

# Lógica

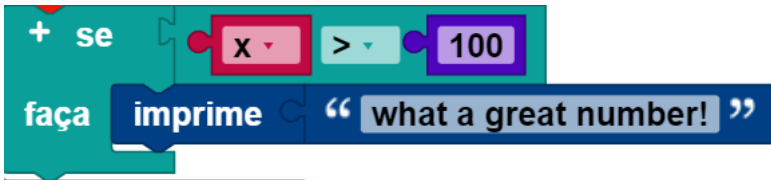
## Instruções condicionais

As instruções condicionais são fundamentais para a programação. Eles tornam possível formular distinções de caso, tais como:

- Se houver um caminho para a esquerda, vire à esquerda.
- Se pontuação = 100, imprima "Muito bem!".

### blocos **se**

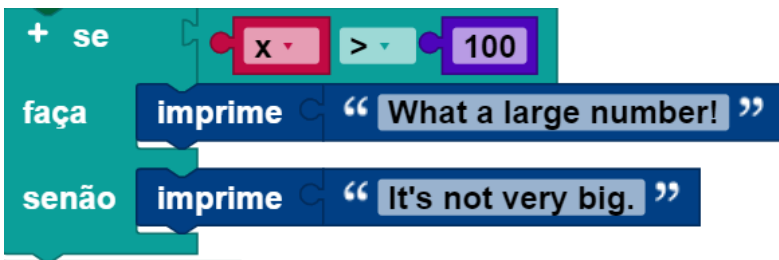
A condição mais simples é um bloco **se**:



Quando isso é executado, o valor da variável **x** é comparado a 100. Quando for maior, "Que grande número!" será emitido. Caso contrário, nada acontecerá.

### blocos **caso contrário**

Também é possível indicar que algo deve acontecer se a condição não for verdadeira, como neste exemplo:

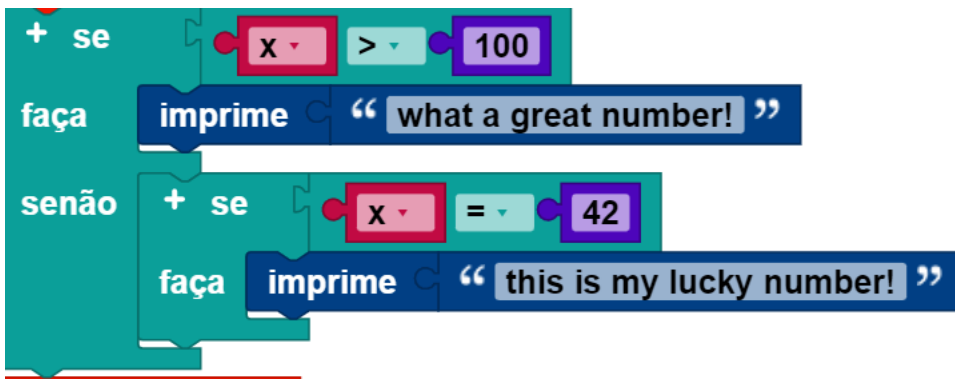


Como no bloco anterior, será emitido "Que grande número!" se  $x > 100$ . Caso contrário, "Isso não é muito grande." é especificado.

Um bloco **se** tem uma seção **caso contrário**, mas não mais de uma.

### blocos **caso contrário, se**

Também é possível testar várias condições com um único bloco **se** adicionando cláusulas **caso contrário se**:

