

HTML5



HTML Tutorial

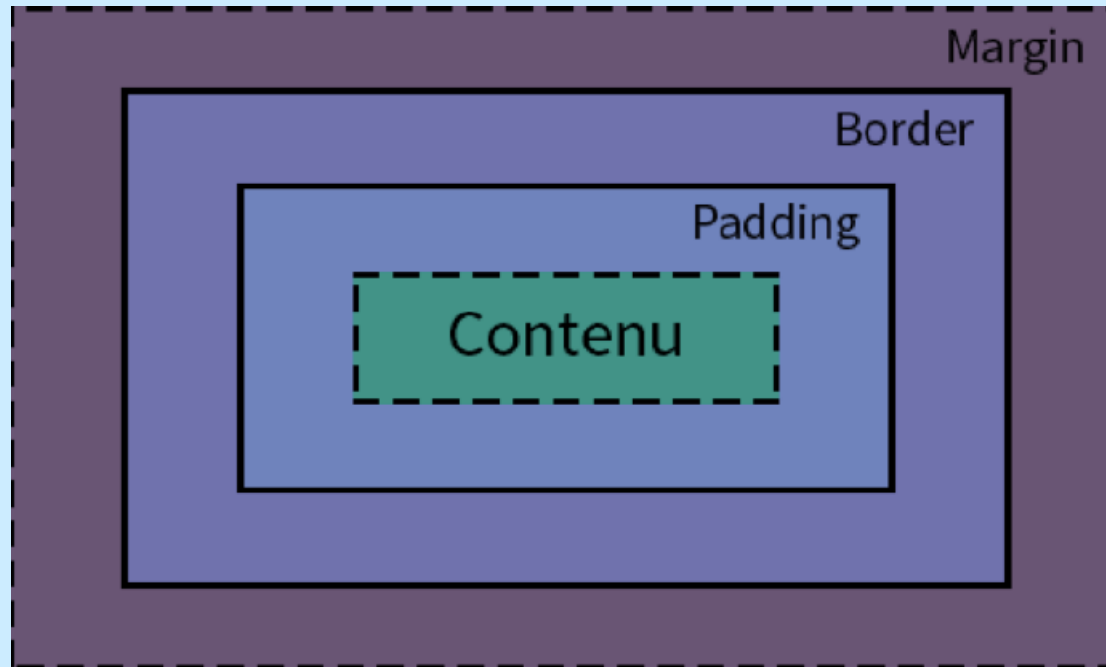
LE MODELE DES BOITES

Importance et définition

- Le **modèle des boîtes** est un concept essentiel et central pour la mise en page d'une page web.
- Le modèle des boîtes nous dit que « tout élément HTML peut être considéré comme une boîte rectangulaire », qu'il soit de type block ou inline.
- Autour de la « boîte » contenant l'élément en soi, on va pouvoir dessiner d'autres boîtes contenant la marge intérieure (padding), les bordures et finalement la marge extérieure de l'élément.

Explication et illustrations du modèle des boîtes

- Première illustration théorique du modèle des boîtes



Explication et illustrations du modèle des boîtes

- La première boîte (la boîte centrale) représente le contenu de l'élément ou l'élément en soi.
- Ensuite, autour du contenu, la deuxième boîte contient en plus la marge intérieure de l'élément qu'on appelle en CSS le "padding".
- La troisième boîte va contenir en plus la bordure de l'élément.
- Enfin, la dernière boîte va également contenir la marge extérieure de l'élément.

Explication et illustrations du modèle des boîtes

- Ainsi :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <div>
      <h1>Le modèle des boîtes</h1>
      <p>Un premier paragraphe.</p>
      <p>Un autre paragraphe</p>
    </div>
  </body>
</html>
```

Explication et illustrations du modèle des boîtes

- (Suite)



- Pour tester le code : <http://jsbin.com/gopega/61/edit?html,css,output>

HTML Tutorial

LA LARGEUR (WIDTH) ET LA
HAUTEUR (HEIGHT) EN CSS

Types d'éléments HTML et valeurs par défaut

- Tout élément HTML possède une largeur et une hauteur. La hauteur par défaut d'un élément HTML est déterminée par son contenu. Ainsi, un paragraphe, selon qu'il occupe une ou deux lignes, ne possèdera pas la même hauteur par défaut.
- La largeur d'un élément, en revanche, est avant tout déterminée par son type : en effet, les éléments de type block occuperont par défaut tout l'espace disponible. Les éléments de type inline, en revanche, n'occuperont que la largeur nécessaire à leur contenu.

Modifier la hauteur et la largeur d'un élément

- Pour modifier la hauteur par défaut d'un élément, on utilise la propriété `height`. Pour modifier la largeur par défaut d'un élément, on utilise la propriété `width`.
- Ces propriétés peuvent prendre des valeurs absolues, relatives (utiles dans le cas d'un site responsive) ou la valeur « auto » (la hauteur sera calculée automatiquement par le navigateur).

Modifier la hauteur et la largeur d'un élément

- Soit par exemple l'exemple suivant où l'on définit la largeur et la hauteur d'un élément div. On lui donne également une couleur de fond afin de bien voir sa taille.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <div>
      <h1>Le modèle des boîtes</h1>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

```
div{
  background-color: #088;
  width: 500px;
  height:300px;
}
```

Modifier la hauteur et la largeur d'un élément

- (Suite)



- Pour tester le code : <http://jsbin.com/gopega/62/edit?html,css,output>

Hauteur, largeur et modèle des boîtes

- Pour régler efficacement la hauteur et la largeur d'un élément, il faut avant tout bien avoir compris le modèle des boîtes.
- Si l'on ne fait pas attention, on risque de créer des éléments plus grands au final que leurs éléments parents, ce qui est généralement considéré comme une mauvaise pratique en HTML et en CSS.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <div>
      <h1>Le modèle des boîtes</h1>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

- Pour tester le code page suivante :
<http://jsbin.com/gopega/63/edit?html,css,output>

Hauteur, largeur et modèle des boîtes

- (Suite)



Hauteur, largeur et modèle des boîtes

- Dans l'exemple précédent, on a donné une largeur égale à 400px au paragraphe portant l'attribut `class="para1"`. Cependant, on a également défini des marges intérieures de 60px de chaque côté ainsi qu'une bordure de 5px.
- L'élément p fait donc au final $400 + 2*60 + 2*5 = 530\text{px}$ de large, soit 30px de plus que le div, son élément parent, ce qui fait qu'il déborde !

HTML Tutorial

LES BORDURES EN CSS

Définir des bordures en CSS

- L'espace pris par la bordure va se trouver entre la marge intérieure et la marge extérieure d'un élément HTML.
- Nous pouvons définir les bordures d'un élément de différentes manières en CSS : soit en utilisant les trois propriétés **border-style**, **border-width** et **border-color**, soit en utilisant directement la notation courte **border**.
- Exemple :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <div>
      <h1>Les bordures</h1>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

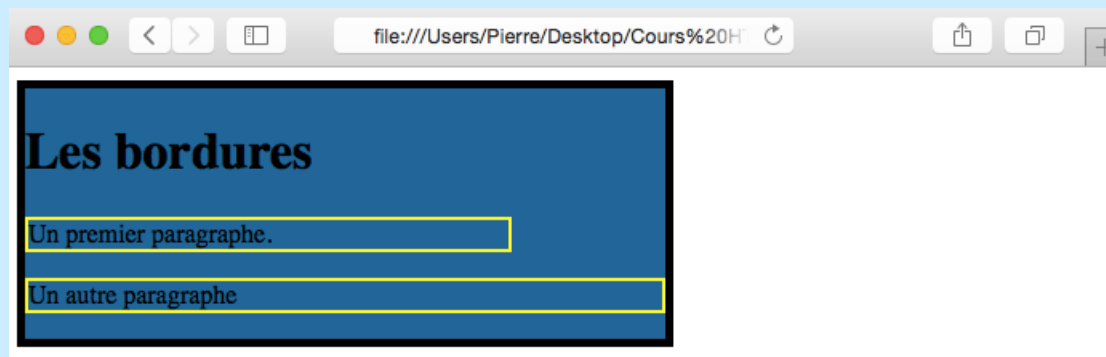
Définir des bordures en CSS

- (Suite)

