

CSS

Boites et mise en page

Remarque : style de `` et `<div>`

- Les éléments de structuration `` et `<div>` ne sont normalement pas visibles mais permettent de regrouper un ensemble d'éléments pour leur donner des attributs de style commun.

`...` élément *inline* : ne génère pas de saut à la ligne ni avant ni après

Ne peut pas comprendre de blocs (ex: `<p>`)

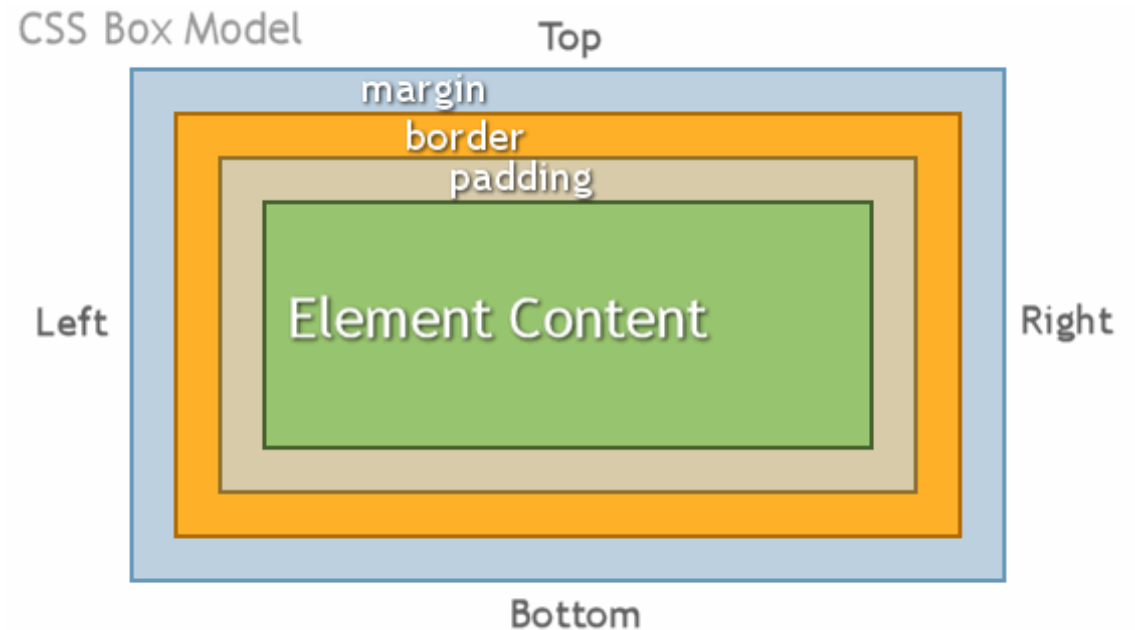
`<div>...</div>` élément *bloc* : un saut de ligne avant et après chaque section div.

Peut comprendre des blocs (ex: `<p>`)

- Tous les éléments contenus héritent certaines propriétés de leur élément contenant (couleur de fond, texte, etc)
- mais ils peuvent éventuellement les redéfinir
- **Remarque.** les tailles relatives de texte (%, smaller, larger) sont relative à la taille héritée par l'élément contenant

Le modèle de boîte

Tous les éléments d'une page HTML sont entourés d'une boîte virtuelle qui définit leurs dimensions



Content : la zone où tous les elements contenus sont disposés

Border : la bordure de la boîte

Padding : le remplissage (à l'intérieur de la bordure)

distance entre la bordure de la boite et son contenu

Margin : la marge (à l'extérieur de la boite)

distance entre la bordure de la boite et tout autre element extérieur (y compris son élément contenant)

Le modèle de boîte

Tous les éléments d'une page HTML sont entourés d'une boîte virtuelle qui définit leurs dimensions

