

Introdução ao Javascript

Disciplina: Linguagem de Programação



Conteúdos:

Introdução ao Javascript.

Habilidade(s):

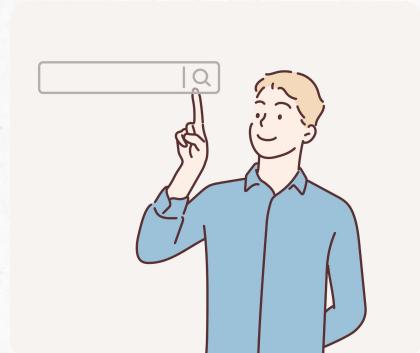
- Introdução às regras JavaScript;
- Utilização de ambientes de desenvolvimento de *software*.

Bloco 1

Vamos explorar o surgimento da linguagem JavaScript!

O que você prefere?

Na hora de pesquisar um assunto na internet, a qual grupo você pertence?



Pesquiso no Google. Eu prefiro saber algo através dos *sites*.



Abro as redes sociais. É a melhor plataforma para uma fofoca!



Pergunto aos meus amigos. Eles são todos bem informados.

Na hora de pesquisar...

Todos nós temos métodos diferentes.
Entretanto, não é raro que algumas pessoas
prefiram pesquisar por meios específicos.





Uma saída são os navegadores!

Eles são ferramentas que permitem que as pessoas acessem e explorem a web. Você consegue identificar alguns deles a seguir?

Google Chrome

Opera GX

Microsoft Edge

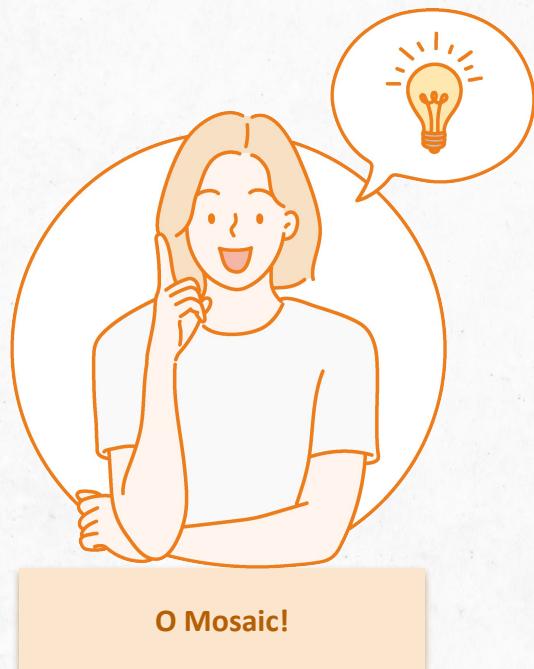
Safari

► □ ○ ▶ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷ □ ○ ▷

Você sabe qual foi o *primeiro* navegador a ser criado?



Dê palpites!



O Mosaic!

E aí, acertou?

NCSA Mosaic: Document View

File Navigate Options Annotate Documents Manuals Help

Document Title: National Center for Supercomputing Applications at the Univ
Document URL: <http://www.ncsa.illinois.edu/SDG/Software/Mosaic/NCSAMosaic>

2911

National Center for Supercomputing Applications at the University of Illinois

[National Center for Supercomputing Applications University of Illinois at Urbana-Champaign](#)

Contact us  

Site menu

- About NCSA
 - Careers
 - Facilities
 - Leaders
 - Contact Us

Search Keyword: []

Back Forward Home Reload Open... Save As... Clone New Window Close Window

A screenshot of a Windows-style web browser window titled "NCSA Mosaic: Document View". The menu bar includes File, Navigate, Options, Annotate, Documents, Manuals, and Help. The title bar shows the document title as "National Center for Supercomputing Applications at the Univ" and the URL as "http://www.ncsa.illinois.edu/SDG/Software/Mosaic/NCSAMosaic". The main content area displays the text "2911" and "National Center for Supercomputing Applications at the University of Illinois", followed by a link to "National Center for Supercomputing Applications University of Illinois at Urbana-Champaign". Below this is the NCSA logo (a stylized 'A' and 'S') and the University of Illinois logo (a 'U' inside a square). A "Site menu" section lists "About NCSA" with sub-options for Careers, Facilities, Leaders, and Contact Us, each preceded by an unchecked checkbox. At the bottom is a search bar and a toolbar with Back, Forward, Home, Reload, Open..., Save As..., Clone, New Window, and Close Window buttons.

