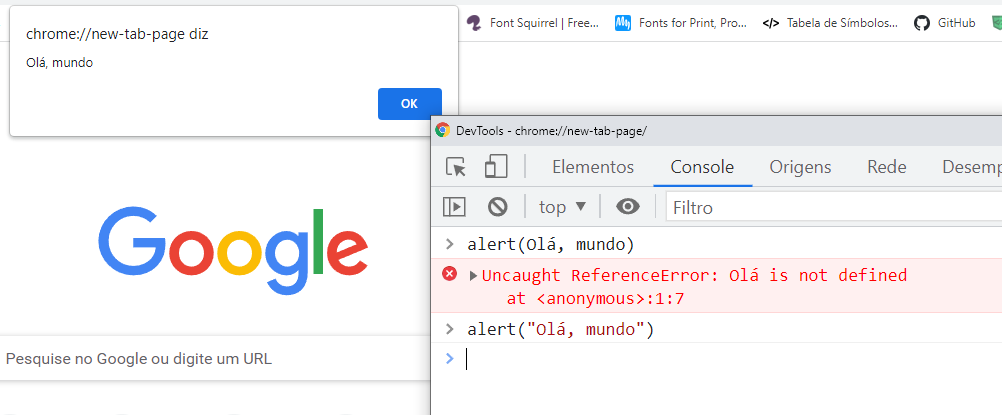
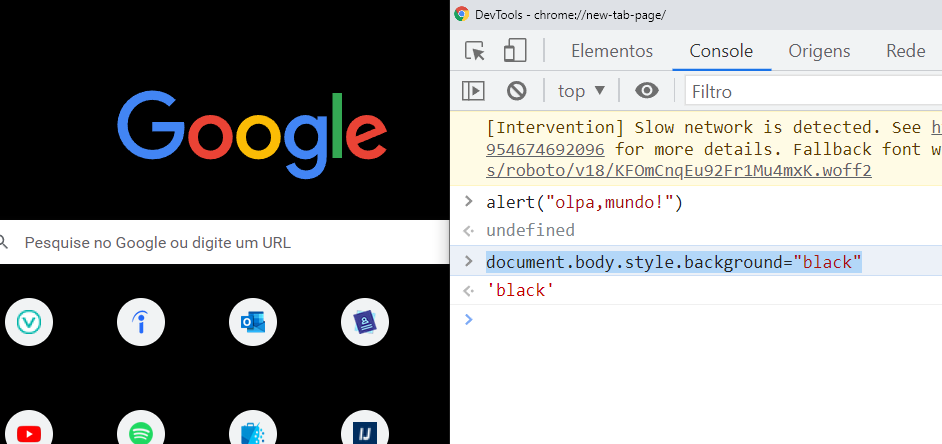
**JavaScript**

Podemos programar mensagens de exibição no google Chrome usando a ferramenta **inspecionar** > **Devtools** > **Console,** inserimos a tag **alert (“**mensagem**”)** e damos **enter, exemplo abaixo:**



Usando essa declaração **document.body.style.background="black",** podemos modificar a cor de fundo da tela inicial do google chrome, **exemplo abaixo:**



Podemos inserir mensagens de alerta usando a tag **script** onde antes inserimos a tag **window.alert (‘mensagem’);** **exemplo abaixo:**

<body>

    <h1>Olá, Mundo!</h1><br>

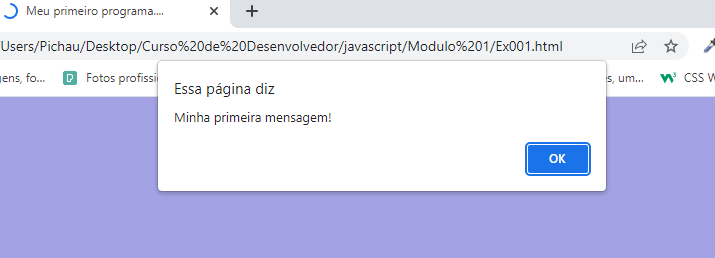
    <p>Já me livrei da maldição</p>

    <script>

        window.alert('Minha primeira mensagem!');

