

# CSS3 - transform

- Les transformations en CSS
- Support par les navigateurs
- Transformations 2D et propriétés associées
- Combiner plusieurs transformations et changer d'origine
- Transformations 3D et propriétés associées
- Perspective, effet Flip


# Les transformations CSS











- Une **transformation** est un ensemble de propriétés CSS qui vont permettre de faire :
  - des rotations
  - des mises à l'échelle
  - des torsions / inclinaisons
  - des déplacements
  - en 2D ou en 3D
- Souvent, on va utiliser les transformations en **combinaison** avec les **transitions** pour créer des effets graphiques et des **animations**

# Support par les navigateurs (2D)

## CSS3 2D Transforms - WD

Method of transforming an element including rotating, scaling, etc.  
Includes support for `transform` as well as `transform-origin` properties.

Usage % of all users  ?  
Global 97.36%  
unprefixed: 96.34%

Current aligned	Usage relative	Date relative	Apply filters	Show all	?										
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baic Brow:
					10.1										
					11.5 										
6-8		2-3			12.1										
9 	12-16 	3.5-15 	4-35 	3.1-8 	15-22 	3.2-8.4 									
10	17-81	16-77	36-81	9-13	23-68	9-13.3		2.1-4.4.4 	12-12.1				4-11.2		
11 	83	78	83	13.1	69	13.5	all	81	46	81	68	12.12	12.0	10.4	7.1
		79-80	84-86	14-TP		14.0									

Les propriétés CSS3 liées aux transformations **2D** sont supportées depuis de nombreuses versions















<https://caniuse.com/#search=css3%20transformation>

# Support par les navigateurs (3D)

## CSS3 3D Transforms - WD

Method of transforming an element in the third dimension using the `transform` property. Includes support for the `perspective` property to set the perspective in z-space and the `backface-visibility` property to toggle display of the reverse side of a 3D-transformed element.

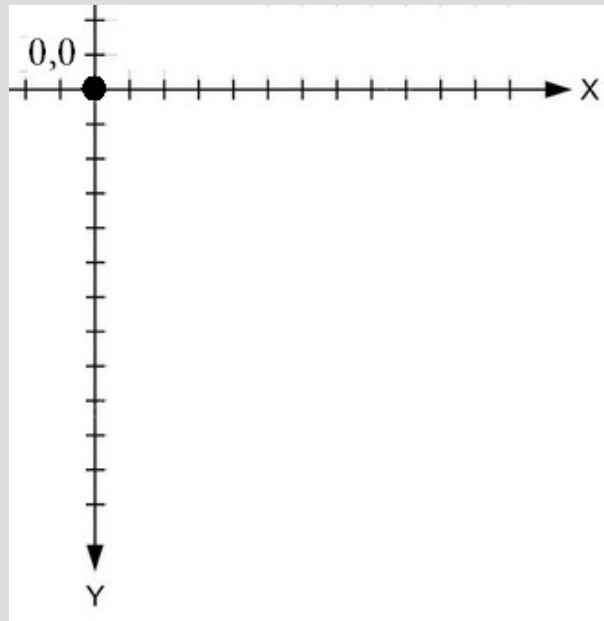
Usage % of all users    
Global 95.6% + 1.4% = 97%  
unprefixed: 94.93% + 1.4% = 96.33%

Current aligned	Usage relative	Date relative	Apply filters	Show all	?										
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baic Brow:
		2-9	4-11	3.1-3.2	10-12.1										
6-9		10-15 	12-35 	4-8 	15-22 	3.2-8.4 		2.1-2.3							
 10	12-81	16-77	36-81	 9-13	23-68	 9-13.3		3-4.4.4 	12-12.1				4-11.2		
 11	83	78	83	 13.1	69	 13.5	all	81	46	81	68	12.12	12.0	10.4	7.1
		79-80	84-86	 14-TP		 14.0									

Les propriétés CSS3 liées aux transformations **3D** sont supportées depuis de nombreuses versions

<https://caniuse.com/#search=css3%20transformation>

# Ecran et axes X,Y,...Z



- Origine : Coin supérieur gauche de l'écran ( $X=0, Y=0$ )
- Axe X : vers la droite
- Axe Y : vers le bas
- *Axe Z : vers l'observateur (pour la 3D)*

# Positionnement

- Les éléments subissant une **transformation** se comportent, **par défaut**, comme s'ils étaient **positionnés en position:relative**; car l'espace « libéré » par leur transformation n'est pas comblé et **l'élément transformé se retrouve dans un « calque »** superposé au reste de la page.

