Le layout avec Flexbox

Développement Web

Plan du cours

- Les conteneurs flexibles
- La direction
- Les propriétés wrap et flow
- L’alignement et le stretch
- Les propriétés des Flexbox
- Exercices

- À la fin de ce cours vous serez en mesure de :
  - Créer le layout complet d’une page en utilisant les Flexbox.

Introduction

Les container flexibles

la direction

les propriétés wrap et flow

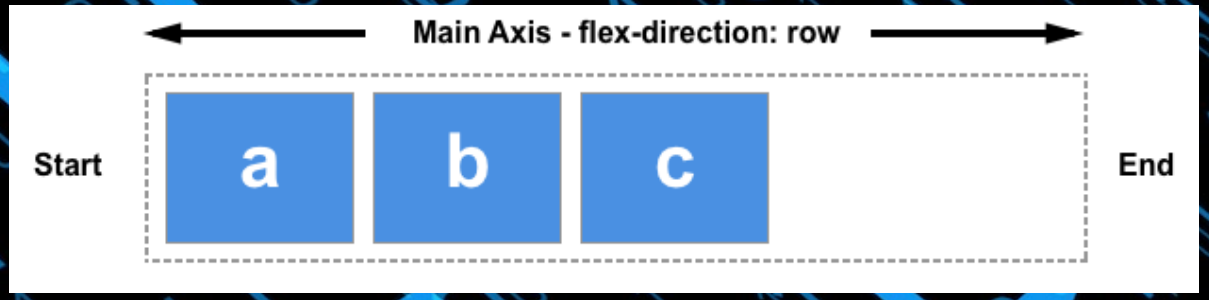
Introduction

- **`Flexbox`** permet de gérer plus facilement la disposition des éléments HTML en permettant facilement de les aligner et de définir l'espace les entourant, et ce même si leur taille est inconnue ou dynamique.
- La première idée derrière les **`Flexbox`** est que les **conteneurs** modifient la hauteur et la largeur des éléments qu'ils contiennent pour optimiser l'espace libre.
  - Un conteneur flexible peut ainsi étendre ou diminuer la taille des éléments HTML qu'il contient.
- **`Flexbox`** permet de définir des propriétés sur les **conteneurs flexibles** et sur les éléments qu'ils contiennent, les éléments flex.

Définir un conteneur et son axe principal

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

- Les éléments peuvent donc s'aligner :
  - horizontalement de gauche vers la droite avec **row**
  - horizontalement de droite vers la gauche **row-reverse**
  - verticalement de haut en bas avec **column**
  - verticalement de bas en haut avec **column-reverse**



Aller ou non à la ligne suivante ?

- Une propriété permet de définir si les éléments peuvent ou non aller à la ligne : **`flex-wrap.`**

```
.container{
  flex-wrap: wrap;
}
```

- Ainsi les éléments contenus peuvent :
  - être sur une ligne maximum obligatoirement avec **nowrap**
  - être sur plusieurs lignes avec plus d'éléments sur la ligne la plus haute **wrap**
  - être sur plusieurs lignes avec plus d'éléments sur la ligne la plus basse **wrap-reverse**
- A noter qu'avec nowrap, la taille des éléments sera modifiée pour tenir sur la ligne.
  - **Si par exemple l'axe principal est horizontal, alors leur width sera diminuée si l'ensemble des éléments ne tient pas sur la ligne**.
  - **Leur width diminuera jusqu'à une largeur minimale qui correspond à la largeur de leur contenu. Si il n'y a toujours pas assez de place sur la ligne, les éléments dépasseront de la ligne et seront hors zone d'affichage**.

Le raccourci `flex-flow`

- Une propriété permet de définir flex-direction et flex-wrap en une seule fois, il s'agit de la propriété **`flex-flow`**.
- La première valeur passée est utilisée pour flex-direction et la seconde pour flex-wrap.
- On peut ainsi écrire par exemple directement :

```
.container{
  flex-flow: row wrap;
}
```


02-Flex Flow.html

L’alignement et stretch

L'alignement et l'espacement sur l'axe principal

- **`justify-content`** permet de définir comment l'espace libre est réparti entre les éléments
  - les éléments sont resserrés au début de la ligne avec **flex-start**
  - les éléments sont resserrés à la fin de la ligne avec **flex-end**
  - les éléments sont centrés avec **center**
  - les éléments sont distribués sur une ligne avec une distance égale entre eux, le premier élément et le dernier élément sont collés au début et à la fin avec **space-between**
  - l'espace entre les éléments est réparti avec une unité d'espace de chaque côté de chaque élément avec **space-around**
  - l'espace entre les éléments est réparti de manière égale avec **space-evenly**

