


CSS3 - transition

- Support par les navigateurs
- Principe des transitions
- Propriétés CSS3 associées
- Ordre et sens des transitions
- Propriété raccourcie
- Transition pilotée par JavaScript











Support par les navigateurs

CSS3 Transitions - WD

Simple method of animating certain properties of an element, with ability to define property, duration, delay and timing function.

Current aligned Usage relative Date relative Filtered All 

Usage % of all users  ?
Global 98.07% + 0.01% = 98.08%
unprefixed: 98.01%

Chrome	Edge *	Safari	Firefox	Opera	IE	Chrome for Android	Safari on iOS *	Samsung Internet	Opera Mini *	Opera Mobile *	UC Browser for Android	Android Browser *	Firefox for Android	QQ Browser	Baidu Browser	KaiOS Browser
			2-3.6													
		1 3.1-5 	1 4 	10.1			1 3.2-5.1 									
4-25 		5.1-6 	5-15 	11.5 	6-9		6.1 			12 		2.1-4.3 				
26-118	12-118	6.1-17.0	16-119	12.1-103	10		7-17.0	4-22		12.1		4.4-4.4.4				2.5
119	119	17.1	120	104	11	119	17.1	23	all	73	15.5	119	119	13.1	13.18	3.1
120-122		17.2-TP	121-123				17.2									

Les propriétés CSS3 liées aux transitions sont supportées depuis de nombreuses versions

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

Principe des transitions

- Une **transition** permet de **passer** d'un **état de départ** à un **état d'arrivée** en un certain temps
- Par exemple, le passage de la souris (**:hover**)
- On définit les propriétés CSS d'un élément et au survol, on peut **changer certaines propriétés**
- Au lieu que le **changement** soit abrupt, il sera **progressif** en jouant sur plusieurs paramètres

Exemple

transition01.html

```
1 <!DOCTYPE html>
2 <html lang="fr">
3 <head>
4   <meta charset="UTF-8">
5   <title>Transitions</title>
6   <link type="text/css" rel="stylesheet"
7     href="transition.css" />
8 </head>
9 <body>
10   <h1>Principe des transitions</h1>
11   <div id="element">
12     <p>Survolez-moi !</p>
13   </div>
14 </body>
15 </html>
```

transition.css

```
1 body {
2   background-color: bisque;
3 }
4
5 #element {
6   width: 300px;
7   height: 300px;
8   background-color: aqua;
9   font-size: 24px;
10  font-weight: bold;
11  text-align: center;
12  line-height: 300px;
13  border: 3px solid black;
14  border-radius: 20px;
15 }
16
17 #element p {
18   margin: 0;
19   padding: 0;
20 }
21
22 #element:hover {
23   border-radius: 50%;
24 }
```

Propriétés des transitions

- Il y a principalement quatre propriétés liées aux transitions :
 - **transition-duration**
 - **transition-delay**
 - **transition-property**
 - **transition-timing-function**
- Et enfin une propriété raccourcie qui regroupe les précédentes : **transition**
(dans l'ordre : transition-property transition-duration transition-timing-function transition-delay séparés par un espace)

transition-duration

- Cette propriété permet de définir la **durée** de la transition d'un état à l'autre
- On donne une valeur exprimée en **secondes** (ex : 2s) ou en **millisecondes** (ex : 500ms) pour définir la durée.
La valeur **par défaut** étant de **0s**
- **Par exemple :**
On ajoute dans `#element {...}`
`transition-duration: 2s;`

transition-delay

- Cette propriété permet de définir le **délai** à attendre **avant que la transition** d'un état à l'autre ne **démarre** réellement
- On donne une valeur exprimée en **secondes** (ex : 2s) ou en **millisecondes** (ex : 500ms) pour définir le délai.
La valeur **par défaut** étant de **0s**
- **Par exemple :**
On ajoute dans `#element {...}`
`transition-delay: 500ms;`

transition-property

- Pendant une transition, on peut **modifier zéro, une ou plusieurs propriétés** en même temps
- Cette propriété permet de définir **quelle(s) propriété(s) change(nt)** pendant la transition
- On donne la liste des propriétés (séparées par une virgule) qui changent pendant la transition. La valeur **par défaut** étant **all**
- **Par exemple :**
On ajoute dans `#element {...}`
`transition-property: border-radius, background-color;`

Modifier plusieurs propriétés

- Supposons que dans `#element: hover {...}`, plusieurs propriétés ont été modifiées
- Celles qui sont renseignées dans `property-transition` changeront progressivement pendant la transition, les autres le feront instantanément (exemple 1)
- On peut paramétrer différemment les propriétés impliquées dans la transition (exemple 2)

