boîtes et flux

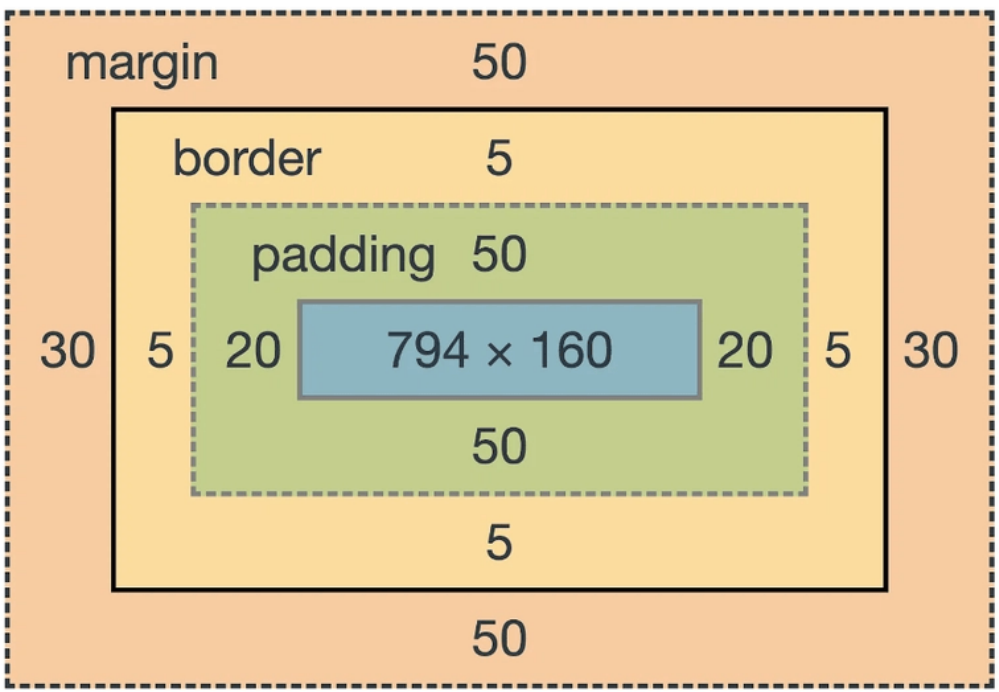
la base de CSS

formation CSS

tous les slides et exercices

https://css.formations.alsacreations.fr/

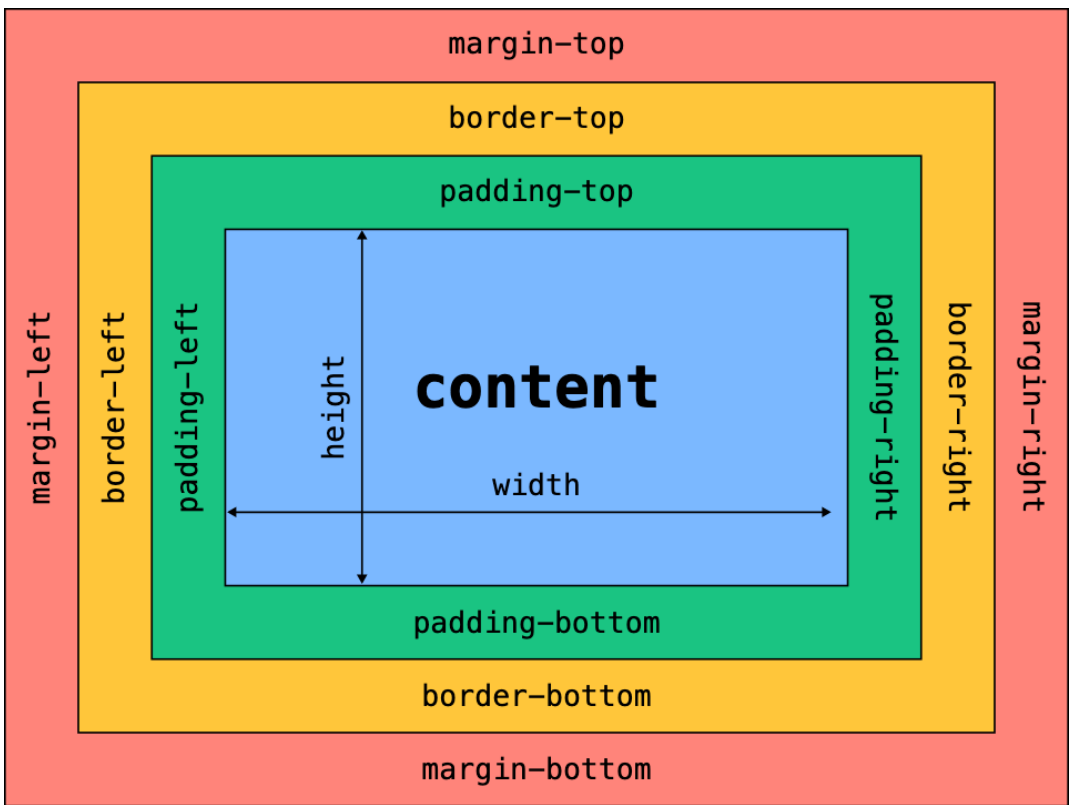
Le Modèle de Boîte



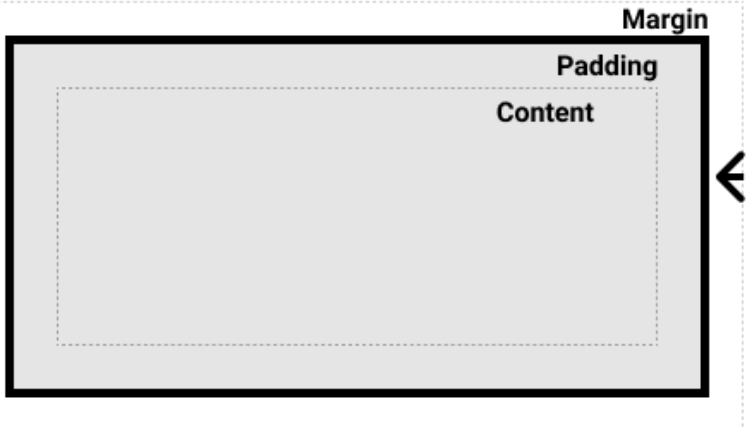
Le Modèle de Boîte

Le navigateur dessine les éléments HTML sous forme de "boîtes".

Quatre zones composent chaque boîte : content, padding, border et margin.



