

# FORMAÇÃO ACELERADA EM PROGRAMAÇÃO

Módulo Javascript O.O.  
Aula 1

Raoni Kulesza

INSTITUIÇÃO EXECUTORA

**SOFTEx**  
RECIFE

COORDENADORA

**MCTI**  
FUTURO

**Softex**

APOIO

MINISTÉRIO DA  
CIÊNCIA, TECNOLOGIA  
E INOVAÇÃO

GOVERNO FEDERAL  
**BRASIL**  
UNIÃO E RECONSTRUÇÃO

# Revisão Javascript

abstract	arguments	await	boolean
break	byte	case	catch
char	class*	const	continue
debugger	default	delete	do
double	else	enum*	eval
export*	extends*	false	final
finally	float	for	function
goto	if	implements	import
in	instanceof	int	interface
let*	long	native	new
null	package	private	protected
public	return	short	static
super*	switch	synchronized	this
throw	throws	transient	true
try	typeof	var	void
volatile	while	with	yield

# Document Object Model (DOM)

# Entendendo o DOM

- Quando uma página da Web é carregada, o navegador cria um DOM (*Document Object Model*) da página
- O modelo HTML DOM é construído como uma árvore de objetos

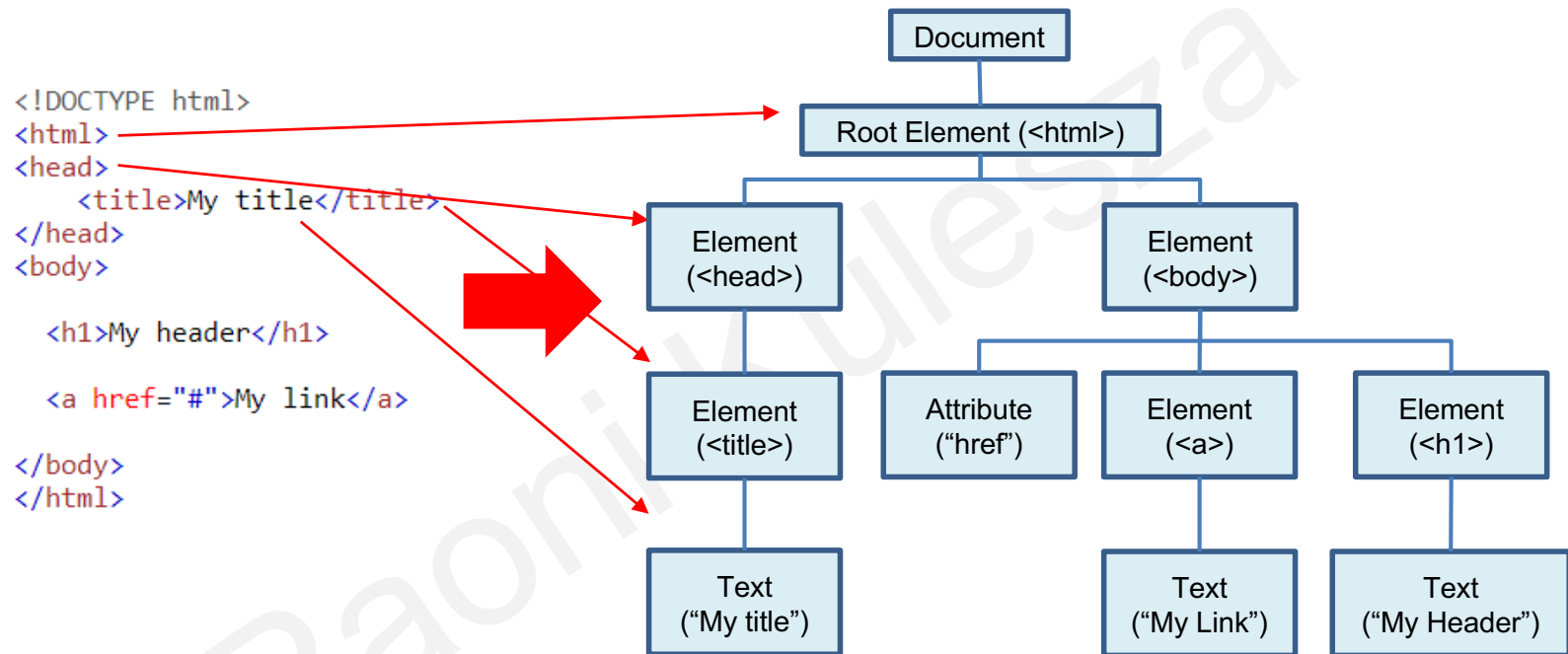
# Entendendo o DOM

- Com o modelo de objeto, o JavaScript obtém todo o poder necessário para criar HTML dinâmico:
  - JavaScript pode **alterar** todos os **elementos** HTML na página
  - JavaScript pode **alterar** todos os **atributos** HTML na página
  - JavaScript pode **alterar** todos os **estilos CSS** na página
  - JavaScript pode **remover elementos e atributos** HTML existentes

# Entendendo o DOM

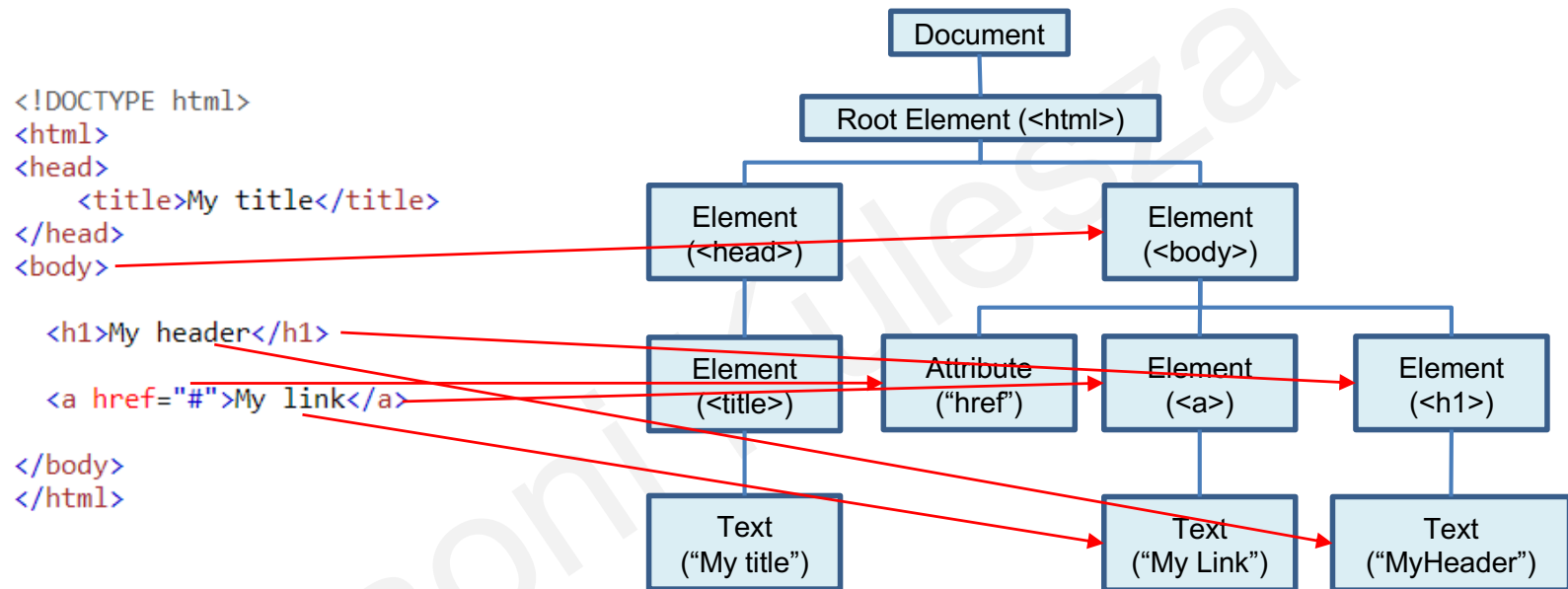
- Com o modelo de objeto, o JavaScript obtém todo o poder necessário para criar HTML dinâmico:
  - JavaScript pode **adicionar novos elementos e atributos** HTML
  - JavaScript pode **reagir** a todos os **eventos** HTML existentes na página
  - JavaScript pode **criar** novos **eventos** HTML na página

