

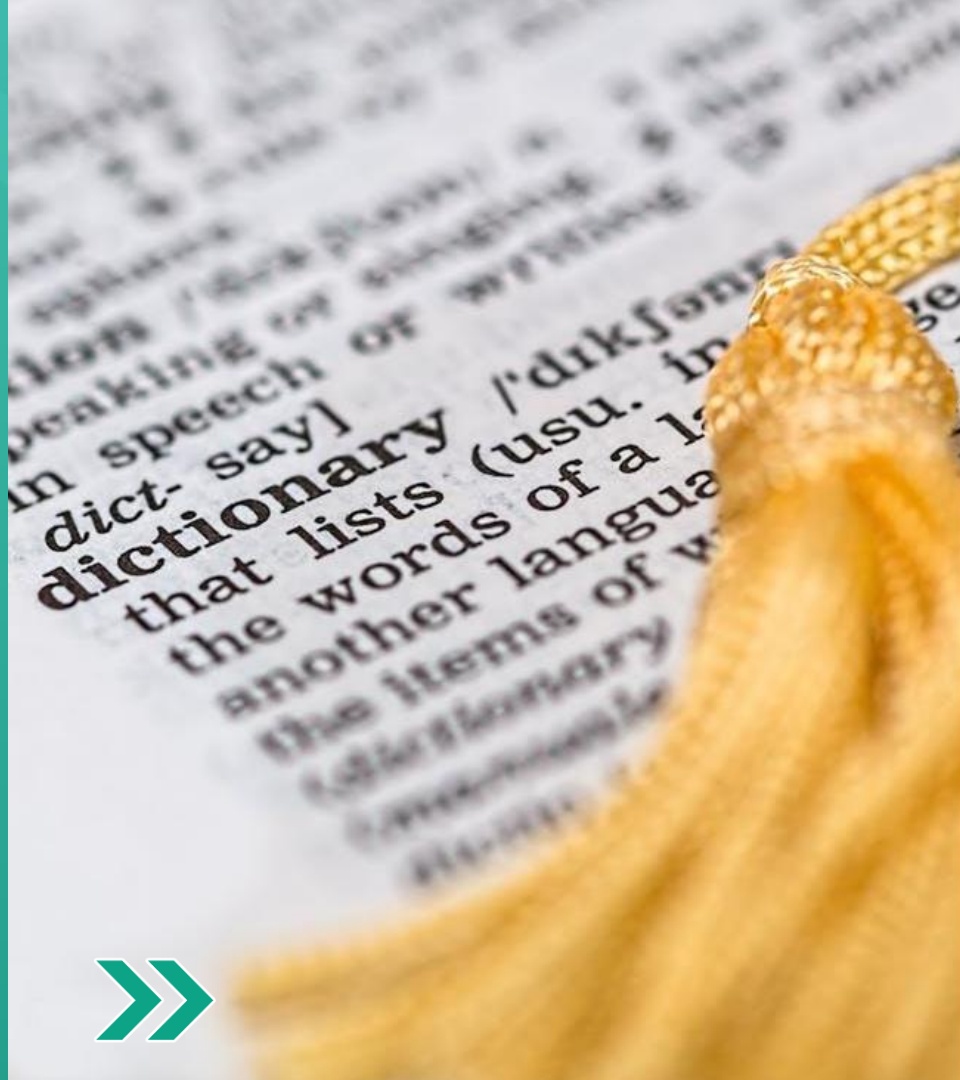


escola  
britânica de  
artes criativas  
& tecnologia

# Front-End



# GLOSSÁRIO



# Integre HTML com JavaScript



Dica: para encontrar rapidamente a palavra que procura aperte o comando CTRL+F e digite o termo que deseja achar.

