Langage CSS

CSS (Cascade Style Sheet)

Priorité dans les CSS

```css
#content  nav li a.on:hover {
 color: red;
}
```

1 ID (#content), 2 classe+pseudo-classe (.on :hover), 3 éléments (nav li a) => 123.

```css
nav.verte li.gras a.on:hover {
 color: green;
}
```

0 ID, 4 classe+pseudo-classe (.on :hover .verte .gras), 3 éléments (nav li a) => 33.

- **Types de médias**
  - all : tous
  - braille : système tactile
  - handheld : système portable
  - print : imprimante
  - projection : système de projection
  - screen : moniteur
  - tv : télévision

Gestion des "média query"

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

```
<link rel="stylesheet" media="screen and (max-width: 1280px)"
href="petite_resolution.css" />
```

```
/* Paragraphes en bleu par défaut */

P {
    color: blue;
}

/* Nouvelles règles si la fenêtre fait au plus 1024px de large */

@media screen and (max-width: 1024px)
{
    p {
        color: red;
        background-color: black;
        font-size: 1.2em;
    }
}
```

Gestion des "média query"

```
<meta name="viewport" content="width=device-width" />
```

```
body {
font-size : 100%
}

 @media only screen and (max-device-width:980px) {
 body {
 font-size:120%;
 }}
```

Les deux exemples suivants ciblent les écrans de largeur inférieure à 640 pixels grâce à la règle ***max-width*** *associée à la valeur* ***640px***.

Modèle de boîtes

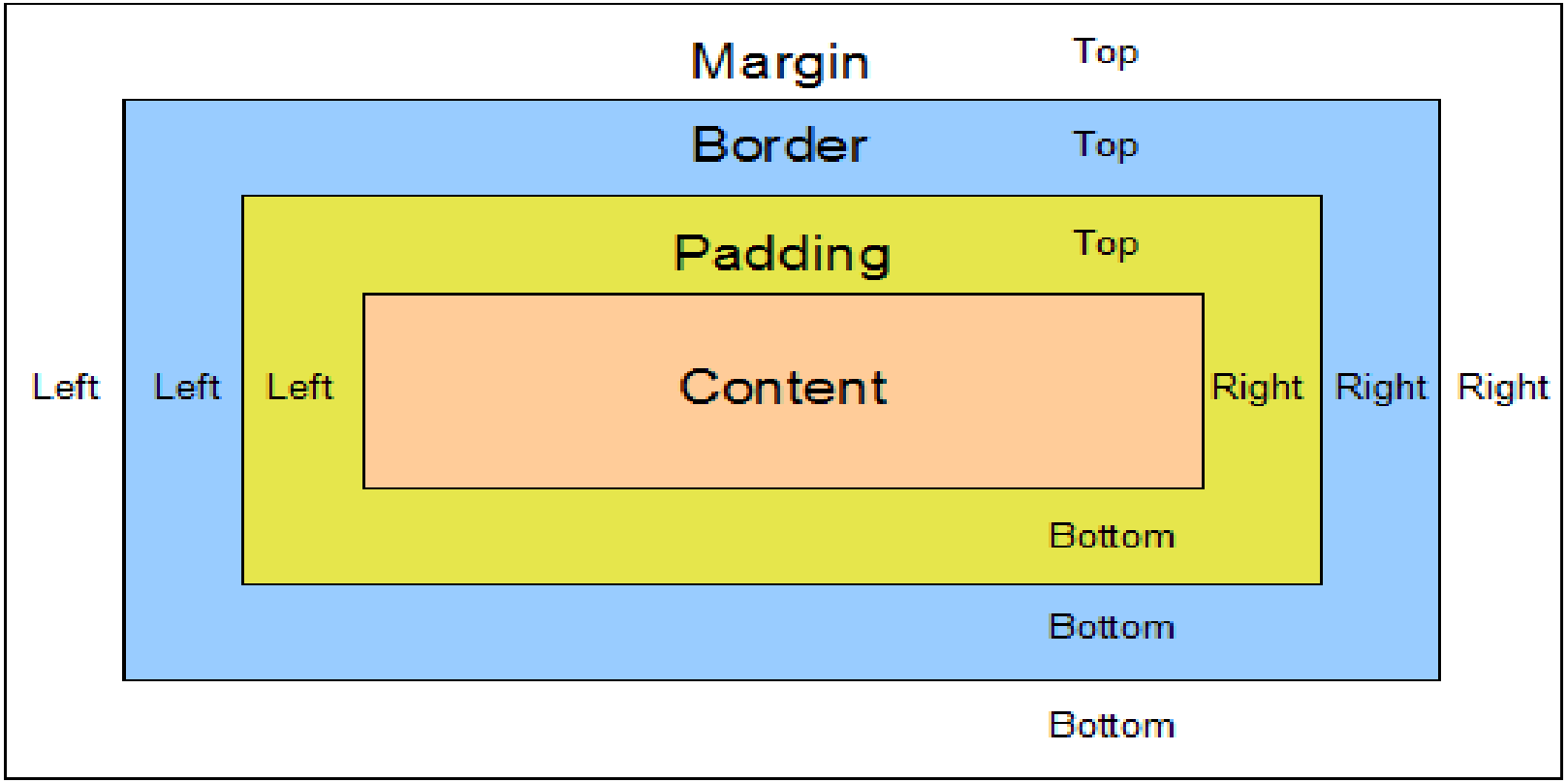
- Tous les éléments HTML sont considérés comme des boîtes
- Les deux grands modes d'affichage des boîtes sont :
  - Le mode ligne (display : *inline*) : <a>,
  - Le mode bloc (display : *block*) : <p>,
- Le positionnement final des éléments dépend de :
  - La taille de la page Web
  - Du remplissage
  - Des bordures
  - Des marges