```
div{
  background-color: #269;
  width: 400px;
  border: 5px solid black;
}

.para1{
  /*Le contenu du paragraphe va prendre 75% de la largeur de
  *son élément parent, c'est-à-dire du div*/
  width: 75%;
  border-width: 2px;
  border-style: solid;
  border-color: yellow;
}

.para2{
  border: 2px solid yellow;
}
```



- Pour tester le code : <http://jsbin.com/gopega/64/edit?html,css,output>

Définir des bordures en CSS

- Il est identique d'utiliser les trois propriétés de type border- ou directement la propriété border avec les trois valeurs requises.
- Noter qu'en utilisant la propriété border on doit préciser dans l'ordre les valeurs associées à border-width, border-style et enfin border-color.

Les styles de bordures

- La propriété border-style peut prendre huit valeurs différentes qui vont définir l'allure générale (le style) de notre bordure.



Les styles de bordures

- Les valeurs les plus couramment utilisées sont solid, dotted et dashed.
- Attention avec les valeurs groove, ridge, inset et outset qui restent parfois encore mal supportées par certains navigateurs. Ces quatre dernières valeurs sont censées ajouter un effet 3D à nos bordures.
- Exemple utilisant différentes valeurs pour border-style :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <div>
      <h1>Les bordures</h1>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

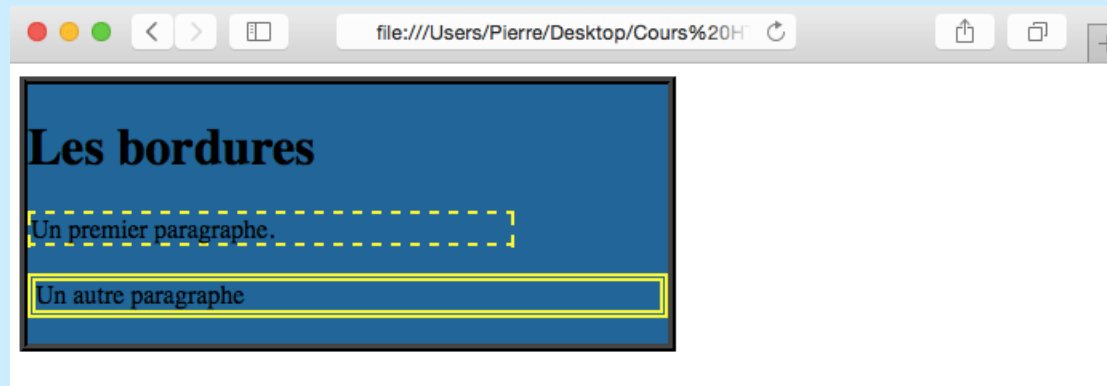
Les styles de bordures

- (Suite)

```
div{
  background-color: #269;
  width: 400px;
  border: 5px ridge #444;
}

.para1{
  /*Le contenu du paragraphe va prendre 75% de la largeur de
  *son élément parent, c'est-à-dire du div*/
  width: 75%;
  border-width: 2px;
  border-style: dashed;
  border-color: yellow;
}

.para2{
  border: 5px double yellow;
}
```



- Pour tester le code : <http://jsbin.com/gopega/65/edit?html,css,output>

Bordures haute, droite, basse et gauche

- Le CSS permet également de définir des bordures différentes au dessus, à droite, en dessous et à gauche des éléments HTML.
- Pour cela, il faut ajouter les mots clefs top (haut), right (droit), bottom (bas) et left (gauche) dans le nom des propriétés.
- Exemple :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <div>
      <h1>Les bordures</h1>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

Bordures haute, droite, basse et gauche

- (Suite)

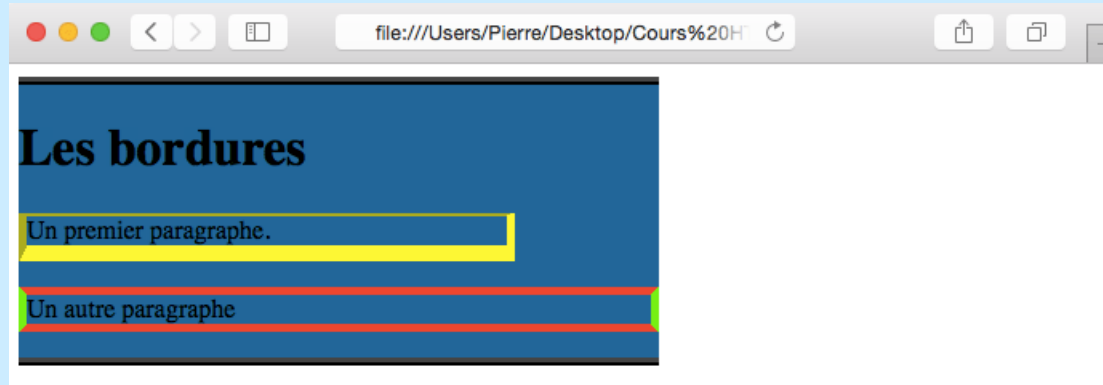
```
div{
  background-color: #269;
  width: 400px;
  border-bottom: 5px ridge #444;
  border-top: 5px ridge #444;
}

.para1{
  width: 75%;
  border-top-width: 2px;
  border-bottom-width: 10px;
  border-right-width: 5px;
  border-left-width: 5px;
  border-style: inset;
  border-color: yellow;
}

.para2{
  border-top: 5px solid red;
  border-right: 5px solid lime;
  border-bottom: 5px solid red;
  border-left: 5px solid lime;
}
```


Bordures haute, droite, basse et gauche

- (Suite)



- Pour tester le code : <http://jsbin.com/gopega/66/edit?html,css,output>

Les bordures arrondies

- L'une des fonctionnalités nouvelles apportées par le CSS3 qui figure sans conteste parmi les plus attendues par les développeurs a été la possibilité de créer des bordures arrondies.
- On peut créer des bordures arrondies avec la propriété **border-radius**. Cette propriété va prendre une valeur en pixels qui va correspondre à la valeur du radius de la bordure, et elle s'utilise de concert avec la propriété border.
- En outre, on pourra définir des bordures plus ou moins arrondies de chaque côté d'un élément HTML, cette fois-ci en utilisant les mots clefs **top-right**, **bottom-right**, **bottom-left**, **top-left**.

Les bordures arrondies

- Illustration :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

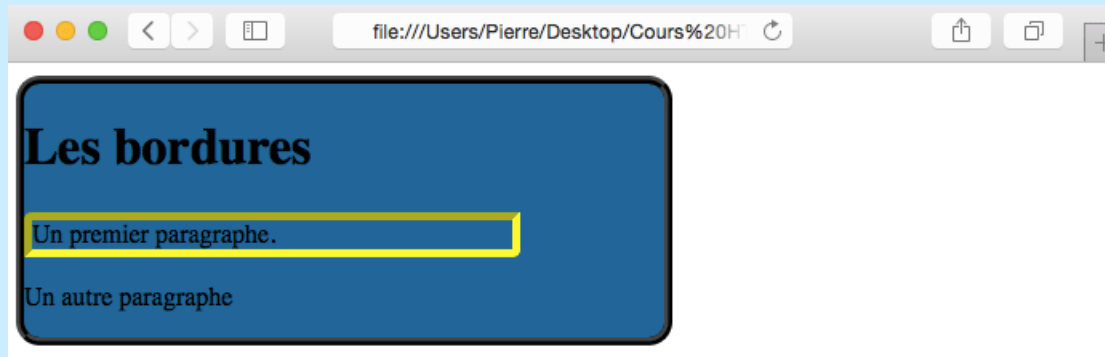
  <body>
    <div>
      <h1>Les bordures</h1>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

```
div{
  background-color: #269;
  width: 400px;
  border: 5px ridge #444;
  border-radius: 15px;
}

