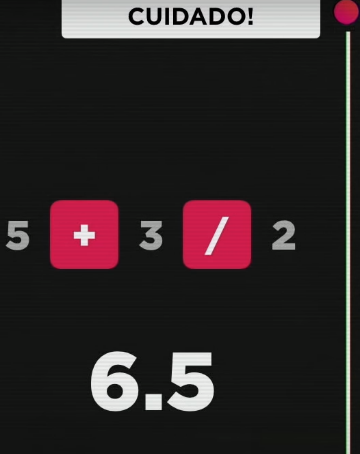
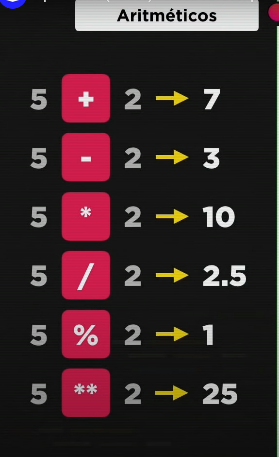
**Operadores**

Na imagem abaixo podemos ver os operadores aritméticos, esses operadores são os mais usados, existem vários outros, mas esses são os que mais vamos usar.

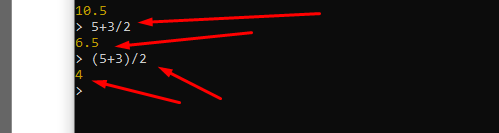


Como vemos na imagem ao lado, nas operações aritméticas, a resposta é 6.5, isso se da porque a divisão é feita antes da soma.

Tome cuidado quando for programar, pois mesmo que não seja apresentado erro, o resultado pode ser errado, pois existe erro de sintaxe.

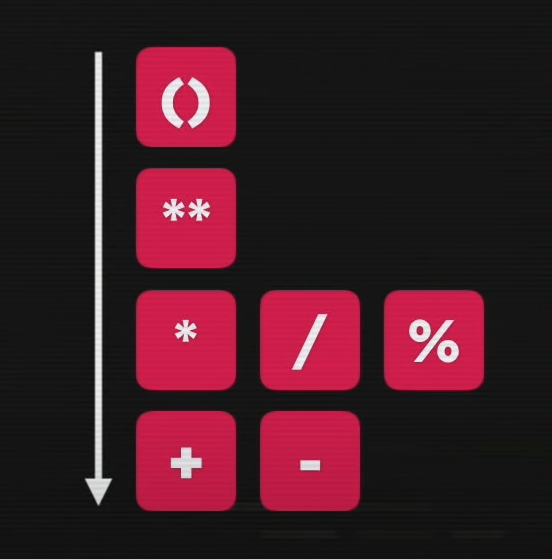
Quando você quiser forçar uma operação, ou seja, se você que o sistema some antes de dividir, basta que você coloque a soma entre parêntese.

Veja o exemplo na imagem a seguir:



Note que quando usamos a entre parêntese ele soma e depois divide, dando resultado 4.

***Ordem de precedência de operações.***

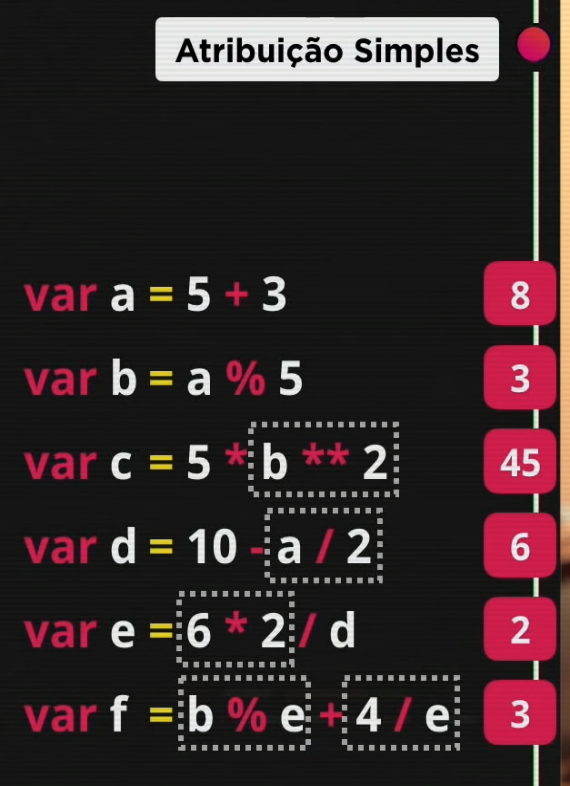


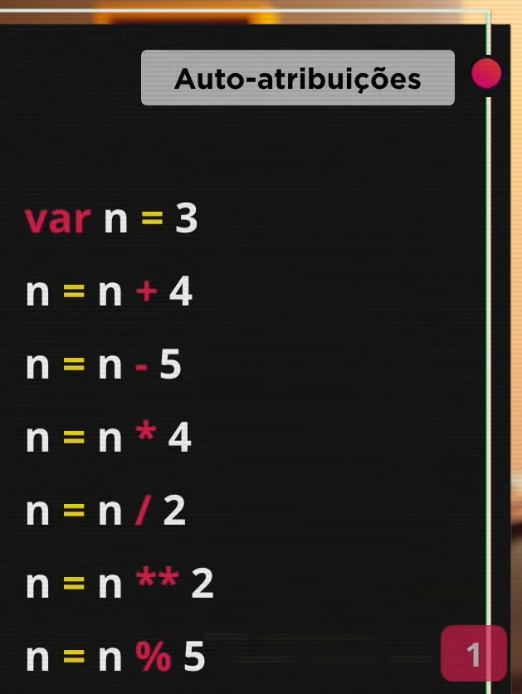
Essa é ordem precedência dos operadores matemáticos, primeiro ele executa o que está entre parêntese, depois o que é potência, multiplicação, divisão, porcentagem, soma e subtração.

