

# The right way to use setInterval() and setTimeout() in Javascript

July 14, 2012

Most of the [tutorials](#) and [examples](#) for using [setInterval\(\)](#) and [setTimeout\(\)](#) describe the first parameter (which represents the function to execute) as a string, like this,

```
setTimeout("count()", 1000);
```

Even the normally reliable [O'Reilly men](#) do it [this way](#), too. [Stephen Chapman](#) is one of the few who [gets it right](#).

While this technique works, it has two problems. First, if the function you want to pass has parameters of its own, escaping and formatting them into a string properly is a mess, even in a simple example like this one,

```
setTimeout('window.alert(\'Hello!\')', 2000);
```

and it can get [even more complicated](#).

Second, this technique uses [eval\(\)](#) to execute the function, [which is evil](#), and to be avoided.<sup>1</sup>

Using a [javascript closure](#) is a better approach:

```
setTimeout(function () {  
    // do some stuff here  
}, 1000);
```

This makes sending parameters to the underlying function easy,

```
setTimeout(function (a, b, c) {  
    // do some stuff here  
}, 1000);
```

and it avoids using [eval\(\)](#) entirely.

[1] While not exactly related, there's more in this vein at the hilarious [\(the \(axis-of \(eval\)\)\)](#) blog, which I found on [news.lispnyc.org](#) recently.

---

Archived from the original at <http://denis.papathanasiou.org/>

 Bitcoin Donate: [14TM4ADKJbaGEi8Qr8dh4KfPBQmjTshkZ2](https://www.blockchain.com/transaction/14TM4ADKJbaGEi8Qr8dh4KfPBQmjTshkZ2)