

# Mise en page avancé

## Box model, flex box & grid

Quelques notions de bases sont nécessaires pour pouvoir faire de la mise en page.

Ces concepts sont décrits sur MDN ([developer.mozilla.org](https://developer.mozilla.org)) :

- Box-model : [https://developer.mozilla.org/fr/docs/Apprendre/CSS/Introduction\\_à\\_CSS/Le\\_modèle\\_de\\_boîte](https://developer.mozilla.org/fr/docs/Apprendre/CSS/Introduction_à_CSS/Le_modèle_de_boîte)
- Flexbox : [https://developer.mozilla.org/fr/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout/Concepts\\_de\\_base\\_flexbox](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Flexible_Box_Layout/Concepts_de_base_flexbox)
- Grid : [https://developer.mozilla.org/fr/docs/Web/CSS/CSS\\_Grid\\_Layout/Les\\_concepts\\_de\\_base](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Grid_Layout/Les_concepts_de_base)

## Le modèle de boîte

### box-model

Les balises HTML sont représentées par des rectangles. Il en existe plusieurs types dont les trois principaux sont **block**, **inline-block** et **inline**.

#### display: block

Les éléments “en bloc” sont des éléments qui vont s’empiler verticalement les uns à la suite des autres. Par défaut ils vont prendre **toute la largeur** disponible dans leur **parent**<sup>1</sup>. Leur hauteur, si elle n’est pas contrainte, s’adaptera au contenu. Vous pouvez modifier leurs tailles et leurs marges. C’est le **modèle par défaut** de la plupart des balises HTML.

#### display: inline

Les éléments “en ligne” sont des éléments qui vont se comporter comme les mots d’un texte. Ils vont s’intégrer au flux du texte (horizontalement) et passer à la ligne. Leur taille sera la plus petite possible pour contenir leur contenu. Vous ne pouvez pas en changer la taille, seulement les marges extérieurs.

#### display: inline-block

Ces éléments se comporteront à la fois comme des blocs : vous pourrez en modifier leur dimensions, et comme des éléments en ligne : Il ne prendront que la place nécessaire et seront disposés les uns à la suite des autres de gauche à droite (dans le sens de lecture).

---

<sup>1</sup> Parent : la balise qui contient cet élément (ou la fenêtre du navigateur)

## Les boîtes flexibles flexbox

Ce modèle de boîte va nous permettre de gérer la répartition d'éléments sur un axe horizontal (ou vertical). Il va nous permettre de créer des colonnes (ou des rangés), ce sera la méthode privilégiée pour gérer une mise en page à une dimension. Nous utiliserons les flexbox pour disposer côte à côte l'image et le texte des actualités ou encore pour disposer les conseils de lecture de la page d'accueil.

Un élément HTML dont le modèle de boîte est flexbox verra ses **enfants** alignés selon l'axe principal (horizontal par défaut).

### HTML

```
<div class="parent">
  <div class="enfant"> ... </div>
  <div class="enfant"> ... </div>
</div>
```

### CSS

```
.parent {
  display: flex;
}
```

Quelques propriétés des flexbox en vrac :

- flex-grow / flex-shrink : gérer les proportions entre les éléments
- align-items : alignement "vertical"
- justify-content : alignement "horizontal"

## Les grilles de mise en page grid

C'est l'outil le plus complet pour effectuer la mise en page. Il va offrir la possibilité de déclarer une grille dans laquelle le contenu viendra se disposer. Nous utiliserons cette méthode pour gérer les listes d'actualités. C'est un outil très puissant !

Il existe un site complet sur la mise en page avec grid (anglais) :  
**<https://gridbyexample.com/>** par **Rachel Andrew**.

D'autres ressources sont à votre disposition :

- **<https://la-cascade.io/css-grid-layout-guide-complet/>**
- **<https://www.gregoirenoyelle.com/declaration-terminologie-css-grid-layout/>**