**Using setInterval() and setTimeout() with JavaScript**

[setInterval()](https://www.w3schools.com/jsref/met_win_setinterval.asp) and [setTimeout()](https://www.w3schools.com/jsref/met_win_settimeout.asp) functions are the quickest solutions to loop functions with JavaScript, often used for a simple animation effects.

For example, setInterval(function(){functionA()}, 1000) executes functionA() every second. On the other hand, setTimeout(function(){functionA()}, 1000) executes functionA() once after waiting for 1 seconds.

Both can be stopped by [clearInterval()](https://www.w3schools.com/jsref/met_win_clearinterval.asp) and [clearTimeout()](https://www.w3schools.com/jsref/met_win_cleartimeout.asp) by passing the returned values from respective functions.

<pre class=" toolbar:2 lang:js font-size:12px decode:true" >

var id = setInterval(function(){functionA()}, 1000) executes functionA();

clearInterval(id);

var id = setTimeout(function(){functionA()}, 1000) executes functionA();

clearTimeout(id);

</pre>

setTimeout() can be used to loop functions recursively. The difference from using setInterval() is that the recursive way with setTimeout() calls the function next time only when the first function completes the execution. setInterval() calls the function next time regardless of the execution completion as soon as the time is up. If your function needs to complete the execution before the subsequent call (e.g. AJAX call), it is better to use setTimeout() recursively.

<pre class=" toolbar:2 lang:js font-size:12px decode:true" >

</pre>

Note that neither timer does not guarantee 100% precision. If the precision is critical, they are not recommended to use.

Let’s have a look at examples.

1. **Counter**

The function below display the count from 10 to 0 and goes back to 10. setInterval() is called when the window is loaded.

<pre class=" toolbar:2 lang:js font-size:12px decode:true" >

</pre>

1. **Button Flashing**

Cheap, but effective, trick to flash the button by changing the button color through CSS.

<pre class=" toolbar:2 lang:js font-size:12px decode:true" >

</pre>

1. **AJAX call loop**

As I mentioned before, when you want to ensure the function to complete execution and can be lenient to the timing of execution, it is better to use setTimeout() recursively. In this example, the function is doing AJAX call to get data from an API endpoint (using [JSONPlaceholder](https://jsonplaceholder.typicode.com/)). In this way, it only executes the function next time after AJAX call completes.

<pre class=" toolbar:2 lang:js font-size:12px decode:true" >

</pre>

1. **Header Text Slider**

This moves a header text from left to right by changing the left CSS property. Another easy and effective trick. The similar animation reference can be found [here](https://www.w3schools.com/howto/howto_js_animate.asp).

<pre class=" toolbar:2 lang:js font-size:12px decode:true" >

</pre>

1. **Header Animation**

When you first visit the website, it loads the first header text. Then subsequently, changes text every second. After 3 seconds, the animation finishes until the page is loaded again. This is another easy and effective trick that many websites use.

<pre class=" toolbar:2 lang:js font-size:12px decode:true" >

</pre>