

Dia 2 - Node.js

Criando um programa Javascript

Centro de Alta Performance - SECITECI

SECITECI
SECRETARIA DE
ESTADO DE CIÊNCIA E
TECNOLOGIA E INOVAÇÃO



GOVERNO DE
**MATO
GROSSO**

<Fic_Dev>
Programador de Sistemas

Roteiro

- 1 Desafio
- 2 Objetivo da aula
- 3 Primeiro programa javascript
 - Usando o node.js
- 4 IDE VSCode
- 5 Aspectos gerais da sintaxe
 - Identação
 - Comentários
 - Identificadores
 - Atribuição
 - Escopo global e local
- 6 Entrada e saída de dados

Organização de Código em JavaScript

Você recebeu um código JavaScript para calcular a soma, a média dos números presentes em um array e exibir a classificação de cada número como par ou ímpar. No entanto, o código está desorganizado, com problemas de indentação e falta de comentários. Sua tarefa é organizar o código corretamente, corrigir a indentação, adicionar comentários relevantes em cada seção do código e identificar os escopos das variáveis utilizadas.

Ah, e não se esqueça. O código tem de fazer o que foi pedido.

Objetivo da disciplina

Entender alguns princípios da sintaxe da linguagem Javascript.

Saber trabalhar com comentários, indentação, atribuição e rastreamento de algoritmos.

Saber trabalhar com uma IDE.

Begin

- Interpretador Javascript - analisa e interpreta o código fonte escrito em JavaScript.
- Gera uma representação interna do código que pode ser executada pelo computador.
- Rodar um código javascript:
 - Usar console do navegador
 - Usar o Node.js
 - Usar IDE

Usando o navegador

```
<html>
  <head>
    <title>Meu tutorial JavaScript</title>
  </head>
  <body>
    <h1>Meu tutorial JavaScript</h1>
    <script>
      console.log("Hello world");
    </script>
  </body>
</html>
```

Com o node.js

- ❶ Crie um arquivo .js com o seguinte código: `console.log("Hello, world!");`
- ❷ Abra o terminal do seu sistema operacional.
- ❸ Navegue até o diretório onde o seu arquivo JavaScript está localizado usando o comando `cd`. `cd /Documentos`
- ❹ Execute o código JavaScript usando o comando `node` seguido do nome do arquivo. `» node meuCodigo.js`
- ❺ O resultado da execução do código será exibido no terminal.

A saída será exibida no terminal :

Hello, world!

VSCode

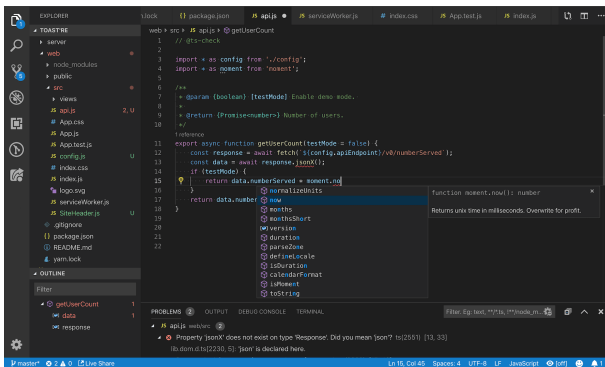


Figura: IDE VS Code

VSCode

- Crie um arquivo .js pela IDE
- Abra o terminal da IDE
- Execute o comando visto no slide anterior

Sintaxe

- Case sensitive
- Identação
- Ponto-e-vírgula
- Comentários
- Variáveis
- Constantes
- Tipos de dados
- Operadores
- Estruturas de controle, como como if/else, switch, for, while...

