# Exercices module Flexbox

## Préparation

*Réalisez un modèle de page pour tous les exercices du module flexbox*

***Dans la structure de base de la page html, créez cinq blocs parents***
*Chaque bloc est identifié*

- *par une classe* **`class="flex-container"`**. *qui permettra d'attribuer les propriétés css communes à ces 5 blocs*
- *par un identifiant* **`id="flex-container1"`** etc*. pour les propriétés spécifiques à chaque bloc.*

```
<div class="flex-container" id="flex-container1"></div>
<div class="flex-container" id="flex-container2"></div>
    .
    .
<div class="flex-container" id="flex-container5"></div>
```

Insérer dans ces blocs parents, 3 blocs enfants identifiés par la classe **`class="element"`**.
Le contenu du premier bloc : élément1
Le contenu du deuxième bloc : élément2 élément2
Le contenu du troisième bloc : élément3 élément3 élément3

```
<div class="flex-container" id="flex-container1">
      <div class="element">élément1</div>
      <div class="element">element2 élément2</div>
      <div class="element">element3 élément3 élément3</div>
</div>
```

## Résultat

***#flex-container1***


***#flex-container2***


***Etc.***

*Selon les différents exercices vous devrez ajouter ou supprimer des blocs flex-container en fonction du nombre de valeurs dont bénéficient les différentes propriétés et ajouter des éléments en fonction du nombre d'éléments à afficher.*

**Dans la partie head de la page**, faire un lien vers la feuille de style css **`"flex-style.css"`** qui se trouve dans le dossier **`"flex-css".`** Sur cette page seront codées les propriétés communes à tous les blocs (celles-ci étant valables pour tous les exercices).

A la suite de cette déclaration insérez les balises **`<style></style>`** qui serviront à coder les propriétés spécifiques aux blocs pour chaque exercice.

```
<link rel="stylesheet" href="le lien vers le fichier" media="screen" type="text/css" />
<style>
</style>
```

***Enregistrez cette page sous flex-modele.html***

***Dans la feuille de style css codez :***

*La propriété* ***`box-sizing :border-box`***

Tous les blocs flex-container ont une couleur de fond **`#ffe764`** et un **`padding`** de 2px.
Les blocs element ont une **`bordure`** de 1px noire et un **`padding`** de 10px.
La police de caractère **`century gothic`**, en majuscules et de **`couleur blanche`**.

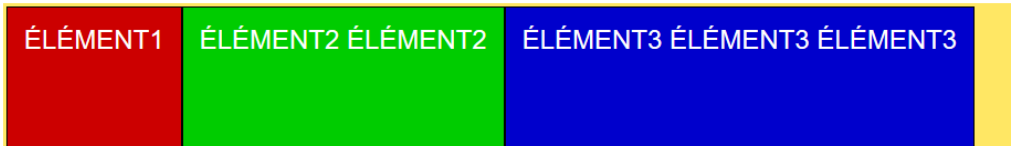
Pour cibler les blocs enfants utilisez le pseudo-sélecteur du type **`nth-of-type`**.
Le premier bloc est de couleur rouge #c00
Le second est vert : #0c0
Le troisième bleu : #00c;
Certains exercices nécessitent cinq blocs. Lorsque vous créerez ces autres blocs :
Le quatrième sera orange : orange
Le cinquième et dernier bleu marine : navy.

***Enregistrez cette page sous flex-style.css dans son dossier.***

## Exercice1. Appliquez la propriété display : flex sur le bloc parent

**Cette propriété permet d'agencer, en mode flexbox, les éléments qui se trouvent à l'intérieur du bloc flex-container1.** Par défaut tous les éléments se positionnent les uns à côté des autres. Si aucune largeur n'est précisée, la largeur de chaque bloc est celle de son contenu et sa hauteur est celle du bloc parent. Tant qu'on ne donne aucune autre instruction, les blocs restent positionnés les uns à côté des autres, quel que soit le contenu.