Marges et bordures

Marges intérieures et extérieures

Pour chaque élément HTML on peut définir :

- L'espacement qui le séparera des autres éléments : `(margin)`
- et les espacements intérieurs dont il peut bénéficier `(padding)`.



Les marges

Propriété	Description
margin	Positionner toutes les marges
margin-bottom	Positionner la marge du bas
margin-left	Positionner la marge de gauche
margin-right	Positionner la marge de droite
margin-top	Positionner la marge de haut

```
h1 {
    margin: 0 0 50px 0;
}

h2 {
    margin: 20px 0 0 0;
}

div {
    width: 300px;
    margin: auto;
    border: 1px solid red;
}
```

```
div {
    border: 1px solid red;
    margin-left: 100px;
    margin-right: 20px;
    margin-top : auto;
    margin-bottom : auto;
}

p.ex1 {
    margin-left: auto;
}
```

Les paddings

Propriété	Description
padding	Positionner tous les paddings
padding-bottom	Positionner le padding du bas
padding-left	Positionner le padding de gauche
padding-right	Positionner le padding de droite
padding-top	Positionner le padding de haut

```
div {
    width: 300px;
    padding: 25px;
}
```

25px : Tous les paddings

```
div {
    padding: 25px 50px;
}
```

25px : Paddings haut et bas
50px : Paddings gauche et droite

```
div {
    padding: 25px 50px 75px;
}
```

25px : Paddings haut
50px : Paddings gauche et droite
75px : Padding bas

La mise en page avec Flexbox

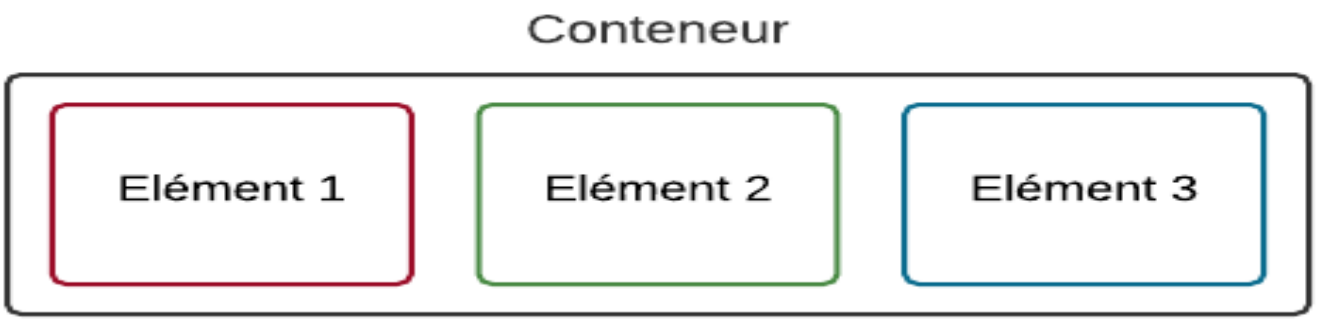
Le principe de la mise en page avec Flexbox est le suivant :

vous définissez ***un conteneur***, et à l'intérieur vous placez plusieurs éléments.