03-justify-content.html

Le comportement des éléments sur l'axe opposé

- **`align-items`** définit le comportement des éléments sur l'axe opposé.
  - les éléments sont resserrés au début de l'axe secondaire avec **flex-start**
  - les éléments sont au début de l'axe secondaire alignés le long de leur baseline avec **baseline**
  - les éléments sont resserrés à la fin de l'axe secondaire avec **flex-end**
  - les éléments sont centrés sur l'axe secondaire avec **center**
  - les éléments sont étirés dans l'espace disponible avec **stretch**, **tout en respectant les contraintes de taille imposées avec height, width, max-height, min-height**

Lorsqu'il y a plusieurs lignes dans un container

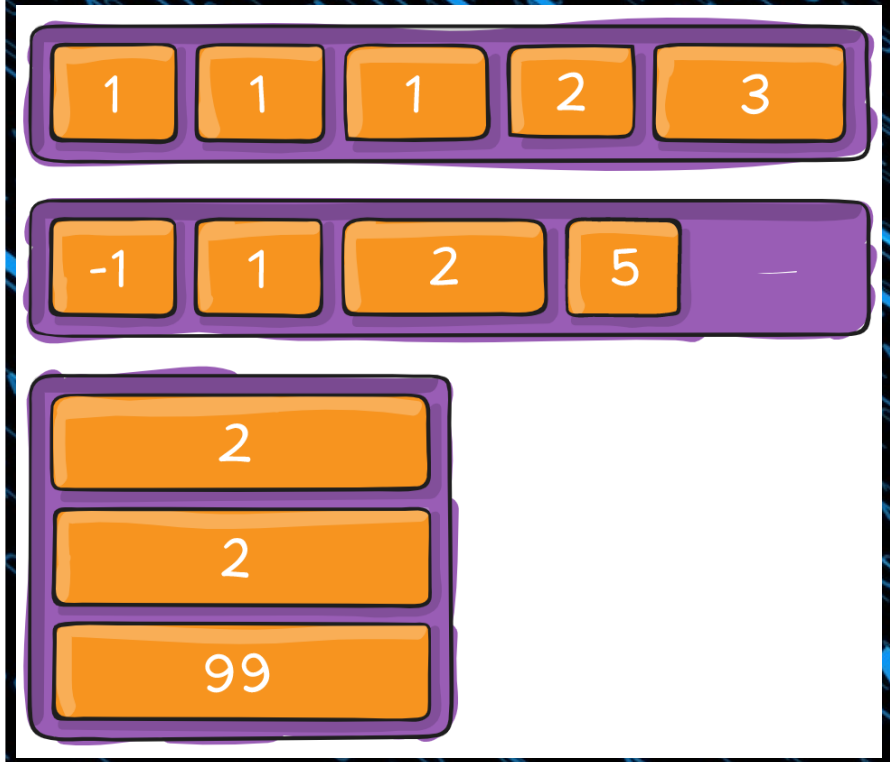
- **`align-content`** définit l'espacement entre les lignes d'un conteneur flexible à l'intérieur duquel il existe plusieurs lignes d'éléments.
  - Cette propriété agit comme justify-content mais sur l'autre axe.
    - C'est-à-dire que si les éléments sont alignés horizontalement, et que le conteneur contient plusieurs lignes d'éléments, alors elle **définit l'espacement vertical entre les lignes.**
  - les lignes sont resserrés au début de l'axe secondaire avec **flex-start**
  - les lignes sont resserrés à la fin de l'axe secondaire avec **flex-end**
  - les lignes sont centrés avec **center**
  - les lignes sont réparties avec la première collée au début et la dernière à la fin de l'axe secondaire **space-between**.
  - les lignes sont réparties avec un espacement identique entre chaque **space-around**.
  - les lignes sont étirés dans l'espace disponible avec **stretch**

