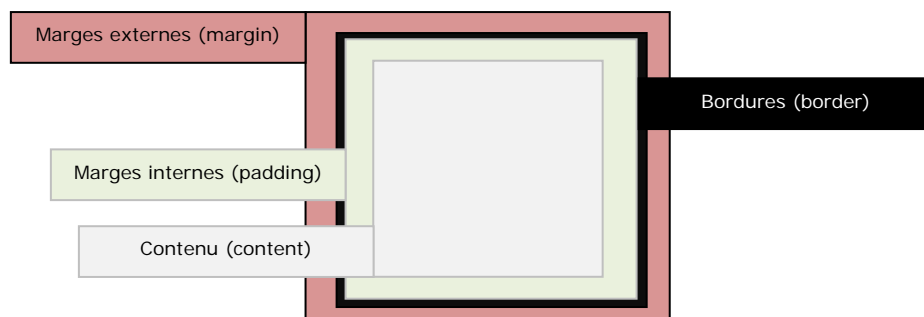


## 2. Les blocs en CSS

Tous les éléments structurant une page HTML peuvent être considérés comme des blocs rectangulaires occupant toute la largeur de la page. Il s'agit en CSS de positionner les blocs comme des boîtes juxtaposées, imbriquées les unes dans les autres, de gérer leurs dimensions, leurs bordures et marges.

### A. Constitution du bloc

Un bloc est constitué de plusieurs épaisseurs : les marges internes, les bordures et, à l'intérieur, le contenu. On peut ajouter des marges externes pour séparer les blocs les uns des autres.



### B. Gérer l'aspect et les dimensions des blocs

Les dimensions, largeur et hauteur d'un bloc → les propriétés **width** et **height**

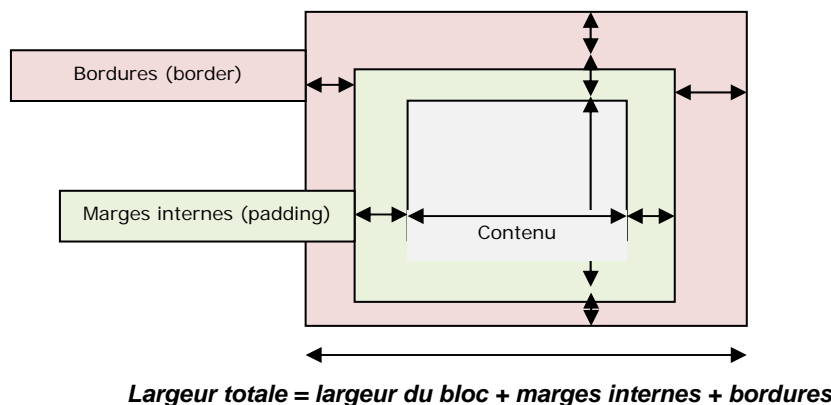
Par défaut un élément de type bloc occupe 100% de la largeur et sa hauteur correspond à celle de son contenu.

Les propriétés « **width** » et « **height** » permettent de définir la largeur et la hauteur d'un bloc. Les tailles sont exprimées en px, em ou %.

Syntaxe css pour attribuer une largeur et une hauteur au bloc header

```
header {  
    width :790px ;  
    height :60px ;  
}
```

## Calcul des dimensions des blocs en fonction des marges et bordures



**Une propriété css : `box-sizing` permet de s'affranchir de ces calculs, lorsqu'on déclare une largeur de bloc de 600px, le bloc fera 600px même si on ajoute marges internes et bordures.**

```
* {  
  -moz-box-sizing: border-box;  
  -webkit-box-sizing: border-box;  
  box-sizing: border-box;  
}
```

## Gestion des débordements → la propriété `overflow`

Généralement le contenu d'une boîte est enfermé à l'intérieur de celle-ci. Il arrive, lorsque l'on précise la hauteur du bloc, que le contenu déborde.

L'utilisation de la propriété `overflow` va permettre de gérer ces débordements → ses valeurs :

**visible** : le débordement est visible (valeur par défaut)

**hidden** : le débordement est masqué (pas accessible)

**scroll** : il y a affichage automatique d'une barre de défilement, le texte débordant reste accessible

**auto** : une barre de défilement apparaît seulement s'il y a débordement.

Syntaxe css pour indiquer la hauteur des blocs section et l'apparition d'une barre de défilement si le contenu est trop important.

```
section {  
  height: 100px;  
  overflow: auto;  
}
```

## Les bordures → les propriétés `border`

Les propriétés `border` permettent de définir les divers aspects des bordures d'encadrement : le type, la dimension, la couleur. Elles s'appliquent aux éléments de type « bloc » ainsi qu'à certains éléments « en-ligne » comme les images.

