

Flexbox – Deux Point Importants

- ❑ Les Types d'éléments
- ❑ Les Axes

Les Type d'éléments

- ❑ Le Flex Container (le parent)
- ❑ Le(s) Flex Item(s) (les enfants)

Propriétés (déclarations) du Flex Container

❑ `display : flex`

(nous permet d'agir sur les enfants)

❑ `flex-direction:`

(nous permet de déterminer l'axe principal)

➤ `row` | `row-reverse` | `column` | `column-reverse`

❑ `flex-wrap:`

(autorise le retour à la ligne au cas où un enfant déborderait sur l'axe secondaire)

➤ `wrap` | `nowrap` | `wrap-reverse`

❑ `justify-content :`

(nous permet d'aligner les éléments enfants par rapport à l'axe principal)

➤ `flex-start` | `flex-end` | `center` | `space-between` | `space-around`

Propriétés (déclarations) du Flex Container

❑ `align-items` :

(aligne les éléments par rapport à l'axe secondaire)

➤ `center` | `stretch` | `flex-start` | `flex-end` | `baseline`

❑ `align-content` :

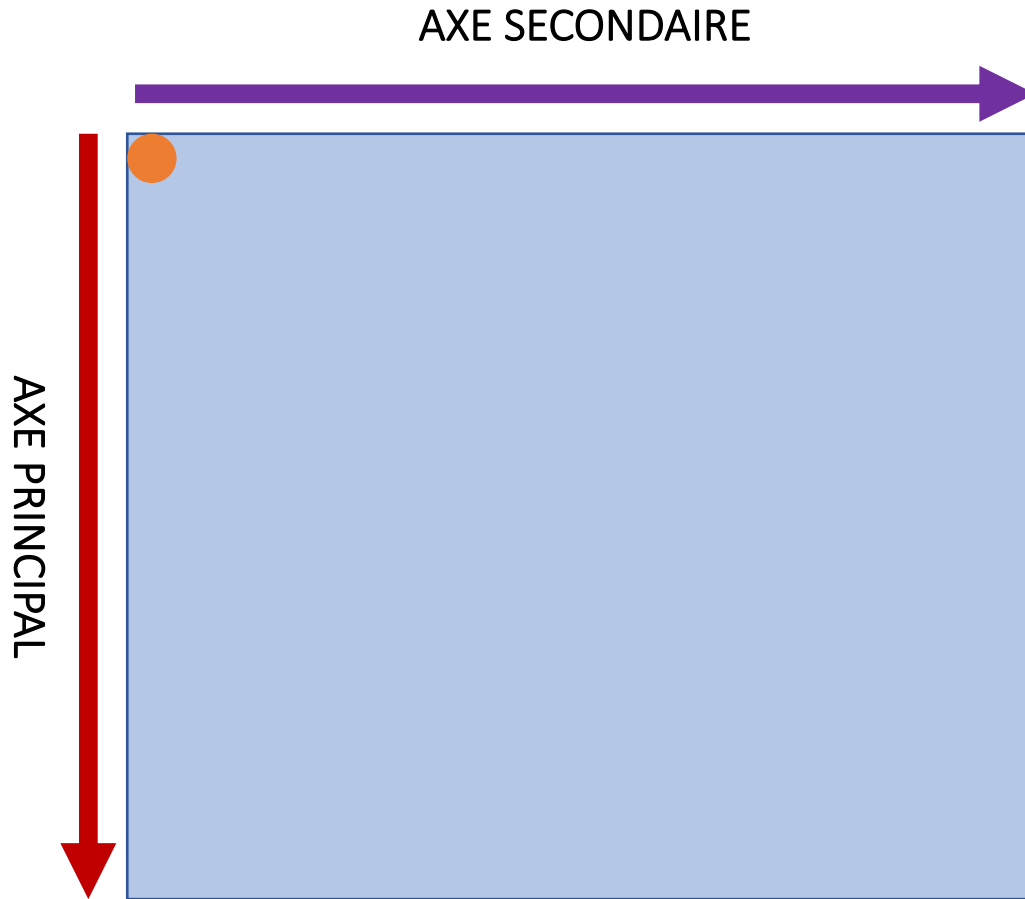
(aligne l'ensemble des éléments enfants par rapport à l'axe secondaire)