.para1{
  width: 75%;
  border-width: 5px;
  border-style: inset;
  border-color: yellow;
  border-top-left-radius: 5px;
  border-bottom-right-radius: 5px;
}
```

Les bordures arrondies

- (Suite)



- Pour tester le code : <http://jsbin.com/gopega/67/edit?html,css,output>

HTML Tutorial

LES MARGES EN CSS

Marges intérieures et marges extérieures

- En CSS, on distingue deux types de marges : les marges intérieures (le padding) et les marges extérieures (le 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 l'élément, celle-ci s'applique également dans l'espace correspondant aux marges intérieures.
- Les marges extérieures, au contraire, vont définir l'espace autour d'un élément. Les marges extérieures se trouvent en dehors des bordures d'un élément et servent généralement à éloigner un élément d'un autre. Elles ne sont pas affectées par la couleur de fond donnée à un élément.

Les marges intérieures

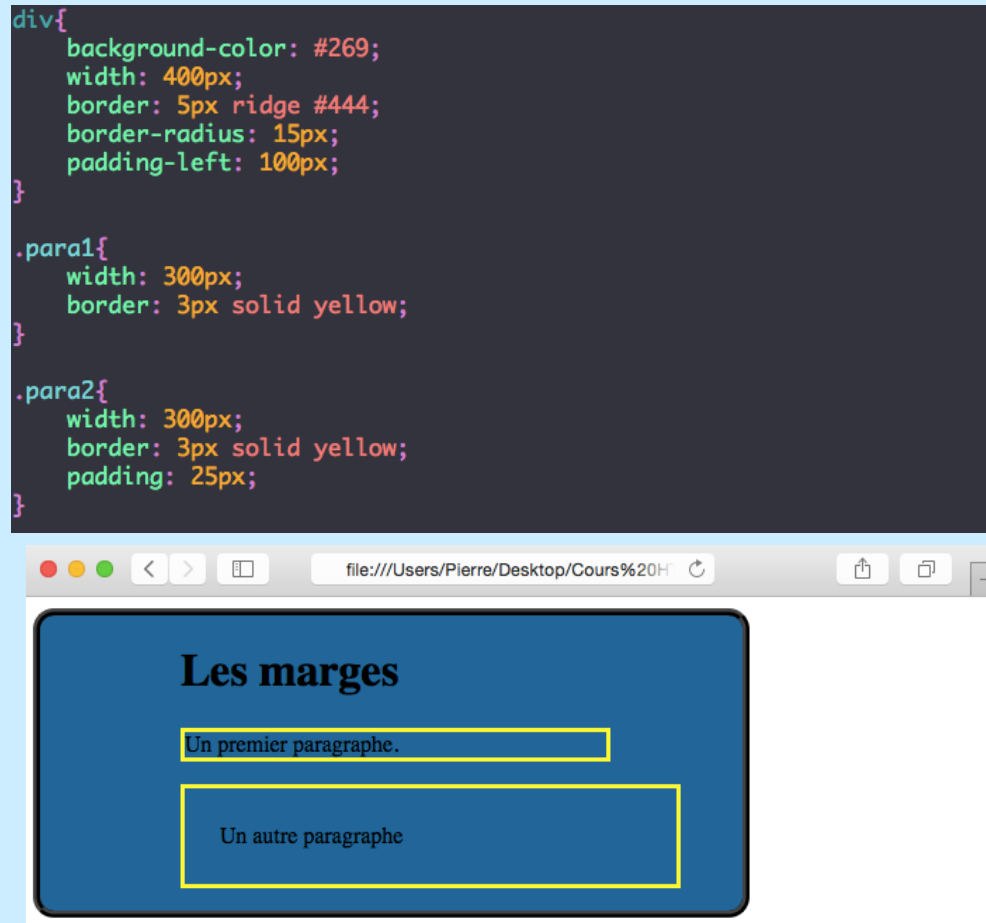
- Les marges intérieures d'un élément sont définies en CSS grâce à la propriété padding. Cette propriété peut prendre des valeurs absolues (px) ou relatives (%).
- Noter que l'on peut définir des marges intérieures haute, droite, basse et gauche de tailles diverses avec padding-top, padding-right, padding-bottom et padding-left.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <div>
      <h1>Les marges</h1>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

Les marges intérieures

- (Suite)



- Pour tester le code : <http://jsbin.com/gopega/68/edit?html,css,output>

Les marges extérieures

- Les marges extérieures sont définies en CSS grâce à la propriété `margin`.
- Cette propriété accepte en valeurs des valeurs absolues, relatives, ainsi que les valeurs `inherit` ou `auto`. Cette dernière valeur, `auto`, va souvent être utile pour centrer horizontalement un élément par rapport à un élément parent.
- De même que précédemment, on peut définir des marges extérieures haute, droite, basse et gauche d'importances diverses grâce à `margin-top`, `margin-right`, `margin-bottom` et `margin-left`.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <div>
      <h1>Les marges</h1>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

Les marges extérieures

- (Suite)



- Pour tester le code : <http://jsbin.com/gopega/69/edit?html,css,output>

Les marges extérieures

- On peut également centrer horizontalement un élément par rapport à son élément parent en utilisant la valeur auto. Pour que cela fonctionne, il faut que deux critères soient réunis : l'élément doit être de type block et doit posséder une largeur définie.
- Voici un exemple avec l'élément div :



- Pour tester le code : <http://jsbin.com/gopega/70/edit?html,css,output>

Notations raccourcies padding et margin

- On peut aussi définir des marges intérieures et extérieures de tailles différentes pour chaque côté de nos éléments en utilisant directement les propriétés padding et margin plutôt que padding-top, padding-right, etc.
- Pour cela, on enchaîne les quatre valeurs à la suite. La première valeur correspond à la marge haute, la seconde à la marge droite, la troisième à la marge basse et la dernière à la marge gauche.
- On peut également ne préciser que deux valeurs. Dans ce cas, la première correspond aux marges haute et basse et la seconde aux marges droite et gauche.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

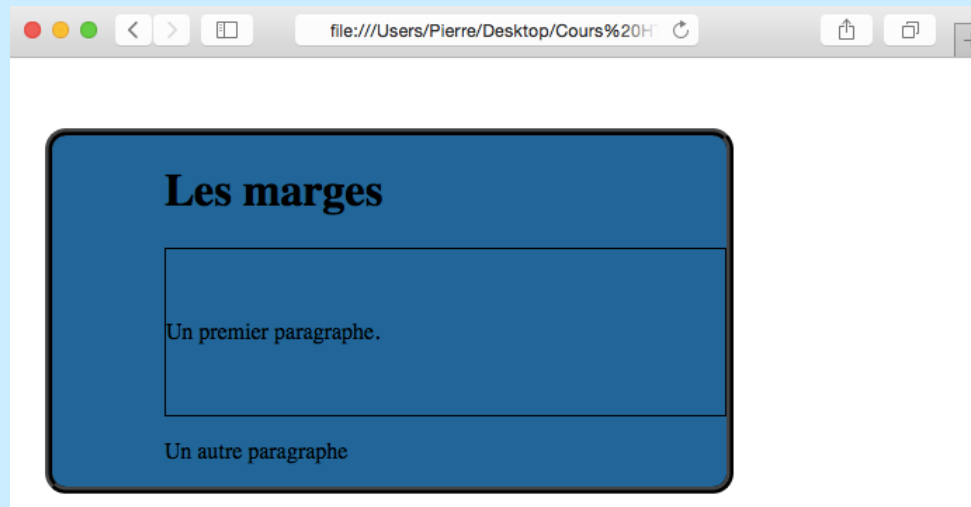
  <body>
    <div>
      <h1>Les marges</h1>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

Notations raccourcies padding et margin

- (Suite)

```
div{
  background-color: #269;
  width: 400px;
  border: 5px ridge #444;
  border-radius: 15px;
  margin: 50px 0px 0px 20px;
  padding: 0px 0px 0px 80px;
}