# Entendendo o DOM





# Entendendo o DOM



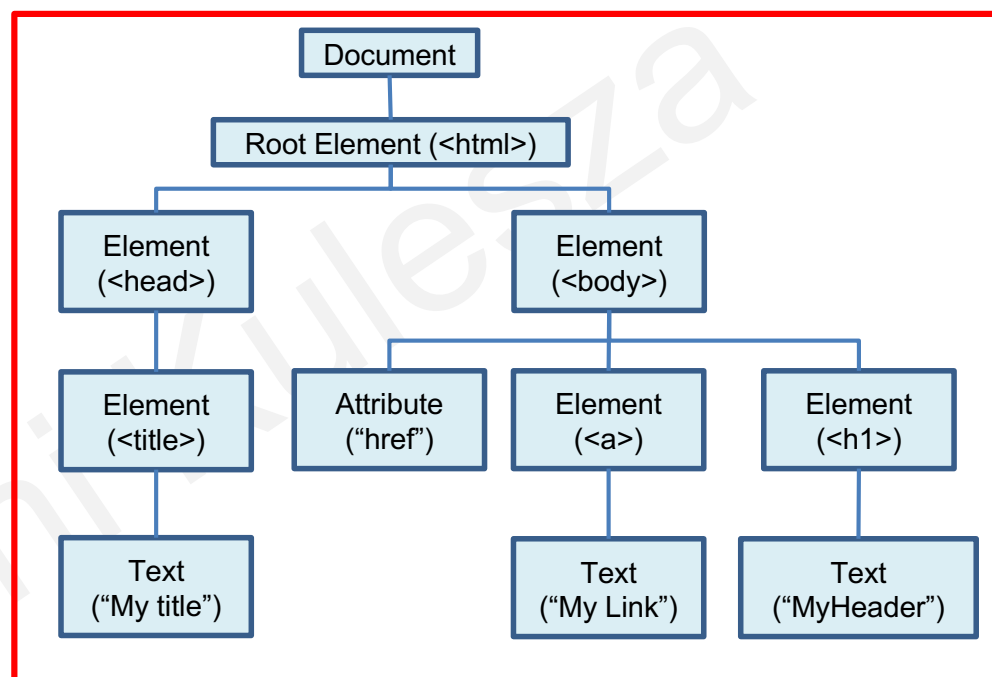
# Entendendo o DOM

```
<!DOCTYPE html>
<html>
<head>
  <title>My title</title>
</head>
<body>

  <h1>My header</h1>

  <a href="#">My link</a>

</body>
</html>
```



É assim que o navegador 'vê' a página da web

Ao atribuir ids a elementos, podemos usar JavaScript para acessar (e modificar)

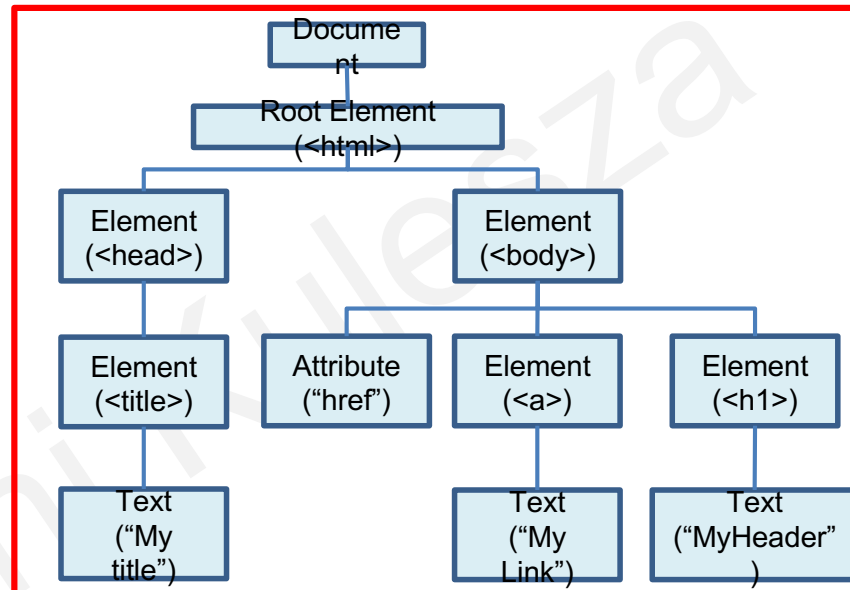
# Entendendo o DOM

```
<!DOCTYPE html>
<html>
<head>
  <title>My title</title>
</head>
<body>

  <h1>My header</h1>

  <a href="#">My link</a>

</body>
</html>
```