*Indiquez une hauteur de 180px au bloc parent.*

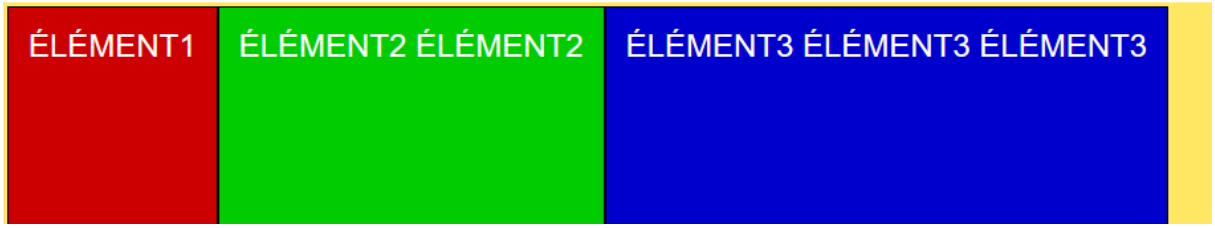



*Modifiez ensuite cette hauteur, indiquez 100vh (100vh correspond à une hauteur de fenêtre d'ordinateur) et observez.*
*Le contenu est étiré sur toute la hauteur de la fenêtre.*

## Exercice2. La propriété flex-direction

La **propriété flex-direction** (utilisée avec **display:flex**) permet de positionner les blocs selon un axe principal (main-*axis*) qui peut être l'axe horizontal ou l'axe vertical.
Ajoutez la propriété display-direction et sa valeur row (valeur par défaut, **flex-direction :row ;**) sur le bloc flex-container1. L'axe principal est l'axe horizontal. Les blocs restent positionnés les uns à côté des autres.
Poursuivez avec la valeur row-reverse sur le bloc flex-container2 (les éléments à l'intérieur du bloc conteneur apparaissent côte à côte maisen ordre inverse) puis avec les valeurs column et column reverse sur les deux blocs flex-container 3 et 4.

1. **row**




2. **row-reverse**


3. **column**


4. **column-reverse**


## Exercice3. La propriété flex-wrap

La **propriété flex-wrap** utilisée avec les propriétés **display:flex et flex-direction** permet de gérer l'affichage des blocs lorsqu'il n'y a plus d'espace.

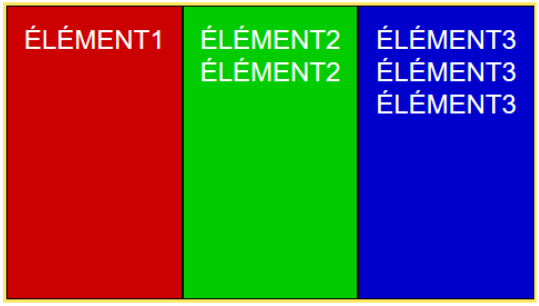
Pour diminuer l'espace horizontal , réduisez la taille de la fenêtre du navigateur ou diminuez la largeur du flex-container.

### 3.a. la propriété flex-wrap utilisée avec les propriétés display:flex et flex-direction:row

**1. Dans le bloc flex-container1 indiquez la valeur nowrap** (par défaut) : les blocs sont alignés sans retour à la ligne quelle que soit leur largeur d'origine.

**2. Dans le bloc flex-container2 indiquez la valeur wrap** : s'il n'y a plus suffisamment de place horizontalement, les blocs passent à la ligne. La largeur d'origine est prise en compte.
**3. Dans le bloc flex-container3 indiquez la valeur wrap-reverse** : s'il n'y a plus suffisamment de place, les blocs passent à la ligne, les lignes sont organisées en ordre inverse

