

JavaScript: Fundamentos para Desenvolvimento Web Interativo - Turma 2025B

1.2 Integração do JavaScript com o HTML

Neste curso, os participantes serão guiados por uma jornada de aprendizado que abrange desde os conceitos básicos até técnicas avançadas de programação com JavaScript para web. Integrar o código JavaScript com páginas web é uma prática fundamental no desenvolvimento web, permitindo que estas páginas web sejam interativas e dinâmicas.

No decorrer deste documento, veremos as três formas que você pode utilizar para integrar JavaScript em documentos HTML.

Usando atributos de eventos de elementos HTML

Você pode adicionar o código JavaScript diretamente aos **atributos de eventos de elementos HTML**. Ele também pode ser chamado de **JavaScript Inline**. Esta técnica permite que você defina comportamentos de resposta a eventos do usuário, como cliques, sem precisar escrever código adicional fora do elemento HTML.

Para demonstrar esta prática, considere o uso do arquivo chamado **index.html** de nosso projeto de exemplo (**meu-projeto-javascript**). Altere este arquivo, escrevendo o código demonstrado no quadro abaixo:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de Atributos de Evento HTML</title>
</head>
<body>
  <h1>Olá, visitante!</h1>
  <button onclick="alert('Bem-vindo ao nosso site!');">Clique Aqui</button>
</body>
</html>
```

Abra o arquivo **index.html** no navegador, clique no botão “**Clique Aqui**” e observe o resultado conforme figura abaixo:

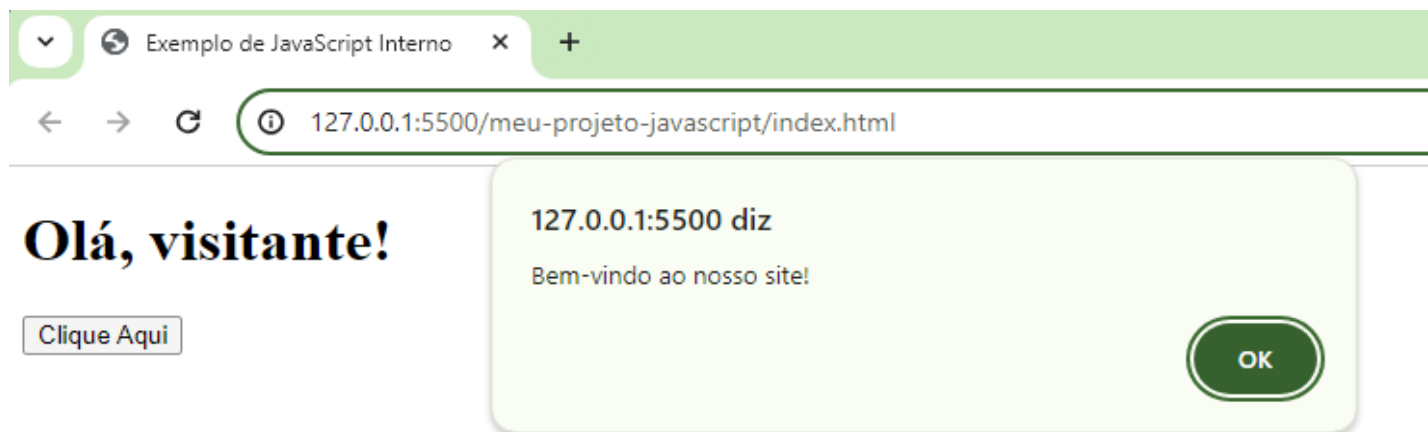


Figura 1 - Exemplo de uso de JavaScript Inline

Descrição da imagem: No navegador, o código cria um botão com o texto "Clique Aqui". Quando o botão é clicado, uma mensagem pop-up de alerta aparece com o texto "Bem-vindo ao nosso site!".

OBS: Esta não é a prática recomendada para scripts mais extensos ou para uma melhor manutenção do código, mas é útil para pequenas interações.

Incorporando o JavaScript diretamente no HTML

Você pode escrever um código **JavaScript diretamente dentro de um documento HTML** usando a tag **<script>**. Este método é útil para scripts pequenos ou para testes rápidos. Isso é chamado de **JavaScript interno**. Normalmente, os blocos de script são colocados na seção **<head>** ou antes do fechamento da tag **<body>** do documento (para garantir que eles não afetem o carregamento da página).

Altere o arquivo **index.html** com o seguinte código:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemplo de JavaScript Interno</title>
</head>
<body>

  <h1>Olá, visitante!</h1>
  <button onclick="saudacao();">Clique Aqui</button>

  <script>
    function saudacao() {
      alert("Bem-vindo ao nosso site!");
    }
  </script>

</body>
</html>
```