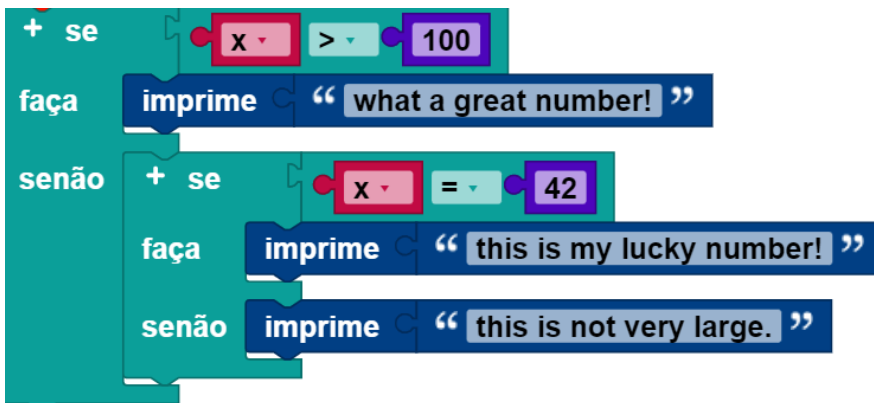


O bloco primeiro verifica se  $x > 100$  e retorna "Que número grande!", se for esse o caso. Se este não for o caso, ele verifica se  $x = 42$ . Em caso afirmativo, ele diz "Este é meu número da sorte!". Caso contrário, nada acontecerá.

Um bloco **se** pode ter qualquer número de seções **caso contrário se**. As condições são avaliadas de cima para baixo até que uma seja satisfeita, ou até que não haja mais condições.

## blocos **caso contrário**, **caso contrário**

Blocos **se** podem ter tanto seções **caso contrário se** como **caso contrário**:

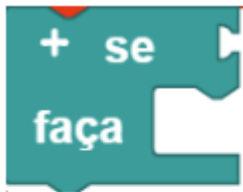


A seção **caso contrário** garante que uma ação será executada mesmo se nenhuma das condições anteriores for verdadeira.

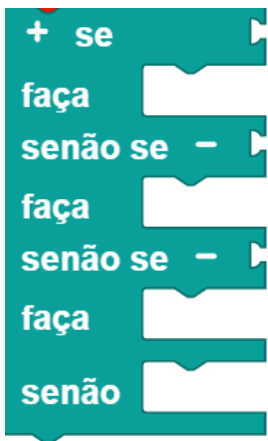
Uma seção **caso contrário** pode aparecer após qualquer número de seções **caso contrário se**, incluindo zero, e então será obtido um bloco **se caso contrário** perfeitamente normal.

## Modificação de bloco

Apenas o bloco **se** simples e o bloco **se caso contrário** aparecem na barra de ferramentas:



Para adicionar cláusulas **caso contrário se** e **caso contrário**, clique no símbolo (+). As cláusulas **caso contrário se** - podem ser removidas com o símbolo (-):



Observe que as formas dos blocos permitem qualquer número de sub-blocos **caso contrário se** a serem adicionados, mas apenas até um bloco **se**.

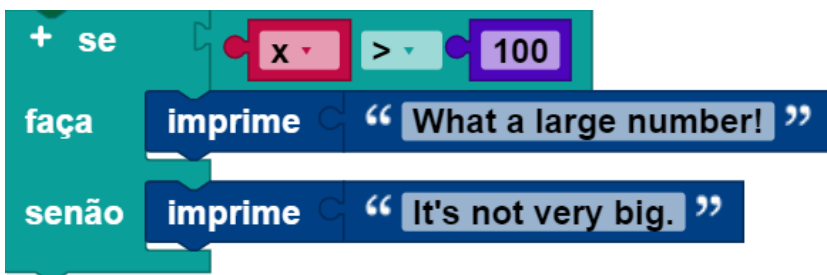
## Lógica booleana

A lógica booleana é um sistema matemático simples que possui dois valores:

- verdadeiro
- falso

Os blocos lógicos no ROBO Pro Coding geralmente existem para controlar condições e loops.

Aqui está um exemplo:



Se o valor da variável *x* for maior que 100, a condição é verdadeira e o texto "Que número grande!" é emitido. Se o valor de *x* não for maior que 100, a condição será falsa e "Isso não é muito grande." é emitido. Os valores booleanos também podem ser armazenados em variáveis e repassados para funções, assim como números, texto e valores de lista.