Imaginez un carton dans lequel vous rangez plusieurs objets : c'est le principe !

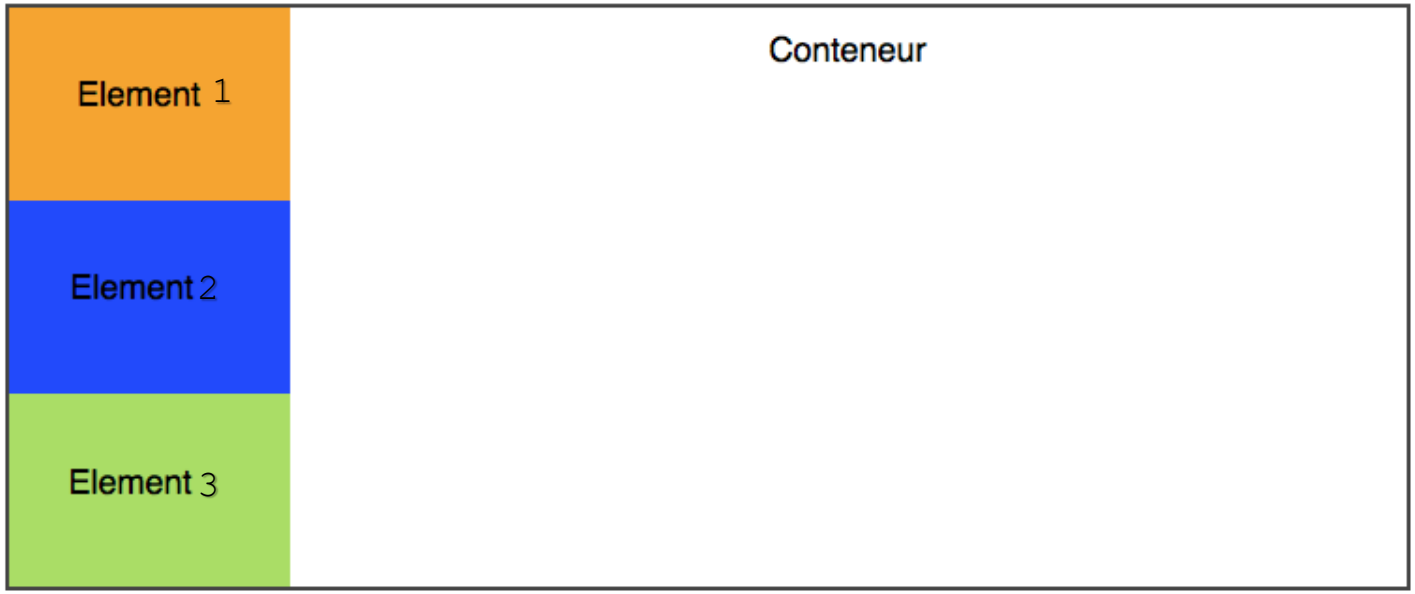
Sur une même page web, vous pouvez sans problème avoir plusieurs conteneurs (plusieurs cartons si vous préférez).

Ce sera à vous d'en créer autant que nécessaire pour obtenir la mise en page voulue.



La mise en page avec Flexbox

```
<div id="conteneur">
    <div class="element">Elément 1</div>
    <div class="element">Elément 2</div>
    <div class="element">Elément 3</div>
</div>
```




```
#conteneur
{
    display: flex;
}
```

La direction

Flexbox nous permet d'agencer ces éléments dans le sens que l'on veut.

