

Introdução ao JavaScript e Configuração do Ambiente no VS Code (4h)

O que é JavaScript?

JavaScript é uma linguagem de programação **interpretada**, utilizada principalmente para criar **interatividade** em páginas web. Ele roda nos navegadores e também pode ser usado no backend com Node.js.

Principais Aplicações do JavaScript

- Desenvolvimento Web (Frontend e Backend)
 - Aplicações Mobile (com frameworks como React Native)
 - Desenvolvimento de APIs e Microserviços
 - Automação de tarefas
-

Configuração do Ambiente no VS Code

Antes de começar a programar em JavaScript, precisamos configurar o ambiente.

Instalando o VS Code

O **Visual Studio Code (VS Code)** é um editor de código leve e poderoso para JavaScript. **Baixe e instale** em: <https://code.visualstudio.com/>

Instalando o Node.js

O Node.js permite executar JavaScript fora do navegador, útil para testar códigos no terminal. **Baixe e instale** em: <https://nodejs.org/>

Verifique a instalação:

Abra o **terminal** no VS Code (Ctrl + `) e digite:

```
node -v
```

Se aparecer uma versão (v18.16.0, por exemplo), está instalado corretamente.

Configurando um Projeto JavaScript no VS Code

1. **Criar uma pasta para o projeto**
 - No VS Code, vá em Arquivo → Abrir Pasta e selecione uma pasta.
2. **Criar um arquivo JavaScript**
 - No VS Code, clique com o botão direito na pasta do projeto e escolha **Novo Arquivo**.
 - Nomeie como **script.js**.
3. **Testando o JavaScript**
 - Abra o **terminal** no VS Code (Ctrl + `).
 - Digite:

```
node script.js
```

4.

Se tudo estiver certo, o terminal executará seu código JavaScript.

Conceitos Básicos de JavaScript

Variáveis e Constantes

Para armazenar valores, usamos `let`, `var` e `const`.

```
var nome = "João"; // Pode ser redeclarada  
let idade = 25; // Pode ser reatribuída  
const PI = 3.1415; // Não pode ser alterado  
  
console.log(nome, idade, PI);
```

Tipos de Dados no JavaScript

Em JavaScript, os dados podem ser classificados em diferentes tipos.

Tipos Primitivos

Os tipos primitivos são valores simples e imutáveis.

Tipo	Descrição	Exemplo
String	Cadeia de caracteres	"Olá, mundo!"
Number	Números inteiros e decimais	42, 3.14
Boolean	Verdadeiro ou falso	true, false
Undefined	Variável sem valor definido	let x; // undefined
Null	Representa ausência de valor	let y = null;

Exemplo no Código

```
let nome = "João"; // String
let idade = 25; // Number
let aprovado = true; // Boolean
let nota; // Undefined
let resultado = null; // Null

console.log(nome, idade, aprovado, nota, resultado);
```

Variáveis e Constantes

Para armazenar dados, usamos **variáveis** (let, var) e **constantes** (const).

Diferença entre let, var e const

Palavra-chave	Pode ser redefinir?	Escopo
var	Sim	Global ou função
let	Sim	Bloco de código

const	Não	Bloco de código
-------	-----	-----------------

Exemplo

```
var cidade = "São Paulo"; // Pode ser redeclarada
let pais = "Brasil"; // Pode mudar de valor
const PI = 3.1415; // Não pode ser alterado

console.log(cidade, pais, PI);
```

Operadores Básicos

Os operadores são usados para manipular valores.

Operadores Aritméticos

Operador	Operação
+	Adição
-	Subtração
*	Multiplicação
/	Divisão
%	Módulo (resto da divisão)

```
let a = 10;
let b = 3;
console.log(a + b); // 13
console.log(a - b); // 7
console.log(a * b); // 30
console.log(a / b); // 3.33
console.log(a % b); // 1
```

Operadores Relacionais

Comparam valores e retornam `true` ou `false`.

Operador	Comparação	Exemplo
>	Maior que	10 > 5 (true)
<	Menor que	5 < 10 (true)
>=	Maior ou igual	10 >= 10 (true)
<=	Menor ou igual	5 <= 10 (true)
==	Igual	"10" == 10 (true)
===	Estritamente igual	"10" === 10 (false)

```
console.log(10 > 5); // true
console.log(10 === "10"); // false (tipos diferentes)
console.log(10 == "10"); // true (conversão implícita)
```

Operadores Lógicos

Usados para combinar expressões.

