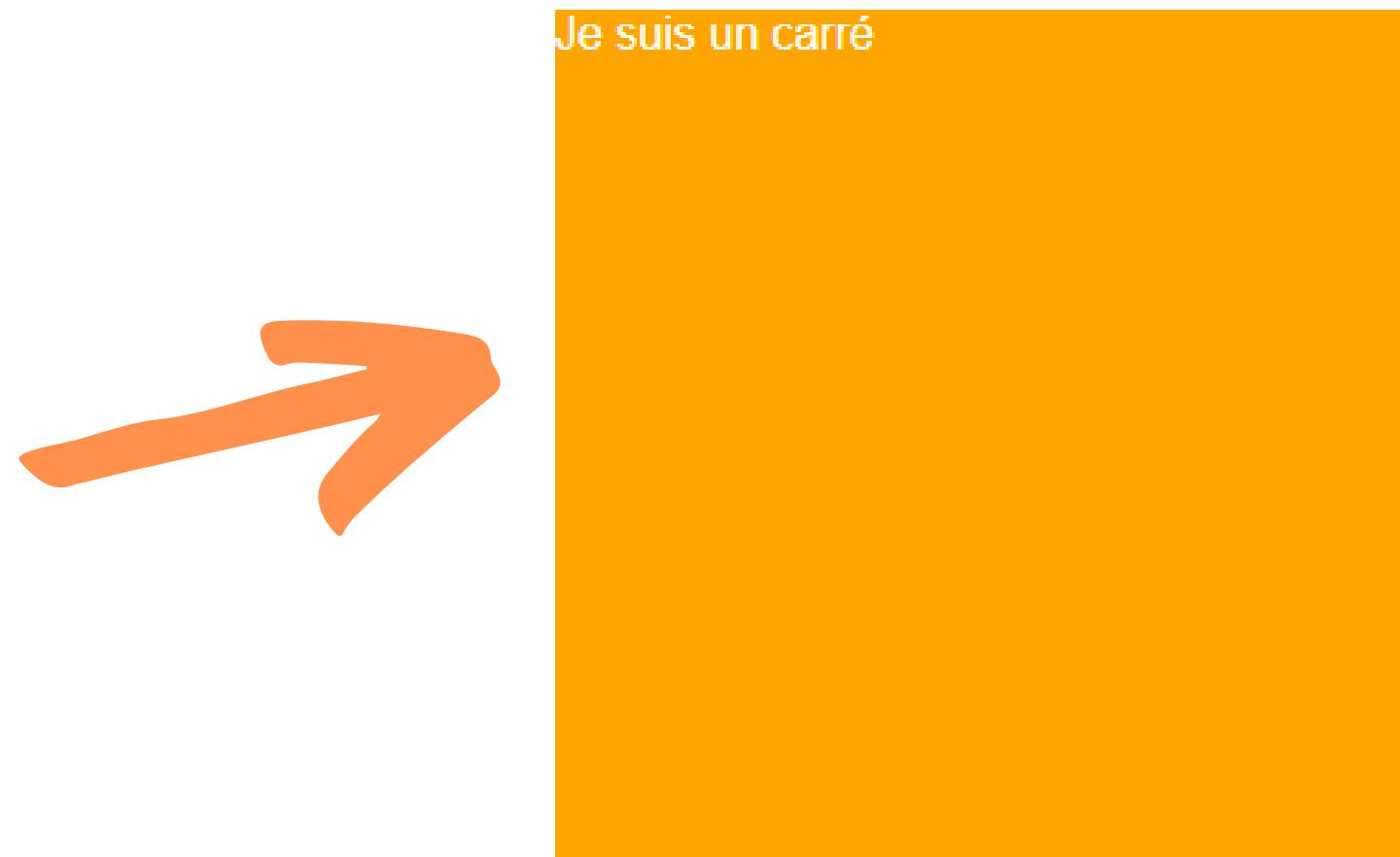
Par défaut, les éléments HTML ont une taille naturelle.

Par exemple, un élément de type bloc comme un `<p>` aura une largeur 100% par rapport à l'écran alors qu'un élément de type inline comme `<a>` n'aura que la largeur dont il a besoin.

Il est cependant possible de modifier la taille des éléments de type bloc.



```
1 .square {  
2     width: 300px;  
3     height: 300px;  
4     background-color: orange;  
5 }
```



