

CSS Transitions et transformations

1.	TRANSITIONS	2
A.	SANS TRANSITION.....	2
B.	AVEC TRANSITION	2
	<i>Propriétés</i>	<i>2</i>
	<i>Trigger</i>	<i>2</i>
	<i>Raccourci</i>	<i>2</i>
	<i>Avec préfixes pour les différents navigateurs.....</i>	<i>3</i>
	<i>Définir plusieurs propriétés</i>	<i>3</i>
	<i>Panneau Transitions de Dreamweaver.....</i>	<i>3</i>
C.	EXEMPLES	4
2.	ANIMATIONS AVEC KEYFRAMES	5
3.	TRANSFORMATIONS	7
A.	DEPLACER UN ELEMENT (TRANSLATE OU TRANSLATEX, TRANSLATEY).....	7
B.	REDIMENSIONNER UN ELEMENT (SCALE OU SCALEX / SCALEY)	7
C.	TOURNER UN ELEMENT (ROTATE).....	8
D.	« INCLINER » UN ELEMENT (SKEW OU SKEWX / SKEWY)	8
E.	TRANSFORM-ORIGIN.....	8
4.	TRANSFORMATIONS 3D.....	9
A.	DEPLACEMENT (TRANSLATE3D)	9
B.	ROTATION (ROTATE3D).....	10
C.	SCALE (SCALE3D)	11
D.	PERSPECTIVE-ORIGIN	11
E.	EXEMPLES	12
	<i>Flip</i>	<i>12</i>
	<i>Création d'un cube 3d.....</i>	<i>14</i>
5.	RESSOURCES	15

1. Transitions

2 états : début et fin

a. Sans transition

```
a{
  background-color:Orange;
}
a:hover{
  background-color:Red;
}
```

b. Avec transition

Propriétés

- « transition-property » : la propriété à animer, changer
- « transition-duration » : durée de l'animation
- « transition-delay » : délai avant de jouer l'animation
- « transition-timing-function » fonction easing (ease,linear, ease-in,ease-out,ease-in-out,etc.)

Trigger

- :hover
- :focus
- :active
- :target
- :checked
- :disabled

```
a{
  background-color:Orange;

  transition-property:background-color;
  transition-duration:1s;
  transition-delay:500ms;
  transition-timing-function:ease-out;
}
a:hover{
  background-color:Red;
}
```

Raccourci

```
a{
  background-color: Orange;
  transition: all 1s ease-out 500ms;
}
a:hover{
  background-color:Red;
}
```

Propriété, durée, délai, easing

« all » pour animer toutes les propriétés ou indiquer seulement le nom des propriétés désirées

Avec préfixes pour les différents navigateurs

```
a{
  background-color: Orange;
  -webkit-transition: all 1s ease-out 500ms;
  -o-transition: all 1s ease-out 500ms;
  transition: all 1s ease-out 500ms;
}
a:hover{
  background-color:Red;
}
```

Définir plusieurs propriétés

```
a{
  background-color: Orange;
  color:white;
  text-decoration: none;
  -webkit-transition: background-color 1s ease-out .5s, color 2s;
  -o-transition: background-color 1s ease-out .5s, color 2s;
  transition: background-color 1s ease-out .5s, color 2s;
}
a:hover{
  background-color:Red;
  color:dodgerblue;
  text-decoration: none;
}
```