Também deve ser evidente que o DOM pode crescer para ser extremamente complexo à medida que a complexidade de uma determinada página aumenta

# Classes e IDs

# Classes e IDs

Vimos ao longo do modulo passado com CSS como as **classes** são importantes para estilizar páginas da Web

Porém, **IDs** são a principal maneira pela qual o JavaScript acessa e modifica dinamicamente páginas da Web e seus elementos

# Exemplos

Um dos muitos métodos JavaScript é `getElementById()`

Este método faz parte da classe `document` e é assim invocado usando a notação dot (".") (Uma das semelhanças da linguagem com Java), ou seja,

```
document.getElementById("id")
```

class

method

id

# Exemplos

```
<h2>What Can JavaScript Do?</h2>
```

```
<p id="demo">  
  Short cuts make delays,<br>  
  but inns make longer ones.  
</p>
```

```
<button type="button" onclick="document.getElementById('demo').innerHTML =  
  'Frodo'">Who?  
</button>
```

## What Can JavaScript Do?

Short cuts make delays,  
but inns make longer ones.

Who?



## What Can JavaScript Do?

Frodo

Who?

Javascript integrado com HTML



# Javascript e HTML

## The `<script>` Element

In HTML, JavaScript code must be inserted between

```
<body>
```

```
<h2>JavaScript in Body</h2>
```

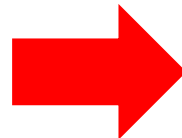
```
<p id="demo"></p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML =  
  "Do not meddle in the affairs of Wizards, for" +  
  "<br>they are subtle and quick to anger.";
```

```
</script>
```

```
</body>
```



### JavaScript in Body

Do not meddle in the affairs of Wizards, for  
they are subtle and quick to anger.

# Javascript e HTML

## JavaScript Functions and Events

A JavaScript function is a block of JavaScript code, that can be executed when "called" for

For example, a function can be called when an event occurs, like when the user clicks a button

We will talk more about functions and events later

# Javascript e HTML

## JavaScript in `<head>`

In this example, a JavaScript function is placed in the `<head>` section of an HTML page

The function is invoked (called) when a button is clicked

# Javascript e HTML

<head>

```
<head>
<script>
function myFunction() {
    document.getElementById("demo").innerHTML =
        "Gurney Halleck, Dune";
}
</script>
</head>
```

```
<body style="font-family:sans-serif;">
```

```
<h2>JavaScript in Head</h2>
```

```
<p id="demo">
    If wishes were fishes, we'd all cast nets
</p>
```

```
<button type="button" onclick="myFunction()">Who?</button>
```

```
</body>
```

## JavaScript in Head

If wishes were fishes, we'd all cast nets

Who?

## JavaScript in Head

Gurney Halleck, Dune

Who?

# Javascript e HTML

## JavaScript in `<body>`

In this example, a JavaScript function is placed in the `<body>` section of an HTML page

The function is invoked (called) when a button is clicked

# Javascript e HTML

<body>

```
<body>
<h2>JavaScript in Body</h2>

<p id="demo">
  Happiness consists in getting enough sleep.<br>
  Just that, nothing more.
</p>

<button type="button" onclick="myFunction()">
  Who?
</button>

<script>
  function myFunction() {
    document.getElementById("demo").innerHTML =
      "Robert A. Heinlein";
  }
</script>
</body>
```

## JavaScript in Body

Happiness consists in getting enough sleep.  
Just that, nothing more.

Who?

## JavaScript in Body

Robert A. Heinlein

Who?

# Javascript e HTML

- JavaScript externo
  - Scripts can also be placed in external files
  - External scripts are practical when the same code is used in many different web pages
  - JavaScript files have the file extension **.js**
  - To use an external script, put the name of the script file in the **src** (source) attribute of a `<script>` tag

# Javascript e HTML

## externo

```
<p id="demo">  
Progress isn't made by early risers. It's made<br>  
by lazy men trying to find easier ways to do <br>  
something.  
</p>
```

```
<button type="button" onclick="myFunction()">  
Try it  
</button>
```

```
<p>  
(myFunction is stored in an external<br>  
file called "myScript.js" in the /js<br>  
folder)  
</p>
```

```
<script src="js/myScript.js"></script>
```

Progress isn't made by early risers. It's made by lazy men trying to find easier ways to do something.