.paral{
  padding: 50px 0px;
  border: 1px solid black;
}
```



- Pour tester le code : <http://jsbin.com/gopega/71/edit?html,css,output>

HTML Tutorial

LES OMBRES DES BOITES

La propriété box-shadow

- On peut créer des ombres autour des boîtes (à l'extérieur ou à l'intérieur des bordures) grâce à la propriété CSS **box-shadow**.
- Tout comme la propriété text-shadow, box-shadow va prendre quatre valeurs différentes dans l'ordre suivant :
 - Le déplacement horizontal (vers la droite ou la gauche) de l'ombre ;
 - Le déplacement vertical (vers le bas ou le haut) de l'ombre ;
 - Le rayon de propagation de l'ombre ;
 - La couleur de l'ombre.
- Noter que l'on peut également créer des ombres intérieures (à l'intérieur des bordures d'un élément) en rajoutant une cinquième valeur à box—shadow : la valeur « **inset** ».

La propriété box-shadow

- Exemples :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <div>
      <h1>Les ombres des boîtes</h1>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```


La propriété box-shadow

- (Suite)

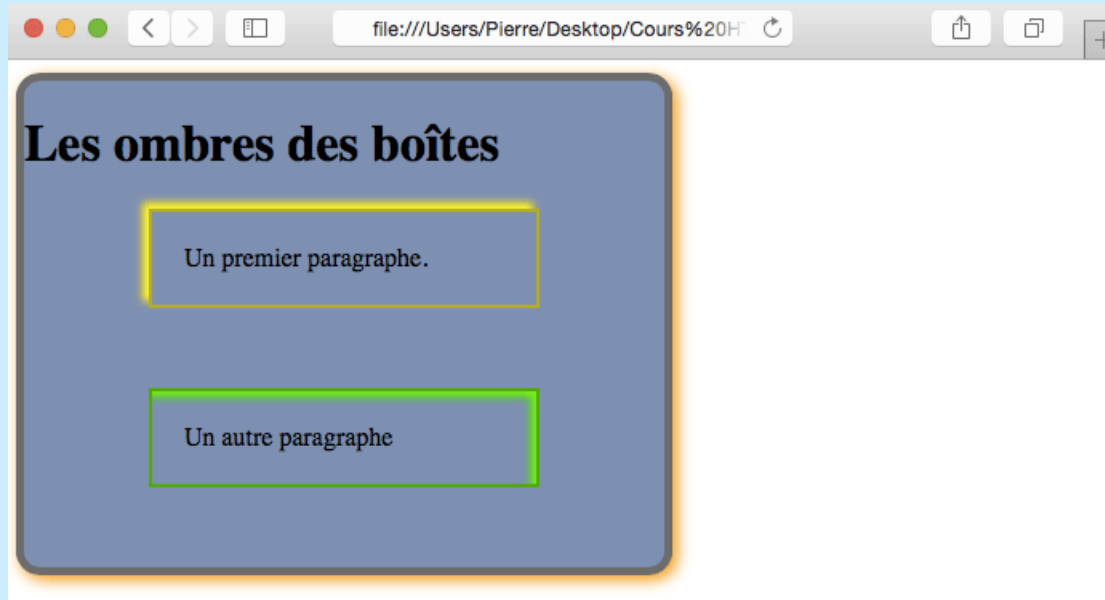
```
div{
  background-color: #89B;
  width: 400px;
  border: 5px solid #777;
  border-radius: 15px;
  box-shadow: 3px 3px 9px orange;
}

.para1{
  width: 50%;
  margin: 0px auto;
  padding: 20px;
  border: 2px solid #BB0;
  box-shadow: -4px -4px 5px yellow;
}

.para2{
  width: 50%;
  margin: 50px auto;
  padding: 20px;
  border: 2px solid #0B0;
  box-shadow: -4px 4px 5px lime inset;
}
```

La propriété box-shadow

- (Suite)



- Pour tester le code : <http://jsbin.com/gopega/72/edit?html,css,output>

La propriété box-shadow

Explication :

- On commence par créer une ombre de couleur orange autour de l'élément div. On applique une projection de 3 pixels vers la droite et 3 pixels vers le bas pour cette première ombre, ainsi qu'un rayon de propagation de 9 pixels.
- Pour l'ombre autour de notre premier paragraphe, on a cette fois-ci indiqué des déplacements horizontal et vertical négatifs, ce qui a pour effet de projeter notre ombre en haut à gauche plutôt qu'en bas à droite.
- Enfin, on utilise la valeur inset pour l'ombre de notre second paragraphe afin de créer une ombre intérieure plutôt qu'extérieure.

HTML Tutorial

LA PROPRIETE CSS DISPLAY

La propriété CSS display

- Par défaut, les éléments HTML vont s'afficher soit sous forme de «bloc», soit «en ligne». C'est la différence majeure entre les éléments de type block et inline.
- La propriété display est une propriété CSS puissante, qui va permettre de maîtriser la façon dont un élément va être affiché, et ainsi de gérer la mise en page des éléments.
- La propriété display peut prendre de nombreuses valeurs permettant de gérer précisément la façon dont chaque élément va être affiché. Nous allons voir les valeurs les plus utilisées.

Display : inline

- Lorsque l'on donne la valeur inline à la propriété display, l'élément ciblé va se comporter comme un élément de type inline et donc n'occuper que la largeur qui lui est strictement nécessaire.
- Par exemple, en appliquant un display:inline à des éléments HTML p, ceux-ci vont venir se coller les uns à côté des autres selon la place disponible.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

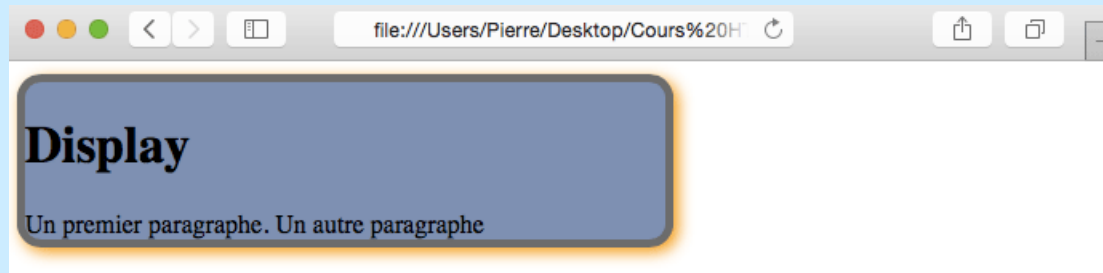
  <body>
    <div>
      <h1>Display</h1>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

Display : inline

- (Suite)

```
div{
  background-color: #89B;
  width: 400px;
  border: 5px solid #777;
  border-radius: 15px;
  box-shadow: 3px 3px 9px orange;
}

.para1, .para2{
  display: inline;
}
```



- Pour tester le code : <http://jsbin.com/gopega/73/edit?html,css,output>

Display : block

- En attribuant la valeur block à la propriété display, les éléments ciblés vont se comporter comme des éléments HTML de type block et ainsi prendre toute la largeur disponible.
- On peut par exemple appliquer un display:block à un élément HTML span(qui est par défaut de type inline) afin d'observer la modification du comportement de celui-ci.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <div>
      <h1>Display</h1>
      <p class="para1">Un <span>premier</span> paragraphe.</p>
      <p class="para2">Un <span id="sp2">autre</span> paragraphe</p>
    </div>
  </body>
</html>
```


Display : block

- (Suite)



- Pour tester le code : <http://jsbin.com/gopega/74/edit?html,css,output>

Display : block

- Dans cet exemple, on a appliqué une bordure aux deux éléments HTML span afin de bien voir la modification du comportement.
- Ensuite, on n'a appliqué un display:block qu'au deuxième élément span. Celui-ci se comporte alors comme un élément de type block et occupe toute la largeur disponible.

Display : inline-block

- En HTML, les éléments sont soit de type inline, soit de type block. La propriété display va permettre de créer des éléments d'un troisième type qui est une combinaison des deux premiers grâce à sa valeur inline-block.
- La valeur inline-block emprunte certaines propriétés des éléments de type block et certaines propriétés des éléments de type inline.
- Ainsi, l'élément en soi va être de type inline tandis que ce qu'il contient (« l'intérieur de la boîte ») va être considéré comme étant de type block.
- Cela va permettre entre autres d'afficher plusieurs éléments côte-à-côte tout en maîtrisant précisément la taille de chacun d'entre eux.