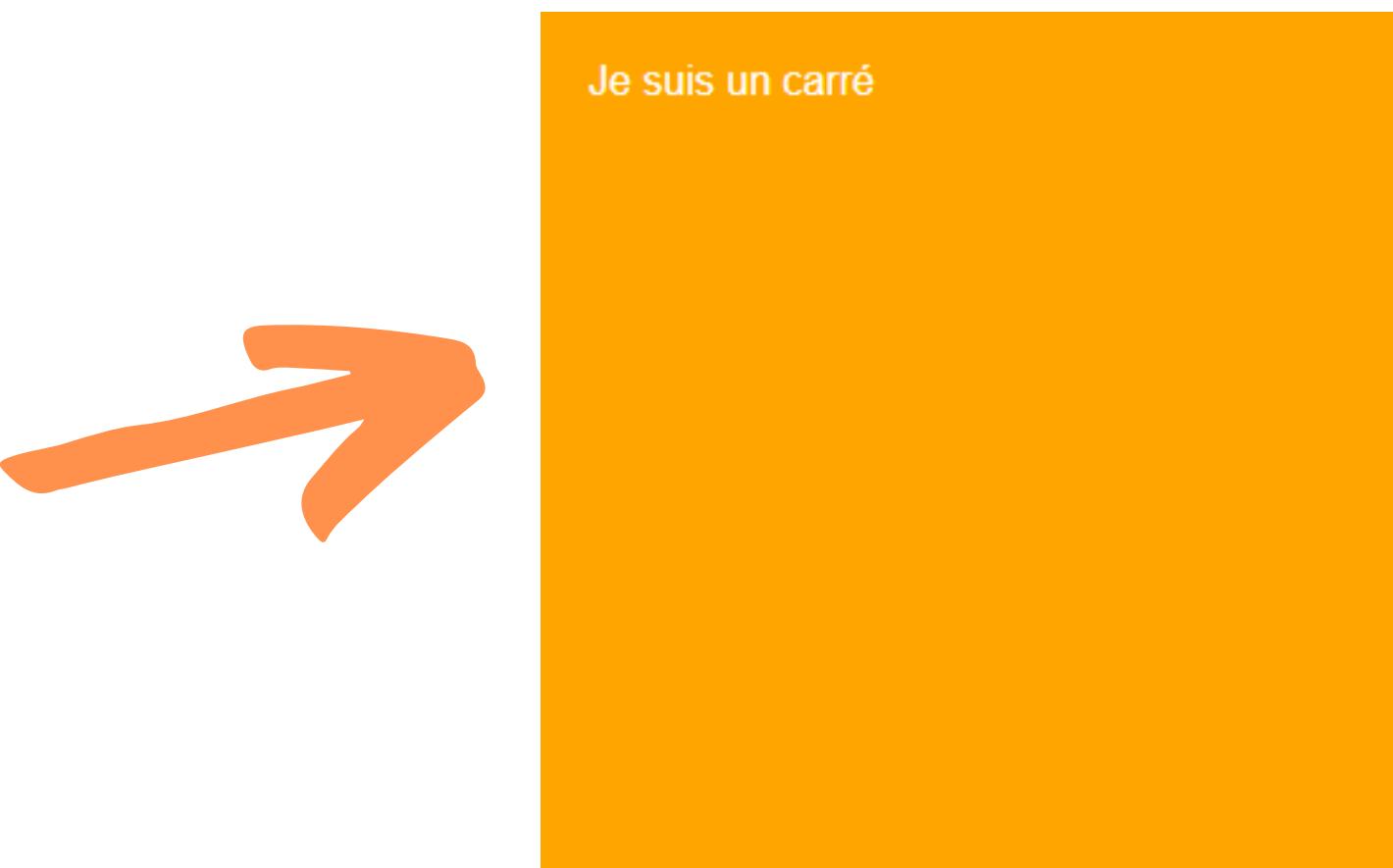
Le Modèle de Boîte

2. Le padding

Le padding représente l'espace entre le contenu d'un élément et sa bordure. Il s'agit d'une propriété qui permet de créer de l'espace autour du contenu, à l'intérieur de la boîte définie, augmentant ainsi l'espace visuel autour du contenu.



```
1 .square {  
2   width: 300px;  
3   height: 300px;  
4   background-color: orange;  
5   padding: 20px;  
6 }
```

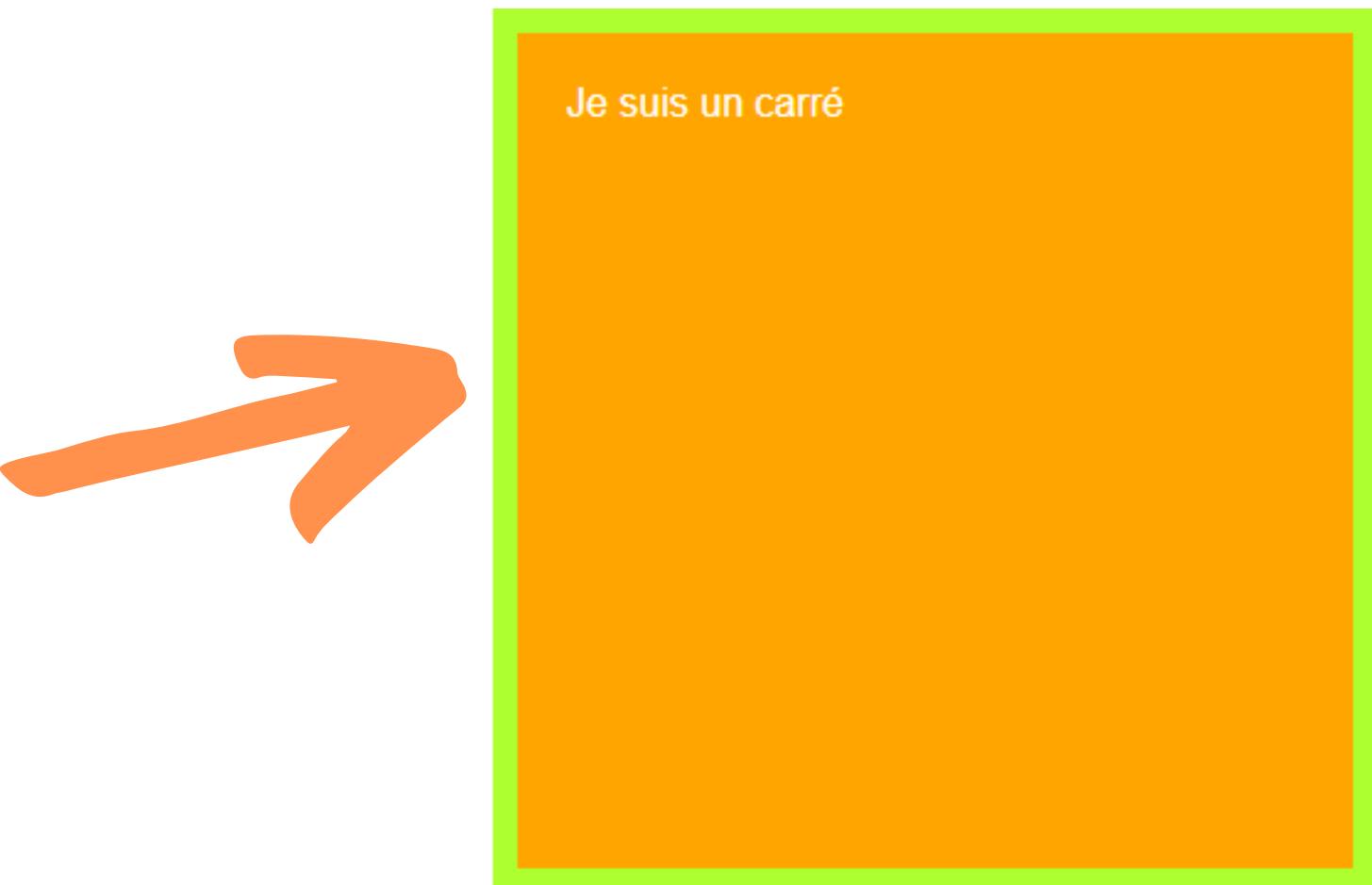


Le Modèle de Boîte

3. La bordure

La propriété **border** en CSS est utilisée pour définir une bordure autour d'un élément HTML. Elle fait également partie du modèle de boîte et se situe entre le padding et la margin.

```
● ● ●  
1 .square {  
2   width: 300px;  
3   height: 300px;  
4   background-color: orange;  
5   padding: 20px;  
6   border: 10px solid greenyellow;  
7 }
```



