

Flexbox

1. Qu'est-ce que Flexbox ?

Flexbox, ou "Flexible Box", est un modèle de mise en page introduit en CSS. Il est conçu pour gérer l'alignement, la direction et la distribution des éléments dans un conteneur, même si leurs tailles sont dynamiques ou inconnues. Contrairement à Grid, Flexbox fonctionne principalement sur un seul axe (horizontal ou vertical).

1.1. Les principaux concepts de Flexbox

Flexbox repose sur un **conteneur flex** (élément parent) et des **éléments flex** (enfants).

Les propriétés du conteneur contrôlent leur disposition, tandis que les propriétés des enfants ajustent leur comportement.

1.1.1. Conteneur flex

Un conteneur flex est activé avec la propriété `display: flex;`. Voici les propriétés essentielles illustrées par des exemples de code.

1.1.1.1. Propriétés principales (avec exemples de code HTML et CSS)

1.2. Exemple 1 : `display` et direction des éléments avec `flex-direction`

HTML :

html

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

CSS :

```
.flex-container {  
  display: flex;  
  flex-direction: row; /* Aligne horizontalement (par défaut) */  
  gap: 10px; /* Espace entre les éléments */  
}  
  
.flex-container div {  
  background-color: lightblue;  
  padding: 10px;  
  text-align: center;  
}
```

Explication : Par défaut, les éléments suivent une direction de gauche à droite (`row`). On peut la changer en colonne avec `flex-direction: column;`.

1.3. Exemple 2 : Espacement avec `justify-content`

HTML :

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

CSS :

```
.flex-container {  
  display: flex;  
  justify-content: space-between; /* Espace équitablement entre les éléments */  
}  
  
.flex-container div {  
  background-color: lightcoral;  
  padding: 10px;  
  text-align: center;  
}
```

Explication : La propriété `justify-content` gère l'espacement sur l'axe principal. D'autres valeurs possibles :

- `center` : Centre tous les éléments.
- `space-around` : Égalise les espaces autour chaque élément.

1.4. Exemple 3 : Alignement sur l'axe croisé avec `align-items`

HTML :

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

CSS :

```
.flex-container {
  display: flex;
  align-items: center; /* Aligne les enfants au centre verticalement */
  height: 150px; /* Nécessaire pour voir l'alignement verticalement */
}

.flex-container div {
  background-color: lightgreen;
  padding: 10px;
  text-align: center;
}
```

Explication : La propriété `align-items` gère l'alignement des éléments sur l'axe croisé (perpendiculaire). Autres options :

- `flex-start` : Aligne en haut.
- `flex-end` : Aligne en bas.

1.5. Exemple 4 : Contrôle individuel avec `align-self`

HTML :

```
<div class="flex-container">
  <div>1</div>
  <div class="custom-align">2</div>
  <div>3</div>
</div>
```

CSS :

```
.flex-container {
  display: flex;
  align-items: flex-start; /* Par défaut, tous sont alignés en haut */
  height: 150px;
}

.flex-container .custom-align {
  align-self: flex-end; /* Aligne cet élément en bas */
}
```

```

}

.flex-container div {
  background-color: lightpink;
  padding: 10px;
  text-align: center;
}

```

Explication : `align-self` permet de modifier l'alignement d'un élément spécifique, indépendamment des autres.

1.6. Exemple 5 : Taille flexible avec `flex-grow`

HTML :

html

```

<div class="flex-container">
  <div class="grow">1</div>
  <div>2</div>
  <div>3</div>
</div>

```

CSS :

CSS

```

.flex-container {
  display: flex;
}

.flex-container .grow {
  flex-grow: 2; /* Cet élément occupe 2 fois plus d'espace */
}

.flex-container div {
  background-color: lightskyblue;
  padding: 10px;
  text-align: center;
}

```

Explication : La propriété `flex-grow` contrôle la façon dont un élément occupe l'espace disponible. Une valeur plus grande attribue plus d'espace.

