Exercício de Fixação JavaScript

Nome: Maria Thaise

Data: 02-10-2025

Seções no documento:

● 1. Variáveis e Tipos

● 2. Operadores e Expressões

● 3. Estruturas de Controle

● 4. Loops e Repetições

● 5. Funções

● 6. Mini-casos práticos

● 7. Reflexão

2.1 Variáveis e Tipos

● Qual a diferença entre var, let e const?

var não se usa mais

let variável que pode ser alterada

const variável que é constante não pode ser alterada

● Liste os tipos primitivos do JavaScript com exemplos.

1- String

let nome = "João";

let sobrenome = 'Silva';

let mensagem = `Olá, ${nome}!`; // Template literal

2-Number

let idade = 25;

let preco = 19.99;

let temperatura = -5;

let infinito = Infinity;

3-Boolean

let isAtivo = true;

let isAdmin = false;

let maiorDeIdade = idade >= 18;

4-Undefined

let variavelIndefinida;

console.log(variavelIndefinida); // undefined

let obj = {};

console.log(obj.propriedadeInexistente); // undefined

5-Null

let valorNulo = null;

let elemento = document.getElementById("inexistente"); // null

6-Symbol

let simbolo1 = Symbol("descricao");

let simbolo2 = Symbol("descricao");

console.log(simbolo1 === simbolo2); // false - cada Symbol é único

// Symbol bem conhecido

let iterador = Symbol.iterator;

7-BigInt

let numeroGrande = 9007199254740991n;

let outroBigInt = BigInt("123456789012345678901234567890");

console.log(numeroGrande + 1n); // 9007199254740992n

● Qual a diferença entre null e undefined?

Null representa ausência intencional de valor

Undefined variável declarada mas não inicializada

● Explique == e ===.

=== Igualdade estrita compara dois valores para igualdade

== Igualdade ampla compara dois valores para a igualdade, após converter ambos os valores para um tipo comum

2.2 Operadores e Expressões

● Liste operadores matemáticos: +, -, \*, /.

* Soma
* Subtração

\* Multiplicação

/ Divisão

● Liste operadores lógicos: &&, ||, !.

&& - Retorna verdadeiro apenas se ambas condições forem verdadeiras

|| - Retorna verdadeiro se pelo menos uma das condições forem verdadeiras

! - Inverte o valor da condição

● Preveja os resultados: ○ "5" + 2 ○ true + 1

2.3 Estruturas de Controle

● Explique if, else if e else.

if(se) Executa um bloco de código se a condição for verdadeira

else if(senão se) Verifica outra condição se a condição anterior for falsa

else(senão) Executa o bloco de código se a condição do if for falso

● Como usar switch?

Se a condição for correspondida,o programa executa as instruções [associadas .](http://associadas.se)

[Se](http://associadas.se) múltiplos casos corresponderem o valor,o primeiro caso que corresponder é selecionado, mesmo se os casos não forem iguais entre si.

● Escreva um exemplo de verificação de maioridade.

let idade = 25;

let faixaEtaria;

// Determinando a faixa etária primeiro

if (idade < 0) faixaEtaria = "invalida";

else if (idade < 12) faixaEtaria = "crianca";

else if (idade < 18) faixaEtaria = "adolescente";

else if (idade < 60) faixaEtaria = "adulto";

else faixaEtaria = "idoso";

switch (faixaEtaria) {

case "crianca":

console.log("👶 Criança - Menor de idade");

console.log("Foco em estudos e brincadeiras!");

break;

case "adolescente":

console.log("👦 Adolescente - Menor de idade");

console.log("Prepare-se para a maioridade!");

break;

case "adulto":

console.log("👨 Adulto - Maior de idade");

console.log("Você tem todos os direitos e responsabilidades civis.");

break;

case "idoso":

console.log("👴 Idoso - Maior de idade");

console.log("Você tem direitos adicionais como idoso.");

break;

case "invalida":

console.log("❌ Idade inválida!");

break;

default:

console.log("Categoria não reconhecida");

}

2.4 Loops e Repetições

● Liste os tipos de loops: for, while, do...while.

for- Executa o bloco de código um número específico várias vezes

while- Executa um bloco de código enquanto uma condição for verdadeira

do…while- Executa um bloco de código pelo menos uma vez,depois repete enquanto a condição for verdadeira

● Escreva mentalmente como imprimir números de 1 a 5.

● Explique break e quando usá-lo.

O break é uma instrução que interrompe imediatamente a execução de :

Loops(for,while,do…while), estrutura switch e bloco de códigos rotulados.

2.5 Funções ● O que é uma função?

● Diferença entre função declarada e função expressa.

função declarada - È uma função principal importante do sistema

função expressa - Precisa de funções como valores

● Crie uma função que recebe um nome e retorna saudação.

function saudar(nome) {

return `Olá, ${nome}!`;

}

// Uso

console.log(saudar("Maria")); // "Olá, Maria!"

console.log(saudar("João")); // "Olá, João!"

2.6 Mini-casos práticos

● Verificação de número par ou ímpar.

function verificarParImpar(numero) {

if (numero % 2 === 0) {

return "par";

} else {

return "ímpar";

}

}

● Criação mental de uma lista de compras (array).

● Somar números de 1 a 10 usando loop.

let soma = 0;

for (let i = 1; i <= 10; i++) {

soma += i;

}

2.7 Reflexão

● Por que conhecer tipos e operadores ajuda a programar melhor?

**tipos e operadores são a base da lógica de programação.** Saber usá-los corretamente torna o código **mais eficiente, legível e confiável.**

● Por que usar console.log() é importante para debug?

Usar **console.log()** é importante para debug porque ele funciona como uma “lupa” para entender o que está acontecendo dentro do código.

● Como planejar variáveis, funções e loops antes de programar?

**planejar é prever a estrutura** → quais dados usar (variáveis), como dividir o problema (funções) e onde repetir tarefas (loops). Isso deixa o código mais **organizado, reutilizável e fácil de entender**.