

# 7 : LES MARGES ET LES BORDURES

## I. Les marges

### Explications :

En CSS, nous allons devoir distinguer deux types de marges : les marges intérieures ("padding") et les marges extérieures ("margin").

Les marges intérieures se trouvent entre le contenu de l'élément et sa bordure. Ainsi, définir une marge intérieure importante va éloigner la bordure de l'élément de son contenu. Si on définit une couleur de fond pour notre élément, celle-ci s'applique également dans l'espace correspondant aux marges intérieures.

### La propriété CSS padding :

Nous allons pouvoir ajouter des marges internes à un élément grâce à la propriété CSS **padding**. Notez déjà que la propriété **padding** est la version raccourcie des propriétés **padding-top**, **padding-left**, **padding-bottom** et **padding-right** qui vont servir à définir les marges internes de chaque côté d'un élément.

Ces propriétés vont pouvoir accepter deux types de valeurs :

- Des valeurs de type longueur, généralement en **px** ou en **em** ;
- Des valeurs de type pourcentage. Dans ce cas, le % indiqué est calculé par rapport à la taille de l'élément parent.

Notez qu'il n'est pas possible de passer des valeurs de **padding** négatives.

### Ajouter une même marge interne de chaque côté d'un élément :

Pour définir une même marge interne de chaque côté d'un élément, il suffit de passer la valeur de la marge intérieure que l'on souhaite appliquer à l'élément à la propriété **padding** en CSS.

Par exemple, pour appliquer une marge intérieure de 25px de chaque côté d'un élément, on écrira en CSS **padding: 25px**. Pour que la marge intérieure soit égale à 10% de la taille de l'élément parent de notre élément, on écrira **padding: 10%** tout simplement.

Pour bien constater l'effet de la propriété **padding**, je vous conseille d'ajouter une couleur de fond ou une bordure aux éléments pour les exemples suivants.

Notez qu'on va tout aussi bien pouvoir utiliser les propriétés **padding-top**, **padding-left**, **padding-bottom** et **padding-right** et leur passer la même valeur pour arriver au même résultat. C'est juste plus long à écrire !

```

<!DOCTYPE html>
<html>
  <head>
    <title>Cours HTML et CSS</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="cours.css">
  </head>

  <body>
    <h1>Un titre de niveau 1</h1>

    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un deuxième paragraphe</p>
    <p class="p3">Un troisième paragraphe</p>
  </body>
</html>

```

index.html

```

p{
  background-color: lightGreen;
  border: 1px solid black;
}

.p2{
  padding: 20px;
}

.p3{
  padding-top: 20px;
  padding-right: 20px;
  padding-bottom: 20px;
  padding-left: 20px;
}

```

cours.css

Résultat :



## Définir des marges internes différentes de chaque côté d'un élément :

Nous allons également pouvoir appliquer des marges intérieures de tailles différentes de chaque côté d'un élément.

Pour cela, nous avons deux façons de faire : soit en passant plusieurs valeurs à la propriété padding, soit en utilisant les propriétés **padding-top**, **padding-left**, **padding-bottom** et **padding-right**.

On va en effet pouvoir passer entre 1 et 4 valeurs à la propriété raccourcie padding :

- En passant une valeur à padding, la valeur passée définira le comportement des 4 marges intérieures de l'élément ;
- En passant deux valeurs à padding, la première valeur passée définira le comportement des marges intérieures supérieure et inférieure de l'élément tandis que la seconde valeur définira le comportement des marges intérieures gauche et droite de l'élément ;
- En passant trois valeurs à padding, la première valeur passée définira le comportement de la marge interne supérieure, la deuxième définira le comportement des marges intérieures gauche et droite tandis que la troisième définira le comportement de la marge interne basse ;
- En passant quatre valeurs à padding, la première valeur passée définira le comportement de la marge interne supérieure, la deuxième définira le comportement de la marge interne droite, la troisième celui de la marge interne basse et la quatrième celui de la marge interne gauche.

