

Javascript timing events

Last Updated giovedì, 29 novembre 2007

With Javascript you can execute functions or statements not immediately but after a specified time interval. The two methods (of the HTML DOM Window object) are:

- setTimeout() - execute a code some time in the future
- clearTimeout() - cancel the setTimeout()

setTimeout()

Syntax: `var t=setTimeout(function, delay*)` *(in milliseconds: 2 seconds = 2 * 1000)

setTimeout() returns a value we have stored in the 't' variable. You can refer to this value if you want to cancel the setTimeout().

The first parameter is a string containing a Javascript statement or function.

The second parameter specifies how many milliseconds you want to delay the your function or statement.

The example below shows a message after a 5 seconds delay `var t=setTimeout("alert('5 seconds')",5000);`
clearTimeout()

Syntax: `clearTimeout(setTimeout_variable)` example: <html>

<head>

<script type="text/javascript">

var t

```
function time(){
    setTimeout("alert()",5000);
}
```

```
function stoptime(){
    clearTimeout(t);
}
```

</script>

</head>

<form>

<input type="button" value="count" onClick="time()">

<input type="button" value="stop" onClick="stoptime()">

</form>

</html> This example shows the two timing functions: the first button starts a 5 second delay before an alert message and the second one stops this counter cleaning the operation.