

# Les ombrages en CSS3

Grâce à des propriétés telles que `box-shadow`, `text-shadow` et `border-colors`, les feuilles de style offrent la possibilité de créer des ombrages sur les différents éléments de votre page sans nécessiter d'image décorative et de découpes intempestives. Tout n'est pas rose puisque certains navigateurs à la traîne ne reconnaissent pas encore ces différentes propriétés et nécessitent des adaptations personnalisées. Faisons le tour du propriétaire...

## Box-shadow

La propriété CSS `box-shadow` a été incluse dans le module "borders" de CSS3 » <http://www.w3.org/TR/2002/WD-css3-border-20021107/#the-box-shadow> et permet de générer une ombre portée sur n'importe quel élément HTML.

Parmi les différentes valeurs utilisables, il est possible d'indiquer le décalage vertical et horizontal ainsi que la force du dégradé. Sans oublier la couleur, bien-entendu. La propriété s'applique sur la boîte de l'élément, et non sur sa bordure. **L'ombrage n'affecte pas la taille de la boîte de l'élément.**

En voici une illustration ([Voir le résultat sur votre navigateur » /xmedia/tuto/exemples/ombrage/box-shadow.html](#)) :



```
img {  
  box-shadow: 8px 8px 0px #aaa;  
}
```

## Syntaxe

- La première valeur indique le décalage horizontal vers la droite (ici 8px)
- le deuxième correspond au décalage vertical vers le bas (ici 8px)
- le chiffre suivant indique la force du dégradé (ici 0px)
- et enfin, la couleur (ici #aaa)

### Compatibilité et utilisation concrète

Seuls les navigateurs modernes, reconnaissent la propriété `box-shadow`. Il s'agit d'Internet Explorer 9+, Firefox 4+, Chrome 10+, Safari 5.1+, Opera 10.5+, Opera Mobile 11+.

La propriété doit être préfixée sur des versions antérieures par `-moz-` pour Firefox <4, ou `-webkit-` pour Chrome <10, Safari <5.1 et Android ainsi que Safari Mobile <5.0, pour fonctionner sur les premières versions de ces navigateurs à avoir accepté les ombrages, ce qui devient donc `-moz-box-shadow` et `-webkit-box-shadow`.

**Par souci de clarté, l'article se passe de ces préfixes.** Les exemples construits utilisent naturellement ces préfixes pour fonctionner et anticiper une implémentation complète sur tous les navigateurs dans le futur.

Voici une illustration d'ombre portée dégradée ([Voir le résultat sur votre navigateur » /xmedia/tuto/exemples/ombrage/box-shadow.html](#)) :



```
img {
  box-shadow: 8px 8px 12px #aaa;
}
```

Et voici enfin une variante sous forme de "halo" dégradé ([Voir le résultat sur votre navigateur » /xmedia/tuto/exemples/ombrage/box-shadow.html](#)) :



```
img {
  box-shadow: 1px 1px 12px #555;
}
```

## Adaptation pour Internet Explorer <9

Le filtre **Shadow** » <http://msdn.microsoft.com/en-us/library/ms533086%28VS.85%29.aspx>

⚠ Internet Explorer, jusqu'à la version IE8 incluse, ne reconnaît pas la propriété **box-shadow**.

Cependant, tout comme [pour la transparence](/tuto/lire/909-CSS-transparence-couleur-rgba.html) » </tuto/lire/909-CSS-transparence-couleur-rgba.html> de couleur, il existe un filtre propriétaire Microsoft permettant d'obtenir un résultat similaire. Il s'agit du filtre **Shadow** » <http://msdn.microsoft.com/en-us/library/ms533086%28VS.85%29.aspx> (cousin du filtre **DropShadow**).

Il suffit d'appliquer ce filtre à l'élément en indiquant une couleur, une direction (en degrés) et l'intensité du dégradé.