Modèle de boîte standard




```
1 .box {
2   box-sizing: content-box;
3 }
```

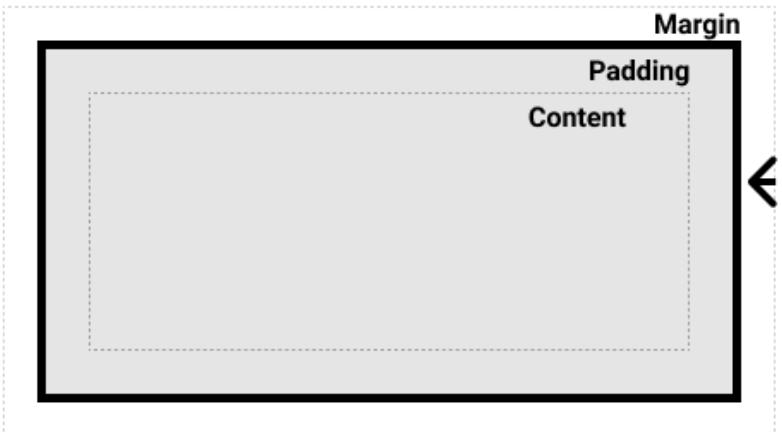
valeur par défaut

Les valeurs de width et height s'appliquent à la zone de contenu.

Donc la "vraie" largeur de la boite :

width + padding + border

Modèle de boîte moderne




```
1  .box {
2    box-sizing: border-box;
3  }
```

Les valeurs de width et height s'appliquent à la zone de bordure.

