

# APPLIQUER LES TRANSFORMATIONS 2D ET 3D EN CSS

# transform

Une transformation en CSS s'applique sur n'importe quel élément grâce à la propriété **transform**.

Cette propriété transform peut prendre comme valeur une multitude de ***fonctions de transformation***, dont certaines agissent en 2D et d'autres en 3D.

```
transform: fonctionA() fonctionB() fonctionC() ... ;
```

Ces fonctions s'appliquent dans l'ordre défini, et sont séparés par des **espaces**.

# Fonctions de transformation : translate()

Les fonctions de “translation” ont pour objectif de **décaler le rendu visuel** de l’objet dans l’espace (mais pas dans le flux).

**translateX**(x)

**translateY**(y)

**translateZ**(z)

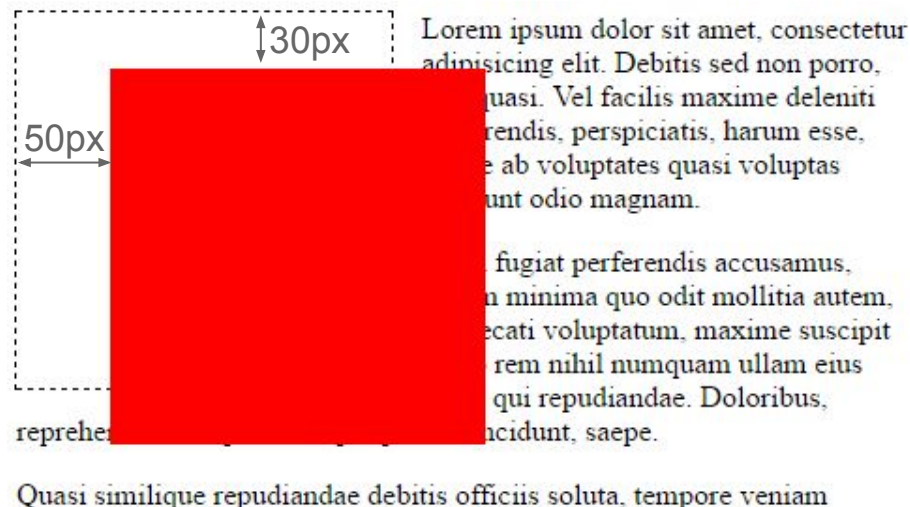
**translate**(x, y)

**translate3d**(x, y, z)

# Fonctions de transformation : translate()

Un exemple d'utilisation :

```
div {  
  float: left;  
  width: 200px;  
  height: 200px;  
  background-color: red;  
  transform: translate(50px, 30px);  
}
```



# Fonctions de transformation : scale()

Les fonctions de “scaling” permettent de jouer sur la taille de l’élément à l’écran. Elles prennent un nombre entier comme multiplicateur :

**scaleX**(2) /\* 2 fois sa taille en X \*/

**scaleY**(1.5) /\* 1,5 fois sa taille en Y \*/

**scaleZ**(3) /\* 3 fois sa taille en Z \*/

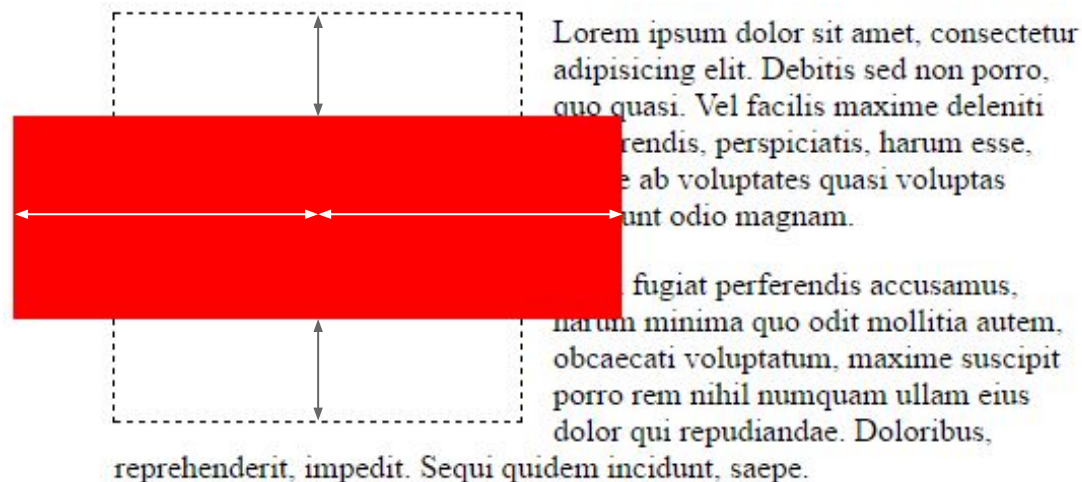
**scale**(2, 1.5)

