JavaScript setInterval

Summary: in this tutorial, you will learn how to use the JavaScript `setInterval()` to repeatedly call a function with a fixed delay between each call.

Introduction to JavaScript setInterval()

The `setInterval()` is a method of the `window` object. The `setInterval()` repeatedly calls a function with a fixed delay between each call.

The following illustrates the syntax of the `setInterval()`:

```
let intervalID = setInterval(callback, delay,[arg1, arg2, ...]);
```

In this syntax:

- The `callback` is a callback function to be executed every `delay` milliseconds.
- The `delay` is the time (in milliseconds) that the timer should delay between executions of the callback function.
- The `arg1`, ... `argN` are the arguments that are passed to the callback function.

The `setInterval()` returns a numeric, non-zero number that identifies the created timer. You can pass the `intervalID` to the `clearInterval()` to cancel the timeout.

Note that the `setInterval()` works like the `setTimeout()` but it repeatedly executes a callback once every specified delay.

JavaScript setInterval() example

The following example uses the `setInterval()` and `clearInterval()` to change the color of a heading once a second once you press the Start button. If you stop the button, the `clearInterval()` will cancel the timeout.

```html
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8" />
  <title>JavaScript setInterval Demo</title>

  <script>
    let intervalID;

    function toggleColor() {
      let e = document.getElementById('flashtext');
      e.style.color = e.style.color == 'red' ? 'blue' : 'red';
    }

    function stop() {
      clearInterval(intervalID);
    }

    function start() {
      intervalID = setInterval(toggleColor, 1000);
    }
  </script>
</head>

<body>
  <p id="flashtext">JavScript setInterval Demo</p>
  <button onclick="start()">Start</button>
  <button onclick="stop()">Stop</button>

</body>
</html>
```

Output:

JavaScript setInterval Demo

Start Stop

Summary

- The `setInterval()` repeatedly calls a function once a fixed delay between each call.

- The `setInterval()` returns a `timeoutID` that can be passed to the `clearInterval()` to cancel the timeout.