FIAP

GRADUAÇÃO

TDS
FRONT-END DESIGN ENGINEERING

Prof. Alexandre Carlos profalexandre.jesus@fiap.com.br
Prof. Luís Carlos lsilva@fiap.com.br

Javascript

EVENTOS

O QUE SÃO EVENTOS?

Toda a ação que o usuário realiza em uma página pode gerar um evento. Desde de quando ele carrega a página no navegador até quando move ou clica o mouse em um botão podemos captar (ouvir) estas ações e criar respostas para estas interações.

Em uma simples ação do usuário podemos captar um ou mais eventos diferentes. Por exemplo, somente em um elemento da tela podemos captar mais de 6 eventos diferentes só com a utilização do mouse.

COMO PODEMOS ESCUTAR UM EVENTO?

Usando um evento inline

A maneira mais simples, mas não ideal para escutar um evento é inserindo ele na tag HTML do elemento que estamos esperando a ação do usuário.

```
<h1>Eventos Javascript</h1>
<button type="button" onclick="alert('Exemplo do evento click inline')">
  Clique aqui!</button>
```

Reparem que o evento **onclick**, que aprece em forma de atributo, esta chamando uma linha de comando javascript, ou seja, quando o evento **onclick** é executado pelo usuário, quando ele clica no botão, o javascript capta (escuta) esta ação e executa a ação que está no evento.

OBS. Quando usamos eventos inline que contenham uma string no seu valor, devemos intercalar o uso de das aspas simples e dupla.

COMO PODEMOS ESCUTAR UM EVENTO?

Atribuindo uma função anônima.

Outra maneira é atribuindo a um elemento referenciado dentro do código javascript, uma função anônima, contendo as instruções que devem ser executadas. A função pode ser anônima pois só será chamada pelo evento, ou seja, quando o usuário executar a ação esperada.

HTML

```
<h1>Eventos Javascript</h1>
<button type="button" id="btnMensagem">
  Clique aqui!</button>
```

JAVASCRIPT

```
document.querySelector('#btnMensagem').onclick = function(){
  alert('Este evento está totalmente no arquivo externo!')
}
```

Reparem que o evento **onclick** esta como propriedade do elemento button. Trabalhando desta forma o código fica mais limpo, pois não temos instruções javascript dentro do nosso documento HTML.

OBS. Devemos evitar ao máximo o uso de código Javascript e CSS dentro do HTML.

COMO PODEMOS ESCUTAR UM EVENTO?

Usando a Arrow Function.

Uma maneira mais atual de criar funções anônimas é a Arrow Function, ela traz várias facilidades e é uma forma de resumirmos a função. Compare uma função simples e uma arrow function:

Function

```
document.querySelector('#btnMsn').onclick = function(){
  console.log("Ele clicou!");
}
```

Arrow Function

```
document.querySelector('#btnMsn').onclick = ()=>{
  console.log("Ele clicou!");
}
```

Arrow Function simplificada

```
document.querySelector('#btnMsn').onclick = ()=>console.log("Ele clicou!")
```

Reparem que substituímos o **function**( ){ } por ( )**=>**{ } .

COMO PODEMOS ESCUTAR UM EVENTO?

Método addEventListener()

Temos uma forma para deixar o código mais organizado, é utilizando o método **addEventListener()**. Ele é um método que recebe dois parâmetros: o evento javascript e a função que será executada quando este evento for executado.

HTML

```
<h1>Eventos Javascript</h1>
<button type="button" id="btnMensagem">
  Clique aqui!</button>
```

JAVASCRIPT

```
document.querySelector('#btnMensagem').addEventListener('click',function(){
  alert('Este evento está totalmente no arquivo externo!')
})
```

OBS. Dentro do addEventListener(), os eventos javascript não iniciam com **“on”**.

COMO PODEMOS ESCUTAR UM EVENTO?

Organizando o código com addEventListener()

Usando o método **addEventListener()**, podemos organizar as chamadas do nosso código.