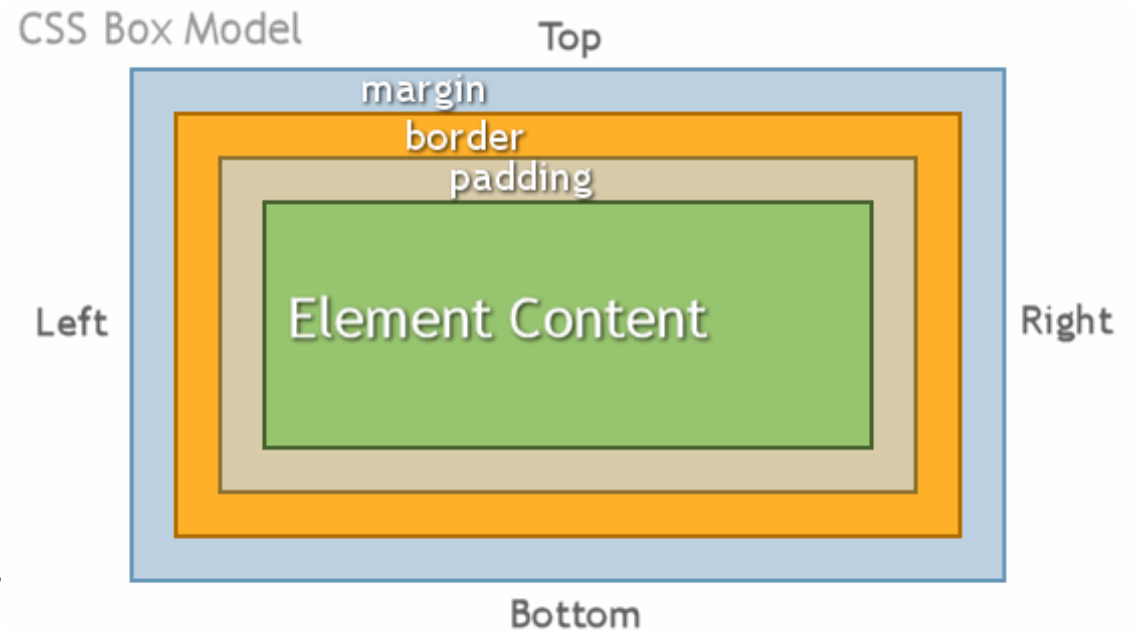
CSS permet de modifier le dimensions de ces différentes parties avec les propriétés suivantes

width, height

La largeur et la hauteur du contenu (*content*) de la boite

padding, margin, border

la taille du remplissage, marge et bordure de la boite



Quelques éléments de style pour les boîtes

`border: épaisseur style couleur`

épaisseur: taille ou thin, medium, thick

style: dotted, dashed, solid, double, ...

`background-color: couleur`

définit la couleur de fond de *content* + *padding*

(le *border* est par défaut noir)

Le modèle de boîte : exemple

```
<head>
<style>
div {
    background-color: #A8F569;
    width: 200px;
    height: 150px;
    padding: 25px;
    border: 15px solid;
    margin : 50px;
}
</style>
</head>
<body>
<div>Lorem ipsum dolor sit amet, ...</div>
</body>
```

Le modèle de boîte : remarques

- Si `width` (`height`) n'est pas spécifié, la largeur (hauteur) du contenu est automatique : déterminée par les autres contraintes (e.g. marges, autres elements, tailles des elements contenus, type de *display*...)
- `width` et `height` spécifient uniquement la taille du contenu (*content*)
- Pour avoir la taille effective d'une boîte (jusqu'au bord), il faut ajouter le `padding` et le `border`

Le modèle de boîte : dimensions

height: auto | taille | %

width: auto | taille | %

auto : le navigateur détermine la taille (en fonction des autres contraintes)

taille : valeur suivie de l'unité

% : pourcentage

padding: taille | %

margin: auto | taille | %

Peuvent aussi être suivi de `-left`, `-top`, `-right`, `-bottom` pour modifier la dimension dans une direction seulement

min-width : taille | %

max-width : taille | %

Fixent une valeur minimale / maximale pour la `width` et la `height` dans les cas où celles-là peuvent varier (selon le contenu, la taille de la fenêtre, ...)

