

TUTORIAIS JAVASCRIPT

Como utilizar o Objeto Window Timing

Copyright 2013 – Todos os Direitos Reservados
Jorge Eider F. da Silva

Proibida a reprodução deste documento no todo ou em parte por quaisquer meios, seja digital, eletrônico ou impresso sem a expressa autorização do autor por escrito. Os infratores serão punidos de acordo com a Lei.

Como utilizar o Objeto Window Timing

Introdução

Em JavaScript é possível executar algum código em intervalos de tempo pré-estabelecidos. Isso é chamado de eventos de tempo. É muito fácil utilizar eventos de tempo em JavaScript.

Os dois métodos utilizados para isso são:

- **setInterval()** – executa uma função várias vezes de acordo com intervalos de tempo especificados.
- **setTimeout()** – executa uma função uma vez, após esperar um número especificado de milissegundos.

Vejamos como utilizar cada um desses métodos:

O método setInterval()

Esse método esperará um número especificado de milissegundos, e então passa a executar uma função especificada, e continuará a executar a função uma vez a cada intervalo de tempo dado.

Sua sintaxe é a seguinte:

```
window.setInterval("função javascript",milissegundos);
```

- O primeiro parâmetro deverá ser uma função.
- O segundo parâmetro deverá ser um valor numérico que representa o intervalo de tempo entre cada execução. Existem 1000 milissegundos em um segundo.

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

Vejamos um exemplo prático:

Exemplo 1

Nesse exemplo vamos utilizar o método **setInterval()** para mostrar uma caixa de diálogo aparecer a cada 3 segundos.

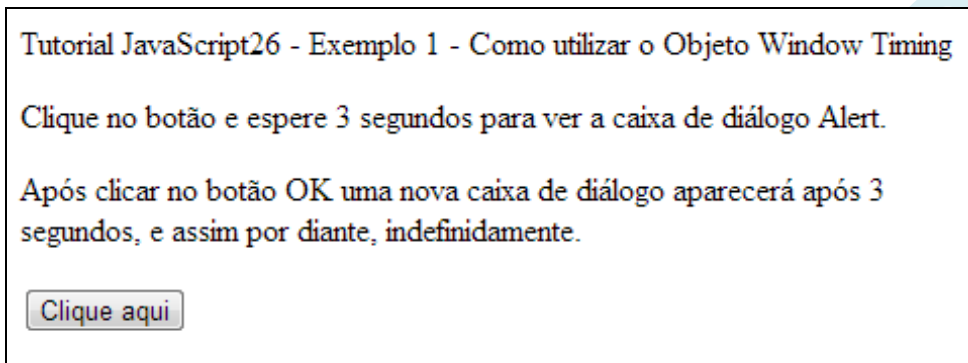
Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js26-exemplo1.html**.

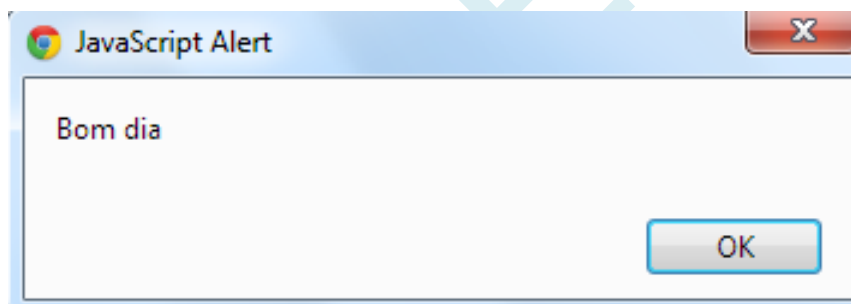
```
<!DOCTYPE html>
<html>
Tutorial JavaScript26 - Exemplo 1 - Como utilizar o Objeto Window Timing
<head>
  <title>Tutorial JavaScript26 - Exemplo 1 - Como utilizar o Objeto Window Timing</title><p>
</head>
<body>
<p>Clique no botão e espere 3 segundos para ver a caixa de diálogo Alert.</p>
```

```
<p>Após clicar no botão OK uma nova caixa de diálogo aparecerá após 3<br>segundos, e assim por diante, indefinidamente.</p>
<button onclick="tempo()">Clique aqui</button>
<script>
function tempo()
{
setInterval(function(){alert("Bom dia")},3000);
}
</script></body>
</html>
```

2. Após a execução do código no browser será mostrado o seguinte:



3. Quando o botão **Clique aqui** for clicado será mostrada a seguinte caixa de diálogo:



Esse exemplo mostra como o método **setInterval()** funciona e que você poderá utilizá-lo para várias finalidades em uma página web.

Exemplo 2

Nesse exemplo vamos utilizar o método **setInterval()** para mostrar a hora atual. O método será usado para executar a função uma vez a cada 1 segundo, tal como um relógio digital.

Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js26-exemplo2.html**.

```
<!DOCTYPE html>
<html>
Tutorial JavaScript26 - Exemplo 2 - Como utilizar o Objeto Window Timing
<head>
  <title>Tutorial JavaScript26 - Exemplo 2 - Como utilizar o Objeto Window Ti-
ming</title><p>
</head>
<body>
<p>Relógio digital mostrando a hora do sistema:</p>
```

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

<script>
var x=setInterval(function(){tempo()},1000);

function tempo()
{
var d=new Date();
var t=d.toLocaleTimeString();
document.getElementById("demo").innerHTML=t;
}
</script>
</body>
</html>
```

2. Após a execução do código no browser será mostrado o seguinte:

Tutorial JavaScript26 - Exemplo 2 - Como utilizar o Objeto Window Timing

Relógio digital mostrando a hora do sistema:

9:13:34 AM

Quando parar uma execução

O método **clearInterval()** é utilizado interromper quaisquer execuções da função especificada no método **setInterval()**.

Sua sintaxe é a seguinte:

```
window.clearInterval(intervalVariavel);
```

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

Quando você for usar o método **clearInterval()** você deverá utilizar uma variável global conforme mostra o exemplo abaixo:

```
mVar=setInterval("javascript function",milliseconds);
```

Dessa forma você poderá interromper a execução chamando o método **clearInterval()**. Vejamos como fazer com um exemplo prático.

Exemplo 3

Nesse exemplo vamos utilizar o exemplo anterior acrescentado de um botão para que se possa interromper a execução do relógio.

Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js26-exemplo3.html**.

```
<!DOCTYPE html>
<html>
Tutorial JavaScript26 - Exemplo 3 - Como utilizar o Objeto Window Timing
<head>
```

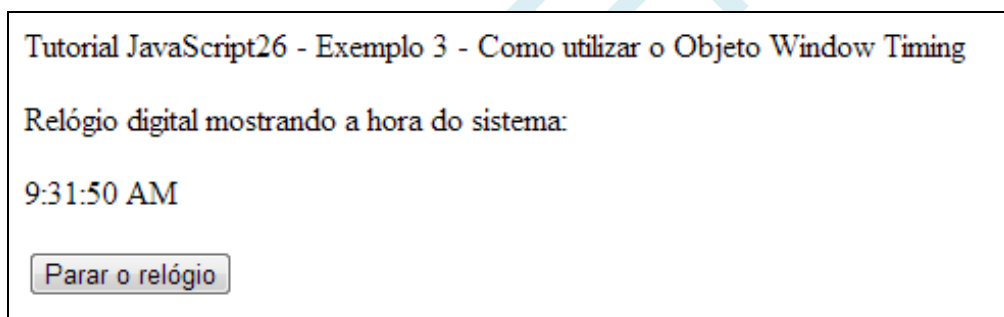
```

<title>Tutorial JavaScript26 - Exemplo 3 - Como utilizar o Objeto Window Timing</title><p>
</head>
<body>
<p>Relógio digital mostrando a hora do sistema:</p>
<p id="demo"></p>
<button onclick="pararTempo()">Parar o relógio</button>

<script>
var mVar=setInterval(function(){tempo()},1000);
function tempo()
{
var d=new Date();
var t=d.toLocaleTimeString();
document.getElementById("demo").innerHTML=t;
}
function pararTempo()
{
clearInterval(mVar);
}
</script>
</body>
</html>

```

2. Após a execução do código no browser será mostrado o seguinte:



O método setTimeout()

Esse método esperará um número específico de milissegundos, e então executará a função especificada.

Sua sintaxe é a seguinte:

```

window.setTimeout("javascript function",milliseconds);

```

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

O primeiro parâmetro desse método deverá ser uma função. O segundo parâmetro indicar quantos milissegundos, a partir do momento atual, você quer executar o primeiro parâmetro.

Vejamos um exemplo prático:

Exemplo 4

Nesse exemplo vamos utilizar o método **setTimeout()** para mostrar uma caixa de diálogo 3 segundos após o botão ser clicado.

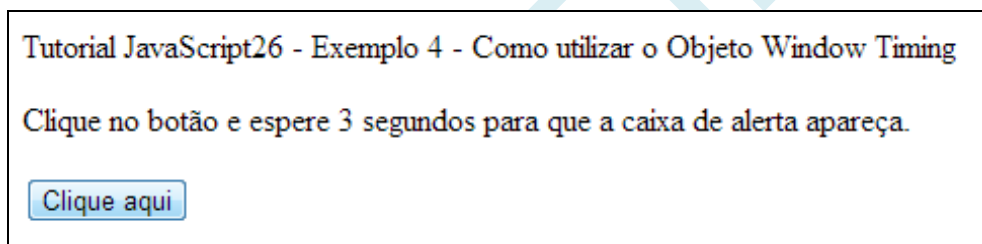
Para isso:

1. Digite o código abaixo no seu editor de texto e salve-o como: **js26-exemplo4.html**.

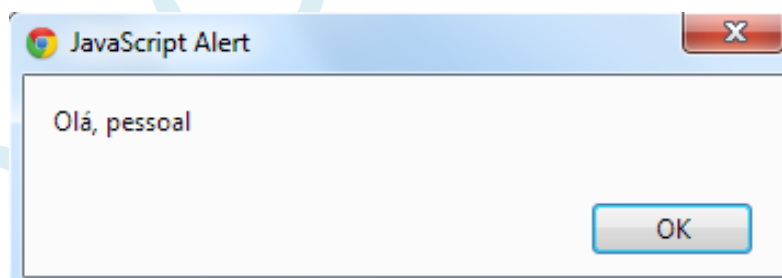
```
<!DOCTYPE html>
<html>
Tutorial JavaScript26 - Exemplo 4 - Como utilizar o Objeto Window Timing
<head>
  <title>Tutorial JavaScript26 - Exemplo 4 - Como utilizar o Objeto Window Ti-
ming</title><p>
</head>
<body>
<p>Clique no botão e espere 3 segundos para que a caixa de alerta apareça.</p>
<button onclick="tempo()">Clique aqui</button>

