‘Animated Text’ Tutorial

By: Chad Clites
For: extant.digital
Date: 11/08/215
Email: chad@extant.digital

Real World Use Case:

As a designer, I want to draw attention to my new sales promo by creating a banner featuring animated text. The example below shows billowy clouds that move across the background, moving from right to left.

whitemask.png

Steps:

1. Create a mask.
2. Generate the HTML
3. Styling
4. JavaScript

Introduction:

I have been doing web development for quite a few years, and never had the opportunity to do much in the way of animating background images. I have had the idea of using an animated background to animate text, and this is a result of that idea.

There are a number of ways I could have used to accomplish the same effect. In fact, it would have been easier to do this using CSS animations. I could have also done this using vanilla JavaScript. I chose to use a mix of HTML and JQuery. As often is the case, there are usually any number of ways to accomplish what you want.

Create Mask:

For this part, one will need 3rd party image editing software to create a text 'mask'. The basic idea is to create a stencil in which the background of the text is transparent. If you know how to use image editing software, this is a relatively trivial exercise. If not, it is easy enough to find tutorials on how to create one's own masks.

Create HTML:

Again, this is reasonably trivial. Here is a barebones HTML skeleton:

```
<!DOCTYPE html>
<html>
  <head>
    <title>&amp;amp;Animated&amp;amp; Background</title>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/2.1.4/jquery.min.js"></script>
  </head>

  <body>

    <div class="content">
          <img src="whitemask.png" alt="extant digital heading"/>
    </div>

  </body>
</html>
```

This is a simple page. The only two things of interest are the JQuery import, and the 'content' . The div serves as our holding container.

Styling:

Since this is such a small example, I embedded my stylings directly in to the html.

```
.content, img{
   width: 800px;
   height: 200px;
}

.content {
        background: #00688B url(clouds.png) repeat 0 bottom;
        background-size: cover;
        margin: 10px;
        border: none;
}
```

We only have two declarations. The first declaration sets the size of the image mask and content container. The second declaration sets the background image, and orders the browser to scale the background image to the containing . The background image is a .png with a transparent background that I found online.

JavaScript:

Again, since this is such a small example, I embedded the JavaScript directly in the <head>. This isn't my JavaScript. I found a JSFiddle that did what I want, and borrowed it. (http://jsfiddle.net/3tQsc/)

```
<script>
	var scrollSpeed = 70;

	// set the default position
	var current = 0;

	// set the direction
	var direction = 'h';

	function bgscroll() {

		// 1 pixel row at a time
		current -= 1;

		// move the background with backgrond-position css properties
		$('div.content').css("backgroundPosition", (direction == 'h') ? current + "px -23px " :
"0 " + current + "px");

	}

	//Calls the scrolling function repeatedly
	setInterval(bgscroll, scrollSpeed);
</script>
```

For the most part, this should be relatively easy to understand. The only tricky part is the JQuery ternary statement inside the bgscroll function. The JQuery .css() function is a standard JQuery function to allow one to set properties of the GUI.

Since I copied the code I needed directly from the JSFiddle, there are a few parts that are unnecessary. The 'direction' flag is hard coded, so no matter what, the JQuery statement is going to move the background image 23px to the left, every 70 milliseconds.

TIP: For those that are not familiar with ternary evaluations, they are a shorthand syntax that replaces if/else syntax. So this:

```
var current =- 1;

if (direction == 'h'){
   current + "px -23px ";
}else{
   "0 " + current + "px";
}
```

Replaces:

```
$('div.content').css("backgroundPosition", (direction == 'h') ? current + "px -23px " : "0 " + current + "px");
```

Conclusion:

This is a fairly simple example of an effect that can be achieved with a minimum of code.