

This site uses cookies to improve the user experience.

jQuery

1. [jQuery Tutorial](#)
2. [jQuery Overview](#)
3. [jQuery Installation](#)
4. [jQuery \\$\(document\).ready\(\)](#)
5. [jQuery Selectors](#)
6. [jQuery Traversal](#)
7. [jQuery & CSS](#)
8. [jQuery & DOM](#)
9. [jQuery & Events](#)
10. **[jQuery Effects](#)**
11. [jQuery AJAX](#)
12. [jQuery Deferred Objects](#)
13. [jQuery Plugins](#)
14. [jQuery - Generating a Table of Contents](#)
15. [jQuery - Creating an Expandable Tree](#)
16. [jQuery Critique](#)

# jQuery Effects

- [show\(\)](#)
- [hide\(\)](#)
- [toggle\(\)](#)
- [slideDown\(\)](#)
- [slideUp\(\)](#)
- [slideToggle\(\)](#)
- [fadeIn\(\)](#)
- [fadeOut\(\)](#)
- [fadeToggle\(\)](#)
- [fadeTo\(\)](#)
- [animate\(\)](#)
  - [Camel Case](#)
  - [Toggle Animation](#)
  - [Stopping Animation](#)

Last update: 2014-10-04

jQuery enables you to apply certain standard effects to HTML elements in your page. Each effect carried out by calling a special effect function in JQuery. The most common of these effect functi

- [show\(\)](#)
- [hide\(\)](#)
- [toggle\(\)](#)
- [slideDown\(\)](#)
- [slideUp\(\)](#)
- [slideToggle\(\)](#)
- [fadeIn\(\)](#)
- [fadeOut\(\)](#)
- [fadeToggle\(\)](#)
- [fadeTo\(\)](#)
- [animate\(\)](#)

Here are a few examples. Click the buttons to see the results. Start with the "Show" button.

ShowHideToggle

Slide DownSlide UpSlide Toggle

Fade InFade OutFade ToggleFade To 0.5Fade To 1.0

Animate 1Animate 2Animate 3Animate Toggle

## show()

The `show()` method can show a hidden element, for instance a `div` or an `img` element. Here is a sir example:

```
$('#theImg').show();
```

An HTML element can be hidden via the `hide()` method, or via CSS, like this:

```

```

You can pass a speed to the `show()` method which tells JQuery how fast to show the element. Yo between three predefined speeds, or a custom number of milliseconds. Here are a few examples

```
$('#theImg').show('slow');
$('#theImg').show('medium');
$('#theImg').show('fast');
$('#theImg').show(2000);
```

Note: In Firefox it may look better to put an `img` inside a `div`, and then hide and show the `div`, than and showing the `img` directly. You'll have to play around with that, to see what works best for your

and showing the `img` directly. You'll have to play around with that, to see what works best for your

It is also possible to pass a function to `show()` which is executed once the element is fully shown. a few examples of that:

```
$('#theImg').show(function(){ alert('shown');} );  
$('#theImg').show(2000, function(){ alert('shown');} );
```

## hide()

The JQuery `hide()` method can hide an HTML element. Just like with `show()`, there are a few speed can choose from, including setting your own speed in milliseconds. Here are a few code example

```
$('#theDiv').hide();  
$('#theDiv').hide('slow');  
$('#theDiv').hide('medium');  
$('#theDiv').hide('fast');  
$('#theDiv').hide(2000);
```

It is also possible to pass a function to `hide()` which is executed once the element is fully hidden. how:

```
$('#theDiv').hide(function(){ alert('shown');} );  
$('#theDiv').hide(2000, function(){ alert('shown');} );
```

## toggle()

The JQuery `toggle()` function toggles the visibility state of an HTML element. In other words, if the is hidden, it will be shown. If it is shown, it will be hidden. Again, you can set the speed of the toggle a parameter. Here are a few examples:

```
$('#theDiv').toggle();  
$('#theDiv').toggle('slow');  
$('#theDiv').toggle('medium');  
$('#theDiv').toggle('fast');  
$('#theDiv').toggle(2000);
```

It is also possible to pass a function to `toggle()` which is executed once the element is fully toggle is how:

```
$('#theDiv').toggle(function(){ alert('shown');} );  
$('#theDiv').toggle(2000, function(){ alert('shown');} );
```

