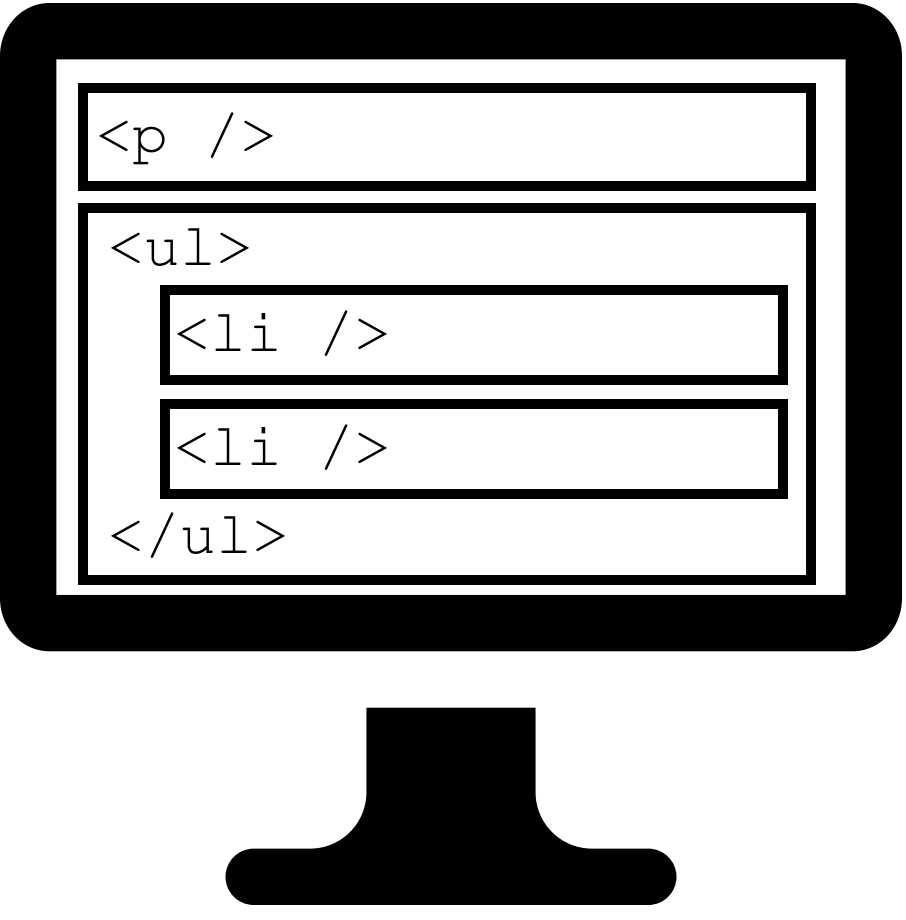
Conception de documents et d'interfaces numériques

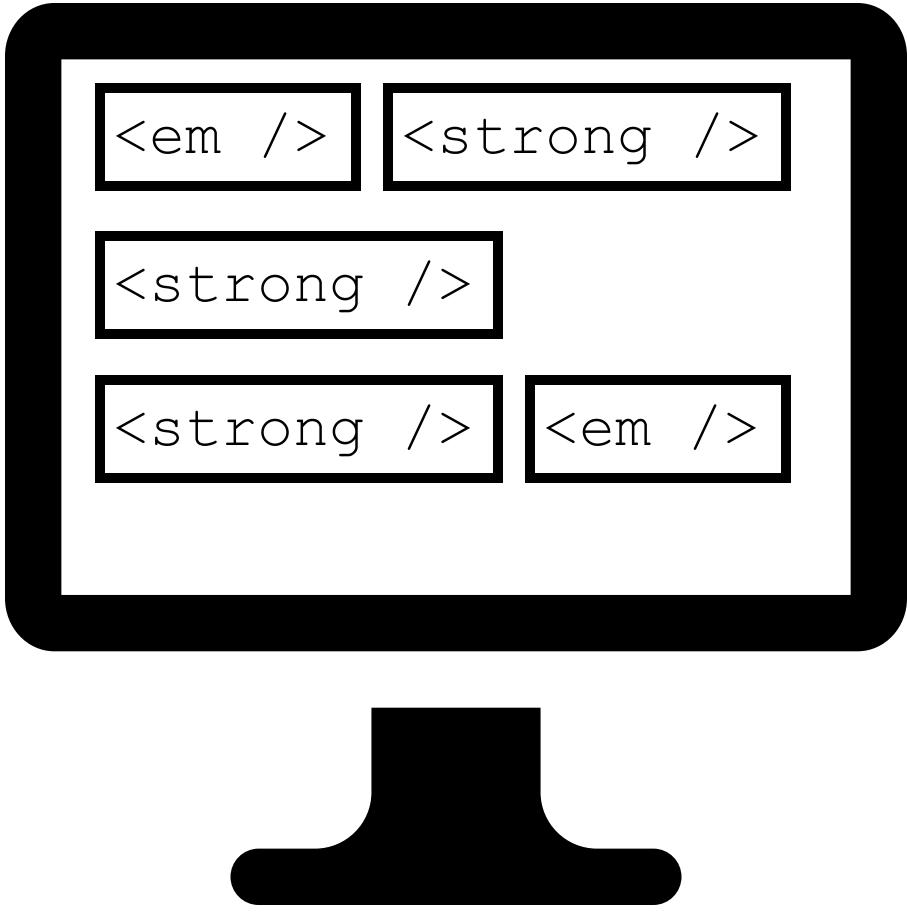
CSS : Box model et mise en page

*Pour CSS, chaque élément HTML est une « boîte » à laquelle on applique différentes propriétés. Ces propriétés déterminent comment et où chaque boîte apparaît dans la page. On appelle « **Box Model** » l'ensemble des règles qui permettent de modifier une boîte.*

Block



Inline



```
<h1>Hello world</h1>

<p>
  Les paragraphes sont de type « block » par défaut.
  En revanche, les éléments <em>em</em> et
  <strong>strong</strong> sont de type « inline »
  par défaut.
</p>

h1,
p {
  background-color: #8395a7;
}

em,
strong {
  background-color: #0abde3;
}
```

Hello world

Les paragraphes sont de type « block » par défaut. En revanche, les éléments *em* et **strong** sont de type « inline » par défaut.

- *Les éléments « block » apparaissent toujours les uns sous les autres*
- Les éléments « block » font automatiquement 100% de la largeur de leur élément parent
- La hauteur des éléments « block » dépend de leur contenu
- La largeur des éléments « inline » dépend de leur contenu

Passer un élément « inline » en « block », et inversement

```
h1,
p {
  background-color: #8395a7;
}

em,
strong {
  background-color: #0abde3;
  display: block;
}
```

Hello world

Les paragraphes sont de type « block » par défaut. En revanche, les éléments

em

et

strong

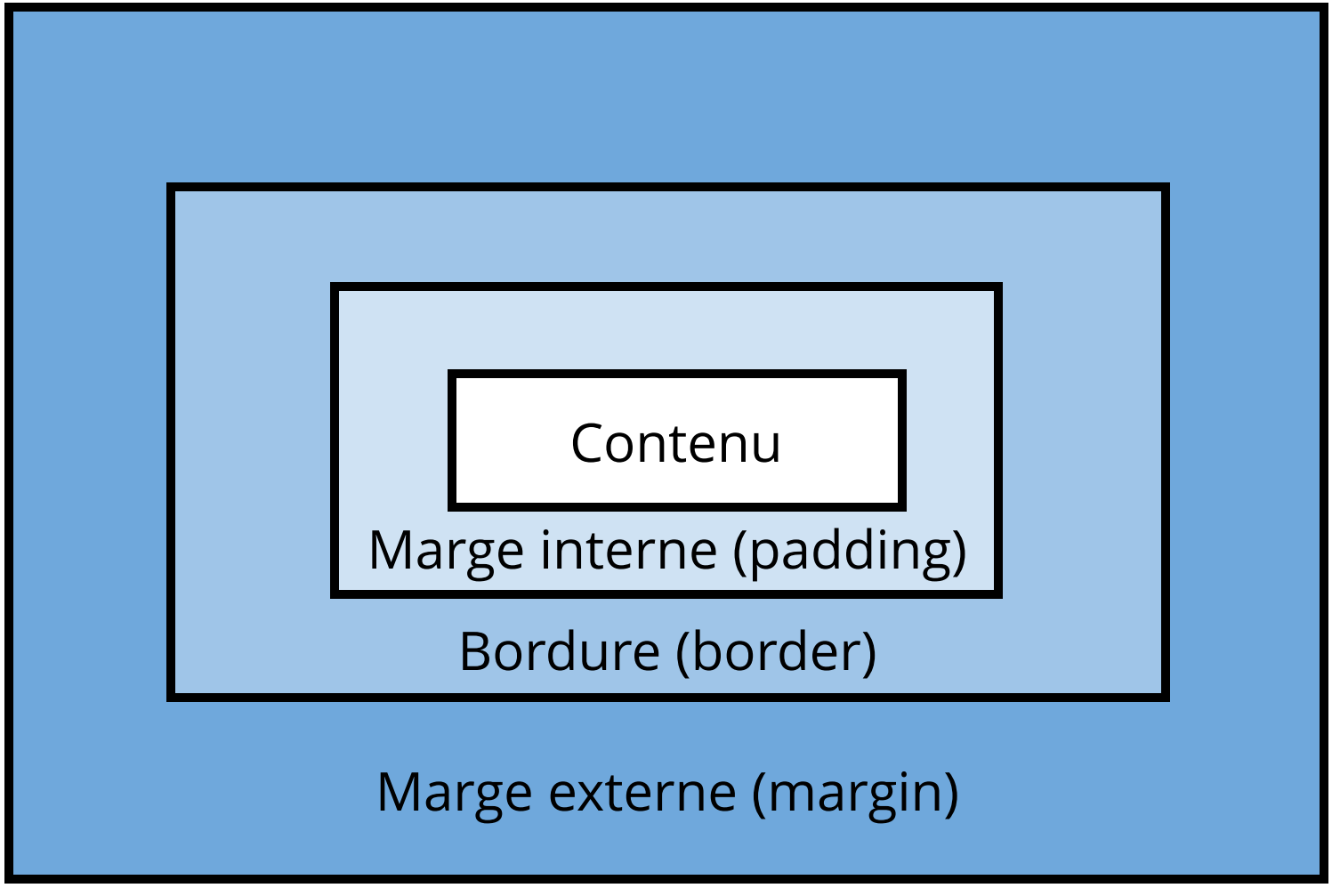
sont de type « inline » par défaut.

```
h1,
p {
  background-color: #8395a7;
}

em,
strong {
  background-color: #0abde3;
  display: inline;
}
```

Hello world

Les paragraphes sont de type « block » par défaut. En revanche, les éléments *em* et **strong** sont de type « inline » par défaut.

Contenu, marges internes, marges externes, bordures...


Contenu
Marge interne (padding)
Bordure (border)
Marge externe (margin)

Marges intérieures (padding)

```
h1 {
  padding-top: 50px;
  padding-right: 50px;
  padding-bottom: 50px;
  padding-left: 50px;
}
```

padding - MDN

Hello world

Les paragraphes sont de type « block » par défaut. En revanche, les éléments *em* et **strong** sont de type « inline » par défaut.


Hello world
h1 | 304 × 178
Les paragraphes sont de type « block » par défaut.
éléments em et strong sont de type « inline » par défaut.

```
h1 {
  padding-top: 50px;
  padding-right: 10px;
  padding-bottom: 30px;
  padding-left: 20px;
}
```

padding - MDN


Hello world
h1 | 304 × 119
Les paragraphes sont de type « block » par défaut.
éléments em et strong sont de type « inline » par défaut.

```
h1 {
   padding: 50px 10px 30px 20px;
}
```

top right bottom left

padding - MDN

```
h1 {
    padding: 50px 10px 30px;
}
```

top

horizontal (left / right)

bottom

padding - MDN

```
h1 {
    padding: 50px 10px;
}
```

vertical (top / bottom)

horizontal (left / right)

padding - MDN

```
h1 {
    padding: 50px;
}
```

vertical
(top / bottom)
+ horizontal
(left / right)

padding - MDN

Bordures (border)

Les bordures ont 3 aspects :

- *Une épaisseur : border-width*
- *Un style (trait plein, ligne discontinue, pointillés) : border-style*
- *Une couleur : border-color*

```
h1 {
  border-style: solid;
  border-width: 5px;
  border-color: red;
}
```

border - MDN

Hello world

Les paragraphes sont de type « block » par défaut. En revanche, les éléments *em* et **strong** sont de type « inline » par défaut.

```
h1 {
    border: 5px solid red;
}
```

width style color

border - MDN

Marges externes (margin)

```
h1 {
   margin-bottom: 50px;
}
```

margin - MDN

Hello world

Les paragraphes sont de type « block » par défaut. En revanche, les éléments *em* et **strong** sont de type « inline » par défaut.