Si on choisit d'utiliser les propriétés **padding-top**, **padding-left**, **padding-bottom** et **padding-right**, il suffit de suivre le nom pour comprendre à quelle marge interne est liée chaque propriété (top = haute, left = gauche, bottom = basse, right = droite).

```
<!DOCTYPE html>
<html>
  <head>
    <title>Cours HTML et CSS</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="cours.css">
  </head>

  <body>
    <h1>Un titre de niveau 1</h1>

    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un deuxième paragraphe</p>
    <p class="p3">Un troisième paragraphe</p>
  </body>
</html>
```

index.html

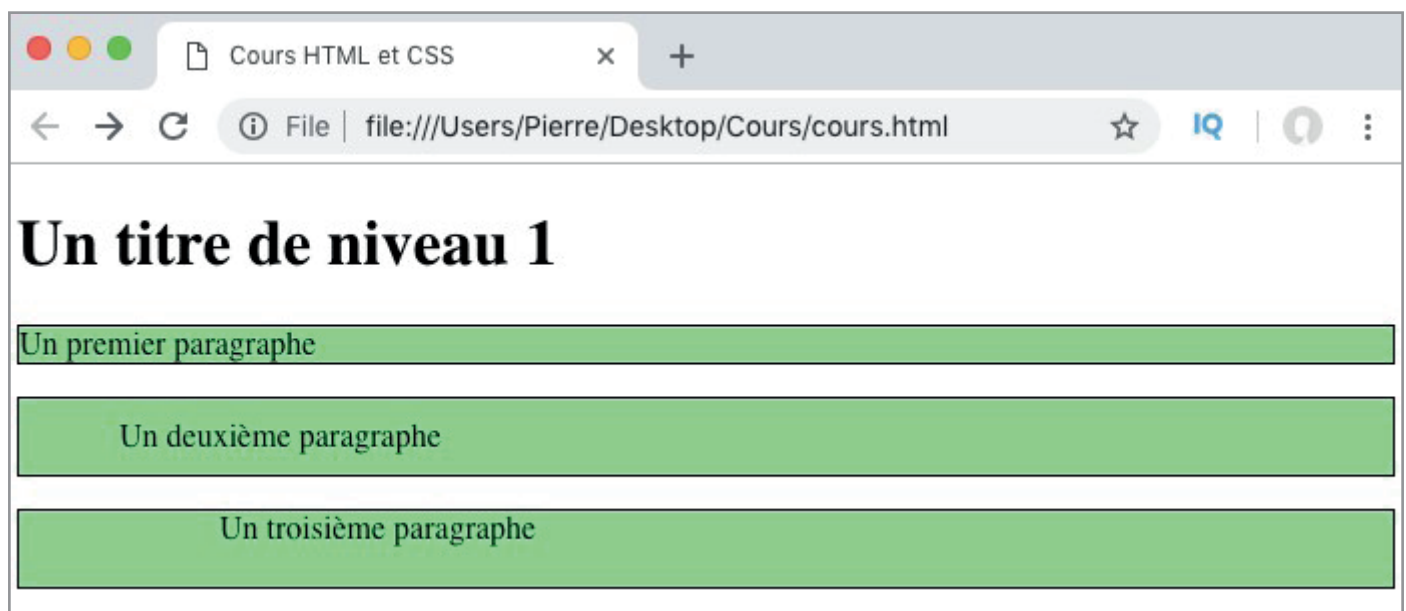
```
p{
  background-color: lightGreen;
  border: 1px solid black;
}

/*Padding top et bottom = 10px
 *Padding left et right = 50px*/
.p2{
  padding: 10px 50px;
}

.p3{
  padding-left: 100px;
  padding-bottom: 20px;
}
```

cours.css

Résultat :



## L'effet du padding sur les éléments environnants :

L'ajout de marges internes ou padding va augmenter la taille totale de l'élément.

En effet, la valeur donnée au padding va venir s'ajouter à celles des propriétés **width** et **height** par défaut.

Regardez plutôt le code ci-dessous pour vous en convaincre :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Cours HTML et CSS</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="cours.css">
  </head>

  <body>
    <h1>Un titre de niveau 1</h1>

    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un deuxième paragraphe</p>
    <p class="p3">Un troisième paragraphe</p>

    <p class="p4">Un quatrième paragraphe avec
    un <span>élément span</span> à l'intérieur</p>

