# JavaScript HTML DOM EventListener

[❮ 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_htmldom_events.asp&usg=ALkJrhiZQT_Zvm9jPLzoal1SNNvi_PprJg)[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_htmldom_navigation.asp&usg=ALkJrhg96aZ_EKauFuKYpnHbjbEEkhTDpA)

## O método addEventListener ()

### Exemplo

Adicione um ouvinte de eventos que dispara quando um usuário clica em um botão:

document.getElementById("myBtn").addEventListener("click", displayDate);

[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_addeventlistener_displaydate&usg=ALkJrhi_HQqqgr8oUan3srJuuo-pYOjhJw)

O método addEventListener () atribui um manipulador de eventos ao elemento especificado.

O método addEventListener () atribui um manipulador de eventos a um elemento sem substituir os manipuladores de eventos existentes.

Você pode adicionar muitos manipuladores de eventos a um elemento.

Você pode adicionar muitos manipuladores de eventos do mesmo tipo a um elemento, ou seja, dois eventos "clicar".

Você pode adicionar ouvintes de eventos a qualquer objeto DOM não apenas elementos HTML. ou seja, o objeto da janela.

O método addEventListener () facilita o controle de como o evento reage a borbulhar.

Ao usar o método addEventListener (), o JavaScript é separado da marcação HTML, para uma melhor legibilidade e permite que você adicione ouvintes de eventos mesmo quando você não controla a marcação HTML.

Você pode remover facilmente um ouvinte de eventos usando o método removeEventListener ().

## Sintaxe

element .addEventListener( event, function, useCapture );

O primeiro parâmetro é o tipo do evento (como "clique" ou "mousedown").

O segundo parâmetro é a função que queremos chamar quando ocorre o evento.

O terceiro parâmetro é um valor booleano que especifica se deseja usar borbulhamento de eventos ou captura de eventos. Esse parâmetro é opcional.

Observe que você não usa o prefixo "on" para o evento; use "clique" em vez de "clicar".

## Adicione um manipulador de eventos a um elemento

### Exemplo

Alerta "Olá Mundo!" quando o usuário clica em um elemento:

element .addEventListener("click", function(){ alert("Hello World!"); });

[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_addeventlistener_add&usg=ALkJrhjBayojHDptSroqDEPxPuyBXOcMIw)

Você também pode se referir a uma função "nomeada" externa:

### Exemplo

Alerta "Olá Mundo!" quando o usuário clica em um elemento:

element .addEventListener("click", myFunction);  
  
function myFunction() {  
    alert ("Hello World!");  
}

[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_addeventlistener_add2&usg=ALkJrhgn_bM4SSRjjH7jz2qS2-BmYPpO5A)

## Adicione muitos manipuladores de eventos ao mesmo elemento

O método addEventListener () permite que você adicione muitos eventos ao mesmo elemento, sem substituir eventos existentes:

### Exemplo

element .addEventListener("click", myFunction);  
element .addEventListener("click", mySecondFunction);

[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_addeventlistener_add_many&usg=ALkJrhhnKawABW62sOp1WyzAHjBbm0pRmw)

Você pode adicionar eventos de diferentes tipos ao mesmo elemento:

### Exemplo

element .addEventListener("mouseover", myFunction);  
element .addEventListener("click", mySecondFunction);  
element .addEventListener("mouseout", myThirdFunction);

[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_addeventlistener_add_many2&usg=ALkJrhjGgYi71xalO5PXnkqVAkIZolcKlg)

## Adicionar um manipulador de eventos ao objeto de janela

O método addEventListener () permite que você adicione ouvintes de eventos em qualquer objeto HTML DOM, como elementos HTML, o documento HTML, o objeto de janela ou outros objetos que suportam eventos, como o objeto xmlHttpRequest.