# Rotation

- On utilise la propriété **transform** avec la valeur **rotate(...)** et on définit l'angle de rotation
- L'angle est exprimé généralement en degrés (**deg**) et peut être **positif ou négatif**
- Il y a des variations possibles (en 2D) :
  - **rotateX(...)** : rotation autour de l'axe des X
  - **rotateY(...)** : rotation autour de l'axe des Y

Voir Exemple 1

# HTML pour les exemples

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Transformations CSS</title>
  <link type="text/css" rel="stylesheet" href="transform.css" />
</head>
<body>
  <h1>Les transformations CSS</h1>
  <main>
    <div id="element">
      <p>Survolez-moi !<br>Je me transforme...</p>
    </div>
    <div id="commentaires">
      <p>Les transformations permettent de faire :</p>
      <ul>
        <li>des rotations</li>
        <li>des mises à l'échelle</li>
        <li>des torsions / inclinaisons</li>
        <li>des déplacements</li>
        <li>en 2D ou en 3D</li>
      </ul>
      <p>Les éléments subissant une transformation se comportent, par défaut, comme s'ils étaient positionnés <strong>en position:relative;</strong> car l'espace « libéré » par leur transformation n'est pas comblé et l'élément transformé se retrouve dans un « calque » superposé au reste de la page.</p>
    </div>
  </main>
</body>
</html>
```



# CSS pour les exemples

```
body {
  background-image: url(axes.png);
  background-repeat: no-repeat;
  background-position: 100px 100px;
}

#commentaires {
  font-size: 1.5em;
  margin-left: 650px;
  margin-top: 100px;
}

#element {
  width :200px;
  height:200px;
  border: 3px solid black;
  border-radius: 10px;

  position: absolute;
  top : 233px;
  left: 233px;
}

#element p {
  text-align: center;
  line-height: 100px;
  margin:0;
  background-color: rgba(150,0,0,0.80);
  color:yellow;
  font-weight: bold;
  font-size: 125%;
}
```

# Exemple 1

```
#element {
  width :200px;
  height:200px;
  border: 3px solid black;
  border-radius: 10px;

  position: absolute;
  top : 233px;
  left: 233px;

  /* une transition pour voir le mouvement au ralenti */
  transition: transform 2s linear;
}

/* Rotation */
#element:hover {
  transform: rotate(90deg);
}
```

Une rotation de 90 degrés (1/4 de tour) vers la droite (sens horlogique)

# Mise à l'échelle

- On utilise la propriété **transform** avec la valeur **scale(...)** et on définit le facteur d'échelle
- Le facteur d'échelle est une valeur numérique  $>1$  (**agrandit**) ou une valeur numérique  $<1$  (**diminue**)
- Il y a des variations possibles (en 2D) :
  - **scaleX(...)** : mise à l'échelle autour de l'axe des X
  - **scaleY(...)** : mise à l'échelle autour de l'axe des Y
  - **scale(...,...)** : mise à l'échelle autour des 2 axes

Voir Exemple 2

# Exemple 2

```
#element {  
    width :200px;  
    height:200px;  
    border: 3px solid black;  
    border-radius: 10px;  
  
    position: absolute;  
    top : 233px;  
    left: 233px;  
  
    /* une transition pour voir le mouvement au ralenti */  
    transition: transform 2s linear;  
}  
  
/* Echelle */  
#element:hover {  
    transform: scale(1.2);  
}
```

Une augmentation de la taille de 20 % (1.2 = 120%)

# Torsion

- On utilise la propriété **transform** avec la valeur **skew(...)** et on définit l'angle de torsion
- L'angle est exprimé généralement en degrés (**deg**) et peut être **positif ou négatif**
- Il y a des variations possibles (en 2D) :
  - **skewX(...)** : torsion autour de l'axe des X
  - **skewY(...)** : torsion autour de l'axe des Y
  - **skew(...,...)** : torsion autour des 2 axes

Voir Exemple 3

# Exemple 3