➤ `stretch` | `flex-start` | `flex-end` | `space-between` | `space-around` | `center`

➤ il faut avoir plusieurs lignes ou `flex-wrap` différent de `nowrap`

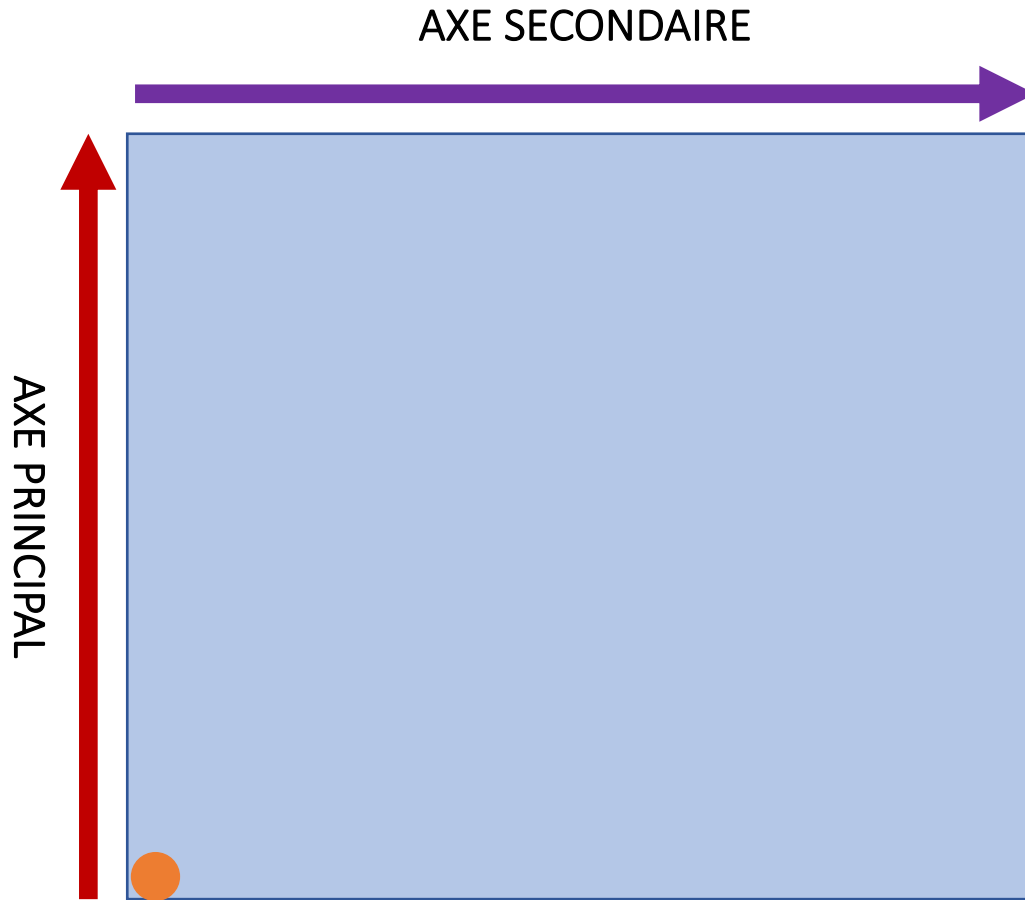
Toutes ces propriétés ci dessus ont des conséquences sur les enfants
(flex items)