Quelle est la différence entre margin et padding ?

Le padding est affecté par le background, pas la margin

```
h1 {
    margin-bottom: 50px;
    padding: 20px;
}
```

margin - MDN

Hello world

Les paragraphes sont de type « block » par défaut. En revanche, les éléments *em* et **strong** sont de type « inline » par défaut.

Le padding est inclut dans la zone de click, pas la margin

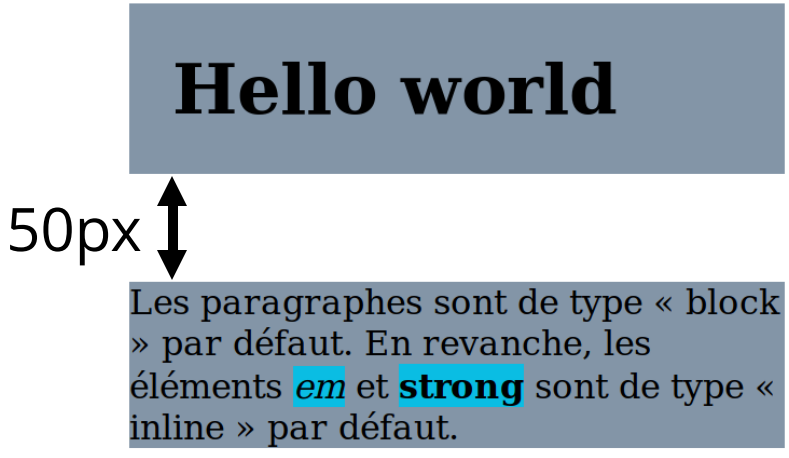
```
<a href="#">
  Pouet
</a>
```

```
a {
  display: block;
  padding: 20px;
  margin: 20px;
  background-color: #0abde3;
}
```

Les margin verticales fusionnent entre elles

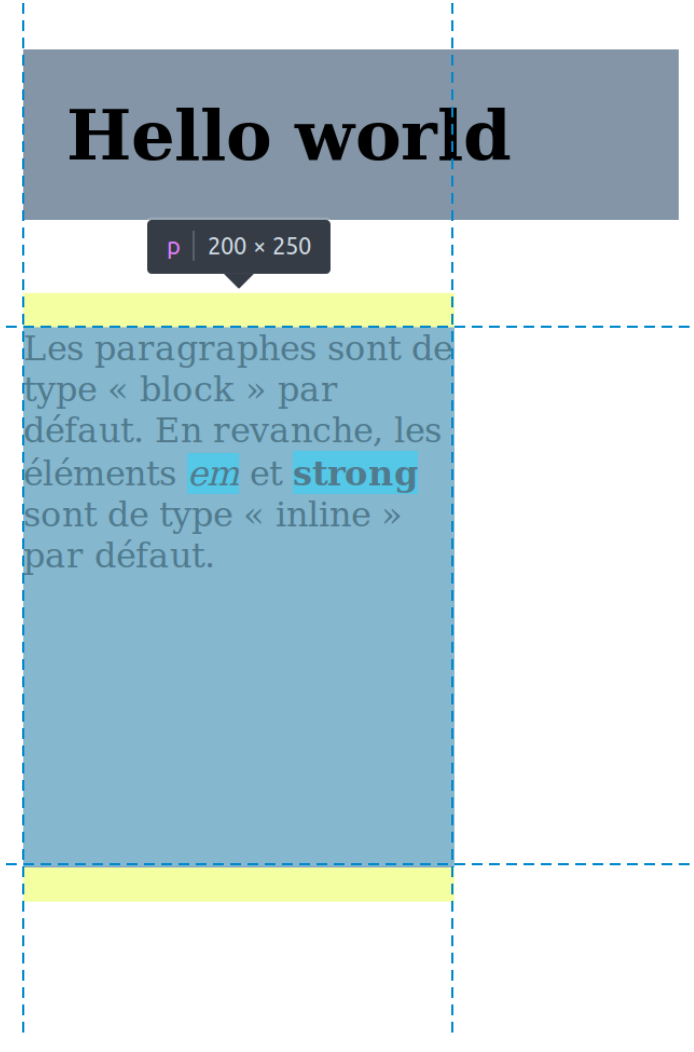
```
h1 {
  margin-bottom: 50px;
}

p {
  margin-top: 50px;
}
```

Mastering margin collapsing - MDN


Hello world
50px
Les paragraphes sont de type « block » par défaut. En revanche, les éléments em et strong sont de type « inline » par défaut.

Donner des dimensions précises à un élément

```
p {
  background-color: #8395a7;
  width: 200px;
  height: 250px;
}
```


Hello world
p | 200 × 250
Les paragraphes sont de type « block » par défaut. En revanche, les éléments em et strong sont de type « inline » par défaut.

```
p {
  background-color: #8395a7;
  width: 200px;
  height: 80px;
}
```

Hello world

Les paragraphes sont de type « block » par défaut. En revanche, les éléments *em* et **strong** sont de type « inline » par défaut.

```
p {
  background-color: #8395a7;
  width: 200px;
  height: 80px;
  overflow: hidden;
}
```

Hello world

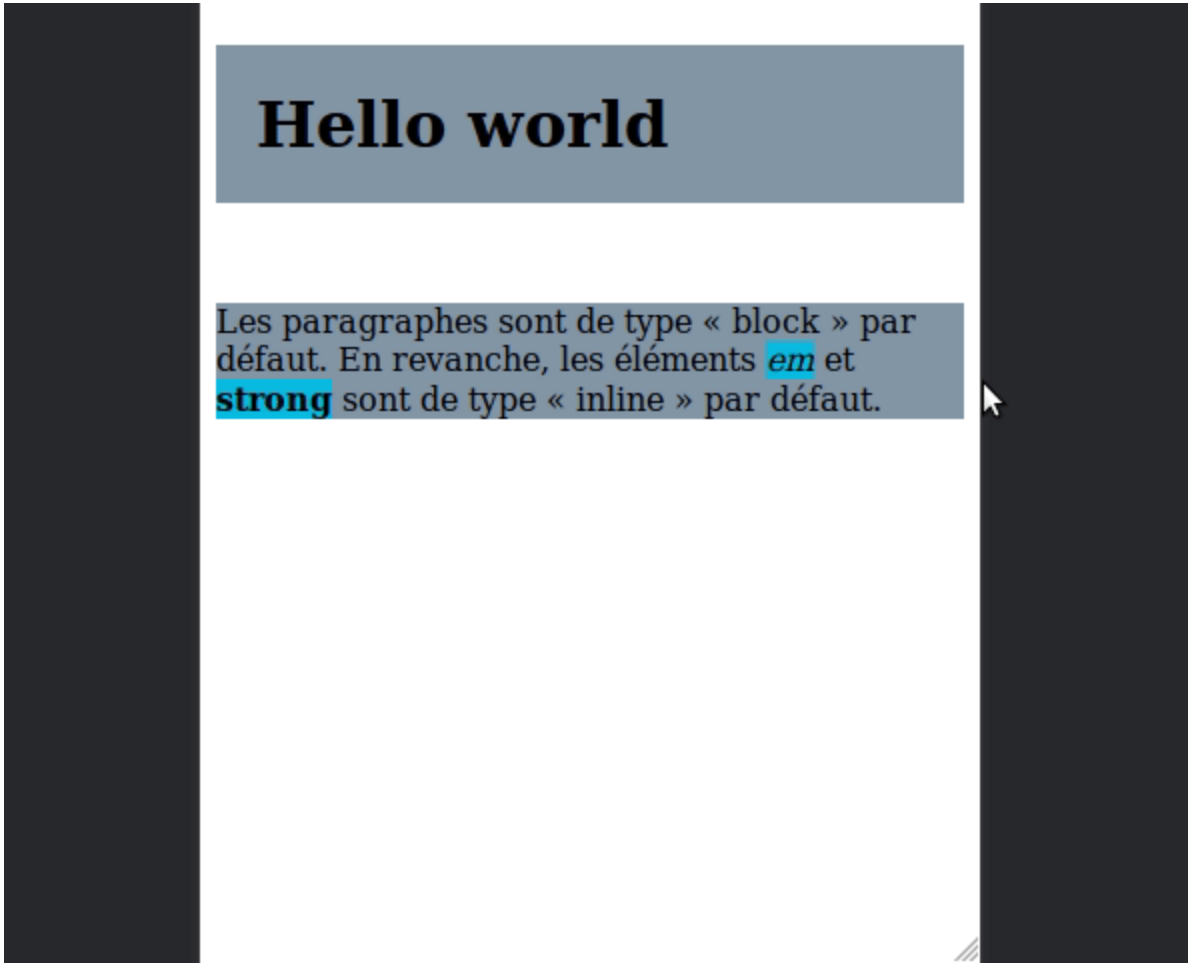
Les paragraphes sont de type « block » par défaut. En revanche, les éléments *em* et **strong**

```
p {
  background-color: #8395a7;
  width: 200px;
  height: 80px;
  overflow: scroll;
}
```

Hello world

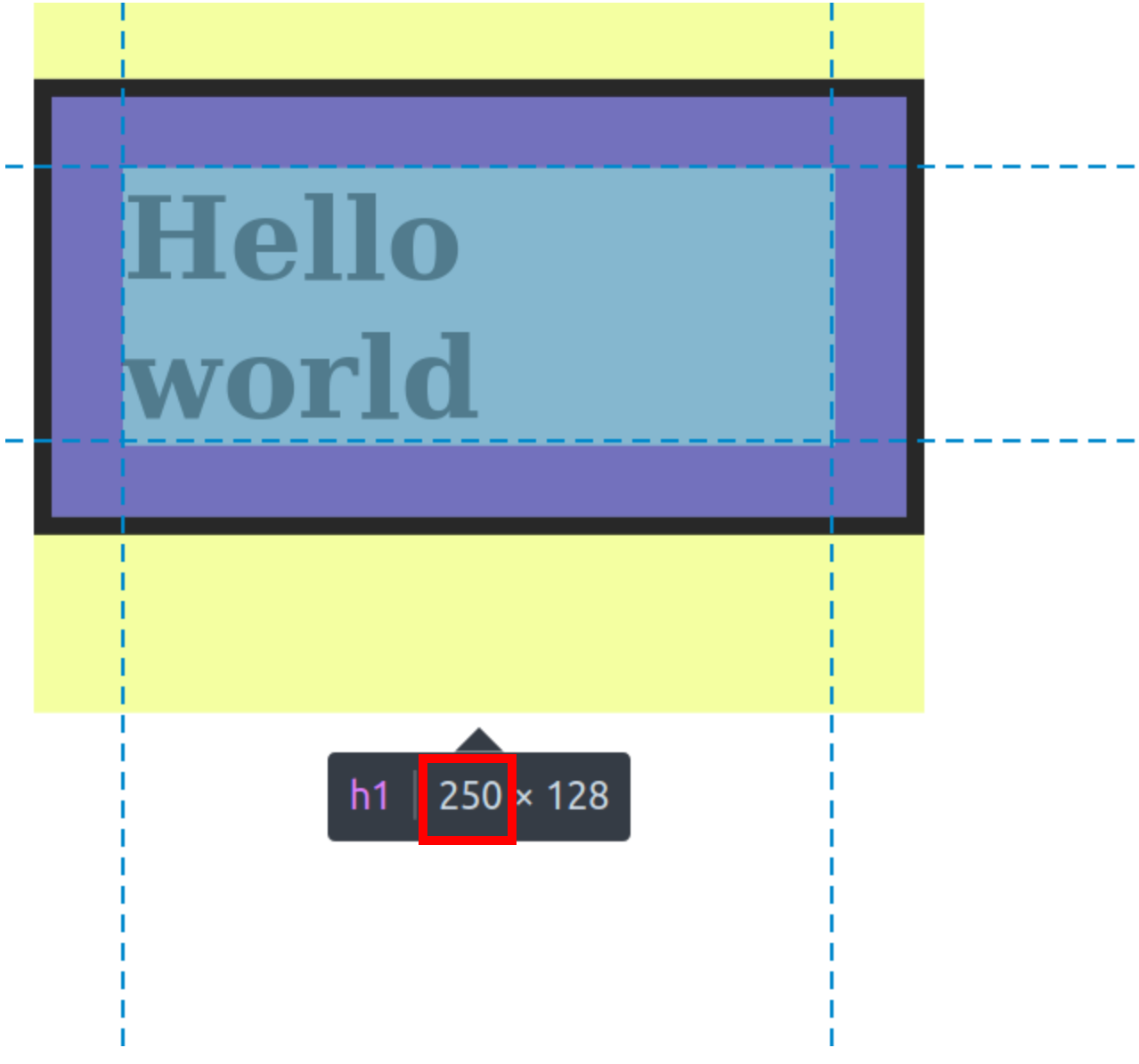
Les paragraphes sont de type « block » par défaut. En revanche, les éléments *em* et **strong**

```
p {
    max-width: 400px;
}
```


Hello world
Les paragraphes sont de type « block » par défaut. En revanche, les éléments em et strong sont de type « inline » par défaut.

