

3. LES TRANSFORMATIONS EN CSS3

Introduction

Les transformations permettent de modifier, avec des propriétés CSS, l'affichage d'éléments HTML d'une page. Une fois affiché de manière « classique », un élément HTML pourra « subir » des transformations. Pour effectuer une transformation, la propriété `transform` est employée. Cette propriété utilise ensuite des fonctions pour appliquer telle ou telle transformation.

La propriété CSS `transform` permet d'appliquer des transformations en deux dimensions sur un élément : *rotation, décalage, zoom, déformation et perspective*.

Les préfixes propriétaires des navigateurs ne sont plus indispensables pour les transformations 2D¹ mais applicables aux transformations 3D².

1. `rotate` : rotation

La fonction `rotate` utilise des valeurs exprimées en degrés (deg) ou en radians (rad), elle permet de tourner les éléments.

```
/* pour tourner les éléments de classe rotation de 45 degrés dans le sens des aiguilles d'une montre*/
.rotation {
  -moz-transform: rotate(45deg);
  -webkit-transform: rotate(45deg);
  -o-transform: rotate(45deg);
  transform: rotate(45deg);
}
```

2. `scale, scaleX, scaleY` : fonction de zoom

La fonction `scale` permet une mise à l'échelle d'un élément selon un ratio, sur une échelle de 0 à 1. 1 étant la taille initiale, une valeur inférieure à 1 aura pour conséquence de rapetisser l'élément, un chiffre supérieur à 1 va l'agrandir.

La place occupée dans le flux demeure identique : si l'élément est agrandi, ses frères ne seront pas poussés en conséquence.

Mise à l'échelle proportionnelle

```
.diminution {
  -moz-transform: scale(0.5);
  -webkit-transform: scale(0.5);
  -o-transform: scale(0.5);
  transform: scale(0.5);
}
```

Mise à l'échelle non proportionnelle

Si vous spécifiez deux valeurs, la première est la mise à l'échelle horizontale, la deuxième, la mise à l'échelle verticale.

```
.echelle {
```

¹ <http://caniuse.com/#feat=transforms2d>

² <http://caniuse.com/#feat=transforms3d>

```
-moz-transform: scale(.5,.2);
-webkit-transform: scale(.5,.2);
-o-transform: scale(.5,.2);
transform: scale(.5,.2);
}
```

Mise à l'échelle dans une seule direction

Les fonctions scaleX : pour la mise à l'échelle horizontale et scaleY : pour la mise à l'échelle verticale.
Dans cet exemple, la mise à l'échelle est uniquement horizontale :

```
.echelle {
    -moz-transform: scaleX(.5);
    -webkit-transform: scaleX(.5);
    -o-transform: scaleX(.5);
    transform: scaleX(.5);
}
```

3. translate, translateX, translateY : déplacement

Ces fonctions permettent de réaliser un déplacement, sur une distance spécifiée, par rapport à sa position d'origine et selon le point de référence.

Cet exemple déplace l'image de 250 pixels horizontalement et 50 pixels verticalement :

```
.deplacement {
    -moz-transform: translate(250px,50px);
    -webkit-transform: translate(250px,50px);
    -o-transform: translate(250px,50px);
    transform: translate(250px,50px);
}
```

Une translation sur l'axe X (translateX) ou Y (translateY) uniquement peut être opérée.

Déplacement horizontal de 20 pixels :

```
.deplacement {
    -moz-transform: translateX(20px);
    -webkit-transform: translateX(20px);
    -o-transform: translateX(20px);
    transform: translateX(20px);
}
```

4. skewX, skewY : déformation (inclinaison)

La fonction skew permet d'effectuer une déformation de l'élément sur les deux axes (d'obliquer la forme d'un élément). L'unité peut s'exprimer en degrés : deg ou en radians : rad.

```
/* pour incliner un élément dans les 2 axes de 15 degrés*/
.deformation {
-moz-transform: skew(15deg);
-webkit-transform: skew(15deg);
-o-transform: skew(15deg);
transform: skew(15deg);
}
```

```
/* pour une déformation horizontale de 20 degrés et une déformation
verticale de 5 degrés */
.deformation {
    -moz-transform: skew(20deg,-5deg);
    -webkit-transform: skew(20deg,-5deg);
    -o-transform: skew(20deg,-5deg);
    transform: skew(20deg,-5deg);
}
```

5. Le point de référence

Par défaut, toutes les transformations ont comme point de référence le centre de l'élément. Ce point de référence sert de départ pour les calculs des transformations. Ce point de référence peut être modifié avec la propriété `transform-origin`. La valeur spécifiée indique alors le point de référence.

