



# CSS : les alignements

---

Par Robert DIASSÉ

---

## Le Flexbox et le Positionnement en CSS

Le positionnement des éléments en CSS est essentiel pour créer des mises en page flexibles et réactives. Deux techniques importantes sont largement utilisées : le Flexbox et les propriétés de positionnement.

### 1. Flexbox :

Le Flexbox est un modèle de disposition qui permet de disposer les éléments de manière dynamique dans un conteneur, en utilisant des lignes et des colonnes. Voici les concepts clés du Flexbox :

**Conteneur Flex** : L'élément parent qui contient les éléments flexibles. Vous définissez un conteneur flex en ajoutant `display: flex;` ou `display: inline-flex;` à son style CSS.

**Axe Principal et Secondaire** : Le Flexbox a deux axes : l'axe principal et l'axe secondaire. L'axe principal est déterminé par la direction de `flex-direction` (horizontale par défaut), tandis que l'axe secondaire est perpendiculaire à l'axe principal. (horizontal = `row`, vertical = `column`, horizontal-inversé = `row-reverse`, vertical-inversé = `column-reverse`)

Exemple

```
div{
  display: flex;
  flex-direction: row; // horizontal par défaut
}
```

**Alignement** : Vous pouvez aligner les éléments flexibles sur l'axe principal avec la propriété `justify-content` avec les valeurs suivant:

- `flex-start` : Les éléments sont alignés au début du conteneur.
- `flex-end` : Les éléments sont alignés à la fin du conteneur.
- `center` : Les éléments sont centrés dans le conteneur.
- `space-between` : Les éléments sont répartis de manière égale avec un espacement entre eux, le premier élément aligné au début du conteneur et le dernier élément aligné à la fin.
- `space-around` : Les éléments sont répartis de manière égale avec un espacement autour d'eux.

et sur l'axe secondaire en utilisant la propriété `align-items` avec les valeurs suivantes:

- stretch : Les éléments sont étirés pour remplir le conteneur.
- flex-start : Les éléments sont alignés au début du conteneur.
- flex-end : Les éléments sont alignés à la fin du conteneur.
- center : Les éléments sont centrés dans le conteneur.
- baseline : Les éléments sont alignés basé sur leur ligne de base.

**Ordre des Éléments** : Vous pouvez contrôler l'ordre des éléments flexibles avec la propriété `order`.

**Largeur Flexible** : Les éléments flexibles peuvent avoir une largeur flexible en utilisant la propriété `flex`.  
exemple:

HTML :

```
<div class="container">
  <div class="item item1">Premier élément</div>
  <div class="item item2">Deuxième élément</div>
  <div class="item item3">Troisième élément</div>
</div>
```

CSS :

```
.container {
  display: flex;
}

.item {
  border: 1px solid black;
  padding: 10px;
  margin: 5px;
}

.item1 {
  order: 2; /* Change l'ordre du premier élément */
}

.item2 {
  flex: 1; /* Définit une largeur flexible pour le deuxième élément */
}

.item3 {
  order: 1; /* Change l'ordre du troisième élément */
}
```

Dans cet exemple :

- Le premier élément (`item1`) est déplacé à la deuxième position en utilisant `order: 2`.
- Le deuxième élément (`item2`) aura une largeur flexible grâce à `flex: 1`, ce qui signifie qu'il va remplir l'espace disponible dans le conteneur après avoir pris en compte les dimensions des autres éléments.

- Le troisième élément (`item3`) est déplacé à la première position en utilisant `order: 1`.

## 2. Positionnement :

Le positionnement en CSS permet de placer les éléments de manière précise sur une page web. Voici les principales propriétés de positionnement :

- **Position Relative** : L'élément est positionné relativement par rapport à sa position normale. Vous pouvez déplacer l'élément en utilisant les propriétés `top`, `right`, `bottom` et `left`.

```
.relative {  
  position: relative;  
  top: 10px;  
  left: 20px;  
}
```

l'élément avec la classe `.relative` sera déplacé de 10 pixels vers le bas et de 20 pixels vers la droite par rapport à sa position normale.

- **Position Absolue** : L'élément est positionné par rapport à son premier parent positionné. Si aucun parent positionné n'existe, il est positionné par rapport au bloc contenant. Utilisez `position: absolute`;

```
.absolute {  
  position: absolute;  
  top: 50px;  
  left: 50px;  
}
```

l'élément avec la classe `.absolute` sera positionné par rapport à son premier parent positionné.

- **Position Fixe** : L'élément est positionné par rapport à la fenêtre du navigateur. Il reste fixe même lorsque vous faites défiler la page. Utilisez `position: fixed`;

```
.fixed {  
  position: fixed;  
  top: 0;  
  right: 0;  
}
```

l'élément avec la classe `.fixed` restera positionné dans le coin supérieur droit de la fenêtre du navigateur même lorsque vous faites défiler la page.

- **Position Collante** : L'élément se comporte comme s'il était relatif jusqu'à ce qu'il atteigne un certain point lors du défilement, où il devient fixe. Utilisez `position: sticky`;

```
.sticky {  
  position: sticky;  
  top: 20px;  
}
```

l'élément avec la classe `.sticky` se comportera comme un élément relatif jusqu'à ce qu'il atteigne 20 pixels depuis le haut du conteneur parent, après quoi il restera collé à cet endroit pendant le défilement.

### 3. Unités de Mesure :

Pour dimensionner les éléments, vous pouvez utiliser différentes unités comme :

- **Pixels (px)** : Une unité de mesure fixe.
- **Pourcentage (%)** : Relative à la taille du parent. Par exemple, si un élément enfant a une largeur définie à 50%, cela signifie qu'il occupe 50% de la largeur de son parent.
- **Viewport Height (vh)** : Relative à la hauteur de la fenêtre du navigateur. Par défaut, 1 vh équivaut à 1% de la hauteur de la fenêtre du navigateur. Donc, si la fenêtre du navigateur a une hauteur de 1000 pixels, 1 vh équivaudra à 10 pixels.
- **Viewport Width (vw)** : Relative à la largeur de la fenêtre du navigateur. De manière similaire, 1 vw équivaut à 1% de la largeur de la fenêtre du navigateur par défaut.
- **Rem** : Relative à la taille de la police racine. Par défaut, 1 rem est égal à la taille de la police racine définie dans le fichier CSS ou par le navigateur. Habituellement, la valeur par défaut de la police racine est de 16 pixels.
- **Em** : Relative à la taille de la police de l'élément. Par défaut, 1 em est égal à la taille de la police de l'élément lui-même. Si la taille de la police de l'élément parent est de 16 pixels, alors 1 em sera égal à 16 pixels.

Ces techniques et propriétés permettent de créer des mises en page flexibles et bien alignées pour vos projets web.