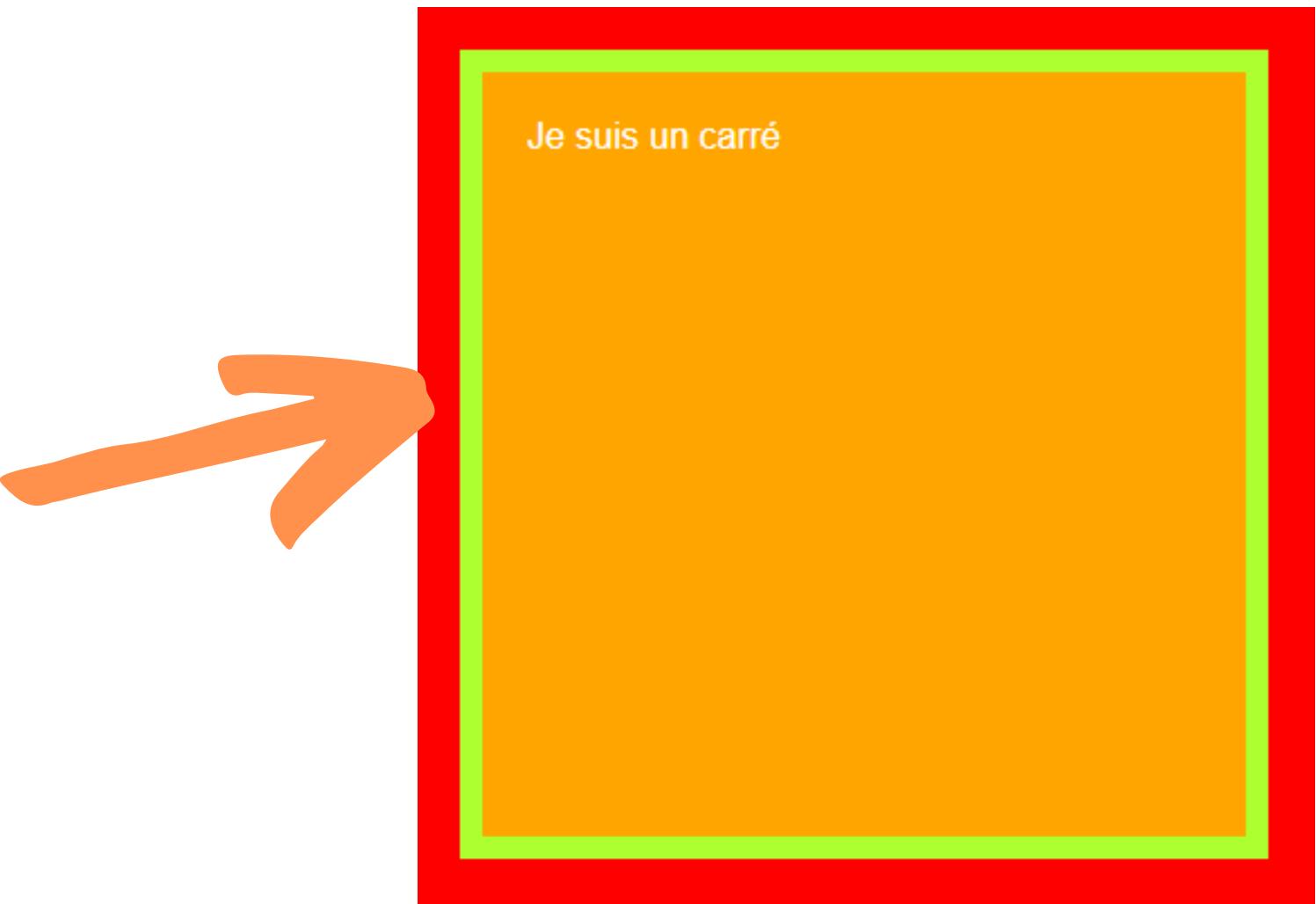
Le Modèle de Boîte

4. Le margin

La propriété **margin** est utilisée pour créer un espace autour des éléments.

Elle fait partie modèle de boîte et permet de définir l'espace entre les éléments HTML sur la page, aidant à structurer la disposition de la page de manière aérée et ordonnée.

```
● ● ●  
1 .square {  
2   width: 300px;  
3   height: 300px;  
4   background-color: orange;  
5   padding: 20px;  
6   border: 10px solid greenyellow;  
7   margin: 20px;  
8 }
```



Margin & Padding & Bordure

Pour définir le padding ou le margin d'un élément, vous avez plusieurs manières

1. Une seule valeur:



```
1 .margin-padding {  
2   margin: 20px;  
3 }
```

2. Plusieurs valeurs



```
1 .margin-padding {  
2   margin: 20px 15px 10px 5px;  
3 }
```

3. Valeur individuelle



```
1 .margin-padding {  
2   margin-top: 20px;  
3   margin-right: 15px;  
4   margin-bottom: 10px;  
5   margin-left: 10px  
6 }
```

+ Ajouter une bordure



```
1 .border {  
2   border: 1px solid red;  
3 }
```

Définir une image de fond

On utilise la propriété background-image pour définir une image de fond sur un élément HTML.



```
1 .background-image {  
2     background-image: url(..../img/dog.jpg);  
3     background-size: cover;  
4     background-position: center;  
5     background-repeat: no-repeat;  
6 }
```

Modifier le type d'affichage

La propriété **display** est extrêmement importante et est utilisée pour contrôler la manière dont les éléments sont affichés sur la page. Elle détermine le type d'affichage d'un élément et influe sur le comportement de l'élément en relation avec les autres éléments autour.



```
1 .display {  
2   display: block;  
3 }
```



```
1 .display {  
2   display: inline;  
3 }
```



```
1 .display {  
2   display: inline-block;  
3 }
```



```
1 .display {  
2   display: none;  
3 }
```



```
1 .display {  
2   display: flex;  
3 }
```

Les pseudos-classes

Les pseudo-classes sont utilisées pour définir un état particulier d'un élément. Il sera possible de styliser un élément lorsqu'il est survolé par la souris, lorsqu'il est cliqué, lorsqu'il a le focus, et bien d'autres situations.

```
● ● ●  
1 a:hover {  
2     color: red;  
3 }  
4  
5 button:active {  
6     background-color: blue;  
7 }  
8  
9 input:focus {  
10    border-color: green;  
11 }  
12  
13 a:visited {  
14     color: black;  
15 }
```

```
● ● ●  
1 ul li:first-child {  
2     font-weight: bold;  
3 }  
4  
5 ul li:last-child {  
6     font-style: italic;  
7 }  
8  
9 ul li:nth-child(3) {  
10    text-decoration: underline;  
11 }  
12  
13 ul li:not(:first-child) {  
14     color: purple;  
15 }
```