Donc la largeur totale de la boite :

width

Modèle de boîte moderne

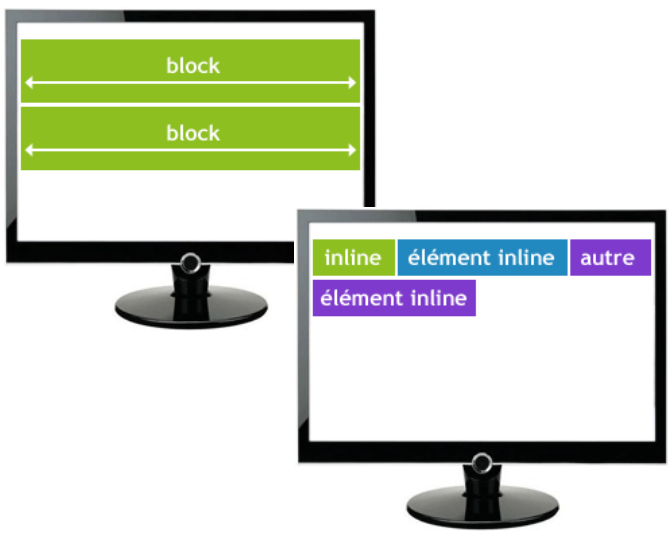
à mettre dans tout fichier Reset CSS

```
*,
*::before,
*::after {
    box-sizing: border-box;
}
```

Exercice d'observation

pour se réveiller en douceur !

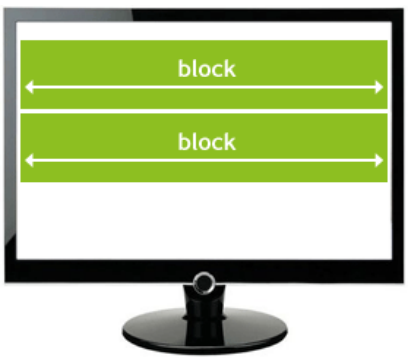
Types de boîtes



Il existe deux types de boîtes :

- Les boîtes de type **bloc**
- Les boîtes de type **en ligne**
- Le type d'une boîte est défini par la valeur de sa propriété `display`
- Tous les éléments HTML sont associés à une valeur de `display`

Boîtes en "bloc"



Flux :

- Passent à la ligne (s'affichent sous la boîte précédente)

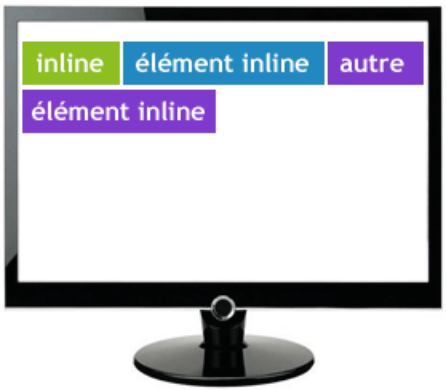
Taille par défaut :

- Occupent toute la largeur disponible dans leur parent
- Occupent la hauteur de leur contenu

Exemples (HTML) :
, <p>, <h1>... <h6>, <ul>, <li>, <form>, <header>, <footer>, etc.

Exemples (CSS) :
display: block | flex | grid | table | list-item, etc.

Boîtes "en ligne"



Flux :

- S'affichent à côté de la boîte précédente (si possible)

Taille par défaut :

- Occupent la largeur de leur contenu
- Occupent la hauteur de leur contenu

Exemples (HTML) :
, <a>, , , <img>, <input>, <button>, etc.

Exemples (CSS) :
display: inline | inline-block | inline-flex | inline-grid | inline-table, etc.

⚠️ Particularité : on ne peut pas appliquer de dimensions (width, height) à un élément en `display: inline`

QUIZ !

"une question de taille"

Largeur 1

```css
a {
  width: 100px;
  height: 100%;
  padding: 5px;
}
```

Largeur 1

```css
a {
  width: 100px;
  height: 100%;
  padding: 5px;
}
```

la taille de son contenu

l'élément <a> est un élément "inline" donc ni width ni height ne s'appliquent 🤷‍♂️

Largeur 2

```
1 p {
2    width: 100px;
3    padding: 5px;
4 }
```

Largeur 2

```css
p {
  width: 100px;
  padding: 5px;
}
```

largeur = 110px

width + padding à droite + padding à gauche

Largeur 3

```css
p {
  width: 100px;
  padding: 5px;
  border: 1px;
  margin: 10px;
}
```

Largeur 3

```css
p {
  width: 100px;
  padding: 5px;
  border: 1px;
  margin: 10px;
}
```

largeur = 112px

width + padding à droite + padding à gauche + border à droite + border à gauche (les margin sont à l'extérieur de la boîte)

Largeur 4

```css
p {
  width: 100px;
  padding: 10%;
  border: 1em;
}
```

Largeur 4

```css
p {
  width: 100px;
  padding: 10%;
  border: 1em;
}
```

largeur = bah euh... 🤷

difficile de calculer avec des unités différentes

Largeur 5

```css
p {
  width: 100px;
  padding: 10%;
  border: 1em;
  box-sizing: border-box;
}
```