Veja na imagem ao lado que eles funcionam de cima para baixo, assim como a seta está apontando para baixo.

Os operadores que estão na mesma linha, no caso de estarem em uma mesma operação, eles funcionarão da esquerda para a direita, no caso ao lado onde temos 3 operadores, fazemos primeiro a multiplicação, depois a divisão e depois a porcentagem. Logo o que temos 2 fazemos primeiro a soma e depois a subtração.

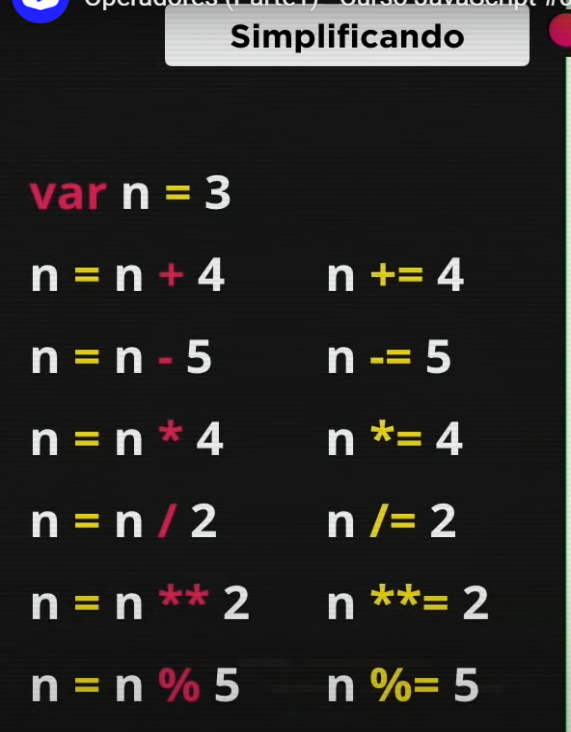
Vejamos abaixo alguns exemplos simples de ordem de precedência, para os operadores já vistos:

Vamos considerar na imagem ao lado que cada letra fica valendo o valor que ela recebeu antes.



O ‘var’ significa que estamos declarando uma variável, e a letra que procede é a variável em si.

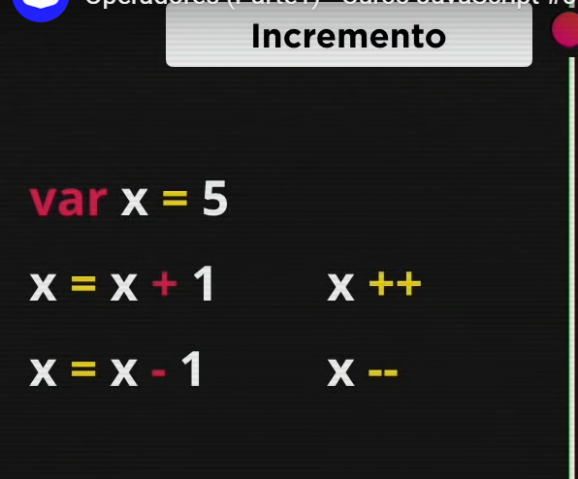
Veja os exemplos de ordem de precedência e analise cada um deles, pratique no seu node, coloque outros valores, até que toda a ordem de precedência seja totalmente entendida por você.



Maneira simplificada de fazer as auto atribuições.

Veja que em vez de se usar as duas letras n para uma receber a outra e depois o número, colocamos somente um ‘n’ seguido da operação mais o sinal de atribuição que é o igual (=).

Esse tipo de sintaxe é aceito pela grande maioria das linguagens de programação que existe hoje em dia, então não se preocupe em aprender essas sintaxes para usar somente em JavaScript, use-as também em outras linguagens de programação sem problema nenhum, e lembre-se sempre, pratique, pois somente assistir aula não te fazer aprender, o que nos faz aprender é a prática, invente exemplos e faça suas próprias operações aritméticas, eu garanto que você vai aprender muito mais.

Esses são as formas de fazer incremento, esse incremento pode funcionar para qualquer valor, porem no exemplo ao lado vemos somente com 1.

Esse método pode ser usado com a estrutura de repetição ‘enquanto’, que delimitará um valor inicial e um valor final, e enquanto esse valor final não for atingido ele não para de fazer a operação. Porem tome cuidado com os loops infinitos, pois eles acontecem muito na construção de sistemas, e podem comprometer seriamente se acontecer em um sistema profissional e trazer sérios problemas a ele.