HTML

```html
<h1>Eventos Javascript</h1>
<button type="button" id="btnMensagem">
  Clique aqui!</button>
```

JAVASCRIPT

```javascript
var botao = document.querySelector('#btnMensagem')
botao.addEventListener('click',clicou)
botao.addEventListener('mouseenter',entrou)

function clicou(){
  botao.innerHTML = "Clicou!"
}
function entrou(){
  botao.innerHTML = "Entrou!"
}
```

Agora temos dois eventos do mesmo elemento, organizados e logo abaixo suas funções.

COMO PODEMOS ESCUTAR UM EVENTO?

Usando o removeEventListener() para remover uma função

As vezes podemos precisar desativar uma função dependendo da ação do usuário ou por outro critério. Nestes casos podemos utilizar o método **removeEventListener().**

HTML

```html
<h1>Eventos Javascript</h1>
<button type="button" id="btnMensagem">
  Clique aqui!</button>
```

JAVASCRIPT

```javascript
var botao = document.querySelector('#btnMensagem')
botao.addEventListener('click',clicou)
botao.addEventListener('mouseenter',entrou)
botao.addEventListener('mouseout', saiu)

function clicou(){
  botao.innerHTML = "Clicou não entra nem sai mais!"
  botao.removeEventListener('mouseenter',entrou)
  botao.removeEventListener('mouseout', saiu)
}
function entrou(){ botao.innerHTML = "Entrou!" }
function saiu(){ botao.innerHTML = "Saiu!" }
```

Foi usado o removeEventListener para, através do click, desativar os eventos de entrada e saída.

COMO PODEMOS ESCUTAR UM EVENTO?

Usando o addEventListener() para um array de elementos

Como já vimos nas aulas anteriores quando pegamos elementos pela classe ou nome da tag os métodos nos retornam um array de objetos. Para não termos que atribuir o evento para estes elementos um-a-um podemos utilizar o **foreach**.

HTML

```html
<h1>Exemplos para eventos em multiplos objetos</h1>
<button type="button" class="item">Clique Aqui!</button><br>
<button type="button" class="item">Clique Aqui!</button><br>
<button type="button" class="item">Clique Aqui!</button><br>
<button type="button" class="item">Clique Aqui!</button><br>
<button type="button" class="item">Clique Aqui!</button><br>
<button type="button" class="item">Clique Aqui!</button><br>
```

JAVASCRIPT

```javascript
var botoes = document.querySelectorAll('.item')
botoes.forEach((item) => {
  item.addEventListener('click',clicar)
});

function clicar() {
  console.log("clicou");
}
```

COMO PODEMOS ESCUTAR UM EVENTO?

Usando o addEventListener() para um array de elementos

Podemos melhorar ainda mais este código usando o arrow function.

HTML

```
<h1>Exemplos para eventos em multiplos objetos</h1>
<button type="button" class="item">Clique Aqui!</button><br>
<button type="button" class="item">Clique Aqui!</button><br>
<button type="button" class="item">Clique Aqui!</button><br>
<button type="button" class="item">Clique Aqui!</button><br>
<button type="button" class="item">Clique Aqui!</button><br>
<button type="button" class="item">Clique Aqui!</button><br>
```

JAVASCRIPT

```
var botoes = document.querySelectorAll('.item')
botoes.forEach((item) => {
  item.addEventListener('click',()=>console.log("clicou"))
});
```

Fica bem mais enxuto.

ALGUNS EVENTOS JAVASCRIPT

ALGUNS EVENTOS JAVASCRIPT

Eventos com o mouse

EVENTOS COM O MOUSE

mouseenter ou onmouseenter

Evento disparado quando o ponteiro do mouse se move para cima de um elemento.

HTML

```
<style media="screen">
  #div1{
    width: 200px; height: 150px; color: white;
    text-align: center; line-height: 150px;
    background-color: gray; font-family:arial;
  }
</style>
<h1>Eventos Javascript</h1>
<div id="div1"> Exemplos com Mouse</div>
```

JAVASCRIPT

```
var minhaDiv = document.querySelector('#div1')
minhaDiv.addEventListener('mouseenter',entrar)
function entrar(){
  minhaDiv.innerHTML = 'Você entrou'
  minhaDiv.style.backgroundColor = 'blue'
}
```

OBS. Para todos os eventos, eles só funcionam sem o “on” no início se estiverem sendo usados com o addEventListener.

