

Développement Web - CSS-

Groupe des étudiants : CIR1

Maxime Lorentz

E-Mail : maxime.lorentz@gmail.com



Sortez moi du flux !



Rappel : le flux HTML

C'est l'ordre dans lequel apparaissent les balises dans le code HTML est celui dans lequel les boîtes sont affichées et s'empilent dans le document. Ce schéma de positionnement par défaut se nomme le **flux courant**. Les règles de positionnement dans le flux courant sont relativement simples et intuitives. Chaque élément :

- est situé sur le même plan que les autres éléments dans le flux
- se place **le plus haut possible** et **le plus à gauche possible au sein de son parent**
- est dépendant de l'élément frère précédent : 2 éléments de type block s'empilent, 2 éléments de type inline se suivent

A heading

One November night in the year 1782, so the story runs, two brothers sat over their winter fire in the little French town of Annonay, watching the grey smoke-wreaths from the hearth curl up the wide chimney.

One

Two

Three

Their names were **Stephen and Joseph Montgolfier**, they were papermakers by trade, and were noted as possessing thoughtful minds and a deep interest in all scientific knowledge and new discovery.

On voit ici l'empilement des éléments :
la balise H1, suivi d'un paragraphe etc.

Les exceptions

Tous les éléments d'un document sont dans le flux à l'exception :

- des éléments flottants
- des éléments avec `position: absolute` ou avec `position: fixed`
- de l'élément racine (`html`)

Les éléments flottants

La propriété **float** indique qu'un élément doit être retiré du flux normal et doit être placé sur le côté droit ou sur le côté gauche de son conteneur.

- **rôle** : sortir du flux normal
- **valeurs** : left | right | none | inline-start * | inline-end *
- **valeur par défaut** : none
- **appliqué à** : tous les éléments

* supporté uniquement par firefox

Comment se remettre dans le flux ?

Lorsque l'on retire un élément du flux normal, nous avons altéré l'écoulement logique du flux.

Pour nous permettre de revenir dans le flux logique, il faut utiliser la propriété **clear**

- **rôle** : du flux normal
- **valeurs** : none | left | right | both
- **valeur par défaut** : none
- **appliqué à** : tous les éléments

Position absolute - fixed

Un **élément positionné de façon absolue** est un élément dont la propriété de position calculée est `absolute` ou `fixed`. Dans ce cas, les propriétés [top](#), [bottom](#), [right](#) et [left](#) indiquent les distances entre les bords de l'élément et les bords du bloc englobant (c'est-à-dire l'ancêtre par rapport auquel l'élément est positionné). Si l'élément possède des marges, elles sont ajoutées aux décalages.

Structuré un document html

On peut structurer un document avec juste les positions absolute, relative, des tableaux, les propriétés float.

Mais on se heurte vite à des difficultés.



