**JavaScript**

// // (única linha) ou /\*\*/ (comentário mais de uma linha)

Node

Atalho para entrar no terminal node: crtl+shift+”

Para sair, digite: .exit

Para limpar, digite: clear ou Ctrl+L

Cuidado com o nome das variáveis, no uso de letras maiúsculas

Para somar os números no prompt, temos que nos atentar, que o prompt sempre vai entender o input como string, então para colocar número, devemos converter os valores

Para converter um número para número inteiro: Number.parseInt(n)

Para converter um número para número com virgula: Number.parseFloat(n)

Concatenar e polarização

Para concatenar os valores, usamos a aspas simples (botão embaixo da tecla ESC)

Já para fazer a polarização, usamos a crase SHIFT+ASPAS (a aspas do lado do P)

var nome = window.prompt('Qual seu nome?')

        document.write(`Seu nome tem ${nome.length} letras.`)

        document.write(`Seu nome em maiúsculo fica  ${nome.toUpperCase()} <br/>`)

        document.write(`Seu nome em minúsculo fica ${nome.toLowerCase()} <br/>`)

**Formatando números:**

Se você tiver um número, por exemplo 123.3 mas quer que apareça depois do ponto, duas casas decimais, você pode usar o toFixed(quantidade de casas que você quer depois do ponto.)

Texto

Descrição gerada automaticamente

**Operadores**

/ divisão (exemplo 5/2 = 2.5 que é o número embaixo da chave)

% resto da divisão (exemplo 5/2 = 2, sobra 1 lá embaixo, então a resposta é 1)

\*\* elevado: 5\*\* = 25

Precedência de operadores:

Homem com óculos de grau

Descrição gerada automaticamente

Guardando os valores dentro de uma variável

Homem com óculos de grau

Descrição gerada automaticamente

Quando você tem 1 variável e quer fazer uma conta usando a mesma variável, você pode simplificar usando o +=

Por exemplo:

Var Num = 3

Num = num + 1

Nesse caso, o resultado ficaria 4. Mas quando você quer outro valor mas na mesma variável, fica assim

Exemplo:

Var Num = 3

Num = Num + 4

Para usar a simplificação ficaria: Num+=4 (mas só pode usar += se for a mesma variável)

Calendário

Descrição gerada automaticamente

**Operadores relacionais**

Nos valores relacionais:

!= significa diferente

== significa igualdade de grandeza

=== significa igualdade de grandeza e tipo. Por exemplo:

5 == 5 o resultado é true pois são iguais

5 == ’5’ apesar de estar dentro de aspas e parecer ser um string, o resultado é true, porque ele esta comparando se o valor igual, não se um valor é number ou não, string ou não.

5 === ‘5’ já aqui, ele valida se o valor e tipo são iguais

**Operadores relacionais**

! - Negação

É um operador unário, ou seja, vai sempre negar um elemento.

Texto

Descrição gerada automaticamente

&& - conjunção

É um operador binário, ou seja, com dois elementos. Os dois elementos têm que ser true para ser TRUE.

Calculadora preta com letras brancas

Descrição gerada automaticamente

|| - disjunção

É um operador binário, ou seja, com dois elementos. Basta um elemento ser true para ser TRUE.

Tela de celular com texto preto sobre fundo branco

Descrição gerada automaticamente com confiança baixa

**Sequência de operadores aritméticos e relacionais**

**Primeiro você resolve os aritméticos, depois os operadores relacionais e depois os lógicos**

Texto

Descrição gerada automaticamente

Resumindo a sequência de operadores:

Tela de um aparelho celular

Descrição gerada automaticamente

**Operadores ternário**

! - Negação

Texto

Descrição gerada automaticamente