



MAY **CSS3** BE
WITH YOU

CSS3 LES EFFETS AVANCÉS



MAY CSS3 BE WITH YOU

- #Les lecteurs pour afficher les effets CSS3 P.3
- #La Transformation 2D avec CSS3 P.4
- #La Transformation 3D avec CSS3 P.5
- #Les effets de transition avec CSS3 P. 6
- #Transformation 3D et transition CSS3 P. 7
- #Les animations en CSS3 P. 8
- #Les propriétés des animations CSS3 P.9
- #Les dégradés de couleurs en CSS3 P.10



LES LECTEURS POUR AFFICHER LES EFFETS CSS3

Les nouvelles propriétés CSS3 ont besoin de faire appel à des lecteurs pour s'afficher correctement selon le navigateur qui les lit, il faut se référer au site w3schools.com pour savoir quel(s) constructeur(s) utiliser selon la propriété CSS3. Néanmoins, certains navigateurs affichent parfaitement le CSS3 et c'est pourquoi, en plus des lecteurs, il faut toujours également faire appel à la propriété standard.

Le sélecteur

Les propriété peuvent être appliquées à tous types de selecteurs.

```
div {  
  -ms-propriété: méthode(valeur); /* IE 9 */  
  -webkit-propriété: méthode(valeur); /* Chrome, Safari */  
  -o-propriété: méthode(valeur); /* Opera */  
  -moz-propriété: méthode(valeur); /* Firefox */  
  propriété: méthode(valeur); /* Standard */  
}
```

Les constructeurs

Selon la propriété utilisée, il faut appeler les bons constructeurs

Méthodes/Valeurs

Chaque propriété doit respecter une orthographe stricte.



LA TRANSFORMATION 2D AVEC CSS3

La propriété transform en CSS3 permet de modifier l'aspect d'un élément du DOM sur deux dimensions. Cette propriété dispose de six méthodes différentes : translate(), rotate(), scale(), skewX(), skewY() et matrix().

translate()

Déplace un élément de sa position actuel selon les axes X et Y.

```
#divUn {  
  -ms-transform: translate(50px,100px); /* IE 9 */  
  -webkit-transform: translate(50px,100px); /* Safari */  
  transform: translate(50px,100px);  
}
```

rotate()

Fait tourner un élément dans le sens horraire ou anti-horraire selon un degré donné.

```
#divDeux {  
  -ms-transform: rotate(20deg); /* IE 9 */  
  -webkit-transform: rotate(20deg); /* Safari */  
  transform: rotate(20deg);  
}
```

skewX(), skewY()

Transforme un élément sur un axe diagonal X ou Y.

```
#div3Trois {  
  -ms-transform: skewY(20deg); /* IE 9 */  
  -webkit-transform: skewY(20deg); /* Safari */  
  transform: skewY(20deg);  
}
```

scale()

Augmente ou diminue les propriétés height et width d'un élément.

```
#divQuatre {  
  -ms-transform: scale(2,3); /* IE 9 */  
  -webkit-transform: scale(2,3); /* Safari */  
  transform: scale(2,3);  
}
```

matrix()

Combinaison de toutes les méthodes en une seule ligne.

```
#divCinq {  
  -ms-transform: matrix(1, -0.3, 0, 1, 0, 0); /* IE 9 */  
  -webkit-transform: matrix(1, -0.3, 0, 1, 0, 0); /* Safari */  
  transform: matrix(1, -0.3, 0, 1, 0, 0);  
}
```



LA TRANSFORMATION 3D AVEC CSS3

La propriété rotate en CSS3 permet de modifier l'aspect d'un élément du DOM sur trois dimensions. Cette propriété dispose de trois méthodes différentes : rotateX(), rotateY() et rotateZ().

rotateX()

Tourne un élément autour de son axes X.



```
#divUn {  
  -webkit-transform: rotateX(150deg); /* Safari */  
  transform: rotateX(150deg);  
}
```

rotateZ()

Tourne un élément autour de son axes Z
(centre de l'élément).



```
#divDeux {  
  -webkit-transform: rotateZ(90deg); /* Safari */  
  transform: rotateZ(90deg);  
}
```

rotateY()

Tourne un élément autour de son axes Y.



```
#div3Trois {  
  -webkit-transform: rotateY(120deg); /* Safari */  
  transform: rotateY(120deg);  
}
```



LES EFFETS DE TRANSITION AVEC CSS3

La propriété transition en CSS3 permet de modifier les valeurs d'un élément en douceur pour déclencher des animations.

Transition simple

Permet de modifier la propriété et de spécifier la durée de la transition.

```
#div1 {  
  -webkit-transition: width 2s; /* Safari */  
  transition: width 2s;  
}
```

Transition multiple

Permet de modifier plusieurs propriétés et de spécifier la durée des transitions.

```
#div2 {  
  -webkit-transition: width 2s, height 4s; /* Safari */  
  transition: width 2s, height 4s;  
}
```

Delai avant transition

Permet de définir un temps en seconde avant que la transition commence.

```
#div3 {  
  -webkit-transition-delay: 1s; /* Safari */  
  transition-delay: 1s;  
}
```

Effets de transition

Il est possible de définir une des cinq courbes qui correspondent à des effets de transition.

```
#div4 {  
  -webkit-transition-timing-function: linear; /* Safari */  
  transition-timing-function: linear;  
}
```

Ecriture compressée

Il est possible de définir les propriétés une par une ou en une seule ligne.

```
#div5 {  
  -webkit-transition: width 2s height 4s linear; /* Safari */  
  transition: width 2s height 4s linear;  
}
```