    </script>

</body>



Podemos inserir uma pergunta de confirmação para o usuário usando a tag **confirm** onde antes inserimos a **window.confirm(‘mensagem’);** **exemplo abaixo:**

<body>

    <h1>Olá, Mundo!</h1><br>

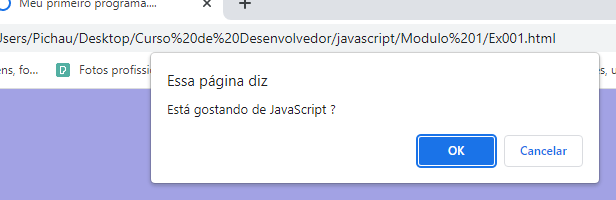
    <p>Já me livrei da maldição</p>

    <script>

        window.confirm('está gostando de JavaScript ?');

    </script>

</body>



Podemos fazer uma pergunta ao usuário usando a tag **prompt** onde antes inserimos **window.prompt(‘mensagem’); exemplo abaixo:**

<body>

    <h1>Olá, Mundo!</h1><br>

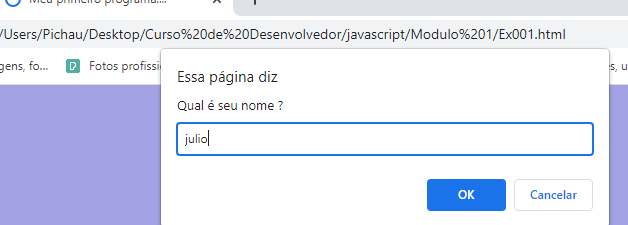
    <p>Já me livrei da maldição</p>

    <script>

        window.prompt('Qual é seu nome ?');

    </script>

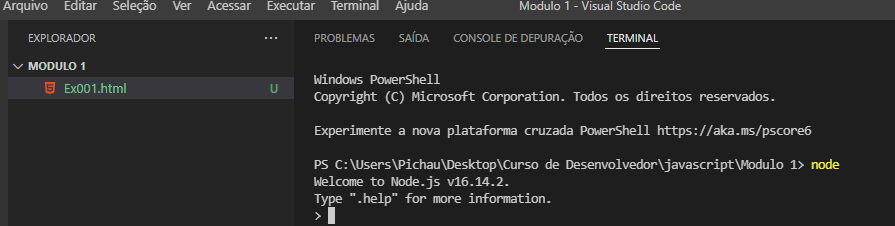
</body>



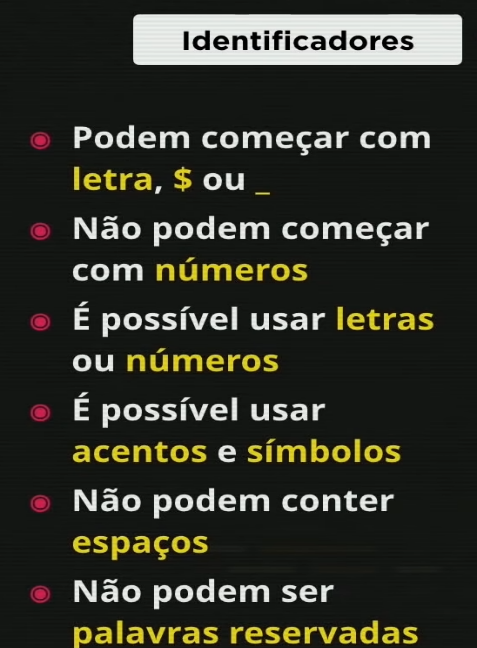
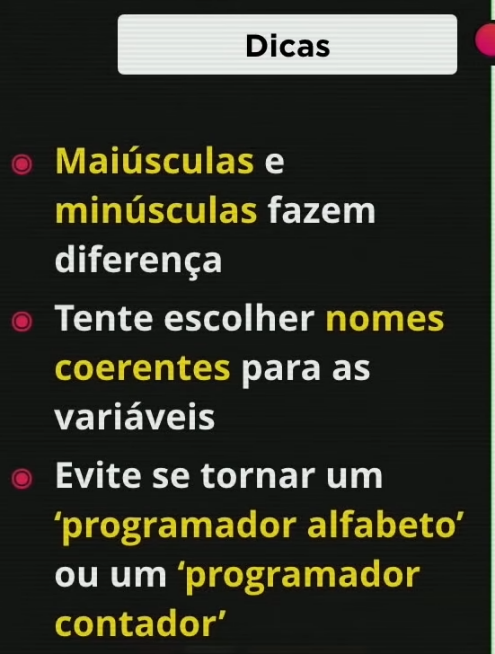
Para inserirmos comentários de até uma linha dentro do **Javascript** usamos a tag **//** e para inserirmos comentários com mais de 1 linha inserimos /\* \*/, lembrando que **/\* \*/** serve para o **CSS**, já para o **HMTL** usamos **<!—**