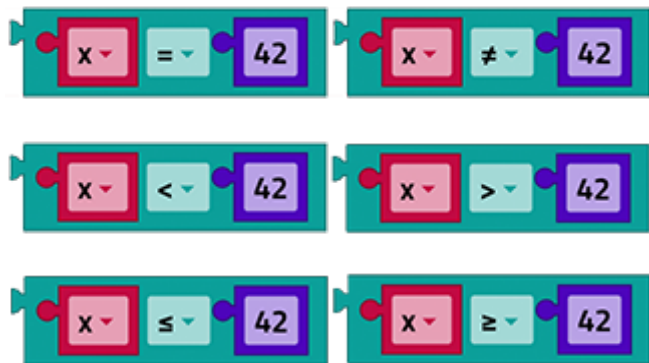
Se um bloco espera um valor booleano como entrada, uma entrada ausente é interpretada como **falsa**. Os valores não booleanos não podem ser inseridos diretamente onde os valores booleanos são esperados, embora seja possível (mas não aconselhável) armazenar um valor não booleano em uma variável e, em seguida, inseri-lo na entrada de condição. Este método não é recomendado e seu comportamento pode mudar em versões futuras do ROBO Pro Coding.

## Valores

Um único bloco com uma lista suspensa indicando **verdadeiro** ou **falso** pode ser usado para obter um valor booleano:

## Operadores de comparação

Existem seis operadores de comparação. Cada um recebe duas entradas (geralmente dois números) e o operador de comparação retorna **verdadeiro** ou **falso**, dependendo de como as entradas são comparadas.



Os seis operadores são: igual a, não igual a, menor que, maior que, menor ou igual a, maior ou igual a.

## Operadores lógicos

O bloco **e** retorna **verdadeiro** se e somente se seus dois valores de entrada forem verdadeiros.



O bloco **ou** retorna **verdadeiro** se pelo menos um de seus dois valores de entrada for verdadeiro.



## não

O bloco **não** converte uma entrada booleana em seu oposto. Por exemplo, o resultado é:



falso.

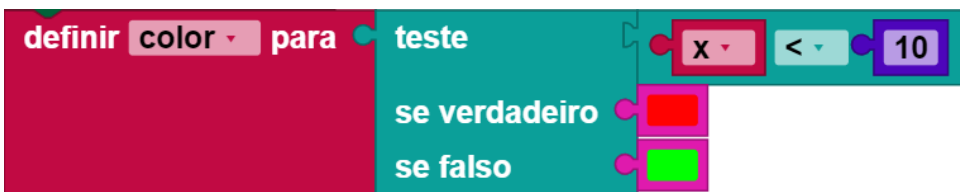
Se nenhuma entrada for feita, o valor é considerado **verdadeiro**, de modo que o bloco a seguir gera o valor **falso**:



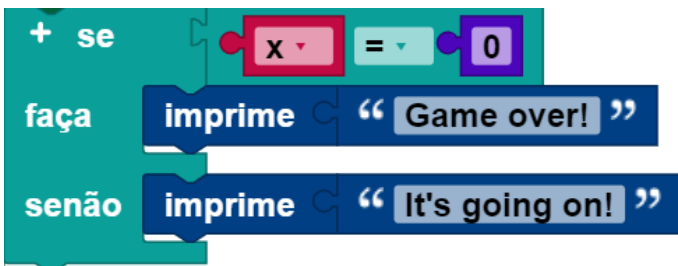
No entanto, não é recomendável deixar uma entrada em branco.

## três operadores

Os três operadores se comportam como um bloco **se-caso contrário** em miniatura. São necessários três valores de entrada: o primeiro valor de entrada é a condição booleana a ser testada, o segundo valor de entrada é o valor que deve ser retornado se o teste resultar em **verdadeiro**, o terceiro valor de entrada é o valor que deve ser retornado se o teste resultar em falso. No exemplo a seguir, a variável **cor** é definida como vermelho se a variável **x** for menor que 10, caso contrário, a variável **cor** é definida como verde.



Um bloco de três sempre pode ser substituído por um bloco **se-caso contrário**. Os dois exemplos a seguir são exatamente iguais.



Revision #8

Created 17 November 2021 21:07:27 by Admin

Updated 10 November 2024 15:43:52 by phuesing