1. **nowrap**



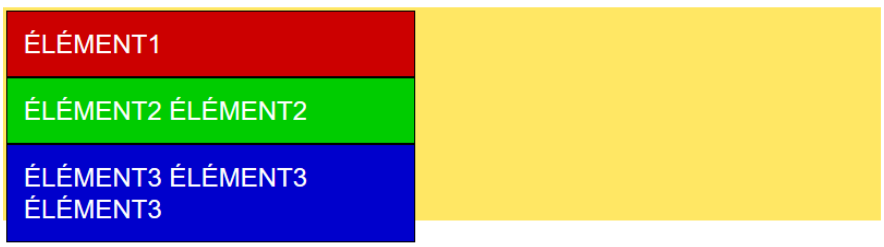

2. **wrap**


3. **wrap-reverse**

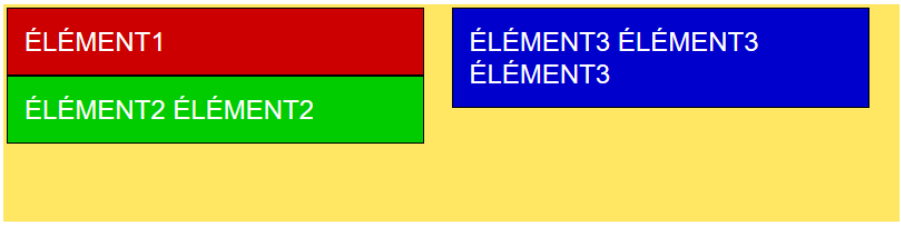

### 3.b. la propriété flex-wrap utilisée avec les propriétés display:flex et flex-direction:column

**1. Dans le bloc flex-container1 indiquez la valeur nowrap** (par défaut) : les blocs sont alignés les uns en dessous des autres sans passage sur une autre colonne.
**2. Dans le bloc flex-container2 indiquez la valeur wrap** : s'il n'y a plus suffisamment de place verticalement, les blocs passent sur une seconde colonne.
**3. Dans le bloc flex-container3 indiquez la valeur wrap-reverse** : s'il n'y a plus suffisamment de place, les blocs passent sur une seconde colonne en ordre inverse

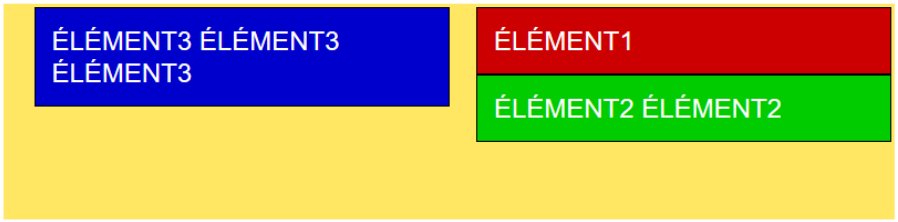
1. **nowrap**



2. **wrap**




3. **wrap-reverse**




### 3.c. Essayez de réaliser ces deux derniers exercices avec la propriété flex-flow

La propriété flex-flow est une propriété raccourcie qui permet d'indiquer en une seule déclaration les valeurs de flex-direction et flex-wrap.

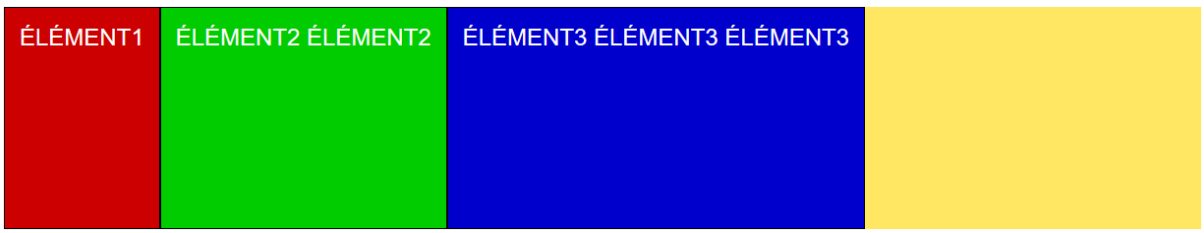
## Exercice 4. La propriété justify-content

La propriété justify-content permet de spécifier la façon dont les blocs occupent l'espace sur l'axe principal quand l'espace n'est pas totalement comblé.

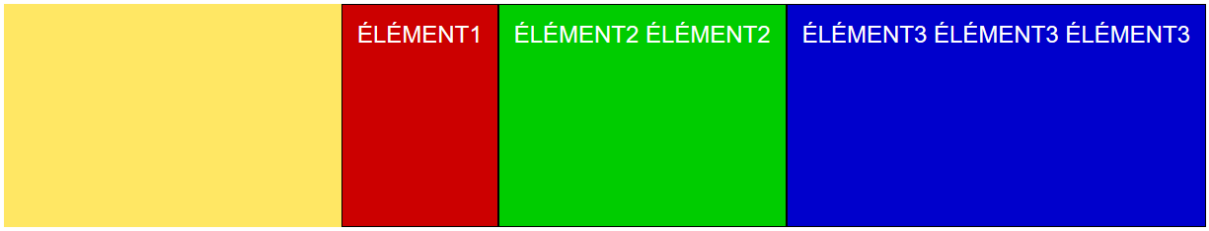
### 4.a. la propriété justify-content utilisée avec les propriétés display:flex et flex-direction:row

**1. Dans le bloc flex-container1 indiquez la valeur flex-start** (par défaut) : Les éléments sont alignés au début de l'axe horizontal.
**2. Dans le bloc flex-container2 indiquez la valeur flex-end** : Les éléments sont alignés à la fin de l'axe horizontal.
**3. Dans le bloc flex-container3 indiquez la valeur center** : Les éléments sont centrés sur l'axe horizontal.
**4. Dans le bloc flex-container4 indiquez la valeur space-between** : Les éléments occupent tout l'axe horizontal, il y a des espaces entre eux.
**5. Dans le bloc flex-container5 indiquez la valeur space-around** : Les éléments occupent tout l'axe horizontal, il y a des espaces entre les éléments et également en début et en fin de ligne.