Largeur 5

```css
p {
  width: 100px;
  padding: 10%;
  border: 1em;
  box-sizing: border-box;
}
```

largeur = 100px

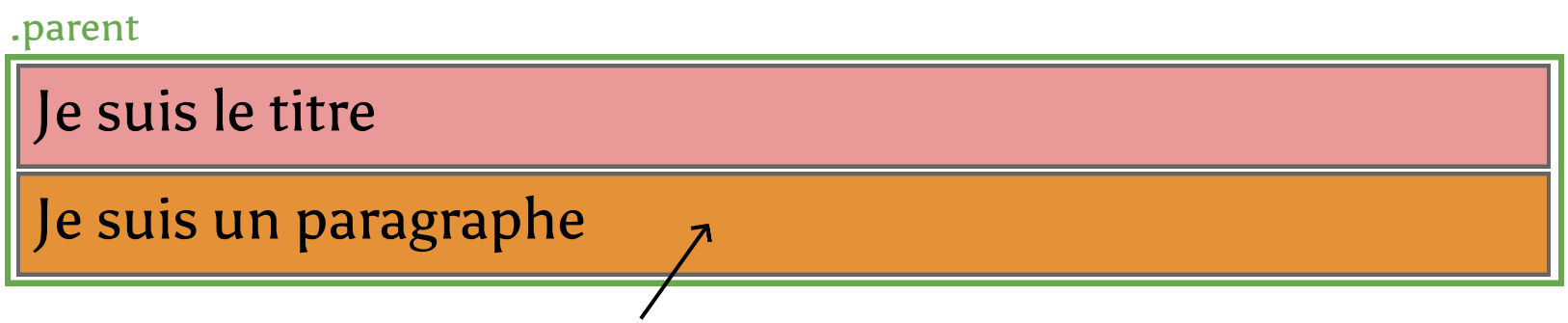
avec border-box, la largeur de la boîte correspond simplement à la valeur de width

Flux
et positionnement

Le Flux

```html
<div class="parent">
  <h1>Je suis le titre</h1>
  <p>Je suis un paragraphe</p>
</div>
```

```css
h1 {background: pink;}
p {background: chocolate;}
```



Je me place le plus haut possible à gauche dans mon parent, **à la suite de l'élément précédent** *(et en plus je prends toute la largeur car je suis de type Bloc)*

Le Flux

```html
<div class="parent">
  <h1>Je suis le titre</h1>
  <p>Je suis un <span>paragraphe</span></p>
</div>
```

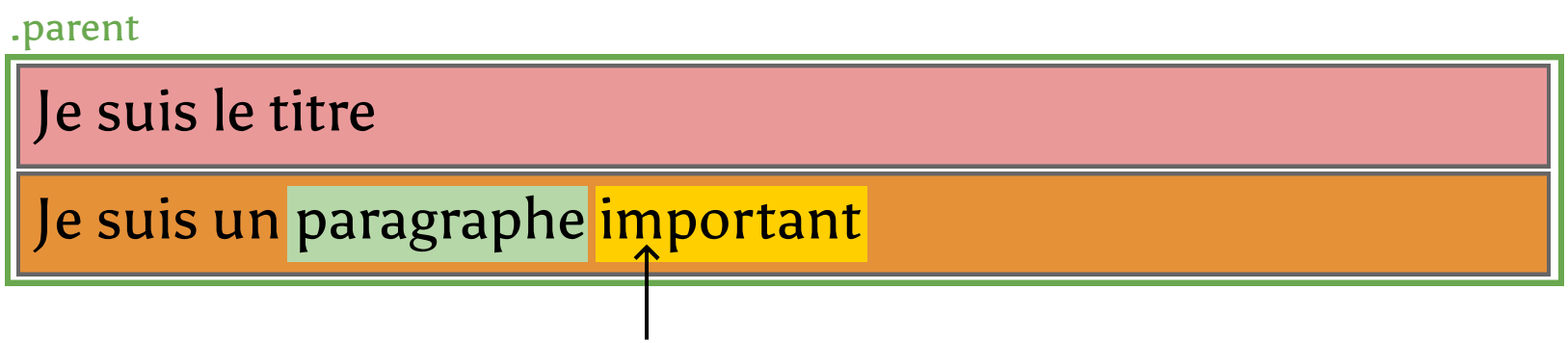
```css
h1 {background: pink;}
p {background: chocolate;}
span {background: lightgreen;}
```