Try it

(myFunction is stored in an external file called "myScript.js" in the /js folder)

Robert A. Heinlein

Who?

(myFunction is stored in an external file called "myScript.js" in the /js folder)



# Javascript e HTML

externo

- Vantagens do JavaScript externo
  - Separa HTML e código
  - Torna HTML e JavaScript mais fáceis de ler e manter
  - Arquivos JavaScript armazenados em cache podem acelerar o carregamento de páginas
- Para adicionar vários arquivos de script a uma página - use várias tags de script

```
<script src="myScript1.js"></script>  
<script src="myScript2.js"></script>
```

INSTITUIÇÃO EXECUTORA

COORDENADORA

APOIO

FORMAÇÃO ACELERADA  
EM PROGRAMACÃO

SOFTEX

MCTI

Softex

MINISTÉRIO DA  
CIÊNCIA, TECNOLOGIA

GOVERNO FEDERAL

BRAZIL

# JavaScript Output

# JS Output

- O JavaScript pode "exibir" dados de diferentes maneiras:
  - HTML element: `innerHTML`
  - HTML output: `document.write()`
  - Alert pop-up: `window.alert()`
  - Console no navegador: `console.log()`

# JS Output - innerHTML

Para acessar um elemento HTML, o JavaScript pode usar o método

`document.getElementById("id")`

`id` identifica o elemento HTML

`innerHTML` define o conteúdo HTML

Alterando a propriedade `innerHTML` de um elemento HTML é uma maneira comum de exibir dados em HTML

# JS Output - innerHTML Example

```
<h2>Isaac Asimov</h2>
```

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

```
<script>  
document.getElementById("demo").innerHTML =  
  "I do not fear computers.<br>" +  
  "I fear the lack of them."  
</script>
```



**Isaac Asimov**

I do not fear computers.  
I fear the lack of them.

# JS Output - document.write()

Para fins de teste, é conveniente usar document.write()

```
<h2>Isaac Asimov</h2>
```

```
<script>  
  document.write("Isaac Asimov quote<br>" +  
    "Should appear here");  
</script>
```



**Isaac Asimov**

Isaac Asimov quote  
Should appear here



# JS Output - window.alert()

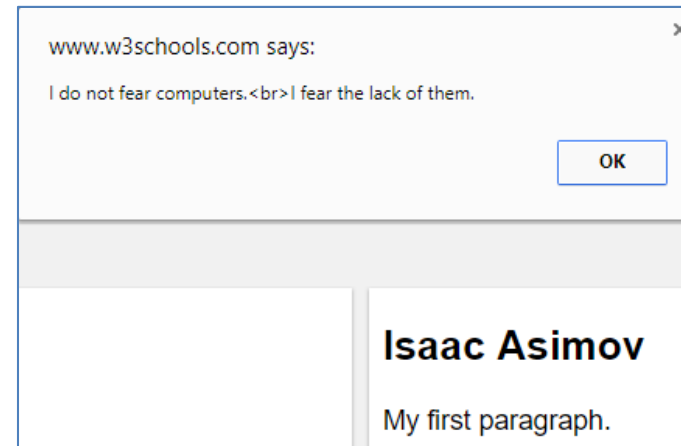
Você pode usar uma caixa de alerta para exibir dados

```
<h2>Isaac Asimov</h2>
```



```
<script>  
window.alert("I do not fear computers.<br>" +  
"I fear the lack of them.");  
</script>
```

Note: The alert box appears as a drop-down box at the top of the browser window. The button must be clicked in order to return to the page (Pressing 'Enter' also works)



# JavaScript - Eventos



# Eventos Javascript

- Eventos HTML são coisas que acontecem com elementos HTML
- Quando o JavaScript é usado em páginas HTML, o JavaScript pode reagir a esses eventos
- Um evento HTML pode ser algo que o navegador faz, ou algo que um usuário faz

Aqui estão alguns exemplos de eventos HTML:

- Uma página da Web HTML terminou de carregar
- Um campo de entrada HTML foi alterado
- Um botão HTML foi clicado

# Eventos Javascript

- Muitas vezes, quando os eventos acontecem, você quer fazer algo
- JavaScript permite executar código quando eventos são detectados
- HTML permite que atributos do manipulador de eventos, com código JavaScript, sejam adicionados à elementos HTML

Com aspas simples:

```
<element event='some JavaScript'>
```

Com aspas duplas:

```
<element event="some JavaScript">
```

# Eventos Javascript

```
<button onclick="document.getElementById('demo').  
    innerHTML=Date()">The time is?</button>
```

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



The time is?

