

## Structurez votre page

### Structurez votre page

Les balises structurantes de HTML5

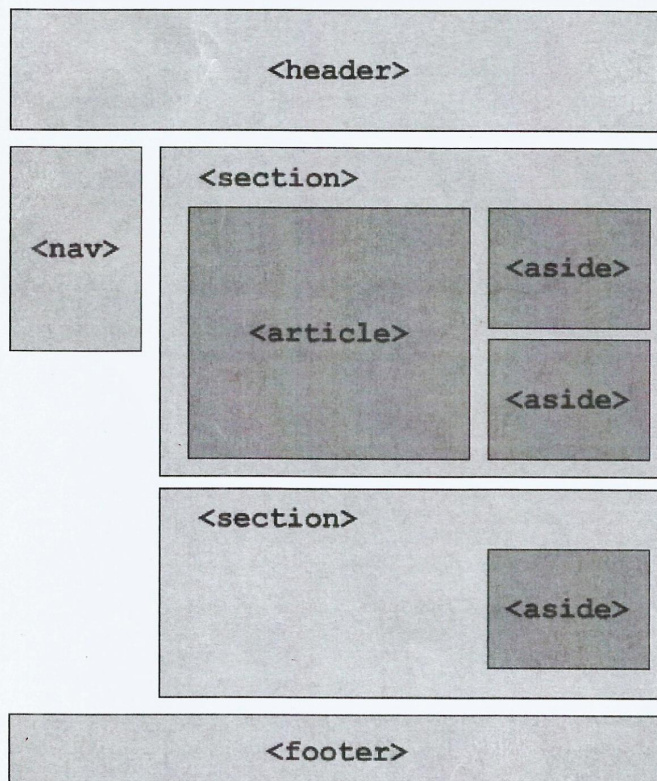
Il s'agit des nouvelles balises introduites par HTML5 pour structurer nos pages :

- `<header>` : l'en-tête
- `<footer>` : le pied de page
- `<nav>` : Il regroupe tous les principaux liens de navigation
- `<section>` : une section de page, sert à regrouper des contenus en fonction de leur thématique.
- `<aside>` : informations complémentaires
- `<article>` : un article indépendant

Ces balises peuvent être imbriquées les unes dans les autres. Ainsi, une section peut avoir son propre en-tête.

Ces balises ne s'occupent pas de la mise en page. Elles servent seulement à indiquer à l'ordinateur le sens du texte qu'elles contiennent. On pourrait très bien placer l'en-tête en bas de la page, si on le souhaitait.

#### Résumé



### ***Aligner sur l'axe secondaire***

align-items

Il permet de changer l'alignement des éléments sur l'axe secondaire. Il peut prendre ces valeurs :

- stretch : les éléments sont étirés sur tout l'axe (valeur par défaut) ;
- flex-start : alignés au début ;
- flex-end : alignés à la fin ;
- center : alignés au centre ;
- baseline : alignés sur la ligne de base (semblable à flex-start).

Exemple :

```
#conteneur {
  display: flex;

  justify-content: center;
  align-items: center;

  height: 400px;
  border: 2px blue solid;
}
```

Une autre manière pour obtenir le centrage vertical et horizontal est de :

- Dire que votre conteneur est une flexbox et
- établissez des marges automatiques sur les éléments à l'intérieur

Exemple :

```
#conteneur {
{
  display: flex;
}

.element_1, .element_2, element_3
{
  margin: auto;
}
```

### ***Aligner un seul élément***

align-self

Il permet de faire une exception pour un seul des éléments sur l'axe secondaire :

```
#conteneur {
  display: flex;
  justify-content: center;
  align-items: center;

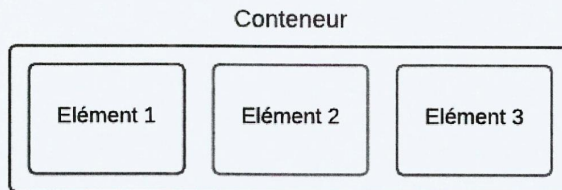
  height: 400px;
  border: 2px blue solid;
}

.element_2 {
  background-color: blue;
```

# Flexbox

## Un conteneur, des éléments

Le principe de la mise en page avec Flexbox est simple : vous définissez un conteneur, et à l'intérieur vous placez plusieurs éléments. Sur une même page web, vous pouvez sans problème avoir plusieurs conteneurs.



Le conteneur est une balise HTML, et les éléments sont d'autres balises HTML à l'intérieur :

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <div id="conteneur">
      <div class="element_1">Element 1 </div>
      <div class="element_2">Element 2</div>
      <div class="element_3">Element 3</div>
    </div>
  </body>
</html>
```

