

Content Rotator



Using jQuery, Timers & Recursion

The Finished Project

This is a great, fairly simple, useful script that you could put on any web page for rotating content. It might be great for customer testimonials, for example.

It uses timers and recursion to loop through paragraphs inside a div.

Pet Supply Customer Comments

“Wow, you guys have the best selection of gecko supplies within a hundred miles!”
— Sally Smith

Project Setup

The setup for this project is simple. There is just a div with some paragraphs in it.

When the page loads, the div will be empty. Then the first paragraph fades in over two seconds. It stays on the screen for seven seconds, then fades out over two seconds.

Then the next paragraph fades in, waits seven seconds and then fades out.

After the last paragraph fades out, go back and fade the first paragraph in and repeat the process, indefinitely.

Pet Supply Customer Comments

“Thanks, I love my pair of black kittens.”

— John Jones

“Wow, you guys have the best selection of gecko supplies within a hundred miles!”

— Sally Smith

“My pet llama is eating all my house plants.”

— Bill Mead

“Your chickens rock. I always get the best egg layers from your store”

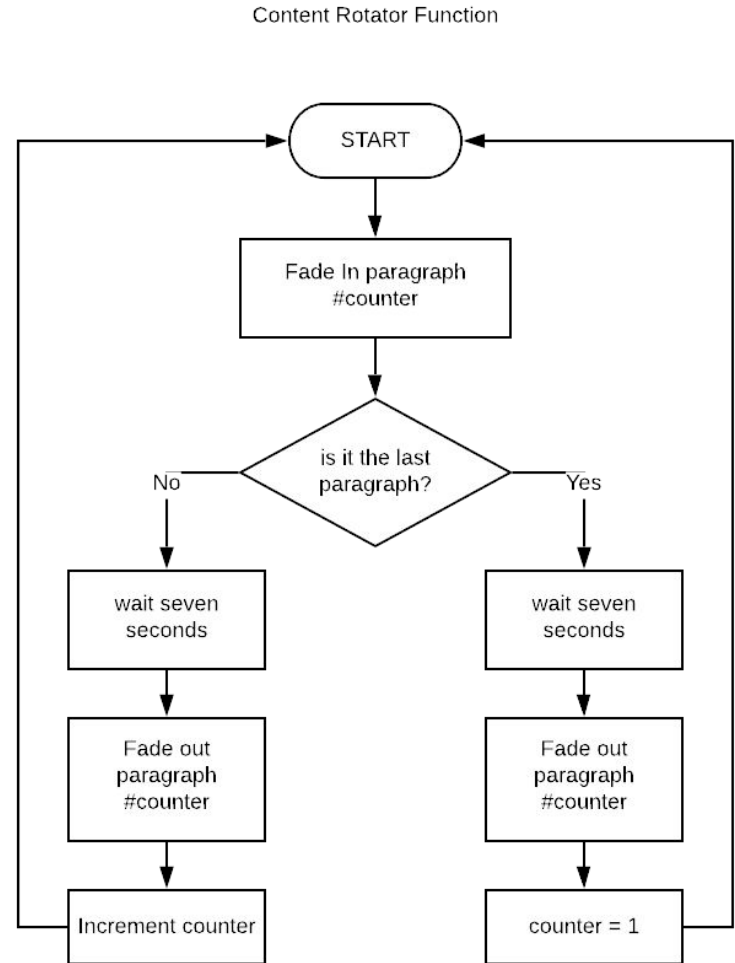
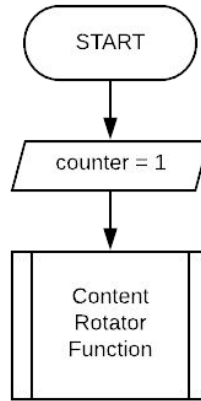
— George Jefferson.

Script Diagram

Even though the script is fairly simple, it can be confusing to think through the events.

A diagram of the script in this case can help. Hopefully you can see the script never ends.