Mon Aug 07 2017 10:21:38 GMT-0400 (Eastern Daylight Time)

# Eventos Javascript

O código altera o conteúdo de seu próprio elemento (usando `this.innerHTML`)

```
<button onclick="this.innerHTML=Date()">  
  The time is?  
</button>
```



The time is?



Mon Aug 07 2017 10:26:14 GMT-0400 (Eastern Daylight Time)

# Eventos Javascript

O código JavaScript geralmente tem várias linhas  
É mais comum ver atributos de evento chamando funções

```
<p>  
  Click the button to display the date.  
</p>
```

```
<button onclick="displayDate()">  
  The time is?  
</button>
```

```
<script>  
function displayDate() {  
  document.getElementById("demo").  
    innerHTML = Date();  
}  
</script>
```



Click the button to display the date.

The time is?



Click the button to display the date.

The time is?

Mon Aug 07 2017 10:40:22 GMT-0400 (Eastern Daylight Time)

# Eventos Javascript

Aqui está uma lista de alguns eventos HTML comuns

Evento	Descrição
onchange	Um elemento HTML foi alterado
onclick	O usuário clica em um elemento HTML
onmouseover	O usuário move o mouse sobre um elemento HTML
onmouseout	O usuário move o mouse para longe de um elemento HTML
onkeydown	O usuário pressiona uma tecla do teclado
onload	O navegador terminou de carregar a página

# Eventos Javascript

- Os manipuladores de eventos podem ser usados para manipular e verificar a entrada do usuário, as ações do usuário e as ações do navegador:
  - Coisas que devem ser feitas sempre que uma página é carregada
  - Coisas que devem ser feitas quando a página é fechada
  - Ação que deve ser executada quando um usuário clica em um botão
  - Conteúdo que deve ser verificado quando um usuário insere dados

# Eventos Javascript

- Muitos métodos diferentes podem ser usados para permitir que o JavaScript funcione com eventos:
- Os atributos de evento HTML podem executar código JavaScript diretamente
- Os atributos de evento HTML podem chamar funções JavaScript
- Você pode atribuir suas próprias funções de manipulador de eventos a elementos HTML
- Você pode impedir que eventos sejam enviados ou manipulados



# Eventos e Inputs - Exemplo

# Exercício 1

## Programa Olá Você!

Nome:

Mostrar

Olá Marina

# index.html

```
<!DOCTYPE html>
<html lang="pt-br">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Exemplo 2.1</title>
</head>

<body>
  <h1> Programa Olá Você! </h1>
  <p> Nome: <input type="text" id="nome">
    <input type="button" value="Mostrar"> </p>
  <p id="resposta"></p>
</body>
```

index.js

```
function mostrarOla() {  
  var nome = document.getElementById("nome").value;  
  document.getElementById("resposta").innerHTML = "Olá " + nome;  
}
```

index.js

```
<script src="arquivo.js"></script>
```

```
var mostrar = document.getElementById("mostrar");  
mostrar.onclick = mostrarOla;
```

# JavaScript - Event Listeners

# JavaScript - Event Listeners

- Uma maneira mais elegante de reagir a eventos em um site é criando ouvintes de eventos
- Remove o 'trigger' do HTML e o coloca no JS
- Semelhantes aos que vimos, com algumas diferenças

# JavaScript - Event Listeners

```
document.getElementById('btn1').addEventListener('click', function() {  
    //code goes here  
});
```

Isso aparece no código JavaScript, não no HTML



# JavaScript - Event Listeners

```
<body>
  <div class='container'>
    <p id='demo'>&lt;&lt; Click here &gt;&gt;</p>
    <p id='demo2'></p>
    <span id='btn1' class='btn'>Click</span>
  </div>

  <script src='handlers.js'></script>
</body>
```