Les pseudos-éléments

Les pseudo-éléments CSS sont utilisés pour styliser des parties spécifiques d'un élément sélectionné qui ne sont généralement pas accessibles avec les sélecteurs CSS classiques.



```
1 .element::before {  
2   content: "Avant";  
3 }  
  
4 .element::after {  
5   content: "Après";  
6 }  
7 }
```



```
1 p::first-line {  
2   font-weight: bold;  
3 }  
  
4  
5 p::first-letter {  
6   font-size: 2em;  
7 }  
  
8  
9 p::selection {  
10  background-color: yellow;  
11 }
```

PROJET 1

The Underground

Les positions

La positive relative

Un élément avec **position: relative;** est positionné par rapport à sa position d'origine dans le flux normal du document. Vous pouvez ensuite utiliser les propriétés top, right, bottom, et left pour déplacer l'élément par rapport à là où il serait normalement, sans affecter la position des autres éléments.



```
1 .box-relative {  
2   width: 100px;  
3   height: 100px;  
4   background-color: midnightblue;  
5   position: relative;  
6   left: 50px;  
7   top: 20px;  
8 }
```

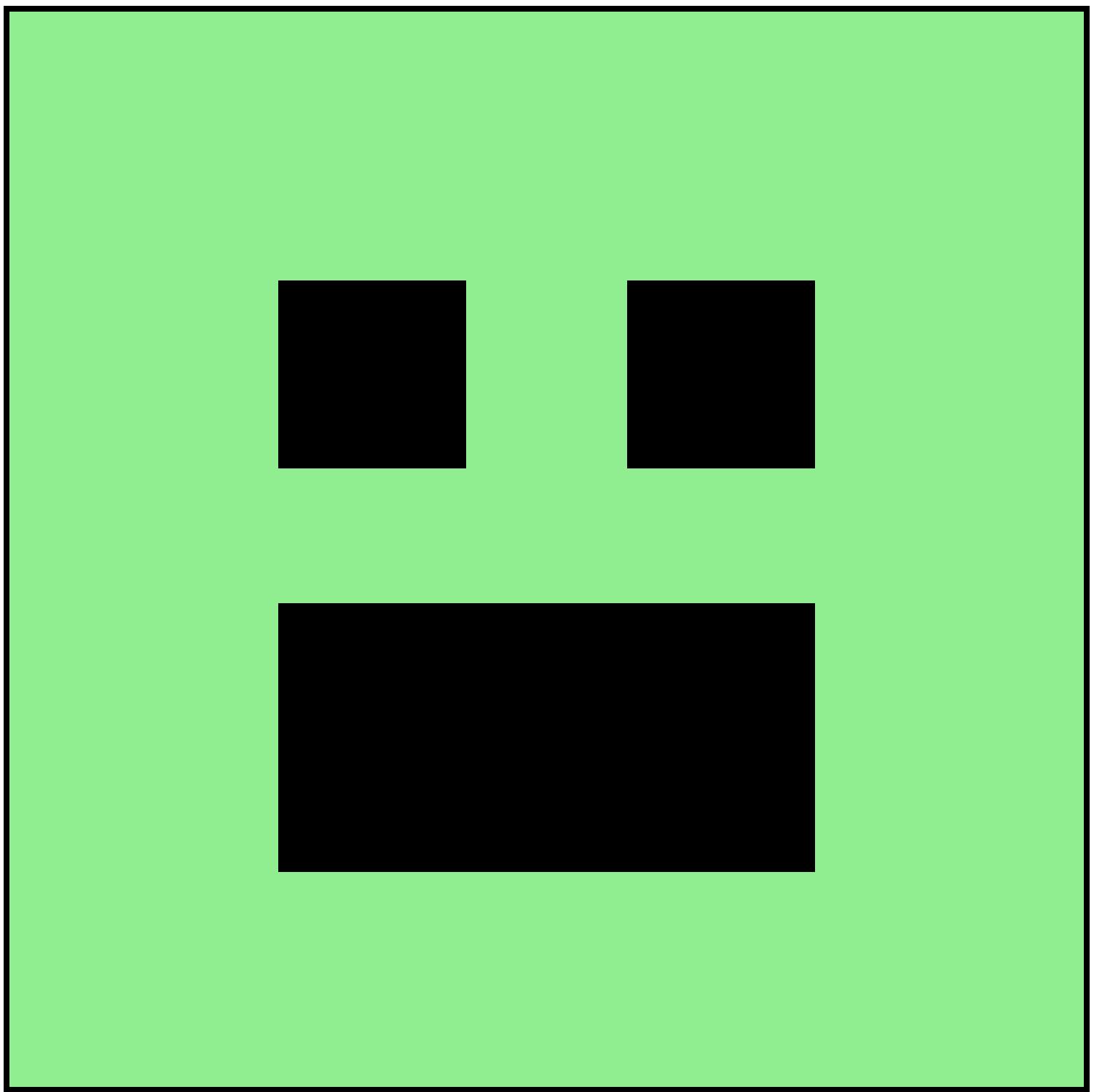
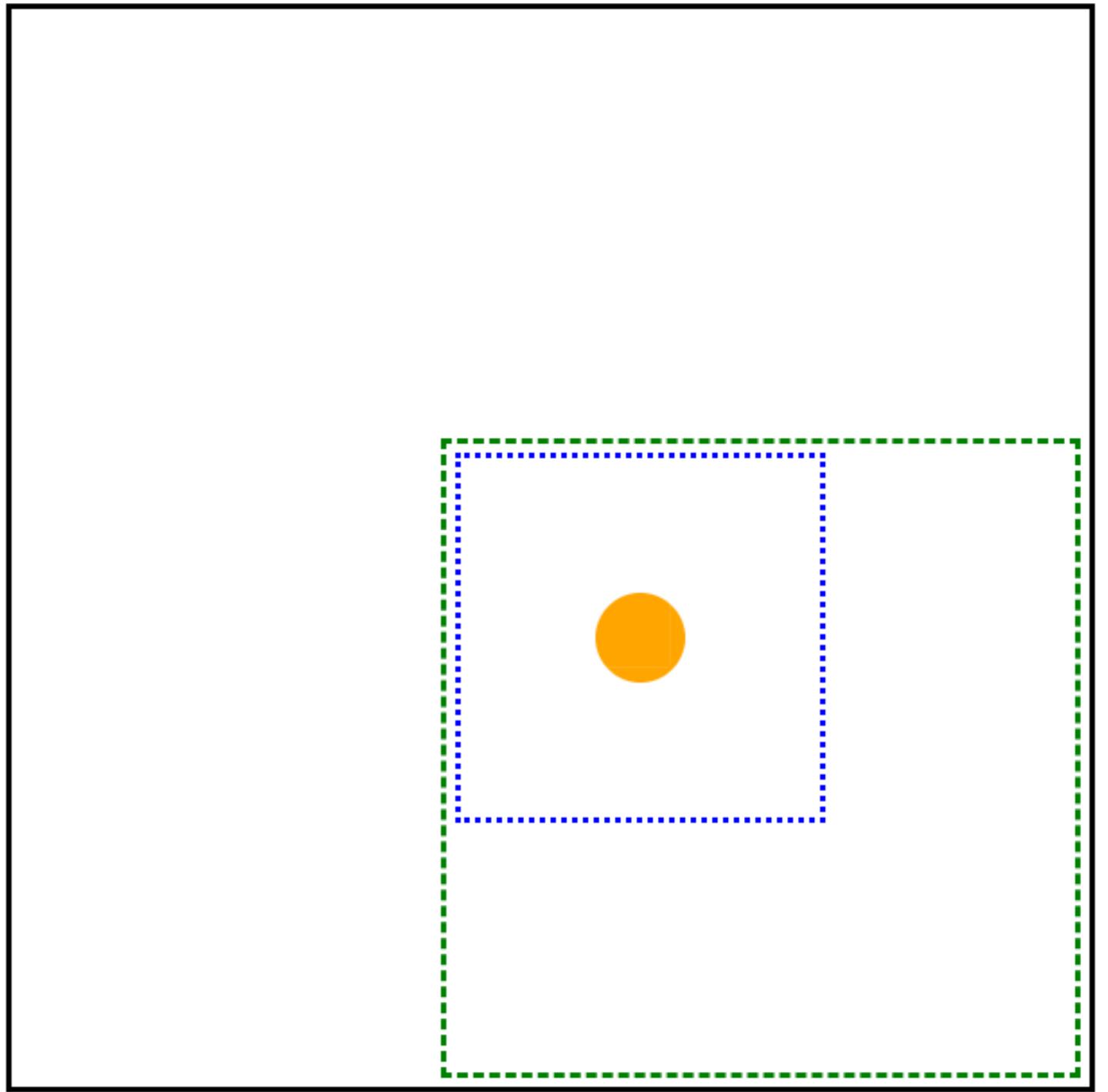
La positive absolute