Can you spot where the `setTimeout()` function will go?



Starting the Script

It's important to note that the CSS includes a rule to initially set the paragraphs so that they do not display at all.

Then on the script file, add a counter and assign it a value of 1.

Create the stub for the contentRotator function, then run the function.

```
#container p { display:none; }
```

```
(function(){  
  "use strict";  
  
  let counter=1;  
  
  function contentRotator(){  
    //do stuff here  
  }  
  
  contentRotator();  
  
})();
```

Fade in a Paragraph

```
let counter=1;

function contentRotator(){

    $('#container p:nth-child(${counter})').fadeIn( 2000 );

}
```

Add the line to the function that fades in a paragraph. The first time the script runs, counter will be 1, and it will fade in the first paragraph.

Note the use of tick marks and the template literal variable for `${counter}`.

Add a Callback Function

```
$(`#container p:nth-child(${counter})`).fadeIn( 2000, function(){} );
```

After the two second fadeIn animation has completed, you want to add a callback function.

The rest of the script will go between the two curly braces in that callback function

The rest of the script goes in here.



Add the if/else Statement

```
function contentRotator(){  
    $('#container p:nth-child(${counter})').fadeIn( 2000, function(){  
        if( $(this).is("#container p:last-child") ){  
            }  
        else{  
            }  
        } ); //end fadeIn callback function  
    }  
}
```

This if statement will check to see if you are on the last paragraph in the div.

Note the use of the “this” keyword and the is() method, which returns true or false.

What Happens in the Selection

Inside the if/else statement, the script will wait seven seconds, after setTimeout finishes, it will fade the paragraph out.

Then the counter is either set back to 1 (if you are on the last paragraph), or incremented, if the last paragraph is not the one showing.

Then the contentRotator calls itself. This is the recursive part of the program.

```
if( $(this).is("#container p:last-child") ){
    //wait seven seconds
    //fade the paragraph out
    //set the counter back to 1
    //run contentRotator again
}
else{
    //wait seven seconds
    //fade the paragraph out
    //increment the counter
    //run contentRotator again
}
```

Else Statement

Start with the else statement, since that is what will happen most the time.

Add the setTimeout method which will run a function after waiting seven seconds.

Everything else for the else statement will go inside the curly braces for the callback function that runs when seven seconds are up.

```
if( $(this).is("#container p:last-child") ){  
    //wait seven seconds  
    //fade the paragraph out  
    //set the counter back to 1  
    //run contentRotator again  
}  
else{  
    setTimeout( function(){ /*Do stuff here*/ }, 7000 );  
}
```

Fade Out the Paragraph

```
else{
  setTimeout( function(){

    $('#container p:nth-child(${counter})').fadeOut(2000);

  }, 7000 );
}
```

After the seven seconds are up, fade out the paragraph over the course of two seconds.

When that animation is over, you need to increment the counter and then call `contentRotator()` again.

To do this after the `fadeOut` is done, you need another callback function.

Can you add that and do the rest?

Add the Final Callback Function

```
else{
  setTimeout( function(){

    $('#container p:nth-child(${counter})').fadeOut(2000, function(){

      counter++;
      contentRotator();

    });

  }, 7000 );
}
```

Add the final callback function and now the else should be working. If you're not showing the last paragraph, wait seven seconds, then fade out the paragraph. When that animation is over, increment the counter and run the program again.

The if statement is the same, except the counter gets set back to 1. Can you do that one yourself?

The If Statement

```
if( $(this).is("#container p:last-child") ){
    setTimeout( function(){

        $('#container p:nth-child(${counter})').fadeOut(2000, function(){

            counter = 1;
            contentRotator();

        } );

    }, 7000 );
}
```

You can see the if statement is the same, except that if you are on the final paragraph, set the counter back to one, so that when the program calls itself again it goes back and gets the first paragraph.

Summary

Hopefully, by doing this project, you saw that by using callback functions, timers and recursion you can make a content rotator like this work without a lot of code.