Para inserirmos variáveis usamos as tag **var** ou **let**, dentro do Javascript o sinal de **=** significa recebe.

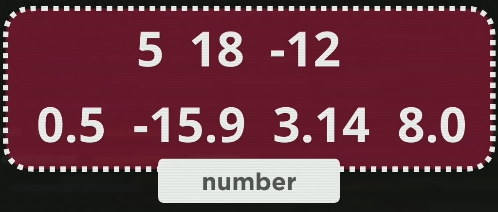
Podemos acessar o terminal dentro do **VScode** clicando em **terminal** > **novo terminal**, já dentro do terminal digitamos a palavra **node**, isso fará com que possamos abrir a tela do node dentro do **VScode**, exemplo abaixo:



Abaixo estão algumas características que os **identificadores** em **Javascript** precisam e podem ter:

Abaixo temos 3 tipos de dados **primordiais (data types – tipos de dados)**:

Usamos a tag **window.alert**(‘mensagem, ‘ + resposta do usuário), abaixo temos um exemplo de **string de concatenação:**

<body>

    <h1>Olá, Mundo!</h1><br>

    <p>Já me livrei da maldição</p>

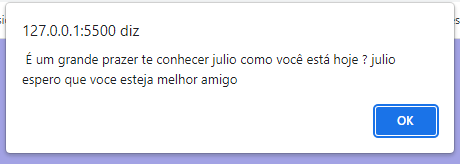
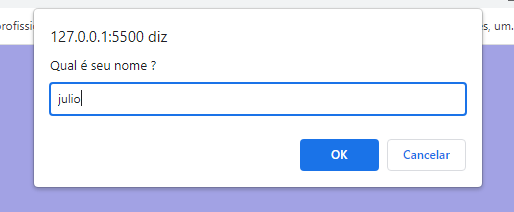
    <script>

       var nome = window.prompt('Qual é seu nome ?'); //vai ´perguntar o nome..

       window.alert(' É um grande prazer te conhecer ' +nome+ ' como você está hoje ? ' +nome+ ' espero que voce esteja melhor amigo ') // usamos o sinal de + para juntar a resposta que o usuário vai nos fornecer, isso precedente com ''

    </script>

</body>



Abaixo temos um exemplo de **string template**:

<body>

    <h1>Olá, Mundo!</h1><br>

    <p>Já me livrei da maldição</p>

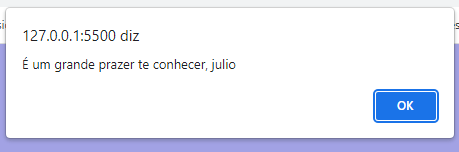
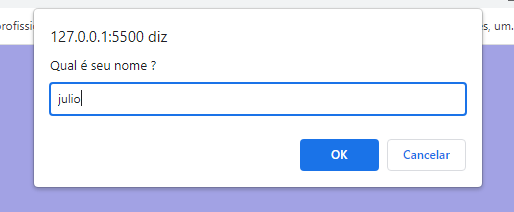
    <script>

       var nome = window.prompt('Qual é seu nome ?'); //vai ´perguntar o nome..

       window.alert(`É um grande prazer te conhecer, ${nome}`) // usamos o sinal de + para juntar a resposta que o usuario vai nos fornecer, isso precedente com ''