Le modèle de boîte : dimensions

Unités de taille : pas d'espace entre valeur et unité

mm, cm, in

pt : 1/72 inch (point), pc : 12 points (pica)

px : pixel (1 point sur l'écran)

em : largeur du caractère 'm' de la police courante

ex : hauteur du caractère 'x' de la police courante

Pourcentage :

- width : % par rapport à width de l'élément contenant
- height : % par rapport à la height de l'élément contenant (si non automatique)
- padding, margin : % par rapport à la width de l'élément contenant

Le modèle de boîte : *box-sizing*

Par défaut `width` et `height` font reference à la taille du contenu de la boîte (*element content*)

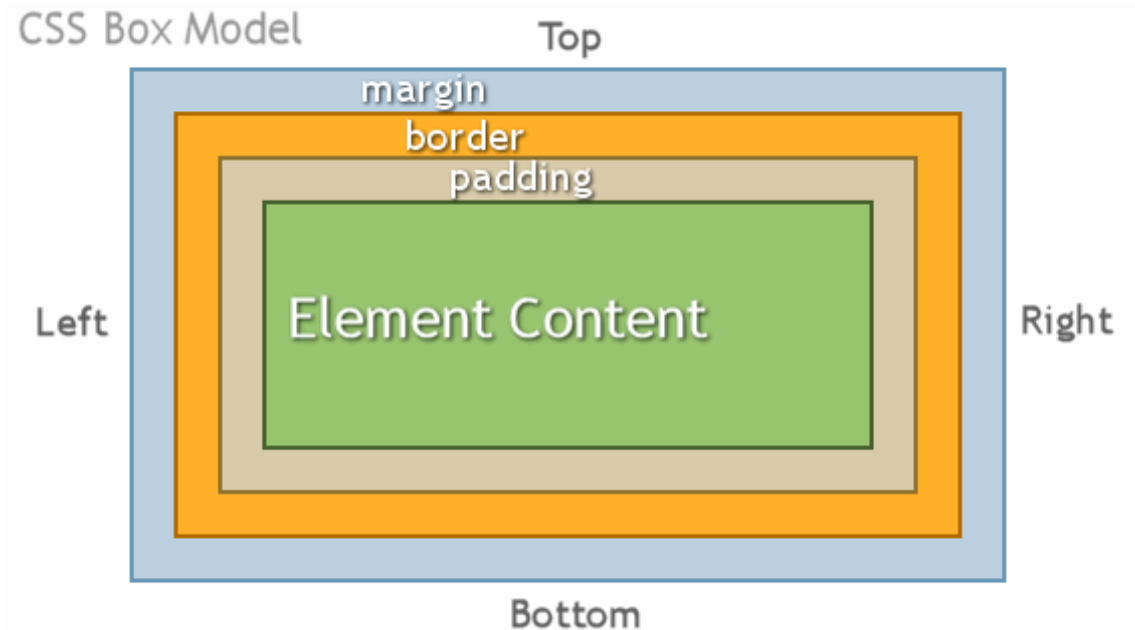
Mais cela peut être modifié (nouveau en CSS3) :

`box-sizing: border-box;`

width et height font reference à la largeur et hauteur de `content + padding + border`

`box-sizing: content-box;`

comportement par default



box-sizing : exemple

```
<head> <style>
#outerdiv{
  border : 3px solid red;
  width : 300px;
  height : 300px;
  margin : 50px;
}
#innerdiv {
  background-color: #A8F569;
  width: 100%;
  padding: 25px;
  border: 15px solid pink;
  /* box-sizing : border-box; */
}
</style> </head>

<body>
<div id="outerdiv">
  <div id="innerdiv">.. </div>
</div>
</body>
```

Mise en page CSS

Des propriétés CSS peuvent être utilisées pour contrôler la disposition des boîtes dans la page

- type de rendu de boîte (propriété `display`)
- positionnement (propriétés `position` et `float`)

Type de rendu des boites (*display*)