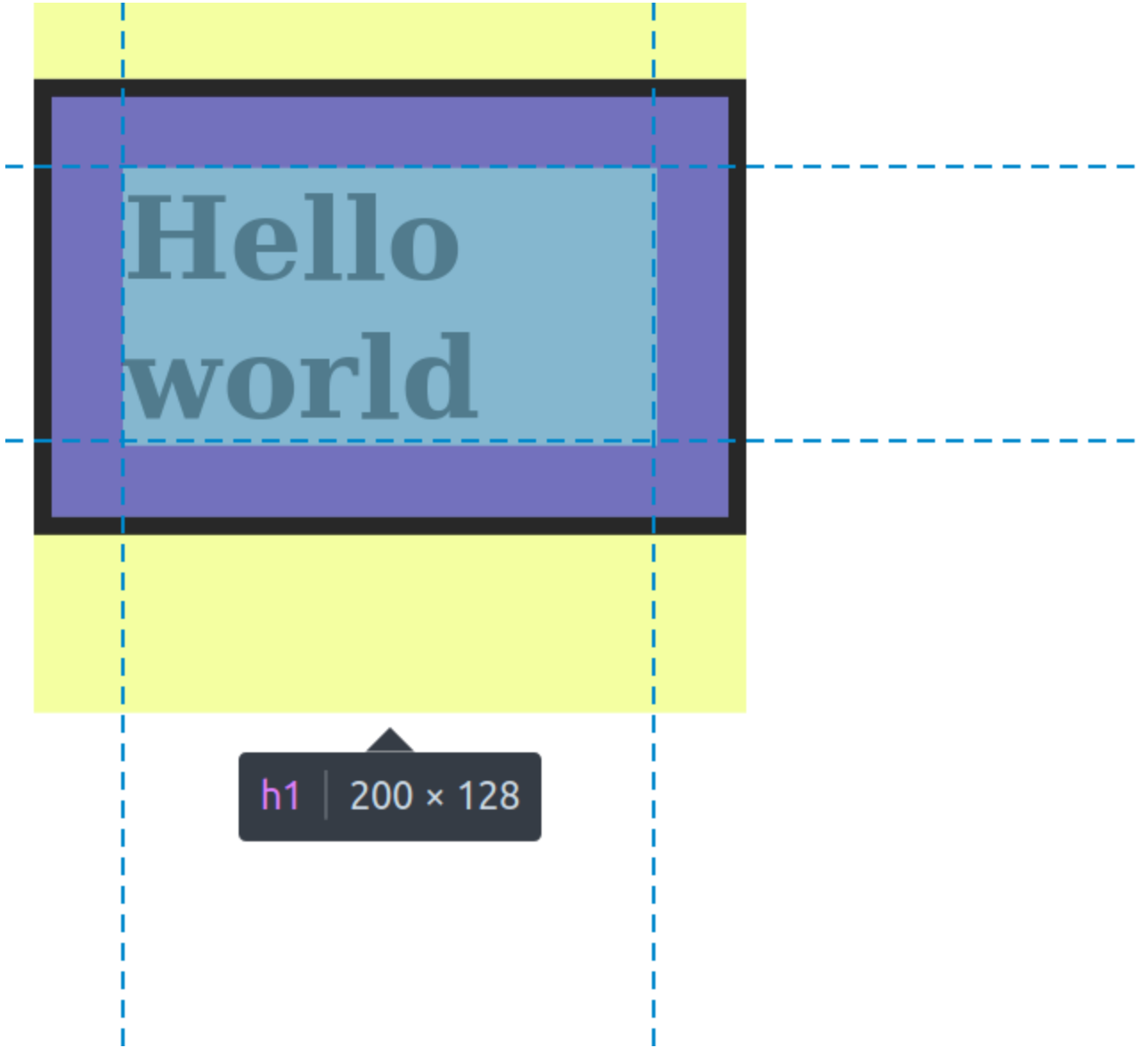
Par défaut, width réelle = width + padding + border

```
h1 {
   width: 200px;
   padding: 20px;
   border: 5px solid black;
}
```


Hello
world
h1
250 × 128

```
h1 {
  width: 200px;
  padding: 20px;
  border: 5px solid black;
  box-sizing: border-box;
}
```

box-sizing - MDN


Hello
world
h1
200 × 128

```
* {
  box-sizing: border-box;
}
```

Aligner les boîtes

Par défaut, tous les éléments sont alignés à gauche. Mais on peut évidemment modifier l'alignement.

```
h1 {
   width: 200px;
   margin-left: auto;
}
```

Hello world

Les paragraphes sont de type « block » par défaut. En revanche, les éléments *em* et **strong** sont de type « inline » par défaut.

```
h1 {
  width: 200px;
  margin-left: auto;
  margin-right: auto;
}
```

Hello world

Les paragraphes sont de type « block » par défaut. En revanche, les éléments *em* et **strong** sont de type « inline » par défaut.

Par défaut, l'élément body possède une margin. Celle-ci est rarement souhaitable. Généralement, on annule directement cette marge.

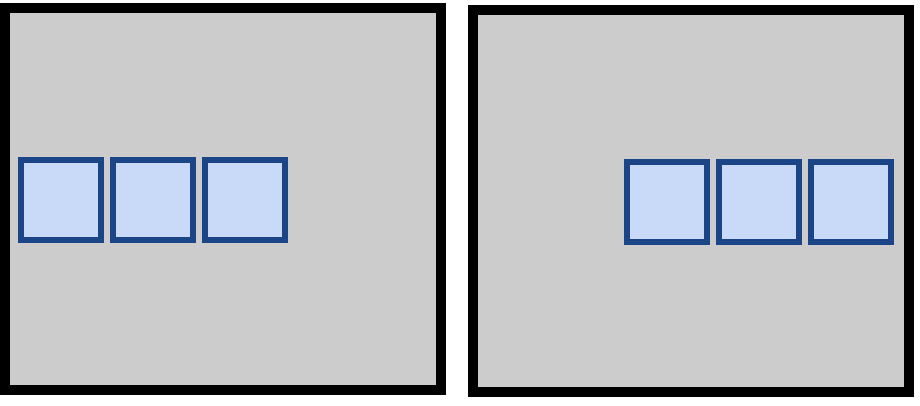
```
body {
    margin: 0;
}
```

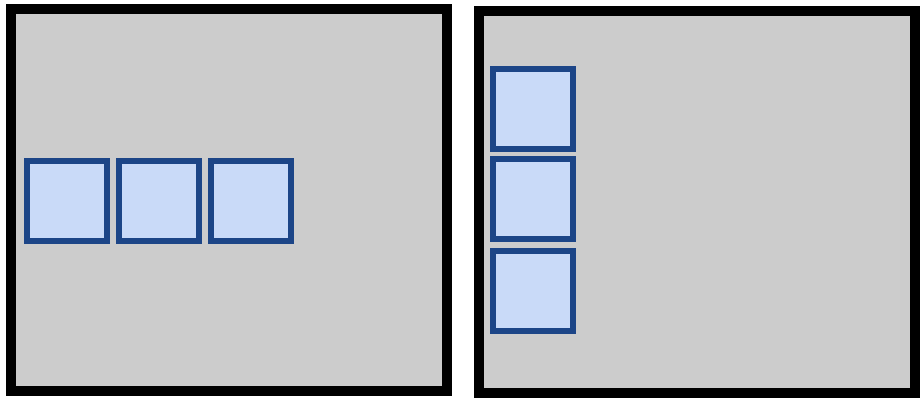
```
* {
  box-sizing: border-box;
}

body {
  margin: 0;
}
```