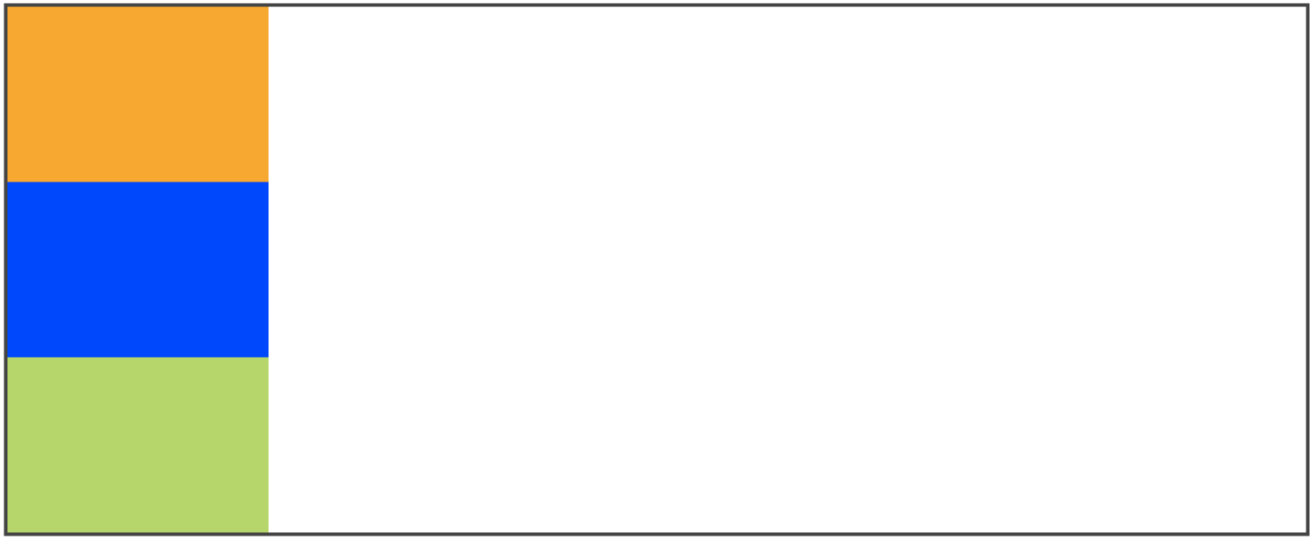
Avec *flex-direction* , on peut les positionner verticalement ou encore les inverser.

Il peut prendre les valeurs suivantes :

- *row* : organisés sur une ligne (par défaut)
- *column* : organisés sur une colonne
- *row-reverse* : organisés sur une ligne, mais en ordre inversé
- *column-reverse* : organisés sur une colonne, mais en ordre inversé