    </script>

</body>



Podemos converter **string** (nomes) para **numbers** (números) usando 2 tags, sendo elas:

**Number.parseInt(numero) –** usamos para ter retorno de números inteiros

**Number.parseFloat(numero) –** usamos para ter retorno de números quebrados

Exemplo abaixo usando **Number.parseInt – inteiros com string de concatenação**

<body>

    <h1>Olá, Mundo!</h1><br>

    <p>Já me livrei da maldição</p>

    <script>

        var n1 = Number.parseInt(window.prompt('Digite um número:')) //string

        var n2 = Number.parseInt(window.prompt('Digite outro número:')) //string

        var s = n1 + n2

        window.alert('A soma dos valores totais é ' + s)

        // (number+numer) server para adição

        // (string+string) serve para concatenação

    </script>

</body>

Exemplo abaixo usando **Number.parseFloat – números quebrados com string de concatenação**

<body>

    <h1>Olá, Mundo!</h1><br>

    <p>Já me livrei da maldição</p>

    <script>

        var n1 = Number.parseFloat(window.prompt('Digite um número:')) //string

        var n2 = Number.parseFloat(window.prompt('Digite outro número:')) //string

        var s = n1 + n2

        window.alert('A soma dos valores totais é ' + s)

        // (number+number) server para adição

        // (string+string) serve para concatenação

    </script>

</body>

Com a nova versão do **Javascript** usando a tag **Number(numero)** o próprio programa detecta se o **número** é **inteiro** ou **quebrado** e faz a soma**, exemplo abaixo com string de template:**

<body>

    <h1>Olá, Mundo!</h1><br>

    <p>Já me livrei da maldição</p>

    <script>

        var n1 = *Number*(window.prompt('Digite um número:')) //string

        var n2 = *Number*(window.prompt('Digite outro número:')) //string

        var s = n1 + n2

        window.alert(`A soma dos valores totais é ${s}`)

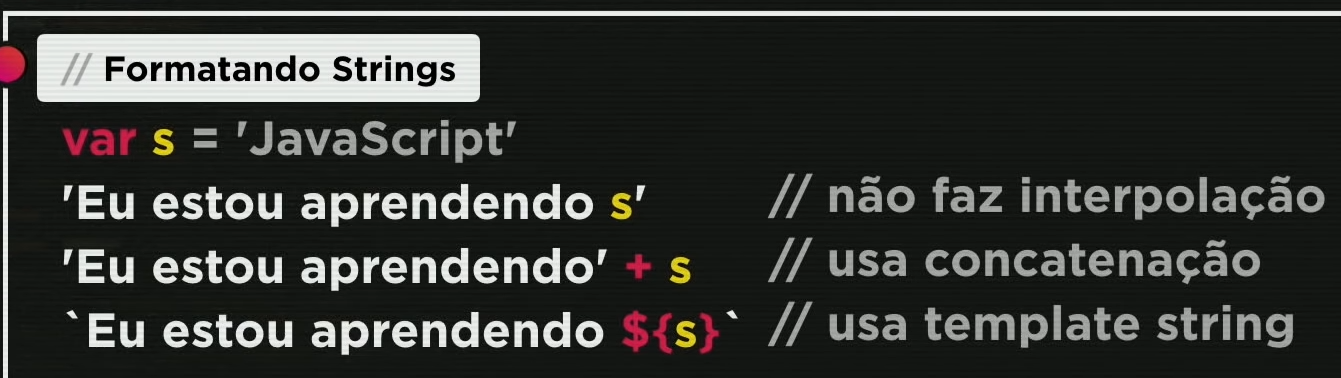
        // (number+number) server para adição