1. **flex-start**



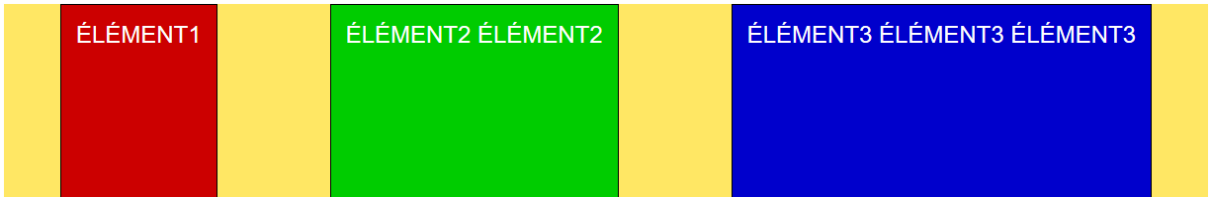
2. **flex-end**




3. **center**


4. **space-between**


5. **space-around**




### 4.b. la propriété justify-content utilisée avec les propriétés display:flex et flex-direction:column

**1. Dans le bloc flex-container1 indiquez la valeur flex-start** (par défaut) : Les éléments sont alignés au sommet de l'axe vertical.
**2. Dans le bloc flex-container2 indiquez la valeur flex-end** : Les éléments sont alignés en bas de l'axe vertical.
**3. Dans le bloc flex-container3 indiquez la valeur center** : Les éléments sont centrés sur l'axe vertical.
**4. Dans le bloc flex-container4 indiquez la valeur space-between** : Les éléments occupent tout l'axe vertical, il y a des espaces entre eux.
**5. Dans le bloc flex-container5 indiquez la valeur space-around** : Les éléments occupent tout l'axe vertical, il y a des espaces entre les éléments et également en début et en fin de ligne.

**1. flex-start**


**2. flex-end**


**3. center**


**4. Space-between**


**5. space-around**


## Exercice 5. La propriété align-items

La propriété align-items permet de positionner les blocs selon un axe secondaire (*cross-axis*). L'axe secondaire est l'axe vertical si l'axe principal est l'axe horizontal (flex-direction :row ) et inversement l'axe secondaire sera l'axe horizontal si l'axe principal a été défini par flex-direction :column.

### 5.a. la propriété align-items utilisée avec les propriétés display:flex et flex-direction:row

**1. Dans le bloc flex-container1 indiquez la valeur stretch** (par défaut)

**2. Dans le bloc flex-container2 indiquez la valeur center**

**1. stretch**


**2. center**


### 5.b. la propriété align-items utilisée avec les propriétés display:flex et flex-direction:column

**1. Dans le bloc flex-container1 indiquez la valeur stretch** (par défaut)
**2. Dans le bloc flex-container2 indiquez la valeur center**

**1. stretch**

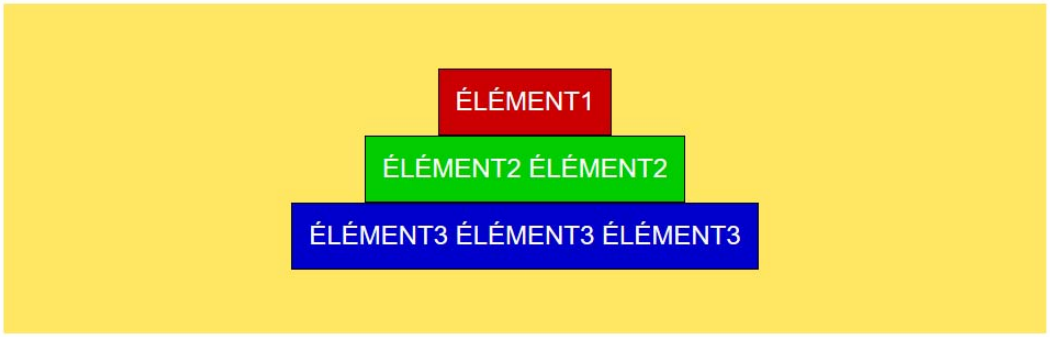

**2.center**


## Exercice 6. Centrer horizontalement et verticalement avec les propriétés justify-content et align-items

Utilisées avec display :flex  et flex-direction, les propriétés justify-content:center; et align-items: center ; permettent de centrer des éléments horizontalement et verticalement dans le bloc parent