- Chaque élément a un type de rendu par défaut (valeur de la propriété `display`)
- Les deux types de rendu principaux sont :

bloc : `display: block`

en-ligne : `display: inline`

- Exemples d'elements de type bloc : `<div>` `<p>` `<h1>` `<section>` `<footer>` etc
- Exemples d'elements de type en-ligne : `` `` `<a>`

Bloc et en-ligne

Les elements de type **bloc** génèrent un saut de ligne avant et après, et leur boite (y compris la marge) occupe toute la largeur de la page

- la taille du contenu peut être contrôlé avec `width` et `height`
- la `width automatique` est maximale dans la largeur de la page
- la `height automatique` est minimale (i.e. déterminée par la hauteur des elements contenus)

Les elements de type **en-ligne** ne génèrent pas de saut de ligne, ni avant ni après, et leur boite (y compris la marge) occupe juste la place nécessaire :

- la `width` et la `height automatiques` sont minimales (déterminées par la taille des elements contenus)
- la taille du contenu **ne peut pas** être contrôlée avec `width` et `height`
- les marges verticales sont nulles et ne peuvent pas être contrôlées

D'autres types de rendu

- La valeur par défaut de `display` peut être modifiée pour changer le type de rendu d'une boîte

```
img {  
    display : block;  
}
```

- D'autres valeurs que `inline` et `block` sont disponibles pour la propriété `display` :
 - `inline-block`
 - `flex` (nouveau en CSS3)
 - `table` (affiche comme un element `<table>`)
 - `table-cell` (affiche comme un élément `<tr>`)
 - `none` (desactive l'affichage)

et bien d'autres... <https://developer.mozilla.org/fr/docs/Web/CSS/display>

inline-block

Avec `display: inline-block`

la boîte se comporte comme un élément en ligne

- ne génère pas de saut de ligne, ni avant ni après,
- la `width` et la `height` automatiques sont minimales (déterminées par la taille des éléments contenus)

mais peut être manipulée comme un bloc :

- la taille du contenu et toutes les marges peuvent être contrôlés avec `width`, `height` et `margin`
- **remarque** : petite marge horizontale automatique entre deux inline-block adjacents (jamais collés!)

Positionnement CSS

- Par défaut les boites sont disposées une après l'autre (avec des sauts de ligne ou pas, selon le display)
- et cela dans l'ordre des éléments dans le document.
- On appelle cela le **flux des éléments**
- Ce comportement correspond à la valeur par défaut (valeur `static`) de la propriété `position` de chaque element
- mais la valeur de cette propriété peut être modifiée , ce qui altère le flux des éléments

`position: static | relative | absolute | fixed`

Positionnement relatif

```
sélecteur {  
    ...  
    position: relative  
    top: taille  
    bottom: taille  
    right: taille  
    left: taille  
}
```

- la boîte est déplacée par rapport à sa position par défaut
- le déplacement relatif par rapport à son bord haut / bas / droit / gauche est celui spécifié par *top* / *bottom* / *right* / *left*
- les autres éléments se positionnent comme si la boîte était dans sa position par défaut (le flux n'est pas altéré)

Positionnement relatif : exemple

```
div{  
    background-color: pink;  
    width: 70%;  
    height: 100px;  
}  
#div2{  
    background-color: orange;  
    position: relative;  
    top: 20px;  
    left: 100px;  
}
```

```
...  
<body>  
  <div> div 1, static </div>  
  <div id ="div2"> div 2, relative  
  </div>  
  <div> div 3, static </div>  
</body>  
...
```

03-relative.html

Positionnement absolu

```
sélecteur {  
    ...  
    position: absolute  
    top: taille  
    bottom: taille  
    right: taille  
    left: taille  
}
```

- la boîte est retirée du flux
- les autres éléments se positionnent comme si la boîte n'était pas présente (flux altéré)
- la boîte se positionne à distance `top` / `bottom` / `right` / `left` par rapport au bord haut / bas / droit / gauche de son premier ancêtre ayant position non-static (ou de la page entière, si cet ancêtre n'existe pas)