**scale3d**(2, 1.5, 3)

# Fonctions de transformation : scale()

Un exemple d'utilisation :

```
div {  
  float: left;  
  width: 200px;  
  height: 200px;  
  background-color: red;  
  transform: scale(1.5, 0.5);  
}
```



# Fonctions de transformation : rotate()

Les fonctions de “rotation” permettent de faire tourner les éléments sur les 3 différents axes.

Elles prennent une valeur en degrés, avec l’unité CSS “deg”:

**rotateX**(180deg) /\* rotation de 180 degrés sur l’axe X \*/

**rotateY**(45deg) /\* rotation de 45 degrés sur l’axe Y \*/

**rotateZ**(360deg) /\* rotation de 360 degrés sur l’axe Z \*/

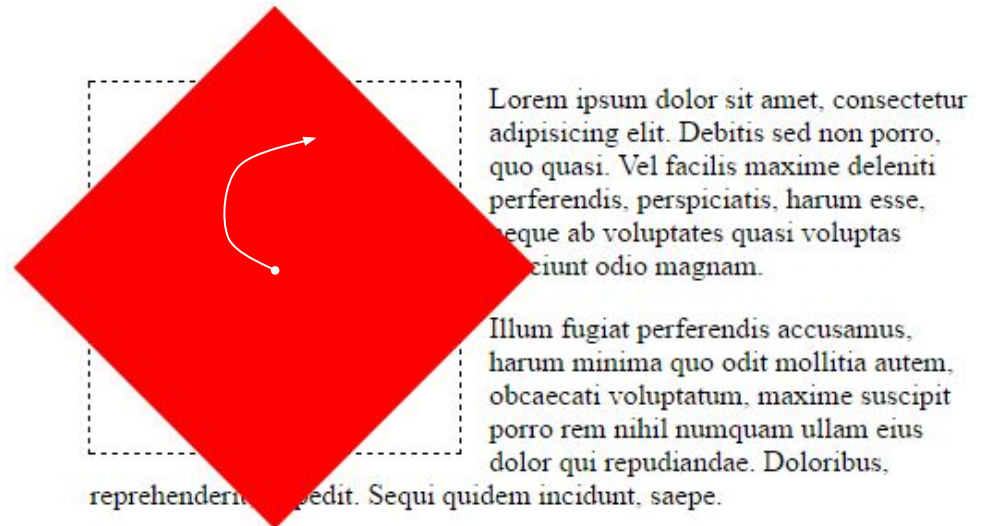
**rotate**(-90deg) /\* rotation de -90 degrés sur l’axe Z \*/

**rotate3d**(180deg, 45deg, 360deg)

# Fonctions de transformation : rotate()

Un exemple d'utilisation :

```
div {  
  float: left;  
  width: 200px;  
  height: 200px;  
  background-color: red;  
  transform: rotate(45deg);  
}
```