Display : inline-block

- (Suite)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <div>
      <h1>Display</h1>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

Display : inline-block

- (Suite)

```
div{
  background-color: #89B;
  width: 400px;
  border: 5px solid #777;
  border-radius: 15px;
  box-shadow: 3px 3px 9px orange;
}

.para1{
  display: inline-block;
  border: 1px solid black;
  width: 55%;
}

.para2{
  display: inline-block;
  border: 1px solid black;
  width: 40%;
}
```



- Pour tester le code : <http://jsbin.com/gopega/75/edit?html,css,output>

Display : none

- La valeur none permet tout simplement de ne pas afficher un élément.
- Cela peut se révéler très pratique, notamment lorsqu'on veut modifier l'affichage des pages selon l'appareil utilisé par les visiteurs (ordinateur de bureau, téléphone portable, tablette, etc.).

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <div>
      <h1>Display</h1>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

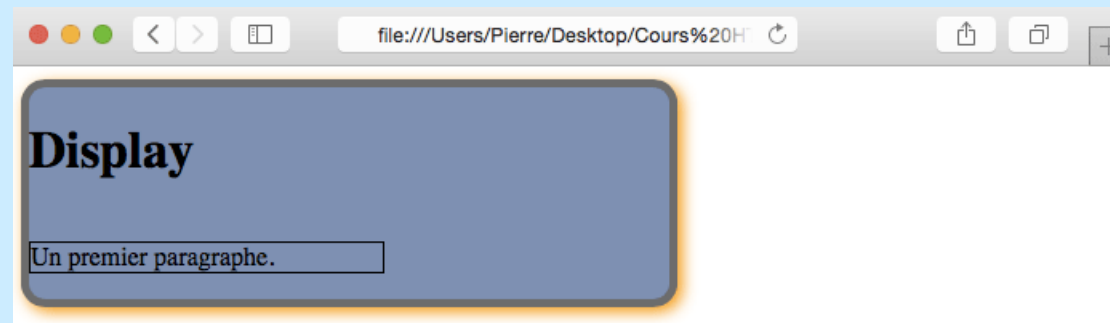
Display : none

- (Suite)

```
div{
  background-color: #89B;
  width: 400px;
  border: 5px solid #777;
  border-radius: 15px;
  box-shadow: 3px 3px 9px orange;
}

.para1{
  display: inline-block;
  border: 1px solid black;
  width: 55%;
}

.para2{
  display: none;
}
```



- Pour tester le code : <http://jsbin.com/gopega/78/edit?html,css,output>

HTML Tutorial

LE POSITIONNEMENT EN CSS

La propriété CSS position

- La propriété CSS position permet de positionner les différents éléments HTML de façon absolue ou relativement par rapport à d'autres éléments HTML ou par rapport à leur place d'origine entre autres.
- La propriété CSS position supporte quatre valeurs différentes :
 - static ;
 - relative ;
 - fixed ;
 - absolute.
- On utilise la propriété position conjointement aux propriétés top, right, bottom et left afin de positionner précisément les éléments.

Position : static

- La valeur static correspond au positionnement par défaut des éléments HTML dans une page.
- Tout élément positionné de façon « static » ne pourra pas être affecté par les propriétés top, right, bottom et left.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <div>
      <h1>Position</h1>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

Position : static

- (Suite)



- Pour tester le code : <http://jsbin.com/gopega/79/edit?html,css,output>

Position : relative

- Un élément positionné grâce à `position:relative` va être repositionné relativement par rapport à sa position par défaut.
- Par exemple, si on positionne un élément HTML de façon relative et qu'on lui ajoute `left:50px`, l'élément sera déplacé de 50 pixels vers la droite par rapport à sa position par défaut.



- Pour tester le code : <http://jsbin.com/gopega/80/edit?html,css,output>

Position : fixed

- Un élément HTML possédant un positionnement « fixed » va toujours rester à la même place, même si le visiteur descend ("scroll") dans la page.
- Cette valeur permet de conserver un élément constamment visible, comme un menu ou un sommaire par exemple.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <div>
      <h1>Position</h1>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
      <p>B<br><br><br><br><br>l<br><br><br><br><br>a<br><br><br><br><br></p>
      <p>B<br><br><br><br><br>l<br><br><br><br><br>a<br><br><br><br><br></p>
    </div>
  </body>
</html>
```

Position : fixed

- (Suite)



- Pour tester le code : <http://jsbin.com/gopega/81/edit?html,css,output>

Position : absolute

- La valeur absolue de la propriété position va permettre de positionner un élément de façon relative par rapport à son parent le plus proche auquel on a appliqué un positionnement spécifique (relative, fixed ou absolute).
- Attention : si l'élément auquel on applique position:absolute ne possède pas d'élément parent positionné spécifiquement, celui-ci va se positionner par rapport à la page entière.
- Attention au comportement de la page avec cette option, car la valeur absolue sort totalement un élément HTML du flux normal de la page web.

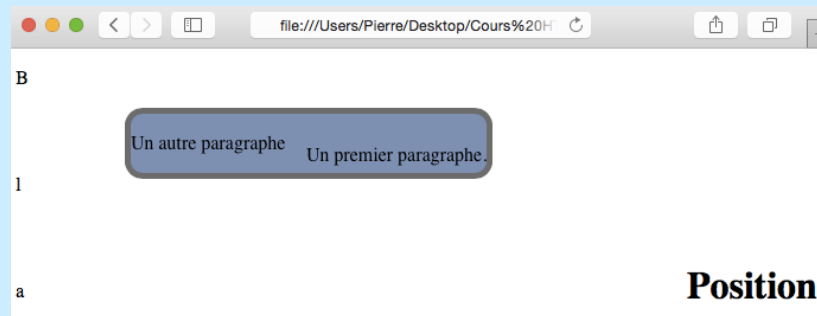
```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Position</h1>
    <!-- Attention, contrairement aux exemples précédents, h1 n'est
    plus à l'intérieur de notre élément div ! -->
    <div>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
    <p>B<br><br><br><br><br>l<br><br><br><br><br>a<br><br><br><br><br></p>
    <p>B<br><br><br><br><br>l<br><br><br><br><br>a<br><br><br><br><br></p>
  </body>
</html>
```

Position : absolute

- (Suite)

```
h1{  
  position: absolute;  
  bottom: 0px;  
  right: 0px;  
}  
  
div{  
  background-color: #89B;  
  width: 300px;  
  border: 5px solid #777;  
  border-radius: 15px;  
  position: fixed;  
  top: 50px;  
  left: 100px;  
}  
  
.para1{  
  position: absolute;  
  right: 0px;  
  top: 10px;  
}
```



- Pour tester le code : <http://jsbin.com/gopega/82/edit?html,css,output>

Le z-index

- La propriété position permet de modifier et de casser le flux normal des éléments HTML dans la page en les positionnant où l'on souhaite dans la page.
- Parfois, en modifiant la position des éléments ou pas, certains vont se chevaucher. Par défaut, l'élément positionné en dernier apparaîtra par dessus les autres. Cependant, on peut choisir quel élément va apparaître au dessus de quel autre grâce à la propriété z-index.
- La propriété z-index va prendre un nombre en valeur : un nombre plus grand positionnera un élément devant un autre ayant un z-index plus petit.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Position et z-index</h1>

    <div>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

Le z-index

- (Suite)

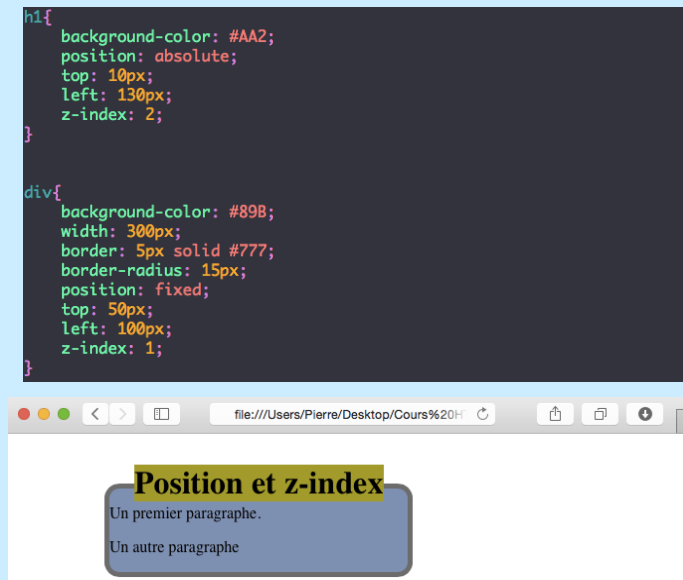
```
h1{
  background-color: #AA2;
  position: absolute;
  top: 10px;
  left: 130px;
}