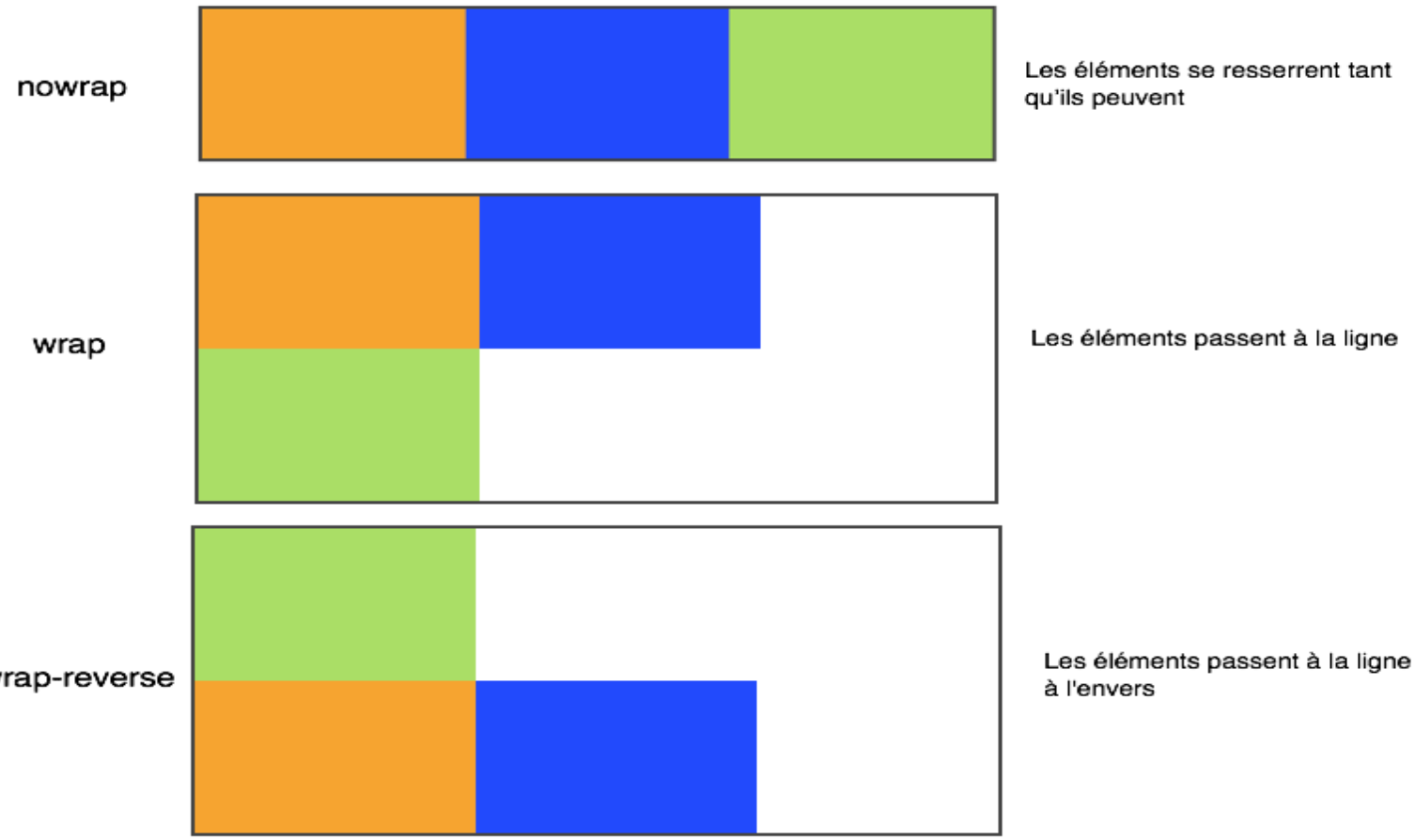
La mise en page avec Flexbox

```
#conteneur {
    display: flex;
    flex-direction: column;
}
```



