# Eventos de sincronização de JavaScript

[❮ Anterior](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/js_popup.asp&usg=ALkJrhgczz8rLmCYgWSmDw9akrwSbczZoQ)[Próximo ❯](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/js_cookies.asp&usg=ALkJrhhnM3UmMP4BmcyDGpoYnkeygQ8i4Q)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | O JavaScript pode ser executado em intervalos de tempo.  Isso é chamado de eventos temporários. |

## Eventos de temporização

O objeto da janela permite a execução do código em intervalos de tempo especificados.

Esses intervalos de tempo são chamados eventos de temporização.

Os dois métodos principais para usar com JavaScript são:

* setTimeout ( função, milissegundos )   
  Executa uma função, depois de esperar um número especificado de milissegundos.
* setInterval ( função, milissegundos )   
  O mesmo que setTimeout (), mas repete a execução da função de forma contínua.

O setTimeout () e setInterval () são ambos os métodos do objeto HTML DOM Window.

## O método setTimeout ()

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

O método **window.setTimeout ()** pode ser escrito sem o prefixo da janela.

O primeiro parâmetro é uma função a ser executada.

O segundo parâmetro indica o número de milissegundos antes da execução.

### Exemplo

Clique em um botão. Aguarde 3 segundos e a página irá alertar "Olá":

<button onclick="setTimeout(myFunction, 3000)">Try it</button>  
  
<script>  
function myFunction() {  
    alert('Hello');  
}  
</script>

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_timing1&usg=ALkJrhiVe0s7PzHXipgJU3VpYUS2SXWf2g)

## Como parar a execução?

O método clearTimeout () interrompe a execução da função especificada em setTimeout ().

window.clearTimeout( timeoutVariable )

O método **window.clearTimeout ()** pode ser escrito sem o prefixo da janela.

O método clearTimeout () usa a variável retornada de setTimeout ():

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

Se a função ainda não foi executada, você pode parar a execução chamando o método clearTimeout ():

### Exemplo

O mesmo exemplo acima, mas com um botão "Parar" adicionado:

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

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_settimeout2&usg=ALkJrhhywW3LWq-MuCjL0lY-HalKL65LEA)

## O método setInterval ()

O método setInterval () repete uma determinada função em cada intervalo de tempo.

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

O método **window.setInterval ()** pode ser escrito sem o prefixo da janela.

O primeiro parâmetro é a função a ser executada.

O segundo parâmetro indica o comprimento do intervalo de tempo entre cada execução.

Este exemplo executa uma função chamada "myTimer" uma vez por segundo (como um relógio digital).

### Exemplo

Exibir a hora atual:

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

[Tente você mesmo "](https://translate.googleusercontent.com/translate_c?depth=1&hl=pt-BR&ie=UTF8&prev=_t&rurl=translate.google.com.br&sl=en&sp=nmt4&tl=pt-BR&u=https://www.w3schools.com/js/tryit.asp%3Ffilename%3Dtryjs_setinterval2&usg=ALkJrhgFfiKJJXX3oN_Qg4W_n7DS3zgskw)

Há 1000 milissegundos em um segundo.

## Como parar a execução?

O método clearInterval () interrompe as execuções da função especificada no método setInterval ().

window.clearInterval( timerVariable )

O método **window.clearInterval (** **)** pode ser escrito sem o prefixo da janela.

O método clearInterval () usa a variável retornada de setInterval ():

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

### Exemplo

O mesmo exemplo acima, mas adicionamos um botão "Parar tempo":

<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>