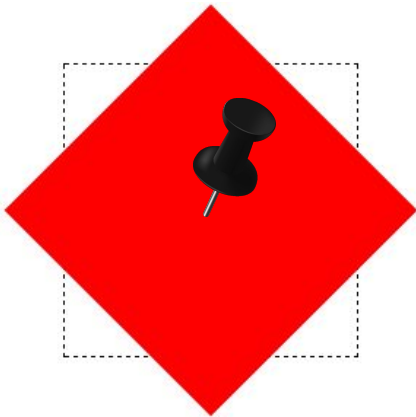




# Point d'origine d'une transformation

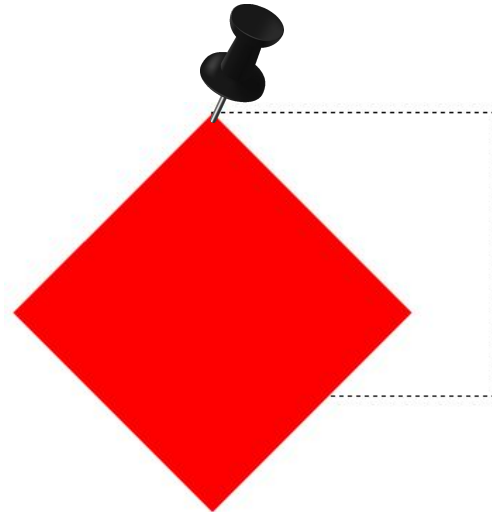
Par défaut, le point d'origine de la transformation est au centre de l'objet (50% 50%).

Il est possible de décaler ce point avec la propriété **transform-origin**.



**transform-origin:** 50% 50%;

(par défaut)



**transform-origin:** 0 0;

# Fonctions de transformation : skew()

Les fonctions de “skew” permettent d'obliquer la forme d'un élément, et prend des valeurs en degrés :

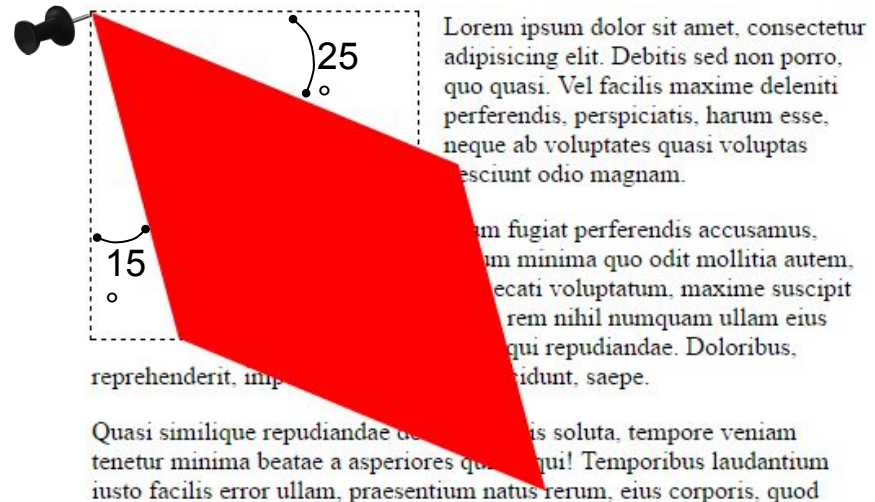
**skewX**(15deg)

**skewY**(45deg)

# Fonctions de transformation : skew()

Un exemple d'utilisation :

```
div {  
  float: left;  
  width: 200px;  
  height: 200px;  
  background-color: red;  
  transform-origin: 0 0;  
  transform: skewX(15deg) skewY(25deg);  
}
```



# Documentation

Documentation officielle :

<https://developer.mozilla.org/fr/docs/Web/CSS/transform>

# Compatibilité navigateurs

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

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

CSS3 2D Transforms <small>WD</small>									
Method of transforming an element including rotating, scaling, etc. Includes support for <code>transform</code> as well as <code>transform-origin</code> properties.									
Global 92.81%									
unprefixed: 79.07%									
Current aligned Usage relative Show all									
IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Chrome for Android
			49						
		47	51					4.4	
8	13	48	52	9.1		9.3		4.4.4	
11	14	49	53	10	39	10	all	52	51