- **Documente HTML com DOM**
- **Desenvolva formulários com JavaScript**
- **Adicione eventos a elementos HTML**
- **Estilize elementos no JavaScript**



# Documente HTML com DOM



# Documente HTML com DOM

## Glossário

### History.back

É usada para voltar para a página anterior no histórico de navegação do navegador. Essa função simula o comportamento do botão "Voltar" do navegador. Ela só irá funcionar se houver um histórico de navegação disponível. Se o usuário estiver na primeira página visitada durante a sessão ou se o histórico de navegação estiver desabilitado, a chamada para `history.back()` não terá efeito.

### Location.pathname

É uma propriedade do objeto `window.location` em JavaScript que retorna o caminho da URL atualmente carregada no navegador. Essa propriedade retorna uma string que representa o caminho e o nome do arquivo da URL.



# Desenvolva formulários com JavaScript



# Desenvolva formulários com JavaScript

## Glossário

### Split

Método usado para **dividir** uma string em um array de strings com base em um separador especificado. O separador pode ser um caractere, uma expressão regular ou uma string.



# Adicione eventos a elementos HTML





# Adicione eventos a elementos HTML

## Glossário

### ● Código bloqueante (ou síncrono)

O JavaScript é considerado um código bloqueante, pois normalmente executa instruções sequencialmente, uma de cada vez, de forma sincronizada. Isso significa que uma instrução é executada e concluída antes que a próxima seja executada.

### ● PreventDefault

É uma função utilizada em JavaScript para evitar o comportamento padrão de um evento. Ele é usado em conjunto com os eventos do DOM (Document Object Model) para impedir que a ação padrão associada a um determinado evento ocorra.



# Estilize elementos no JavaScript



# Estilize elementos no JavaScript

## Glossário

### Evento change

Esse evento é acionado quando o valor de um elemento de entrada é alterado e o foco é perdido desse elemento. Esse evento é comumente usado em elementos de formulário, como campos de texto, caixas de seleção (checkboxes) e botões de rádio (radio buttons), para detectar quando o usuário realiza uma seleção ou digitação e confirma a mudança.

### Evento keyup

Esse evento é acionado quando uma tecla é liberada após ser pressionada em um elemento de entrada. Esse evento permite detectar a ação de soltar uma tecla do teclado e pode ser usado para capturar a entrada de texto em tempo real ou executar ações com base nas teclas pressionadas.

### InnerHTML

É uma propriedade do objeto DOM que permite acessar e modificar o conteúdo HTML de um elemento. Essa propriedade retorna ou define o conteúdo HTML interno de um elemento, incluindo as *tags* HTML e o texto contido nele.



