

9.4 Styles and animation

The `css()` method

jQuery simplifies the process of adding and removing CSS properties to an element's inline style. CSS properties can be added to selected elements with the `css()` method as illustrated in the following animation.

PARTICIPATION
ACTIVITY

9.4.1: Using the `.css()` method.



1 2 3 ← ✓ 2x speed

```
<body style="background-color: peachpuff;  
            color: green;  
            font-size: 20pt">  
  <p>Hello!</p>  
</body>
```

Hello!

```
$( "body" ).css( "background-color", "peachpuff" );  
  
$( "body" ).css({  
  color: "green",  
  "font-size": "20pt"  
});
```

`css()` method adds two CSS properties to `<body>` using an object literal.

Captions ^

1. Browser renders the webpage.
2. `css()` method adds the "background-color" CSS property to the `<body>`.
3. `css()` method adds two CSS properties to `<body>` using an object literal.

[Feedback?](#)

jQuery converts each camel-cased property name to a string that is equivalent to the CSS property name. Ex: `backgroundColor` is converted to `"background-color"`. However, `"background-color"` cannot be used without quotes since the dash is not a legal character in a JavaScript identifier.

PARTICIPATION
ACTIVITY

9.4.2: Using the `.css()` method.



Given the following HTML, what is the effect of each jQuery code segment?

```
<h1 style="border-style: solid">To Do</h1>
<span id="important">Study for exam</span> and
<span>feed the fish</span>
```

1) `$("#important").css("display", "none");`

- "Study for exam" is replaced with a blank area.
- "Study for exam" is no longer visible, and the browser does not leave room for the missing words.
- Nothing because no elements use the class "important".

Correct



The span remains in the DOM even though the span is no longer visible.

2) `$("h1").css({
 "text-decoration": "underline",
 "border-style": "dotted"
});`

- "To Do" is underlined with a dotted border.
- "To Do" is underlined, but the solid border remains.
- Nothing because there is a syntax error in the object literal.

Correct



The new border-style replaces the border-style specified in the HTML.



```
3) $("span").css({
  textDecoration:
  "underline",
  border-style:
  "dotted" });
```

- Both span elements are underlined with a dotted border.
- Nothing because quotes are needed around textDecoration.
- Nothing because there is a syntax error in the object literal.

Correct

The property border-style must either be quoted "border-style" or unquoted as borderStyle.

```
4) let x =
  $("h1").css("border-
  style");
```

- The border around "To Do" is removed.
- x is set to an empty string.
- x is set to "solid".

Correct

Calling css() with only a CSS property returns the current property value.



```
5) $("h1").css("border-
  style", "");
```

- "To Do" continues to have a solid border.
- Nothing because an empty string cannot be used as a property value.
- The solid border around "To Do" is removed.

Correct

An empty string for a property value means that the CSS property should be removed from the element.

**jQuery effects**

jQuery has three sets of animation methods for showing and hiding webpage elements. The animation methods in the table below alter various CSS properties of selected elements to produce animations. The methods take a speed argument that determines the duration of the animation. jQuery animation methods are also called **jQuery effects**.

Table 9.4.1: Animation methods.

Methods	Example	Description
<code>show()</code> <code>hide()</code> <code>toggle()</code>	<code>\$("h1").show("slow");</code> <code>\$("h1").hide("slow");</code> <code>\$("h1").toggle("slow");</code>	Alters width, height, and opacity all at once
<code>fadeIn()</code> <code>fadeOut()</code> <code>fadeToggle()</code>	<code>\$("h1").fadeIn("normal");</code> <code>\$("h1").fadeOut("normal");</code> <code>\$("h1").fadeToggle("normal");</code>	Alters opacity only
<code>slideDown()</code> <code>slideUp()</code> <code>slideToggle()</code>	<code>\$("h1").slideDown("fast");</code> <code>\$("h1").slideUp("fast");</code> <code>\$("h1").slideToggle("fast");</code>	Alters height only

[Feedback?](#)

Table 9.4.2: The speed argument for animation methods.

Argument	Example	Explanation
"slow"	<code>\$("p").show("slow");</code>	0.6 seconds to show the paragraph
"normal"	<code>\$("p").show("normal");</code>	0.4 seconds to show the paragraph
"fast"	<code>\$("p").show("fast");</code>	0.2 seconds to show the paragraph
milliseconds	<code>\$("p").show(1500);</code>	1.5 seconds to show the paragraph

[Feedback?](#)



The following webpage is using jQuery animation methods to show and hide the cat image. Click on each button to see the effect on the image. Then try modifying the JavaScript so that when the image is clicked two things will happen:

1. A solid red border will immediately appear around the image. Use the `css()` method to add the appropriate CSS properties.
2. The image will quickly hide, display, fade out, fade in, slide up, and slide down. Call each of the animation methods to produce the series of animations.