Exemple 1

```
1 ▼ body {
2     background-color: bisque;
3 }
4
5 ▼ #element {
6     width: 300px;
7     height: 300px;
8     background-color: aqua;
9     font-size: 24px;
10    font-weight: bold;
11    text-align: center;
12    line-height: 300px;
13    border: 3px solid black;
14    border-radius: 20px;
15
16    /* transition */
17    transition-duration: 2s;
18    transition-delay: 500ms;
19    transition-property: border-radius;
20 }
21
22 ▼ #element p {
23     margin: 0;
24     padding: 0;
25 }
26
27 ▼ #element:hover {
28     border-radius: 50%;
29     background-color: coral;
30 }
```

- 2 propriétés sont modifiées pendant le passage d'un état à l'autre :
 - border-radius
 - background-color
- Seulement une va changer progressivement pendant la transition :
 - border-radius
 - durée : 2 secondes et délai : 500 millisecondes

Exemple 2

```
1 ▼ body {
2     background-color: bisque;
3 }
4
5 ▼ #element {
6     width: 300px;
7     height: 300px;
8     background-color: aqua;
9     font-size: 24px;
10    font-weight: bold;
11    text-align: center;
12    line-height: 300px;
13    border: 3px solid black;
14    border-radius: 20px;
15
16    /* transition */
17    transition-duration: 2s, 1s;
18    transition-delay: 0s, 2s;
19    transition-property: border-radius, background-color;
20 }
21
22 ▼ #element p {
23     margin: 0;
24     padding: 0;
25 }
26
27 ▼ #element:hover {
28     border-radius: 50%;
29     background-color: coral;
30     border-color: blue;
31 }
```

- dans #element:hover, trois propriétés vont être modifiées
 - **border-radius** démarre avec un délai de 0s et dure 2s
 - **background-color** démarre après 2s et dure 1s
 - **border-color** change instantanément

transition-timing-function

- Cette propriété permet de définir une **courbe de vitesse de progression** pendant la transition (sous forme d'une courbe de Bézier)
- Les valeurs possibles sont : ease, linear, ease-in, ease-out, ease-in-out, cubic-bezier(n,n,n,n)
La valeur **par défaut** étant **ease**
- Cela donne la possibilité de régler la vitesse, l'accélération et le ralentissement de la transition

transition-timing-function



ease



linear



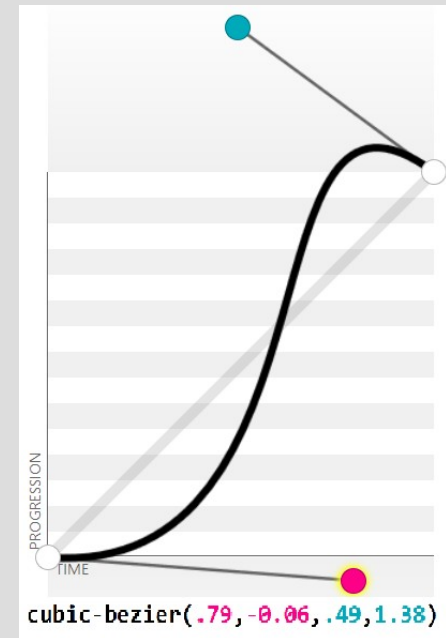
ease-in



ease-out



ease-in-out





Courbe de Bézier

Pour définir les paramètres de la courbe de Bézier
<https://cubic-bezier.com>

cubic-bezier(.17, .67, .83, .67) COPY ▾


Preview & compare GO!

Duration: 1 second




Library IMPORT EXPORT


Click on a curve to compare it with the current one.




ease




linear



ease-in



ease-out



ease-in-out

Tip: Right click on any library curve and select "Copy Link Address" to get a permalink to it which you can share with others

Sens des transitions

- Les transitions ne **doivent pas être** forcément **symétriques** (de l'état A vers B ou de B vers A)
- Si on définit dans un sens, la propriété **transition-property:none**; alors le changement est instantané car il n'y a pas de transition (exemple 3)
- On peut aussi définir des **transitions différentes** en modifiant les paramètres **dans un sens ou dans l'autre** (exemple 4)