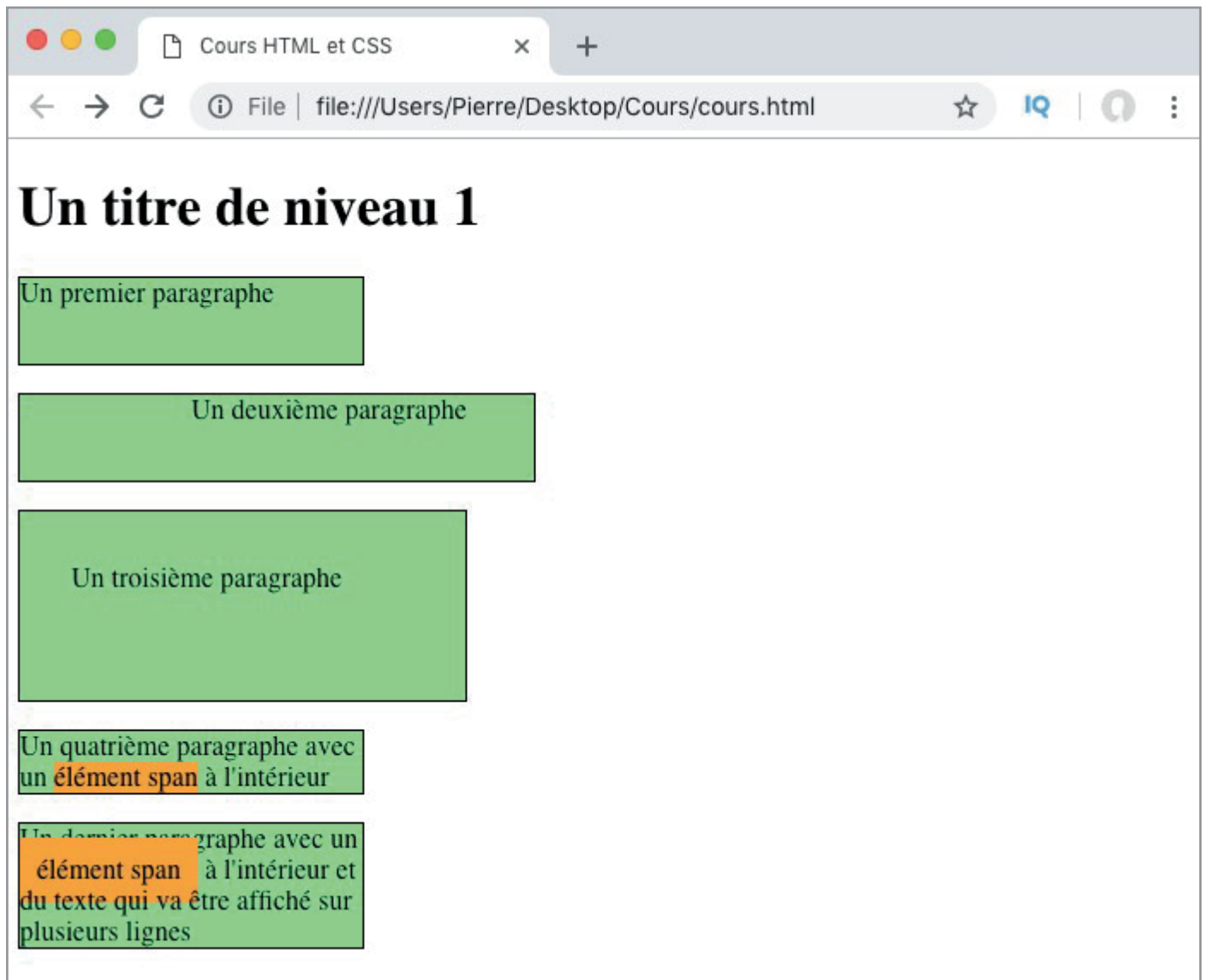
    <p class="p4">Un dernier paragraphe avec un
    <span class="s1">élément span</span> à l'intérieur et du
    texte qui va être affiché sur plusieurs lignes</p>
  </body>
</html>
```

index.html