```
#element {  
    width :200px;  
    height:200px;  
    border: 3px solid black;  
    border-radius: 10px;  
  
    position: absolute;  
    top : 233px;  
    left: 233px;  
  
    /* une transition pour voir le mouvement au ralenti */  
    transition: transform 2s linear;  
}  
  
/* Inclinaison */  
#element:hover {  
    transform: skew(30deg);  
}
```

Une inclinaison (torsion) de 30 degrés

# Déplacement

- On utilise la propriété **transform** avec la valeur **translate(...)** et on définit le déplacement
- Le déplacement est exprimé par un nombre (px,em,...) et peut être **positif ou négatif**
- Il y a des variations possibles (en 2D) :
  - **translateX(...)** : déplacement sur l'axe des X
  - **translateY(...)** : déplacement sur l'axe des Y
  - **translate(..., ...)** : déplacement sur les 2 axes

Voir Exemple 4

# Exemple 4

```
#element {  
    width :200px;  
    height:200px;  
    border: 3px solid black;  
    border-radius: 10px;  
  
    position: absolute;  
    top : 233px;  
    left: 233px;  
  
    /* une transition pour voir le mouvement au ralenti */  
    transition: transform 2s linear;  
}  
  
/* Déplacement */  
#element:hover {  
    transform: translate(250px,50px);  
}
```

Un déplacement de 250px à droite (axe X) et de 50px en bas (axe Y)



# Plusieurs transformations

- Si on souhaite appliquer à un élément **plusieurs transformations**, il suffit de les placer dans la **même propriété transform, séparées par un espace**

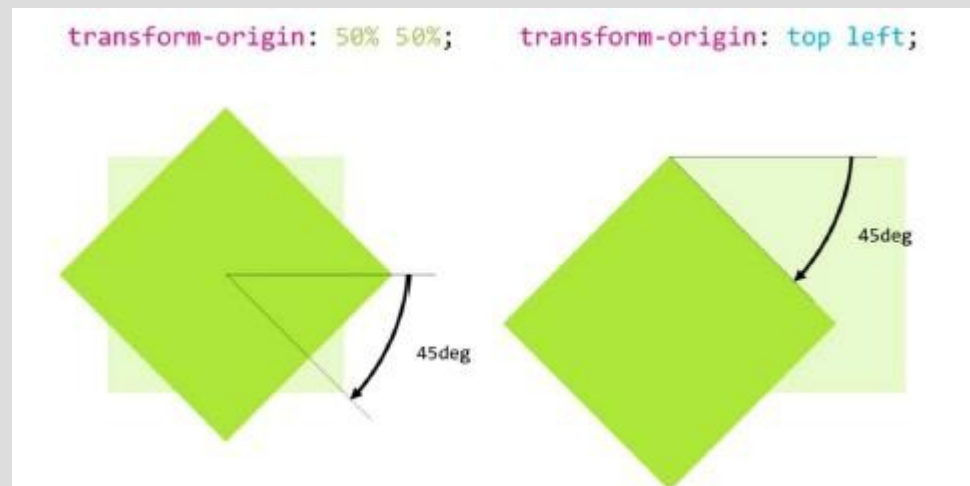
transform : rotate(45deg) translateX(100px) ;

- Attention à l'ordre, le résultat peut être différent
- Il existe aussi une propriété **matrix(...)** avec six paramètres qui résume plusieurs transformations via une formule mathématique complexe

<http://angrytools.com/css-generator/transform/>

# Changement d'origine

- Par défaut, toutes les transformations ont pour origine le centre de l'élément, (X=50%,Y=50%)
- La propriété **transform-origin** permet de modifier l'origine. Les valeurs peuvent être exprimées en %, en px (>0 ou <0) ou avec des mots-clés (top, bottom, left, right, center).



# Exemple 5

```
#element {
    width :200px;
    height:200px;
    border: 3px solid black;
    border-radius: 10px;

    position: absolute;
    top : 233px;
    left: 233px;

    /* une transition pour voir le mouvement au ralenti */
    transition: transform 2s linear;

    /* L'origine est le coin supérieur gauche de l'élément */
    transform-origin: top left;      /* équivalent à 0% 0% */
}

/* Déplacement */
#element:hover {
    transform: rotate(90deg);
}
```