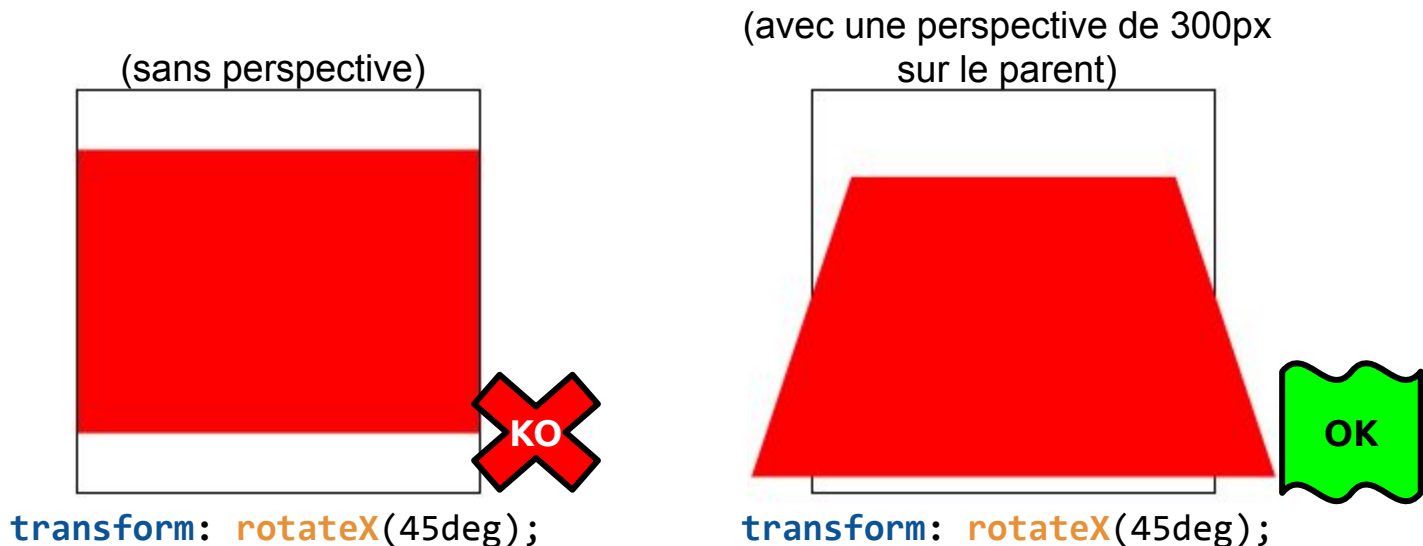
CSS3 3D Transforms <small>WD</small>									
Method of transforming an element in the third dimension using the <code>transform</code> property. Includes support for the <code>perspective</code> property to set the perspective in z-space and the <code>backface-visibility</code> property to toggle display of the reverse side of a 3D-transformed element.									
Global 86.9% + 5.27% = 92.17%									
unprefixed: 73.69% + 5.27% = 78.96%									
Current aligned Usage relative Show all									
IE	Edge	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Android Browser	Chrome for Android
			49						
		47	51					4.4	
8	13	48	52	9.1		9.3		4.4.4	
11	14	49	53	10	39	10	all	52	51

