

# Les transformations en CSS3

par debray jerome ([Dji programmation web2 et design](#)) ([Blog](#))

Date de publication :

Dernière mise à jour :

Dans cet article, je vais présenter les transformations 2D utilisables via CSS3 et qui permettent un grand nombre de possibilités visuelles grâce à la propriété **transform**.

Quand ce sera nécessaire, il sera indiqué les préfixes des navigateurs pour les propriétés CSS.

**Compatibilité** : *Chrome, Safari, Opera, Firefox.*

I - Un point d'origine.....	3
II - A savoir.....	3
III - La rotation.....	3
IV - Le redimensionnement.....	4
V - La distorsion.....	5
VI - La translation.....	6
VII - Exemples.....	6
VIII - Remerciements.....	6

## I - Un point d'origine

Les différents exemples qui seront présentés peuvent avoir un point d'origine différent si on le spécifie, grâce à la propriété **transform-origin**. Cette propriété prend deux types de valeurs soit **numérique** (100 px, 50 %...) soit **alphabétique** (left, top, right, bottom).

Voici un exemple :

```
-webkit-transform-origin: 0 0;
-moz-transform-origin: 0 0;
-o-transform-origin: 0 0;
transform-origin: 0 0;
```

ou

```
-webkit-transform-origin: top left;
-moz-transform-origin: top left;
-o-transform-origin: top left;
transform-origin: top left;
```

Par défaut, le point d'origine est *top left*.

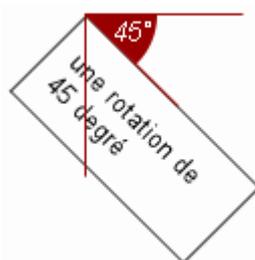
## II - A savoir

La propriété **transform** autorise plusieurs valeurs de transformation à la suite. Ainsi vous pourrez coupler les différents types de transformation entre eux :

```
-webkit-transform: skew(30deg, 15deg) translate(10px, 0px) rotate(-30deg);
-moz-transform: skew(30deg, 15deg) translate(10px, 0px) rotate(-30deg);
-o-transform: skew(30deg, 15deg) translate(10px, 0px) rotate(-30deg);
transform: skew(30deg, 15deg) translate(10px, 0px) rotate(-30deg);
```



## III - La rotation



La rotation est possible via la propriété **transform** prenant pour valeur **rotate(x)**. L'argument de **rotate** correspond à la valeur de l'angle de rotation et peut être négatif.

```
-webkit-transform: rotate(45deg);
-moz-transform: rotate(45deg);
-o-transform: rotate(45deg);
transform: rotate(45deg);
```

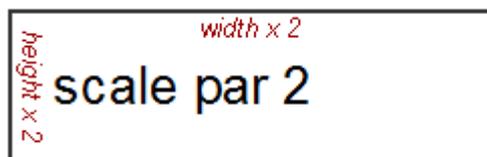
**Attention** : Sous les navigateurs Webkit (Chrome et Safari), on ne peut pas appliquer la transformation à n'importe quel tag HTML ( par exemple le span ne permet pas de transformation de type rotation).

## IV - Le redimensionnement

La valeur **scale** de la propriété **transform** permet de dilater/redimensionner un élément. Cette valeur peut prendre **un ou deux arguments** : **scale(x)** ou **scale(x, y)**.

Avec une seule valeur, le **x** correspond au coefficient de dilatation en **largeur (width)** et en **hauteur (height)**. Avec deux valeurs, **x** correspond à la **largeur (width)** et **y** à la **hauteur (height)**.

```
-webkit-transform: scale(2);
-moz-transform: scale(2);
-o-transform: scale(2);
transform: scale(2);
```



ou

```
-webkit-transform: scale(2, 1);
-moz-transform: scale(2, 1);
-o-transform: scale(2, 1);
transform: scale(2, 1);
```



Il est à noter que l'on peut utiliser les sous-valeurs de **scale** : **scaleX** et **scaleY**.

Exemple:

```
transform: scaleX(2) scaleY(1);
```

ou

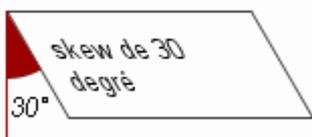
```
transform: scaleX(2);
```

## V - La distorsion

La valeur **skew** de la propriété **transform** permet de tordre un élément. Cette valeur peut prendre **un ou deux arguments** (unité en degré).

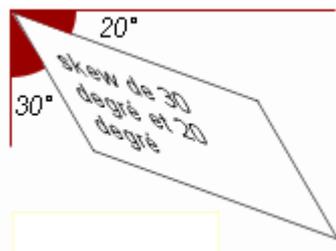
S'il n'y a qu'un argument, cela équivaut à une distorsion horizontale. S'il y a le deuxième argument, celui-ci contrôlera la distorsion verticale.

### glissement horizontal :



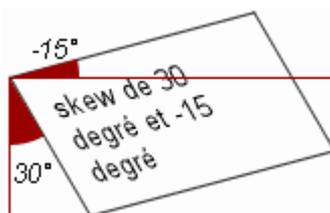
```
-webkit-transform: skew(30deg);
-moz-transform: skew(30deg);
-o-transform: skew(30deg);
transform: skew(30deg);
```

### glissement horizontal et vertical :



```
-webkit-transform: skew(30deg, 20deg);
-moz-transform: skew(30deg, 20deg);
-o-transform: skew(30deg, 20deg);
transform: skew(30deg, 20deg);
```

Les arguments de la valeur **skew** peuvent être négatifs :



```
-webkit-transform: skew(30deg, -15deg);
-moz-transform: skew(30deg, -15deg);
-o-transform: skew(30deg, -15deg);
transform: skew(30deg, -15deg);
```

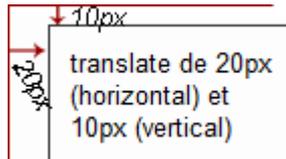
Tout comme **scale**, **skew** peut être utilisé sous cette forme :

```
transform:skewY(30deg) skewX(deg);
```

## VI - La translation

La valeur **translate** de la propriété **transform** permet de déplacer un élément en **x** et **y** par rapport à sa position d'origine. Cette valeur peut prendre un ou deux arguments.

Le premier argument correspond à une translation en **x** (horizontale) et le deuxième argument à une translation en **y** (verticale).



```
-webkit-transform: translate(20px, 10px);
-moz-transform: translate(20px, 10px);
-o-transform: translate(20px, 10px);
transform: translate(20px, 10px);
/*translation horizontale de 20px et translation verticale de 10px*/
```

```
-webkit-transform: translate(20px);
-moz-transform: translate(20px);
-o-transform: translate(20px);
transform: translate(20px);
/*translation horizontale et verticale de 20px*/
```

Ces arguments peuvent être négatifs.

Tout comme **scale** et **skew**, **translate** peut être utilisé sous cette forme :

```
transform: translateY(20px) translateX(10px);
```

## VII - Exemples

<http://debray-jerome.developpez.com/demos/transformations.html>

## VIII - Remerciements

Merci à **Sepia** pour sa relecture attentive et ses conseils.