Voici ce que donne une ombre portée "classique" à l'aide du filtre **Shadow** sur Internet Explorer ([Voir le résultat sur votre navigateur](#) » </xmedia/tuto/exemples/ombrage/box-shadow-hack.html>) :



```
img {
  filter:progid:DXImageTransform.Microsoft.Shadow(color='#aaaaaa', Direction=135,
  Strength=12);
  zoom: 1;
}
```

Pour éviter de gêner les bons élèves, nous pouvons appliquer cette bidouille uniquement sur Internet Explorer à l'aide d'un [commentaire conditionnel](/astuce/lire/48-commentaires-conditionnels.html) » </astuce/lire/48-commentaires-conditionnels.html> .

Au final, les styles CSS devraient ressembler à ceci :

```
<style type="text/css">
.ombrage {
  border: 5px solid #fff;
  -moz-box-shadow: 8px 8px 12px #aaa;
  -webkit-box-shadow: 8px 8px 12px #aaa;
  box-shadow: 8px 8px 12px #555;
}
</style>
```

```
<!--[if lte IE 8]>
<style type="text/css">
.ombrage {
    filter:progid:DXImageTransform.Microsoft.Shadow(color='#aaaaaa', Direction=135,
Strength=12);
    zoom: 1;
}
</style>
<![endif]-->
```

Le filtre **Glow** » <http://msdn.microsoft.com/en-us/library/ms532995%28VS.85%29.aspx>

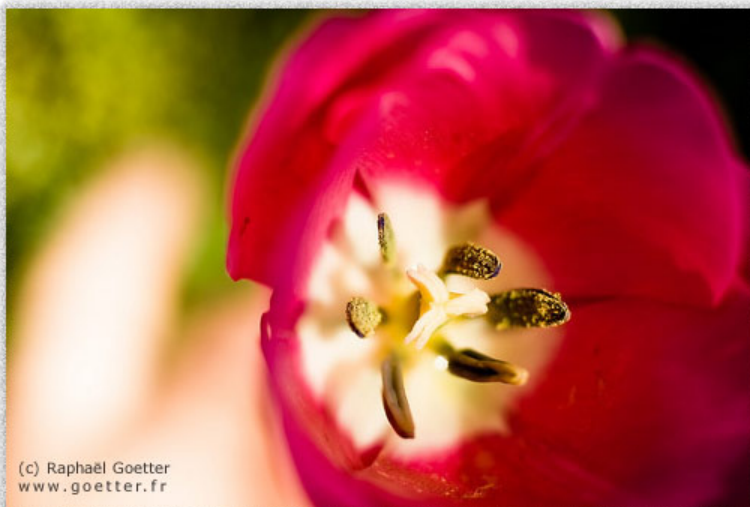
La syntaxe :

```
filter:progid:DXImageTransform.Microsoft.Glow(Color=#c0c0c0,Strength=8);
```

L'attribut **Color** désigne la couleur du halo.

L'attribut **Strength** désigne la taille de l'effet.

Voici une illustration : ([Voir le résultat sur votre navigateur » /xmedia/tuto/exemples/ombrage/glow-ie.html](#) )



Le rendu est loin d'être parfait... mais dans certains (rares?) cas, il peut suffire.

Code utilisé :

```
<!--[if lte IE 8]>
<style type="text/css">
.ombrage {
    filter:progid:DXImageTransform.Microsoft.Glow(Color=#c0c0c0,Strength=8);
    zoom: 1;
}
</style>
<![endif]-->
```

## Outil WYSIWYG

Pour vous familiariser avec le concept de **box-shadow** et pouvoir jouer en direct avec cette propriété et ses effets, sachez qu'il existe [un outil WYSIWYG publié chez Westciv.com](http://www.westciv.com/tools/boxshadows/index.html) » <http://www.westciv.com/tools/boxshadows/index.html>

# Text-shadow

`text-shadow` » <http://www.yoyodesign.org/doc/w3c/css2/text.html#text-shadow-props> est une propriété permettant de **produire une ombre portée sur le texte de contenu sur lequel elle est appliquée**.