HTML

CSS

JavaScript

```
6   <button id="toggle-btn">toggle</button>
7 </div>
8 <div>
9   <button id="fadein-btn">fadeIn</button>
10  <button id="fadeout-btn">fadeOut</button>
11  <button id="fadetoggle-btn">fadeToggle</button>
12 </div>
13 <div>
14   <button id="slidedown-btn">slideDown</button>
15   <button id="slideup-btn">slideUp</button>
16   <button id="slidetoggle-btn">slideToggle</button>
17 </div>
18
19 ` elements during a 2 second interval.

- True  
 False

**Correct**

The interval will be only 2 milliseconds.  
`$( "h1" ).toggle(2000)` uses a 2 second interval.



- 4) The `slideDown()` method will slide the element down until the element is no longer visible.

- True  
 False

**Correct**

`slideDown()` slides the element down to make the element visible.



[Feedback?](#)

## Controlling animation order

Sometimes developers want to perform an operation after an effect completes. Ex: Display a "Thank you" message after the user has typed information into a form that gradually disappears. However, code that is written after a call to an animation method will execute before the animation is complete. To execute code after the animation is complete, the jQuery animation methods can be passed a callback function.

**PARTICIPATION ACTIVITY**

## 9.4.5: Waiting for an animation to complete.



1 **2** ◀  2x speed

```
<body>
 <h1>jQuery effects</h1>
 <h2>Good stuff!</h2>
</body>

$("h1").fadeOut("slow");
$("h2").fadeOut("slow");

$("h1").fadeIn("slow", function() {
 $("h2").fadeIn("slow");
});
```

**jQuery effects**  
Good stuff!

The first fadeIn() completes before the second fadeIn() is called, so <h2> fades in after <h1> has finished fading in.

Captions ^

1. The fadeOut() methods cause <h1> and <h2> to fade out at the same time.
2. The first fadeIn() completes before the second fadeIn() is called, so <h2> fades in after <h1> has finished fading in.

[Feedback?](#)

When two animations are applied to different elements, the animations occur at the same time. However, when two animations are applied to the same element, the first animation completes before starting the second animation. jQuery queues animations applied to the same element so that each animation finishes before the next begins.

Figure 9.4.1: Queuing animations on the same element.

```
// Both animations occur at the same time
$("h1").fadeOut("slow");
$("h2").fadeOut("slow");

// Animations on the same element are queued
$("h1").fadeIn("slow");
$("h1").slideUp(1000);
```

[Feedback?](#)

Non-animation method calls are not placed in the animation queue.

Figure 9.4.2: Only animation methods are queued.

```
$("h1").hide("slow")
 .addClass("important") // Called immediately, not waiting for
 hide to complete
 .slideDown(1000); // Begins after hide completes
```

[Feedback?](#)

jQuery provides a **queue()** method to aid in queuing code that should be executed after the previous animations complete. The **queue()** method can take the place of the callback function used with the animation methods. The **queue()** method takes a function argument that is passed a function parameter called **next**. The **next** function must be called so the next animation in the queue can be processed. *Good practice is to use the **queue()** method instead of callback functions when developing elaborate animations. Using many nested callback functions can lead to overly complicated code.*

Figure 9.4.3: Using the **.queue()** method.

```
$("h1").hide("slow")
 .queue(function(next) {
 // Add the class after the <h1> hide
 completes
 $(this).addClass("important");

 // Process the next animation in the queue
 next();
 })
 .slideDown(1000);
```

[Feedback?](#)

PARTICIPATION  
ACTIVITY

9.4.6: Animation order.



Refer to the jQuery code below:

```
$("h1").show(1000)
 .queue(function(next) {
 $(this).removeClass("important");
 next();
 })
 .css("color", "red")
 .delay(1000)
 .hide("fast", function() {
 $(this).css("color", "green");
 });
}
```

- 1) The show animation completes before the hide animation begins.

True  
 False

**Correct**

Two animations are applied to the same `<h1>` element, so the animations are queued. The first animation will complete before the second begins.

- 2) `$(this)` in the code above is equivalent to `$( "h1" )`.

True  
 False

**Correct**

`this` is the currently selected DOM node, and calling `$(this)` returns the `jQuery` object representing the DOM node. `$(this)` is more efficient than calling `$( "h1" )`, which forces the JavaScript interpreter to search through the DOM again.

- 3) The show animation completes before the "important" class is removed from `<h1>`.

True  
 False

**Correct**

The `queue()` method adds the anonymous function that is executed after the show animation completes.

- 4) The show animation completes before the CSS color property is set to red.

True  
 False

**Correct**

The `css()` method is not a `jQuery` animation method, so the method call is not queued. The CSS color property is immediately applied when the show animation begins.

- 5) The `delay()` method delays executing the hide animation for one second.

True  
 False

**Correct**

The `delay()` method allows delays to be added to the animation queue.

- 6) The hide animation completes before the

**Correct**

CSS color property is set to green.

- True
- False

The callback function for `hide()` is not called until the hide animation completes.

[Feedback?](#)

## The animate() method

PARTICIPATION  
ACTIVITY

9.4.7: The animate() method.



1 2 3 4 ← ✓ 2x speed

