

jQuery Effects – Animation Method in jQuery

The jQuery `animate()` method is used to create custom animations.

Syntax:

```
$(selector).animate({params},speed,callback);
```

The required `params` parameter defines the CSS properties to be animated.

The optional `speed` parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.

The optional `callback` parameter is a function to be executed after the animation completes.

The following example demonstrates a simple use of the `animate()` method; it moves a `<div>` element to the right, until it has reached a left property of 250px:

Example

```
$("#button").click(function(){  
    $("#div").animate({left: '250px'});  
});
```

jQuery `animate()` - Manipulate Multiple Properties

Notice that multiple properties can be animated at the same time:

Example

```
$("#button").click(function(){  
    $("#div").animate({  
        left: '250px',  
        opacity: '0.5',  
    });  
});
```

```
    height: '150px',  
    width: '150px'  
  });  
});
```

jQuery animate() - Using Relative Values

It is also possible to define relative values (the value is then relative to the element's current value). This is done by putting += or -= in front of the value:

Example

```
$("#button").click(function(){  
  $("#div").animate({  
    left: '250px',  
    height: '+=150px',  
    width: '+=150px'  
  });  
});
```

jQuery animate() - Using Pre-defined Values

You can even specify a property's animation value as "show", "hide", or "toggle":

Example

```
$("#button").click(function(){  
  $("#div").animate({  
    height: 'toggle'  
  });  
});
```

```
});  
});
```

jQuery animate() - Uses Queue Functionality

By default, jQuery comes with queue functionality for animations.

This means that if you write multiple `animate()` calls after each other, jQuery creates an "internal" queue with these method calls. Then it runs the `animate` calls ONE by ONE.

So, if you want to perform different animations after each other, we take advantage of the queue functionality:

Example 1

```
$("#button").click(function(){  
    var div = $("#div");  
    div.animate({height: '300px', opacity: '0.4'}, "slow");  
    div.animate({width: '300px', opacity: '0.8'}, "slow");  
    div.animate({height: '100px', opacity: '0.4'}, "slow");  
    div.animate({width: '100px', opacity: '0.8'}, "slow");  
});
```

The example below first moves the `<div>` element to the right, and then increases the font size of the text:

Example 2

```
$("#button").click(function(){  
    var div = $("#div");  
    div.animate({left: '100px'}, "slow");  
    div.animate({fontSize: '3em'}, "slow");  
});
```

Source Code:

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Animate Method in jQuery</title>
  <style>
    .box {
      width: 300px;
      height: 300px;
    }

    #test1 {
      background-color: aqua;
      border: 2px solid black;
      overflow: auto;
      padding: 10px 10px;
      position: relative;
    }

  </style>
  <script src="jquery-3.7.1.min.js"></script>
</head>

<body>

  <h1> Animate Method in jQuery</h1>
  <div class="box" id="test1">
    <h1>Box Dev</h1>
    <p>Lorem ipsum dolor sit, amet consectetur adipisicing elit. Esse quia
quasi expedita fssimus eum dolor. Iure adipisci voluptates esse cupiditate sunt
nostrum! .
    </p>

  </div>
  <button id="btnAnimate">Animate</button>
```

```
<script>
    $(document).ready(function() {
        $('#btnAnimate').click(function() {
            $('#test1').animate({
                left:'200px',
                width:'400px',
                fontSize:'30px',

            },2000,function(){
                console.log("Animate the width from left");
            })

            $('#test1').animate({
                top:'300px'
            })
        })
    });
</script>
</body>

</html>
```

Thank You