```
p{
  background-color: lightGreen;
  border: 1px solid black;
  width: 200px;
}
.p1, .p2, .p3{
  height: 50px;
}
.p2{
  padding-left: 100px;
}
.p3{
  padding: 30px;
}
span{
  background-color: orange;
}
.s1{
  padding: 10px;
}
```

cours.css

## Résultat :



Ici, on voit clairement que le padding a un impact sur les dimensions totales des éléments, et ceci qu'ils soient de type block ou inline.

Cependant, l'impact sur les éléments environnants ne va pas être le même selon le type d'élément auquel on applique un padding et selon le padding appliqué.

Ici, il y a notamment un cas à noter : celui des marges internes haute et basse d'un élément de type **inline**. Comme vous pouvez le voir ci-dessus, les marges internes haute et basse sont bien appliquées à mon élément **span s1** (on le voit grâce à la couleur en fond de mon élément).

Cependant, le navigateur ne va pas en tenir compte pour l'affichage et le positionnement des autres éléments. C'est la raison pour laquelle **notre élément span** chevauche le texte des lignes précédente et suivante du paragraphe.

Notez par ailleurs ici les différentes « couches » de texte : la ligne de texte précédent celle du **span** va se trouver en dessous de lui tandis que la suivante va être au-dessus (on voit que la couleur de fond du **span** cache le texte précédent mais est sous le texte de la ligne suivante).

Encore une fois, notez que le **padding** est bien appliqué de chaque côté pour chaque type d'éléments, cependant les **padding** haut et bas n'auront pas d'impact sur le positionnement d'un élément **inline** ni sur celui des éléments autour de lui.



# I. Les caractéristiques des bordures en CSS

## Explications :

Les bordures vont être définies par trois caractéristiques en CSS : une épaisseur (ou largeur), un style et une couleur.

Nous allons pouvoir définir ces différentes caractéristiques d'un coup au sein de la propriété raccourcie **border** ou les définir une à une avec chacune des propriétés de type **border-**. Plus précisément :

- La propriété **border-width** va nous permettre de définir la largeur (ou « l'épaisseur ») d'une bordure ;
- La propriété **border-style** va nous permettre de définir le style d'une bordure ;
- La propriété **border-color** va nous permettre de définir la couleur d'une bordure.

## Définir la largeur ou l'épaisseur d'une bordure :

Nous allons pouvoir utiliser différents types de valeurs pour définir la largeur d'une bordure :

- Une valeur de type « mot clef » à choisir parmi **thin** (bordure fine), **medium** (bordure moyenne) et **thick** (bordure épaisse) ;
- Une valeur de type « longueur » **en px ou en em** par exemple.