```
<body>

</body>
```

```
$("img").animate({
 left: "200px",
 height: "-=60"
}, "slow", "linear");

$("img").animate({
 opacity: 0.1,
 top: "+=50"
}, 2000);

$("img").animate({
 opacity: 1,
 width: "200px"
}, function() {
 $(this).css("border", "2px solid red");
});
```



`css()` sets image's border to solid red after animation completes.

Captions ^

1. `animate()` makes image's left become 200px and height become 100px - 60 = 40px with linear easing over 0.6 secs (length of "slow").
2. `animate()` makes image's opacity change to 0.1, top to 0 + 50 = 50px over 2 seconds.
3. `animate()` makes image's opacity change to 1 and width to 200px over 0.4 seconds (default length of time).
4. `css()` sets image's border to solid red after animation completes.

**PARTICIPATION ACTIVITY**

## 9.4.8: The animate() method.



- 1) In the `animate()` property:value map, what does the property value "+=20" do?

**Correct**

Ex: If the width property is 30px, the width becomes 30x + 20px = 50px.



- Sets the property to the number 20.
- Adds 20px to the property's value.
- Subtracts 20px from the property's value.

- 2) What easing function does the code segment below use?

**Correct**

Swing is the easing function used if an easing function is not explicitly specified.



```
$("img").animate({
 opacity: 1,
 width: "200px"
}, function() {

 $(this).css("border",
 "2px solid red");
});
```

- linear
- swing
- No easing function was specified, so no easing was used.

- 3) Can `animate()` animate the transition from one font color to another?

- Yes
- No

**Correct**

jQuery doesn't support non-numeric property animations like colors, but the jQuery UI plugin supports color animations.



**PARTICIPATION  
ACTIVITY**

## 9.4.9: Practice with animate() and queue().



The following webpage displays some help information when the "?" is clicked. The help information is hidden when "?" is clicked a second time.

Make the following modifications so the rendered webpage acts like the expected webpage:

1. Modify the `animate()` call to make the width of "?" increase by 15px and font-size increase by 10px when the "?" is first clicked. Make the width decrease by 15px and font-size decrease by 10px when "?" is clicked again.
2. Use the `queue()` method to change the background of the "?" to orange after the "?" has finished moving to the right. Change the background back to light blue after the "?" has finished moving back to the left.
3. Use the `queue()` method to fade in the help text after the "?" has finished moving to the right. Fade out the help text when the "?" has been clicked a second time but before the "?" moves back to the left.

[HTML](#)[CSS](#)[JavaScript](#)

```
1 <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/
2
3
4 <div id="help" title="Help">
5 ?
6 </div>
7 <div id="help-text">
8 To print the document, select the Print option from th
9 </div>
10
```

[Render webpage](#)[Reset code](#)

**Your webpage****Expected webpage**

▼ View solution

 Explain

--- START FILE: HTML ---

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>

<div id="help" title="Help">
 ?
</div>
<div id="help-text">
 To print the document, select the Print option
 from the File menu.
</div>

<style type="text/css">#help-text {
 background-color: lightblue;
 position: absolute;
 left: 10px;
 top: 10px;
 width: 240px;
 font-size: 12pt;
 padding: 10px;
 display: none;
 z-index: 0;
}
```

```
#help {
 font-size: 26pt;
 font-weight: bold;
 text-align: center;
 color: darkblue;
 background: lightblue;
 position: absolute;
 left: 10px;
 top: 10px;
 width: 40px;
 border-radius: 50px;
 border: darkblue 2px solid;
 cursor: pointer;
 z-index: 2;
}

</style>

<script type="text/javascript">let $helpText = $("#help-text");

$("#help").click(function() {
 if ($helpText.is(":visible")) {
 $(this).queue(function(next) {
 $helpText.fadeOut("fast");
 next();
 })
 .animate({
 left: "10px",
 width: "-=15",
 fontSize: "-=10"
 }, "slow")
 .queue(function(next) {
 $(this).css("background", "lightblue");
 next();
 })
 }
 else {
 $(this).animate({
 left: "240px",
 width: "+=15",
 fontSize: "+=10"
 }, "slow")
 .queue(function(next) {
 $(this).css("background", "orange");
 next();
 })
 .queue(function(next) {
 $helpText.fadeIn("fast");
 })
 }
})
```

```
 next();
 }
}
});
</script>
```

--- END FILE: HTML ---

[Feedback?](#)

Exploring further:

- [jQuery Effects documentation](#)
- [Queue and Dequeue Explained](#)

How was  
this  
section?



[Provide section feedback](#)