Exemple 3

```
/* #element {
    width: 300px;
    height: 300px;
    background-color: aqua;
    font-size: 24px;
    font-weight: bold;
    text-align: center;
    line-height: 300px;
    border: 3px solid black;
    border-radius: 20px;

    /* transition (vers l'état normal) : d'abord la forme ensuite la couleur */
    transition-duration: 2s, 1s;
    transition-delay: 0s, 2s;
    transition-property: border-radius, background-color;
}

/* #element p {
    margin: 0;
    padding: 0;
}

/* #element:hover {
    border-radius: 50%;
    background-color: coral;
    border-color: blue;

    /* transition (vers l'état survolé) : pas de transition */
    transition-property: none;
} */
```


Exemple 4

```
▼ #element {
    width: 300px;
    height: 300px;
    background-color: aqua;
    font-size: 24px;
    font-weight: bold;
    text-align: center;
    line-height: 300px;
    border: 3px solid black;
    border-radius: 20px;

    /* transition (vers l'état normal) : d'abord la forme ensuite la couleur */
    transition-duration: 2s, 1s;
    transition-delay: 0s, 2s;
    transition-property: border-radius, background-color;
}

▼ #element p {
    margin: 0;
    padding: 0;
}

▼ #element:hover {
    border-radius: 50%;
    background-color: coral;
    border-color: blue;

    /* transition (vers l'état survolé) : d'abord la couleur ensuite la forme */
    transition-duration: 2s, 1s;
    transition-delay: 1s, 0s; /* modification des délais pour changer l'ordre */
    transition-property: border-radius, background-color;
}
```

transition

- Il y a une *propriété raccourcie* **transition** qui regroupe les 4 propriétés séparées, par exemple:
transition-duration: 2s, 1s;
transition-property: *border-radius*, background-color;
transition-timing-function: ease, linear;
transition-delay: 0s, 2s;
- Cela devient
transition: *border-radius* 2s ease 0s, background-color 1s linear 2s ;

Dans l'ordre: [property duration timing-function delay]

Transitions et JavaScript

- Pour provoquer une transition, il faut :
 - un état initial
 - un état final
 - un déclencheur pour passer de l'un à l'autre
- Si on ne veut pas réagir au survol avec la souris (:hover), on utilise JavaScript pour déclencher sur un autre événement (Exemple 5)

Exemple 5 - HTML

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <title>Transitions</title>
  <link type="text/css" rel="stylesheet" href="transition.css" />
</head>
<body>
  <h1>Courbe de Bézier</h1>
  <h2>transition-timing-function</h2>
  <div id="course">
    <p id="ease" class="barre">ease</p>
    <p id="linear" class="barre">linear</p>
    <p id="ease_in" class="barre">ease-in</p>
    <p id="ease_out" class="barre">ease-out</p>
    <p id="ease_in_out" class="barre">ease-in-out</p>
    <p id="bezier" class="barre">cubic-bezier</p>
    <em>cubic-bezier(.64,-0.31,.49,1.64)</em>
  </div>
  <br />
  <button type="button" onclick="toggle();">A vos marques, Prêt ?</button>

  <script src="toggle.js">
  </script>
</body>
</html>
```

Exemple 5 - CSS

```
.barre {  
    width:100px;  
    height:50px;  
    line-height: 50px;  
    background-color: blue;  
    color: white;  
    padding-left: 10px;  
}  
  
.barre.go {  
    width:1000px;  
}  
  
#ease {  
    transition: 2s width ease 1s;  
}  
  
#linear {  
    transition: 2s width linear 1s;  
}  
  
#ease_in {  
    transition: 2s width ease-in 1s;  
}  
  
#ease_out {  
    transition: 2s width ease-out 1s;  
}  
  
#ease_in_out {  
    transition: 2s width ease-in-out 1s;  
}  
  
#bezier {  
    transition: 2s width cubic-bezier(.64,-0.31,.49,1.64) 1s;  
}
```

Exemple 5 - JS

```
/* Cette fonction va servir de déclencheur à notre transition */
/* L'état initial correspond à une classe "barre" */
/* L'état final correspond aux classes "barre go" */
/* */
/* Chaque fois qu'on appuie sur le bouton, on bascule d'un */
/* état à l'autre, provoquant le lancement de la transition */

function toggle() {
    "use strict";

    /* Mettre dans un tableau x, tous les éléments p qui sont dans l'élément avec id = course */
    var x = document.querySelectorAll("#course p");

    /* Pour chaque élément du tableau x, tester son attribut class */
    x.forEach(elem => {
        /* S'il a uniquement la classe "barre" */
        if (elem.className === "barre") {
            /* Alors on va ajouter la classe "go" => pour passer à "barre go" */
            elem.className += " go";
        } else {
            /* Sinon ne pas mettre la classe "go" => pour passer à "barre" */
            elem.className = "barre";
        }
    });
}
```

