

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;
})
```

Alguns tipos de eventos

click: evento que é disparado mediante o click no elemento.

submit: evento disparado mediante ao envio de um formulário.

keydown: evento disparado mediante ao ser pressionado alguma tecla, isso ocorre antes da captura da tecla pressionada.

keyup: o evento é disparado ao soltar alguma tecla. Nesse evento a tecla pressionada já é capturada.

keypress: tem a mesma função do evento keydown, porém ele funciona apenas com teclas alfanuméricas.

input: o evento é disparado sempre que tiver entrada de dados no elemento.

Exercício 01

Validar um formulário de login (usuário e senha) ao clicar no botão de envio. Se tiver algum campo vazio deve ser mostrado um alert e impedir o envio.

```
var $inputName = document.querySelector('#inputName');
var $inputSenha = document.querySelector('#inputSenha');
var $botao = document.querySelector('.formulario__button');

$botao.addEventListener('click', function(){

    if(!$inputName.value || !$inputSenha.value){
        alert('Preencha todos os campos');
    } else {
        alert('logou');
    }

})
```

A solução a cima está correta, porém ela possui algumas brechas:

1. O botão lá no HTML deve ter o type submit. Caso contrário o formulário não será enviado.
2. O evento não deve ser associado ao botão e sim ao envio do formulário.
3. Um formulário de login não deve passar os valor pela URL ao ser submetido, para isso o method deve ser igual a post.

Código refatorado:

```
var $formulario = document.querySelector('form')
var $inputName = document.querySelector('#inputName');
var $inputSenha = document.querySelector('#inputSenha');
var $botao = document.querySelector('.formulario__button');

$formulario.addEventListener('submit', function(e){

    if(!$inputName.value || !$inputSenha.value){
        alert('Preencha todos os campos');
    }

})
```

```

        e.preventDefault();
    } else {
        alert('logou');
    }
}

}))

```

Exercício 02

Num textarea com um número máximo de caracteres permitidos, mostrar numa span o número de caracteres restantes e indicar pro usuário quando restam apenas 10 caracteres.

resultado:

```

var $areaConteudo = document.querySelector('#msg');
var $textoDoContador = document.querySelector('#textoDoContador');
var $contador = document.querySelector('#contador');

//Se o JS estiver habilitado o contador é exibido.
$textoDoContador.style.display = 'block';

var caracteresMaximo = $areaConteudo.maxLength;

atualizarContador();

$areaConteudo.addEventListener('input', contarCaracters);

function contarCaracters(e){
    var quantidadeDigitada = this.value.length;
    atualizarContador(quantidadeDigitada);
}

function atualizarContador(quantidade){
    if(quantidade){
        $contador.textContent = caracteresMaximo - quantidade;
    } else {
        $contador.textContent = caracteresMaximo
    }
}

```

Exercício 03

Permitir o envio do formulário somente após o usuário aceitar os termos do contrato (checkbox). Para este exemplo, vamos habilitar ou desabilitar o botão dependendo do

checkbox.

Resolução:

```
var $checkbox = document.querySelector('#checkbox');
var $btnEnvio = document.querySelector('#btnEnvio');

$checkbox.addEventListener('change', function(){
    if($checkbox.checked){
        $btnEnvio.removeAttribute('disabled');
    } else {
        $btnEnvio.disabled = true;
    }
})
```

Exercício 04

Atrelar eventos de click em uma coleção de botões que ao ser clicado, mostra no console o textContent do proprio botão.

Resolução:

```
var $botoes = document.querySelectorAll('button');

for(var i = 0; i < $botoes.length; i++){
    $botoes[i].addEventListener('click', function(){
        console.log(this.textContent);
    })
}
```

Exercício 05

Fazer uma pequena lista de tarefa onde o usuário pode acrescentar um novo item (li) na lista (ul) através de um input e um botão. Esta lista deve está preenchida previamente com três itens quaisquer. Acrescentar eventos de click a cada li para marcar como feito.

Resolução:

```
var $btnAdicionar = document.querySelector('.btnAdicionar');
var $inputTarefa = document.querySelector('.inputTarefa');
var $listaDeTarefa = document.querySelector('.area__listaTarefa')
var $tarefas = document.querySelectorAll('li');
```

```

$btnAdicionar.addEventListener('click', adicionarTarefa);
$inputTarefa.addEventListener('keyup', function(e){
    if(e.keyCode === 13){
        adicionarTarefa();
    }
})

$listaDeTarefa.addEventListener('click', function(e){
    if(e.target.nodeName === 'LI'){
        marcarConcluido(e.target);
        removerTarefa(e.target);
    }
})

function adicionarTarefa(){

    var li = document.createElement('li');
    var texto = document.createTextNode($inputTarefa.value);

    if(texto == ""){
        alert('Adicione um título para a tarefa');
    } else {
        li.appendChild(texto);
        $listaDeTarefa.appendChild(li);

        $inputTarefa.value = ''
    }
}

function marcarConcluido(li){
    li.classList.toggle('feito');
}

function removerTarefa(li){
    var resposta = confirm('Deseja excluir essa tarefa ?');

    if(resposta){
        $listaDeTarefa.removeChild(li);
    }
}

```