05-align-content.html

Les propriétés de Flexbox

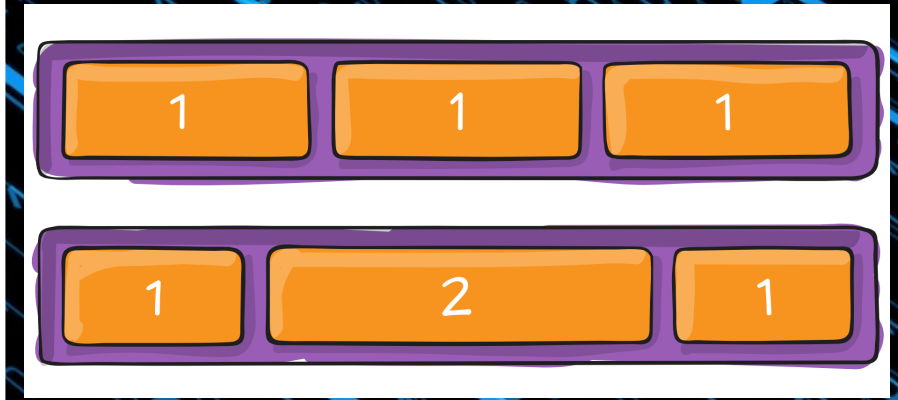
La propriété **order**

- Par défaut les éléments HTML s'affichent dans l'ordre dans lesquels ils sont déclarés.
- La propriété **order** appliquée sur un élément permet de définir son ordre suivant l'orientation de l'axe principal.
- L'ordre par défaut est 0, un élément auquel on attribue un order de -1 viendra donc avant. Un élément avec un order de 1 viendra après.



La propriété `flex-grow`

- L'élément s'étirera et occupera l'espace disponible sur l'axe principal ou une partie de cet espace si les autres éléments peuvent également s'étirer.
- Si tous les éléments ont un flex-grow défini à 1, l'espace disponible sera alors partagé de façon égale entre les éléments qui s'étireront pour occuper toute la place disponible sur l'axe principal.
- Un élément à 2 prendra (1+2+1) / 2 = 50%



La propriété `flex-shrink`

- **La propriété flex-shrink défini la possibilité pour un élément flexible de rétrécir, si nécessaire**.
- S'il n'y a pas suffisamment d'espace dans le conteneur, l'élément peut alors rétrécir s'il a sa propriété flex-shrink supérieure à 0.
- **La valeur par défaut est 1**, ce qui signifie que par défaut, comme nous l'avions vu précédemment, les éléments flexibles vont rétrécir si ils n'ont pas la place disponible.
- Si tous les éléments ont un flex-shrink défini à 1, chaque élément occupera le même espace dans le conteneur.
- Si vous donnez à l'un des éléments une valeur de 2, cet élément rétrécira deux fois plus vite que les autres.

Taille initiale d’un élément `flex`

- La propriété **flex-basis** définit la taille initiale d'un élément flex avant que l'espace disponible ne soit utilisé selon les autres propriétés flex définies.
- La valeur par défaut est auto. Dans ce cas, si une taille pour les éléments est définie elle sera utilisée, sinon c'est la taille du contenu des éléments flexibles qui sera utilisée comme base.

```
.flex-element {
  flex-basis: 100px;
}
```

Le raccourci **flex**

- La propriété **flex** permet de définir en une seule fois les propriétés **flex-grow**, **flex-shrink** et **flex-basis**.
- Par défaut nous avons donc :

```
flex: 0 1 auto;
```

- **flex: auto** sera l'équivament de :

```
flex: 1 1 auto;
```

- **flex: none** sera l'équivament de :

```
flex: 0 0 auto;
```

La propriété `align-self`

- Permet à des éléments flex de **passer outre** les alignements spécifiés par align-items sur le container.
  - l'élément est resserré au début de l'axe secondaire avec **flex-start**
  - l'élément est au début de l'axe secondaire alignés le long de leur baseline avec **baseline**
  - l'éléments est resserré à la fin de l'axe secondaire avec **flex-end**
  - l'éléments est centrés sur l'axe secondaire avec **center**
  - l'élément est étiré dans l'espace disponible avec **stretch**
    - **tout en respectant les contraintes de taille imposées: height, width, max-height, min-height**

Lecture obligatoire

CSS-TRICKS – A Complete Guide to Flexbox

Exercices avec Froggy

Faire une première fois l'exercice en mode débutant

Refaire l'exercice en mode expert

[Flexbox Froggy]