O surgimento do Mosaic

Foi lançado em 1993, quando que a única ferramenta utilizada para criar *sites* era o HTML (*HyperText Markup Language*).

O HTML, por sua vez, é uma linguagem de marcação usada para criar e estruturar páginas da *web*. Com ela, é possível:



Definir a estrutura básica de uma página



Inserir *links*



Incorporar listas



Criar formulários



Além de prático, o HTML era acessível não apenas a engenheiros de *software*, mas também a desenvolvedores e *designers* inexperientes.

Alguma vez você já pensou em fazer um *blog*? Com o HTML, isso seria possível.



Dê um *play* no conhecimento!



Contudo, o Mosaic era bastante limitado. Veja como era a sua interface.



A necessidade de expansão

Tendo recém-lançado o navegador Netscape, Marc Andreessen se viu diante de dois impasses fundamentais.

Ele queria adicionar mais elementos dinâmicos ao *site*, que pudessem torná-lo mais interativo.

Ao mesmo tempo, ele também queria preservar a facilidade de uso do navegador para os desenvolvedores inexperientes.





Daí, surgiu a ideia do Javascript!

Ele tornou a *web* mais dinâmica, funcional e interativa.

A importância do surgimento da linguagem Javascript

O surgimento da linguagem Javascript teve um impacto significativo no desenvolvimento da *web* e na forma como as aplicações são criadas e interagem com os usuários. Agora, era possível:

Utilizar botões, formulários e elementos que respondem aos cliques do usuário.

Verificar se os dados inseridos estão corretos antes de enviá-los.

Permitir a mudança dinâmica do texto.

Criar caixas de diálogo, como alertas e *pop-ups*, para interagir com o usuário.

Facilitar a criação de animações, transições e efeitos visuais.

Permitir a incorporação de áudio e vídeo em páginas *web*.

Linguagens Java e JavaScript: qual é a diferença?

Java

Linguagem de **programação** desenvolvida pela Sun Microsystems.

É conhecida pela sua portabilidade.

É frequentemente usada para criar aplicativos *desktop*, móveis (Android) e empresariais.

Seu desenvolvimento envolve escrever o código, compilá-lo e, depois, executá-lo.

VS

JavaScript

Foi originalmente desenvolvida pela Netscape.

Linguagem de **script** voltada para a *web*.

Usada para tornar as páginas da *web* interativas e dinâmicas.

Sua tipagem é fraca e dinâmica, o que significa que pode variar bastante.

Principais acontecimentos

Assim como todo projeto, antes de se tornar o JavaScript como conhecemos hoje, o seu desenvolvimento enfrentou algumas etapas importantes. Confira:

Criação

Andressen contratou Brendan Eich, cofundador do Mozilla, para criar uma linguagem de *script* para navegadores.

Desafios

Eich enfrentou desafios devido à parceria entre Netscape e Sun Microsystems para integrar o Java ao Netscape 2.0.

Soluções

Para agir rapidamente antes do lançamento beta, Eich lançou o protótipo chamado "Mocha" em maio de 1995, com correções de *bugs*, recursos adicionados e APIs (Interfaces de Programação de Aplicativos) para interagir com o Netscape.