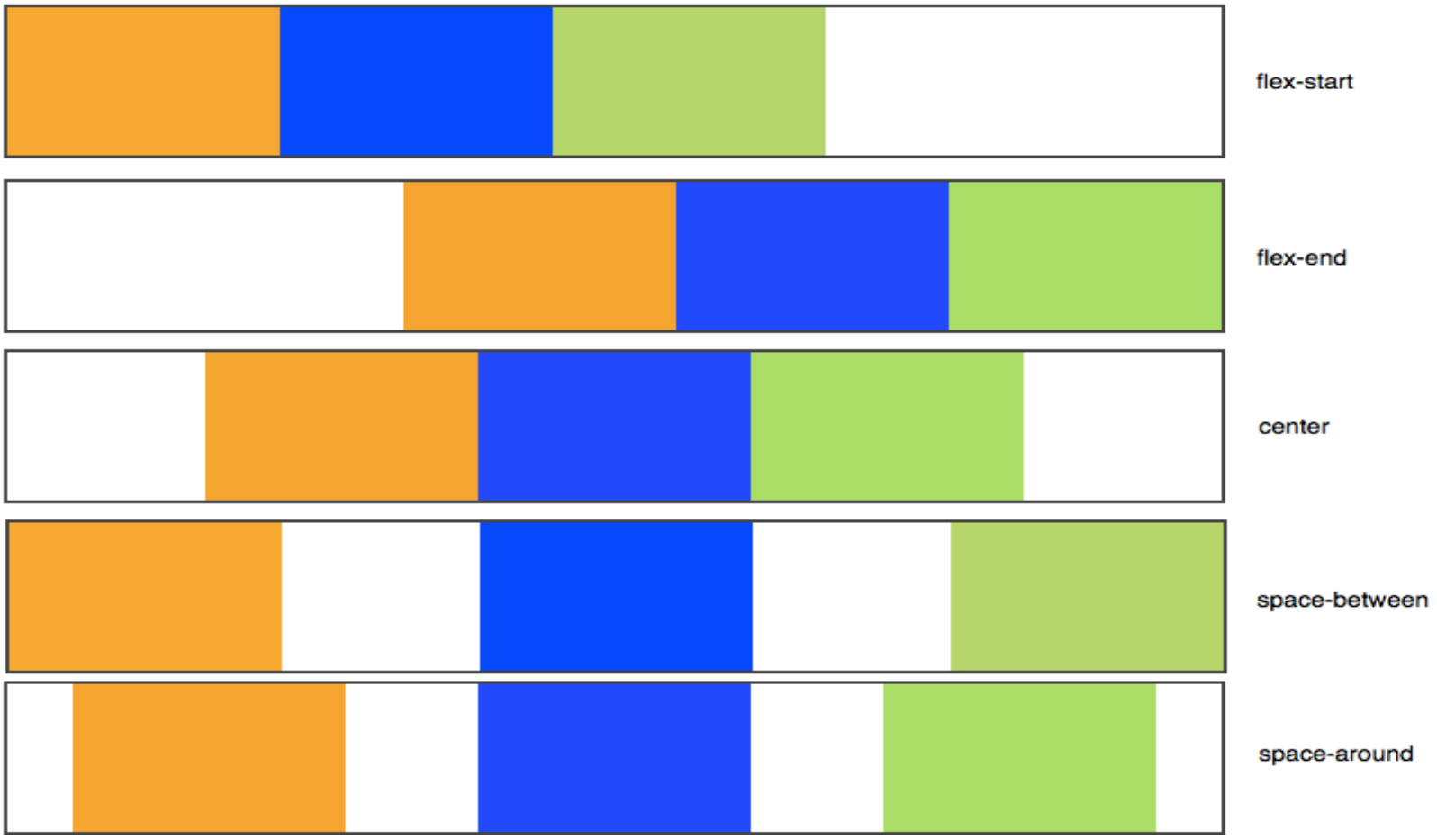
La mise en page avec Flexbox

```
#conteneur {
    display: flex;
    flex-wrap: wrap;
}
```



La mise en page avec Flexbox

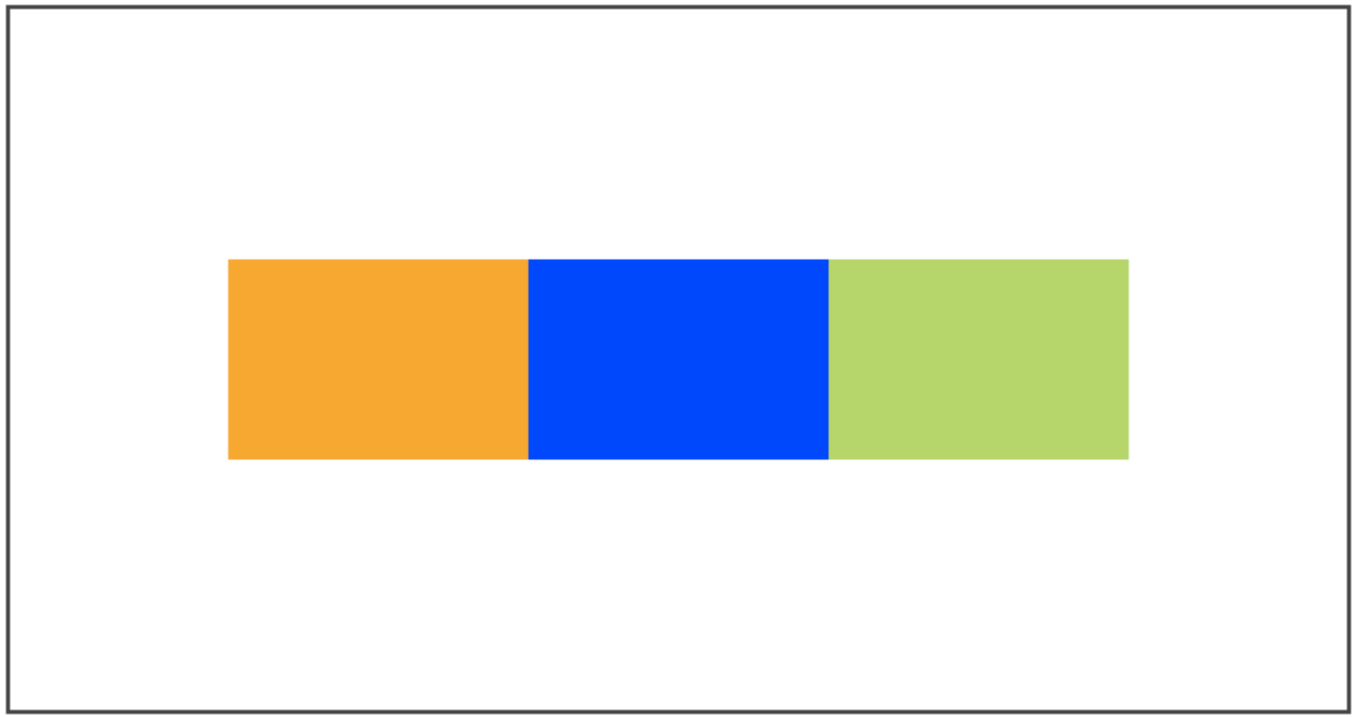
```
#conteneur {
    display: flex;
    justify-content: space-between;
}
```