INSTITUIÇÃO EXECUTORA

COORDENADORA

APOIO

FORMAÇÃO ACELERADA  
EM PROGRAMAÇÃO

SOFTEX

MCTI

Softex

MINISTÉRIO DA  
CIÊNCIA, TECNOLOGIA

GOVERNO FEDERAL

BRAZIL

# JavaScript - Event Listeners

```
document.getElementById('demo').style.cursor = 'pointer';
```

```
document.getElementById('demo').addEventListener('click', function() {  
    let p = document.getElementById('demo');  
    p.innerHTML = "It is so shocking to find out how many<br>" +  
                  "people do not believe that they can learn,<br>" +  
                  "and how many more believe learning to be<br>" +  
                  "difficult.<br>" +  
                  "&nbsp;&nbsp;&nbsp;- Frank Herbert, Dune";  
});
```

# JavaScript - Event Listeners

<< Click here >>



It is so shocking to find out how many people do not believe that they can learn, and how many more believe learning to be difficult.

- Frank Herbert, Dune

# JavaScript - Event Listeners

```
<body>
  <div class='container'>
    <p id='demo'>&lt;&lt; Click here &gt;&gt;</p>
    <p id='demo2'></p>
    <span id='btn1' class='btn'>Click</span>
  </div>

  <script src='handlers.js'></script>
</body>
```

INSTITUIÇÃO EXECUTORA

COORDENADORA

APOIO

FORMAÇÃO ACELERADA  
EM PROGRAMAÇÃO

SOFTEX

MCTI

Softex

MINISTÉRIO DA  
CIÊNCIA, TECNOLOGIA

GOVERNO FEDERAL

BRAZIL

# JavaScript - Event Listeners

```
document.getElementById('btn1').addEventListener('click', function() {  
    let p = document.getElementById('demo2');  
    p.innerHTML = "I must not fear. Fear is the mind-killer.<br>" +  
                  "Fear is the little-death that brings total<br>" +  
                  "obliteration. I will face my fear. I will<br>" +  
                  "permit it to pass over me and through me. And<br>" +  
                  "when it has gone past I will turn the inner<br>" +  
                  "eye to see its path. Where the fear has gone<br>" +  
                  "there will be nothing. Only I will remain.<br>" +  
                  "&nbsp;&nbsp;&nbsp;- Frank Herbert, Dune<br>";  
});
```

# JavaScript - Event Listeners

Click



I must not fear. Fear is the mind-killer.  
Fear is the little-death that brings total  
obliteration. I will face my fear. I will  
permit it to pass over me and through me. And  
when it has gone past I will turn the inner  
eye to see its path. Where the fear has gone  
there will be nothing. Only I will remain.

- Frank Herbert, Dune

Click

# Event Listeners- Document

Há uma variedade de listeners de eventos integrados ao JavaScript (além de 'clicar')

animationcancel event  
animationend event  
animationiteration event  
animationstart event  
copy event  
cut event  
DOMContentLoaded event  
dragend event  
dragenter event  
dragleave event  
dragover event  
dragstart event  
drag event

drop event  
fullscreenchange event  
fullscreenerror event  
gotpointercapture event  
keydown event  
keypress event  
keyup event  
lostpointercapture event  
paste event  
pointercancel event  
pointerdown event  
pointerenter event  
pointerleave event

pointerlockchange event  
pointerlockerror event  
pointermove event  
pointerout event  
pointerover event  
pointerup event  
readystatechange event  
scroll event  
selectionchange event  
selectstart event  
touchcancel event  
touchend event  
touchmove event

touchstart event  
transitioncancel event  
transitionend event  
transitionrun event  
transitionstart event  
visibilitychange event  
wheel event



# Event Listeners - Element

Há uma variedade de listeners de eventos integrados ao JavaScript (além de 'clicar')

afterscriptexecute event  
auxclick event  
beforescriptexecute event  
blur event  
click event  
compositionend event  
compositionstart event  
compositionupdate event  
contextmenu event  
copy event  
cut event  
dblclick event  
DOMActivate event