Un élément avec **position: absolute;** est retiré du flux normal du document et est positionné par rapport à son ancêtre positionné le plus proche. Si aucun ancêtre positionné n'est trouvé, il sera positionné par rapport à l'élément <html>. Les propriétés top, right, bottom, et left déterminent la position de l'élément par rapport à l'élément de référence.

```
● ● ●  
1 .parent {  
2     position: relative;  
3     width: 200px;  
4     height: 200px;  
5     background-color: lightblue;  
6 }  
7  
8 .enfant {  
9     position: absolute;  
10    width: 50px;  
11    height: 50px;  
12    background-color: lightcoral;  
13    bottom: 0;  
14    right: 0;  
15 }
```

Parent

Enfant



FLEXBOX

Flexbox

Flexbox, ou le modèle de boîte flexible, est un outil moderne de mise en page en CSS qui permet de concevoir des structures complexes avec une syntaxe plus propre et prévisible. Flexbox est particulièrement utile lorsque les tailles de vos éléments sont inconnues ou dynamiques.

Concepts de base:

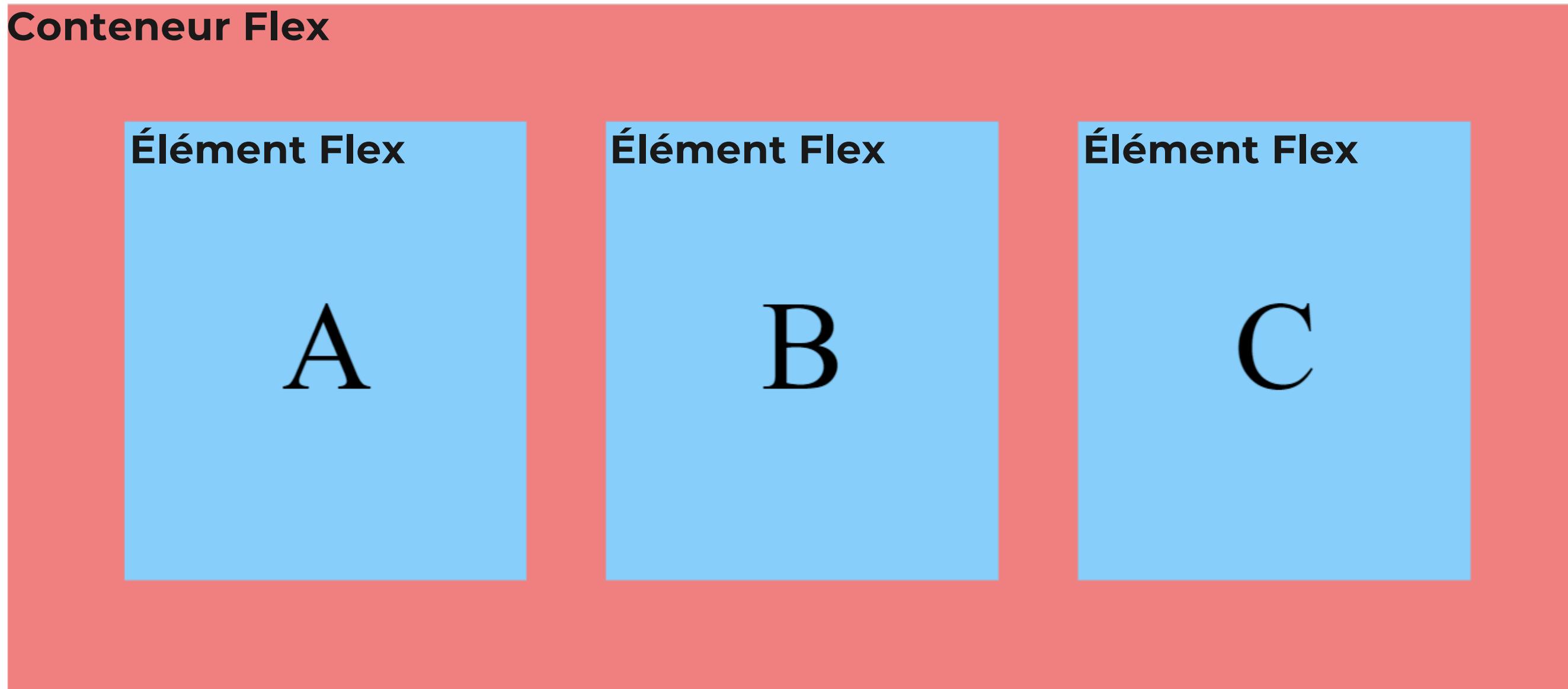
Container Flex:

Quand on parle du "Container Flex", on fait référence à l'élément parent qui aura des éléments enfants structurés par Flexbox. Pour définir un élément comme un Container Flex, on utilise la propriété display avec la valeur flex ou inline-flex.