Flexbox

Alignement

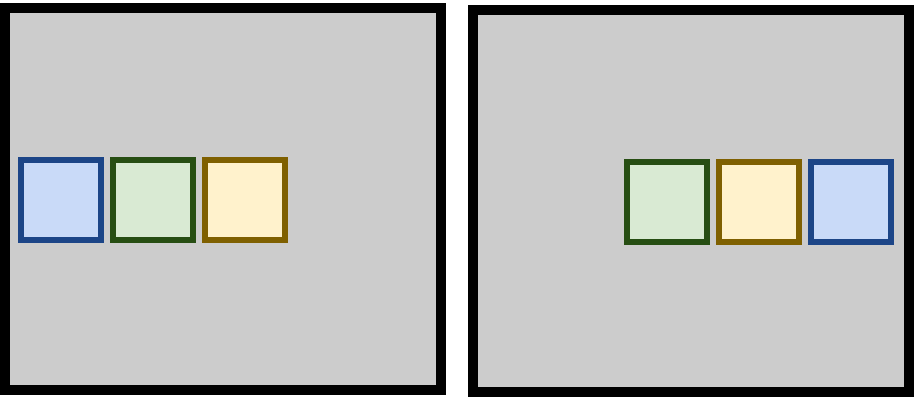
Direction

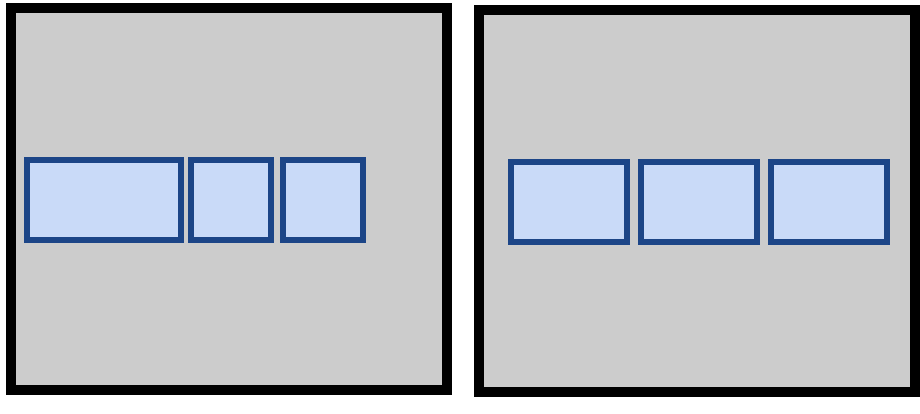




Ordre

Taille



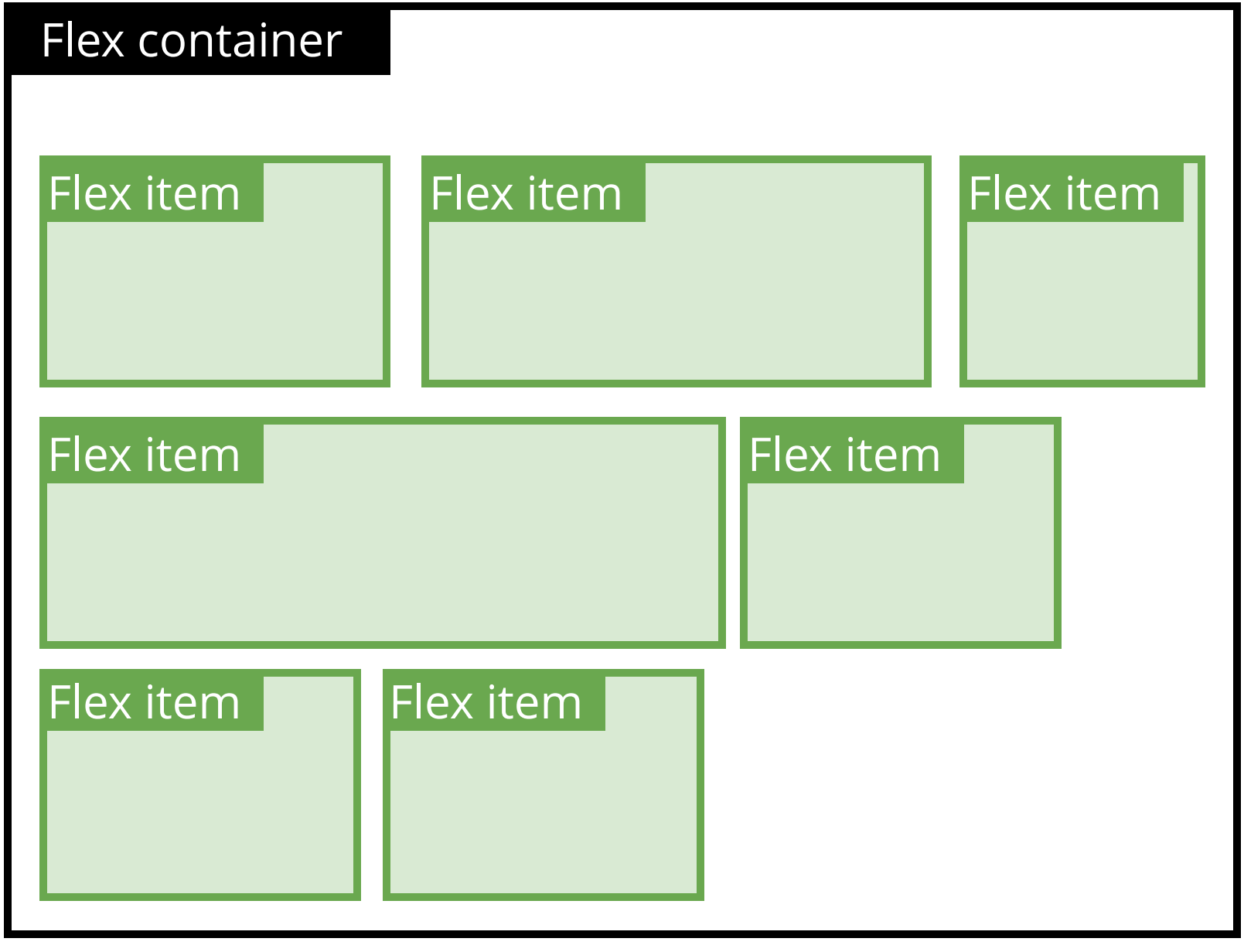


```
<div class="container">
  <div class="item">1</div>
  <div class="item">2</div>
  <div class="item">3</div>
</div>
```

1
2
3

```
.container {
  display: flex;
}
```

123


Flex container
Flex item
Flex item
Flex item
Flex item
Flex item
Flex item
Flex item

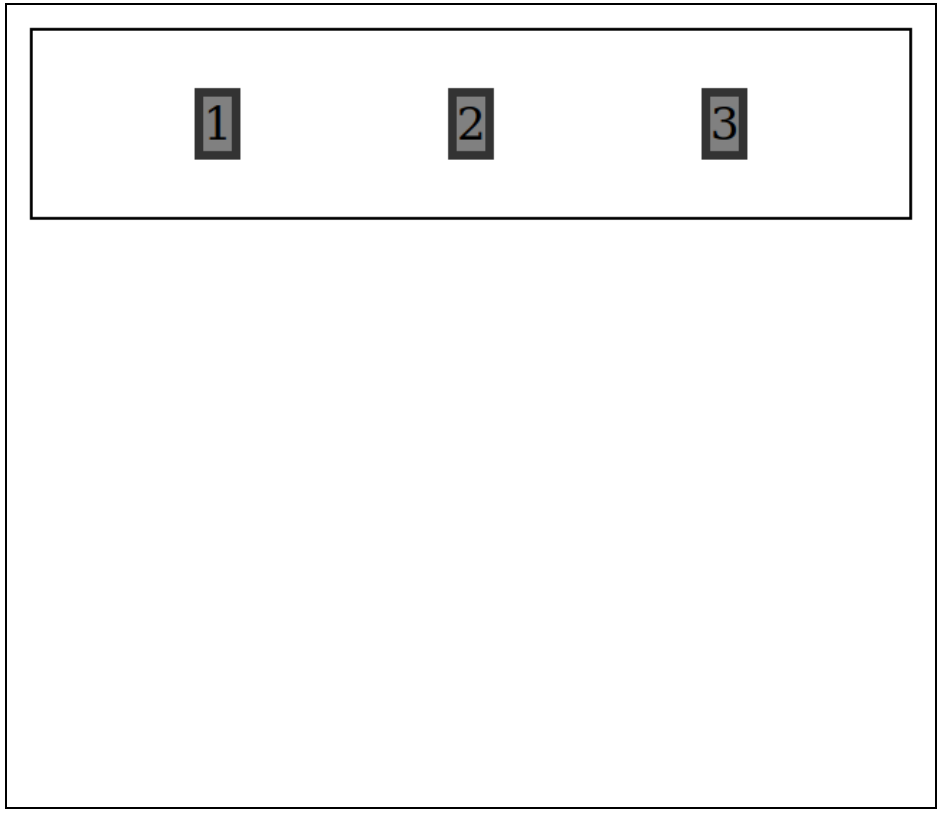
Aligner des éléments horizontalement avec flexbox

```
.container {
  justify-content: flex-start;
}
```

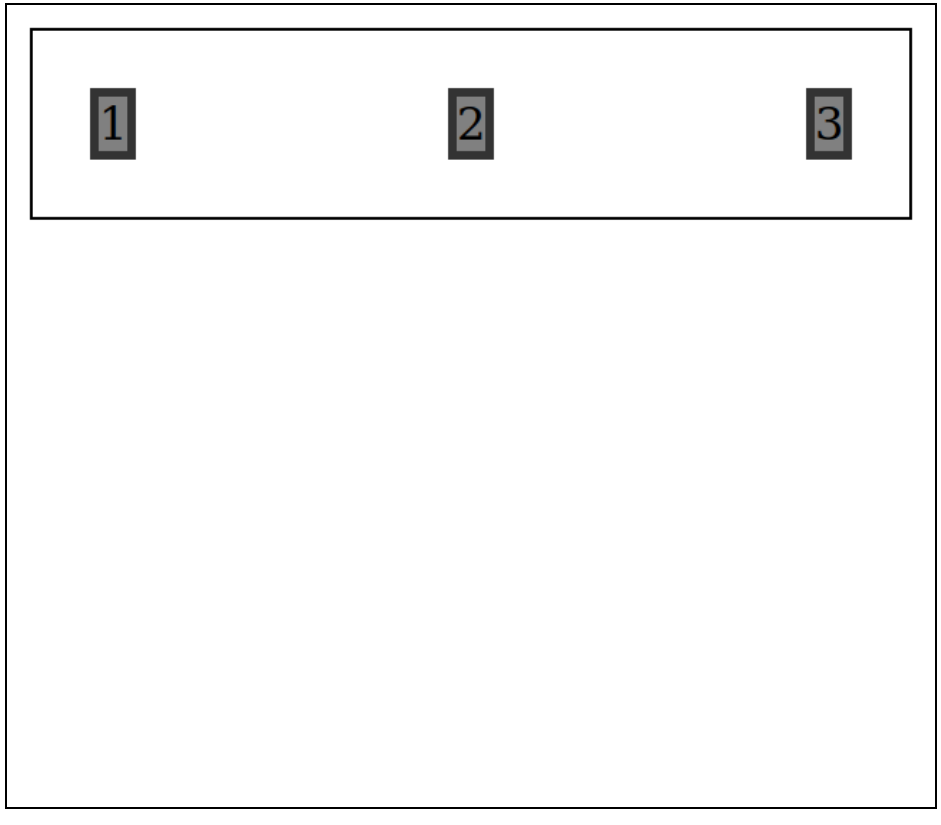
```
.container {
  justify-content: center;
}
```

```
.container {
  justify-content: flex-end;
}
```

```
.container {
  justify-content: space-around;
}
```

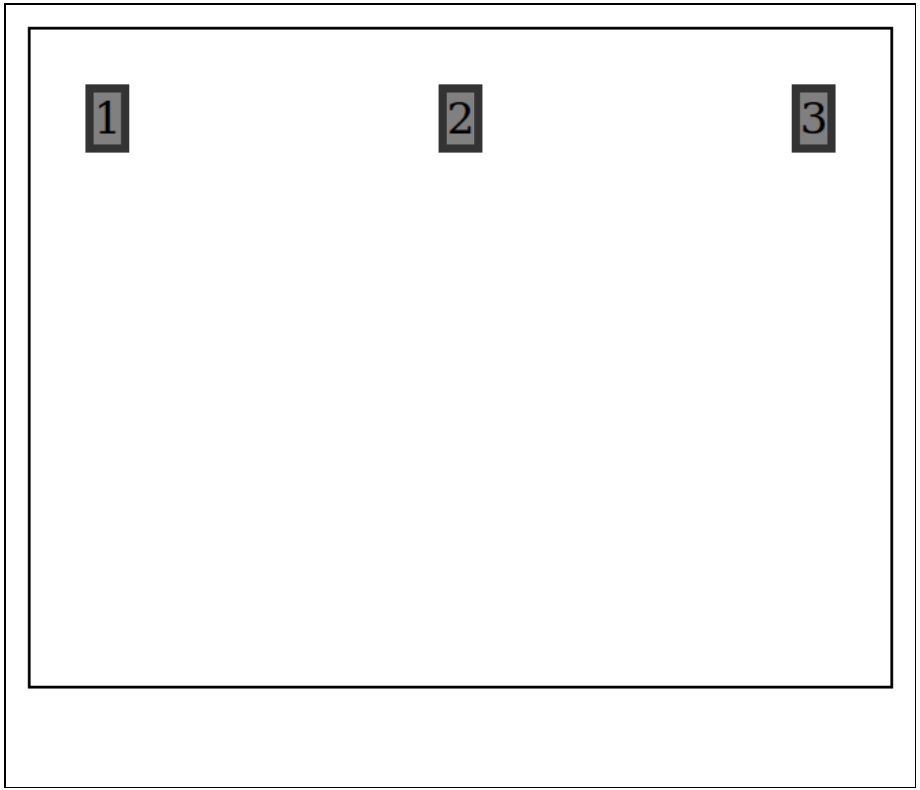


```
.container {
  justify-content: space-between;
}
```