div{
  background-color: #89B;
  width: 300px;
  border: 5px solid #777;
  border-radius: 15px;
  position: fixed;
  top: 50px;
  left: 100px;
}
```



Le z-index

- Ci-dessus, on s'est arrangé pour que le titre h1 et l'élément div se chevauchent. Sans z-index, c'est le dernier élément positionné qui sera « par dessus » l'autre. C'est donc bien le div qui est devant h1.
- On peut inverser l'ordre de ces éléments en indiquant un z-index avec une valeur plus grande pour h1 que pour div comme ceci :



- Pour tester le code : <http://jsbin.com/gopega/83/edit?html,css,output>

Le z-index

- La propriété z-index sera particulièrement utile dans le cas d'un menu déroulant (afin que les rubriques s'affichent devant le reste de la page) ou pour l'affichage de produits dans le cas d'un site e-commerce complexe par exemple dans lequel de nombreux div et autres éléments HTML risquent de se chevaucher.

HTML Tutorial

FLOAT ET CLEAR

La propriété float

La propriété **float** permet de faire « flotter » des éléments HTML à gauche ou à droite dans une page web.

- On utilisera généralement soit la valeur `right` (l'élément ciblé flotte à droite), soit `left` (l'élément ciblé flotte à gauche) avec cette propriété.
- La propriété `float` va faire sortir un élément du flux normal de la page.

Il y a trois choses à retenir lorsque l'on utilise cette propriété :

- Les éléments suivants un élément flottant vont se positionner à côté de celui-ci par défaut. On peut annuler ce comportement grâce à la propriété `clear` ;
- Les éléments positionnés de façon absolue avec `position:absolute` ignoreront la propriété `float` (ils ne pourront pas flotter) ;
- Un élément flottera toujours dans les limites (en terme de largeur) de son élément parent conteneur.

La propriété float

- (Suite)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Float et clear</h1>

    <div>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

La propriété float

- (Suite)

```
h1{
  background-color: #AA2;
  width: 300px;
  float: left;
}

div{
  background-color: #89B;
  border: 5px solid #777;
  width: 300px;
  float: right;
}

.para1{
  float:right;
}
```



- Pour tester le code : <http://jsbin.com/gopega/86/edit?html,css,output>

La propriété float

- Dans cet exemple, on fait flotter l'élément HTML h1 sur la gauche de la page. Ainsi, s'ils ont la place nécessaire, les éléments suivants l'élément h1 viendront se positionner par défaut à côté de celui-ci.
- On fait ensuite flotter l'élément div sur la droite. Celui-ci se positionne donc bien par défaut sur la « même ligne » (imaginaire) que h1 mais à droite de la page.
- Finalement, on fait flotter le premier paragraphe à droite dans le div. Cela a pour effet de placer le deuxième paragraphe sur la même ligne que le premier... et sur sa gauche !

Arrêter le flottement avec clear

La propriété clear permet de contrôler le comportement d'éléments flottants.

- Plus précisément, cette propriété permet d'empêcher des éléments de venir se positionner aux côtés d'éléments flottants.

Cette propriété peut prendre les valeurs suivantes :

- Left : neutralise l'effet d'un float:left ;
- Right : neutralise l'effet d'un float:right ;
- Both : neutralise l'effet d'un float:left et d'un float:right ;
- Il faut bien faire attention à appliquer le clear à l'élément qui va venir flotter par défaut autour de l'élément flottant, pas à celui qui possède la propriété float

Arrêter le flottement avec clear

- (Suite)

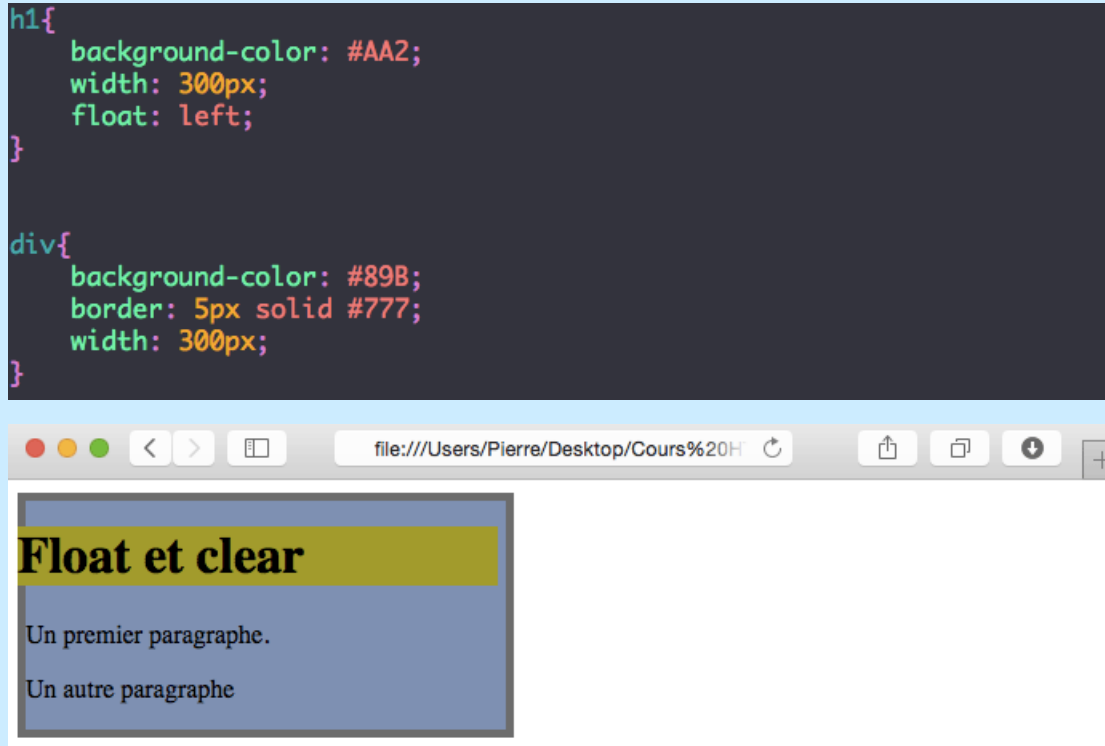
```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Float et clear</h1>

    <div>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

Arrêter le flottement avec clear

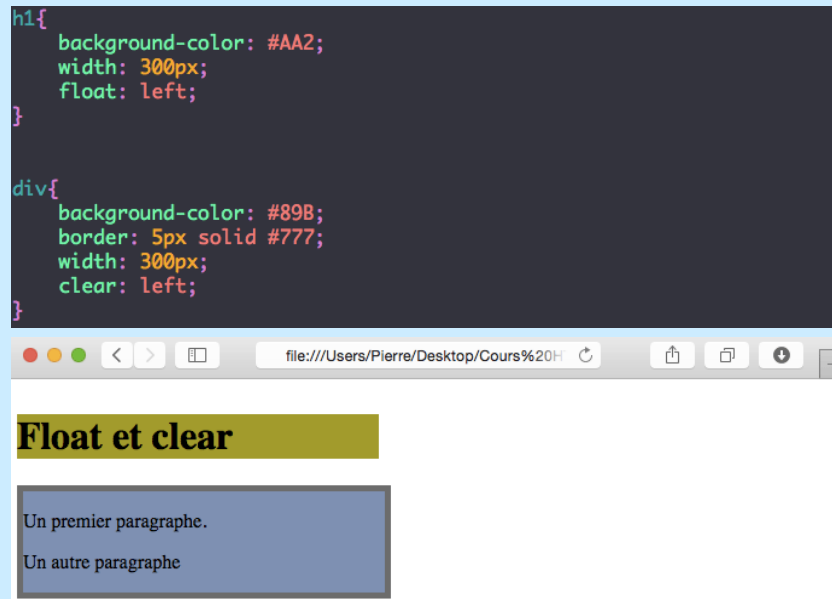
- (Suite)



- Pour tester le code : <http://jsbin.com/gopega/87/edit?html,css,output>

Arrêter le flottement avec clear

- Dans l'exemple ci-dessus, nous avons fait flotter l'élément h1 sur la gauche. Par défaut, le div en dessous du h1 dans notre code HTML va venir se placer sur la même ligne que notre titre et donc se placer autour de notre h1.
- Pour éviter ce comportement, on va spécifier un `clear:left` à l'élément div, afin de l'empêcher de venir flotter autour de notre h1.