Conséquences sur les axes



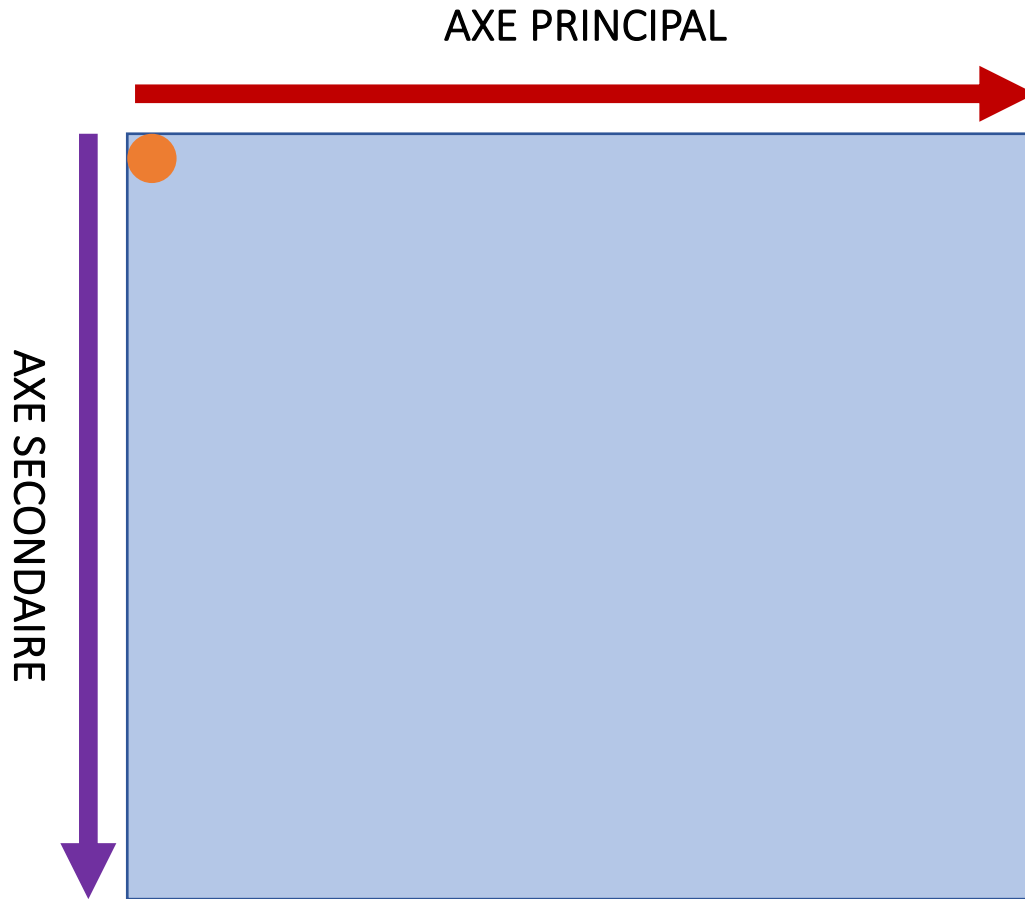
flex-direction: **column**
De haut vers le bas.