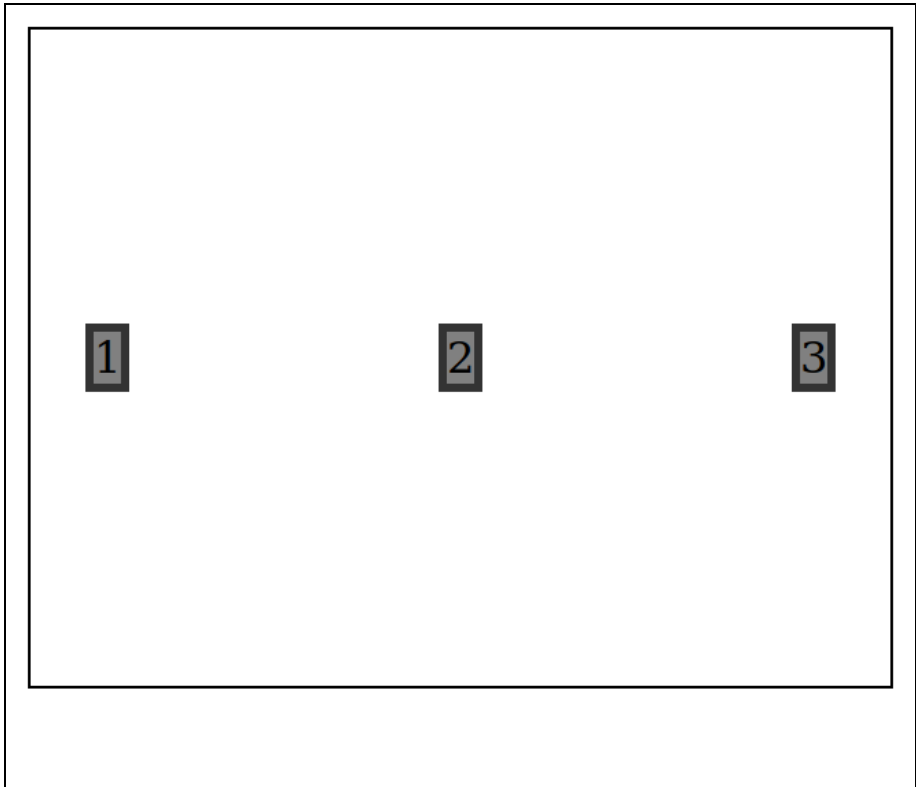


Aligner des éléments verticalement avec flexbox

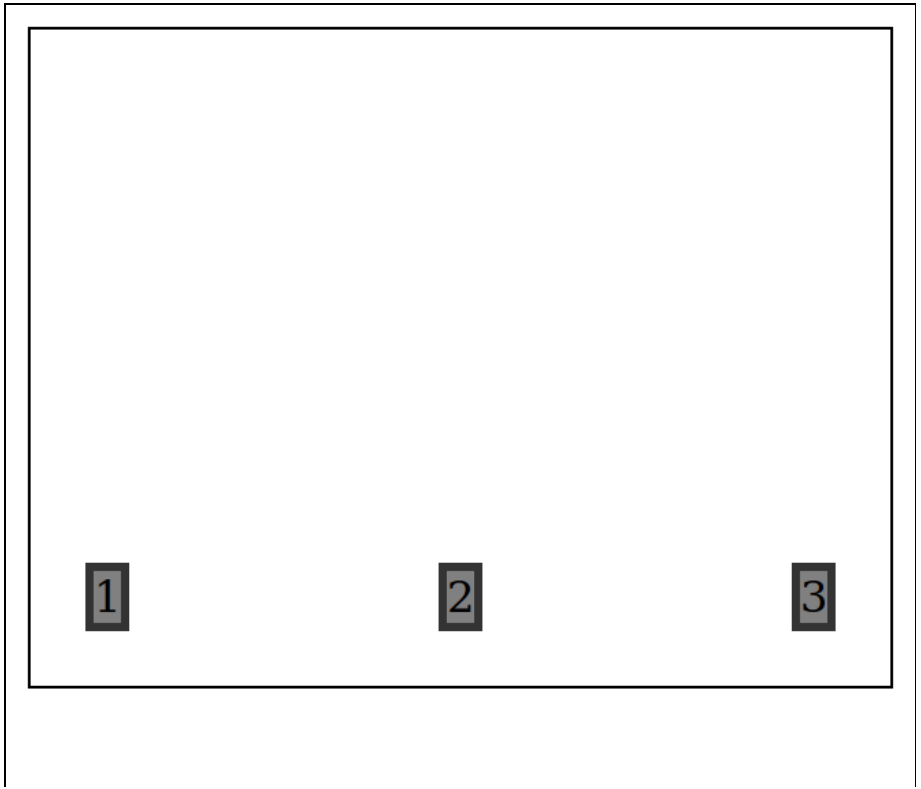
```
.container {
  height: 200px;
  justify-content: space-between;
  align-items: flex-start;
}
```



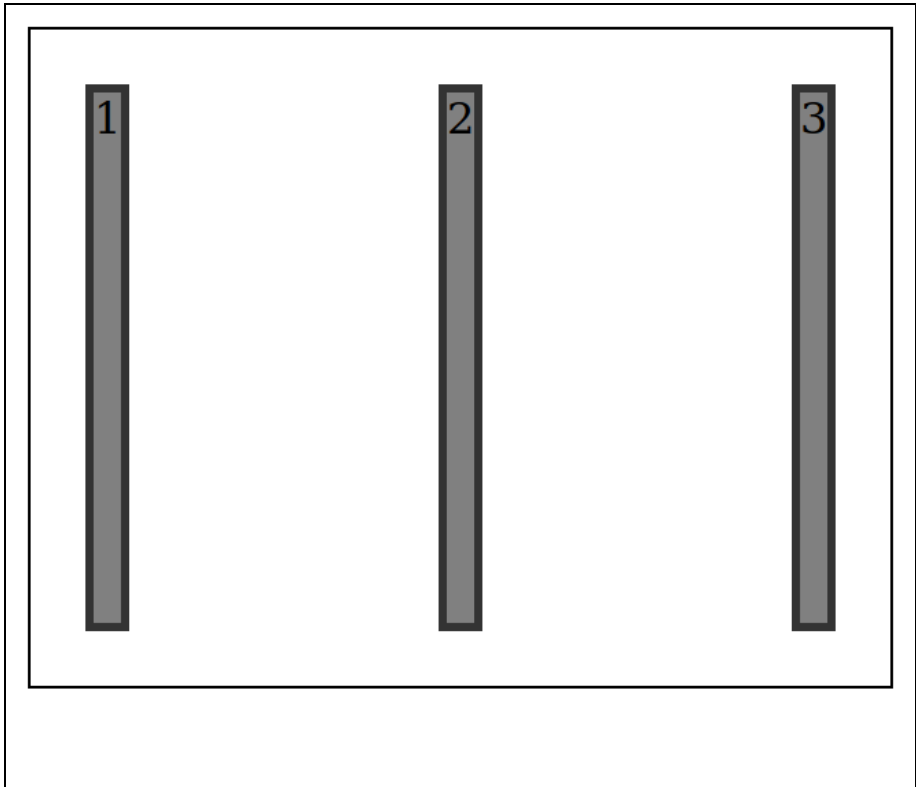
```
.container {
  height: 200px;
  justify-content: space-between;
  align-items: center;
}
```



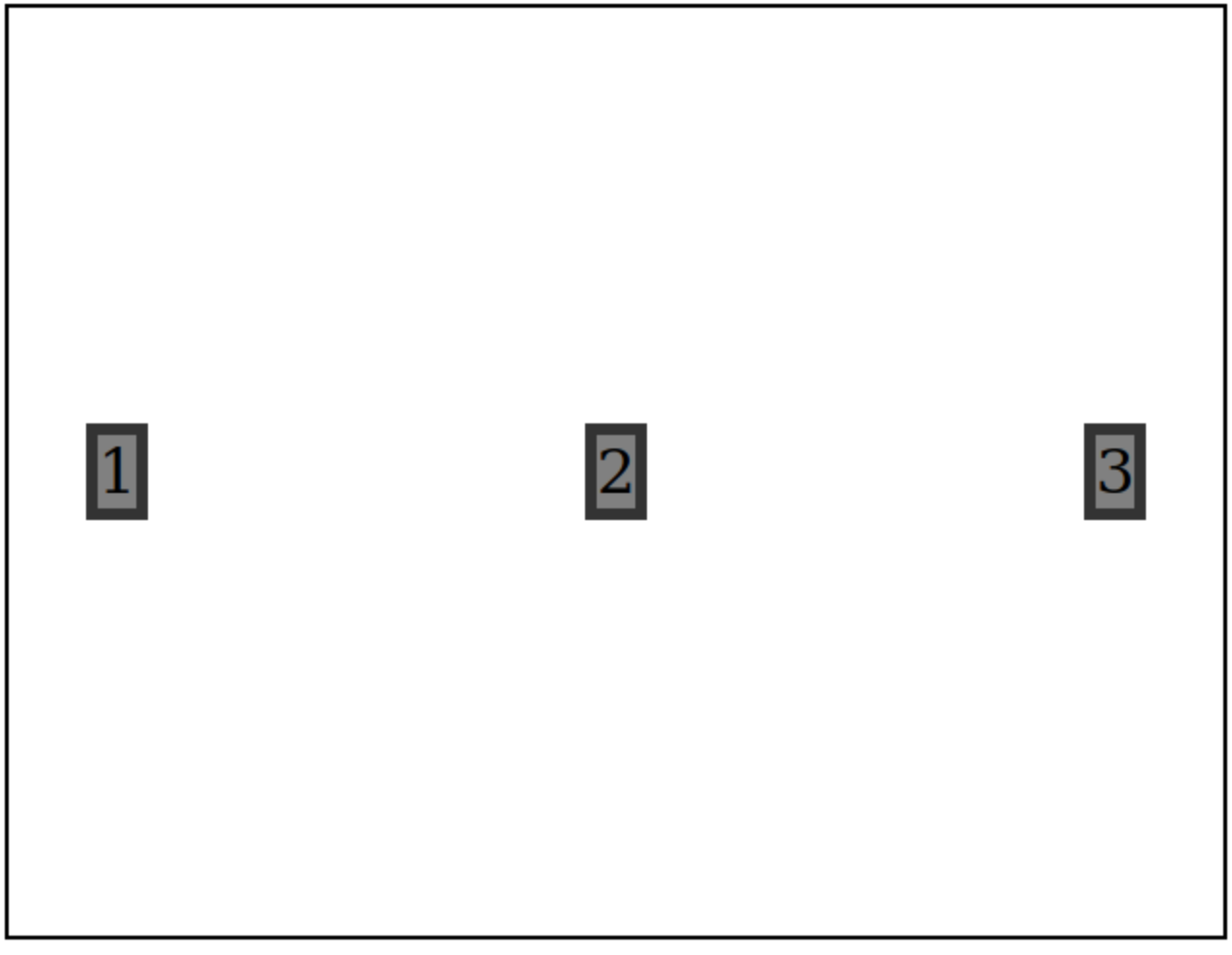
```
.container {
  height: 200px;
  justify-content: space-between;
  align-items: flex-end;
}
```



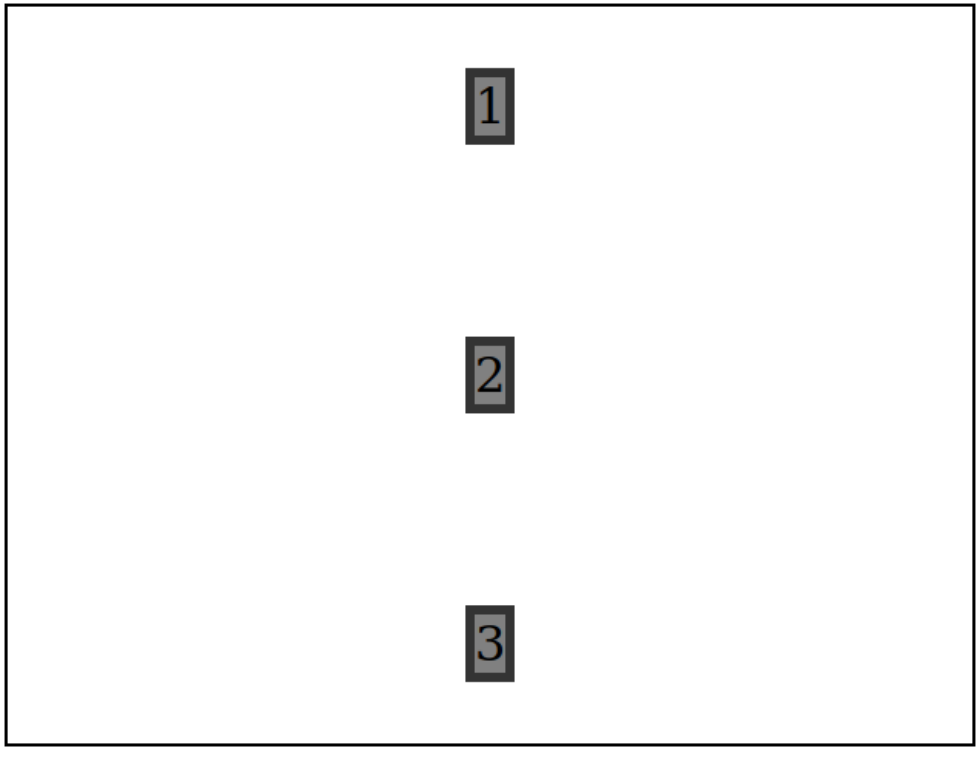
```
.container {
  height: 200px;
  justify-content: space-between;
  align-items: stretch;
}
```



