Controlling Devices Over the Internet > Introduction to Javascript

Built-in Javascript functions

In this lesson, we will be learning about Javascript's awesome SetTimeout and SetInterval function.

1. setTimeout()

The Set timeout function executes a function or set of code statements once after a certain interval of time. It is useful in cases where we want some code to execute once after an interval and not immediately.

Suppose, we want to build a webpage which allows the user to press a button and after 2 seconds, we want to display an alert to the user.

For this, we will need to create a simple HTML page with a button. The code for showing a simple HTML page with a button is as follows below,

```
<!DOCTYPE html>
<html>
   <head>
      <title>Javascript Timeout</title>
   </head>
   <body>
      <h1>Javascript Timeout function</h1>
      <button>Hello guys</button>
   </body>
</html>
```

Create a new file called as js_timeout.html on your desktop and enter the code as shown above. Save the file and open the file via your internet browser like Google Chrome or Mozilla Firefox. It will look similar to this, (For legibility, I have zoomed in on the page.)

The page contains a button which has the text “Hello guys”. Notice that this button does not do anything when we click on it as we have not associated any behavior with it till now.

For associating a behavior with the button, we will add an attribute called as onclick which tells the button what to do when the user clicks on the button. In this case, we want it to execute our function “say_hello()”. The say_hello() function displays an alert to the user after 3 seconds when the user clicks on the button.

Create a new tag in the same document under the tag of the document and add the code for displaying the alert. We will use the setTimeout() function of Javascript to display the alert to the user after 3 seconds.

```
<!DOCTYPE html>
 <html>
 <head>
 <title>Javascript Timeout</title>
 </head>

 <body>
 <h1>Javascript Timeout function</h1>
 <button  onclick="say_hello()">Hello guys</button>
 </body>

 <script>
 function  say_hello() {
 setTimeout(function(){ alert("Hello"); }, 3000);
 }
 </script>
 </html>
```

Inside the say_hello() function, we have given an instruction to the browser to display the alert “Hello” after 3000 milliseconds which is 3 seconds.

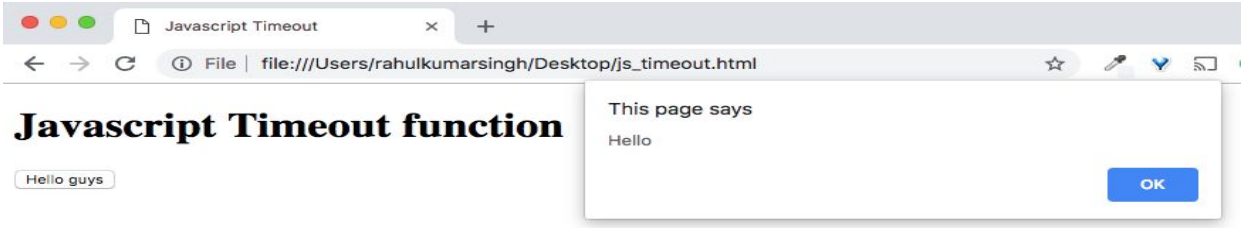
1. setInterval()

The Set interval function executes a function or code statements repeatedly after a set interval. It is useful in cases where we want some code to execute repeatedly at a set interval.
To demonstrate the functionality of the setInterval() function, we will be creating a simple counter which increments a counter value and updates it on the browser display after every second.

First, create a new file called as js_interval.html and create a view which shows a simple heading, a button and an empty div(which we will use to display the counter). The code looks like this,

```
<!DOCTYPE html>
<html>
   <head>
      <title>Javascript Interval</title>
   </head>
   <body>
      <h1>Javascript Interval function</h1>
      <button>Start Timer</button>
      <br>
      <div></div>
   </body>
</html>
```

Save the file and open it in your favorite browser to see what you have built.



Now, we want the button to execute a function to start the timer when the user clicks on the “Start Timer” button. For this, we will use the “onclick” attribute of the button to execute our function which is start_timer().

```
<!DOCTYPE html>
<html>
   <head>
      <title>Javascript Interval</title>
   </head>
   <body>
      <h1>Javascript Interval function</h1>
      <button onclick="start_timer()">Start Timer</button>
      <br>
      <div id="counter"></div>
   </body>
   <script>
      var counter = 0;
      function start_timer(){
        setInterval(increment_counter, 1000);
      }

        function  increment_counter() {
             counter_element = document.getElementById("counter");
        counter_element.innerHTML = counter;
        counter = counter + 1;
      }
   </script>
```



```
</html>
```

Next, we have given an ID to the element that we have created with “counter”. This ID uniquely identifies the div element on the page and this will be the place where the counter will be displayed to the user.

Inside the script tag, we have created a global variable called as “counter” with the initial value of 0.

The function start_timer() which gets executed after the button is clicked is defined here. This function just has a single line which is the setInterval() function which has two arguments currently i.e. the name of the function to execute along with the interval in milliseconds.

The function increment_counter() does the actual work of displaying the counter to the user on the page. The function first finds the elements which have the id as “counter” with the help of the “document.getElementById()” function and sets the HTML content to the counter value using the “innerHTML” function. Next, the function increments the counter value by one and exits.

Since the counter variable is global, the increment_counter() function modifies the global value of the counter variable, sets the HTML content and increments the global counter.

The end result looks like this,