Neste exemplo, a função **saudacao()** é definida dentro de uma tag **<script>** e é chamada quando clicamos no botão **"Clique Aqui"**, graças ao atributo **onclick** da tag **<button>**. Observe que o código JavaScript está junto com o código HTML da página.

Abra o arquivo **index.html** no navegador e observe o resultado conforme figura abaixo:

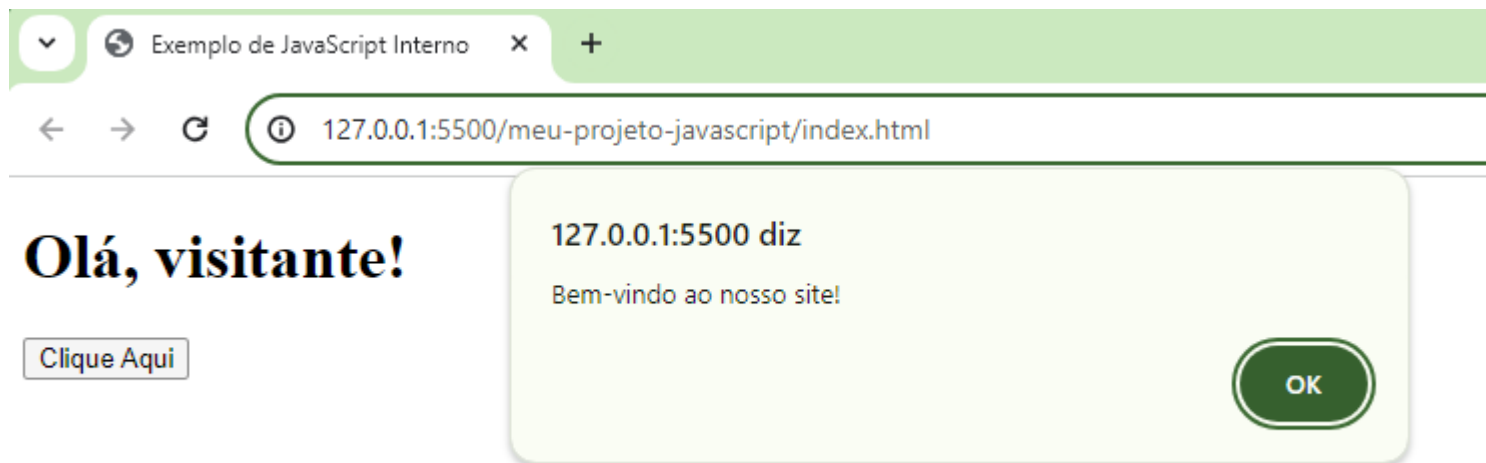


Figura 2 - Exemplo de uso de JavaScript diretamente dentro de um documento HTML

Descrição da imagem: No navegador, o código cria um botão com o texto "Clique Aqui". Quando o botão é clicado, a função JavaScript `saudacao()` é chamada, exibindo uma mensagem pop-up de alerta com o texto "Bem-vindo ao nosso site!".

Observe que não houve alteração para o usuário final.

Usando um arquivo JavaScript externo

Para manter o **código JavaScript separado do HTML**, facilitando a manutenção e reutilização, você pode colocar o JavaScript em um arquivo externo (normalmente com a extensão `.js`) e referenciá-lo no HTML. Isso é feito usando o atributo `src` da tag `<script>`, que contém o caminho para o arquivo JavaScript.