## slideDown()

The `slideDown()` function slides down a hidden HTML element, so it becomes visible. Like with the and `hide()` functions, you can set the speed as a parameter to the function. Here are a few exam

```
$('#theDiv').slideDown();  
$('#theDiv').slideDown('slow');  
$('#theDiv').slideDown('medium');  
$('#theDiv').slideDown('fast');  
$('#theDiv').slideDown(2000);
```

You can also pass a callback function to `slideDown()` which is executed once the sliding is fully done are a few example:

```
$('#theDiv').slideDown(function(){ alert('done');} );  
$('#theDiv').slideDown(2000, function(){ alert('done');} );
```

## slideUp()

The `slideUp()` function hides a visible HTML element by sliding it up. Like with the `show()` and `hide` functions, you can set the speed as a parameter to the function. Here are a few examples:

```
$('#theDiv').slideUp();  
$('#theDiv').slideUp('slow');  
$('#theDiv').slideUp('medium');  
$('#theDiv').slideUp('fast');  
$('#theDiv').slideUp(2000);
```

You can also pass a callback function to `slideUp()` which is executed once the sliding is fully done are a few example:

```
$('#theDiv').slideUp(function(){ alert('done');} );  
$('#theDiv').slideUp(2000, function(){ alert('done');} );
```

## slideToggle()

The `slideToggle()` function shows an HTML element if it is hidden, or hides it if visible. It does so the HTML element up and down. Here are a few examples:

```
$('#theDiv').slideToggle();
$('#theDiv').slideToggle('slow');
$('#theDiv').slideToggle('medium');
$('#theDiv').slideToggle('fast');
$('#theDiv').slideToggle(2000);
```

You can also pass a callback function to `slideToggle()` which is executed once the sliding is fully done. Here are a few examples:

```
$('#theDiv').slideToggle(function(){ alert('done'); } );
$('#theDiv').slideToggle(2000, function(){ alert('done'); } );
```

## fadeIn()

The `fadeIn()` function fades in a hidden HTML element. Like with the `show()` and `hide()` functions, set the speed as a parameter to the function. Here are a few examples:

```
$('#theDiv').fadeIn();
$('#theDiv').fadeIn('slow');
$('#theDiv').fadeIn('medium');
$('#theDiv').fadeIn('fast');
$('#theDiv').fadeIn(2000);
```

You can also pass a callback function to `fadeIn()` which is executed once the fading is fully done. Here are a few examples:

```
$('#theDiv').fadeIn(function(){ alert('done'); } );
$('#theDiv').fadeIn(2000, function(){ alert('done'); } );
```

## fadeOut()

The `fadeOut()` function fades out a visible HTML element. Like with the `show()` and `hide()` functions, set the speed as a parameter to the function. Here are a few examples:

```
$('#theDiv').fadeOut();
$('#theDiv').fadeOut('slow');
$('#theDiv').fadeOut('medium');
$('#theDiv').fadeOut('fast');
$('#theDiv').fadeOut(2000);
```

You can also pass a callback function to `fadeOut()` which is executed once the fading is fully done. Here are a few examples:

```
$('#theDiv').fadeOut(function(){ alert('done'); } );
$('#theDiv').fadeOut(2000, function(){ alert('done'); } );
```

## fadeToggle()

The `fadeToggle()` function shows an HTML element if it is hidden, or hides it if visible. It does so by toggling the HTML element in and out. Here are a few examples:

```
$('#theDiv').fadeToggle();
$('#theDiv').fadeToggle('slow');
$('#theDiv').fadeToggle('medium');
$('#theDiv').fadeToggle('fast');
$('#theDiv').fadeToggle(2000);
```

You can also pass a callback function to `fadeToggle()` which is executed once the fading is fully done. Here are a few examples:

```
$('#theDiv').fadeToggle(function(){ alert('done'); } );
$('#theDiv').fadeToggle(2000, function(){ alert('done'); } );
```

## fadeTo()

The `fadeTo()` function enables you to fade an HTML element partially in or out, making it transparent. You pass a parameter between 0 and 1 which tells how much to fade the element. 0 is fully transparent and 1 is fully opaque (shown). Like with the `show()` and `hide()` functions, you can set the speed as a parameter to the function. Here are a few examples:

```
$('#theDiv').fadeTo(0.5);
$('#theDiv').fadeTo('slow', 0.5);
$('#theDiv').fadeTo('medium', 0.5);
$('#theDiv').fadeTo('fast', 0.5);
$('#theDiv').fadeTo(2000, 0.5);
```

You can also pass a callback function to `fadeTo()` which is executed once the fading is fully done.

a few example:

```
$('#theDiv').fadeTo(0.5, function(){ alert('done');} );
$('#theDiv').fadeTo(2000, 0.5, function(){ alert('done');} );
```