<script>
function tempo()
{
setTimeout(function(){alert("Olá, pessoal")},3000);
}
</script></body>
</html>
```

2. Após a execução do código no browser será mostrado o seguinte:



3. Após o botão **Clique aqui** ser clicado será mostrada a seguinte caixa de diálogo:



Como parar uma execução

O método **clearTimeout()** é utilizado interromper a execução da função especificada no método **setTimeout()**.

Sua sintaxe é a seguinte:

```
window.clearTimeout(timeoutVariavel);
```

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

Quando você for usar o método **clearTimeout()** você deverá utilizar uma variável global conforme mostra o exemplo abaixo:

```
mVar=setTimeout("javascript function",milliseconds);
```

Dessa forma você poderá interromper a execução chamando o método **clearTimeout()**.

Vejamos como fazer isso com um exemplo prático.

Exemplo 5

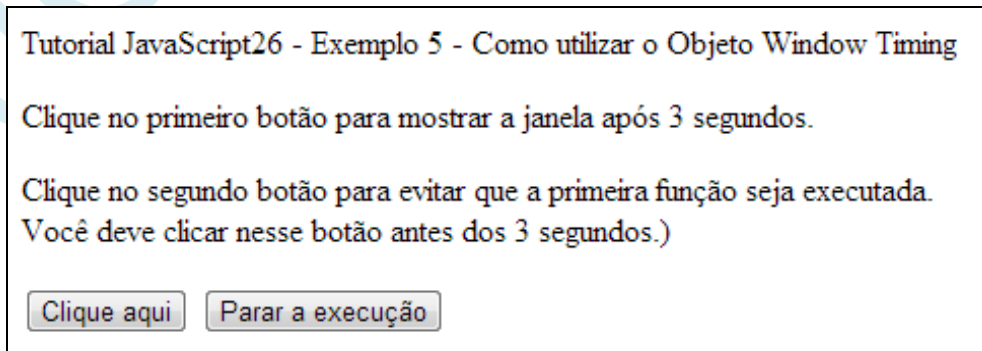
Nesse exemplo vamos utilizar o exemplo anterior acrescentado de um botão para que se possa interromper a execução do método **setTimeout()**.

Para isso:

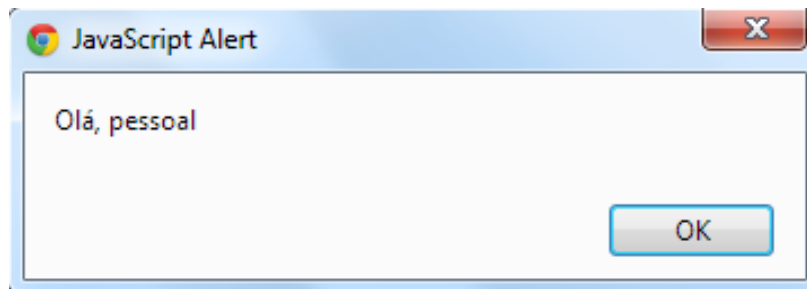
1. Digite o código abaixo no seu editor de texto e salve-o como: **js26-exemplo5.html**.

```
<!DOCTYPE html>
<html>
Tutorial JavaScript26 - Exemplo 5 - Como utilizar o Objeto Window Timing
<head>
  <title>Tutorial JavaScript26 - Exemplo 5 - Como utilizar o Objeto Window Ti-
ming</title><p>
</head>
<body>
<p>Clique no primeiro botão para mostrar a janela após 3 segundos.</p>
<p>Clique no segundo botão para evitar que a primeira função seja executada.<br>
Você deve clicar nesse botão antes dos 3 segundos.</p>
<button onclick="mostrarJanela()">Clique aqui</button>
<button onclick="pararExecucao()">Parar a execução</button>
<script>
var mVar;
function mostrarJanela()
{
mVar=setTimeout(function(){alert("Olá, pessoal")},3000);
}
function pararExecucao()
{
clearTimeout(mVar);
}
</script>
</body>
</html>
```

2. Após a execução do código no browser será mostrado o seguinte:



3. Após o botão **Clique aqui** ser clicado será mostrada a seguinte caixa de diálogo:



4. Para interromper a execução, é preciso clicar no botão Parar a execução antes do tempo especificado, nesse caso, três segundos.

Todos os métodos explicados nesse tutorial são muito utilizados em testes e simulados de múltipla escolha.

JORGE ELDER