

# jQuery Animations - The animate() Method

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:

```
<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js"></script>

<script>

$(document).ready(function()

{

    $("button").click(function()

    {

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

    }

});

}
```

```
);
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<button>Start Animation</button>
```

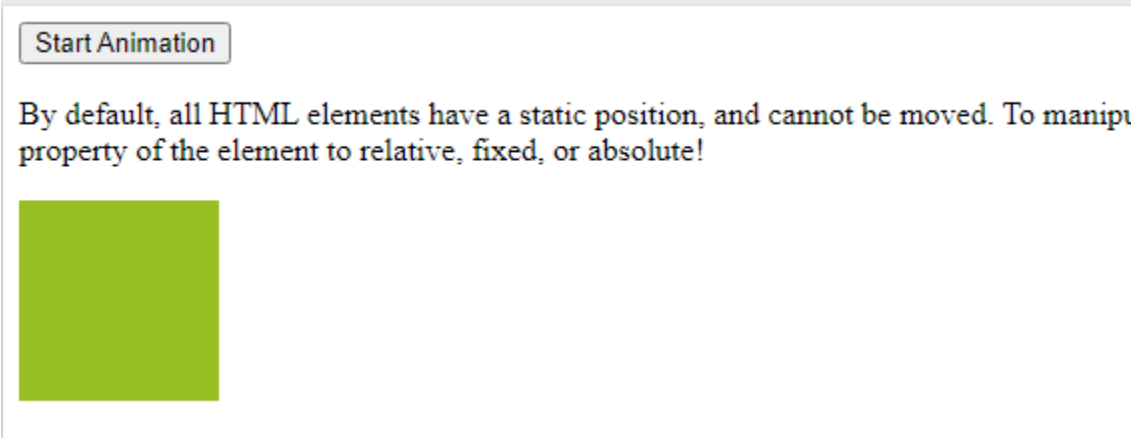
<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>

```
<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>
```

```
</body>
```

```
</html>
```

Output:-

A screenshot of a web browser window. At the top, there is a button with the text "Start Animation". Below the button, there is a paragraph of text: "By default, all HTML elements have a static position, and cannot be moved. To manipulate property of the element to relative, fixed, or absolute!". Below the text, there is a solid green square.

By default, all HTML elements have a static position, and cannot be moved. To manipulate property of the element to relative, fixed, or absolute!



# jQuery animate() - Manipulate Multiple Properties

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

```
<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js"></script>

<script>

$(document).ready(function(){

    $("button").click(function(){

        $("div").animate({

            left: '250px',

            opacity: '0.5',

            height: '150px',

            width: '150px'

        });

    });

});

</script>

</head>

<body>

<button>Start Animation</button>
```

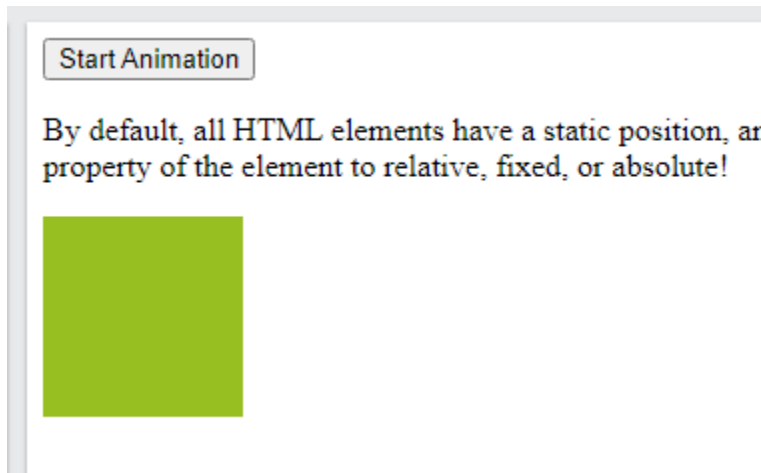
<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>

```
<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>
```

```
</body>
```

```
</html>
```

Output:-



And after click on Start animation you will see



By default, all HTML elements have a static position, and cannot be moved. To n property of the element to relative, fixed, or absolute!



# 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:

```
<!DOCTYPE html>

<html>

<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js"></script>

<script>

$(document).ready(function(){

    $("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");

    });

});

</script>

</head>

<body>
```

```
<button>Start Animation</button>
```

<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>

```
<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>
```

```
</body>
```

```
</html>
```

Output:-



By default, all HTML elements have a static position property of the element to relative, fixed,



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

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js"></script>
```

```
<script>
```

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

```
  });
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<button>Start Animation</button>
```

<p>By default, all HTML elements have a static position, and cannot be moved. To manipulate the position, remember to first set the CSS position property of the element to relative, fixed, or absolute!</p>

```
<div style="background:#98bf21;height:100px;width:200px;position:absolute;">HELLO</div>
```

```
</body>
```

```
</html>
```

Output:-

Start Animation

By default, all HTML elements have a static position, and can have the position property of the element to relative, fixed, or absolute!

HELLO

