

CSS transition-timing-function Property

Example

A transition effect with the same speed from start to end:

```
div {  
  transition-timing-function: linear;  
}
```

Definition and Usage

The `transition-timing-function` property specifies the speed curve of the transition effect.

This property allows a transition effect to change speed over its duration.

Default value:	ease
Inherited:	no
Animatable:	no. Read about animatable
Version:	CSS3
JavaScript syntax:	<code>object.style.transitionTimingFunction="linear"</code>

Browser Support

The numbers in the table specify the first browser version that fully supports the property.

Numbers followed by `-webkit-`, `-moz-` or `-o-` specify the first version that worked with a prefix.

Property					
transition-timing-function	26.0 4.0 -webkit-	10.0	16.0 4.0 -moz-	6.1 3.1 -webkit-	12.1 10.5 -o-

CSS Syntax

```
transition-timing-function: linear|ease|ease-in|ease-out|ease-in-out|step-start|step-end|steps(int,start|end)|cubic-bezier(n,n,n,n)|initial|inherit;
```

Property Values

Value	Description
ease	Default value. Specifies a transition effect with a slow start, then fast, then end slowly (equivalent to cubic-bezier(0.25,0.1,0.25,1))
linear	Specifies a transition effect with the same speed from start to end (equivalent to cubic-bezier(0,0,1,1))
ease-in	Specifies a transition effect with a slow start (equivalent to cubic-bezier(0.42,0,1,1))
ease-out	Specifies a transition effect with a slow end (equivalent to cubic-bezier(0,0,0.58,1))
ease-in-out	Specifies a transition effect with a slow start and end (equivalent to cubic-bezier(0.42,0,0.58,1))
step-start	Equivalent to steps(1, start)
step-end	Equivalent to steps(1, end)
steps(int,start end)	Specifies a stepping function, with two parameters. The first parameter specifies the number of intervals in the function.

It must be a positive integer (greater than 0). The second parameter, which is optional, is either the value "start" or "end", and specifies the point at which the change of values occur within the interval. If the second parameter is omitted, it is given the value "end"

cubic-bezier(*n,n,n,n*).

Define your own values in the cubic-bezier function. Possible values are numeric values from 0 to 1

initial

Sets this property to its default value. [Read about *initial*](#)

inherit

Inherits this property from its parent element. [Read about *inherit*](#)

Tip: Try the different values in the examples below to understand how it works!

More Examples

Example

To better understand the different function values: Here are five different div elements with five different values:

```
#div1 {transition-timing-function: linear;}
#div2 {transition-timing-function: ease;}
#div3 {transition-timing-function: ease-in;}
#div4 {transition-timing-function: ease-out;}
#div5 {transition-timing-function: ease-in-out;}
```

Example

Same as the example above, but the speed curves are specified with the cubic-bezier function:

```
#div1 {transition-timing-function: cubic-bezier(0,0,1,1);}
#div2 {transition-timing-function: cubic-bezier(0.25,0.1,0.25,1);}
#div3 {transition-timing-function: cubic-bezier(0.42,0,1,1);}
```

```
#div4 {transition-timing-function: cubic-bezier(0,0,0.58,1);}
#div5 {transition-timing-function: cubic-bezier(0.42,0,0.58,1);}
```
