

Eventos

Um evento é uma ação que é executada por algo ou alguém (usuário ou browser). O elemento HTML que dispara o evento é conhecido como disparador.

O browser possui alguns eventos padrão, no qual não precisamos construir, como: clicar em um link e submeter um formulário.

Dentro de eventos também temos o event Handler (manipulador de eventos), que a função que vai ser executada ao disparar o evento.]

Level DOM

Os level do DOM em events são a forma de declarar o evento ao elemento disparador.

Um forma mais antiga é a conhecida com Level 0, nela utilizamos o atributo `onclick`, pode ser utilizada tanto no HTML como no JS.

Exemplo de utilização no HTML:

```
<button onclick="exemploFuncao()">Click aqui</button>
```

Também pode-se utilizar direto no JS, afim de separar as camadas de conteúdo com as de desenvolvimento. Exemplo:

```
var button = document.querySelector('button');  
  
button.onclick = exemploFuncao;
```

Observe que ao atribuir a função pelo JS não colocamos o “()” ao lado do nome da função, porque se não a função será executada e o que será definido ao atributo onclick será o retorno dessa função.

Uma outra forma de atribuir um evento é utilizando o DOM level 2, que é a forma mais recomendada. Exemplo:

```
var button = document.querySelector('button');  
  
button.addEventListener('click', funcao);
```

aqui utilizamos um método do DOM chamado `addEventListener`, que é um escutador de evento digamos. Ele irá ficar monitorando e esperando a ação passada para disparar o evento. Nesse método podemos passar dois parâmetros, o primeiro é o o tipo de ação que irá disparar o evento, o segundo é o nome da função que será executada.

As vantagens de utilizar o level 2 é que o mesmo atributo pode ter vários eventos atribuídos a ele, diferente do level 0 no qual isso não é permitido.

Propagação

Um evento ele é propagado do elemento mais interno para o elemento externo. Isso é, todo evento que tiver nesse meio caminho será disparado.

Um exemplo:

HTML

```
<div class='div1'>  
  Conteúdo externo  
  <div class='div2'>  
    Conteúdo interno  
  </div>  
</div>
```

JS

```
var div1 = document.querySelector('.div1');  
var div2 = document.querySelector('.div2');  
  
div1.addEventListener('click', function(){  
  alert('Div externa.');})  
  
div2.addEventListener('click', function(){  
  alert('Div interna.');})
```

No exemplo acima ao clicar na div interna o evento da div externa será disparado.

Podemos reverter a ordem dessa propagação, ao invés de propagar de dentro para fora, propagar de fora para dentro. Para isso basta passar um terceiro parâmetro, que vai ser um boolean com o valor `true`. Exemplo:

```
div.addEventListener('click', function, true);
```

Objeto evento

Quando se atrela uma função a um evento é possível passar um parâmetro a essa função. O parâmetro passado vira um objeto que pode ser acessado pela função. Esse objeto trás várias informações adicionais sobre o evento.

Exemplo

```
elemento.addEventListener('click', function(evento){  
    console.log(evento)  
})
```

Dentro do objeto evento teremos alguns métodos e propriedades, exemplo:

```
elemento.addEventListener('click', function(evento){  
    //Para a propagação de evento a partir desse elemento  
    evento.stopPropagation();  
    //Retorna o objeto do elemento no qual foi atrelado o evento.  
    evento.currentTarget;  
    //Retorna quem disparou o evento  
    evento.target;  
    //Retorna o elemento no qual o evento foi atrelado  
    evento.this;  
})
```