Les valeurs acceptées sont :

- Des **pourcentages**, par défaut la valeur est de 50% 50%, soit au milieu de l'élément.
- Des **mots-clés** : `left`, `center`, `right`, `top`, `center`, `bottom`.
- Des valeurs exprimées en **pixels**

Ex : point de référence à l'origine

`Transform-origin:50% 50%;`

Il est possible de changer cette valeur en utilisant un mot-clef de position (`top`, `right`, `bottom`, `left`) suivi d'une valeur chiffrée dont l'unité peut varier (`px`, `%`, etc.)

```
div {
    transform-origin: top 0 left 0;
}
```

6. Appliquer toutes les transformations

La propriété `transform` accepte plusieurs fonctions les unes à la suite des autres, il suffit d'indiquer les fonctions que vous souhaitez utiliser.

```
.transformation {
    -moz-transform: scale(.7) rotate(3deg) translateX(5px) skewX(-5deg);
    -webkit-transform: scale(.7) rotate(3deg) translateX(5px) skewX(-5deg);
    -o-transform: scale(.7) rotate(3deg) translateX(5px) skewX(-5deg);
    transform: scale(.7) rotate(3deg) translateX(5px) skewX(-5deg);
}
```

Lecture Web :

<http://www.alsacreations.com/article/lire/1418-css3-transformations-2d.html>

Générateurs

<http://www.css3maker.com/css3-transform.html>

<http://css3generator.com/>

<http://www.westciv.com/tools/transforms/index.html>

7. La transformation 3D

La propriété CSS transform 3D fait partie du CSS 3D Transforms Module Level 3. Elle permet de transformer un élément dans un espace en trois dimensions.

Les transformations 3D sont basées sur la même propriété que les transformations 2D, à savoir transform.

En revanche, de nouvelles fonctions de transformations sont disponibles :

rotateX(angle), rotateY(angle) et rotateZ(angle) pour des rotations autour de chaque axe.

rotate3d(x,y,z,angle) pour une rotation autour du vecteur 3D x,y,z.

translateZ(z) pour des translations sur l'axe Z (profondeur).

translate3d(x,y,z) pour des translations en 3D.

scaleZ(z) pour des changements d'échelle sur l'axe Z

scale3d(x,y,z) pour des changements d'échelle en 3D.

matrix3d() spécifie une matrice de transformation 4x4.

De nouvelles propriétés existent pour gérer cet affichage en 3D :

perspective représente la distance entre la caméra et la scène (l'internaute et l'écran). Sans perspective, l'impression de 3D n'existe pas.

perspective-origin pour l'origine de la perspective (Là où les points de fuite des éléments se rejoignent).

transform-style: flat|preserve-3d défini si les éléments inclus subissent les transformations en 3D (La valeur flat indique que les éléments enfants reposent à "plat" sur leur parent et la valeur preserve-3d indique que les éléments reposent dans le plan 3D géré par le parent).

backface-visibility: visible|hidden spécifie si les faces arrière des éléments sont visibles ou non.

Liens

<http://www.html5-css3.fr/css3/transformations-3d-css3> (Bon article explicatif)

<http://thewebrocks.com/demos/3D-css-tester/> (comprendre en testant)

<http://www.creativebloq.com/css3/20-stunning-examples-css-3d-transforms-11112759> (exemples d'utilisation)

http://ie.microsoft.com/testdrive/graphics/hands-on-css3/hands-on_3d-transforms.htm (générateur)