Les propriétés : **border**, **border-top**, **border-right**, **border-bottom** et **border-left** (bordure), **border-width** (largeur), **border-color** (couleur), **border-style** (trait du contour)

- La propriété **border-width** permet de définir l'épaisseur des bordures → valeurs : **px** ou **thin** (fin), **medium** (moyen), **thick** (épais)
  - La propriété **border-color** permet de définir la couleur des bordures → valeur : **couleur**.
  - La propriété **border-style** permet de définir le style des bordures → valeurs : **none**, **hidden** (cachée, elle n'apparaîtra jamais), **dotted** (pointillés), **dashed** (tirets), **solid** (trait simple), **double** (trait double), **groove** (en relief), **ridge**, **inset** et **outset** (effets 3D)
1. Lorsque les 4 bordures sont identiques, il est possible de réaliser un encadrement en utilisant la **propriété raccourcie** :

```
selecteur {  
    border : 'valeur de border-width' 'valeur de border-style' 'valeur de border-color' ;  
}
```

Les valeurs des éléments sont énoncées les unes à la suite des autres, peu importe l'ordre, et sont séparées par des espaces.

Syntaxe CSS pour tous les paragraphes, ils seront entourés d'une bordure

```
p {  
    border: 1px solid #c00;  
} /* l'ordre n'importe pas, on obtient un paragraphe entouré d'une bordure rouge, au  
   trait plein et d'une largeur de 1px */
```

Résultat

2. Il est possible également d'énoncer successivement [pour les 4 côtés] les valeurs des propriétés → **border-style** / **border-color** / **border-width**.  
Pour chaque propriété, les valeurs sont mentionnées les unes à la suite des autres en respectant un ordre : haut, droit, bas, gauche, séparées par des espaces (\*)

\* Règles : les valeurs sont indiquées à partir du haut, dans le sens des aiguilles d'une montre : **top** → **right** → **bottom** → **left**. Lorsque la valeur d'un ou plusieurs côtés est absente, la valeur attribuée correspond à celle du côté qui lui est opposé.

- a. S'il n'y a qu'une valeur spécifiée, elle s'applique aux bords des 4 côtés ;
- b. S'il y a deux valeurs, la première s'applique aux bords du haut et du bas, la seconde aux bords de droite et de gauche ;
- c. Quatre valeurs : haut, droite, bas et gauche (toujours dans cet ordre). Ceci vaut pour les propriétés **margin** et **padding**.

selecteur {

**border-style**: 'valeur de border- style du haut' 'valeur de border- style à droite' 'valeur de border- style du bas' 'valeur de border- style à gauche';

**border-color**: 'valeur de border-color du haut' 'valeur de border- color à droite' 'valeur de border-color en bas' 'valeur de border-color à gauche';

**border-width**: 'valeur de border-width en haut' 'valeur de border-width à droite' 'valeur de border-width en bas' 'valeur de border-width à gauche';

}

### Syntaxe CSS

p {

border-style : solid dashed solid dotted;

/\* l'ordre importe → style de la bordure : 'solid' pour le haut, 'dashed' à droite, 'solid' en bas et 'dotted' à gauche gauche \*/

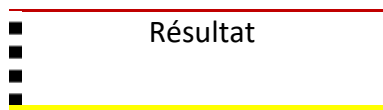
border-color : red orange yellow black;

/\* l'ordre importe → couleur de la bordure 'red' en haut, 'orange' à droite, 'yellow' en bas, 'black' à gauche \*/

border-width : 1px 2px 3px 4px;

/\* l'ordre importe → largeur de la bordure 1px haut, 2px à droite, 3px en bas, 4px à gauche \*/

}



3. Il est possible enfin de définir les bordures séparément avec les propriétés : border-top, border-right, **border-bottom** et **border-left** de la façon suivante :

selecteur {

**border-top** : 'valeur de border-width' 'valeur de border-color' 'valeur de border-style';

**border-right** : 'valeur de border-width' 'valeur de border-color' 'valeur de border-style';

**border-bottom** : 'valeur de border-width' 'valeur de border-color' 'valeur de border-style';

**border-left** : 'valeur de border-width' 'valeur de border-color' 'valeur de border-style';

}

### Syntaxe CSS

p {

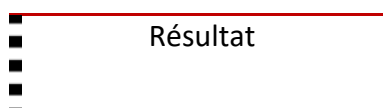
border-top: 1px solid red;

border-right: 2px dashed orange;

border-bottom: 3px solid yellow;

border-left: 4px dotted black;

}



**IMPORTANT : Pour qu'une bordure soit affichée, il faut nécessairement avoir défini le style.**

4. Comme il est possible de supprimer le soulignement par défaut des liens avec la propriété CSS : `text-decoration:none`; il est également possible de supprimer la bordure (bleue) par défaut de toutes les images sur lesquelles un lien a été codé en donnant la valeur **none** ou **0** à la propriété **border**