FlexBox



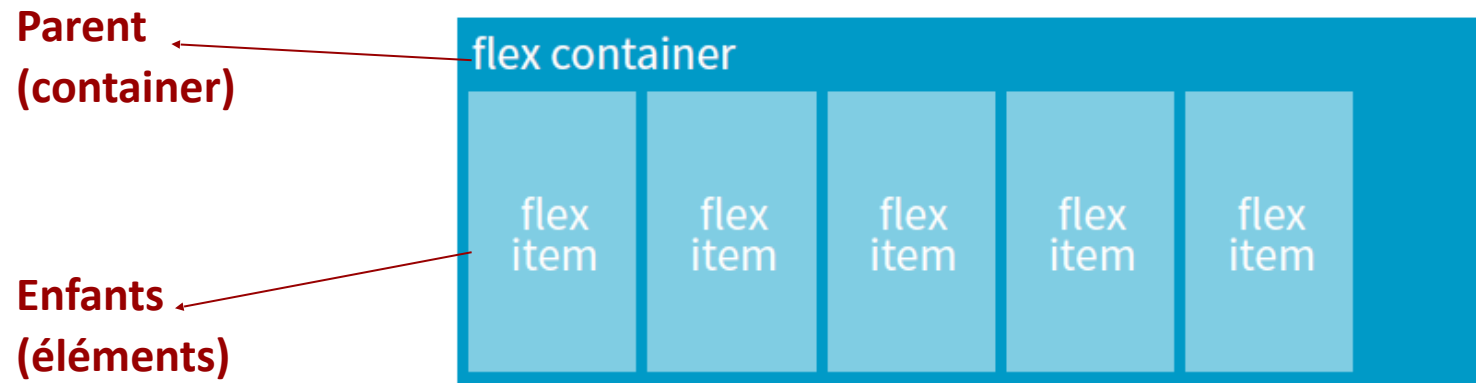
Définition

- **FlexBox** (= Flexible Boxes) est un module pour la présentation des composants d'une page web



Définition

- **FlexBox** (= Flexible Boxes) est un module pour la présentation des composants d'une page web
- **Principe :**
 - Définition d'un conteneur
 - Insertion de plusieurs éléments dans le conteneur



Définition

- **FlexBox** (= Flexible Boxes) est un module pour la présentation des composants d'une page web

- **Principe :**

- Définition d'un conteneur
- Insertion de plusieurs élément dans le conteneur



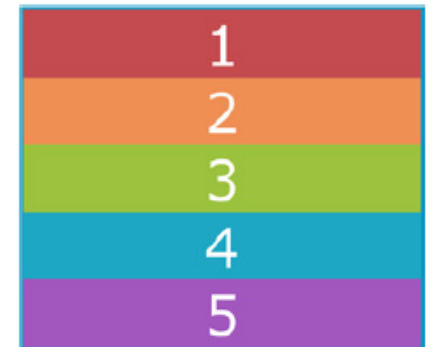
- **Déclaration :**

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

```
<div id="container">  
  <div class="box box1">1</div>  
  <div class="box box2">2</div>  
  <div class="box box3">3</div>  
  <div class="box box4">4</div>  
  <div class="box box5">5</div>  
</div>
```



flex layout mode



block layout mode (sans flex)

La direction : flex-direction

- **rôle** : définir la façon d'agencement des éléments
- **valeurs** : row | column | row-reverse | column-reverse
- **valeur par défaut** : row
- **appliqué aux** : éléments du conteneur flex