### 6.a. Centrer des éléments selon l'axe horizontal flex-direction :row


### 6.b. Centrer des éléments selon l'axe vertical flex-direction :column

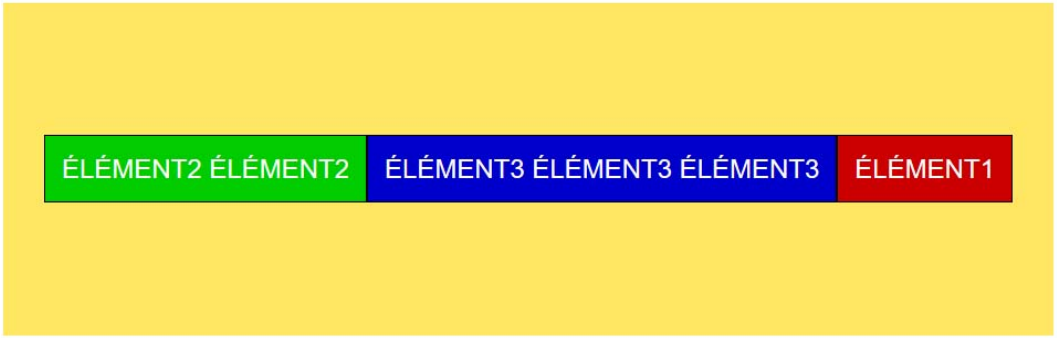



## 7. La propriété order

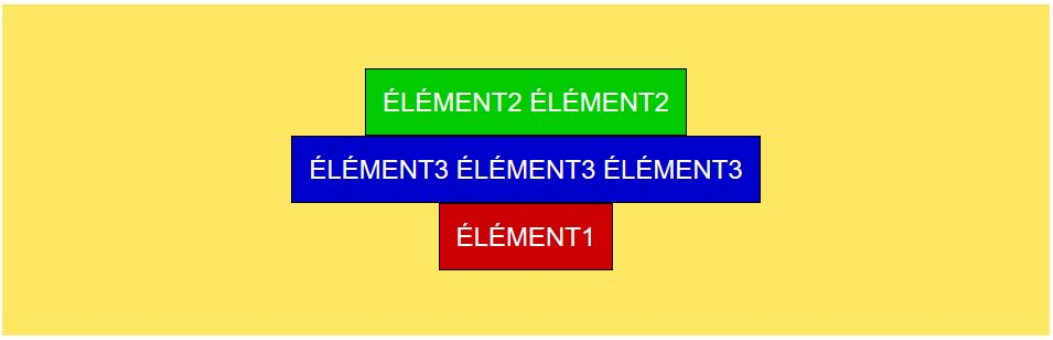
Il est possible de modifier l'ordre des éléments avec la propriété order dont la **valeur est un nombre** qui désigne la place de l'élément. Cette propriété s'applique sur l'élément enfant.

Sur les blocs précédemment construits modifiez l'ordre des éléments. Utilisez les pseudos classes du type nth-of-type pour cibler les éléments enfants.