Je suis de type Inline donc ma taille est celle de mon contenu et je m'affiche à côté des éléments précédents

Le Flux

```html
<div class="parent">
  <h1>Je suis le titre</h1>
  <p>Je suis un <span>paragraphe</span>
    <strong>important</strong>
  </p>
</div>
```

```css
h1 {background: pink;}
p {background: chocolate;}
span {background: lightgreen;}
strong {background: yellow;}
```



Je suis de type Inline donc ma taille est celle de mon contenu et je m'affiche à côté des éléments précédents

Le Flux

“ Le Flux d'un document décrit le comportement naturel d'affichage des éléments
(selon l'ordre d'apparition dans le HTML)

Spécificités du Flux

- C'est le placement **par défaut**
- L'affichage d'un élément dépend de sa valeur de `display`
- Les éléments de type **Bloc** s'affichent naturellement les uns sous les autres
- Les éléments de type **En ligne** s'affichent côte à côte
- Les éléments **interagissent** entre eux (se poussent, s'agrandissent, etc.) car ils sont sur le même plan

Sortir du Flux

```
<div class="parent">
  <div class="salade">salade</div>
  <div class="tomate">tomate</div>
  <div class="oignons">oignons frits et…</div>
  <div class="choucroute">choucroute</div>
</div>
```

Sortir du Flux

.parent

Pour le moment, tous les éléments sont placés naturellement dans le Flux les uns sous les autres puisque ce sont des (donc de type Bloc)

Sortir du Flux

exemple : positionnement absolu

.parent


```
.tomate {
  position: absolute;
}
```

L'élément .tomate passe en position absolute

Sortir du Flux

exemple : positionnement absolu

tomate

.parent

salade

oignons frits et
petits lardons

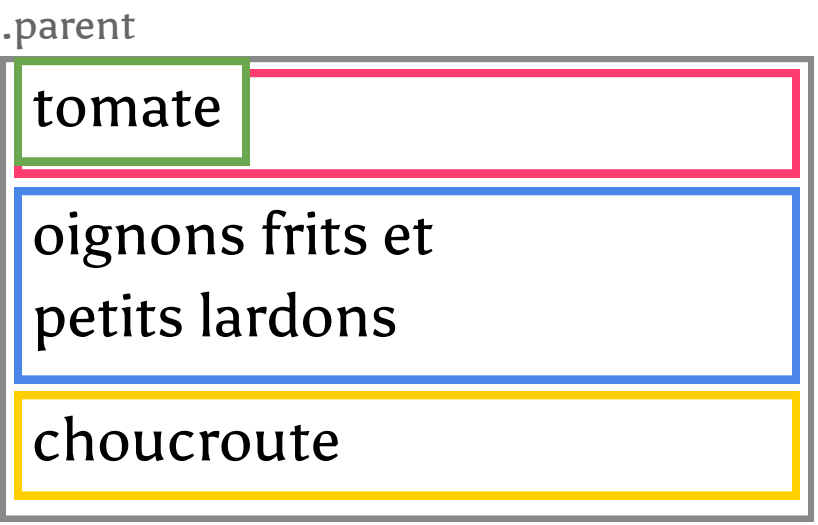
choucroute

L'élément .tomate passe en position absolute,
placé en coordonnée top: 0

```css
.tomate {
  position: absolute;
  top: 0;
}
```

Sortir du Flux

exemple : positionnement absolu




```
.parent {
  position: relative;
/* je suis référent de tomate */
}
.tomate {
  position: absolute;
  top: 0;
}
```

L'élément .tomate passe en position absolute, placé en coordonnée top: 0... de son référent .parent

Spécificités

du positionnement Absolu

- Mode de positionnement **hors du Flux**
- Il prend la taille de son contenu (par défaut)
- Le déclencheur est `position: absolute`
- L'élément se place sur un calque au-dessus du plan initial (Flux)
- Les coordonnées dépendent des propriétés `top` , `right` , `bottom` et `left`
- Le référent est un élément **ancêtre** placé en `position: relative` (pour simplifier)

Fil rouge

Project Wallace

propriétés : display et position

https://www.projectwallace.com/analyze-css