```
#conteneur {
  display: flex;
}

.element_1 {
  background-color: chocolate;
  width: 200px;
  height: 100px;
}

.element_2 {
  background-color: blue;
  width: 200px;
  height: 100px;
}

.element_3 {
  background-color: chartreuse;
  width: 200px;
  height: 100px;
}
```



Découvrons maintenant Flexbox par l'application des propriétés Flexbox au conteneur :

`display: flex;`

Les blocs se placent par défaut côte à côte.

`flex-direction: column;`

Positionner les éléments verticalement, horizontalement ou encore les inverser. Il peut prendre les valeurs suivantes :

- `row` : organisés sur une ligne (par défaut) ;
- `column` : organisés sur une colonne ;
- `row-reverse` : organisés sur une ligne, mais en ordre inversé ;
- `column-reverse` : organisés sur une colonne, mais en ordre inversé.

### ***Le retour à la ligne***

`flex-wrap: wrap;`

Par défaut, les blocs essaient de rester sur la même ligne s'ils n'ont pas la place. Si vous voulez, vous pouvez demander à ce que les blocs aillent à la ligne lorsqu'ils n'ont plus la place, avec `flex-wrap` qui peut prendre ces valeurs :

- `nowrap` : pas de retour à la ligne (par défaut) ;
- `wrap` : les éléments vont à la ligne lorsqu'il n'y a plus la place ;
- `wrap-reverse` : les éléments vont à la ligne, lorsqu'il n'y a plus la place, en sens inverse.

### ***Aligner sur l'axe principal***

`justify-content`

Partons sur des éléments organisés horizontalement. Pour changer leur alignement, on va utiliser `justify-content`, qui peut prendre ces valeurs :

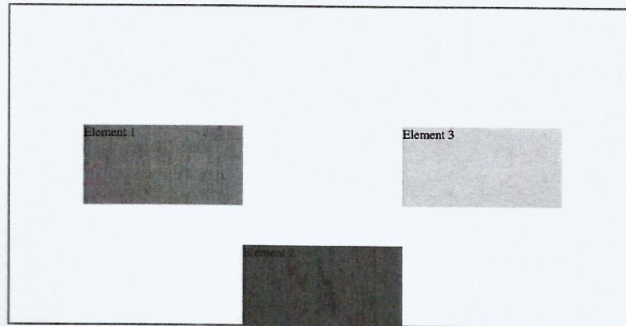
- `flex-start` : alignés au début (par défaut) ;
- `flex-end` : alignés à la fin ;
- `center` : alignés au centre ;
- `space-between` : les éléments sont étirés sur tout l'axe (il y a de l'espace entre eux) ;
- `space-around` : idem, les éléments sont étirés sur tout l'axe, mais ils laissent aussi de l'espace sur les extrémités.

**ça marche aussi si vos éléments sont dans une direction verticale. Voici un exemple :**

```
#conteneur {  
  display: flex;  
  flex-direction: column;  
  flex-wrap: wrap;  
  justify-content: center;  
  
  height: 400px;  
  border: 2px blue solid;  
}
```

```
width: 200px;
height: 100px;
align-self: flex-end;
}
```

Résultat :



### Répartir plusieurs lignes

align-content

Si vous avez plusieurs lignes dans votre Flexbox, vous pouvez choisir comment celles-ci seront réparties avec align-content .

➔ Cette propriété n'a aucun effet s'il n'y a qu'une seule ligne dans la Flexbox.

### Exemple

```
<div id="conteneur">
  <div class="element">Element 1</div>
  <div class="element">Element 2</div>
  <div class="element">Element 3</div>
  <div class="element">Element 4</div>
  <div class="element">Element 5</div>
  <div class="element">Element 6</div>
  <div class="element">Element 7</div>
  <div class="element">Element 8</div>
  <div class="element">Element 9</div>
  <div class="element">Element 10</div>
  <div class="element">Element 11</div>
  <div class="element">Element 12</div>
</div>
```

J'autorise mes éléments à aller à la ligne avec flex-wrap :

```
#conteneur
{
  display: flex;
  flex-wrap: wrap;

  border: 1px blue solid;

  height: 600px;
  align-content: space-between;
}
```



```
.element
{
    border: 1px blue solid;
    width: 200px;
    height: 100px;
}
```

*align-content* peut prendre ces valeurs :

- **flex-start** : les éléments sont placés au début ;
- **flex-end** : les éléments sont placés à la fin ;
- **center** : les éléments sont placés au centre ;
- **space-between** : les éléments sont séparés avec de l'espace entre eux ;
- **space-around** : idem, mais il y a aussi de l'espace au début et à la fin ;
- **stretch** (par défaut) : les éléments s'étirent pour occuper tout l'espace.

### Modifier l'ordre des éléments en CSS

Sans changer le code HTML, nous pouvons modifier l'ordre des éléments en CSS grâce à la propriété *order* .

Reprenons l'exemple précédent :

```
<div id="conteneur">
    <div class="element">Element 1</div>
    <div class="element">Element 2</div>
    <div class="element">Element 3</div>
</div>
```

```
...
.element:nth-child(1) {
    order: 2;
}
.element:nth-child(2) {
    order: 3;
}
.element:nth-child(3) {
    order: 1;
}
...
```

### faire grossir ou maigrir les éléments

Avec la propriété *flex* , nous pouvons permettre à un élément de grossir pour occuper tout l'espace restant.

```
...
.element:nth-child(1) {
    order: 2;
    flex: 1;
}
...
```

## En résumé

- Il existe plusieurs techniques pour positionner les blocs sur la page. Flexbox est la technique la plus récente et de loin la plus puissante, que je vous recommande d'utiliser.
- Le principe de Flexbox est d'avoir un conteneur, avec plusieurs éléments à l'intérieur. Avec `display: flex`; sur le conteneur, les éléments à l'intérieur sont agencés en mode Flexbox (horizontalement, par défaut).
- Flexbox peut gérer toutes les directions. Avec `flex-direction`, on peut indiquer si les éléments sont agencés horizontalement (par défaut) ou verticalement. Cela définit ce qu'on appelle l'*axe principal*.
- L'alignement des éléments se fait sur l'axe principal avec `justify-content`, et sur l'axe secondaire avec `align-items`.
- Avec `flex-wrap`, on peut autoriser les éléments à revenir à la ligne s'ils n'ont plus d'espace.
- S'il y a plusieurs lignes, on peut indiquer comment les lignes doivent se répartir entre elles avec `align-content`.
- Chaque élément peut être réagencé en CSS avec `order` (pas besoin de toucher au code HTML !).
- Avec la super-propriété `flex`, on peut autoriser nos éléments à occuper plus ou moins d'espace restant.
- Flexbox, c'est cool.