- Pour tester le code : <http://jsbin.com/sipujup/23/edit?html,css,output>

Gestion du débordement (l'overflow)

Il faut faire bien attention en utilisant la propriété float car celle-ci sort complètement un élément HTML du flux normal de notre page web.

- Ainsi, si un élément flottant est plus grand que son élément parent conteneur, celui-ci va déborder de son conteneur verticalement.
- Afin d'éviter ce comportement, on peut utiliser la propriété overflow qui permet de cacher (ou pas) ce qui va dépasser.

Cette propriété peut prendre les valeurs suivantes :

- Visible : valeur par défaut (rien ne sera coupé) ;
- Hidden : ce qui dépasse sera coupé ;
- Scroll : coupe de qui dépasse et ajoute une barre de défilement afin d'avoir accès à tout le contenu ;
- Auto : Laisse le navigateur décider du meilleur choix ;
- Initial et inherit.

Gestion du débordement (l'overflow)

Soit quelques exemples utilisant ces différentes valeurs.

- Commençons déjà par constater le problème avec un overflow:visible par défaut.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Float et clear</h1>

    <div>
      <p class="para1">Un premier paragraphe, <br><br>
      qui va dépasser de notre div.</p>
      <p class="para2">Un autre paragraphe, <br><br>
      qui va également dépasser.</p>
    </div>
  </body>
</html>
```

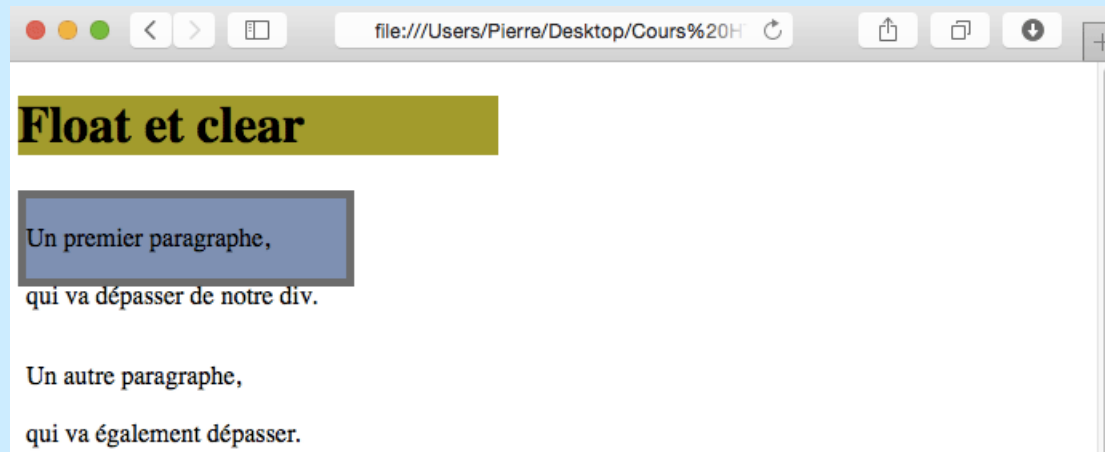
Gestion du débordement (l'overflow)

- (Suite)

```
h1{
  background-color: #AA2;
  width: 300px;
}

div{
  background-color: #89B;
  border: 5px solid #777;
  width: 200px;
  height: 50px;
}

.para1, .para2{
  float: left;
}
```



Gestion du débordement (l'overflow)

- Observons maintenant la différence avec un `overflow:hidden`. Tout ce qui dépasse va être coupé et caché.



- Pour tester le code : <http://jsbin.com/gopega/89/edit?html,css,output>

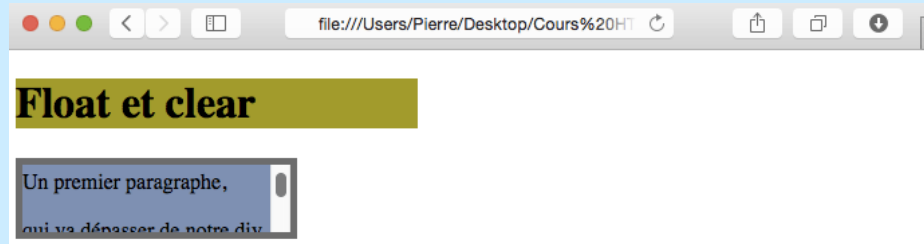
Gestion du débordement (l'overflow)

- Finalement, regardons à nouveau le comportement avec un `overflow:auto` (qui correspond généralement, en cas de dépassement, à `overflow:scroll` pour des navigateurs modernes).

```
h1{
  background-color: #AA2;
  width: 300px;
}

div{
  background-color: #89B;
  border: 5px solid #777;
  width: 200px;
  height: 50px;
  overflow: auto;
}

.para1, .para2{
  float: left;
}
```



- Pour tester le code : <http://jsbin.com/gopega/90/edit?html,css,output>

Gestion du débordement (l'overflow)

- Comme on peut le voir, dans ce dernier exemple, une barre de défilement est apparue permet d'avoir accès à tout le contenu caché.

HTML Tutorial

ALIGNEMENT EN CSS

Aligner des éléments HTML grâce au CSS

Ici, on expose trois manières couramment utilisées pour **aligner ou centrer des éléments HTML dans une page** :

- En utilisant la propriété `margin` ;
- En utilisant la propriété `position` ;
- En utilisant la propriété `float`.

Centrer horizontalement avec margin

- Dans l'exemple, on indique deux valeurs à la propriété margin : une valeur pour les marges haute et basse (au choix) et la valeur auto pour les marges gauche et droite, ce qui va avoir pour effet de centrer l'élément HTML.
- Attention cependant : se rappeler que l'élément à centrer doit être de type block et posséder une largeur définie pour pouvoir être centré correctement.

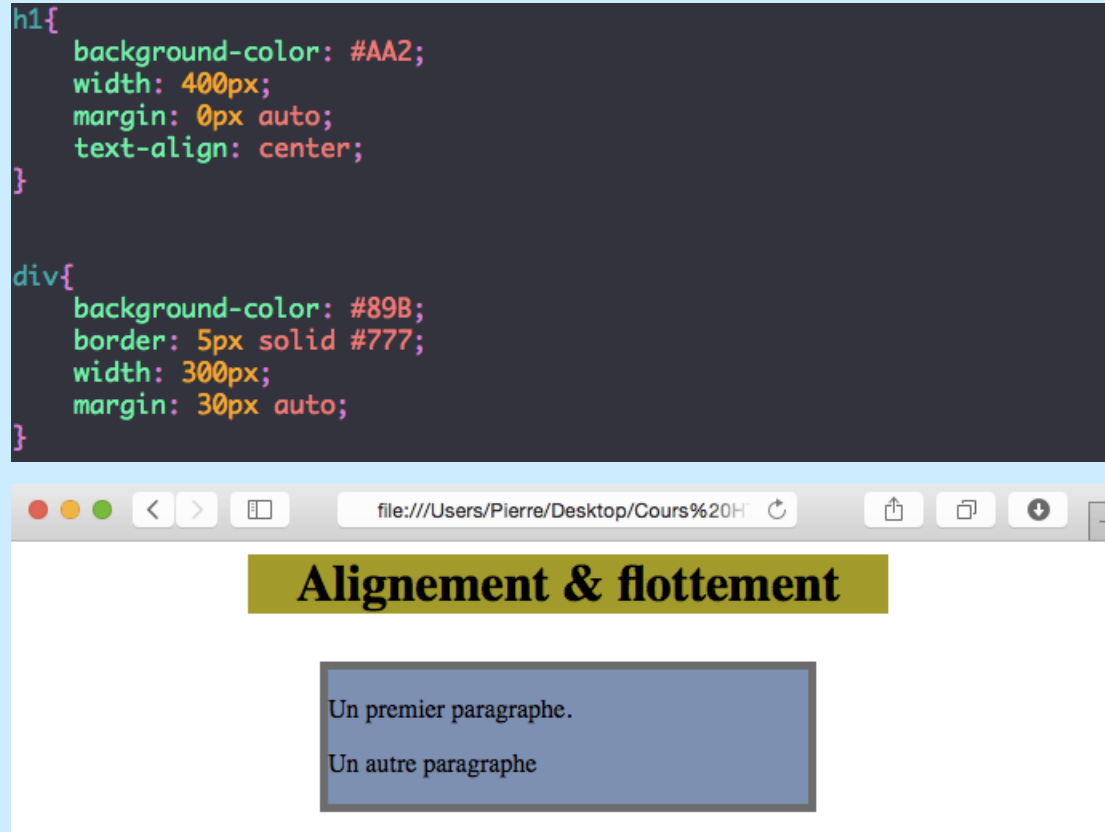
```
<!DOCTYPE html>
<html>
  <head>
    <title>Le modèle des boîtes</title>
    <meta charset= "utf-8">
    <link rel="stylesheet" href="styles.css">
  </head>