Généralement, nous utiliserons des unités en **px** pour définir la largeur de nos bordures.

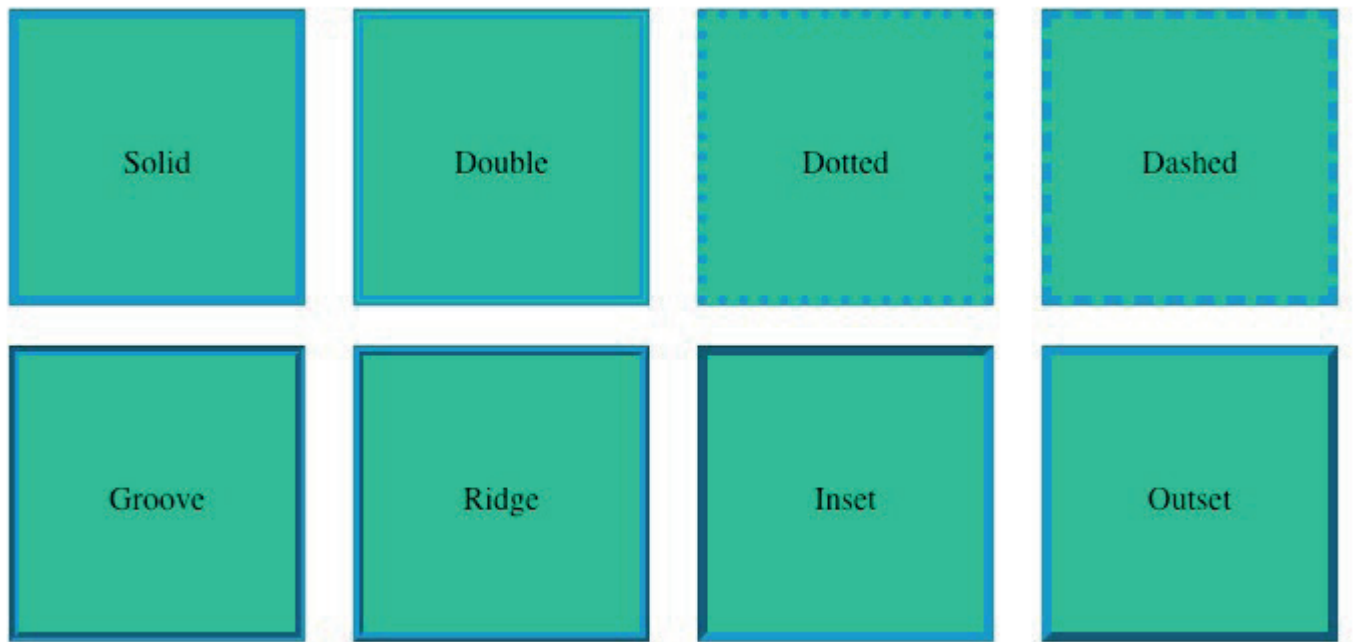
## Choisir le style d'une bordure :

Le « style » d'une bordure correspond à son aspect : une bordure peut prendre la forme d'un simple trait, d'un trait double, ou être constituée de pointillés, avoir un effet 3D, etc.

Pour définir le style d'une bordure, nous allons devoir choisir parmi les mots clefs suivants :

Valeur	Description
<b>solid</b>	Bordure solide simple (un trait)
<b>double</b>	Bordure solide double
<b>dotted</b>	Bordure en pointillés
<b>dashed</b>	Bordure constituée de tirets
<b>groove</b>	Bordure incrustée avec effet 3D. L'effet produit est l'inverse de <b>ridge</b>
<b>ridge</b>	Bordure en relief avec effet 3D. L'effet produit est l'inverse de <b>groove</b>
<b>inset</b>	La bordure donne l'effet que la boîte représentant l'élément est enfoncée dans la page. L'effet produit est l'inverse de <b>outset</b>
<b>outset</b>	La bordure donne l'effet que la boîte représentant l'élément est en relief par rapport au reste de la page. L'effet produit est l'inverse de <b>inset</b>

Chaque mot clef va créer un type de bordure différent. Voici à quoi correspond chaque effet :



## Définir la couleur d'une bordure :

Finalement, nous allons devoir définir la couleur d'une bordure. Pour cela, nous allons pouvoir piocher parmi toutes les valeurs de type « couleur » connues et notamment :

- Les notations de type « nom de couleur » ;
- Les notations hexadécimales ;
- Les notations RGB() et RGBa() ;
- Les notations HSL() et HSLa().

## Exemples de création de bordures en CSS :

La suite de cette leçon va être pour nous l'occasion de s'exercer et de créer toutes sortes de bordures. Je vous rappelle ici que nous aborderons les bordures arrondies dans la prochaine leçon.

Commençons déjà avec des exemples simples de bordures créées en CSS en appliquant ce que nous avons vu ci-dessous.

Ici, nous allons créer 4 types de bordures différentes : une bordure simple, une bordure double, une avec effet 3D de type « ridge » et une avec un style « outset ».

Nous allons pouvoir faire cela en utilisant soit les propriétés **border-width**, **border-style** et **border-color**, soit la notation raccourcie **border**.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Cours HTML et CSS</title>
    <meta charset="utf-8">
    <link rel="stylesheet" href="cours.css">
  </head>
  <body>
    <h1>Titre de niveau 1</h1>
    <p class="p1">Solid</p>
    <p class="p2">Double</p>
    <p class="p3">Ridge</p>
    <p class="p4">Outset</p>
  </body>
</html>
```