        // (string+string) serve para concatenação

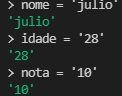
    </script>

</body>

**Usando strings abaixo:**

****

**Propriedades das variáveis:**

****

**Um exemplo de template string abaixo:**

****

**Um exemplo de string usando a concatenação (soma) abaixo:**

****

Abaixo um exemplo de declaração usando a **string template:**

<body>

    <h1>Olá, Mundo!</h1><br>

    <p>Já me livrei da maldição</p>

    <script>

        var n1 = *Number*(window.prompt('Digite um número:')) //string

        var n2 = *Number*(window.prompt('Digite outro número:')) //string

        var s = n1 + n2

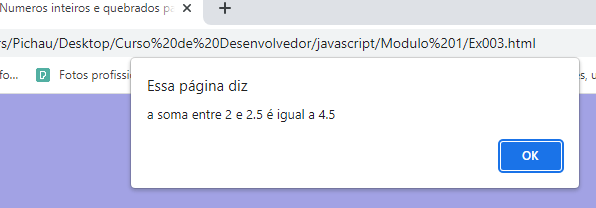
        window.alert(`A soma dos valores totais é ${s}`)

        // (number+number) server para adição

        // (string+string) serve para concatenação

    </script>

</body>

****

Abaixo um exemplo de declaração usando a **string de concatenação:**

<body>

    <h1>Olá, Mundo!</h1><br>

    <p>Já me livrei da maldição</p>

    <script>

        var n1 = *Number*(window.prompt('Digite um número:')) //string

        var n2 = *Number*(window.prompt('Digite outro número:')) //string

        var s = n1 + n2

        window.alert('A soma dos valores totais é ' + s)

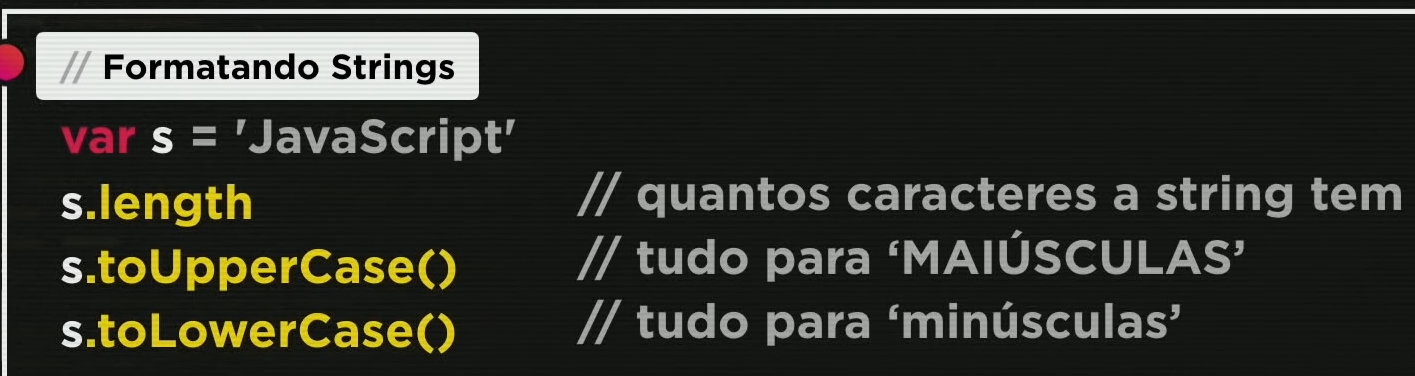
        // (number+number) server para adição

        // (string+string) serve para concatenação

    </script>

</body>

Abaixo mais variações de **strings** que podemos usar**:**



Podemos atribuir nome à variável, assim quando digitamos o cliente nos dar um nome a mesma vai retorna com a resposta digitada em texto dentro do site,além disso podemos atribuir um estilo em CSS, adicionarmos **parágrafos**, colocar a **fonte** em **negrito** ou colocar as **letras** **maiúsculas** ou **minúsculas**, **exemplo abaixo:**

    <style>

        body{

            font: normal 20pt arial;

        }

    </style>

</head>

<body>

    <script>

        var nome = window.prompt(`Qual e seu nome ?`)

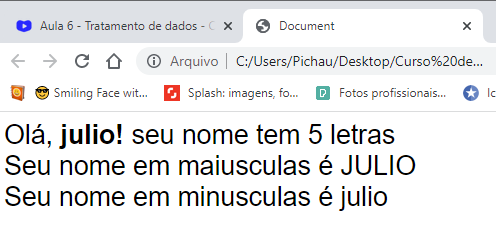
        document.write(`Olá, <strong>${nome}!</strong> seu nome tem ${nome.length} letras <br>`)

        document.write(`Seu nome em maiusculas é ${nome.toUpperCase()}<br>`)