# Bons estudos!



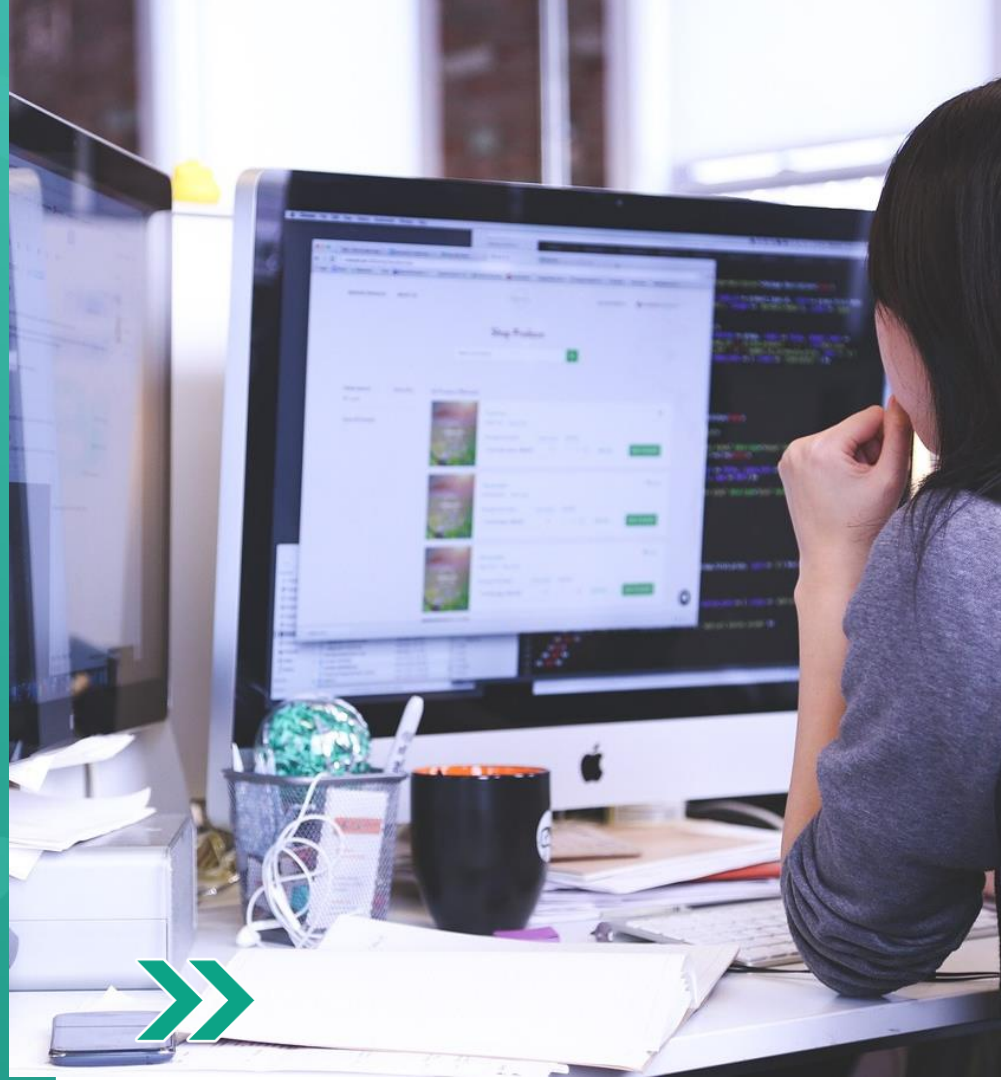


escola  
britânica de  
artes criativas  
& tecnologia

# Front-End

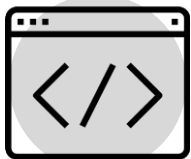


# BOAS PRÁTICAS



# Integre HTML com JavaScript

- **Documente HTML com DOM**
- **Adicione eventos a elementos HTML**



Confira boas práticas da comunidade de Front-End por assunto relacionado às aulas.



# Documente HTML com DOM



## **Location.pathname:**

O `location.pathname` retorna apenas o caminho e o nome do arquivo da URL, não incluindo o domínio, o protocolo ou outros componentes da URL. Se você precisar obter a URL completa, incluindo o domínio e o protocolo, pode usar `location.href` em vez de `location.pathname`.





# Adicione eventos a elementos HTML

Além dos eventos estudados, existem vários outros disponíveis no JavaScript que podem ser usados para manipular a interação do usuário com os elementos da página. Acompanhe!



## Eventos de mouse:

- **click:** Ocorre quando o usuário clica em um elemento.
- **mouseover:** Ocorre quando o cursor do mouse é colocado sobre um elemento.
- **mouseout:** Ocorre quando o cursor do mouse deixa um elemento.
- **mousedown:** Ocorre quando o botão do mouse é pressionado em um elemento.
- **mouseup:** Ocorre quando o botão do mouse é liberado em um elemento.



## Eventos de teclado:

- **keydown:** Ocorre quando uma tecla é pressionada.
- **keypress:** Ocorre quando uma tecla é pressionada e liberada.

# Adicione eventos a elementos HTML

Além dos eventos estudados, existem vários outros disponíveis no JavaScript que podem ser usados para manipular a interação do usuário com os elementos da página. Acompanhe!



