

Fiche les transformations en CSS3

La **propriété transform** permet de manipuler un élément HTML sur les axes X et Y (horizontal et vertical) grâce à des fonctions diverses de transformation graphique. Il est possible de modifier l'apparence d'un élément grâce à cet ensemble de fonctions

- Translation avec **translate**(valeur sur x, valeur sur y) - permet de déplacer l'élément.
- Mise à l'échelle avec **scale**(valeur coeff) - permet de faire un zoom sur un objet
- Rotation avec **rotate**(valeur en deg) - permet de faire pivoter un élément sur lui-même.
- Inclinaison avec **skewX**(valeur en deg) et **skewY**(valeur en deg) - permet de créer une inclinaison sur un objet

Sous la forme >> transform: **function(value);** *avec utilisation des préfixes propriétaires.*

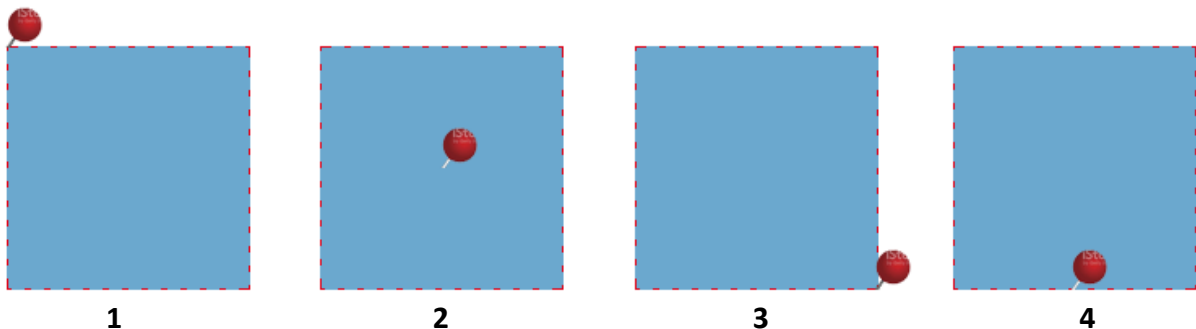
Pour pouvoir appliquer des transformations, nous avons besoin de **définir un point d'origine** (d'ancrage) de la transformation. La propriété transform-origin permet de définir ce point d'origine.

Par défaut la valeur initiale de cette propriété est le centre de l'élément.

Si l'on veut définir le point en haut et à gauche :

transform-origin: top left;

Quelques exemples de positionnement du point d'origine :



- | | |
|---------------------|--|
| 1. en haut à gauche | transform-origin: top left; |
| 2. au centre | transform-origin: center center; (valeur par défaut) |
| 3. en bas à droite | transform-origin: bottom right; |
| 4. en bas au centre | transform-origin: bottom center ; |

I. Les fonctions de la propriété transform

Une fois l'origine choisie, nous pouvons effectuer des transformations sur les éléments avec la propriété transform suivie de fonctions.

1. La fonction translate()

Elle permet d'effectuer une translation (un déplacement) de l'élément sur les axes X et Y et s'écrit sous cette forme >> *transform: **translate**(x,y);*

Les deux valeurs peuvent être positives ou négatives. La valeur y est **optionnelle**. Par défaut elle est à zéro.

-webkit-transform: translate(10px, 10px);

-moz-transform: translate(10px, 10px);

-o-transform: translate(10px, 10px);

-ms-transform: translate(10px, 10px);

transform: translate(10px, 10px);

Cette propriété peut s'écrire également sous la forme >> transform: translateX(value) translateY(value);

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

Les fonctions translateX() et translateY() permettent surtout de réaliser une translation sur un seul axe axes, l'axe des X (translateX()) ou l'axe des Y (translateY())

transform: translateX(value); ou transform: translateY(value);

2. La fonction scale().

Cette fonction permet d'agir sur l'échelle (les dimensions) de l'élément. La valeur initiale est **1**, tandis que les valeurs supérieures à 1 créent un effet d'agrandissement, et les valeurs inférieures créent un effet de réduction.

Elle s'écrit sous cette forme >> *transform: **scale**(valeur);*

-webkit-transform: scale(2);

-moz-transform: scale(2);

-o-transform: scale(2);

transform: scale(2);

! Si vous agrandissez une image elle peut devenir floue, pixélisée. Il est préférable de l'insérer dès le départ dans sa taille zoomée et modifier ses dimensions en css.

Les fonctions scaleX() et scaleY()

Sur le même principe que pour les fonctions dérivées de translate, ces deux fonctions permettent de définir indépendamment les valeurs x et y.

Il est également possible d'écrire sous cette forme : **scale(x,y)**

-webkit-transform: scale(.75, 1.25);

-moz-transform: scale(.75, 1.25);

-o-transform: scale(.75, 1.25);

transform: scale(.75, 1.25);

Essai : Introduire un texte et essayez avec scale(1,-1), scale(-1,1), scale(-1,-1)

3. La fonction rotate()

Il s'agit d'une des plus simples fonctions à comprendre. Comme son nom l'indique, elle permet d'effectuer une **rotation** de l'élément ciblé.

Elle s'écrit sous cette forme >> *transform: **rotate**(value en deg);*

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

Attention cependant, le point d'origine de la transformation a son importance.

4. La fonction skewX() et skewY()

Ces fonctions permettent d'obliquer la forme d'un élément.

```
transform: skewX(valeur en deg);  
-webkit-transform: skewX(30deg);  
-moz-transform: skewX(30deg);  
-o-transform: skewX(30deg);  
-ms-transform: skewX(30deg);  
transform: skewX(30deg);
```

transform: skewX(30deg);
ou en changeant les valeurs de X et Y : *transform: skewX(30deg) skewY(30deg);*

2. Ordre des déclarations

La propriété transform accepte plusieurs fonctions les unes à la suite des autres, comme proposé précédemment. Il est possible d'effectuer ces transformations combinées en espaçant les fonctions d'un simple caractère blanc

```
transform : function1(value) function2(value) function3(value);
```

L'ordre de déclaration des fonctions a son importance.