EVENTOS COM O MOUSE

mouseout ou onmouseout

Evento disparado quando o ponteiro do mouse é movido para fora de um elemento ou um de seus filhos.

HTML

```
<style media="screen">
  #div1{
    width: 200px; height: 150px; color: white;
    text-align: center; line-height: 150px;
    background-color: gray; font-family:arial;
  }
</style>
<h1>Eventos Javascript</h1>
<div id="div1"> Exemplos com Mouse</div>
```

JAVASCRIPT

```
var minhaDiv = document.querySelector('#div1')
minhaDiv.addEventListener('mouseout',sair)
function sair(){
  minhaDiv.innerHTML = 'Você saiu'
  minhaDiv.style.backgroundColor = 'green'
}
```

EVENTOS COM O MOUSE

click ou onclick

Evento disparado quando se clica em um elemento.

HTML

```
<style media="screen">
  #div1{
    width: 200px; height: 150px; color: white;
    text-align: center; line-height: 150px;
    background-color: gray; font-family:arial;
  }
</style>
<h1>Eventos Javascript</h1>
<div id="div1"> Exemplos com Mouse</div>
```

JAVASCRIPT

```
var minhaDiv = document.querySelector('#div1')
minhaDiv.addEventListener('click', clicar)
function clicar(){
  minhaDiv.innerHTML = 'Você clicou!'
  minhaDiv.style.backgroundColor = 'orange'
}
```

EVENTOS COM O MOUSE

contextmenu ou oncontextmenu

Evento disparado quando se clica com o botão direito do mouse sobre um elemento.

HTML

```
<style media="screen">
  #div1{
    width: 200px; height: 150px; color: white;
    text-align: center; line-height: 150px;
    background-color: gray; font-family:arial;
  }
</style>
<h1>Eventos Javascript</h1>
<div id="div1"> Exemplos com Mouse</div>
```

JAVASCRIPT

```
var minhaDiv = document.querySelector('#div1')
minhaDiv.addEventListener('contextmenu',clickDireito)
function clickDireito(){
  event.preventDefault()
  minhaDiv.innerHTML = 'Click direito'
  minhaDiv.style.backgroundColor = 'pink'
}
```

OBS. O 'event.peventDefault()' , se tiver, cancela a ação padrão do evento.

EVENTOS COM O MOUSE

dblclick ou ondblclick

Evento disparado quando é aplicado um clique duplo sobre um elemento.

HTML

```
<style media="screen">
  #div1{
    width: 200px; height: 150px; color: white;
    text-align: center; line-height: 150px;
    background-color: gray; font-family:arial;
  }
</style>
<h1>Eventos Javascript</h1>
<div id="div1"> Exemplos com Mouse</div>
```

JAVASCRIPT

```
var minhaDiv = document.querySelector('#div1')
minhaDiv.addEventListener('dblclick',clickDuplo)
function clickDuplo(){
  minhaDiv.innerHTML = 'Foram 2 clicks!'
  minhaDiv.style.backgroundColor = 'red'
}
```

EVENTOS COM O MOUSE

mousemove ou onmousemove

Evento disparado quando o ponteiro do mouse se move sobre um elemento.

HTML

```
<style media="screen">
  #div1{
    width: 200px; height: 150px; color: white;
    text-align: center; line-height: 150px;
    background-color: gray; font-family:arial;
  }
</style>
<h1>Eventos Javascript</h1>
<div id="div1"> Exemplos com Mouse</div>
<span></span>
```

JAVASCRIPT

```
var minhaDiv = document.querySelector('#div1')
minhaDiv.addEventListener('mousemove',mover)
function mover(event){
  var x = event.clientX
  var y = event.clientY
  var res = document.querySelector('span')
  res.innerHTML = "Posição X: "+x+" e Y: "+y
}
```

OBS. O 'event.clientX' e 'event.clientY' retornam a posição do mouse na tela.

O 'event.clientX – this.offsetLeft' e 'event.clientY – this.offsetTop' retornam a posição do mouse no elemento.

EVENTOS COM O MOUSE

mousedown ou onmousedown

Evento disparado quando o botão do mouse é pressionado sobre um elemento.

