

4. LES TRANSITIONS EN CSS3

Les CSS3 permettent, avec le module Transition de **passer d'une valeur CSS à une autre avec une transition spécifiée si un événement est détecté au niveau d'un élément**.

Concrètement, le principe de base d'une transition CSS 3 est de permettre un passage en douceur de l'ancienne vers la nouvelle valeur d'une propriété CSS lorsqu'un événement est déclenché. Lorsque la transition est terminée, l'élément reprend ses paramètres CSS initiaux.

Les préfixes propriétaires des navigateurs ne sont plus requis aujourd'hui.

Lecture Web : Can I use <http://www.caniuse.com/>

Récapitulatif des dernières technologies et de leur prise en charge sur les générations de navigateurs.

1. Mettre en place les transitions

1. Le déclencheur

Pour déclencher une transition, il faut qu'un événement soit détecté avec, par exemple, une pseudo-classe `:hover`, `:active` ou `:focus`.

2. La propriété qui est affectée

Vous devez indiquer quelle(s) propriété(s) vous souhaitez utiliser avec la propriété **transition-property**.

Ex : **transition-property : all** (valeur par défaut), toutes les propriétés possibles seront animées
transition-property : propriété (le nom d'une propriété pouvant être animée)

Nom de la propriété	Type
background-color	<i>color</i>
background-image	<i>only gradients</i>
background-position	<i>percentage, length</i>
border-bottom-color	<i>color</i>
border-bottom-width	<i>length</i>
border-color	<i>color</i>
border-left-color	<i>color</i>
border-left-width	<i>length</i>
border-right-color	<i>color</i>
border-right-width	<i>length</i>
border-spacing	<i>length</i>
border-top-color	<i>color</i>
border-top-width	<i>length</i>
border-width	<i>length</i>
bottom	<i>length, percentage</i>
color	<i>color</i>

crop	<i>rectangle</i>
font-size	<i>length, percentage</i>
font-weight	<i>number</i>
grid-*	<i>various</i>
height	<i>length, percentage</i>
left	<i>length, percentage</i>
letter-spacing	<i>length</i>
line-height	<i>number, length, percentage</i>
margin-bottom	<i>length</i>
margin-left	<i>length</i>
margin-right	<i>length</i>
margin-top	<i>length</i>
max-height	<i>length, percentage</i>
max-width	<i>length, percentage</i>
min-height	<i>length, percentage</i>
min-width	<i>length, percentage</i>
opacity	<i>number</i>
outline-color	<i>color</i>
outline-offset	<i>integer</i>
outline-width	<i>length</i>
padding-bottom	<i>length</i>
padding-left	<i>length</i>
padding-right	<i>length</i>
padding-top	<i>length</i>
right	<i>length, percentage</i>
text-indent	<i>length, percentage</i>
text-shadow	<i>shadow</i>
top	<i>length, percentage</i>
vertical-align	<i>keywords, length, percentage</i>
visibility	<i>visibility</i>
width	<i>length, percentage</i>
word-spacing	<i>length, percentage</i>
z-index	<i>integer</i>
zoom	<i>number</i>

3. La durée

Vous indiquez ensuite la durée de cette transition avec la propriété **transition-duration**. Les deux unités de temps acceptées sont **s** : la seconde et **ms** : la milliseconde.

4. L'effet

La méthode d'interpolation (**transition-timing-function**) : **ease**, **linear**, **ease-in**, **ease-out**, **ease-in-out**.

Certaines animations sont plus rapides sur le début et ralentissent sur la fin, d'autres sont constantes dans leur vitesse (linéaires).

- **ease** : la transition est lente au début, puis accélère vers la fin (c'est la valeur par défaut).
- **linear** : la transition se fait à vitesse constante.
- **ease-in** : la transition se fait lentement au début, puis accélère vers la fin.
- **ease-out** : la transition se fait rapidement au début, puis ralentit vers la fin.
- **ease-in-out** : la transition se fait lentement au début, puis accélère, puis ralentit vers la fin.
- **cubic-bezier** : cela permet de paramétrer vous-même la courbe de Bézier pour la cinétique.