  <body>
    <h1>Alignement & flottage</h1>

    <div>
      <p class="para1">Un premier paragraphe.</p>
      <p class="para2">Un autre paragraphe</p>
    </div>
  </body>
</html>
```

Centrer horizontalement avec margin

- (Suite)



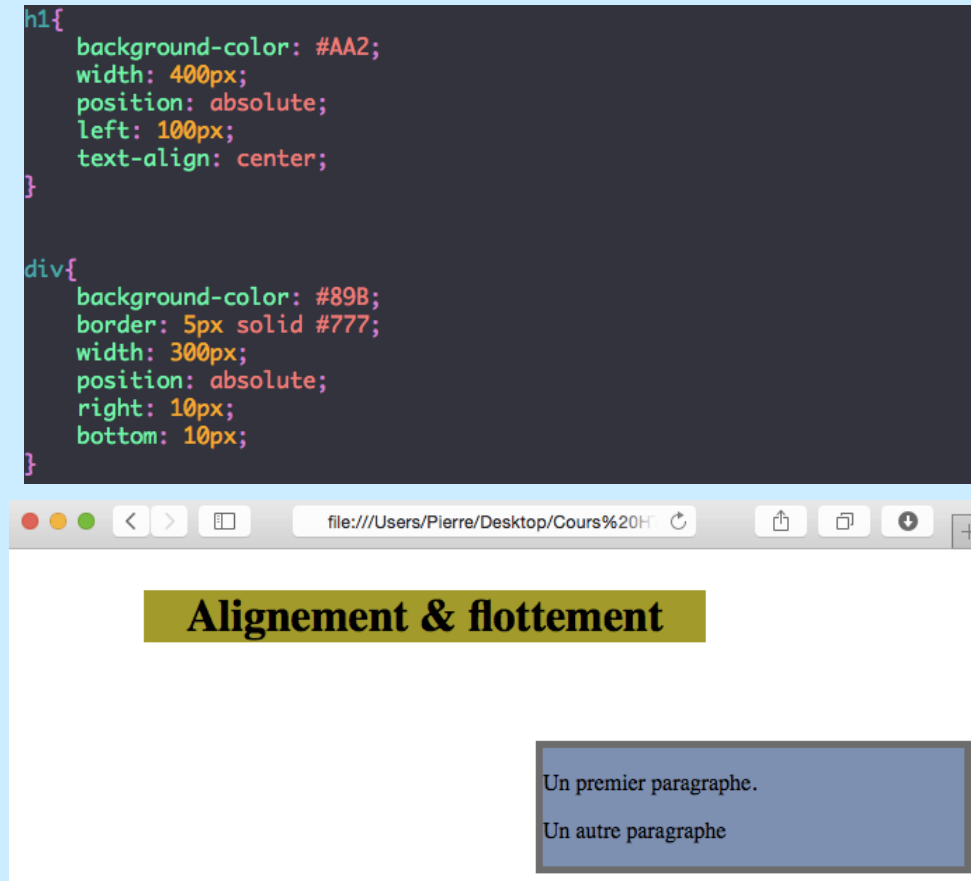
- Pour tester le code : <http://jsbin.com/sipujup/24/edit?html,css,output>

Aligner horizontalement un élément HTML grâce à la propriété CSS position

- On peut également aligner horizontalement un élément HTML grâce à la propriété position utilisée conjointement avec les propriétés left et right. Pour cela, on utilise la valeur absolute de la propriété position
- Lorsque l'on centre des éléments HTML de cette façon, il est bon de faire un « reset CSS », sur la page, c'est-à-dire redéfinir des marges intérieures et extérieures sur l'élément body égales à 0px afin d'éviter des différences d'affichage entre navigateurs.

Aligner horizontalement un élément HTML grâce à la propriété CSS position

- (Suite)



- Pour tester le code :

Aligner horizontalement un élément HTML grâce à la propriété CSS position

- Petite astuce : on peut également centrer horizontalement un élément en utilisant la propriété position , en attribuant la même valeur, en pourcentage, aux propriétés left et right.
- Cependant, afin que cela fonctionne parfaitement, il ne faut pas donner de largeur définie à l'élément HTML ciblé (celui-ci va se recadrer de lui-même selon la valeur attribuée à left et à right et la place disponible).

Aligner horizontalement un élément HTML grâce à la propriété CSS position

- (Suite)

```
h1{
  background-color: #AA2;
  position: absolute;
  left: 20%;
  right: 20%;
  text-align: center;
}

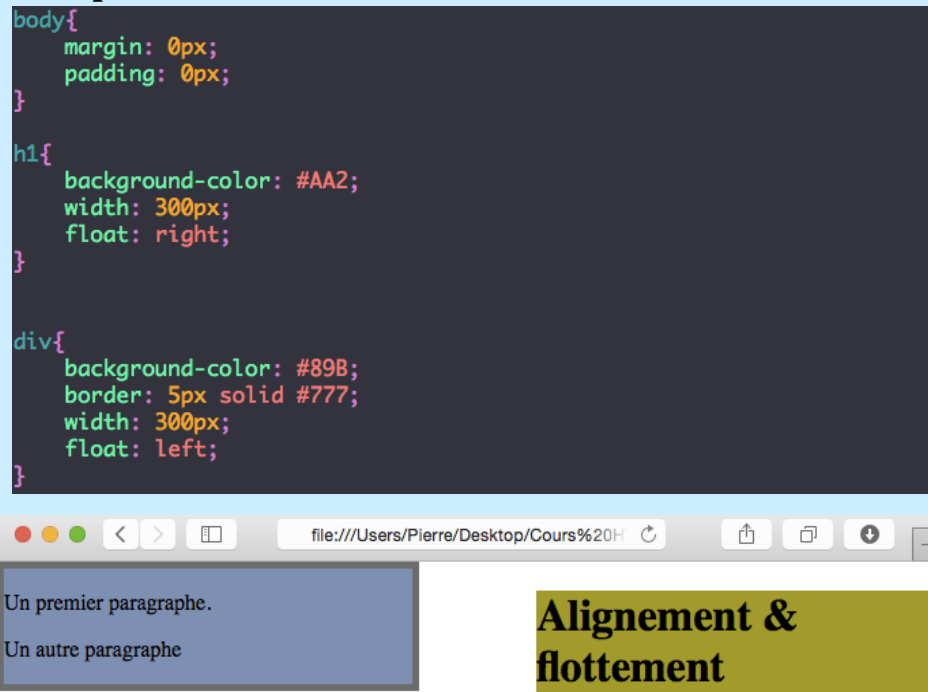
div{
  background-color: #89B;
  border: 5px solid #777;
  width: 300px;
  position: absolute;
  right: 10px;
  bottom: 10px;
}
```



- Pour tester le code : <http://jsbin.com/sipujup/25/edit?html,css,output>

Aligner horizontalement en utilisant float

- On peut également très simplement aligner un élément sur la gauche ou la droite de la page en utilisant la propriété float et ses valeurs left et right.
- Comme pour la propriété position, la propriété float perturbe le flux normal de la page. Le conseil précédent sur le reset CSS est donc tout aussi valable.



- Pour tester le code : <http://jsbin.com/sipujup/26/edit?html,css,output>

-
- (Suite)