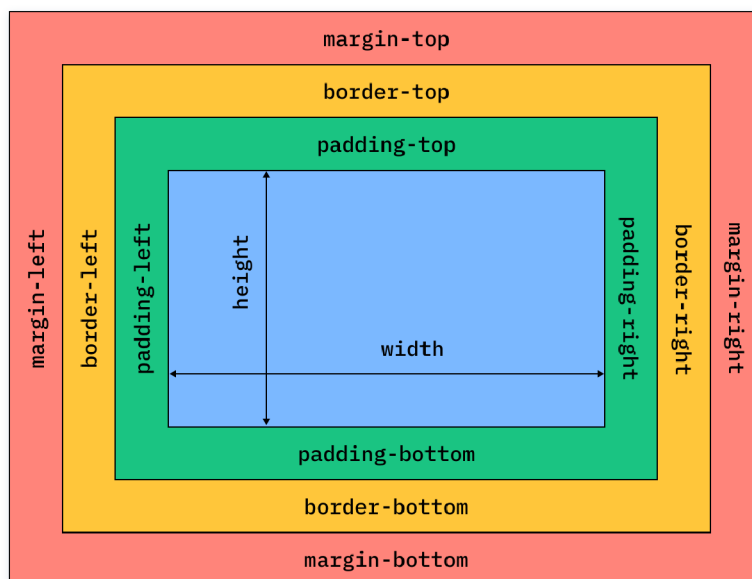


# Mémento CSS – Mise en page et positionnement

## 1. Modèle de boîte (Box Model)

Chaque élément HTML est une **boîte** :



### 1.1. Propriétés principales

- `width` , `height` , `margin` , `padding` , `border`

### 1.2. box-sizing

Contrôle la façon dont les dimensions d'une boîte sont calculées.

Si vous définissez `width: 200px` et `padding: 10px` , la largeur totale sera de 220px avec `content-box` mais toujours 200px avec `border-box` .

- `content-box` (par défaut) : inclut uniquement le contenu
- `border-box` : inclut le padding et la bordure
- Recommandé pour un contrôle plus facile des dimensions

```
.element {
  box-sizing: border-box;
  width: 200px;
  padding: 10px;
  border: 5px solid black;
  margin: 20px;
}
```

CSS

## 2. Positionnement

Type	Exemple	Description
<code>static</code>	(par défaut)	suit le flux normal
<code>relative</code>	décalage ajustable	par rapport à sa position
<code>absolute</code>	positionnée librement	par rapport au parent
<code>fixed</code>	reste à l'écran	même au défilement
<code>sticky</code>	entre fixe et relative	selon le défilement

## 3. Propriété display

- `block` : occupe la largeur entière
- `inline` : s'aligne dans la ligne
- `inline-block` : mix entre les deux
- `flex` / `grid` : pour la mise en page moderne

## 4. Flexbox – Disposition flexible

### 4.1. Exemple de base :

```
.container {
  display: flex;
  justify-content: space-between;
  align-items: center;
}
```

CSS

### 4.2. Propriétés utiles :

Propriété	Valeurs possibles	Rôle
<code>flex-direction</code>	<code>row</code> , <code>column</code>	direction principale
<code>justify-content</code>	<code>center</code> , <code>space-between</code>	alignement horizontal
<code>align-items</code>	<code>center</code> , <code>stretch</code>	alignement vertical
<code>flex-wrap</code>	<code>wrap</code>	retour automatique à la ligne

## 5. CSS Grid – Mise en page moderne

### 5.1. Exemple :

```
.container {  
  display: grid;  
  grid-template-columns: 1fr 2fr;  
  gap: 10px;  
}
```

CSS

### 5.2. Avantages :

- Organisation en **lignes et colonnes**
- Idéal pour les **layouts de tableau de bord**
- Compatible avec les médias responsives

## 6. Sélecteurs avancés

Type	Exemple	Description
Enfant direct	<code>ul &gt; li</code>	cible uniquement l'enfant direct
Voisin adjacent	<code>h2 + p</code>	cible le paragraphe suivant un titre
Frères généraux	<code>p ~ span</code>	tous les spans suivants
Attribut complexe	<code>a[href^="https"]</code>	commence par "https"

## 7. Bonnes pratiques

- Utiliser **Flex** pour les blocs simples, **Grid** pour la structure globale
- Employer `gap` plutôt que `margin` entre éléments
- Combiner `@media` pour l'adaptabilité responsive
- Isoler les styles de mise en page dans un fichier dédié