Produto final

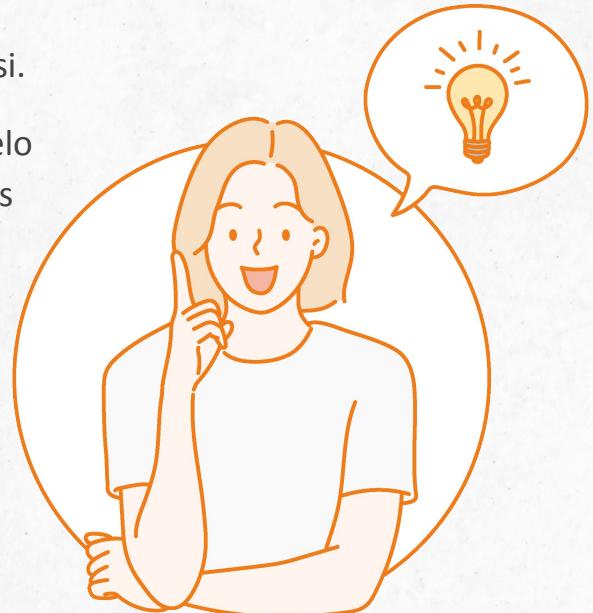
Essa linguagem de *script* eventualmente se tornou JavaScript.

A essencialidade dos APIs

As Interfaces de Programação de Aplicativos são conjuntos de regras e protocolos que permitem que diferentes *softwares* se comuniquem entre si.

Eles estabeleceram a base para o DOM (*Document Object Model*, ou Modelo de Objeto de Documento), que é uma interface usada para interagir com as linguagens **HTML** e **XML**.

O DOM é a representação programática da estrutura e conteúdo de uma página *web* e as APIs permitem que os desenvolvedores accessem e modifiquem essa estrutura.



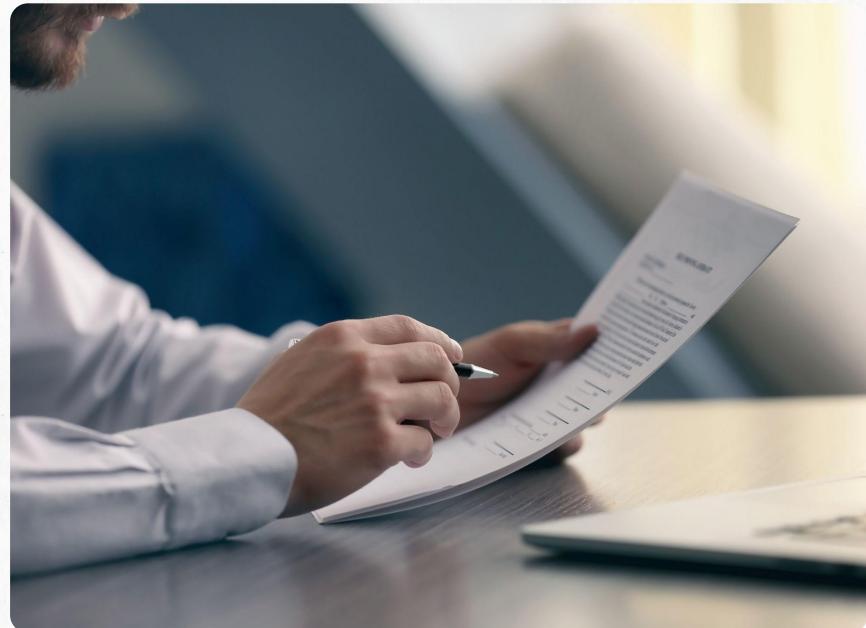
Padronização

ECMAScript

Em novembro de 1996, a Netscape se uniu com a ECMA International para definir as regras e os padrões para JavaScript. Desde então, a linguagem padronizada é chamada de **ECMAScript**.

ECMA-262

Documento que rege a normalização da linguagem.



Bloco 2

Explorando os aspectos da linguagem JavaScript.

Vamos realizar uma dinâmica?

Alguns *sites* notáveis demonstram o uso extensivo de JavaScript para criar experiências interativas.

Separem-se em seis grupos e utilizem os seus celulares para explorar os aplicativos móveis das plataformas a seguir e registrar os elementos dinâmicos que podem ser encontrados.

Google Maps

Facebook

X (antigo Twitter)

YouTube

Netflix

Amazon



JavaScript é uma linguagem multiplataforma

Além de poder ser encontrada em diferentes tipos de *sites* e plataformas, a linguagem JavaScript não é exclusiva da *web*.