TRANSFORMATION 3D ET TRANSITION CSS3

Les propriétés `transition: ...;` et `transform: ...;` ayant besoin des mêmes constructeurs, il est possible de les combiner pour réaliser des transitions de forme. La propriété `transform: ...;` est alors configurée comme une propriété CSS classique.

transition: ...;

Les propriétés de transition s'appliquent sur des éléments inactifs pour être chargées en premier.

transform: ...;

Cette propriété peut être appelée car elle a besoin des mêmes constructeurs que la propriété transition

Application de la propriété transform: ...;

Lorsqu'elles sont combinées avec les propriétés de transition, les propriétés de transformation peuvent être appliquées au survol d'un élément pour animer la transformation.

```
div {  
  width: 100px;  
  height: 100px;  
  background: red;  
}
```

```
-webkit-transition: width 2s, height 2s, -webkit-transform 2s;  
transition: width 2s, height 2s, transform 2s;
```

```
div:hover {  
  width: 300px;  
  height: 300px;  
}
```

```
-webkit-transform: rotate(180deg);  
transform: rotate(180deg);
```



LES ANIMATIONS EN CSS3

Les propriétés d'animation permettent de modifier les styles CSS d'un élément en créant des étapes pour créer une animation homogène. Il faut d'abord définir les @keyframes qui correspondent aux étapes de l'animation et leur donner un nom, ce nom sera ensuite appelé avec la propriété animation-name.

@keyframes

Toutes les propriétés CSS sont disponibles dans les @keyframes et il est possible de définir autant d'étapes que voulu. Pour une animation en deux étapes, utiliser les @keyframes from{...} et to{...}.

@-webkit-keyframes

Constructeur pour Chrome, Safari et Opera.

Animation

Il faut appeler l'animation dans la propriété animation-name et définir une durée en seconde avec la propriété animation-duration.

```
@keyframes animationColorPosition {
  0%   { background-color: red; left:0px; top:0px; }
  50%  { background-color: blue; left:200px; top:200px; }
  100% { background-color: green; left:0px; top:0px; }
}

@-webkit-keyframes animationColorPosition {
  0%   { background-color: red; left:0px; top:0px; }
  50%  { background-color: blue; left:200px; top:200px; }
  100% { background-color: green; left:0px; top:0px; }
}

div {
  width: 100px;
  height: 100px;
  background-color: red;
  animation-name: animationColorPosition;
  animation-duration: 4s;
}
```




LES PROPRIETES DES ANIMATIONS CSS3

Comme les transitions, les animations possèdent différentes propriétés qui permettent de configurer une animation complexe. Comme pour toutes les propriétés CSS multiples, deux méthodes d'écriture sont disponibles.

animation-name: ...;
Nom de l'animation définie par les @keyframes

animation-duration: ...;
Durée de l'animation, en seconde.

animation-timing-function: ...;
Courbe d'effet de transition : linear, ease, ease-in, ease-out, ease-in-out et cubic-bezier(n,n,n,n)

animation-delay: ...;
Temp en secondes avant le lancement de l'animation

animation-iteration-count: ...;
Nombre de cycles de l'animation : un chiffre ou infinite

animation-direction: ...;
Sens de l'animation : normal, reverse, alternate et alternate-reverse

```
div {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
}

div {
  animation-name: animationColorPosition;
  animation-duration: 4s;
  animation-timing-function: linear;
  animation-delay: 2s;
  animation-iteration-count: 3;
  animation-direction: reverse;
}

div {
  animation: animationColorPosition 4s linear 2s 3 reverse;
}
```

METHODE SIMPLIFIÉE
Il est recommandé d'utiliser cette méthode en respectant l'ordre des propriétés, pour alléger le poids du fichier.



LES DEGRADES DE COULEURS EN CSS3

L'application des propriétés de dégradés peut être assez fastidieuse mais permet de réaliser des fonds de couleurs précises et bien plus légères qu'une image. Les dégradés de couleurs peuvent être de deux types : linear-gradient ou radial-gradient.

LES COULEURS

Les couleurs doivent être séparées d'une virgule.
Il est possible d'utiliser toutes les valeurs de couleur : red, #FF0000 ou rgba(255, 0, 0, 1).



```
#degradeUn {  
  background: -webkit-linear-gradient(red, green, blue);  
  background: -o-linear-gradient(red, green, blue);  
  background: -moz-linear-gradient(red, green, blue);  
  background: linear-gradient(red, green, blue);  
}
```

LA DIRECTION

Il est possible de configurer l'orientation d'un dégradé linéaire avec les valeurs suivantes : top, right, bottom, left, top right, etc...



```
#degradeDeux {  
  background: -webkit-linear-gradient(left top, red, green, blue);  
  background: -o-linear-gradient(left top, red, green, blue);  
  background: -moz-linear-gradient(left top, red, green, blue);  
  background: linear-gradient(left top, red, green, blue);  
}
```

L'INTENSITÉ

En ajoutant un pourcentage après la définition d'un couleur, il est possible de modifier l'intensité de la couleur dans le dégradé.



```
#degradeTrois {  
  background: -webkit-radial-gradient(red 5%, green 15%, blue 60%);  
  background: -o-radial-gradient(red 5%, green 15%, blue 60%);  
  background: -moz-radial-gradient(red 5%, green 15%, blue 60%);  
  background: radial-gradient(red 5%, green 15%, blue 60%);  
}
```

LA REPETITION

Il est également possible de répéter le dégradé.



```
#degradeQuatre {  
  background: -webkit-repeating-radial-gradient(red, green, blue);  
  background: -o-linear-repeating-radial-gradient(red, green, blue);  
  background: -moz-repeating-radial-gradient(red, green, blue);  
  background: repeating-radial-gradient(red, green, blue);  
}
```