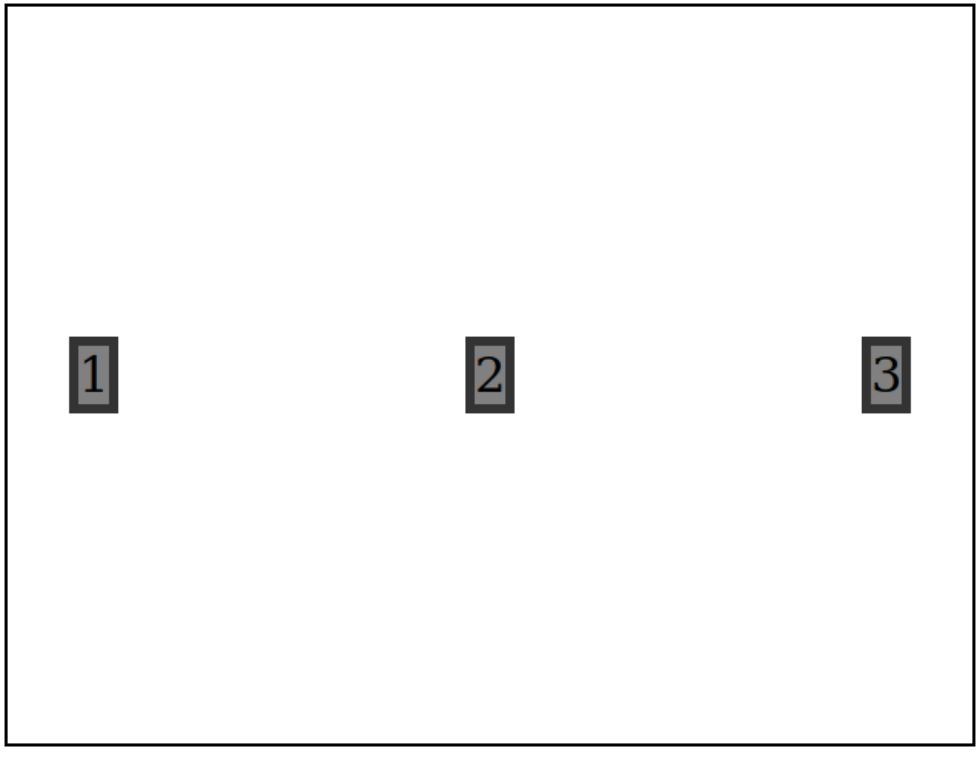
Définir la direction


1
2
3

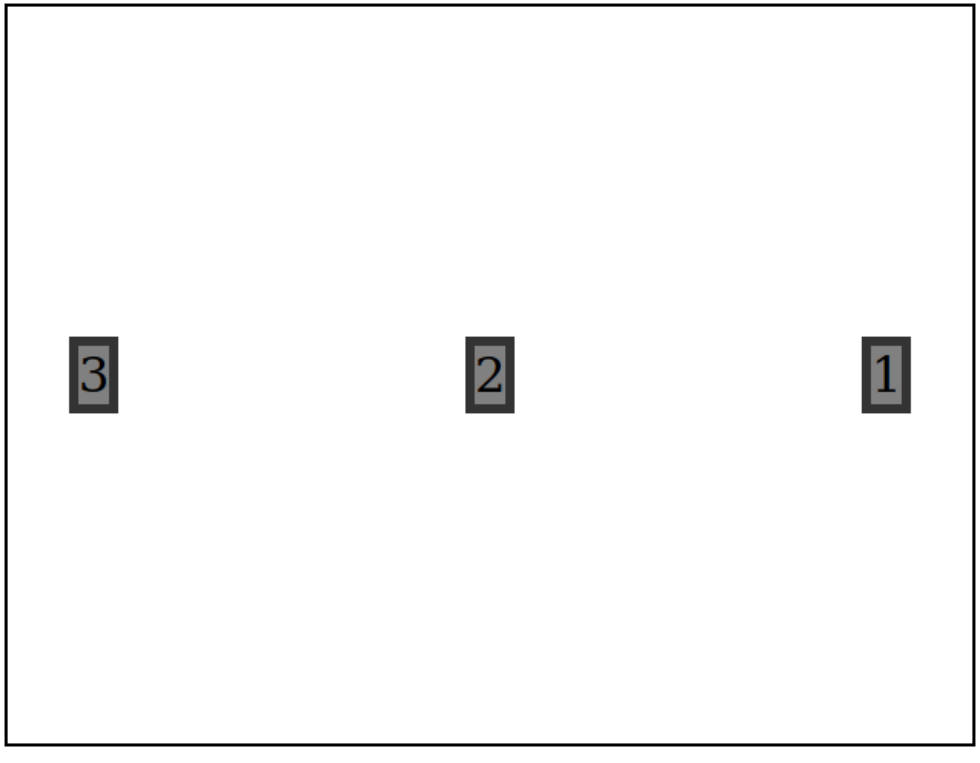
```
.container {
  flex-direction: column;
}
```



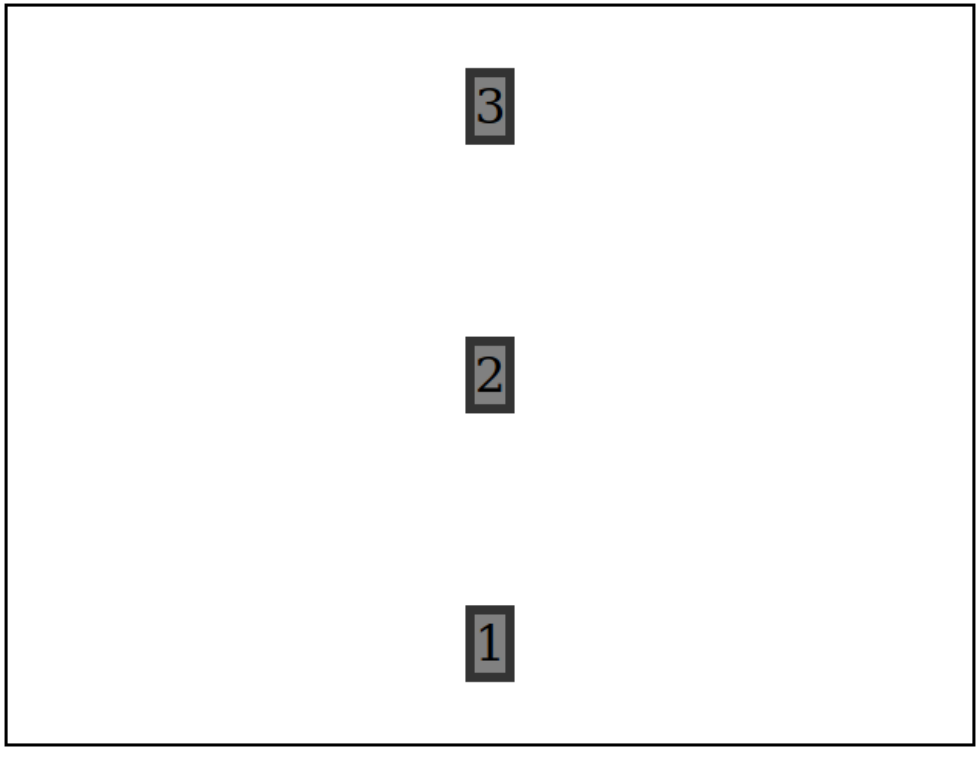
```
.container {
  flex-direction: row;
}
```



```
.container {
  flex-direction: row-reverse;
}
```



```
.container {
  flex-direction: column-reverse;
}
```

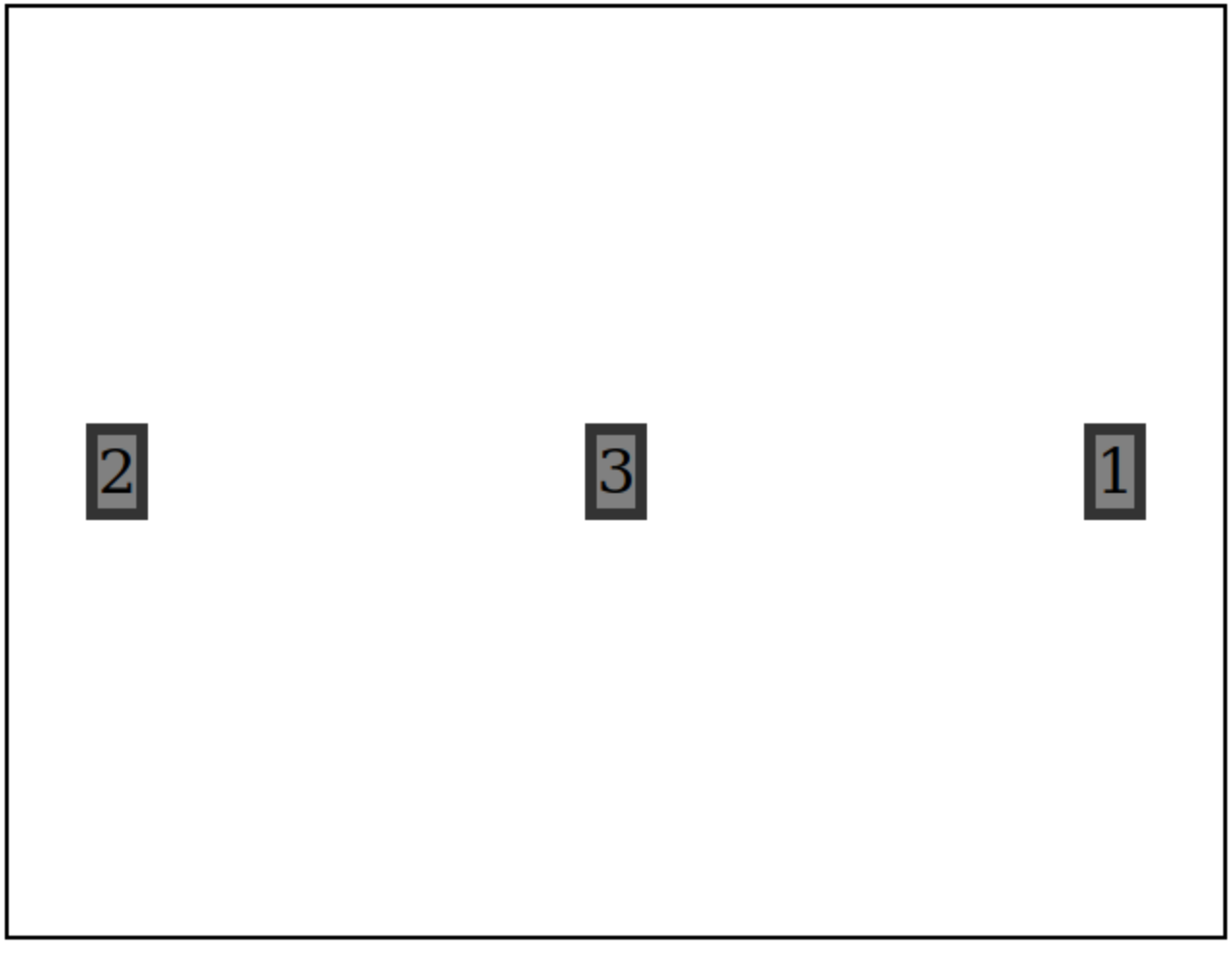


Modifier l'ordre d'affichage des flex items

flex-direction donne l'ordre d'affichage sur le flex container : soit dans l'ordre du HTML, soit dans l'ordre inverse

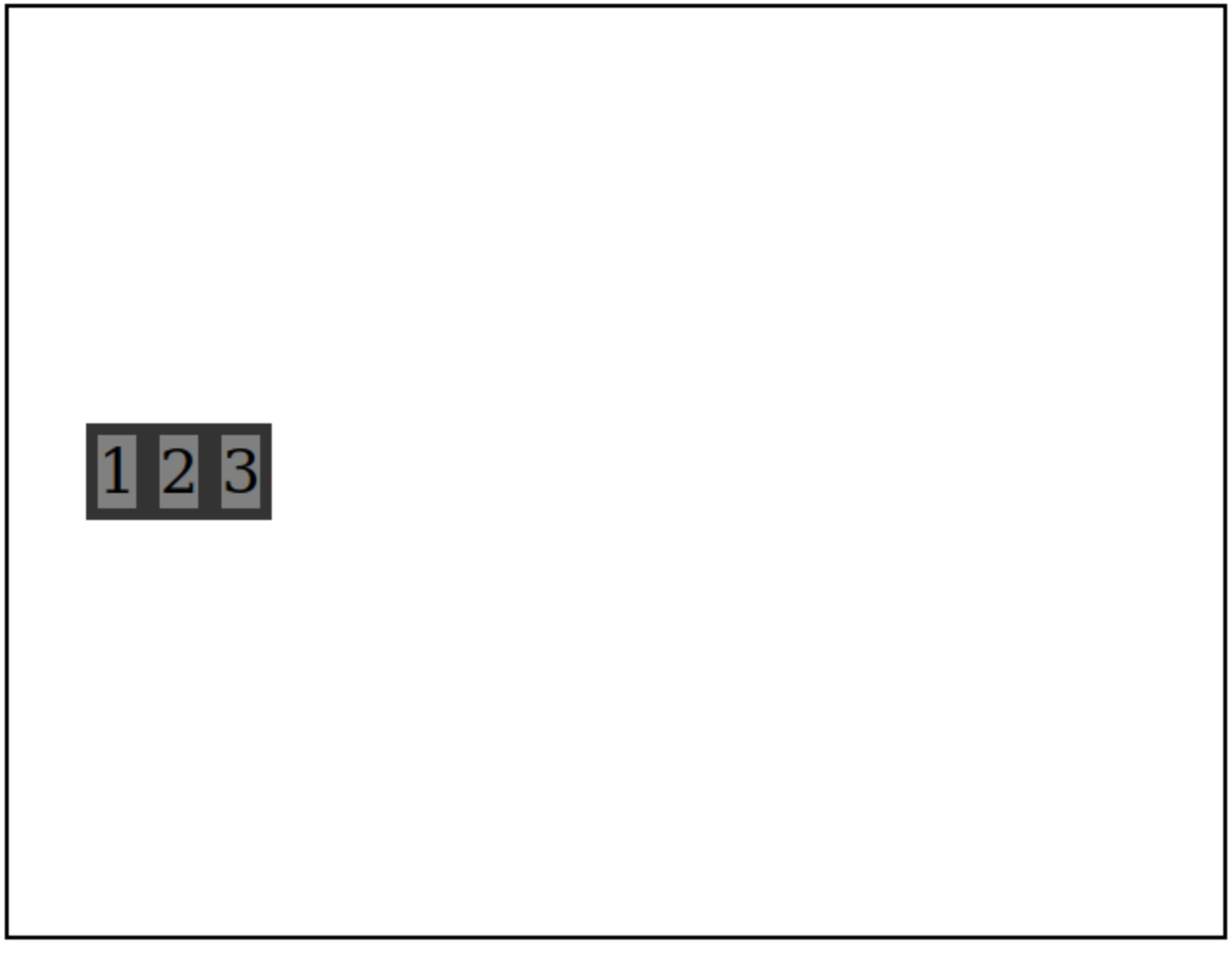
```css
.item:nth-child(1) {
  order: 3;
}
```