Identação - Princípios

- Hierarquia
- Clareza

Identação

```
// código sem indentação  
if (saldo == 0) {  
  let emprestimo = 2;  
  if (emprestimo == 2) {  
    let divida = 3;  
    console.log(divida);  
  }  
}
```

Identação

```
// código com indentação
if (saldo == 0) {
  let emprestimo = 2;
  if (emprestimo == 2) {
    let divida = 3;
    console.log(divida);
  }
}
```

Comentários Simples

- Comentários de uma linha são usados para incluir anotações curtas.
- Começam com //

Exemplo:

```
// Define uma variável chamada 'idade' com o valor de 20  
let idade = 20;
```

Comentário de bloco

- Comentários de várias linhas são usados para comentar blocos de código ou para incluir anotações mais longas.
- Começam com `/*` e terminam com `*/`

Exemplo:

```
/* Esta função calcula o cubo de um número. Para calcular o cubo, o  
número é multiplicado por si mesmo três vezes. */
```

Identificadores

- As palavras-chave `var`, `let` e `const` são usadas em JavaScript para declarar identificadores

Exemplo:

```
var numero = 5; // variável declarada com "var"  
let texto = "Hello"; // variável declarada com "let"  
const pi = 3.14; // constante declarada com "const"
```


Atribuição

- O operador de atribuição básico é o sinal de igual (=).
- Ele atribui um valor a uma variável: $x = 5$
- Essa operação atribui a x o valor 5.

Escopo global

- O escopo global é o escopo mais amplo, onde as variáveis são definidas fora de qualquer bloco de código, como funções ou loops.
- Variáveis globais podem ser acessadas e modificadas em qualquer parte do programa.

Escopo Global

```
//Variáveis globais devem ter nomes descritivos e
//ser declaradas fora de qualquer bloco de código
let contadorGlobal = 0;
function addContadorGlobal() {
    // A variável contadorGlobal pode ser acessada
    // e modificada dentro da função
    contadorGlobal++;
    console.log('Contador global é agora: ${contadorGlobal}');
}
// Chama a função para incrementar a variável globalCount
addContadorGlobal();
```

Escopo local

- O escopo local é o escopo dentro de um bloco de código.
- Variáveis locais só podem ser acessadas dentro do bloco onde foram definidas.

Escopo Local

```
function calculoAreaCirculo(raio) {  
    // Variáveis locais devem ser declaradas dentro  
    // do bloco de código onde são usadas  
    const PI = 3.14159;  
    //A variável area só pode ser acessada dentro desta função  
    let area = PI * raio * raio;  
    console.log('A área do circulo é ${area}');  
}  
// Chama a função para calcular a área  
calculoAreaCirculo(5);
```

Entrada - prompt() - HTML

- `variavel = prompt("Texto que vai aparecer:")`

Entrada de dados

```
<!DOCTYPE html>
<html>
  <head>
    <title>Curso JavaScript Progressivo</title>
    <script type="text/javascript">
      var nome = prompt("Digite seu nome: ")
      document.write("Bem vindo, ", nome);
    </script>
  </head>
</html>
```

Entrada - prompt() - console

```
var nome = prompt("Digite seu nome: ")  
alert(nome)
```


Saída - prompt()

JavaScript consegue escrever os dados de diferentes maneiras:

- Escrevendo em uma caixa de alerta, usando `window.alert()`
- Escrevendo para a saída HTML usando `document.write()`
- Escrevendo em um elemento HTML, usando `innerHTML`
- Escrevendo para o console do navegador, usando `console.log()`

Saída - window.alert()

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First Web Page</h1>
    <p>My first paragraph.</p>
    <script>
      window.alert(5 + 6);
    </script>
  </body>
</html>
```

Saída - document.write()

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First Web Page</h1>
    <p>My first paragraph.</p>
    <script>
      document.write(5 + 6);
    </script>
  </body>
</html>
```

Saída - innerHTML

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First Web Page</h1>
    <p>My First Paragraph</p>
    <p id="demo"></p>
    <script>
      document.getElementById("demo").innerHTML = 5 + 6;
    </script>
  </body>
</html>
```

Saída - console.log()

```
<!DOCTYPE html>
<html>
  <body>
    <h1>My First Web Page</h1>
    <p>My first paragraph.</p>
    <script>
      console.log(5 + 6);
    </script>
  </body>
</html>
```

Questionamentos?

Vamos ao desafio !