Operador	Nome	Exemplo
&&	E (AND)	true && false → false
	OU(OR)	true false → true

!	Negação (NOT)	<code>!true → false</code>
---	---------------	----------------------------

```
let idade = 20;
console.log(idade > 18 && idade < 30); // true
console.log(idade < 18 || idade > 65); // false
console.log(!false); // true
```

Estruturas Condicionais

```
let idade = 18;

if (idade >= 18) {
  console.log("Maior de idade");
} else {
  console.log("Menor de idade");
}
```

Estruturas de Repetição

For

```
for (let i = 1; i <= 5; i++) {
  console.log("Número:", i);
}
```

While

```
let contador = 1;
while (contador <= 5) {
  console.log("Contador:", contador);
  contador++;
}
```

Funções

```
function saudacao(nome) {  
    return "Olá, " + nome;  
}  
  
console.log(saudacao("Ana"));
```

Extensões Úteis para JavaScript no VS Code

No VS Code, vá até **Extensões (Ctrl + Shift + X)** e instale:

ESLint (para corrigir erros de código).

JavaScript (ES6) Code Snippets (atalhos para código).

Prettier - Code Formatter (para organizar código).

Exercícios Práticos

Agora que você já aprendeu os conceitos básicos de JavaScript e configurou o VS Code, pratique com os exercícios abaixo:

Criando e Executando um Script Simples

1. Crie um arquivo `script.js`.
2. Escreva um código que exibe "Olá, Mundo!" no console.
3. Execute no terminal com `node script.js`.

```
console.log("Olá, Mundo!");
```

Criando e Manipulando Variáveis

1. Declare três variáveis (nome, idade e cidade).
2. Exiba uma mensagem no console com esses dados.

```
let nome = "Carlos";
```

```
let idade = 28;
let cidade = "São Paulo";

console.log(`Meu nome é ${nome}, tenho ${idade} anos e moro em
${cidade}.`);
```

Estrutura de Decisão

1. Crie um código que verifica se um número é **par** ou **ímpar**.
2. Exiba a mensagem correspondente no console.

Criando um Contador

1. Escreva um código que conte de **1 a 100** usando um laço for.

Criando uma Função

1. Crie uma função chamada soma() que recebe dois números e retorna a soma.

```
function soma(a, b) {
  return a + b;
}

console.log(soma(5, 3));
```

Quer um desafio?

1. Verificação de Idade

Crie um código que peça a idade do usuário e diga se ele é **menor de idade**, **adulto** ou **idoso**.

2. Tabuada

Crie um **loop for** que imprima a tabuada do número **5**. até 20

3. Verificação de Números Pares

Crie um código que imprima **todos os números pares de 1 a 20**.

Código Calculadora - Situação de Aprendizagem de Sala de Aula 19/02

```
//criar uma calculadora Simples em JS

//import da biblioteca

const prompt = require("prompt-sync") ();


//funções - Operações

//soma

function soma(a, b){

    return (a+b);

}

//subtração

function sub(a,b){

    return (a-b);

}

//multiplicação

function multi(a,b){

    return (a*b);
```

```
}

//divisão

function div(a,b) {

    return (a/b);

}


//menu Operações

function menuOperacoes() {

    console.log("===Calculadora Simples===");

    console.log("| 1. Soma                |");

    console.log("| 2. Subtração            |");

    console.log("| 3. Multiplicação        |");

    console.log("| 4. Divisão              |");

    console.log("=====");

    let operacao = prompt("Escolha a Operação Desejada: ");

    //entrada dos n°s

    let numero1 = Number(prompt("Digite n° 1: "));

    let numero2 = Number(prompt("Digite n° 2: "));

    let resultado;

    //condicional

    switch (operacao) {

        case "1":

            resultado = soma(numero1,numero2);
```

```
        break;

    case "2":

        resultado = sub(numero1,numero2);

        break;

    case "3":

        resultado = multi(numero1,numero2);

        break;

    case "4":

        if(numero2==0){

            console.log("Não Dividirás por Zero!!!");

            resultado = null;

        }else{

            resultado=div(numero1,numero2);

        }

        break;

    default:

        console.log("Operação Inválida")

        break;

} //fim switch

console.log ("Resultado: "+resultado);

}

// execução do programa
```

```
var continuar = true;

while (continuar) {

    menuOperacoes();

    let sair = prompt("1. Continuar | 2. Sair | ");

    if (sair == "2") {

        continuar = false;

        console.log("Saindo...");

    }

}
```