

# JAVASCRIPT - ANIMATION

[http://www.tutorialspoint.com/javascript/javascript\\_animation.htm](http://www.tutorialspoint.com/javascript/javascript_animation.htm)

Copyright © tutorialspoint.com

You can use JavaScript to create a complex animation having, but not limited to, the following elements –

- Fireworks
- Fade Effect
- Roll-in or Roll-out
- Page-in or Page-out
- Object movements

You might be interested in existing JavaScript based animation library: [Script.Aculo.us](http://Script.Aculo.us).

This tutorial provides a basic understanding of how to use JavaScript to create an animation.

JavaScript can be used to move a number of DOM elements `<img/>`, `<div>` or any other HTML element around the page according to some sort of pattern determined by a logical equation or function.

JavaScript provides the following two functions to be frequently used in animation programs.

- **setTimeout***function, duration* – This function calls **function** after **duration** milliseconds from now.
- **setInterval***function, duration* – This function calls **function** after every **duration** milliseconds.
- **clearTimeout***setTimeout, variable* – This function calls clears any timer set by the **setTimeout** functions.

JavaScript can also set a number of attributes of a DOM object including its position on the screen. You can set *top* and *left* attribute of an object to position it anywhere on the screen. Here is its syntax.

```
// Set distance from left edge of the screen.  
object.style.left = distance in pixels or points;  
  
or  
  
// Set distance from top edge of the screen.  
object.style.top = distance in pixels or points;
```

## Manual Animation

So let's implement one simple animation using DOM object properties and JavaScript functions as follows. The following list contains different DOM methods.

- We are using the JavaScript function **getElementById** to get a DOM object and then assigning it to a global variable **imgObj**.
- We have defined an initialization function **init** to initialize **imgObj** where we have set its **position** and **left** attributes.
- We are calling initialization function at the time of window load.
- Finally, we are calling **moveRight** function to increase the left distance by 10 pixels. You could also set it to a negative value to move it to the left side.

## Example

Try the following example.

```

<html>

<head>
  <title>JavaScript Animation</title>

  <script type="text/javascript">
    <!--
      var imgObj = null;

      function init(){
        imgObj = document.getElementById('myImage');
        imgObj.style.position= 'relative';
        imgObj.style.left = '0px';
      }

      function moveRight(){
        imgObj.style.left = parseInt(imgObj.style.left) + 10 + 'px';
      }

      window.onload =init;
    //-->
  </script>

</head>

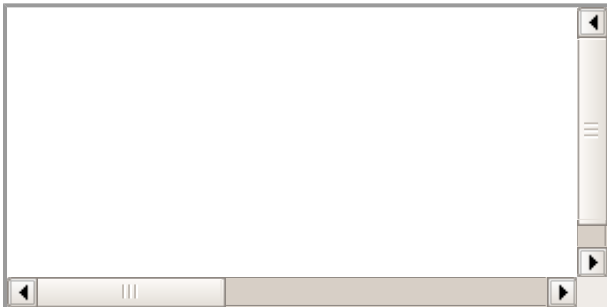
<body>

  <form>
    <img />
    <p>Click button below to move the image to right</p>
    <input type="button" value="Click Me" onclick="moveRight();" />
  </form>

</body>
</html>

```

## Output



## Automated Animation

In the above example, we saw how an image moves to right with every click. We can automate this process by using the JavaScript function **setTimeout** as follows –

Here we have added more methods. So let's see what is new here –

- The **moveRight** function is calling **setTimeout** function to set the position of *imgObj*.
- We have added a new function **stop** to clear the timer set by **setTimeout** function and to set the object at its initial position.

## Example

Try the following example code.

```

<html>

```

```

<head>
  <title>JavaScript Animation</title>

  <script type="text/javascript">
    <!--
      var imgObj = null;
      var animate ;

      function init(){
        imgObj = document.getElementById('myImage');
        imgObj.style.position= 'relative';
        imgObj.style.left = '0px';
      }

      function moveRight(){
        imgObj.style.left = parseInt(imgObj.style.left) + 10 + 'px';
        animate = setTimeout(moveRight,20); // call moveRight in 20msec
      }

      function stop(){
        clearTimeout(animate);
        imgObj.style.left = '0px';
      }

      window.onload =init;
    //-->
  </script>

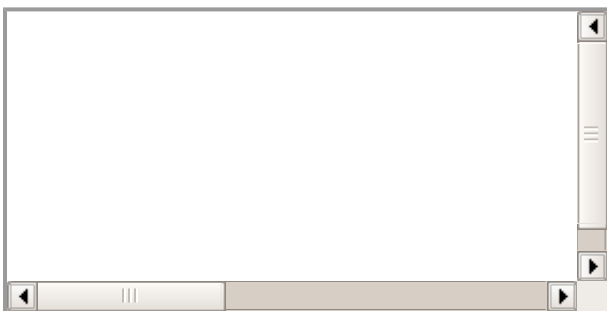
</head>

<body>

  <form>
    <img />
    <p>Click the buttons below to handle animation</p>
    <input type="button" value="Start" onclick="moveRight();" />
    <input type="button" value="Stop" onclick="stop();" />
  </form>

</body>
</html>

```



## Rollover with a Mouse Event

Here is a simple example showing image rollover with a mouse event.

Let's see what we are using in the following example –

- At the time of loading this page, the 'if' statement checks for the existence of the image object. If the image object is unavailable, this block will not be executed.
- The **Image** constructor creates and preloads a new image object called **image1**.
- The src property is assigned the name of the external image file called /images/html.gif.
- Similarly, we have created **image2** object and assigned /images/http.gif in this object.
- The # *hashmark* disables the link so that the browser does not try to go to a URL when clicked.

This link is an image.

- The **onMouseOver** event handler is triggered when the user's mouse moves onto the link, and the **onMouseOut** event handler is triggered when the user's mouse moves away from the link *image*.
- When the mouse moves over the image, the HTTP image changes from the first image to the second one. When the mouse is moved away from the image, the original image is displayed.
- When the mouse is moved away from the link, the initial image `html.gif` will reappear on the screen.

```
<html>

<head>
  <title>Rollover with a Mouse Events</title>

  <script type="text/javascript">
    <!--
      if(document.images){
        var image1 = new Image(); // Preload an image
        image1.src = "/images/html.gif";
        var image2 = new Image(); // Preload second image
        image2.src = "/images/http.gif";
      }
    <!-->
  </script>
</head>

<body>
  <p>Move your mouse over the image to see the result</p>

  <a href="#" onMouseOver="document.myImage.src=image2.src;"
onMouseOut="document.myImage.src=image1.src;">
    
  </a>
</body>
</html>
```