`flex-direction: row; (default)`



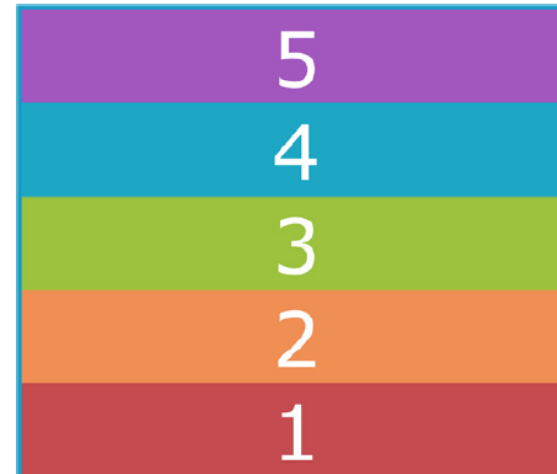
`flex-direction: row-reverse;`



`flex-direction: column;`



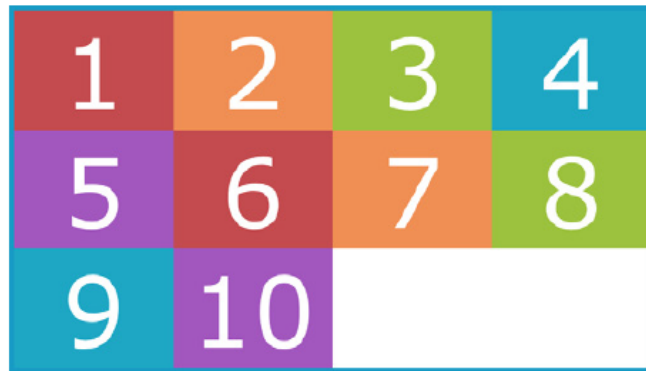
`flex-direction: column-reverse;`



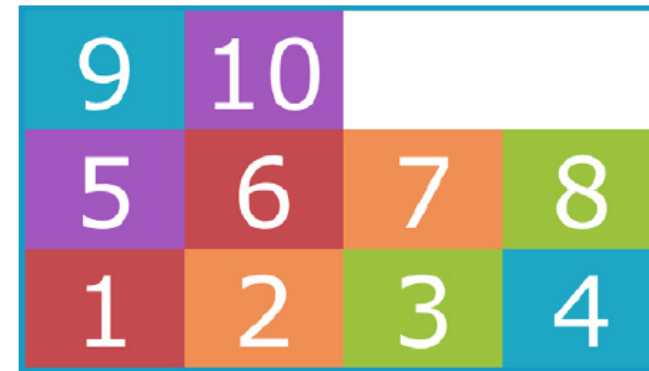
Le retour à la ligne : flex-wrap

- **rôle** : définir la façon d'affichage des éléments (une seule ligne ou plusieurs lignes)
- **valeurs** : nowrap | wrap | wrap-reverse
- **valeur par défaut** : nowrap
- **appliqué aux** : éléments du conteneur flex

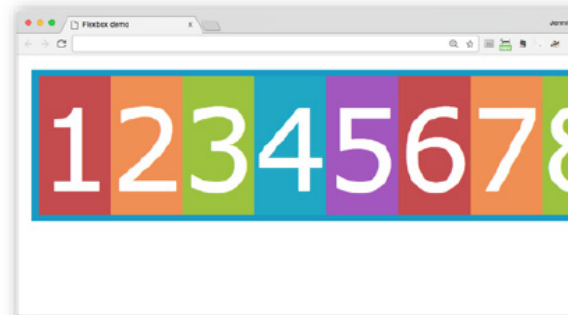
flex-wrap: wrap;



flex-wrap: wrap-reverse;



flex-wrap: nowrap; (default)



Alignement selon l'axe principale : justify-content

- **rôle** : définir la façon dont l'espace doit être réparti entre et autour des éléments selon l'axe principale
- **valeurs** : flex-start | flex-end | center | space-between | space-around
- **valeur par défaut** : flex-start
- **appliqué aux** : éléments du conteneur flex

`justify-content: flex-start;` (default)