Les courbes de Bézier utilisées pour ces effets de cinétique : <http://www.felix-girault.fr/css/les-transitions-css3/>

5. Le décalage

Le temps en seconde avant que la transition ne démarre (**transition-delay**), combien de temps après la détection de l'événement.

2. Exemple

```
img {  
width: 50px; /* état par défaut */  
transition-property: width; /* nous indiquons quelle est la propriété CSS à  
modifier */  
transition-duration: 1s; /* durée de la transition */  
}  
  
img:hover {  
width: 200px;  
/* c'est au survol (:hover) de la souris que va se déclencher la transition.  
L'agrandissement va se faire jusqu'à la position horizontale de 40 pixels */  
}
```

3. La syntaxe courte

Vous pouvez utiliser la syntaxe courte transition en spécifiant ensuite les fonctions séparées par un simple espace, dans cet ordre :

1. **transition-property**
2. **transition-duration**
3. **transition-timing-function**
4. **transition-delay**

Exemple de syntaxe courte pour un déplacement vers la gauche :

```
#deplacement {  
position: absolute;  
left: 20px;  
top: 20px;
```

```
transition: left 2s ease-in 2s;
}
```

4. Les transitions multiples

Vous pouvez effectuer plusieurs transitions, il suffit de séparer chaque transition par une virgule et d'indiquer un délai pour que chaque transition débute en temps voulu.

```
#multiple {
  width: 200px;
  height: 60px;
  background-color: darkblue;
  transition: width 3s linear, height 2s linear 3s, opacity
4s linear 6s;
}
/*
Transition multiple avec trois propriétés modifiées :
• la largeur width, qui dure 3 secondes.
• la hauteur height, qui dure 2 secondes et débute 3 secondes plus tard.
• l'opacité opacity, qui dure 4 secondes et débute 6 secondes plus tard.
*/

#multiple:hover {
  width: 400px;
  height: 120px;
  opacity: .3;
}
```

5. Mouseover et Mouseout sans JavaScript

Nous mettons en place, à priori, la transition sur l'élément initial (et pas sur le hover, active ou focus de celui-ci). Sans nouvelle déclaration de cette propriété transition dans l'état hover, la seule existante s'effectuera de la même manière pour l'état hover et lors du retour à l'état initial.

Pour **différencier l'animation lorsque la souris survole et quitte l'élément** on ajoute une transition sur l'événement hover, cette dernière s'appliquera lors du survol et lorsque le pointeur quitte l'élément, c'est la transition initiale qui s'applique.

```
/*Effet au « Mouseout »*/
.bouton {
  position: absolute;
  top: 400px;
  left: 200px;
  transition: all 0.5s ease-in-out 1s;
}

/*Effet au « Mouseover »*/
.base:hover .bouton {
  position: absolute;
  top: 200px;
  transition: all 5.5s ease 1s;
}
```

6. Les générateurs en ligne

1. CSS3 Generator

En choisissant Transition dans la liste déroulante, CSS3 Generator (<http://css3generator.com/>) vous permet de travailler sur cinq propriétés (*Property*). Vous pouvez spécifier la durée (*Duration*) et la cinétique (*Function*).

2. CSS 3.0 Maker

Avec CSS 3.0 Maker (<http://www.css3maker.com/css3-transition.html>), le travail des transitions ne se fait que sur quelques propriétés. Vous pouvez définir l'événement (*Select Action*) la durée (*Transition Duration*) et la cinétique (*Transition Timing*).

3. Créer vos courbes de Bézier

La propriété `transition-timing-function` permet de créer ses propres courbes de Bézier avec le mot-clé `cubic-bezier` :

```
transition: all 3s cubic-bezier(.98,0,1,.28);
```

Pour vous aider à créer visuellement vos courbes de Bézier : <http://cubic-bezier.com/>

Vous avez sur la gauche un graphique Bézier dans lequel vous pouvez manipuler les deux points directeurs. La « formule » `cubic-bezier` se crée dynamiquement en haut de l'écran. Vous disposez aussi d'un comparateur de cinétique et des courbes prédéfinies.

Lecture Web :

<http://www.alsacreations.com/tuto/lire/874-transitions-css3-proprietes.html>

<http://www.alsacreations.com/tuto/lire/876-transitions-css3-transition-timing-function.html>