



# Introdução ao Javascript

---

# Introdução a Javascript

JavaScript é uma das linguagens de programação mais populares e versáteis do mundo da tecnologia. Criada inicialmente para ser utilizada em páginas web, ela rapidamente evoluiu para ser empregada em diversas áreas, como desenvolvimento front-end, back-end e até mesmo para a criação de aplicativos móveis e jogos.

# JavaScript é Java?

Apesar de JavaScript ter um nome parecido com a linguagem Java, elas são duas linguagens de programação distintas, com propósitos e sintaxes diferentes. A semelhança no nome é principalmente uma estratégia de marketing adotada pela Netscape, a empresa que desenvolveu o JavaScript, no momento em que a linguagem foi lançada.

# JavaScript é Java?

Na década de 1990, a popularidade do Java estava em alta, e a Netscape viu uma oportunidade de aproveitar esse sucesso ao lançar sua nova linguagem de programação. Eles decidiram nomeá-la JavaScript para capitalizar a familiaridade com o Java e atrair a atenção dos desenvolvedores. No entanto, a semelhança entre os nomes é praticamente o único aspecto em comum entre as duas linguagens.

# Características do Javascript

Javascript é uma linguagem client-side (lado do cliente). A programação client-side, em termos simples, refere-se à execução de código ou lógica de programação no lado do cliente, ou seja, no navegador web do usuário.

Quando você visita um site, o navegador faz o download do código HTML, CSS e JavaScript que compõem a página. O HTML define a estrutura e o conteúdo da página, o CSS é responsável pela aparência visual e o JavaScript é a linguagem de programação que permite adicionar interatividade e comportamentos dinâmicos à página.

# Características do Javascript

Dizemos que ao programar, construímos um corpo:

O HTML são os ossos;

O CSS é a pele;

O Javascript é o cérebro;

# Como adicionar Javascript a minha página HTML?

Existem 3 formas de adicionar Javascript a sua página, e apenas dois deles são recomendados. Na programação, precisamos aprender até mesmo o que já caiu em desuso, por que apesar de não ser mais utilizado, não é retirado da linguagem por ter aplicações rodando com esses comandos.

A primeira forma de adicionar Javascript a sua página é o mais recomendado: Criar um arquivo externo com extensão .js e criar todo seu Javascript ali dentro, em seguida conectar esse JS ao seu HTML. Vamos ver na prática:

# Como adicionar Javascript a minha página HTML?

A tag `<script>` é usada para incluir e vincular arquivos JavaScript externos a uma página HTML. Essa tag é inserida no código HTML e fornece ao navegador o caminho para o arquivo JavaScript que deve ser baixado e executado.

A sintaxe básica para vincular um arquivo JavaScript externo usando a tag `<script>` é a seguinte:

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



# Como adicionar Javascript a minha página HTML?

O atributo `src` é utilizado para especificar o caminho do arquivo JavaScript externo. Esse caminho pode ser um URL completo, caso o arquivo esteja hospedado em um servidor externo, ou um caminho relativo ao arquivo HTML atual, caso o arquivo esteja no mesmo servidor ou diretório.

# Como adicionar Javascript a minha página HTML?

Ao incluir a tag `<script>` com o atributo `src`, o navegador fará o download do arquivo JavaScript especificado e executará o código contido nele. É importante observar que a inclusão do arquivo JavaScript externo deve ser feita antes de qualquer código JavaScript que dependa dele.

A tag `<script>` também pode ser usada sem o atributo `src` para incluir código JavaScript diretamente na página HTML, entre as tags `<script>` de abertura e fechamento. Isso é útil quando você precisa incluir pequenos trechos de código JavaScript diretamente na página.

# Como adicionar Javascript a minha página HTML?

Em resumo, a tag `<script>` é usada para vincular arquivos JavaScript externos a uma página HTML, permitindo que o código JavaScript seja baixado e executado pelo navegador. Isso possibilita a separação do código JavaScript em arquivos separados, tornando a estrutura e a manutenção do código mais organizadas e flexíveis.

A forma de implementar Javascript que caiu em desuso citada anteriormente, é colocando a tag `<script>` dentro da tag `<head>`. Vamos entender por quê!