index.html



```

p{
  padding: 20px;
  margin: 20px 10px;
}

.p1{
  border-width: 5px;
  border-style: solid;
  border-color : #09C;
}
.p2{
  border: 15px double #09C;
}
.p3{
  background-color: #0C9;
  border: 5px ridge #09C;
}
.p4{
  background-color: #0C9;
  border: 5px outset #09C;
}

```

cours.css

Résultat :



Ici, nous nous contenterons d'ajouter des bordures différentes autour de nos paragraphes. J'ai également rajouté une couleur de fond aux deux derniers paragraphes afin que l'on puisse bien observer l'effet de 3D des deux dernières bordures.

Notez une chose intéressante par rapport à notre bordure double : la taille de la bordure correspond à la taille totale de la bordure, c'est-à-dire dans ce cas du double trait et de l'espace entre ces deux traits. La largeur totale va être répartie régulièrement : en indiquant une bordure double de **15px**, chaque élément de la bordure (les deux traits plus l'espace entre les traits) va faire 5px de large (15px / 3).

## Définir des bordures différentes pour chaque côté d'un élément :

Le CSS va également nous permettre de définir chacune des quatre bordures de nos éléments indépendamment les unes des autres afin de pouvoir appliquer des effets intéressants.

Nous avons deux façons de faire cela selon que nous utilisons la notation raccourcie **border** pour définir nos bordures ou que nous utilisons chacune des propriétés **border-**.

Dans le cas où nous souhaitons utiliser **border**, nous allons devoir utiliser 4 sous propriétés CSS qui sont :

- **border-top** pour définir l'aspect (taille, style et couleur) de la bordure supérieure de l'élément ;
- **border-right** pour définir l'aspect (taille, style et couleur) de la bordure droite de l'élément ;
- **border-bottom** pour définir l'aspect (taille, style et couleur) de la bordure inférieure de l'élément ;
- **border-left** pour définir l'aspect (taille, style et couleur) de la bordure gauche de l'élément.

Bon à savoir : Les propriétés **border-top**, **border-right**, **border-bottom** et **border-left** sont à nouveau des écritures raccourcies des propriétés CSS **border-top-width**, **border-top-style**, **border-top-color**, etc.

Dans le cas où nous utilisons les propriétés **border-width**, **border-style** et **border-color** alors nous pourrions indiquer 4 valeurs à la suite qui définiront le comportement des bordures supérieure, droite, inférieure et gauche dans cet ordre.

Notez ici qu'il est tout à fait possible de ne renseigner qu'une valeur pour l'une de ces 3 propriétés si vous voulez que vos 4 bordures aient le même comportement. Vous pouvez également ne mentionner que deux valeurs : dans ce cas, la première valeur définira le comportement des bordures supérieure et inférieure tandis que la seconde définira le comportement des bordures droite et gauche de l'élément.

Regardez plutôt les exemples ci-dessous qui résument toutes les situations possibles pour bien comprendre :

```

<!DOCTYPE html>
<html>
  <head>
    <title>Cours HTML et CSS</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="cours.css">
  </head>

  <body>
    <h1>Un titre de niveau 1</h1>

    <p class="p1">Bordure haute 1px tirets rouge foncée,
    droite 2px solide rouge foncée,
    basse 3px tirets rouge foncée,
    gauche 4px solide rouge foncée</p>
    <p class="p2">Idem</p>
    <p class="p3">Idem</p>
    <p class="p4">Bordure gauche 5px solide rouge foncée</p>
    <p class="p5">Bordure basse 2px solide rouge foncée</p>
  </body>
</html>

