

HomeHTMLCSSJAVASCRIPTSQLPYTHONPHPBOOTSTRAP

Navegação DOM

Nós DOM

Coleções DOM

Listas de nós DOM

JS Browser BOM

Janela JS

Tela JS

Localização JS

História JS

JS Navigator

Alerta pop-up JS

JS Timing

Cookies JS

JS Web APIs

Introdução à API da Web

API de formulários da web

API de histórico da web

grammarty

Get Grammarly It's Free

Be best English writer you can be!

the best

Eventos de cronometragem de JavaScript

< Anterior

Próximo >

11121

102

93

84

75

6

JavaScript pode ser executado em intervalos de tempo.

Isso é chamado de eventos de tempo.

Estude online

Mantenha o equilíbrio de

Cresce Exponencialmente

Diariamente | Feedforward

para o futuro que você merece | Torne-se o CEO da sua vida

Atlantic Int'l University

Abrir

COLOR PICKER

COMO NÓS

f

o

in

Obtenha a certificação completando um curso hoje!

W3schools

W3

CERTIFIED

iniciar

JOGO DE CÓDIGOS

Jogar um jogo

Mestrados Online

FUNIBER

Estude a distância. Temos uma ampla gama de programas de formação

FUNIBER

Abrir

Eventos de cronometragem

O `window` objeto permite a execução de código em intervalos de tempo especificados.

Esses intervalos de tempo são chamados de eventos de temporização.

Os dois métodos principais a serem usados com JavaScript são:

- `setTimeout(function, milliseconds)`
Executa uma função, depois de esperar um número especificado de milissegundos.
- `setInterval(function, milliseconds)` O mesmo que `setTimeout()`, mas repete a execução da função continuamente.

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

O método setTimeout ()

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

O `window.setTimeout()` método 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 alertará "Hello":

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

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

Tente você mesmo "

Como parar a execução?

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

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

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

O `clearTimeout()` método 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 `clearTimeout()` método:

Exemplo

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:

```
setInterval(myTimer, 1000);

function myTimer() {
  const 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

Mesmo exemplo acima, mas adicionamos um botão "Tempo de parada":

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

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

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

Tente você mesmo "

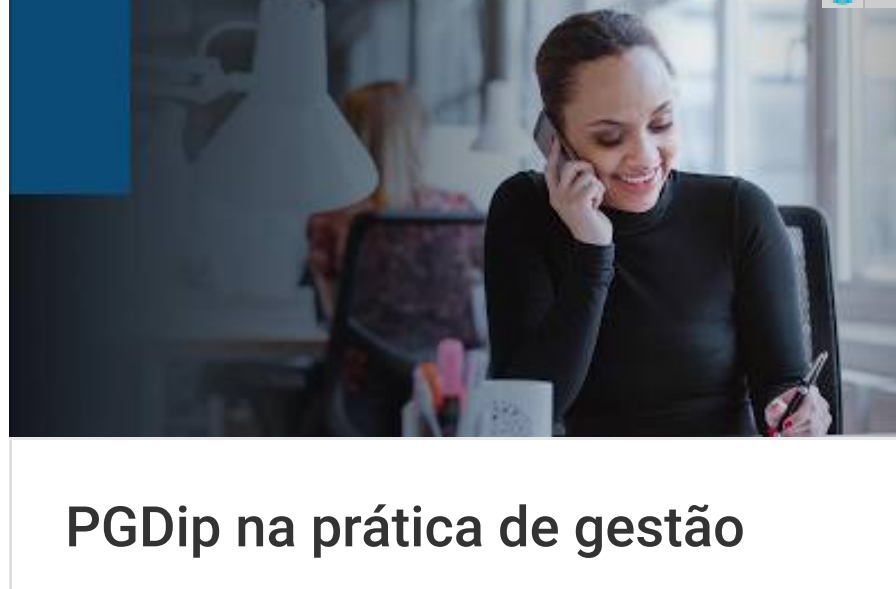
Mais exemplos

[Outro momento simples](#)

[Um relógio criado com um evento de cronometragem](#)

< Anterior

Próximo >



PGDip na prática de gestão

UCT GSB

[Abrir >](#)

Reportar erro

Fórum

Cerca de

Comprar

Principais tutoriais

Tutorial HTML Tutorial
CSS Tutorial
JavaScript
Como fazer Tutorial
SQL Tutorial
Python Tutorial
W3.CSS Tutorial
Bootstrap Tutorial
PHP Tutorial
Java Tutorial
C ++ Tutorial
jQuery Tutorial

Referências principais

HTML Reference
CSS Reference
JavaScript Reference
SQL Reference
Python Reference
W3.CSS Reference
Bootstrap Reference
PHP Reference
HTML Colors
Java Reference
Angular Reference
jQuery Reference

Top Examples

HTML Examples
CSS Examples
JavaScript Examples
How To Examples
SQL Examples
Python Examples
W3.CSS Examples
Bootstrap Examples
PHP Examples
Java Examples
XML Examples
jQuery Examples

Web Courses

HTML Course
CSS Course
JavaScript Course
Front End Course
SQL Course
Python Course
PHP Course
jQuery Course
Java Course
C++ Course
C# Course
XML Course

Get Certified »

W3Schools is optimized for learning and training. Examples might be simplified to improve reading and learning. Tutorials, references, and examples are constantly reviewed to avoid errors, but we cannot warrant full correctness of all content. While using W3Schools, you agree to have read and accepted our terms of use, cookie and privacy policy.

Copyright 1999-2021 by Refsnes Data. All Rights Reserved.
W3Schools is Powered by W3.CSS.