Note:  
If you fade an HTML element to e.g. 0.5, and then call the `fadeOut()` function, and after that the `fadeIn()` function, the HTML element will only be faded back in to 0.5, not to 1.0. You can see this effect by clicking the buttons at the top of the page.

animate()

The JQuery `animate()` function enables you to animate HTML elements in your page. Not every element can be animated. Only numerical CSS attributes can. For instance, you can animate

- `borderWidth`
- `width`
- `height`
- `fontSize`
- `opacity`
- `margin`
- `padding`
- `bottom`
- `left`
- `right`
- `top`
- `wordSpacing`
- etc.

Animating a CSS attribute changes it gradually from the value it had before the animation started to the value specified in the `animate()` function call.

Here are two examples:

```
$('#theDiv').animate({height" : 300}, 'slow');
$('#theDiv').animate({width" : 200}, 'slow');
```

You can animate more than one property at a time, by putting more properties into the object parameter. Here is an example that animates both width and height at the same time, instead of one after the other:

```
$('#theDiv').animate({height" : 250, width:250 }, 'slow');
```

You can also pass a callback function to the `animate()` function, which is executed once the animation is finished. Here is an example:

```
$('#theDiv').animate({width" : 200}, 'slow',
function(){ alert('done'); } );
```

Camel Case

You might have noticed that some of the CSS properties are written in a mixture of uppercase and lowercase letters. This is because JavaScript object properties cannot contain hyphens. For instance, the CSS property `border-width` cannot directly be written as a property of a JavaScript object. This example would fail to be parsed:

```
$('#theDiv').css({ border-width : 1 });
```

To address CSS properties with hyphens in your animations etc. you must either quote the property name or use a camel case version of the property, like this:

```
$('#theDiv').css({ "border-width" : 1 });
```

or use a camel case version of the property, like this:

```
$('#theDiv').css({ borderWidth : 1 });
```

The examples above only showed the property naming used inside the `css()` function. However, the same rules apply inside the `animate()` function.

Toggle Animation

It is possible to "toggle animate" an HTML element. For instance, if you toggle animate the `width` of an element the first time the animation is executed, the `width` will be animated down to 0. Second time you execute the animation, the `width` is animated back to the value it had before the first animation. Here is an example:

```
$('#theDiv').animate({width" : 'toggle'}, 'slow' );
```

You might have to play around a little with this function to get it to work properly. I have experienced that sometimes only one of the given CSS properties are animated etc. and I have not yet found out why (I am doing wrong, in other words). But it seems to work fine if you first toggle the HTML element's visibility.

All Trails	Trail TOC	Page TOC	Previous	Next
------------	-----------	----------	----------	------

visibility with it, and then to invisibility again. If you start the other way around (hide, then show), I have some strange problems. If you know the solution to this problem, I would appreciate an email with explanation.

## Stopping Animation

You can also stop an animation that is ongoing, by calling the `stop()` method on an HTML element like this:

```
$('#theDiv').stop();
```

Stopping an animation might be useful if you are to start a new animation, and you don't want the animations to run at the same time.

That about sums it up for jQuery effects. By combining these effects you can create your own, more advanced effects. For instance mixtures of sliding and fading etc. Spend some time playing around with the possibilities to get a good feeling for how they work, and what you can use them for in your application. Remember, some effects look nice when you see them the first time, but become tiresome after a while.

Next: [jQuery AJAX](#)

0 Comments [tutorials.jenkov.com](#)

♥ Recommend [Share](#)



Start the discussion...

Be the first to comment.

ALSO ON TUTORIALS.JENKOV.COM

### Android Web Apps Using Android WebView

12 comments • a year ago



**Jakob Jenkov** — Hi Eric, to do that you must load the content of the URL into a byte array, manipulate that byte array, and then pass the manipulated byte array on to the WebView. To do so you must intercept the HTTP request that ...

### Android Toast

1 comment • a year ago



**BASAM SRERAM** — very nice tutorial..

### Vert.x Tutorial

4 comments • 7 months ago



**Jakob Jenkov** — As far as I know, Vert.x has support for RxJava, but I don't actually know how RxJava style apps look different from regular V... yet, at least.

### Touchable Web Design

2 comments • a year ago



**Jakob Jenkov** — As far as I know it is now allowed. At least it works in the browsers I have tested it in (I think).

✉ Subscribe

Add Disqus to your site [Add Disqus](#) [Add](#)

Privacy