```

index.html

```

p{
  padding: 20px;
  margin: 20px 10px;
  background-color: #f1f1f1;
}

.p1{
  border-width: 1px 2px 3px 4px;
  border-style: dashed solid dashed solid;
  border-color: #C04 #C04 #C04 #C04; /*Rouge foncé*/
}

.p2{
  border-width: 1px 2px 3px 4px;
  border-style: dashed solid;
  border-color: #C04;
}

.p3{
  border-top: 1px dashed #C04;
  border-right: 2px solid #C04;
  border-bottom: 3px dashed #C04;
  border-left : 4px solid #C04;
}

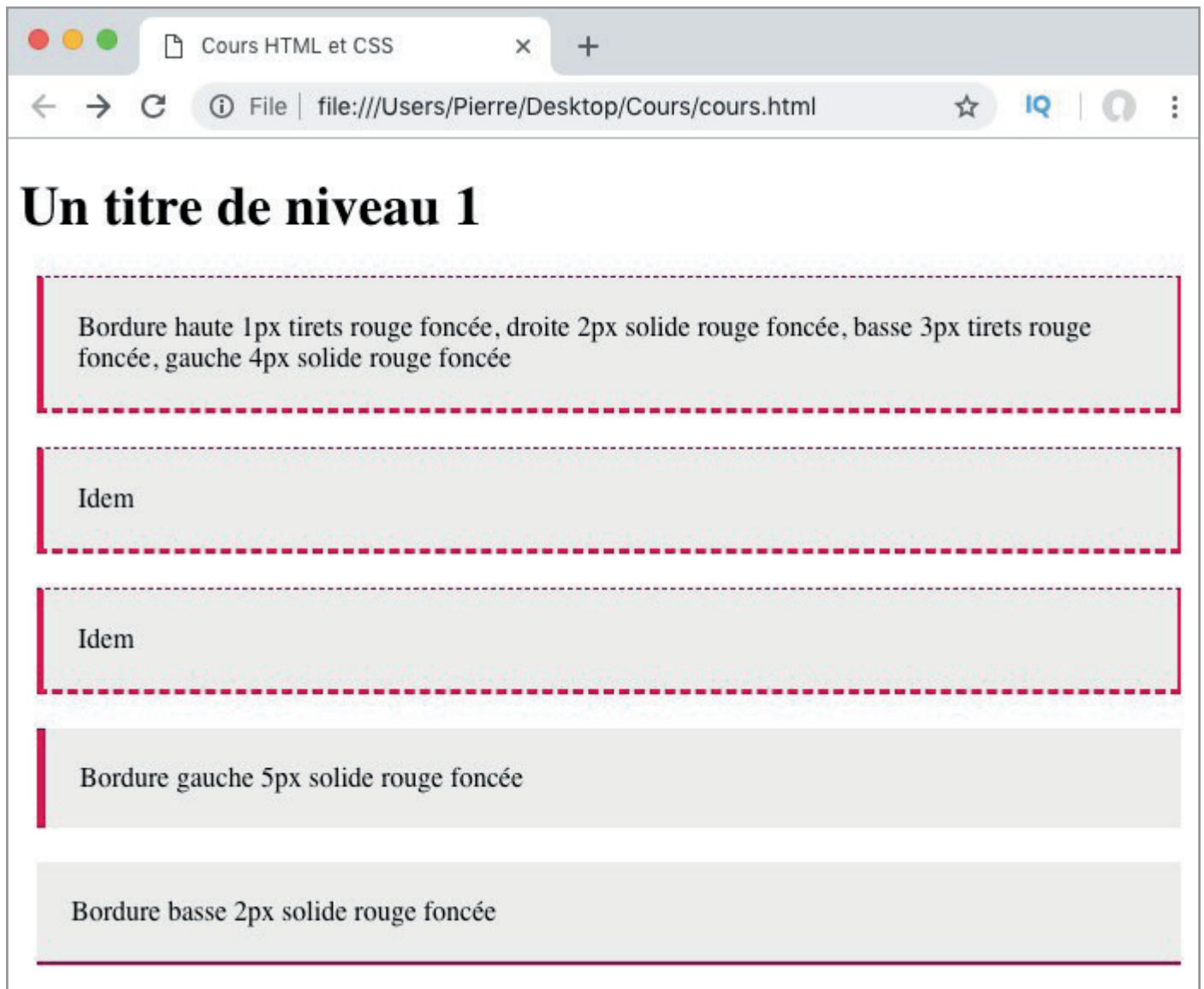
.p4{
  border-left-width: 5px;
  border-left-style: solid;
  border-left-color: #C04;
}

.p5{
  border-bottom: 2px solid #C04;
}

```

cours.css

## Résultat :



## Créer des bordures semi transparentes :

Nous allons encore pouvoir ajouter un effet de transparence à nos bordures en utilisant tout simplement une notation RGBA lorsque l'on précisera la paramètre couleur de notre bordure.

Notez ici qu'utiliser la propriété CSS **opacity** ne produirait pas le comportement voulu puisque l'effet de transparence serait appliqué à l'élément entier et non pas seulement à la bordure comme on le souhaite.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Cours HTML et CSS</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="cours.css">
  </head>

  <body>
    <h1>Un titre de niveau 1</h1>

    <p class="p1">Un premier paragraphe</p>
    <p class="p2">Un deuxième paragraphe</p>
    <p class="p3">Un troisième paragraphe</p>
  </body>
</html>
```

index.html

```

p{
  padding: 20px;
  margin: 20px 10px;
  background-color: lightBlue;
}
.p1{
  border: 5px solid black;
}

.p2{
  border: 5px solid black;
  opacity: 0.5;
}

.p3{
  border: 5px solid RGBA(0, 0, 0, 0.5);
}

```

cours.css

Résultat :



## Les bordures dans le modèle des boîtes :

Les bordures d'un élément HTML se situent entre les marges internes et externes de l'élément. La taille des bordures va par défaut venir s'ajouter aux dimensions de l'élément définies avec les propriétés **width** et **height** ainsi qu'aux marges internes.

Notez que nous allons pouvoir définir des bordures de la même façon pour des éléments de type **block** et **inline** et que ces bordures auront exactement le même comportement.