### Exemplo

Adicione um ouvinte de eventos que dispara quando um usuário redimensiona a janela:

window.addEventListener("resize", function(){  
    document.getElementById("demo").innerHTML = sometext ;  
});

[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_addeventlistener_dom&usg=ALkJrhjQYg_z-lypQb1i_Udc0DhHBUxomQ)

## Passando Parâmetros

Ao passar os valores dos parâmetros, use uma "função anônima" que chama a função especificada com os parâmetros:

### Exemplo

element .addEventListener("click", function(){ myFunction(p1, p2); });

[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_addeventlistener_parameters&usg=ALkJrhjNTTA8muJhfXCNSMFQJXXhQuWH4Q)

## Event Bubbling ou Event Capturing?

Existem duas formas de propagação de eventos no HTML DOM, borbulhando e capturando.

A propagação de eventos é uma maneira de definir a ordem do elemento quando ocorre um evento. Se você tem um elemento <p> dentro de um elemento <div> e o usuário clica no elemento <p>, qual evento de "clique" do elemento deve ser tratado primeiro?

Ao borbulhar, o evento do elemento mais interno é tratado primeiro e depois o externo: o evento de clique do elemento <p> é tratado primeiro, depois o evento de clique do elemento <div>.

Ao capturar o evento do elemento mais externo é tratado primeiro e depois o interior: o evento de clique do elemento <div> será tratado primeiro, depois o evento de clique do elemento <p>.

Com o método addEventListener (), você pode especificar o tipo de propagação usando o parâmetro "useCapture":

addEventListener( event , function , useCapture );

O valor padrão é falso, que usará a propagação borbulhante, quando o valor for definido como verdadeiro, o evento usa a propagação de captura.

### Exemplo

document.getElementById("myP").addEventListener("click", myFunction, true);  
document.getElementById("myDiv").addEventListener("click", myFunction, true);

[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_addeventlistener_usecapture&usg=ALkJrhhK2vSbAv5XXG9v9of0I6V0HBtQgA)

## O método removeEventListener ()

O método removeEventListener () remove os manipuladores de eventos que foram anexados com o método addEventListener ():

### Exemplo

element .removeEventListener("mousemove", myFunction);

[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_addeventlistener_remove&usg=ALkJrhiN0m19-xMhOSEM-YwcnVavTXNOMQ)

## Suporte do navegador

Os números na tabela especificam a primeira versão do navegador que suporta totalmente esses métodos.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method |  |  |  |  |  |
| addEventListener() | 1.0 | 9.0 | 1.0 | 1.0 | 7.0 |
| removeEventListener() | 1.0 | 9.0 | 1.0 | 1.0 | 7.0 |

**Nota:** Os métodos addEventListener () e removeEventListener () não são suportados no IE 8 e versões anteriores e o Opera 6.0 e versões anteriores. No entanto, para essas versões específicas do navegador, você pode usar o método attachEvent () para anexar um manipulador de eventos ao elemento e o método detachEvent () para removê-lo:

element. attachEvent (event, function);element. detachEvent (event, function);

### Exemplo

Solução cross-browser:

var x = document.getElementById("myBtn");  
if (x.addEventListener) {                    // For all major browsers, except IE 8 and earlier  
    x.addEventListener("click", myFunction);  
} else if (x.attachEvent) {                  // For IE 8 and earlier versions  
    x.attachEvent("onclick", myFunction);  
}

[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_addeventlistener_crossbrowser&usg=ALkJrhj1MTCAdBQSZXcrK5nzTWhiG0LEVg)

## Referência do objeto de evento HTML DOM

Para obter uma lista de todos os eventos HTML DOM, consulte nossa [Referência](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/jsref/dom_obj_event.asp&usg=ALkJrhjEBbugo_4p2ALXapJBMlyeZFbQxQ) completa [do objeto de evento HTML DOM](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/jsref/dom_obj_event.asp&usg=ALkJrhjEBbugo_4p2ALXapJBMlyeZFbQxQ) .