On sépare chaque transition par une virgule

Panneau Transitions de Dreamweaver

Nouvelle transition

Règle cible : a

Transition sur : hover

Même règle pour toutes les propriétés

Durée : 1 s

Retard : 500 ms

Fonction de minutage : ease-out

Propriété : background-color

Valeur de fin : Red

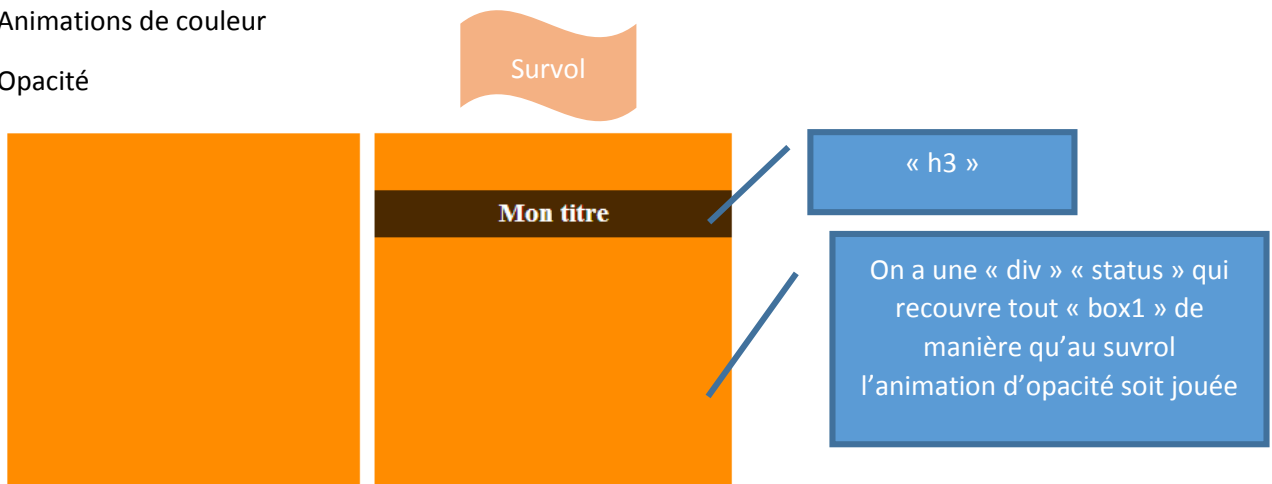
Choisissez où créer la transition : (Ce document uniquement)

Aide Annuler Créer une transition

c. Exemples

Animations de couleur

Opacité



HTML

```
<div id="box1">
  <div class="status">
    <h3>Mon titre</h3>
  </div>
</div>
```

CSS

```
#box1 {
  display: block;
  position: relative;
  width: 250px;
  height: 250px;
  background: darkorange;
  margin: 0 auto;
}
.status {
  position: absolute;
  top: 0;
  height: 100%;
  width: 100%;
  opacity: 0;

  -webkit-transition: opacity 1s ease-in;
  -o-transition: opacity 1s ease-in;
  transition: opacity 1s ease-in;
}
.status h3 {
  margin-top: 40px;
  padding: 5px 0;
  color: white;
  text-align: center;
  background-color: rgba(0, 0, 0, 0.70);
}
.status:hover {
  opacity: .8;
}
```

2. Animations avec keyframes

```
@keyframes myAnimation {
  0%{background-color: orange;}
  50%{background-color: orangered;}
  100%{background-color: red;}
}
```

Avec préfixes

```
@-webkit-keyframes myAnimation {
  0%{background-color: orange;}
  50%{background-color: orangered;}
  100%{background-color: red;}
}
@-o-keyframes myAnimation {
  0%{background-color: orange;}
  50%{background-color: orangered;}
  100%{background-color: red;}
}
@keyframes myAnimation {
  0%{background-color: orange;}
  50%{background-color: orangered;}
  100%{background-color: red;}
}
```

Utilisation

Propriétés :

- « animation-name »
- « animation-duration » durée en s ou ms
- « animation-iteration-count »
- « animation-direction » (normal ou alternate)
- « animation-timing-function »
- « animation-delay » en secondes ou ms (exemple 5s, 500ms)

```
body{
  animation: myAnimation 10s infinite;
}
```

Avec préfixes

```
body{
  -webkit-animation: myAnimation 10s infinite;
  -o-animation: myAnimation 10s infinite;
  animation: myAnimation 10s infinite; /*
}
```

Forme « abrégée » Nom, duration, repeat, direction



Autre exemple

```
#box1{
  height: 250px;
  width: 250px;
  background-color: dodgerblue;
}
#box1:hover{
  animation:myAnimation 2s; /* infinite normal; */
}
@keyframes myAnimation {
  to {
    background-color:darkorange;
  }
  /* from {
    background-color:orange;
  } */
}
```

3. Transformations

On emploie « transform : method »

a. Déplacer un élément (translate ou translateX, translateY)

« translate » ou translateX / translateY si on ne veut modifier qu'une valeur

```
img {
  display: block;
  position: relative;
  width: 560px;
  margin: 30px auto;
}
img:hover {
  -o-transform: translate(50px, 200px);
  -webkit-transform: translate(50px, 200px);
  transform: translate(50px, 200px);
}
```

Déplacement d'une image au survol de 50px vers la droite et 200px vers le bas. Pour un déplacement vers la gauche ou le haut, utiliser des valeurs négatives.

b. Redimensionner un élément (scale ou scaleX / scaleY)

HTML

```
<div id="container">
  
</div>
```

CSS

```
#container {
  position: relative;
  width: 560px;
  margin: 0 auto;
}

img {
  margin: 20px;
  -webkit-transition: all .5s ease;
  -o-transition: all .5s ease;
  transition: all .5s ease;
}

img:hover {
  -webkit-transform: scale(1.5);
  -o-transform: scale(1.5);
  transform: scale(1.5);
}
```