- **Eventos de formulário:**
  - **submit:** Ocorre quando um formulário é submetido.
  - **input:** Ocorre quando o valor de um campo de entrada é alterado.
  
- **Eventos de janelas:**
  - **load:** Ocorre quando a página é completamente carregada.
  - **resize:** Ocorre quando a janela do navegador é redimensionada.
  - **scroll:** Ocorre quando a página é rolada.

# Adicione eventos a elementos HTML



## **Código bloqueante:**

Apesar de o JavaScript ser um código bloqueante, ele também possui recursos assíncronos que permitem contornar o bloqueio e evitar a interrupção da execução sequencial do código.



# Bons estudos!





escola  
britânica de  
artes criativas  
& tecnologia

# Front-End



# Integre HTML com JavaScript



## GUIA DA AULA 5



# Estilize elementos através do JavaScript



JS



Acompanhe aqui  
os temas que  
serão tratados  
na videoaula



## JS

Podemos aplicar estilo aos elementos utilizando a propriedade **style**:

```
document.querySelector('h1').style.color= 'red';
```

**Observação:** Para estilos compostos por duas palavras usamos o padrão camelCase:

```
style.backgroundColor= 'blue';
```

Além da adição de estilos **inline**, podemos manipular classes:

```
document.querySelector('.feedback').classList.add('success');
                        .classList.remove('success');
```

Assim evitamos o uso de CSS inline.







escola  
britânica de  
artes criativas  
& tecnologia

# Front-End



# Integre HTML com JavaScript



## GUIA DA AULA 4



# Adicione eventos a elementos HTML

 Sobre  
eventos



Acompanhe aqui  
os temas que  
serão tratados  
na videoaula



## Sobre eventos

Evento é o nome dado a uma ação de determinado elemento. Por exemplo:

- 1) **Click**: ao clicar em um botão;
- 2) **Submit**: ao enviar um formulário;
- 3) **Change**: ao mudar o conteúdo de um *input*.

Para adicionar um evento a um elemento, usamos a função **addEventListener**, que irá observar o evento e quando ele for disparado executará nosso código.

```
document.querySelector('form').addEventListener(function() {
  alert("O formulário foi enviado");
});
```





escola  
britânica de  
artes criativas  
& tecnologia

# Front-End



# Integre HTML com JavaScript



## GUIA DA AULA 3



# Desenvolva formulários com JavaScript

 Valores



Acompanhe aqui  
os temas que  
serão tratados  
na videoaula



## Valores

Para recuperar ou atribuir valor a um campo:

**let beneficiario= document.getElementById('nome-beneficiario')**

**beneficiario.value//** retorna o valor

**beneficiario.value= 'João Silva' //** atribui o valor

Podemos acessar os atributos inseridos em um campo através do operador ponto:

**beneficiario.type//** retorna o tipo do campo

**beneficiario.id //** retorna o ID atribuído ao campo

**beneficiario.placeholder//** retorna o placeholder atribuído ao campo.





Para habilitar ou desabilitar um botão:

```
let botao = document.getElementById('btn-depositar')
```

```
botao.disabled = true OU false
```





escola  
britânica de  
artes criativas  
& tecnologia

# Front-End



# Integre HTML com JavaScript



## GUIA DA AULA 2



# Selecione elementos com JavaScript

 Critérios



Acompanhe aqui  
os temas que  
serão tratados  
na videoaula



## Critérios

Podemos seleccionar um ou mais elementos através de alguns critérios:

**1) Nome da tag – retorna todos os elementos da tag:**

`document.getElementsByTagName('input')`

**2) ID – retorna apenas um elemento:**

`document.getElementById('btn-depositar')`

**3) Classe – retorna todos os elementos que usam a classe:**

`document.getElementsByClassName('container')`



#### 4) Seletores especiais:

`document.querySelector('input[type=number]')`

#### 5) Ou para retornar todos os itens que atendam ao critério da busca:

`document.querySelectorAll('input[type=number]')`