DOMMouseScroll event  
error event  
focusin event  
focusout event  
focus event  
fullscreenchange event  
fullscreenerror event  
gesturechange event  
gestureend event  
gesturestart event  
keydown event  
keypress event  
keyup event

mousedown event  
mouseenter event  
mouseleave event  
mousemove event  
mouseout event  
mouseover event  
mouseup event  
mousewheel event  
msContentZoom event  
MSGestureChange event  
MSGestureEnd event  
MSGestureHold event  
MSGestureStart event

MSGestureTap event  
MSInertiaStart event  
overflow event  
paste event  
scroll event  
select event  
show event  
touchcancel event  
touchend event  
touchmove event  
touchstart event  
underflow event  
wheel event



# Event Listeners - Window

Há uma variedade de listeners de eventos integrados ao JavaScript (além de 'clicar')

afterprint event  
animationcancel event  
animationend event  
animationiteration event  
animationstart event  
appinstalled event  
beforeprint event  
beforeunload event  
blur event  
copy event  
cut event  
devicemotion event  
deviceorientation event

DOMContentLoaded event  
error event  
focus event  
gamepadconnected event  
gamepaddisconnected event  
hashchange event  
languagechange event  
load event  
messageerror event  
message event  
offline event  
online event  
orientationchange event

pagehide event  
pageshow event  
paste event  
popstate event  
rejectionhandled event  
resize event  
storage event  
transitioncancel event  
transitionend event  
transitionrun event  
transitionstart event  
unhandledrejection event  
unload event

vrdisplayactivate event  
vrdisplayblur event  
vrdisplayconnect event  
vrdisplaydeactivate event  
vrdisplaydisconnect event  
vrdisplayfocus event  
vrdisplaypointerrestricted  
event  
vrdisplaypresentchange event

# Events Listeners e Inputs

```
// declara a função mostrarOla
function mostrarOla() {
    // obtém o conteúdo do campo (com id=) nome
    var nome = document.getElementById("nome").value;
    // exibe no parágrafo (resposta): "Olá " e o nome informado
    document.getElementById("resposta").textContent = "Olá " + nome;
}
// cria uma referência ao botão (com id=) mostrar
var mostrar = document.getElementById("mostrar");
// registra para o botão "mostrar" um ouvinte para o evento click,
// que ao ser clicado irá chamar a função mostrarOla
mostrar.addEventListener("click", mostrarOla);
```

# Perguntas?



**FORMAÇÃO ACELERADA**  
EM PROGRAMACÃO

INSTITUIÇÃO EXECUTORA

**SOFTEX**

**MCTI**

COORDENADORA

**Softex**

APOIO

MINISTÉRIO DA  
CIÊNCIA, TECNOLOGIA

GOVERNO FEDERAL

**BRAZIL**

# Exercício 2

## Vídeo Locadora JS

Título do Filme:

Duração (min):

O Touro Ferdinando

1 hora(s) e 48 minuto(s)

# Exercício 3

Programa que recebe o veículo e valor e calcula o valor de entrada de 50% e o restante pago em 12x sem juros

## Revenda de Veículos JS

Veículo:

Preço RS:

**Promoção: Sanderro 2016**

**Entrada de RS: 15000.00**

**+ 12x de RS: 1250.00**

INSTITUIÇÃO EXECUTORA

COORDENADORA

APOIO

**FORMAÇÃO ACELERADA**  
EM PROGRAMACÃO

**SOFTEX**

**MCTI**

**Softex**

MINISTÉRIO DA  
CIÊNCIA, TECNOLOGIA

GOVERNO FEDERAL  
**BRAZIL**

# Exercício 4

## Restaurante JS

Buffet por Quilo R\$:

Consumo do Cliente (gr):

**Valor a pagar R\$: 28.00**

# Referencias

- [https://www.w3schools.com/js/js\\_htmlDOM.asp](https://www.w3schools.com/js/js_htmlDOM.asp)
- <https://www.w3schools.com/js/default.asp>
- <http://lotrproject.com/quotes/>
- <https://www.brainyquote.com/quotes/quotes/i/isaacasimo100104.html>
- [https://www.goodreads.com/author/quotes/205.Robert\\_A\\_Heinlein](https://www.goodreads.com/author/quotes/205.Robert_A_Heinlein)
- <https://en.wikiquote.org/wiki/Dune>
- <http://www.imdb.com/title/tt0062622/quotes>
- <https://developer.mozilla.org/en-US/docs/Web/Events>