Um exemplo disso são os motores gráficos, como a Unity 3D.

A Unity 3D é uma plataforma de desenvolvimento de jogos e aplicativos em 2D e 3D. Ela é amplamente utilizada na criação de jogos, como Hollow Knight, Among Us e Fall Guys.



Elementos básicos da linguagem Javascript

A linguagem Javascript possui uma **biblioteca padrão de objetos** e um conjunto fundamental de elementos que compõem o **núcleo da linguagem**.

Biblioteca Padrão de Objetos

É um conjunto de objetos pré-definidos que oferecem funcionalidades comuns e úteis.

São parte integrante da linguagem e podem ser usados para realizar diferentes tarefas.

Alguns exemplos de objetos incluem **date**, **string** e **math**.

Núcleos da Linguagem

Operadores: são símbolos ou palavras-chave que realizam operações em valores. Podem ser (+, -, *, /), que executam cálculos matemáticos.

Estruturas de controle: são construções que permitem controlar o fluxo de um programa. Isso inclui estruturas como **if**, **else**, **for**, **while** e **switch**, que permitem tomar decisões e repetir ações.

Declarações: são usadas para definir variáveis e funções. As variáveis são usadas para armazenar valores, enquanto as funções são blocos de código reutilizável.

Núcleos da linguagem

Além dos elementos que a compõem, a linguagem JavaScript pode ser dividida em duas partes.



Front-end



Back-end

Front-end

Também conhecido como *client-side* (lado do cliente), inclui os seguintes aspectos:

se refere à parte da aplicação web com a qual os usuários interagem diretamente, ou seja, a interface do usuário;

inclui tudo o que é visível e que os usuários podem interagir, como menus, botões, formulários, imagens e texto;

pode utilizar HTML (para estruturar o conteúdo), CSS (para estilizar a aparência) e JavaScript (para adicionar interatividade).

Em uma caixinha de perguntas do Instagram, por exemplo, o front-end pode ficar responsável pela aparência, pela disposição dos elementos e pelo estilo.

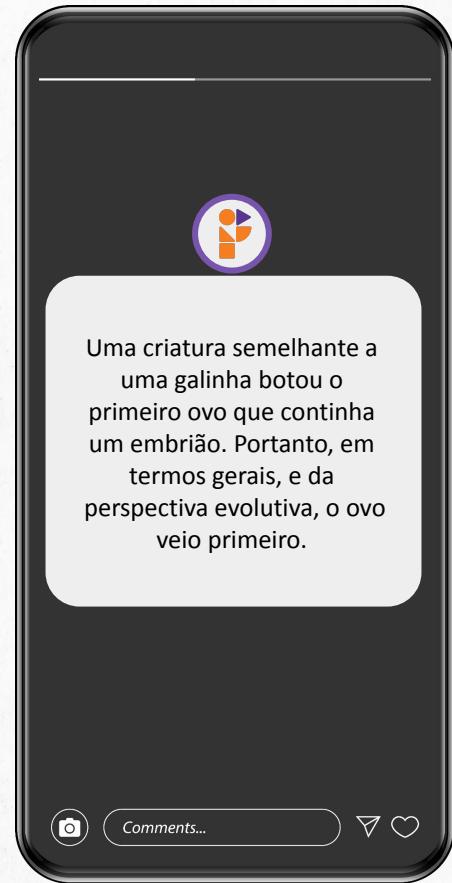


Back-end

Por outro lado, o *back-end* é considerado o *server-side* (lado do servidor). Além de operar nos bastidores, algumas de suas funções incluem:

- lidar com o processamento, armazenamento e gerenciamento de dados;
- englobar o servidor, o banco de dados e toda a lógica de negócios por trás da interface do usuário;
- utilizar linguagens de programação como Python, Java, Ruby, PHP e Node.js, e frameworks como Django, Ruby on Rails e Express.js.

Em uma caixinha de perguntas do Instagram, por exemplo, o *back-end* pode coletar, armazenar e processar as perguntas dos usuários.



Documento *script*

A linguagem JavaScript, como o próprio nome diz, é uma linguagem de *script*.