# Como adicionar Javascript a minha página HTML?

Talvez faça sentido adicionar o que chamamos de cérebro anteriormente dentro da cabeça, mas vamos entender porque não é esse o caso. Um documento HTML é lido de cima para baixo. Ao encontrar a tag `<script>` dentro do `<head>`, o VS Code para de ler HTML para entender o Javascript. Sabemos que a tag `<head>` vem antes do `<body>`, que é tudo aquilo que será visível para o usuário. Enquanto seu Javascript está sendo processado pelo computador, o HTML dentro da tag `<body>` não é lido, logo, nada é mostrado pro usuário. Quanto maior seu Javascript, mais tempo o usuário ficará com a tela em branco!

# Como adicionar Javascript a minha página HTML?

Adicionando javascript interno:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <!--Todo o seu body precisa vim antes da tag <script>-->
  <script>
    //Aqui escrevemos todo o javascript
  </script>
</body>
</html>
```

# Como adicionar Javascript a minha página HTML?

Adicionando javascript externo:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <!--Todo o seu body precisa vim antes da tag <script>-->
  <script src="caminho.js"></script>
</body>
</html>
```



# Como adicionar Javascript a minha página HTML?

Adicionando javascript no <head> (forma que caiu em desuso):

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <script>
    //Todo seu javascript aqui
  </script>
</head>
<body>

</body>
</html>
```

# Mas como é feito na vida real?

Na vida real, sempre usamos o formato de arquivo externo, tanto pro javascript quanto para o CSS. A nível de estudo, não há problema em colocar os arquivos internamente.



# Em nossas palavras...

A forma que caiu em desuso, foi colocar o script dentro a tag head, já que assim, caso o javascript seja muito grande,

O navegador, vai parar de ler o html pra entender o seu javascript, e enquanto isso, so tem uma tela em branco pro usuário

Uma outra forma de fazer, é colocar a tag script, como a ultima tag, dentro do body. Precisa ser a ultima, por que se não, caímos no mesmo problema citado anteriormente.

A terceira e a forma como fazemos na vida real, é criar um arquivo externo com extensão js, e conectar, através da tag script, mas agora, fazendo uso, do atributo src.

# Começando no Javascript

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
<p id="meuP"></p>
<script>
  document.getElementById("meuP").innerHTML = "Javascript"
</script>
</body>
</html>
```

# Começando no Javascript

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

A primeira coisa a se fazer é criar uma tag vazia com um id!

```
<script>  
    document.getElementById("meuP").innerHTML = "Javascript"  
</script>
```

Abrimos nossa tag `<script>` e então, acessamos o documento com o comando `document`. No javascript, ao usar o ponto `(.)` significa entrar dentro daquilo que veio antes dele

# Começando no Javascript

```
<script>  
    document.getElementById("meuP").innerHTML = "Javascript"  
</script>
```

Mas quem é esse documento? Lembra que a primeira tag HTML é a tag `<!DOCTYPE>`? É como se o computador estivesse olhando para todo o seu código. EM seguida, utilizamos o `getElementById`. Traduzindo, pegue o elemento pelo id. Dentro dos parênteses, passamos o nome do id entre as aspas. Podem ser simples ou duplas. Usamos o ponto (.) novamente, e passamos o `innerHTML`. A funcionalidade dele é:

# Começando no Javascript



Javascript

Ao renderizar nossa página, a palavra está na tela. Indo no Inspeccionar, veremos que dentro da nossa tag `<p>`, agora, temos a palavra Javascript, como se estivesse lá o tempo inteiro

# Vamos recapitular?

Vimos que inicialmente javascript foi criado para ser uma linguagem de programação para web, mas evoluiu e agora é capaz de fazer back end e até mesmo jogos com javascript.

Aprendemos também que javascript NADA tem a ver com java. Os nomes se parecem devido a uma jogada de marketing dos criadores do javascript. Inicialmente a linguagem se chamava ECMA Script.

Vimos também que se trata de uma linguagem client-side, ou seja, lado do cliente. Cada vez que pesquisamos uma palavra no google, é feito uma requisição, um pedido, que ele navegue até o servidor do google, e traga todos os sites que tem a ver com a palavra pesquisada.