HTML

```
<style media="screen">
  #div1{
    width: 200px; height: 150px; color: white;
    text-align: center; line-height: 150px;
    background-color: gray; font-family:arial;
  }
</style>
<h1>Eventos Javascript</h1>
<div id="div1"> Exemplos com Mouse</div>
<span></span>
```

JAVASCRIPT

```
var minhaDiv = document.querySelector('#div1')
minhaDiv.addEventListener('mousedown',apertou)
function apertou(){
  minhaDiv.innerHTML = 'Botão apertado!'
  minhaDiv.style.backgroundColor = 'aqua'
}
```

EVENTOS COM O MOUSE

mouseup ou onmouseup

Evento disparado quando o botão do mouse é liberado, despreciando, sobre um elemento.

HTML

```
<style media="screen">
  #div1{
    width: 200px; height: 150px; color: white;
    text-align: center; line-height: 150px;
    background-color: gray; font-family:arial;
  }
</style>
<h1>Eventos Javascript</h1>
<div id="div1"> Exemplos com Mouse</div>
<span></span>
```

JAVASCRIPT

```
var minhaDiv = document.querySelector('#div1')
minhaDiv.addEventListener('mouseup',soltou)
function soltou(){
  minhaDiv.innerHTML = 'Botão solto!'
  minhaDiv.style.backgroundColor = 'lime'
}
```

ALGUNS EVENTOS JAVASCRIPT

Eventos com o teclado

EVENTOS COM O TECLADO

keydown ou onkeydown

Evento disparado quando o usuário pressiona uma tecla.

HTML

```html
<label for="">Digite um texto:</label>
<input type="text" id="idTexto"><br>
<p >Resposta: <span id="res"></span></p>
```

JAVASCRIPT

```javascript
var texto = document.querySelector('#idTexto')
texto.addEventListener('keydown',()=>{
  var resultado = document.querySelector('#res')
  resultado.innerHTML = texto.value
})
```

Repare que, como ele registra o momento em que o usuário aperta a tecla, a primeira letra só aparece quando ele aperta a segunda tecla.

EVENTOS COM O TECLADO

keyup ou onkeyup

Evento disparado quando o usuário libera uma tecla pressionada.

HTML

```html
<label for="">Digite um texto:</label>
<input type="text" id="idTexto"><br>
<p >Resposta: <span id="res"></span></p>
```

JAVASCRIPT

```javascript
var texto = document.querySelector('#idTexto')
texto.addEventListener('keydown',()=>{
  var resultado = document.querySelector('#res')
  resultado.innerHTML =""
  texto.value = ""
})
texto.addEventListener('keyup',(event)=>{
  var resultado = document.querySelector('#res')
  resultado.innerHTML ="A tecla "+texto.value+" = "+event.keyCode
})
```

Agora sim!!! Como ele registra após a tecla ser pressionada, o texto sai junto conforme digitamos.

keyCode – retorna o valor da tecla. Usamos este valor para saber qual tecla está sendo pressionada.

EVENTOS COM O TECLADO

Exemplo

HTML

```
<style media="screen">
  #div1{
    width: 200px; height: 150px; color: white;
    text-align: center; line-height: 150px;
    background-color: gray; font-family:arial;
  }
</style>
<h1>Eventos Javascript</h1>
<div id="div1"> Exemplos com Teclado</div>
```

JAVASCRIPT

```
var px=0;
var py=0;
document.addEventListener('keydown',function(event){
  var obj = document.getElementById('div1');
  obj.style.position = 'absolute'
  var tecla = event.keyCode;
  //37 esquerda 38 cima 39 direita 40 esquerda
  if(tecla==37){
    px-=10;
    obj.style.left=px+"px"
  }else if(tecla==38){
    py-=10;
    obj.style.top=py+"px"
  }else if(tecla==39){
    px+=10;
    obj.style.left=px+"px"
  }else if(tecla==40){
    py+=10;
    obj.style.top=py+"px"
  }
})
```

Movendo a DIV – Neste exemplo usamos as setas do teclado para mover a div pela tela.