Grossit l'image au survol de 2 fois sa largeur et 2 fois hauteur. Pour réduire, indiquer des valeurs en dessous 1

Transition

Etat de fin de l'image

Au survol l'image « grossit »



c. Tourner un élément (rotate)

```
img {
  display: block;
  position: relative;
  width: 560px;
  margin: 30px auto;
}
img:hover {
  -o-transform: rotate(90deg);
  -webkit-transform: rotate(90deg);
  transform: rotate(90deg);
}
```



L'image pivote de 90 degrés vers la droite

d. « Incliner » un élément (skew ou skewX / skewY)

```
img {
  display: block;
  position: relative;
  width: 560px;
  margin: 30px auto;
}
img:hover {
  -o-transform: skew(20deg, 10deg);
  -webkit-transform: skew(20deg, 10deg);
  transform: skew(20deg, 10deg);
}
```



e. Transform-origin

On peut changer le point d'origine de la transformation :

- transform-origin :50% 50%
- transform-origin :50px 0
- transform-origin :left bottom

4. Transformations 3D

Transformation sur 3 dimensions : X, Y et Z

« transform-style » : preserve-3d ou flat

a. Déplacement (translate3d)

HTML

```
<div class="container">
  
</div>
```

CSS

```
.container {
  position: relative;
  width: 500px;
  margin: 30px auto;

  -webkit-perspective: 500px;
  perspective: 500px;
}

.container img {
  position: absolute;

  transform-style: preserve-3d;
  -webkit-transition: transform 2s;
  transition: transform 2s;
}

.container img:hover {
  transform: translateX(100px) translateY(100px) translateZ(-150px);
  -webkit-transform: translateX(100px) translateY(100px) translateZ(-150px);
  -o-transform: translateX(100px) translateY(100px) translateZ(-150px);
}
```

« Perspective » sur le conteneur
(loin :1500px, moyen : 500px, prêt :500px)

Animation sur l'image

Etat de fin au survol de l'image

Ou raccourci

```
.container img:hover {
  transform: translate3d(100px, 100px, -150px);
  -webkit-transform: translate3d(100px, 100px, -150px);
  -o-transform: translate3d(100px, 100px, -150px);
}
```

L'image se déplace vers la droite, le bas et donne l'impression de s'éloigner.



b. Rotation (rotate3d)

HTML

```
<div class="container">
  
</div>
```

CSS

```
.container {
  position: relative;
  width: 500px;
  margin: 30px auto;

  -webkit-perspective: 500px;
  perspective: 500px;
}

.container img {
  position: absolute;

  transform-style: preserve-3d;
  -webkit-transition: transform 2s;
  transition: transform 2s;
}

.container img:hover {
  transform: rotateY(180deg);
  -webkit-transform: rotateY(180deg);
  -o-transform: rotateY(180deg);
}
```

Rotation seulement sur « y »

Au survol l'image pivote



```
.container img:hover {
  transform: rotateX(180deg) rotateY(180deg) rotateZ(180deg);
  -webkit-transform: rotateX(180deg) rotateY(180deg) rotateZ(180deg);
  -o-transform: rotateX(180deg) rotateY(180deg) rotateZ(180deg);
}
```

Ou

```
.container img:hover {
  transform: rotate3d(1, 1, 1, 360deg);
  -webkit-transform: rotate3d(1, 1, 1, 360deg);
  -o-transform: rotate3d(1, 1, 1, 360deg);
}
```

c. Scale (sacle3d)

HTML

```
<div class="container">
  
</div>
```

CSS

```
.container {
  position: relative;
  width: 500px;
  margin: 30px auto;

  -webkit-perspective: 500px;
  perspective: 500px;
}

.container img {
  position: absolute;

  transform-style: preserve-3d;
  -webkit-transition: transform 2s;
  transition: transform 2s;
}

.container img:hover {
  transform: scaleX(1.5) scaleY(1.5) scaleZ(1.5);
  -webkit-transform: scaleX(1.5) scaleY(1.5) scaleZ(1.5);
  -o-transform: scaleX(1.5) scaleY(1.5) scaleZ(1.5);
}
```

Ou raccourci

```
.container img:hover {
  transform: scale3d(1.5, 1.5, 1.5);
  -webkit-transform: scale3d(1.5, 1.5, 1.5);
  -o-transform: scale3d(1.5, 1.5, 1.5);
}
```

