

Coins arrondis CSS

par Pascale Lambert ([mammouthland](#)) ([Blog](#))

Date de publication : 16/09/2010

Dernière mise à jour : 24/11/2010

Pour réaliser des blocs (menus ou autres) avec des coins arrondis, il existe principalement trois méthodes en CSS, qui offrent plus ou moins de fluidité et de possibilités graphiques.

Créer des coins arrondis avec une seule image, coulissante en hauteur et en largeur, permet une certaine liberté graphique et s'adapte à un design fluide.

N'hésitez pas à nous faire part de votre avis sur cet article :

Préambule.....	3
Image de fond.....	3
Code (X)HTML.....	3
Code CSS.....	4
Lectures complémentaires.....	5

Préambule

Les trois méthodes courantes pour créer des coins arrondis sont :

- 1 La propriété CSS 3 border-radius (implémentation très attendue !)
- 2 La technique positionnant 4 images de coins à l'aide de 4 div imbriqués ;
- 3 La technique de l'image coulissante en hauteur.

Chacune de ces méthodes a ses avantages et ses inconvénients.

Cette 4ème méthode, utilisant une seule image adaptable en hauteur et en largeur pour un design fluide, s'inspire des deux dernières pré-citées. Elle souffre aussi de quelques défauts, notamment l'éternelle imbrication de plusieurs div. Cependant, elle a l'avantage d'offrir plus de possibilités graphiques (et donc pour faire autre chose que des coins arrondis) que la seconde technique. Attention cependant à ce que cette image de fond ne soit pas trop lourde !

Fonctionne avec :

- Tous les navigateurs graphiques.

Attributs utilisés :

- background ;
- margin ;
- max-width (inopérant avec IE6) ;
- padding.

Image de fond

Le but est donc de mettre en fond une seule image, très grande, qui pourra s'adapter aux résolutions les plus courantes. La taille des écrans s'agrandissant, j'ai tablé sur une largeur d'environ 2000 pixels, la hauteur en faisant autant. Bien entendu, si vous pensez que vos pages pourront dépasser 2000 pixels de hauteur, il faudra la rallonger d'autant !

L'image a un poids d'un peu moins de 17 ko, elle est donc très légère et se chargera très vite.

Code (X)HTML

Il faut donc 4 div (eh oui...) pour permettre de positionner correctement l'image autour du texte.

```
<div id="gauche">
  <div id="droite">
    <div id="haut">
      <div></div>
    </div><!-- /haut -->
    <h2>Lorem ipsum</h2>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit.
      Mauris vulputate laoreet urna. Integer magna.
      Donec facilisis lectus sed quam.
      Curabitur sit amet lacus id lacus facilisis venenatis.
    </p>
  </div><!-- /droite -->
</div><!-- /gauche -->
```

Code CSS

On commence par positionner le coin bas gauche de l'image de fond à l'aide de la propriété background. Le `margin:auto` est facultatif, c'est juste pour centrer le bloc dans la page. Par contre, la précaution du `max-width`, correspondant à la largeur réelle de l'image, est indispensable pour les résolutions d'écran supérieures à celle-ci : elles sont encore rares, mais suffisantes pour en tenir compte.

Pour rappel, le code :

```
background-color:#E4EFFF;
background-image:url(images/fond-arrondi.png);
background-repeat:no-repeat;
background-position:bottom left;
```

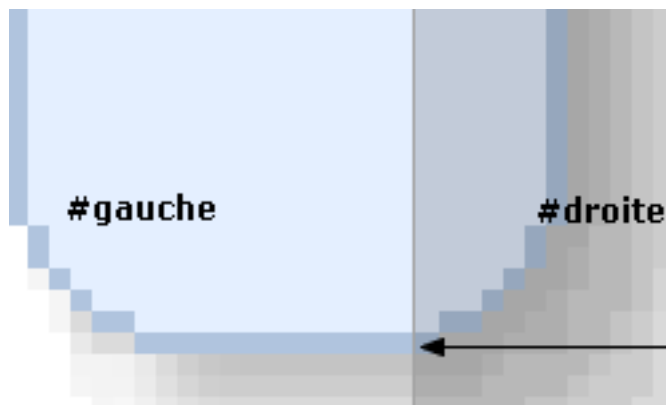
peut s'optimiser par :

```
background:#E4EFFF url(images/fond-arrondi.png) no-repeat bottom left;
```

Cela donne donc :

```
#gauche {
background:#E4EFFF url(images/fond-arrondi.png) no-repeat bottom left;
margin:auto;
max-width:2007px;
}
```

On positionne ensuite le bas de l'image à droite (`bottom right`). C'est donc le premier "coulissement" de l'image en faisant glisser le côté droit sur le côté gauche. Les 7 pixels déclarés en marge de gauche (`margin-left`) correspondent à la largeur de l'arrondi qui ne doit pas être recouvert.




```
#droite {
background:#E4EFFF url(images/fond-arrondi.png) no-repeat bottom right;
margin-left:7px;
padding-bottom:20px;
}
```

Vient ensuite le tour du positionnement du haut de l'image avec l'élément supérieur du bloc délimités par le `<div id="haut">` et l'autre div imbriqué. C'est le second coulissement, cette fois en hauteur. Les tailles de 7 pixels correspondent toujours aux tailles des coins.

```
#haut {  
  background:#E4EFFF url(images/fond-arrondi.png) no-repeat top right;  
  margin-left:-7px;  
  padding:0;  
}  
#haut div {  
  background:#E4EFFF url(images/fond-arrondi.png) no-repeat top left;  
  height:7px;  
  width:7px;  
}
```

Ne reste plus qu'à définir les marges désirées pour que le texte ne colle pas aux bords de l'image sans interférer avec les positionnements précédents.

```
p, h2 {  
  padding:5px 10px;  
  margin:0;  
}
```

 [Voir le résultat](#)

Lectures complémentaires

- [Coins arrondis en CSS de Maurice Svay](#) ;
- [Diverses méthodes pour créer des coins arrondis \(Biboo\)](#) ;
- [Coins arrondis pour designs fluides](#) ;
- [CSS3 info : la propriété border-radius \(en\)](#) ;
- [Générateur de coins arrondis \(en\)](#).