Veja a sintaxe básica no quadro abaixo:

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

Vamos alterar nosso exemplo anterior. Para isso, em nosso projeto de exemplo, vamos alterar o conteúdo do arquivo chamado **script.js** localizado dentro da pasta **scripts**. Neste documento, digite o seguinte código demonstrado no quadro abaixo:

```
function saudacao() {  
    alert("Bem-vindo ao nosso site!");  
}
```

Agora, altere o arquivo **index.html**, alterando o elemento de **<script>**, conforme quadro abaixo:

```
<!DOCTYPE html>  
<html lang="pt-BR">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Exemplo de JavaScript Externo</title>  
</head>  
<body>  
    <h1>Olá, visitante!</h1>  
    <button onclick="saudacao();">Clique Aqui</button>  
  
    <script src="scripts/script.js" defer></script>  
</body>  
</html>
```

Observe que o atributo **defer** é utilizado na tag **<script>** para indicar ao navegador que ele deve prosseguir com o processamento e renderização de todo conteúdo HTML. Ou seja, o script será executado apenas após o HTML completo ser carregado e analisado, garantindo que qualquer manipulação do DOM pelo script ocorra de forma segura e eficaz.

Salve e atualize seu navegador, conforme figura abaixo, e você deverá ver a mesma coisa!

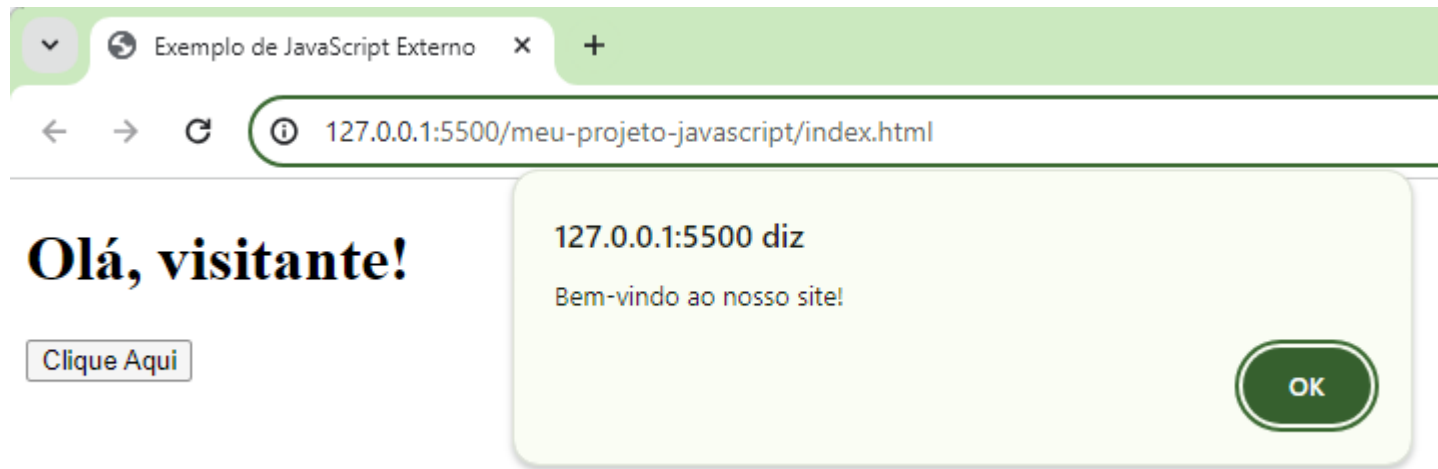


Figura 3 - Exemplo de uso de arquivo JavaScript externo

Descrição da imagem: No navegador, o código cria uma página com um título "Olá, visitante!" e um botão com o texto "Clique Aqui". Quando o botão é clicado, uma função JavaScript chamada `saudacao()` é executada. Esta função está definida em um arquivo externo chamado "script.js", que é carregado e executado de forma assíncrona após o carregamento completo da página.

Observe que funciona igualmente, mas agora nós temos o JavaScript em um arquivo externo. Isso é geralmente uma coisa boa em termos de organização de código, e faz com que seja possível reutilizar o código em múltiplos arquivos HTML. Além disso, o HTML fica mais legível sem grandes pedaços de script no meio dele.

OBS: Recomenda-se usar o JavaScript externo para projetos maiores devido à organização e reutilização do código, além de evitar misturar muito código de script diretamente com o HTML.

◀ 1.1.4 Configuração do ambiente de desenvolvimento

Seguir para...

1.3 Estratégias para carregamento eficiente de códigos JavaScript ►

[Baixar o aplicativo móvel.](#)