Sur une seule dimension l'image donne l'impression de s'étirer

d. Perspective-origin

On peut changer le point d'origine de la perspective :

- perspective-origin :50% 50% (défaut)
- perspective-origin :50% 100% (centré en bas)
- etc.

e. Exemples

Flip



HTML

```
<div class="container">
  <div class="picture">
    <div class="front">
      
    </div>
    <div class="back">
      <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>
    </div>
  </div>
</div>
```

CSS

```
.container{
  position: relative;
  margin: 50px auto;
  width: 250px;
  height: 250px;
  perspective: 500px;
}
.picture{
  width: 100%;
  height: 100%;
  transform-style: preserve-3d;
  transition: all 1s linear;
}
.front{
  position: absolute;
  width: 100%;
  height: 100%;
  z-index: 2;
  backface-visibility: hidden;
  transform: rotateY(0deg);
}
.back{
  position: absolute;
  width: 100%;
  height: 100%;
  padding: 10px;
  text-align: center;
  color: white;
  background-color: dodgerblue;
  backface-visibility: hidden;
  transform: rotateY(180deg);
}
.container:hover .picture{
  transform: rotateY(180deg);
}
```

Texte placé « derrière »

Rotation sur Y par exemple

Rotation sur X

```
.container .back{  
  transform: rotateZ(180deg) rotateY(180deg);  
}  
.container:hover .picture{  
  transform: rotateX(180deg);  
}
```

On tourne en plus le texte de manière à ce qu'il ne soit pas « inversé »



Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Suspendisse nec mattis odio.

Lorem ipsum dolor sit amet,
consectetur adipiscing elit.
Suspendisse nec mattis odio.

Création d'un cube 3d

HTML

```

<div class="container">
  <div class="cube_3d">
    <div class="pane front"><figure>Face</figure></div>
    <div class="pane back"><figure>Derrière</figure></div>
    <div class="pane top"><figure>Haut</figure></div>
    <div class="pane bottom"><figure>Bas</figure></div>
    <div class="pane left"><figure>Gauche</figure></div>
    <div class="pane right"><figure>Droite</figure></div>
  </div>
</div>

```

CSS

```

.container{
  width: 250px;
  height: 250px;
  perspective: 500px;
  margin: 50px auto;
}
.cube_3d{
  position: relative;
  width: 100%;
  height: 100%;
  transform-style: preserve-3d;
  transition: all 1s ease-in-out;
}
.pane{
  position: absolute;
  width: 250px;
  height: 250px;
}
.front{
  background: rgba(0,0,0,.6);
  z-index: 2;
  transform: translate3d(0,0,125px);
}
.back{
  background: darkslategray;
  transform: rotateY(180deg) translate3d(0,0,125px);
}
.top{
  background: blue;
  transform: rotateX(90deg) translate3d(0,0,125px);
}
.bottom{
  background: lightblue;
  transform: rotateX(-90deg) translate3d(0,0,125px);
}
.left{
  background: orangered;
  transform: rotateY(-90deg) translate3d(0,0,125px);
}
.right{
  background: darkred;
  transform: rotateY(90deg) translate3d(0,0,125px);
}

```

Animer le cube

HTML

Ajout de boutons radios au-dessus du conteneur du cube

```
<input type="radio" id="front" name="grp1" checked /><label>Front</label>
<input type="radio" id="back" name="grp1" /><label>Back</label>
<input type="radio" id="left" name="grp1" /><label>Left</label>
<input type="radio" id="right" name="grp1" /><label>Right</label>
<input type="radio" id="top" name="grp1" /><label>Top</label>
<input type="radio" id="bottom" name="grp1" /><label>Bottom</label>

<input type="radio" id="showbackface" name="grp2" checked /><label>Backface</label>
<input type="radio" id="hidebackface" name="grp2" /><label>Caché</label>
```

CSS

```
#front:checked ~ .container .cube_3d{
  transform: rotateY(0deg);
}
#back:checked ~ .container .cube_3d{
  transform: rotateY(180deg);
}
#top:checked ~ .container .cube_3d{
  transform: rotateX(-90deg);
}
#bottom:checked ~ .container .cube_3d{
  transform: rotateX(90deg);
}
#left:checked ~ .container .cube_3d{
  transform: rotateY(90deg);
}
#right:checked ~ .container .cube_3d{
  transform: rotateY(-90deg);
}

#showbackface:checked ~ .container .cube_3d .pane{
  backface-visibility: visible;
}
#hidebackface:checked ~ .container .cube_3d .pane{
  backface-visibility: hidden;
}
```

5. Ressources

<https://github.com/desandro/3dtransforms>