1.7. Exemple 6 : Ordre des éléments avec `order`

HTML :

html

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div class="reorder">3</div>
</div>
```

CSS :

css

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

.flex-container .reorder {
  order: -1; /* Cet élément apparaît en premier */
}

.flex-container div {
  background-color: lightyellow;
  padding: 10px;
  text-align: center;
}
```

Explication : `order` permet de modifier l'ordre naturel des éléments dans le flux de Flexbox.

2. Tableaux des propriétés flexbox

Catégorie	Propriété	Valeurs possibles	Description
Conteneur parent	<code>display</code>	<code>flex</code> , <code>inline-flex</code>	Définit un conteneur Flexbox. C'est la propriété de base pour activer le modèle Flexbox.

Catégorie	Propriété	Valeurs possibles	Description
	<code>flex-direction</code>	<code>row</code> , <code>row-reverse</code> , <code>column</code> , <code>column-reverse</code>	Définit la direction principale (axe principal) pour l'agencement des éléments.
	<code>flex-wrap</code>	<code>nowrap</code> , <code>wrap</code> , <code>wrap-reverse</code>	Définit si les éléments doivent être regroupés sur une seule ligne ou s'ils doivent passer à une nouvelle ligne en cas de manque d'espace.
	<code>justify-content</code>	<code>flex-start</code> , <code>flex-end</code> , <code>center</code> , <code>space-between</code> , <code>space-around</code> , <code>space-evenly</code>	Contrôle l'alignement des éléments le long de l'axe principal.
	<code>align-items</code>	<code>flex-start</code> , <code>flex-end</code> , <code>center</code> , <code>baseline</code> , <code>stretch</code>	Contrôle l'alignement des éléments sur l'axe transversal.
	<code>align-content</code>	<code>flex-start</code> , <code>flex-end</code> , <code>center</code> , <code>space-between</code> , <code>space-around</code> , <code>stretch</code>	Contrôle l'espacement entre les lignes du conteneur si plusieurs lignes sont présentes.
	<code>gap</code>	Valeur en longueur (par exemple <code>px</code> , <code>%</code> , <code>em</code> , etc.)	Définit l'espace entre les lignes et les colonnes des éléments enfants.
Élément enfant	<code>order</code>	Nombre entier (par défaut <code>0</code>)	Définit l'ordre dans lequel un élément apparaît dans le conteneur Flexbox, indépendamment de l'ordre du DOM.
	<code>flex-grow</code>	Nombre positif (par défaut <code>0</code>)	Détermine la capacité d'un élément à croître pour occuper l'espace disponible, en proportion par rapport aux autres éléments.
	<code>flex-shrink</code>	Nombre positif (par défaut <code>1</code>)	Détermine la capacité d'un élément à rétrécir lorsqu'il manque d'espace.
	<code>flex-basis</code>	Valeur en longueur (par exemple <code>px</code> , <code>%</code> , <code>em</code> , etc.) ou <code>auto</code>	Spécifie la taille initiale d'un élément avant que l'espace disponible ne soit réparti.
	<code>align-self</code>	<code>auto</code> , <code>flex-start</code> , <code>flex-end</code> , <code>center</code> , <code>baseline</code> , <code>stretch</code>	Permet d'overrider l'alignement défini par <code>align-items</code> pour un élément individuel.

3. À RETENIR

- **Définition** : Flexbox organise et aligne les éléments sur un seul axe.
- Propriétés essentielles :
 - `display: flex` : Active le mode flex.
 - `flex-direction` : Définit l'axe principal (horizontal ou vertical).
 - `justify-content` : Gère l'espacement sur l'axe principal (ex : centrer, espacer).
 - `align-items` : Contrôle l'alignement sur l'axe croisé.
 - `flex-grow` : Fait grandir un élément en fonction de l'espace disponible.
 - `order` : Modifie l'ordre naturel des éléments.