`justify-content: flex-end;`



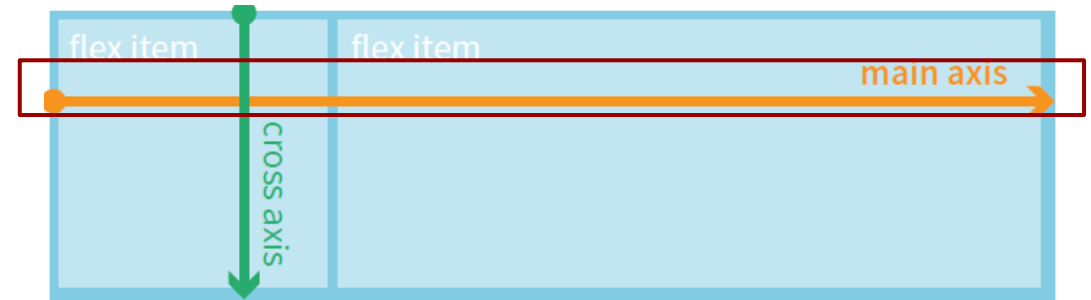
`justify-content: space-between;`



`justify-content: center;`

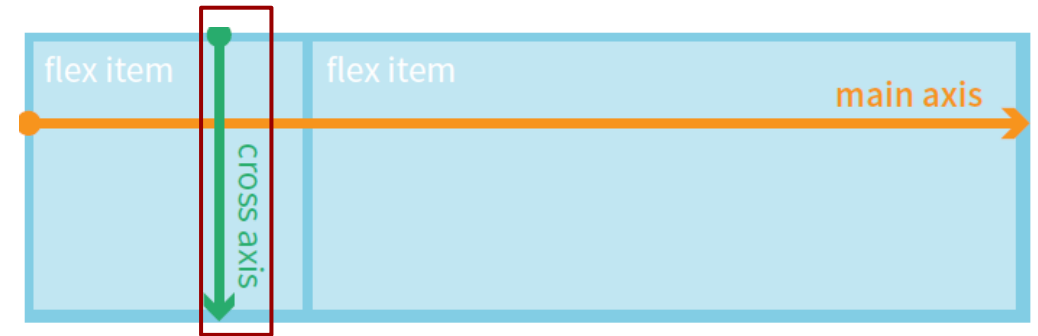


`justify-content: space-around;`

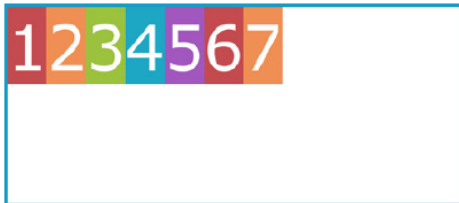


Alignement selon l'axe secondaire : align-items

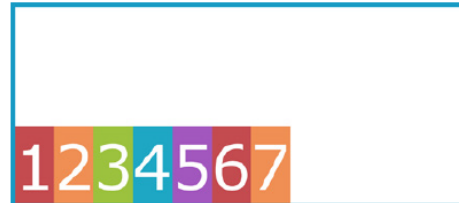
- **rôle** : définir la façon dont l'espace doit être réparti entre et autour des éléments selon l'axe secondaire
- **valeurs** : flex-start | flex-end | center | baseline | stretch
- **valeur par défaut** : stretch
- **appliqué aux** : éléments du conteneur flex



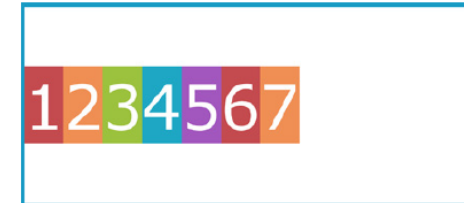
`align-items: flex-start;`



`align-items: flex-end;`



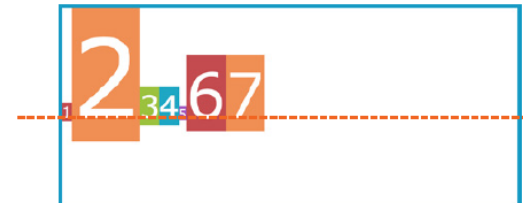
`align-items: center;`



`align-items: stretch;` (default)

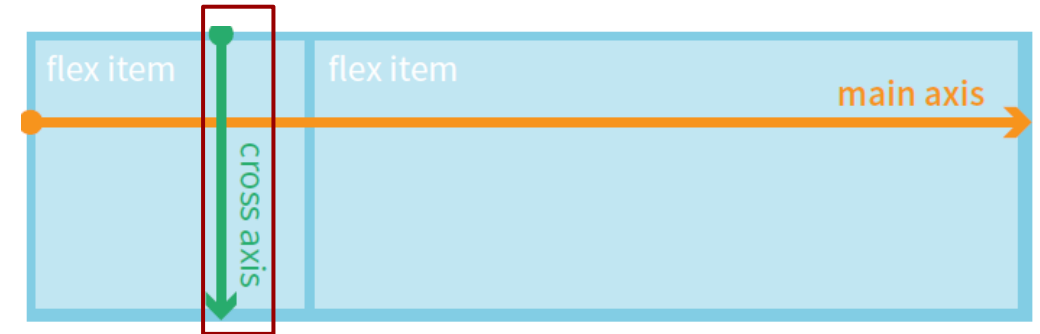


`align-items: baseline;`



Alignement d'un seul élément : align-self

- **rôle** : définir la façon dont l'espace doit être réparti pour un élément selon l'axe secondaire
- **valeurs** : flex-start | flex-end | center | baseline | stretch
- **valeur par défaut** : stretch
- **appliqué aux** : éléments du conteneur flex



```
.box4 {  
  align-self: flex-end;  
}
```

align-self: flex-end;



