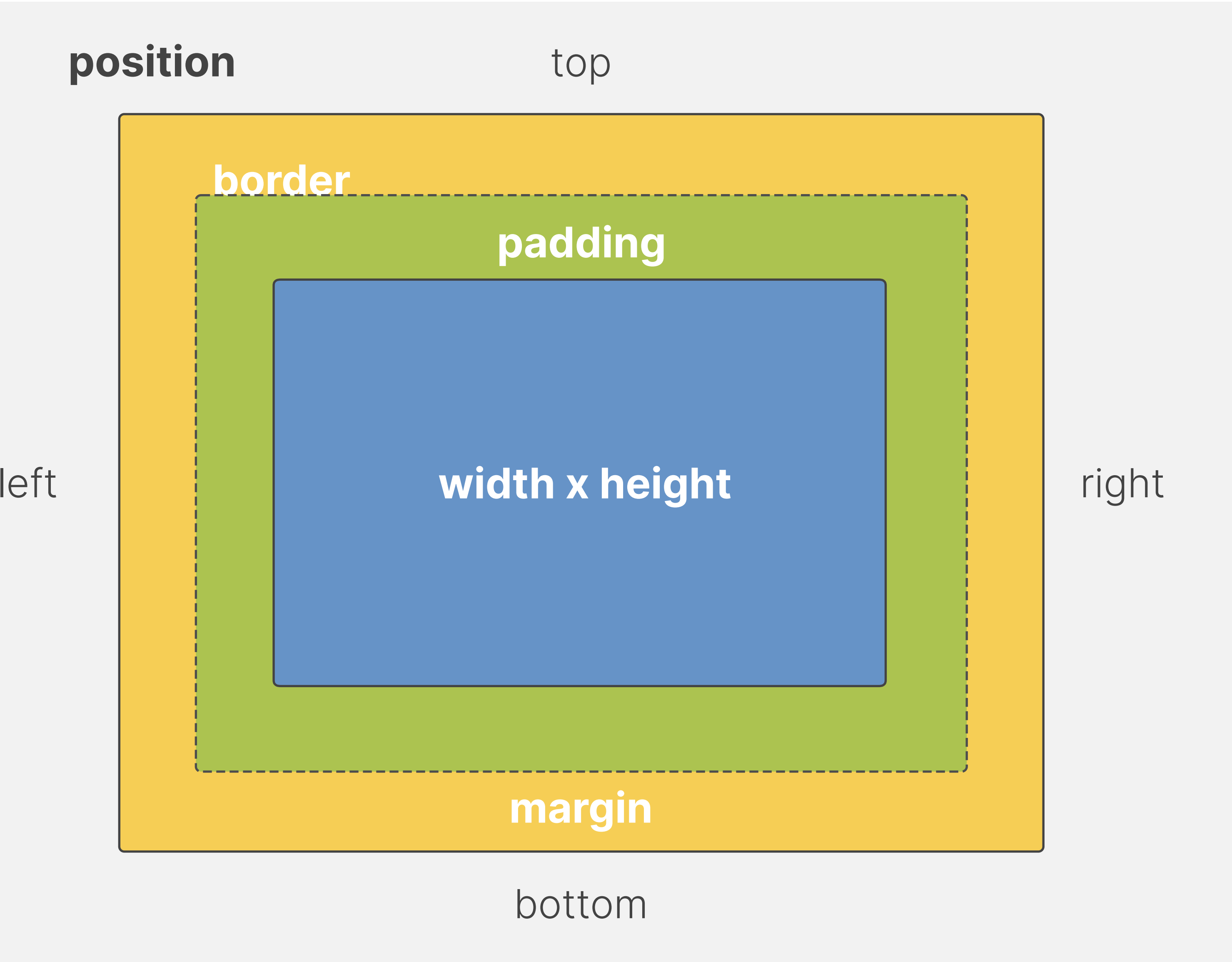


# The Box model



## Tailles et marges

```
article {
  width: 100%;
  height: auto;
  margin: 20px 10px 30px 10px;
  border: 2px solid pink;
  padding: 20px;
}

article {
  width: <size> | <auto>;
  height: <size> | <auto>;
  margin: <top> <right> <bottom> <left>;
  border: <width> <style> <color>;
  padding: <top> <right> <bottom> <left>;
}
```

## Shorthands (raccourcis)

```
.cadre {
  margin-top: 20px;
  margin-right: 20px;
  padding: 20px 10px;
  padding: 20px 10px 30px;
}

.cadre {
  margin-top: <size> | <auto>;
  margin-right: <size> | <auto>;
  padding: <top,bottom> <right,left>;
  padding: <top> <right,left> <bottom>;
}
```

## Unités

- px**: unité absolue en pixel
- %**: unité absolue en pourcentage, relatif au parent
- em**: 1em est égale à la taille de police, relatif à l'élément
- rem**: proportion de la taille de police, relatif au document (**html**)

# Sélecteurs

Ils permettent d'appliquer des styles aux éléments ciblés du html. Les sélecteurs se lisent de droite à gauche, le dernier élément du sélecteur est celui **ciblé**. Il doivent être le plus simple possible. Ils sont sensibles à la casse et ne commencent pas par un chiffre.

```
article .cadre #chapo {
  type   classe  identifiant

  background-color: yellow;
  propriété      valeur
}

<article>
<div class="cadre">
  <p id="introduction">...</p>
</div>
<article>
```

Plusieur genres de sélécteurs sont à votre disposition :

- type**: les éléments dont le nom correspond (balise html)
- .classe**: les éléments dont l'attribut **class** correspond
- #identifiant**: les éléments dont l'attribut **id** (unique) correspond
- parent > .enfant: tous les **.enfant** descendants direct d'un **parent**
- .ancêtre .enfant: tous les **.enfant** descendants d'un **.ancêtre**

## :Pseudo-classes

element:**hover**: lorsque le curseur **survole** l'élément

element:**focus**: lorsque l'élément est **sélectionné**

element:**first-child**: lorsque l'élément est le **premier** de son parent

element:**last-child**: lorsque l'élément est le **dernier** de son parent

# La cascade

Pour choisir quelle style doit prévaloir lors de conflits, il faut suivre les règles de la **cascade** et de la **spécificité**. À importance égale, la propriété la plus basse dans le fichier l'emporte.