Isso significa dizer que o documento *script* é como um conjunto de instruções escritas em JavaScript, muitas vezes chamado de *script* ou arquivo de *script*.

O que isso quer dizer?

É como uma receita de bolo que o seu computador precisa executar em uma linguagem comprehensível para ele.



Qual é a importância do documento *script*?

 **Interatividade do usuário:** cria elementos interativos como botões, formulários e interfaces que respondem a ações do usuário, tornando a experiência mais envolvente.

 **Manipulação do DOM:** acessa e modifica a estrutura da página em tempo real (DOM), alterando conteúdo, estilo e comportamento sem recarregar a página.

 **Requisições assíncronas:** permite que páginas façam solicitações ao servidor sem recarregar a página, fundamental para aplicativos web dinâmicos.

 **Validação de dados:** valida dados em formulários antes do envio, melhorando a qualidade dos dados e a usabilidade geral.

 **Manipulação de *cookies* e armazenamento local:** controla *cookies* para rastreamento e armazena informações temporariamente. Além disso, usa APIs modernas para armazenar dados localmente.

Qual é a importância do documento *script*?



Carregamento dinâmico de conteúdo: carrega conteúdo adicional sem recarregar a página inteira. É útil para atualizações parciais.



Animações e efeitos visuais: cria animações e efeitos visuais impressionantes, tornando a página mais atraente.

Bloco 3

Investigando as principais regras da linguagem JavaScript.

Hora de recapitular

Você acabou de abrir uma empresa de tecnologia e precisa de mais profissionais para compor a sua equipe. Assim, você decide contratar especialistas em ***front-end*** e ***back-end***. Para isso, você precisa avaliar as características gerais do cargo que você está exigindo.

Em um anúncio da internet, o que você colocaria?

Front-end

Procura-se profissional para:

Back-end

Procura-se profissional para:

Sintaxe da linguagem JavaScript

Assim como em um manual de instruções, é imprescindível conhecer a principal estrutura da sintaxe da linguagem JavaScript.

Dessa forma, você pode garantir que o seu código seja válido e funcional para que o computador possa compreender e executar as suas instruções.



Declaração de variáveis

A declaração é seguida pelo nome da variável, um sinal de = (igual) e o valor que você deseja atribuir.

var

Tem escopo de função ou são globais.

```
var nome = "João"
```

let

Tem escopo de bloco.

```
let idade = 30
```

const

É usado para valores que não serão alterados depois de serem definidos.

```
const PI = 3.1416
```

Adicionando comentários

São usados para adicionar explicações ou notas ao código que não serão executadas pelo computador. Eles são úteis para documentar o código, tornando-o mais compreensível para outros desenvolvedores.

Comentários de linha única começam com //, e comentários de várias linhas com /* e */.

Linha única

```
// Este é um comentário de linha única  
  
var idade = 30; // Isso declara uma variável  
chamada idade
```

Várias linhas

```
/*  
Este é um exemplo de um comentário de várias  
linhas.  
Você pode escrever várias linhas de explicação  
aqui.  
*/  
var nome = "Maria";
```

Instruções

É uma ação que o computador deve executar. Dessa ações, o código pode emitir **expressões**, que são partes do código que produzem um valor, como cálculos matemáticos, operações de comparação ou chamadas de funções.

Exemplo de instrução simples (uma única linha)

```
var nome = "Maria"; // Isso é uma instrução que atribui o valor "Maria" à variável 'nome'.
```

Exemplo de expressões

```
var resultado = 5 + 3; // A expressão "5 + 3" produz o valor 8 e é atribuída à variável 'resultado'.
```

Estruturas de controle

Permitem que você controle o fluxo do seu programa.

if

A estrutura *if* permite executar um bloco de código apenas se uma condição for verdadeira.

for

Executa um bloco de código repetidamente por um número específico de vezes.

while

Executa um bloco de código repetidamente enquanto uma condição for verdadeira.

```
var idade = 20;  
if (idade >= 18) {  
    console.log("Você é maior de idade.")
```

```
for (var i = 0; i < 5; i++) {  
    console.log("Número: " + i);  
}
```

```
var contador = 0;  
while (contador < 3) {  
    console.log("Contador: " + contador);  
    contador++;  
}
```

Outras nomenclaturas

Além dos elementos que foram apresentados, também temos:

Funções

São blocos de código reutilizáveis chamados por nome e podem receber parâmetros e retornar valores.