Élément Flex:

Ces éléments flex peuvent être stylisés et manipulés à l'aide des propriétés flexbox spécifiques.

Flexbox



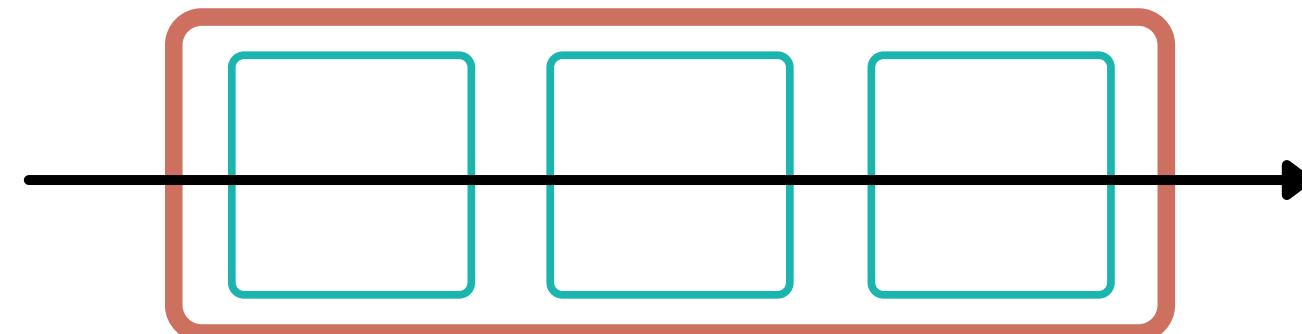
Flexbox

Lorsque vous appliquez la propriété **display: flex;** à un élément, celui-ci devient un Container Flex, et ses enfants directs deviennent des Éléments Flex.



```
1 .flex-container {  
2   display: flex;  
3 }
```

Par défaut, les éléments flex sont disposés selon l'axe horizontal, de gauche à droite

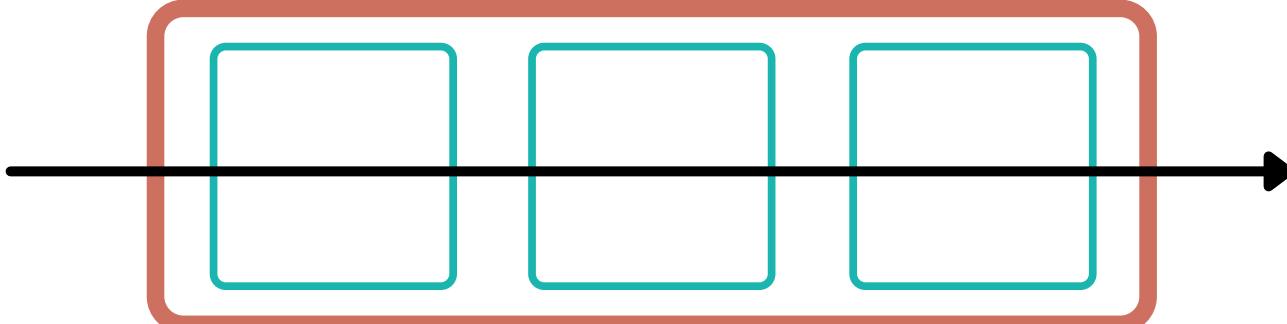


Flexbox

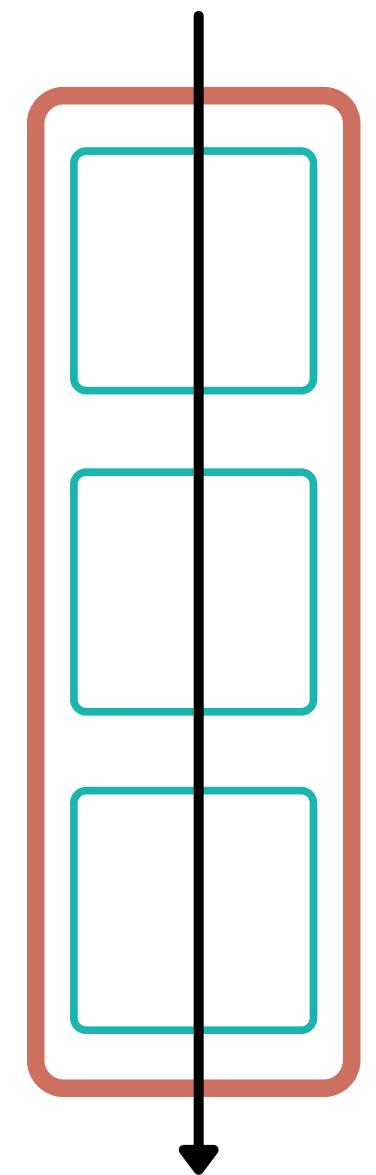
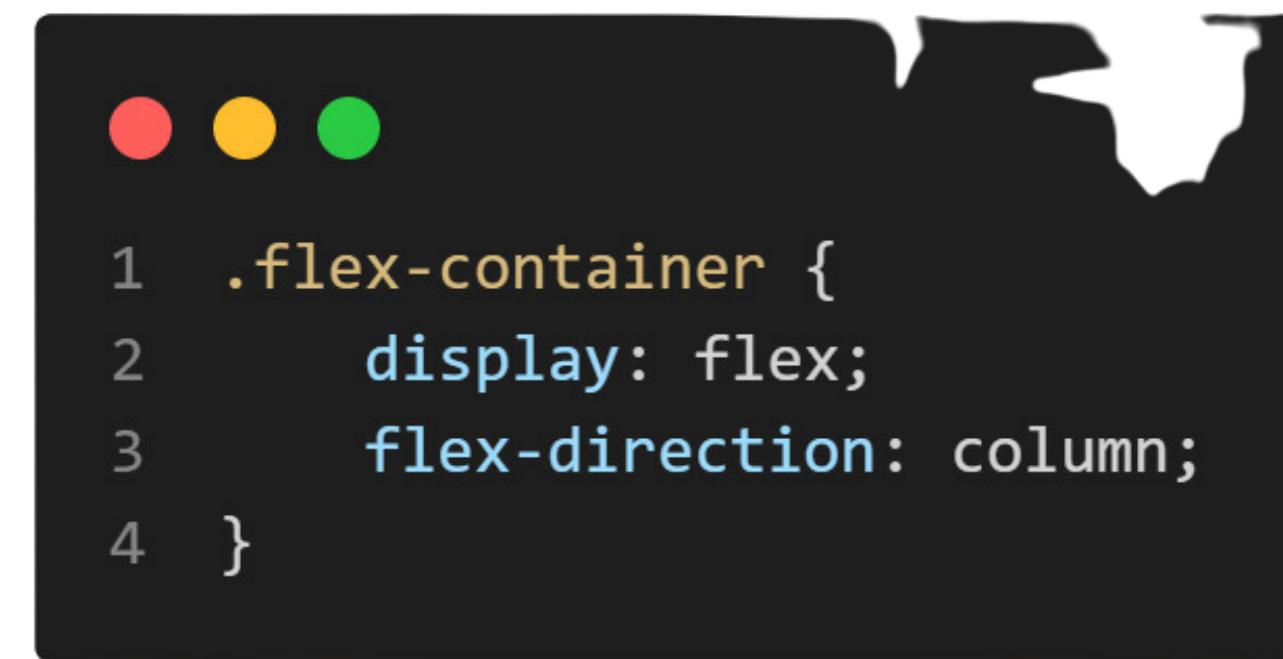
Il est possible de modifier la direction dans laquelle les éléments flex sont placés dans le container flex en déterminant la direction de l'axe principal (et de l'axe transversal) du container flex.



```
1 .flex-container {  
2   display: flex;  
3   flex-direction: row;  
4 }
```