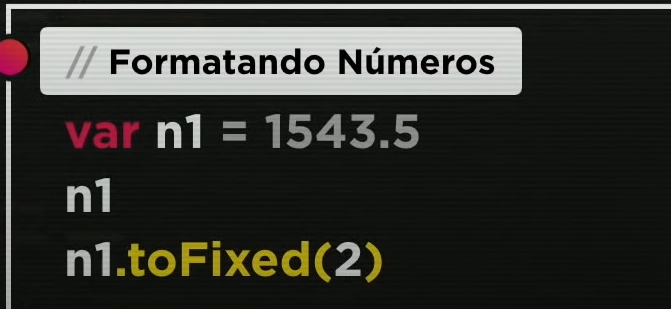
        document.write(`Seu nome em minusculas é ${nome.toLowerCase()}`)

    </script>

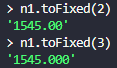
</body>

****

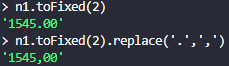
Podemos fazer a formatação de números seguindo as **strings** abaixo:



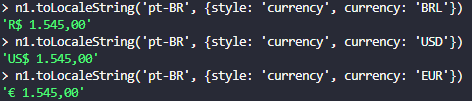
Com essa **string** estamos adicionando uma outra casa decimal usando o **toFixed(2)** em nossa **string**, podemos adicionar quantas casas decimais quisermos somente trocando o número 2.



Além disso podemos trocar a separação dos números de virgula por ponto ou vice e versa usando a declaração **toFixed(2).replace(‘.’,’,’)**

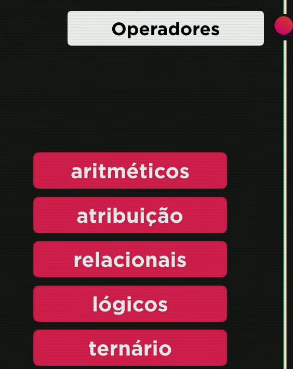


Além disso podemos colocar um estilo em nossa **string** como por exemplo o valor da moeda de cada país.

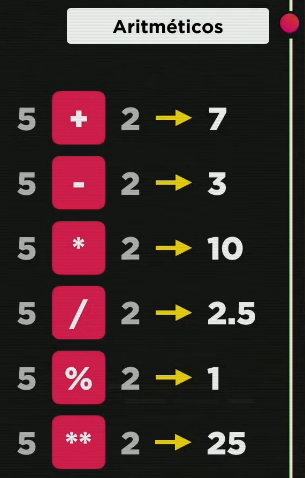


**Operadores em JS**

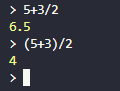
Abaixo temos alguns tipos de **operadores:**



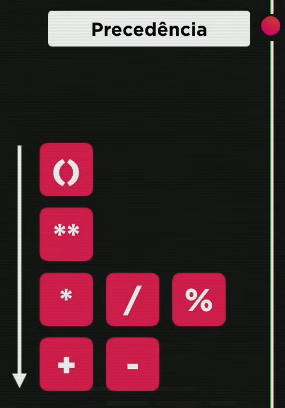
Iniciamos pelos operadores **aritméticos**, eles são usados para fazer **cálculos:**

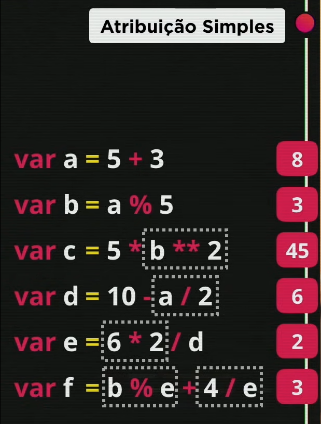


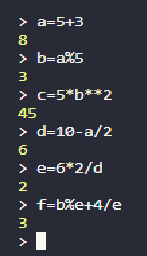
Devemos respeitar e estar atentos as formas de somar, multiplicar e dividir os números, pois existe uma logica chamada **ordem de precedência.**



