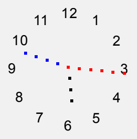
JavaScript Timing Events

❮ Previous Next ❯



JavaScript can be executed in time-intervals.

This is called timing events.

Timing Events

The window object allows execution of code at specified time intervals.

These time intervals are called timing events.

The two key methods to use with JavaScript are:

- setTimeout(*function, milliseconds*)
  Executes a function, after waiting a specified number of milliseconds.
- setInterval(*function, milliseconds*)
  Same as setTimeout(), but repeats the execution of the function continuously.

The setTimeout() and setInterval() are both methods of the HTML DOM Window object.

The setTimeout() Method

```
window.setTimeout(function, milliseconds);
```

The **window.setTimeout()** method can be written without the window prefix.

The first parameter is a function to be executed.

The second parameter indicates the number of milliseconds before execution.

Example

Click a button. Wait 3 seconds, and the page will alert "Hello":

```
<button onclick="setTimeout(myFunction, 3000)">Try it</button>

<script>
function myFunction() {
    alert('Hello');
}
</script>
```

Try it Yourself »

How to Stop the Execution?

The clearTimeout() method stops the execution of the function specified in setTimeout().

```
window.clearTimeout(timeoutVariable)
```

The **window.clearTimeout()** method can be written without the window prefix.

The clearTimeout() method uses the variable returned from setTimeout():

```
myVar = setTimeout(function, milliseconds);
clearTimeout(myVar);
```

If the function has not already been executed, you can stop the execution by calling the clearTimeout() method:

Example

Same example as above, but with an added "Stop" button:

```
<button onclick="myVar = setTimeout(myFunction, 3000)">Try it</button>

<button onclick="clearTimeout(myVar)">Stop it</button>
```

Try it Yourself »

The setInterval() Method

The setInterval() method repeats a given function at every given time-interval.

```
window.setInterval(function, milliseconds);
```

The **window.setInterval()** method can be written without the window prefix.

The first parameter is the function to be executed.

The second parameter indicates the length of the time-interval between each execution.

This example executes a function called "myTimer" once every second (like a digital watch).

Example

Display the current time:

```
var myVar = setInterval(myTimer, 1000);

function myTimer() {
    var d = new Date();
    document.getElementById("demo").innerHTML = d.toLocaleTimeString();
}
```

Try it Yourself »

There are 1000 milliseconds in one second.

How to Stop the Execution?

The clearInterval() method stops the executions of the function specified in the setInterval() method.

```
window.clearInterval(timerVariable)
```

The **window.clearInterval()** method can be written without the window prefix.

The clearInterval() method uses the variable returned from setInterval():

```
myVar = setInterval(function, milliseconds);
clearInterval(myVar);
```

Example

Same example as above, but we have added a "Stop time" button:

```
<p id="demo"></p>

<button onclick="clearInterval(myVar)">Stop time</button>

<script>
var myVar = setInterval(myTimer, 1000);
function myTimer() {
    var d = new Date();
    document.getElementById("demo").innerHTML = d.toLocaleTimeString();
}
</script>
```

Try it Yourself »

More Examples

Another simple timing

A clock created with a timing event

❮ Previous Next ❯

COLOR PICKER

HOW TO

Tabs