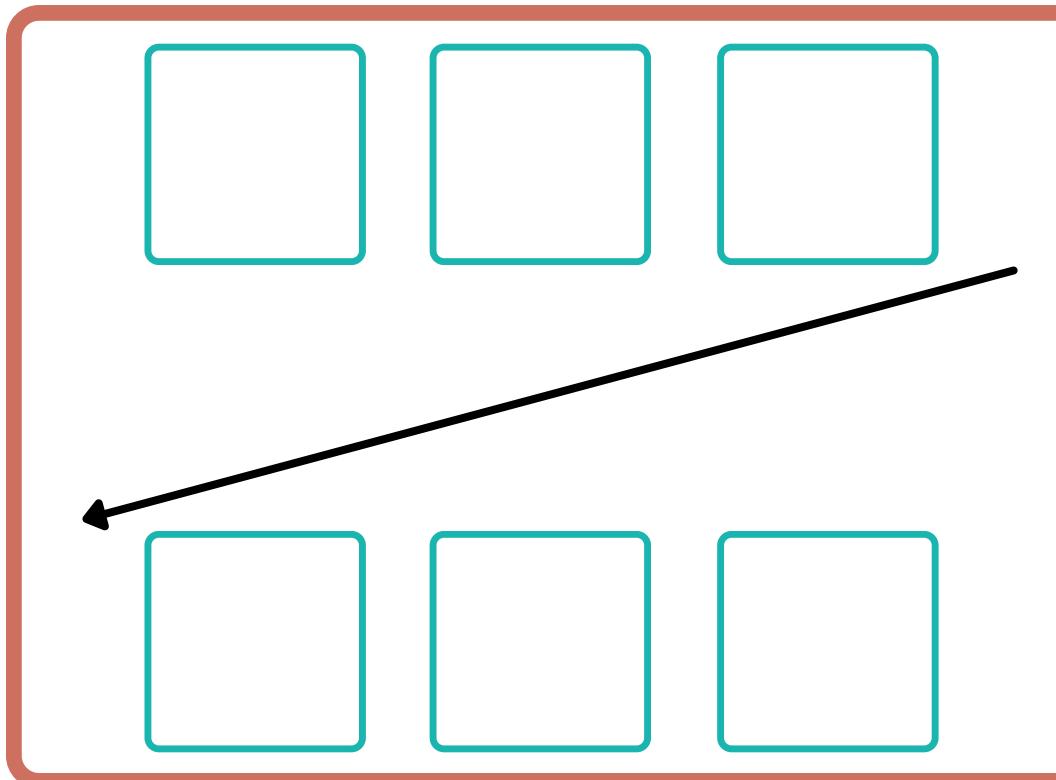
```
1 .flex-container {  
2   display: flex;  
3   flex-direction: column;  
4 }
```



Flexbox

La propriété **flex-wrap** appliquée au conteneur Flex va autoriser les éléments flex à retourner à la ligne si l'espace disponible sur la ligne de base n'est pas suffisant.

- **nowrap**: tout sur une seule ligne (par défaut)
- **wrap**: éléments se divisent sur plusieurs lignes si nécessaire.
- **wrap-reverse**: comme wrap, mais en inversant l'ordre des lignes.

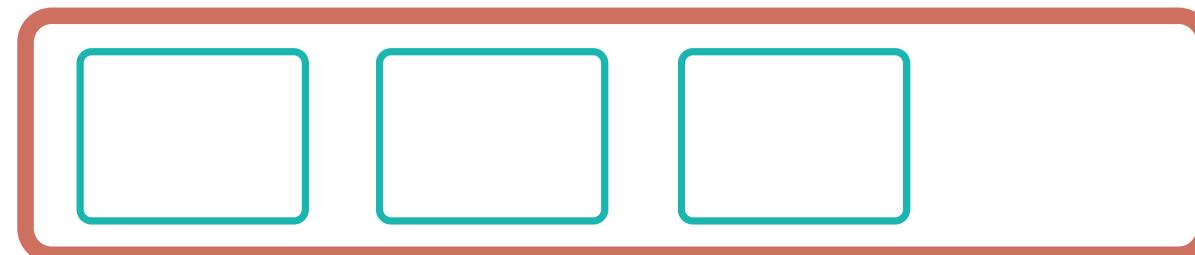


```
1 .flex-container {  
2   display: flex;  
3   flex-wrap: wrap;  
4 }
```

Flexbox

La propriété **justify-content** va définir la manière dont les éléments vont s'aligner sur l'axe principal (défini par flex-direction)