# Vamos recapitular?

Existem 3 formas de adicionar javascript em uma página, sendo que apenas duas ainda são utilizadas:

Colocando a tag `<script>` e criando todo o seu javascript interno, ou utilizando a tag `script` dentro do atributo `src`, que significa procura.

Vimos também que, precisamos fazer todo um caminho com uma sequência de passos bem definida. Usamos a palavra `document` para entrar no documento HTML. Usamos o ponto `(.)` para poder entrar dentro daquilo que veio antes dele.



# Vamos recapitular?

```
<script>  
  document.getElementById("nome do id").innerHTML = "Conteúdo que quero escrever dentro da tag"  
  //Entrando no documento html  
  //Lendo o documento até encontrar aquele que tem o id  
  //Pegando o espaço vazio entre essa tag, e escrevendo o que quer nesse espaço  
</script>
```



# Erros comuns e dicas

Escrever a tag `<script>` na mão e ao invés de SRC, escrever SCR. Se o nome do atributo está escrito de forma correta, o computador não reconhece e não conecta os arquivos.

Escrever o `getElementById` na mão e colocar alguma letra minúscula. Isso também faz com que o computador não reconheça e consequentemente, não faça o que foi pedido!

Se confundir com o uso do `innerHTML`. Ele serve para captar a tag e escrever dentro dela!

Não criar uma tag vazia para colocar o conteúdo que você quer escrever na tela. Se não há um lugar para escrever o conteúdo, ele consequentemente não aparecerá!

# Erros comuns e dicas

O javascript de fato é mais verboso que o python, mas com domínio da lógica e entendimento básico dos comandos, você consegue chegar lá!

Querer estudar diversos conteúdos de uma vez não é prático muito menos funcional. Se não sabe por onde começar, peça ajuda a alguém que já esteja na área.

Não conseguir de primeira é extremamente normal. Mas nada que você faça constante e repetidamente terá o mesmo nível de dificuldade que o início.

Não adianta estudar a teoria e deixar a prática de lado, em nenhuma área da programação!

# Erros comuns e dicas

Dica de ouro: Sim, é muita coisa pra aprender, muito detalhezinho. Não se assuste caso leia a questão e não entenda de primeira. Faça uma lista! O que precisa ser feito primeiro? Escreva com suas palavras. Se necessário, volte ao conteúdo quando estiver programando e veja o que faz aquilo que você quer no momento!

# Avançando no Javascript

Vimos uma forma de escrever na tela, que foi com o `document.getElementById`. Mas existem diversas outras formas:

**Alert:** Abre uma caixa de pop-up. Exemplo:

```
alert("Javascript")
```

Não esqueça que, se quer mostrar uma frase ou palavra, coloque entre aspas. O alert é todo minúsculo.

# Avançando no Javascript

Temos também o `console.log`, que mostra as informações no console do navegador. Basta ir em Inspecionar, e em seguida, console.

O console vai ser seu melhor aliado na hora de descobrir erros no seu código. Se o erro for de sintaxe, aparecerá no console. Caso contrário, o problema está na lógica, e não aparecerá no console.

```
console.log("Javascript")
```

# Avançando no Javascript

O onclick é um atributo de evento do JavaScript que é usado para associar uma ação a um elemento HTML quando ele é clicado. Ele permite que você execute uma função ou um trecho de código quando um determinado evento de clique ocorre em um elemento.

```
<button onclick="alert('Javascript')">Clique Aqui</button>
```

Note que nesse caso, não precisamos de uma tag <script>. O onclick é tanto do HTML quanto do Javascript, e chama algum comando dentro dele. Quando o botão for clicado, aparecerá um alerta com a mensagem Javascript

# Mão no código

Escreva “Hello World” na tela através do javascript  
(`document.getElementById`)

Peça um nome ao usuário e mostre esse nome no console.  
(`console.log()`)

Faça um código que peça o nome do usuário, e ao clicar no botão, mostre uma mensagem personalizada no alerta.  
Exemplo: Seja bem vindo `${nome do usuário}`