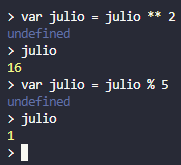
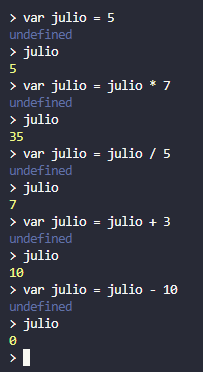
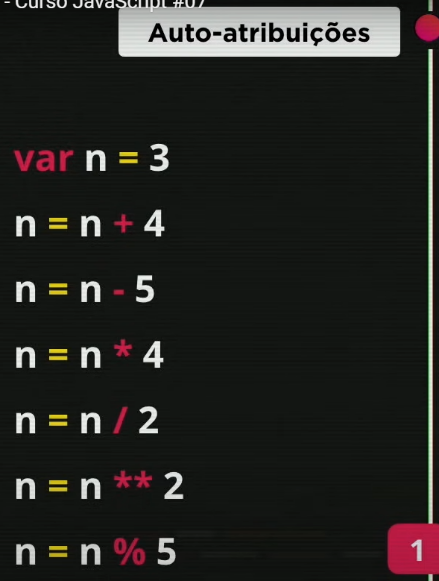
Abaixo está a ordem de **precedência**, o que significa quais logicas serão executas por **primeiro** e assim por sequencia em **ordem decrescente**:

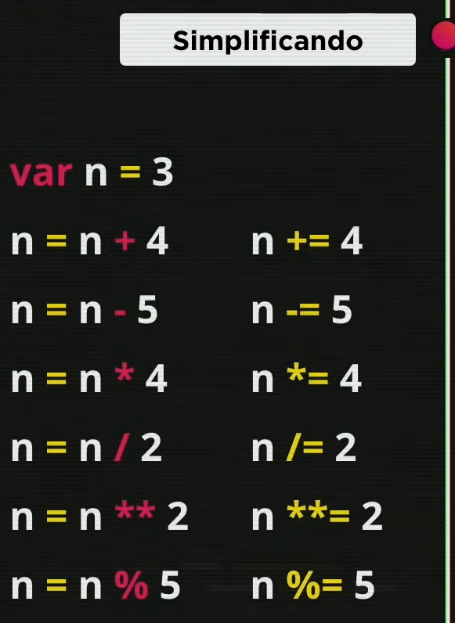
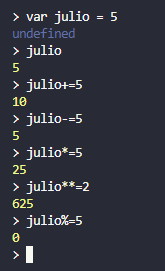


Abaixo algumas operações de **atribuições simples:**

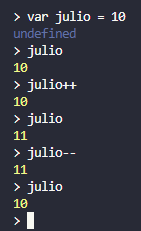
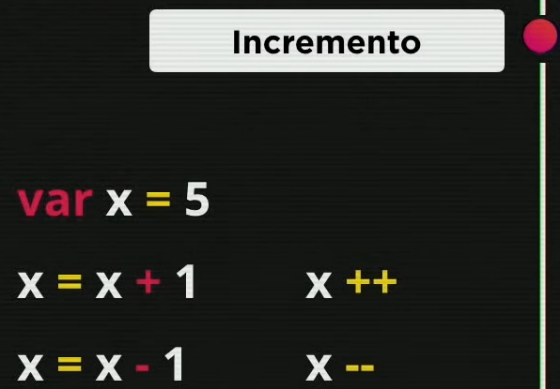


Também temos as **auto atribuições**, que são quando operações que podemos fazer com a mesma variável, mudando assim seus valores:

  
Temos uma maneira de **simplificar** estas auto atribuições, e estas simplificações são usadas por uma grande variedade de linguagens:



Também a uma outra forma chamada de **incremento** onde podemos realizar somas e diminuições atribuindo um valor para a variável e depois o nome da variável seguido de ++ ou -- :



Abaixo veremos outros tipos de operadores, começando pelos **relacionais,** na operação dentro do **node.js** onde temos operadores **relacionais** e **aritméticos**, primeiro resolvemos os **aritméticos** e depois os **relacionais:**