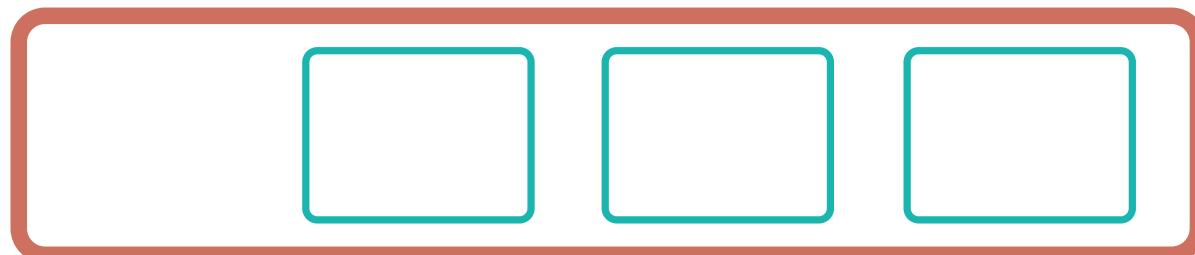
flex-start



space-between



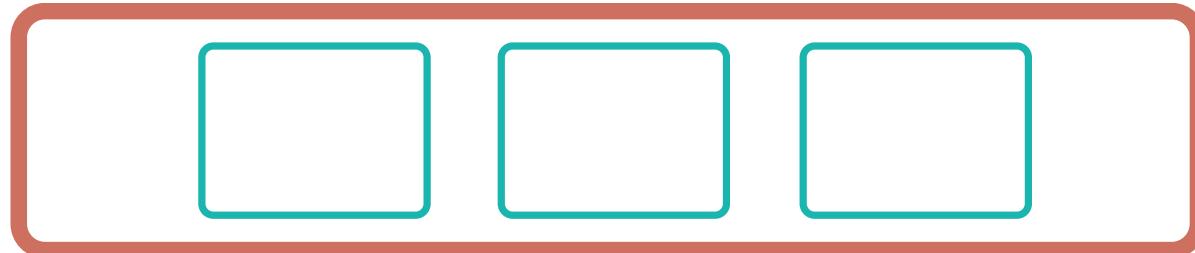
flex-end



space-around



center



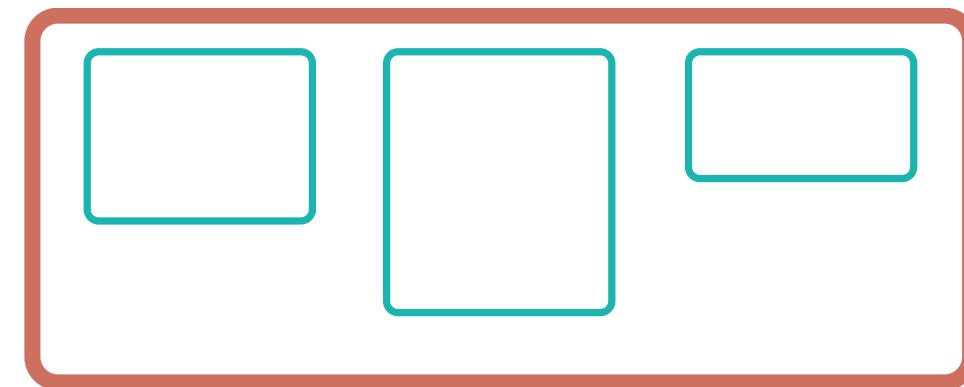
space-evenly



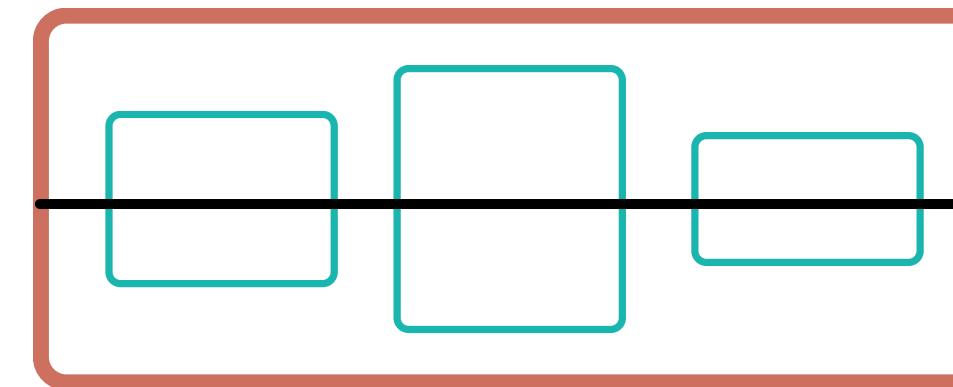
Flexbox

La propriété **align-items** va définir la manière dont les éléments vont être aligné sur l'axe secondaire (perpendiculaire à celle définie par flex-direction)

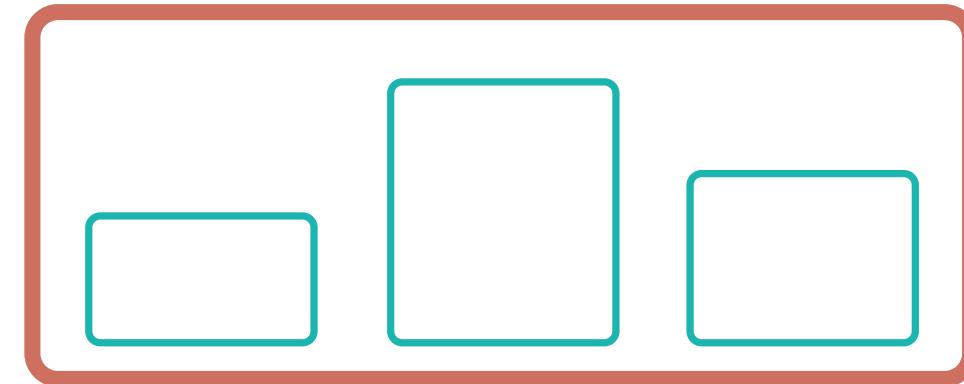
flex-start



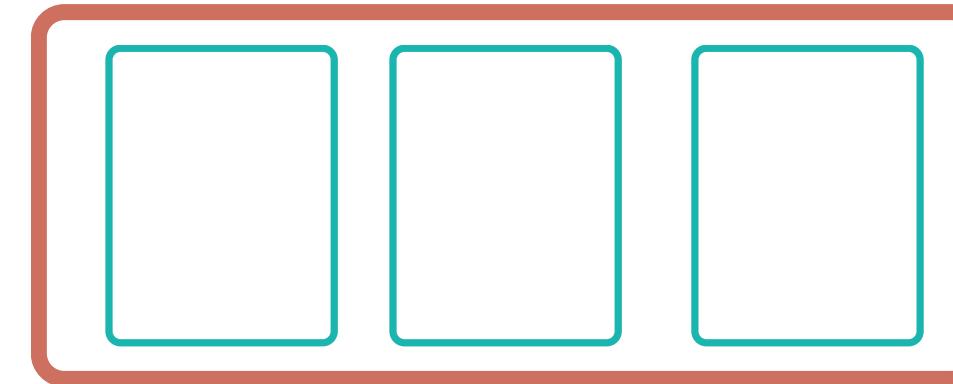
center



flex-end



stretch



PROJET 2

Destinations

PROJET 3

Resident Evil 4