Objetos e *arrays*

Objetos têm pares chave-valor, usados para agrupar informações.
Arrays são listas ordenadas de valores.

Operadores

São utilizados para matemática, comparação e lógica.



Você pode se perguntar

“Por que os comandos em JavaScript são em inglês?”

Quando foi criada em 1990, a programação em inglês era dominante na comunidade. Além disso, o inglês também é amplamente reconhecido como uma língua internacional de comunicação.

Bloco 4

Explorando outros tipos de linguagens de
programação.

Quem sou eu?

Analise os códigos em JavaScript abaixo e diga a qual elemento se refere.

```
if (idade >= 18) {  
    console.log("Você é maior de idade.");
```

```
let contador = 0;
```

```
// Calcula a média dos três números  
let media = (numero1 + numero2 + numero3) / 3;
```

```
let temperaturaCelsius = 25;  
let temperaturaFahrenheit = (temperaturaCelsius * 9/5) + 32
```



Gabarito

E aí, acertou?

```
if (idade >= 18) {  
    console.log("Você é maior de idade.");
```

Estruturas de controle

```
let contador = 0;
```

Declaração de variáveis

```
// Calcula a média dos três números  
let media = (numero1 + numero2 + numero3) / 3;
```

Comentários

```
let temperaturaCelsius = 25;  
let temperaturaFahrenheit = (temperaturaCelsius * 9/5) + 32
```

Instruções

Outras linguagens de programação

Além da linguagem JavaScript, é essencial conhecer outras linguagens de programação para desenvolver aplicativos móveis.

Swift: usada para desenvolver aplicativos iOS (iPhone e iPad).

Kotlin: usada para desenvolver aplicativos Android.

Dart: usada para desenvolver aplicativos com o *framework* Flutter.

Plataformas e ambientes de desenvolvimento

Familiarize-se com as plataformas, ambientes de desenvolvimento e *frameworks* específicos para cada sistema operacional.

Plataformas

Xcode: ambiente de desenvolvimento integrado (IDE) para iOS.

Android Studio: ambiente de desenvolvimento para criação de aplicativos Android.

Visual Studio Code: pode ser usado para desenvolver aplicativos multiplataforma com *frameworks*, como React Native e Flutter.

Frameworks

React Native: para desenvolvimento multiplataforma usando JavaScript e React.

Flutter: para desenvolvimento multiplataforma usando a linguagem Dart.

NativeScript: para desenvolvimento multiplataforma usando JavaScript e tecnologias web.

É importante se atentar para a interface

Além disso, é necessário entender os princípios de *design* responsivo, criação de interface do usuário (UI) e experiência do usuário (UX).

Aprenda a integrar aplicativos com serviços externos, consumir APIs e lidar com dados provenientes da web.

Saiba como armazenar dados localmente no dispositivo, como bancos de dados SQLite e armazenamento local.

Desenvolva habilidades em testes unitários, testes de integração e depuração de aplicativos para garantir que funcionem corretamente.

Aprenda como preparar, compilar e publicar seu aplicativo nas lojas de aplicativos.

Entenda como manter e atualizar seus aplicativos para fornecer correções de *bugs*, melhorias no sistema e novos recursos.

Esteja atualizado com as tendências atuais e emergentes em desenvolvimento móvel, como inteligência artificial e realidade aumentada.

Bloco 5

Explorando os ambientes de desenvolvimento.

Explorando ambientes de desenvolvimento

Para começar a realizar aplicações com JavaScript, será necessário utilizar um ambiente de desenvolvimento.

Na aula de hoje, utilizaremos o **code.org** para explorar os aspectos a seguir:

- criar uma conta;
- criar um projeto;
- iniciar um desenvolvimento;
- apresentar os elementos presentes na tela;
- escrever um código;
- rodar um código.