## Spécifité

l'importance des règles est basée sur leur **spécificité**. Cette valeur est fonction de la complexité du sélécteur: 100 par id, 10 par classe, 1 par type.



## Héritage

Certaines propriétés sont **héritées**, si elles ne sont pas définies, le navigateur utilisera la valeur qu'il trouve chez le plus proche **ancêtre**. Par exemple : **font, text-align, color...**

# Mise en page

## Position

```
.illustration {
  position: relative;
}

.illustration .legende {
  position: absolute; /* sort l'élément du flux, */
  right: 10px; /* sa position est relative */
  bottom: 10px; /* à son plus proche ancêtre positionné */
}
```

## Éléments flexibles

```
.conteneur {
  display: flex; /* boîte flexible */
}

.conteneur .colonne { /* element flexible (enfant du conteneur) */
  flex-basis: 280px; /* taille initiale */
  flex-grow: 1; /* facteur d'agrandissement */
  flex-shrink: 2; /* facteur de rétrécissement */
}
```

## Grilles

```
.actualites {
  display: grid; /* une grille de 4 colonnes égales */
  grid-template-columns: 1fr 1fr 1fr 1fr; /* repeat(4, 1fr) */
  grid-gap: 20px; /* gouttière de 20px */
}

.actualites .article--mis-en-avant {
  /* l'article mis en avant occupera */
  grid-column: span 3; /* trois cellules (cases) horizontales */
  grid-row: span 2; /* et deux verticales */
}
```

## Responsive

```
@media screen and (max-width: 780px) { /* petits écrans */
  .actualites {
    display: initial; /* réinitialise la valeur (annule grid) */
  }
  .actualites .article--mis-en-avant {
    background-color: aliceblue;
  }
}
```

## Animations

```
.actualites {
  transition: 300ms transform ease-in-out, 200ms opacity;
}

.actualites:hover, .actualites:focus {
  opacity: .9;
  transform: translateY(-10px);
}
```



# Décorations

## Fonds

```
.header {
  background-color: pink;
  background-image: url(./fond.png);
  background-position: center top;
  background-repeat: no-repeat;
  background-size: cover;
}
```

## Textes

```
.chapo {
  font-family: "Open Sans", Arial, sans-serif;
  font-size: 1.2rem;
  font-style: italic;
  font-weight: bold;
  line-height: 1.5; /* 1,5 x taille de police courant */
  color: cornflowerblue;
  text-align: center;
  text-decoration: underline;
}
```

## Autres

```
.navigation {
  display: block;
  max-width: 90vw;
  border-radius: 10px;
  list-style: none;
  box-shadow: 2px 3px 5px rgba(0, 0, 0, .8);
  cursor: pointer;
  overflow: hidden;
  opacity: .8;
}
```

# Inline / Block

Toutes les balises visible du html sont représentées par un rectangle par le navigateur.  
La propriété **display** permet d’influer sur la manière dont vas être rendue cette boîte dans le flux.

- block** : Une hauteur automatique, occupe toute la largeur disponible et force le passage à la ligne dans le flux. Des marges laterales automatique permettent de gérer son alignement horizontal.
- inline** : Pas de taille en soit, comme les mots dans un paragraphe.
- inline-block** : A une taille, ne force pas le passage à la ligne, le plus petit possible.

# Bonnes pratiques

## Tailles et marges

Autant que possible, ne jamais attribuer de taille fixe (px).  
Prévilégier les propriétés max-width, min-height et les unités relatives.

Pour simplifier la gestion des marges, utiliser majoritairement des marges basses (margin-bottom).

## Sélecteurs

Les selecteurs doivent être simples (spécificité faible).  
Utiliser uniquement des sélecteurs de type et de classe permettra d’éviter au maximum les conflits. Ne pas utiliser de selecteurs d’identifiant.

## Nomenclature

Les noms de classes doivent représenter la fonction de l’élément, deux éléments identiques auront le même nom.  
Utiliser une convention de nommage tel que BEM.

```
<article class="article">
  <div class="article__content">
    <h3 class="article__title">...</h3>
    <p class="article__chapo">...</p>
  </div>
</article>
```

## Ordre de la source

Pour tirer profit de la cascade, les selecteurs sont ordonnés selon trois règles :

- Du générique (polices, couleurs) au spécifique (composants).
- Du global au local, plus une propriété influence d’éléments, plus elle sera hautee dans la source.
- La spécificité est croissante de haut en le bas de la cascade.

# Ressources

- MDN** : <https://developer.mozilla.org/fr/>
- DevDocs** : <https://devdocs.io/css/>
- Google Fonts** : <https://fonts.google.com/>
- CSS Tricks** : <https://css-tricks.com/>
- Smashing Magazine** : <https://www.smashingmagazine.com/>
- BEM** : <http://getbem.com/>
- ITCSS** : <https://skl.sh/harry>



# Cascading Style Sheets