Expansion d'un élément flex : flex-grow

- **rôle** : définir le facteur d'expansion d'un élément
- **valeurs** : nombre
- **valeur par défaut** : 0
- **appliqué à** : un élément flex

flex-grow : 0



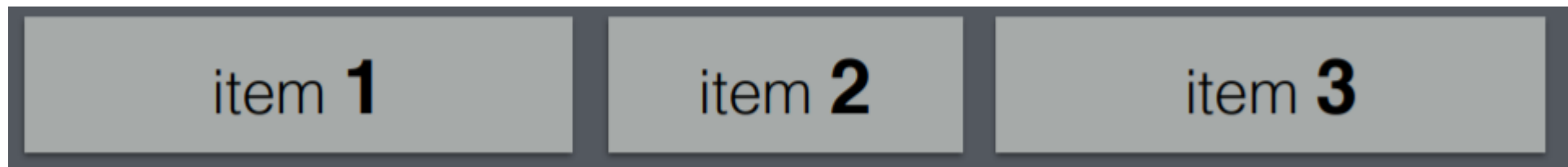
flex-grow : 1



Compression d'un élément flex : flex-shrink

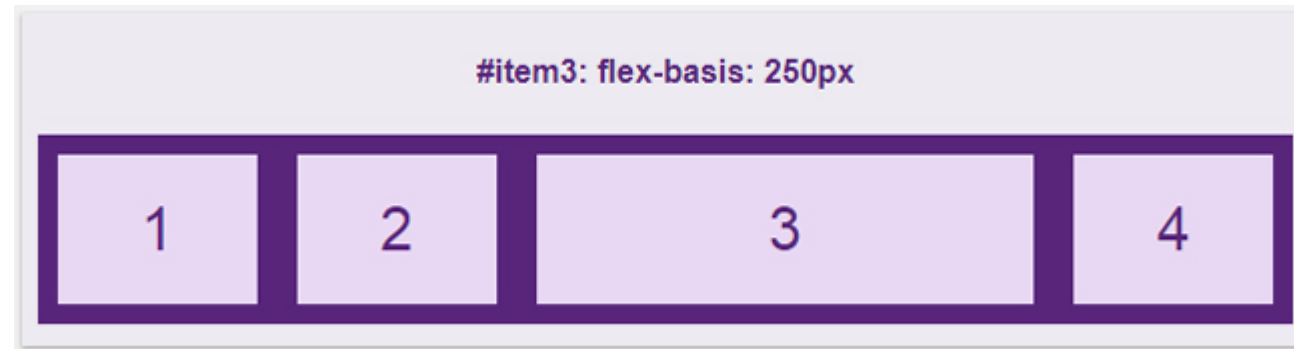
- **rôle** : définir le facteur de compression d'un élément
- **valeurs** : nombre
- **valeur par défaut** : 1
- **appliqué à** : un élément flex

```
#item1, #item3 {  
    display: flex;  
    flex-shrink: 2;  
}
```



Taille initiale d'un élément flex : flex-basis

- **rôle** : définir la taille initiale d'un élément flex
- **valeurs** : length | percentage | content | auto
- **valeur par défaut** : auto
- **appliqué à** : un élément flex



Ordre d'un élément flex : order

- **rôle** : définir l'ordre d'un élément
- **valeurs** : nombre entier
- **valeur par défaut** : 0
- **appliqué à** : un élément flex

```
.box3 {  
  order: 1;  
}
```



Ordre d'un élément flex : order

- **rôle** : définir l'ordre d'un élément
- **valeurs** : nombre entier
- **valeur par défaut** : 0
- **appliqué à** : un élément flex

```
.box2, .box3 {  
  order: 1  
}
```



Ordre d'un élément flex : order

- **rôle** : définir l'ordre d'un élément
- **valeurs** : nombre entier
- **valeur par défaut** : 0
- **appliqué à** : un élément flex

```
.box5 {  
  order: -1  
}
```





Bootstrap 4 utilise flexbox dans son système de colonnade

[bootstrap](#) / dist / css / bootstrap-grid.css

```
.row {  
  display: flex;  
  flex-wrap: wrap;  
  margin-right: -15px;  
  margin-left: -15px;  
}
```

```
.col {  
  flex-basis: 0;  
  flex-grow: 1;  
  max-width: 100%;  
}
```

Source : <https://github.com/twbs/bootstrap/blob/master/dist/css/bootstrap-grid.css>