Positionnement absolu : exemple

```
.parent {  
    margin : 150px;  
    width:70%;  
    border: 3px solid green;  
    /*position: relative */  
}  
#div1, #div3 {  
    background-color: pink;  
    border: 1px solid orange;  
    width: 200px;  
    height: 100px;  
}  
#div2 {  
    background-color: orange;  
    border: 1px solid orange;  
    width:200px;  
    height: 50px;  
    position: absolute;  
    top:30px;  
    left:0px;  
}
```

```
...  
<body>  
  
<div class="parent">  
  
    <div id ="div1"> div 1, static  
    </div>  
    <div id ="div2" > div 2 absolute  
    </div>  
    <div id ="div3"> div 3, static  
    </div>  
  
</div>  
  
</body>  
...
```

03-absolute.html

Positionnement fixe

```
sélecteur {  
    ...  
    position: fixed  
    top: taille  
    bottom: taille  
    right: taille  
    left: taille  
}
```

- identique au positionnement absolu à différence du système de coordonnées :
 - la boîte se positionne à distance `top` / `bottom` / `right` / `left` par rapport au bord haut / bas / droit / gauche de la fenêtre
- conséquence : la boîte est affichée toujours à la même distance des bords de la fenêtre, même quand on fait défiler la page
- exemple identique au cas de position absolue, mais on remplace `absolute` par `fixed`

Positionnement flottant

`float: left | right`

- la boîte est retirée du flux
- les autres éléments se positionnent comme si la boîte n'était pas présente
- en verticale la boîte garde sa position naturelle (celle qu'elle aurait en position statique)
- en horizontale la boîte se positionne le plus à droite (`float: right`) ou le plus à gauche (`float: left`) possible
 - jusqu'à rencontrer le bord de son contenant ou un autre élément flottant

Positionnement flottant : exemple

```
div {  
    margin-top: 20px;  
}  
  
#div1, #div3, #div4 {  
    background-color: pink;  
    height: 150px;  
}  
  
#div2 {  
    background-color: orange;  
    float: right;  
    height: 250px;  
    width: 150px;  
    /*margin-left: 20px*/  
}  
  
...  
  
<body>  
  
<div id = "div1"> div 1, static ...  
</div>  
  
<div id = "div2"> div 2, flottante  
</div>  
  
<div id = "div3"> div 3, static ...  
</div>  
  
<div id = "div4"> div 4, static ...  
</div>  
  
</body>  
...
```


Positionnement flottant

La propriété `clear: left | right | both`

d'un élément impose que l'élément n'ait pas de blocs flottants sur le coté spécifié (gauche / droit / les deux)

```
#div4{  
    clear:right;  
}
```

03-float.html

CSS

compléments et exercices

Éléments de style pour les tables

`table-layout: auto | fixed` la taille des cellules s'adapte au contenu ou est fixée

`border` la bordure; à la fois pour la table et les cellules

`border-spacing: taille` Espace entre cellules d'une table

`border-collapse: collapse` pour fusionner les bords de cellules adjacentes

`background-color, width, height` à la fois pour la table et les cellules

pour mettre en forme le contenu des cellules :

`text-align: left | center | right | justify`

`vertical-align: top | middle | bottom`

`padding : taille`

Exemple de table

```
table, th, td { border : 1px solid grey; }
table { border-collapse: collapse; width: 100%; }

th, td {
    height: 50px;
    text-align: center;
    vertical-align: middle;
    color: green;
}
th {
    background-color: #57B8A4;
    color: white;
}
tr:nth-child(even){
    background-color: #f2f2f2
}
tr:hover {
    background-color: rgba(241, 76, 156, 0.3);
}
```

Exercice I

Créer une page contenant un menu déroulant, composé d'un bloc fixe et un bloc qui déroule quand la souris survole le bloc fixe. Le bloc qui déroule contient une liste de liens.

Ensuite ajouter plusieurs de ces éléments pour former un barre de navigation en haut de la page