Animation in JavaScript

You can use JavaScript to create a complex, unlimited effect.

You can use JavaScript to create a complex, unlimited effect, with the following elements:

1. Fireworks
2. Fade Effect
3. Roll-in or Roll-out
4. Page-in or Page-out
5. Object movements

You should consider the existing effects library built on JavaScript: **Script.Aculo.us** .

This tutorial provides you with a basic understanding of how to use JavaScript to create an effect.

JavaScript can be used to move several DOM elements (

Picture 1 of Animation in JavaScript

,
or any other HTML element) around the page according to a number of patterns is determined by a logical equation or function.

JavaScript provides the following two functions that are used frequently in animation programs.

1. **setTimeout (function, duration)** - This function calls the **Function** after the **duration** (in milliseconds) from now.
2. **setInterval (function, duration)** - This function calls the **function** after each **duration** (in milliseconds).
3. **clearTimeout (setTimeout_variable)** - This function removes any timer set by the setTimeout () function.

JavaScript can also set some properties of a DOM object including its position on the screen. You can set the **top** and **left** properties of an object to determine its position anywhere on the screen. Here is the syntax:

```
// Set distance from left edge of the screen. object . style . left  =  distanc
```

Hand operation animation

We now implement a simple animation using DOM object attributes and JavaScript functions as follows. The following lists the various DOM methods:

We are using the **getElementById ()** function to get a **DOM** object and then assign it to a Global variable, **imgObj** .

We have defined an initialization function **init ()** to initialize **imgObj** , where we have set its **position** and **left** properties.

We are calling the initialization function at the time of loading the window.

Finally, we are calling **moveRight ()** function to increase the left distance by 10 pixels. You can also set it to a negative value to move it to the left.

For example

Try the following example:

```
JavaScript Animation

Click button below to move the image to right
 type = "button" value = "Click Me" onclick = " moveRight (); " />
```

Result

Run the above command to see the result

Automation animation

In the above example, we saw how an image moves right every time you click. We can automate this process using the **setTimeout ()** function as follows:

Below, we add more methods. That is:

The **moveRight ()** function is calling the **setTimeout ()** function to set the position of imgObj.

We added a new function **stop ()** to delete the timer set by the setTimeout () function and to set the object at the initialization position.

For example

You try the following code:

```
JavaScript Animation

Click the buttons below to handle animation
 type = "button" value = "Start" onclick = " moveRight (); " />
 type = "button" value = "Stop" onclick = " stop (); " />
```

Result

Run the above command to see the result

Rollover with a Mouse Event

Below is a simple example to illustrate rollover images with a Mouse Event.

Let's look at what is being used in this example:

At the time of loading this page, the if command checks for the existence of the **Image** object. If this Image object is not available, this block will not be executed.

Image () constructor creates and reloads a new Image object called **image1** .

The src attribute is assigned the name of the external image file called ./images/html.gif.

Similarly, we create the **image2** object and assign ./images/http.gif in this object.

The # disables the link so that the browser doesn't try to reach the URL when clicked. This link is an image.

The **onMouseOver** error handling method is activated when the user moves the mouse over the link (the image), and **onMouseOut** event handler is triggered when the user moves the mouse off the link.

When moving the mouse over the image, the HTTP image changes from the first image to the second image. When moving the mouse out of the image, the original image is displayed.

When moving the mouse off the link, the first initialization image html.gif will reappear on the screen.

```
Rollover with a Mouse Events

Move your mouse over the image to see the result
 href = "#" onMouseOver = " document . myImage . src = image2 . src ; " onMouse
```

Result

Run the above command to see the result

According to Tutorialspoint

Previous article: Form Validation in JavaScript

Next lesson: Multimedia (Multimedia) in JavaScript