### 7.a. Blocs ordonnés selon l'axe horizontal flex-direction :row



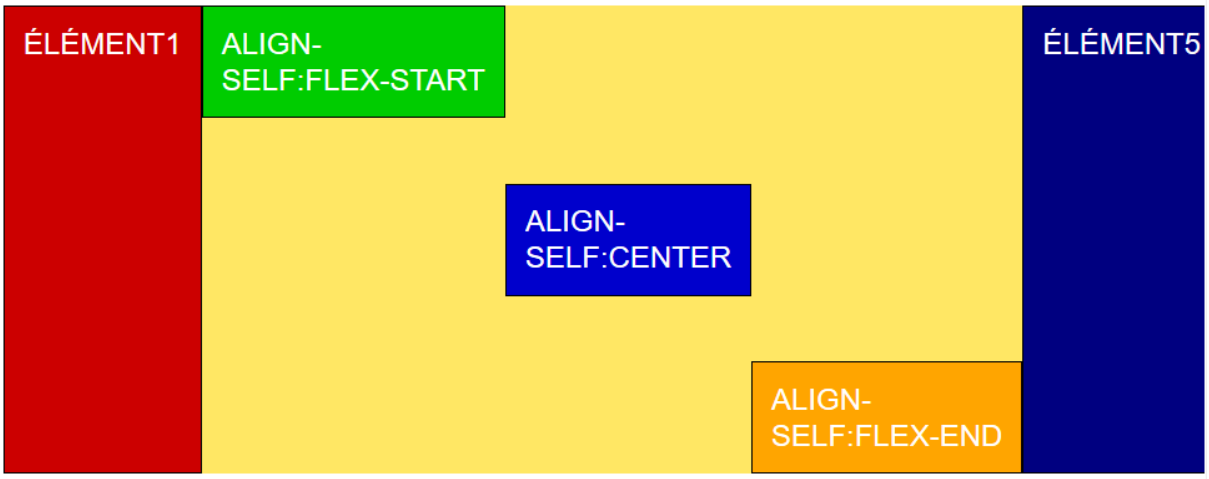

### 7.a. Blocs ordonnés selon l'axe vertical flex-direction :column



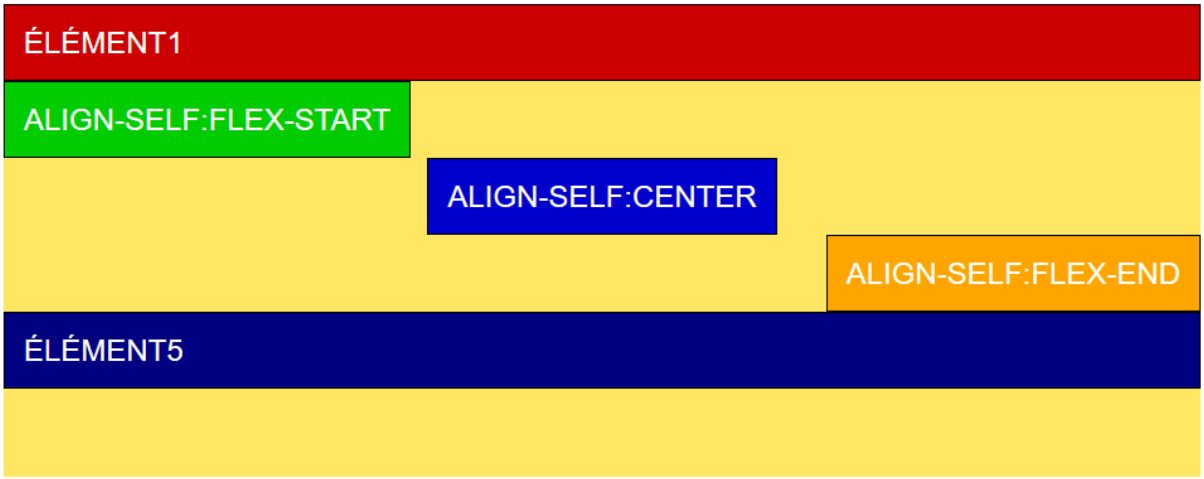

## 8. La propriété align-self

**La propriété align-self** appliqué sur un élément enfant permet de décaler cet élément par rapport aux autres.

### 8.a. Modifier l'alignement de trois blocs avec align-self et les valeurs flex-start, center et flex-end lorsque l'axe défini est horizontal




### 8.a. Modifier l'alignement de trois blocs avec align-self et les valeurs flex-start, center et flex-end lorsque l'axe défini est vertical



## 9. La propriété flex.

La propriété flex est la propriété raccourcie de trois propriétés :

- **flex-grow :** capacité pour un élément à s'étirer dans l'espace restant (valeur par défaut : 0). C'est sur cette propriété que l'on va travailler pour rendre les éléments flexibles. Il suffira de lui donner une valeur supérieure à zéro pour que l'élément ciblé occupe l'espace restant au sein du flex-container.
- **flex-shrink :** capacité pour un élément à se contracter si nécessaire (valeur par défaut : 1)
- **flex-basis :** taille initiale de l'élément avant que l'espace restant ne soit distribué (valeur par défaut : auto)

Pour rendre un élément flexible, il suffit de lui attribuer une valeur de flex-grow (ou flex en raccourci) supérieure à zéro. Cet élément occupera alors l'espace restant au sein du flex-container.

**#container1. flex:initial (valeur par défaut sur tous les éléments)**


**#container2. flex:1 / le dernier élément**


**#container3. flex:1 / les 4e et dernier élément**


**#container4. flex:1 / tous les éléments**


**#container5. flex-basis:150px; / tous les éléments**


**#container6. flex-basis:60px / tous les éléments et flex-grow:1 / élément 2**


Sites sur le sujet

https://css-tricks.com/snippets/css/a-guide-to-flexbox/