ALGUNS EVENTOS JAVASCRIPT

Eventos com formulário

EVENTOS COM FORMULÁRIO

focus ou onfocus

Evento disparado quando o elemento recebe o foco, é selecionado.

HTML

```html
<h1>Eventos Javascript</h1>
<form action="index.html" method="post">
  <label for="idNome">Nome:</label>
  <input type="text" id="idNome">
  <input type="submit" value="Enviar">
  <input type="reset" value="Limpar">
</form>
<span id="acao"></span>
```

JAVASCRIPT

```javascript
var nome = document.querySelector("#idNome")
var span = document.querySelector("#acao")
nome.addEventListener('focus',()=>{
  nome.style.outlineColor = 'blue'
  span.innerHTML = "O usuário acessou o campo"
})
```

EVENTOS COM FORMULÁRIO

blur ou onblur

Evento disparado quando o elemento perde o foco, perde a seleção

HTML

```html
<h1>Eventos Javascript</h1>
<form action="index.html" method="post">
  <label for="idNome">Nome:</label>
  <input type="text" id="idNome">
  <input type="submit" value="Enviar">
  <input type="reset" value="Limpar">
</form>
<span id="acao"></span>
```

JAVASCRIPT

```javascript
var nome = document.querySelector("#idNome")
var span = document.querySelector("#acao")
nome.addEventListener('blur',()=>{
  nome.style.borderColor = 'red'
  span.innerHTML = "O usuário saiu do campo"
})
```

EVENTOS COM FORMULÁRIO

change ou onchange

Evento disparado quando o conteúdo do elemento é alterado, input, select, textarea.

HTML

```
<label for="idBarra">Escala:</label>
<input type="range" id="idBarra">
<span id="idValor">50</span>
```

JAVASCRIPT

```
var range = document.querySelector("#idBarra")
range.addEventListener('change',()=>{
document.querySelector("#idValor").innerHTML = range.value
})
```

input ou oninput

Usando o evento “input” este exemplo funciona em tempo real!

EVENTOS COM FORMULÁRIO

submit ou onsubmit

Evento disparado quando o botão submit é clicado para enviar os dados do formulário.

HTML

```html
<form action="index.html" method="post" id="idForm">
  <label for="idNome">Nome:</label>
  <input type="text" id="idNome" autocomplete="off">
  <input type="submit" value="Enviar">
  <input type="reset" value="Limpar">
```

JAVASCRIPT

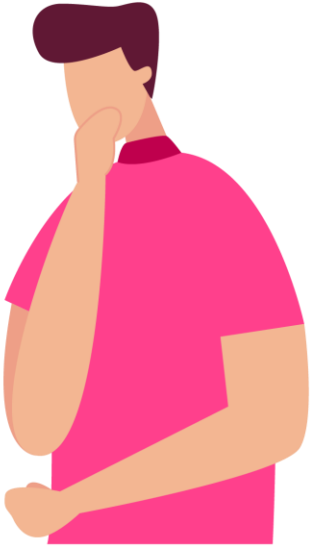
```javascript
var form = document.querySelector("#idForm")
form.addEventListener('submit',()=>{
alert('Obrigado por preencher nossa pesquisa!')
})
```

DÚVIDAS?

- lsilva@fiap.com.br

“A dúvida é o princípio da sabedoria.”

- Aristóteles



3ºCheckpoint – em grupo (máx. 3)

Vamos ver se conseguimos utilizar o que aprendemos até agora!

Crie uma página única, nela faça:

1 – Uma declaração, apenas faltando os dados do declarante, pelo menos 5, e uma caixa com os campos para preenchimento desses dados. Conforme o usuários for preenchendo os campos a declaração deve ser completada.

2 – Crie 3 botões de estilo de texto para que o usuário possa personalizar sua declaração, “Clássico, Moderno e Esporte”. Através destes botões deve ser mudado cor do texto, tipo de fonte e cor de fundo.

3 – Crie uma lista com 10 itens a sua escolha, pode ser carros, times, cantores, etc. e uma div para mostrar a imagem, conforme você for clicando nos itens da lista a imagem deve mudar.

Obs. A página deve estar devidamente estilizada e organizada, isso também contará na avaliação.

OBRIGADO

FIAP


FIAP