Il est possible de spécifier les décalages de l'ombrage, la couleur et sa zone de flou. Ces effets s'appliquent dans l'ordre spécifié et peuvent ainsi se recouvrir, mais ceux-ci ne recouvriront jamais le texte lui-même. **L'ombrage n'affecte pas la taille de la boîte de texte.**

Voici une illustration de cette propriété `text-shadow` en action ([Voir le résultat sur votre navigateur](#) »

</xmedia/tuto/exemples/ombrage/text-shadow.html>) :

Ombrage sur le texte avec CSS3  
**Et encore plus fort !**

(Inspiré de [CSS3.info](#))

```
h1 {  
  text-shadow: 0px 0px 9px #777;  
  color: #fff;  
}
```

## Compatibilité et syntaxe

⚠ A l'instar de `box-shadow`, cette propriété n'est pas reconnue par Internet Explorer (version 8 comprise). Par contre, Opera 9.5+ l'implémente correctement.

Notez également que puisqu'il s'agit d'une propriété finalisée, il n'y a pas lieu de lui affecter un préfixe (-moz-, -webkit-, ...). La propriété demeure `text-shadow` quel que soit le navigateur.

## Adaptation pour Internet Explorer

Là encore, nous allons devoir ruser en appliquant le filtre `Shadow` » <http://msdn.microsoft.com/en-us/library/ms533086%28VS.85%29.aspx> à nos éléments, de la même manière que nous l'avons fait précédemment pour adapter la propriété `box-shadow`.

Le résultat sur Internet Explorer est malheureusement moins excitant que sur les autres navigateurs ([Voir le résultat sur votre navigateur](#) » </xmedia/tuto/exemples/ombrage/text-shadow-hack.html>) :

# Ombrage sur le texte avec

Au final, les styles CSS devraient ressembler à ceci :

```
.ombrage {  
  text-shadow: 2px 2px 4px #999;  
}  
</style>  
<!--[if lte IE 8]>  
<style type="text/css">  
.ombrage {
```

```
filter:progid:DXImageTransform.Microsoft.Shadow(color='#999999', Direction=135,
Strength=4);
zoom: 1;
}
</style>
<![endif]-->
```

## Outil WYSIWYG

Pour vous familiariser avec le concept de `text-shadow` il existe aussi [un outil WYSIWYG publié chez Westciv.com](http://www.westciv.com/tools/shadows/index.html) »  
<http://www.westciv.com/tools/shadows/index.html>

## Border-colors

`border-colors` est une propriété uniquement reconnue par Mozilla Firefox 3.0+ et qui permet de déclarer plusieurs valeurs de couleurs à appliquer sur la bordure d'un élément.

En voici une illustration ([Voir le résultat sur votre navigateur » /xmedia/tuto/exemples/ombrage/border-gradient.html](#)) :



**Bords dégradés en  
CSS**

```
.gradient {
  -moz-border-bottom-colors: #897048 #917953 #a18a66 #b6a488 #c5b59b #d4c5ae #e2d6c4
#eae1d2;
  -moz-border-top-colors: #897048 #917953 #a18a66 #b6a488 #c5b59b #d4c5ae #e2d6c4
#eae1d2;
  -moz-border-left-colors: #897048 #917953 #a18a66 #b6a488 #c5b59b #d4c5ae #e2d6c4
#eae1d2;
  -moz-border-right-colors: #897048 #917953 #a18a66 #b6a488 #c5b59b #d4c5ae #e2d6c4
#eae1d2;
}
```

Puisque cette propriété n'est pas vraiment répandue, nous n'allons pas entrer dans les détails, mais sachez que vous pouvez l'étudier sur [le site ZenElements.com](http://www.zenelements.com/blog/css3-border-color-colour/) » <http://www.zenelements.com/blog/css3-border-color-colour/> .

## Conclusion

Associées aux [coins arrondis](#) » </tuto/lire/891-coins-arrondis-css-sans-images.html> et aux [transparences de couleurs](#) » </tuto/lire/909-CSS-transparence-couleur-rgba.html> , les propriétés d'ombrage sont très puissantes pour une mise en forme avancée, sans utiliser d'images.

En production, il est nécessaire de prendre quelques précautions d'usage et de se poser des questions préalables : puis-je m'offrir une version dégradée (*graceful degradation*) sur les navigateurs à la traîne, voire tout simplement les ignorer ? Quel est mon public et ses attentes ? Combien de temps puis-je passer à bidouiller des équivalences sur IE ?... La réponse est uniquement entre vos mains.