

Apresentação da Disciplina

Dia 3 Organização de código

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 disciplina
- 3 Pré-requisitos
- 4 Conteúdo

Desafio

Crie um programa em JavaScript que receba dois números pelo console. Retorne as seguintes informações: soma, subtração, multiplicação, qual número é maior e se os dois números são diferentes entre si.

Objetivo da disciplina

Entender a sintaxe dos comandos da linguagem.

Saber criar identificadores e constantes.

Saber trabalhar com os vários operadores em JavaScript.

Pré-requisitos

- Sólidos conhecimentos em **Lógica**
- Alto poder de abstração
- Muita Força de Vontade, Persistência e Atenção
- Ser autodidata ;-)

Conteúdo

- **Sintaxe básica do Javascript**

Conteúdo

- Tipos de dados:

JavaScript possui vários tipos de dados, como **números**, **strings**, **booleanos**, **arrays**, **objetos** e muito mais.

Cada tipo de dado é usado para representar um conjunto específico de valores e tem um conjunto específico de operações que podem ser realizadas com ele.

Conteúdo

- Tipos de dados primitivos

Conteúdo - Tipos de dados

- **Number:**

São usados para representar valores numéricos. Podem ser inteiros ou decimais.

```
let idade = 18;  
let altura = 1.76;  
typeof idade; // imprime number  
typeof altura; // imprime number
```

Conteúdo

- **String:**

São usados para representar valores de texto. Podem ser definidos usando aspas simples ou duplas.

```
let nome = "João";  
let sobrenome = 'Silva';  
typeof nome; // imprime string  
typeof sobrenome; // imprime string
```

Conteúdo

- **Booleanos:**

São usados para representar valores verdadeiro ou falso.

```
let aprovado = true;  
let reprovado = false;  
typeof aprovado; // imprime boolean  
typeof reprovado; // imprime boolean
```

Conteúdo

- **Null:**

É usado para representar um valor nulo ou vazio.

```
let valor = null;  
typeof valor; // imprime object
```

Em JavaScript o tipo “**null**” é um valor primitivo que representa a ausência intencional de qualquer valor de objeto

Conteúdo

- **Undefined:**

É usado para representar uma variável que ainda não foi definida.

```
let x;  
let y = undefined;  
typeof x; // imprime undefined  
typeof y; // imprime undefined
```

Conteúdo

- Operadores

Conteúdo

São usados para realizar operações **matemáticas**, **lógicas** e de **atribuição em variáveis**.

Eles são ferramentas essenciais para escrever um **código limpo e eficiente**.

Conteúdo

- Operadores Aritméticos

Os operadores aritméticos são utilizados para realizar operações matemáticas em variáveis.

Precedência:

parênteses - exponenciação - multiplicação, divisão e resto - adição e subtração

Operador	Descrição
+	adição
-	subtração
*	multiplicação
/	divisão
%	resto da divisão
**	exponenciação

Conteúdo

- Exemplos:

```
let x = 3;
```

```
let y = 2;
```

```
// Operações
```

```
let soma = x + y; // 5
```

```
let subtracao = x - y; // 1
```

```
let multi = x * y; // 6
```

```
let div = x / y; // 1.5
```

```
let resto = x % y; // 1
```

```
let potencia = x ** y; // 9
```

```
let precendencia = (x + y) * 2; // 10
```

Conteúdo

- Operadores de Atribuição

Os operadores de atribuição são utilizados para atribuir valores à variáveis. Os mais utilizados são:

Operador	Descrição
=	simples
+=	com adição
-=	com subtração
*=	com multiplicação
/=	com divisão
%=	com resto da divisão
**=	com exponenciação

Conteúdo

- Exemplos:

```
let x = 0;
```

```
// Operações
```

```
x += 2; //  $x = x + 2$ ;
```

```
x -= 2; //  $x = x - 2$ ;
```

<Fic_Dev>

Programador de Sistemas

Conteúdo

- Operadores de Comparação

Permitem que você compare dois valores e verifique se eles são **iguais** ou **diferentes**, **maiores** ou **menores** que um determinado valor.

Eles retornam um valor booleano (**true** ou **false**) dependendo do resultado da comparação.

Conteúdo

Operador	Descrição
==	iguais, sem levar em conta o tipo de dado.
===	iguais, levando em conta o tipo de dado.
!=	diferentes, sem levar em conta o tipo de dado.
!==	diferentes, levando em conta o tipo de dado.
>	maior
<	menor.
>=	maior ou igual.
<=	menor ou igual.

Conteúdo

- Exemplos:

```
let x = 5;  
// Operações  
x == 5 // true  
x == "5" // true  
x === "5" // false  
x != "5" // false  
x !== 5 // false  
x > 3 // true  
x < 10 // true  
x >= 5 // true  
x <= 4 // false
```

<Fic_Dev>

Programador de Sistemas

Questionamentos?

Muito Obrigado !