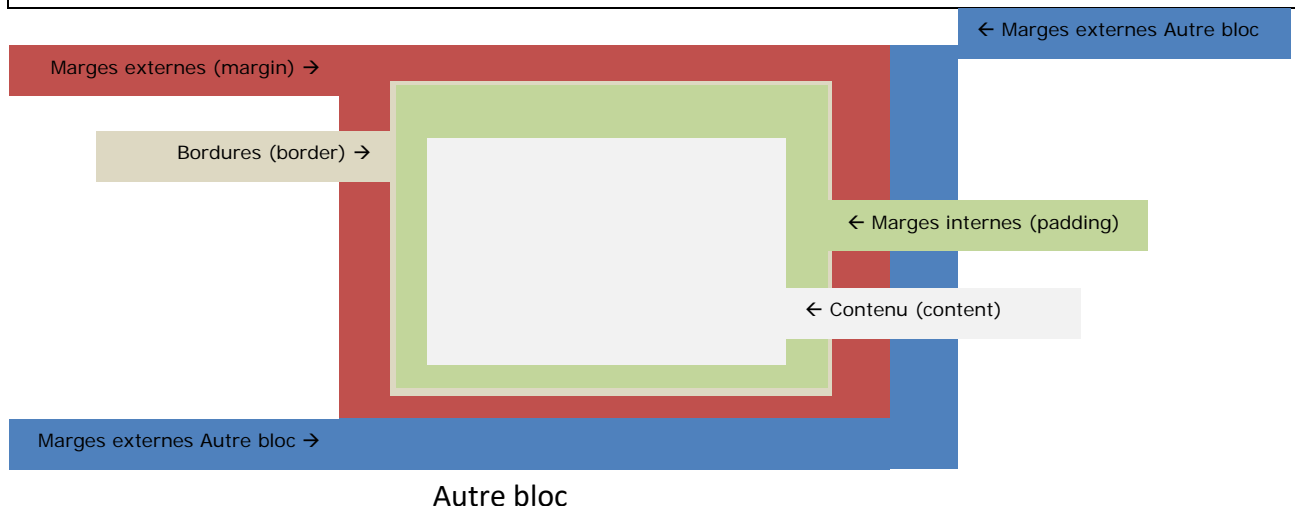
Dans le fichier.html :

```
<p><a href="pages/page1.html"></a></p>
```

#### Syntaxe CSS

```
a img {  
    border: none;  
}
```

Les marges extérieures et intérieures → les propriétés **margin** et **padding**



La propriété **margin** définit les dimensions des marges externes (en px, em, %). L'arrière-plan des marges est toujours transparent. Ces marges vont servir à espacer les blocs.

La propriété **padding** définit la valeur de la marge interne, la distance entre le bord du contenant et le contenu.

Les propriétés : **margin** et **padding** (propriétés raccourcies des marges externes et internes), **margin-top** (marge externe du haut), **margin-right** (marge externe de droite), **margin-bottom** (marge externe du bas) et **margin-left** (marge externe de gauche), **padding-top** (marge interne du haut), **padding-right** (marge interne de droite), **padding-bottom** (marge interne du bas) et **padding-left** (marge interne de gauche).

Les propriétés raccourcies « **margin** » et « **padding** » :

```
selecteur { margin : 'valeur de margin-top' 'valeur de margin-right' 'valeur de margin-bottom' 'valeur de margin-left' ; }
```

`selecteur { padding : 'valeur de padding-top' 'valeur de padding -right' 'valeur de padding -bottom' 'valeur de padding -left' ; }`

### Syntaxe CSS

```
p {  
  margin:5px ;  
} /* s'applique aux 4 marges */  
  
p {  
  margin:10px 20px;  
} /* 10px s'applique aux marges du haut et du bas, 20 px s'appliquent aux marges de droite et de gauche */  
  
p {  
  margin:10px 0 20px 5px;  
} /* haut, droite, bas, gauche */
```

*S'il n'y a qu'une valeur spécifiée, elle s'applique aux marges des 4 côtés ; s'il y a deux valeurs, la première s'applique aux marges du haut et du bas, la seconde aux marges de droite et de gauche ; s'il y a quatre valeurs : haut, droit, bas et gauche (toujours dans cet ordre).*

Il est possible d'utiliser les propriétés qui définissent indépendamment chaque côté : **margin-top**, **margin-right**, **margin-bottom** et **margin-left**, **padding-top**, **padding-right**, **padding-bottom** et **padding-left**.

### Syntaxe CSS

```
p {  
  margin-top:30px;  
  margin-bottom:30px;  
  padding-top:15px;  
}  
  
#container {  
  width: 350px;  
  border: 1px solid red;  
  margin: 0 auto; /* Le bloc sera centré*/  
}
```

Il est possible de centrer les blocs auxquels on a donné une largeur en donnant la valeur « **auto** » aux propriétés **margin**

`margin:auto;`      ou      `margin:0 auto;`