La mise en page avec Flexbox

```
#conteneur {
    display: flex;
    justify-content: center;
    align-items: center;
}
```

Centrer l'élément dans le conteneur



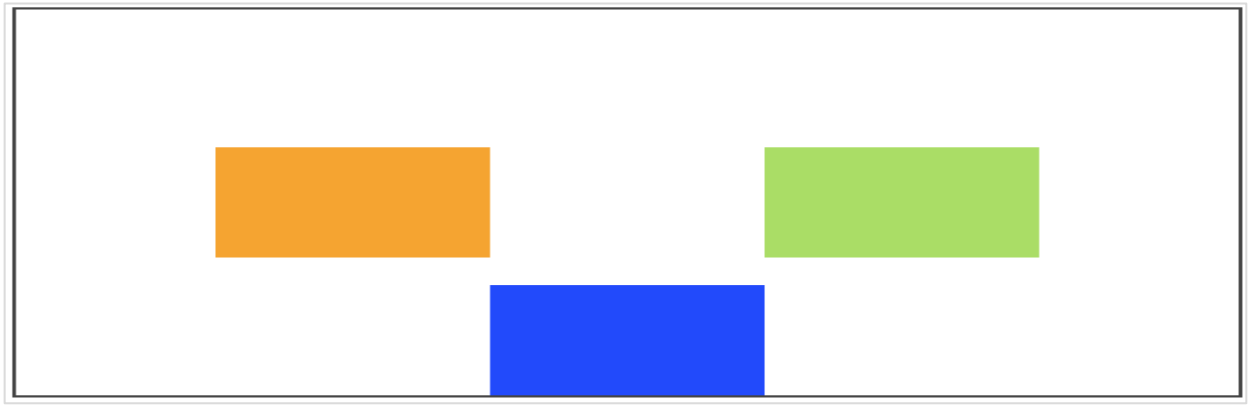
La mise en page avec Flexbox

Aligner un seul élément

Il est possible de faire une exception pour un seul des éléments sur l'axe secondaire avec **align-self** :

```
#conteneur {
    display: flex;
    flex-direction: row;
    justify-content: center;
    align-items: center;
}

.element:nth-child(2) /*        On prend le deuxième bloc élément */
{
    background-color: blue;
    align-self: flex-end; /* Seul ce bloc sera aligné à la fin */
}
```



Propriété : *order*

Pour modifier l'ordre des éléments, on pourra utiliser la proriété : ***order***

```
.element:nth-child(1) {
    order: 3;
}

.element:nth-child(2) {
    order: 1;
}

.element:nth-child(3){
    order: 2;
}
```

Propriété : ***flex***

Agrandir un élément :

```
.element:nth-child(2)
{
    flex: 1;
}
```


```
.element:nth-child(1) {
    flex: 2;
}
.element:nth-child(2) {
    flex: 1;
}
```