3D

# 3D : La perspective

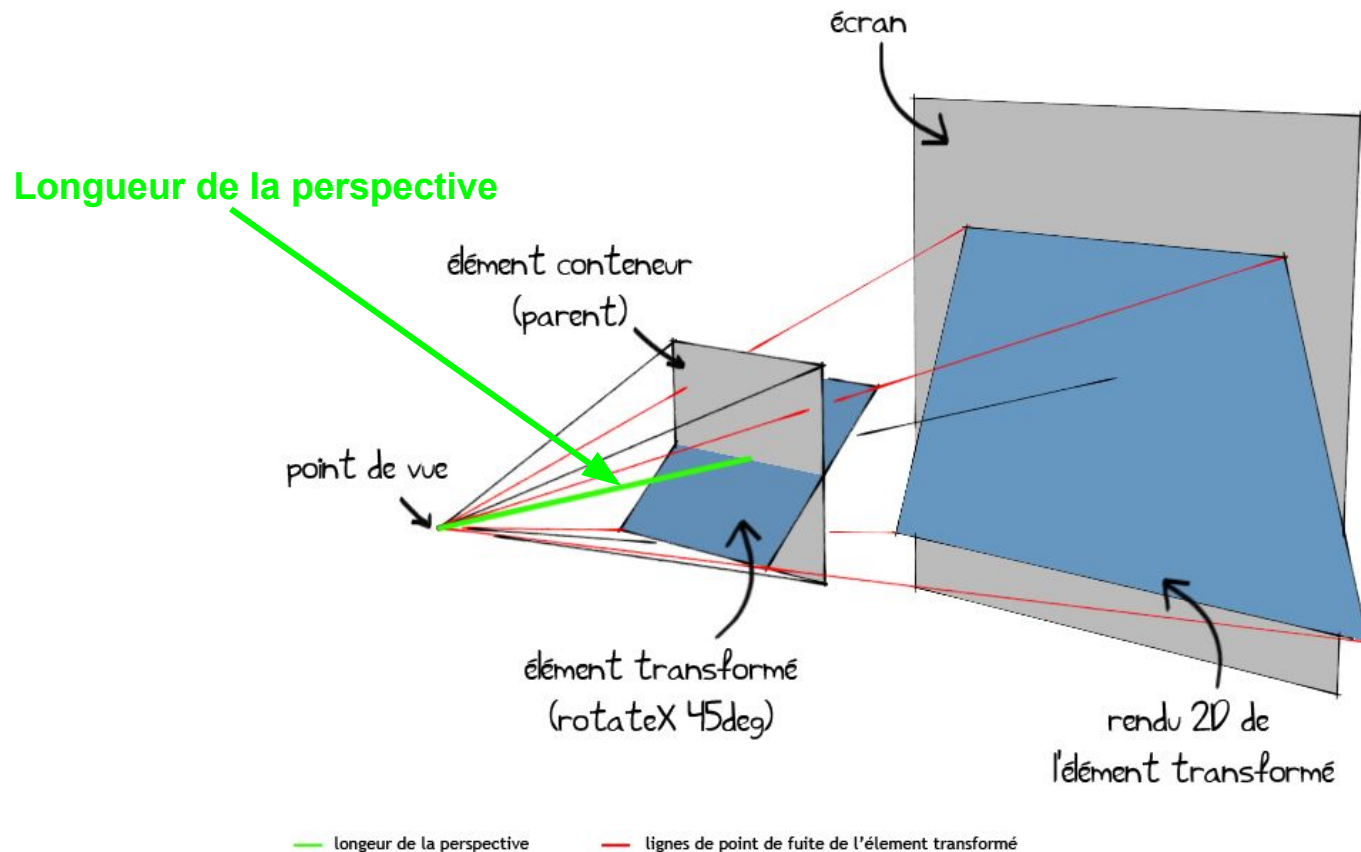
Afin d'appliquer un rendu visuel 3D, il faut obligatoirement appliquer une **perspective** à l'élément conteneur (le parent) de l'objet qui sera en 3D.

En effet, sans la notion de perspective, un objet défini en 3D n'aura pas un rendu en “profondeur”.



# 3D : La perspective

La perspective représente la distance entre le point de vue et l'élément HTML sur lequel la transformation est appliquée.



Cette image est tirée de l'[article sur les transformations 3D en CSS](#), de [Vincent de Oliveira \(@iamvdo\)](#).



# 3D : La perspective

La propriété CSS permettant d'appliquer une perspective s'intitule tout simplement :

**perspective: 500px;**

```
<div class="scene">
  <p>Coucou ! Je fais de la 3D !</p>
</div>
```

```
.scene {
  perspective: 300px;
}

.scene p {
  transform: rotateZ(5deg) rotateY(40deg);
}
```

Applique une perspective de 300px sur le parent <div class="scene">

*Coucou ! Je fais de la 3D !*

Applique des transformations sur l'enfant <p>

# 3D : transform-style

Lorsqu'on travaille en 3D avec plusieurs éléments, on souhaite généralement que ces derniers restent positionnés dans un espace **tridimensionnel**.

Mais par défaut, ce n'est pas le cas !

Le navigateur, afin de gagner en rapidité, va **aplanir** les éléments.

On peut donc le forcer à conserver le positionnement tridimensionnel avec la propriété suivante :

```
transform-style: preserve-3d;  
/* notez que le "d" est en minuscules */  
/* la valeur par défaut est flat */
```

# 3D : transform-style

La même animation, avec un **transform-style** différent :

`transform-style: flat;`  
(par défaut)



`transform-style: preserve-3d;`



# Récapitulatif

Les propriétés à retenir dans cette présentation sont les suivantes :

`transform: rotate(...) translate(...) skew(...) ... ;`  
(applique différentes transformations 2D et/ou 3D dans l'ordre défini)

`transform-origin: 100% 0;`  
(indique le point d'origine d'où s'appliqueront les transformations,  
dans cet exemple : **en haut à droite**)

`perspective: 500px;`  
(applique une perspective de 500px de profondeur sur l'élément ciblé (qui doit être le parent des éléments que l'on veut en 3D))

`transform-style: preserve-3d;`  
(indique au navigateur que les éléments doivent rester positionnés dans un espace tridimensionnel)