order - MDN


2
3
1

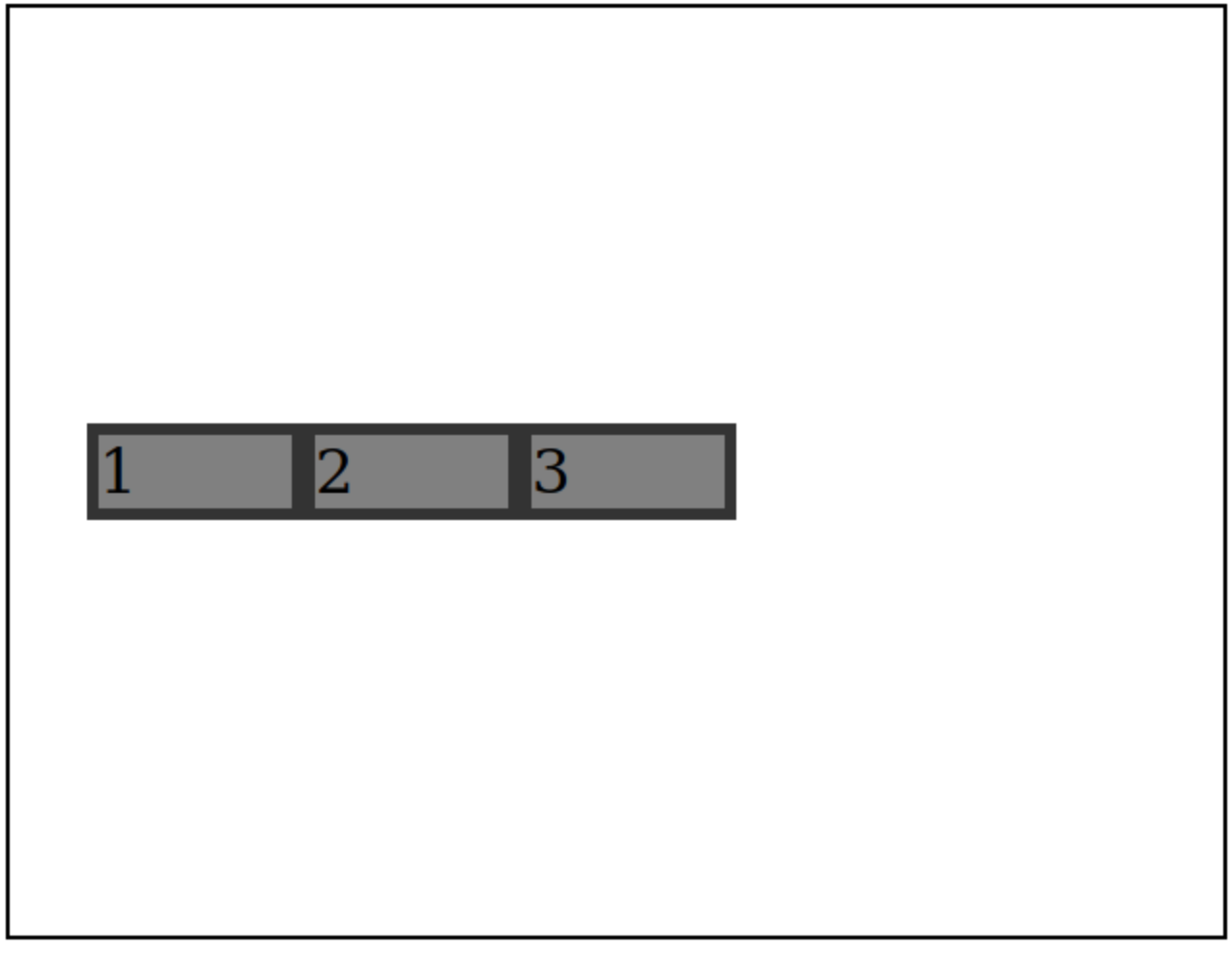
Tirer parti du « flex » de flexbox

```css
.container {
  display: flex;
  align-items: center;
  justify-content: flex-start;
}
```


1 2 3

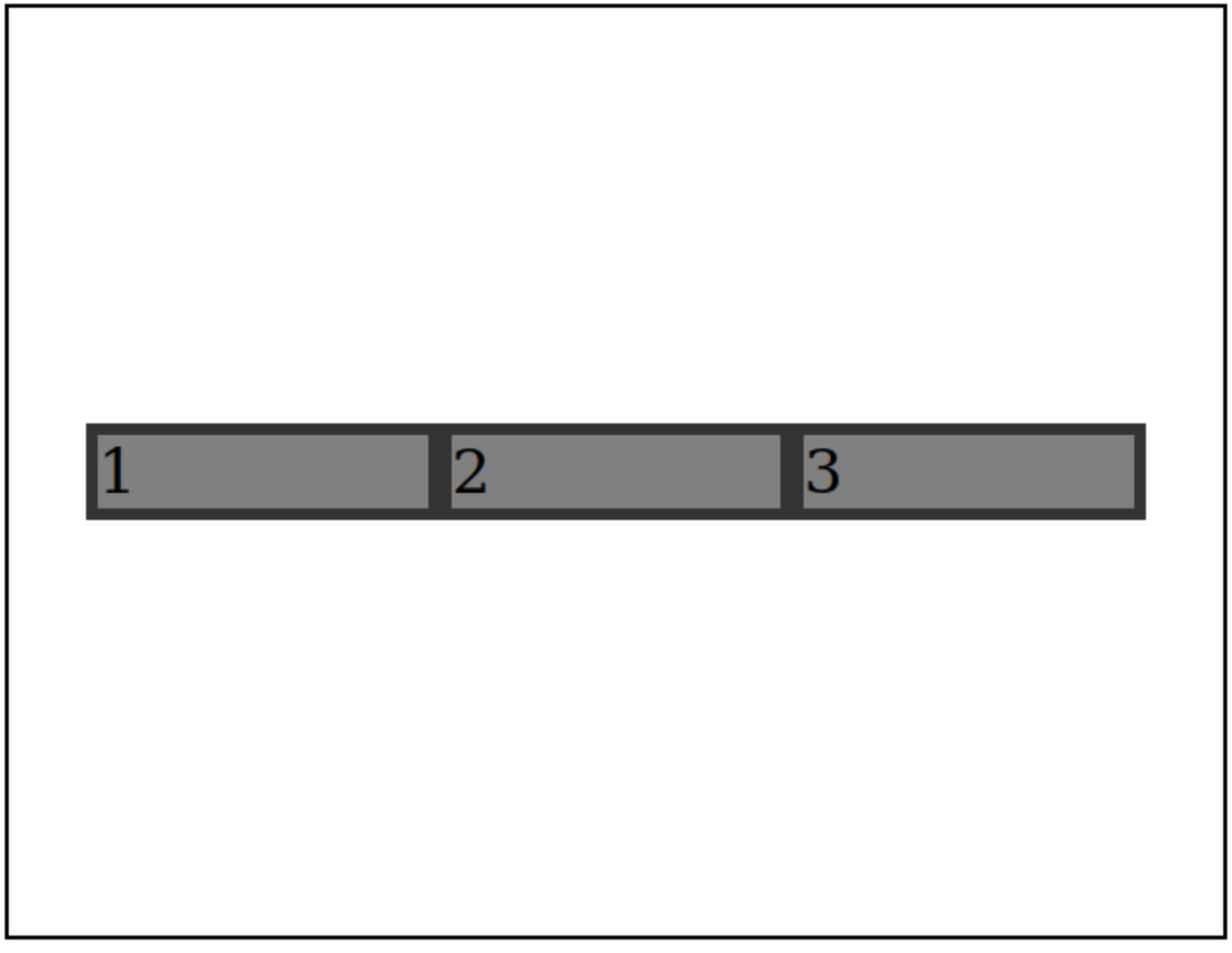
```
.item {
  flex-basis: 50px;
}
```

flex-basis - MDN


1
2
3

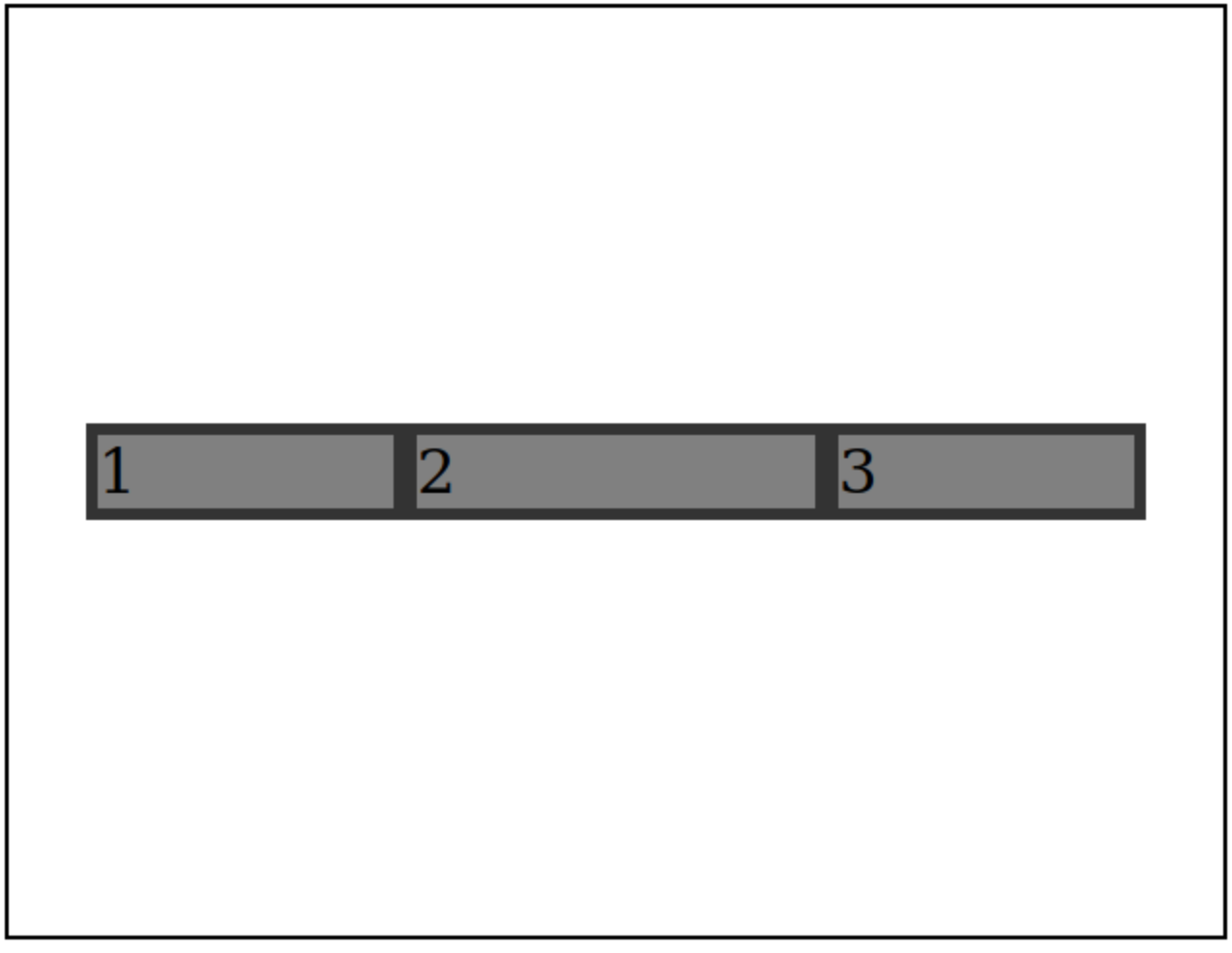
```
.item {
    flex-basis: 50px;
    flex-grow: 1;
}
```