Conséquences sur les axes



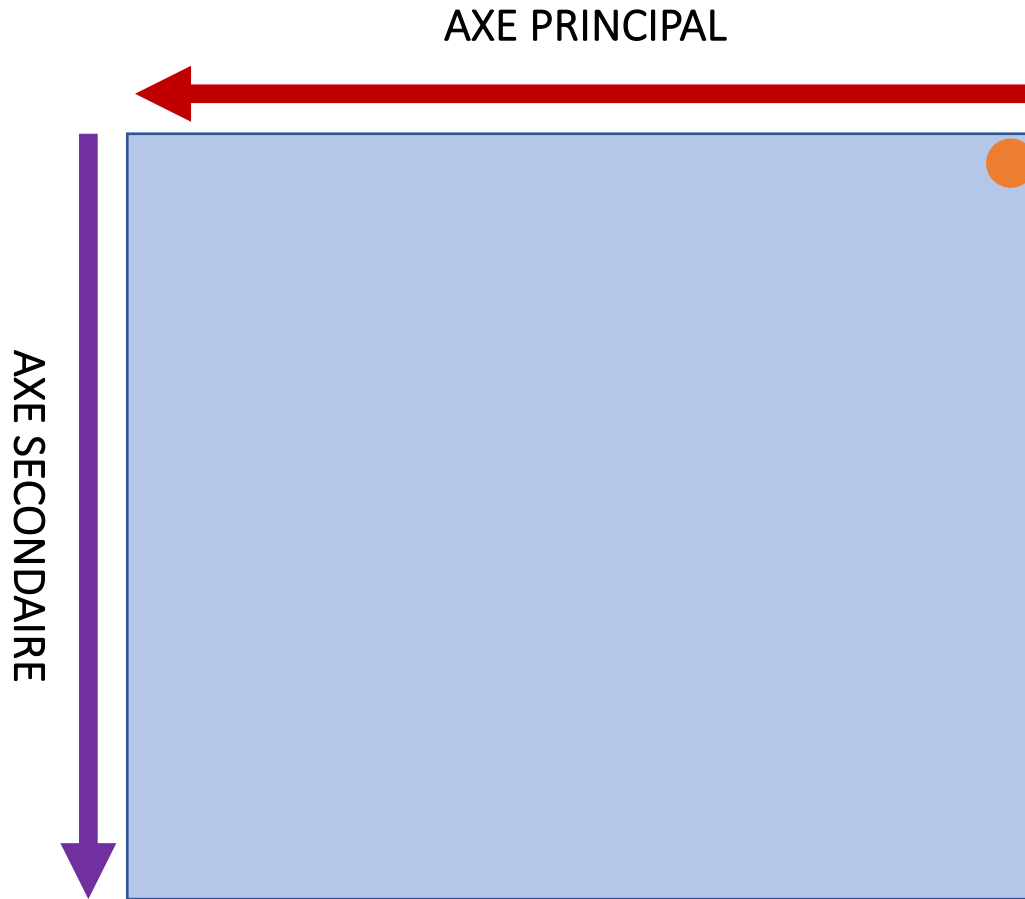
flex-direction: column-reverse
De bas vers le haut.

Conséquences sur les axes



flex-direction: **row**
De gauche à droite.

Conséquences sur les axes



flex-direction: **row-reverse**
De droite à gauche.

Propriétés (déclarations) du Flex Item

❑ **order** : **<nombre>**

(détermine l'ordre d'alignement des enfants. Le plus grand nombre se mettra le plus à droite de **l'axe principale** et ainsi de suite. **0** par défaut);

❑ **flex-grow**: **<nombre>**

(détermine la grandeur d'un enfant par rapport aux autres quand la taille max est atteinte. **0** par défaut)

❑ **flex-shrink**: **<nombre>**

(l'inverse du flex-grow, il retrécit l'élément. **1** par défaut)

- s'applique si la taille de l'élément est supérieure au conteneur.
<https://developer.mozilla.org/fr/docs/Web/CSS/flex-shrink>

❑ **align-self** :

(modifie l'alignement d'un seul élément par rapport à **l'axe secondaire**)

- **flex-start** | flex-end | center | auto

❑ **flex-basis** : **<nombre>**

(détermine la taille d'un élément par rapport à **l'axe principale**)

Flexbox et React Native

- ❑ React Native utilise les Flexbox grâce à **Yoga**.
- ❑ **Yoga** est un moteur open source de mise en page (flexible);
<https://yogalayout.com/>
- ❑ En bref, **yoga** utilise les Flexbox (avec une syntaxe en **camel case**) + certaines autres propriétés;
- ❑ <https://yogalayout.com/docs/> (toutes ces propriétés nous permettent de créer notre mise en page flexible)
- ❑ C'est grâce à **Yoga** que l'on peut utiliser les Flexbox dans React Native.
- ❑ Yoga a aussi un playground <https://yogalayout.com/playground>
- ❑ Quand on parle de layout (disposition, alignement, tout ce qui concerne la mise en page)
- ❑ Propriétés yoga et leurs équivalents css <https://reactnative.dev/docs/layout-props>