Navegue pelo conhecimento!

A large orange rectangular callout box contains the text "Navegue pelo conhecimento!" in bold orange letters. Below the text is an icon of a globe with a cursor arrow pointing towards it, symbolizing global navigation or search.

Bloco 6

Chegou a hora de praticar tudo que o vimos até aqui!

Chegou a hora de colocar tudo em prática!

Crie uma calculadora de Índice de Massa Corporal (IMC) em JavaScript.

O IMC é uma medida que ajuda a avaliar se uma pessoa está com um peso saudável com base em sua altura e peso. Confira as regras!

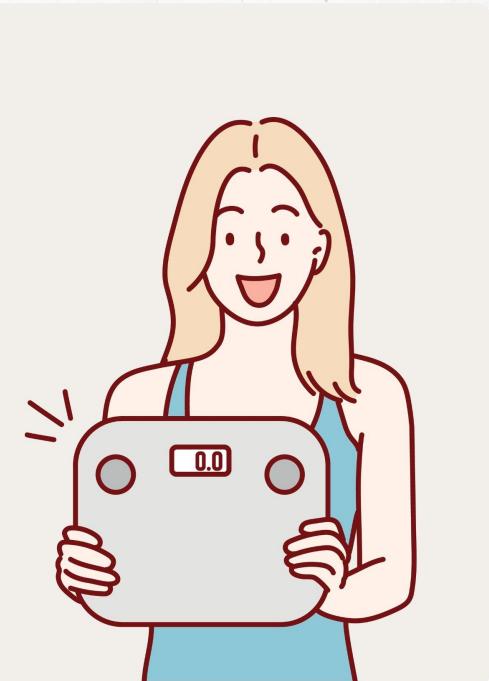
Crie uma página HTML simples com campos de entrada para o peso e a altura, além de um botão de "Calcular IMC".

Use JavaScript para capturar os valores inseridos pelo usuário e calcular o IMC.

Exiba o resultado do cálculo do IMC na página, informando se o usuário está abaixo do peso, com peso normal, com sobre peso ou obeso.

Forneça um *feedback* ao usuário de acordo com o resultado do IMC.

Adicione comentários ao seu código para explicar o que cada parte faz.

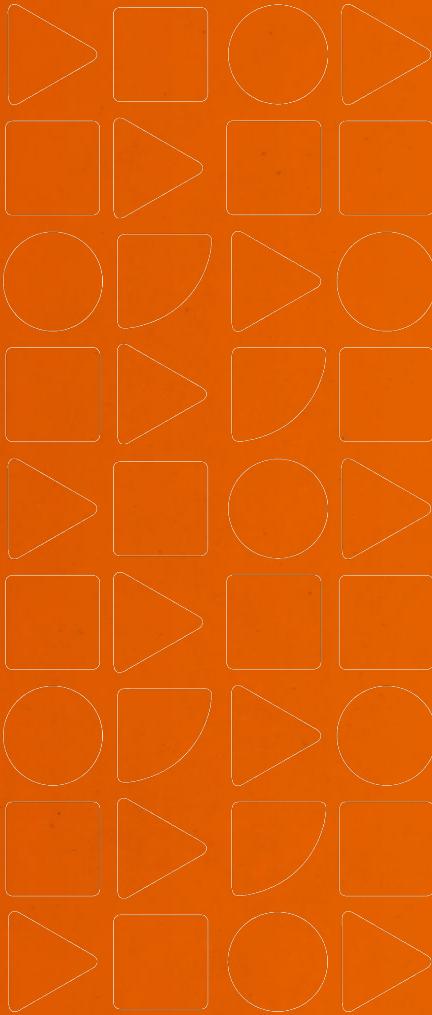


Fechamento

O que eu já sabia?

O que eu não sabia?

O que eu quero aprender?



Referências Bibliográficas

PROZ EDUCAÇÃO. *Apostila de Desenvolvimento para Dispositivos Móveis I.* 2023.