flex-grow - MDN


1
2
3

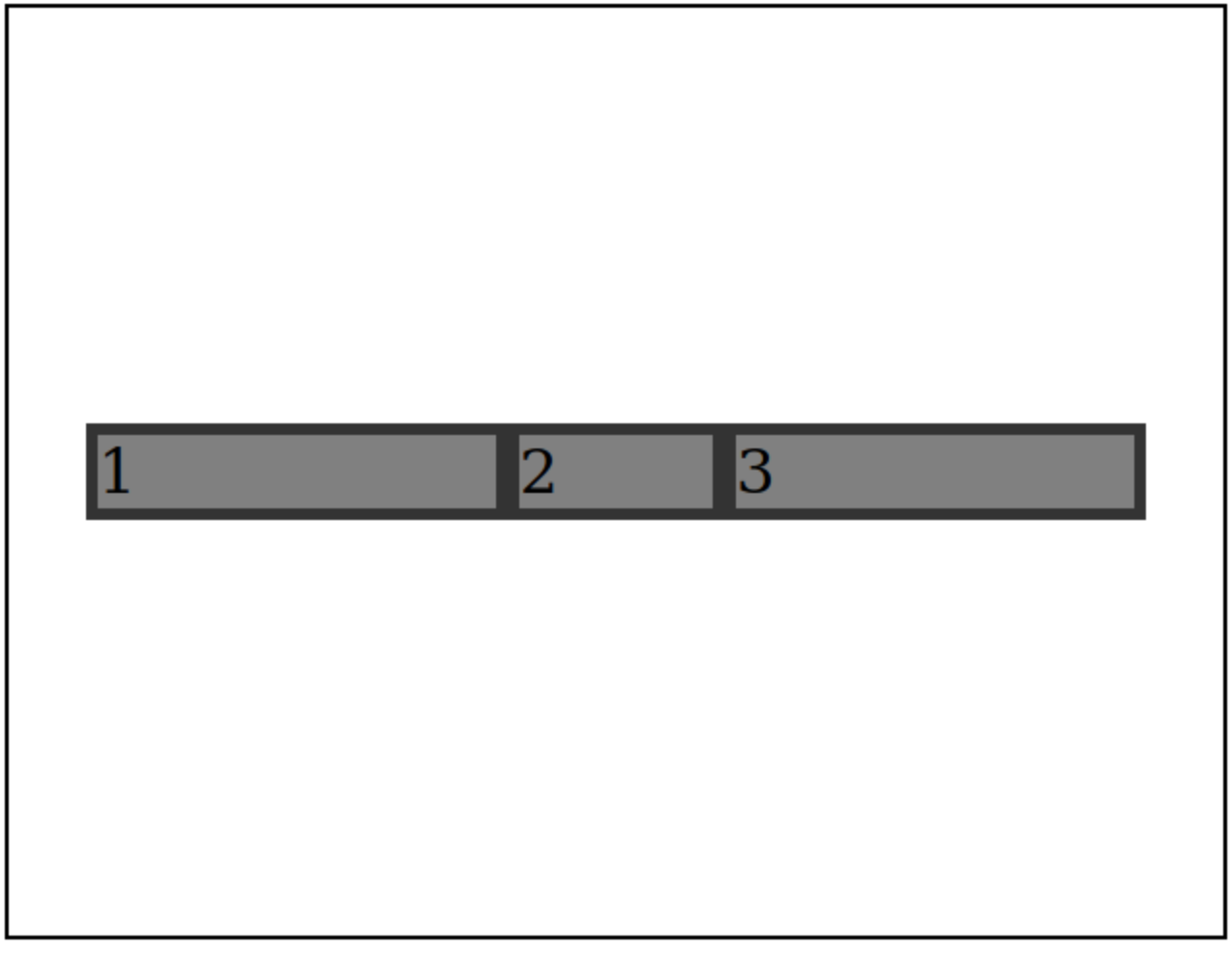
```
.item {
  flex-basis: 50px;
  flex-grow: 1;
}

.item:nth-child(2) {
  flex-grow: 2;
}
```

flex-grow - MDN

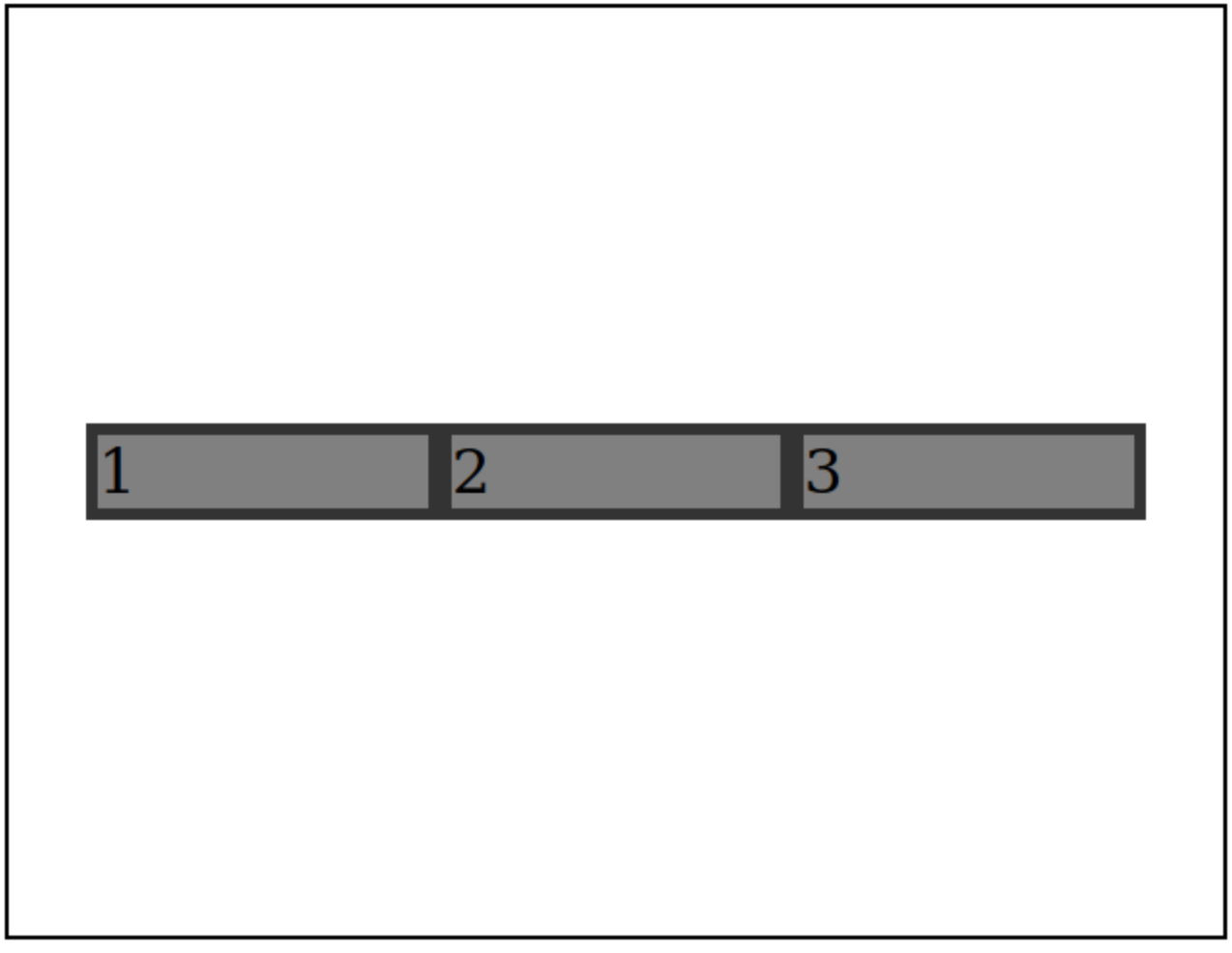

1
2
3

```
.item {
  flex-basis: 50px;
  flex-grow: 1;
}

.item:nth-child(2) {
  flex-grow: initial; /* or 0 */
}
```

flex-grow - MDN

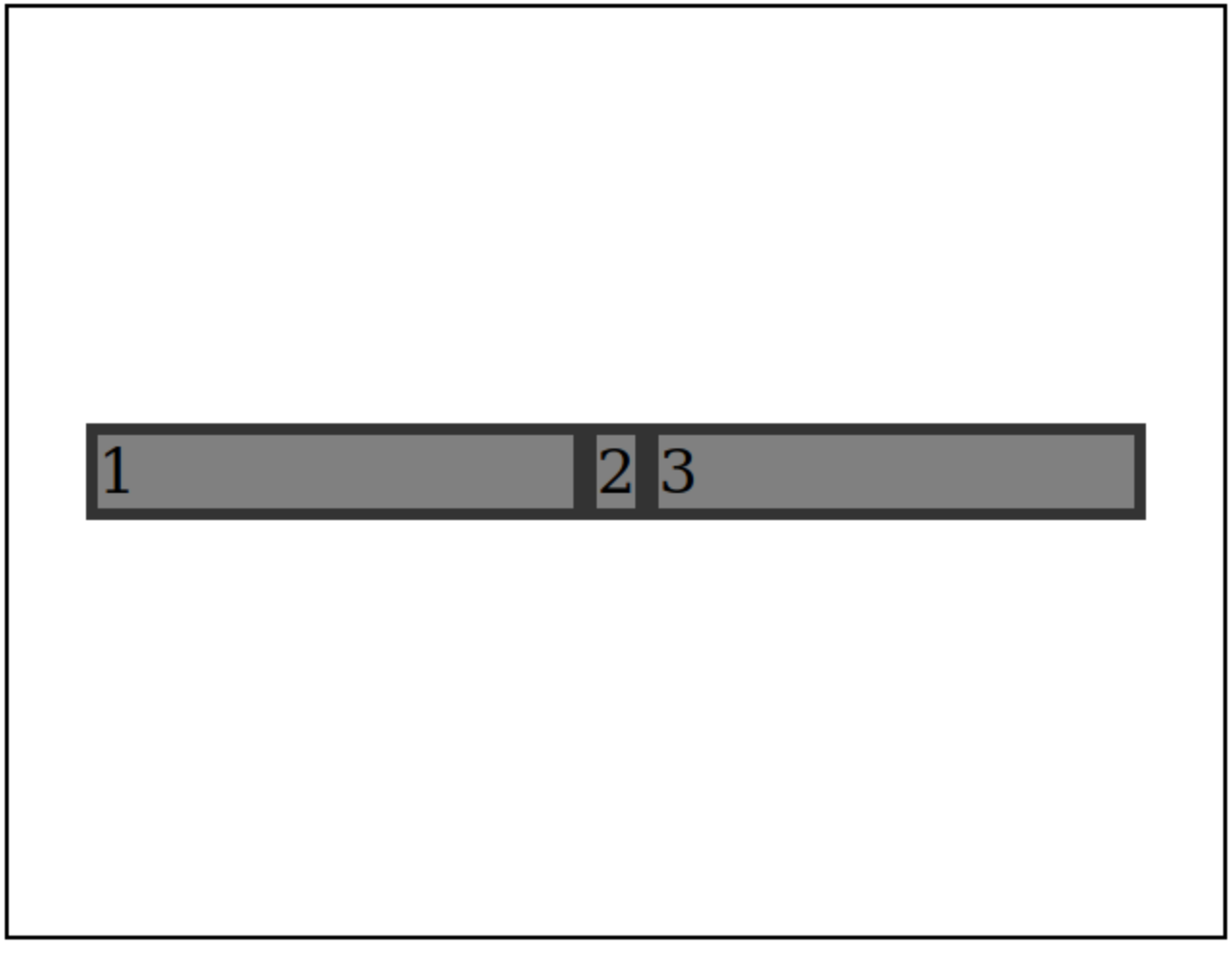

1
2
3

```
.item {
  flex-basis: 500px;
}
```


1
2
3

```
.item {
  flex-basis: 500px;
}

.item:nth-child(2) {
  flex-shrink: 2;
}
```

flex-shrink - MDN


1
2
3

Récapitulons

- `display: flex;` crée un flex container
- `justify-content` permet de définir l'alignement horizontal des flex items
- `align-items` permet de définir l'alignement vertical des flex items
- `flex-direction` permet d'afficher les items en ligne ou en colonne
- On peut modifier l'ordre global avec les directions `row-reverse` et `column-reverse`
- On peut modifier l'ordre plus finement en utilisant `order` sur les flex items
- On peut donner une taille initiale aux flex items avec `flex-basis`
- On peut modifier le facteur d'aggrandissement et de rétrécissement avec `flex-grow` et `flex-shrink`

Soufflez, c'est fini.

TD