



Aula 06

Operadores relacionais

Prof. Jalerson Lima

Apresentação

Nessa aula iremos continuar aprendendo sobre os operadores. Dessa vez vamos abordar os operadores relacionais, que também são conhecidos como operadores de comparação e nos permitem realizar comparações entre valores numéricos e variáveis.

Objetivos

1. Compreender a utilidade dos operadores relacionais;
2. Aprender como usar os operadores relacionais com valores numéricos e variáveis.

1. Introdução

Os operadores relacionais, também conhecidos como operadores de comparação, nos permitem realizar comparações entre dois valores distintos (ou duas variáveis). Mas antes de começarmos a discutir sobre esses operadores, vamos aprender um pouco sobre valores booleanos.

Um valor booleano é um tipo de dado primitivo que só pode assumir dois possíveis valores: verdadeiro, também conhecido como 1 (um) ou *true*; e falso, também conhecido como 0 (zero) ou *false*. Os valores booleanos vêm da Lógica de Boole, criada pelo inglês George Boole (TECMUNDO, 2015).

Mas por que estamos discutindo valores booleanos? Porque os operadores relacionais, na maioria das vezes, dão como retorno (resposta) valores booleanos. Ou seja, quando você fizer a comparação de dois valores ou variáveis usando operadores relacionais, o resultado será *true* (verdadeiro) ou *false* (falso). Há um único operador que resulta num valor inteiro ao invés de um valor booleano, que será apresentado a seguir.

2. Operadores relacionais em Ruby

A linguagem Ruby conta com sete operadores relacionais apresentados na Tabela 1.

Tabela 1 - Operadores relacionais em Ruby

Operador	Nome	Exemplo	Descrição
<code>==</code>	Igualdade	<code>a == b</code>	Verifica se o valor de <code>a</code> é igual ao valor de <code>b</code> .
<code>!=</code>	Diferente	<code>a != b</code>	Verifica se o valor de <code>a</code> é diferente do valor de <code>b</code> .
<code>></code>	Maior que	<code>a > b</code>	Verifica se o valor de <code>a</code> é maior do que o valor de <code>b</code> .
<code><</code>	Menor que	<code>a < b</code>	Verifica se o valor de <code>a</code> é menor que o valor de <code>b</code> .
<code>>=</code>	Maior ou igual	<code>a >= b</code>	Verifica se o valor de <code>a</code> é maior ou igual ao valor de <code>b</code> .
<code><=</code>	Menor ou igual	<code>a <= b</code>	Verifica se o valor de <code>a</code> é menor ou igual ao valor de <code>b</code> .
<code><=></code>	-	<code>a <=> b</code>	Retorna 0 se <code>a</code> for igual a <code>b</code> . Retorna 1 se <code>a</code> for maior que <code>b</code> e retorna -1 se <code>a</code> for menor que <code>b</code> .

Vamos ilustrar o uso dos operadores relacionais através de alguns exemplos. Confira a seguir.

```
irb(main):001:0> 1 == 1
=> true
irb(main):002:0> 1 == 2
=> false
```

```

irb(main):003:0> 1 != 1
=> false
irb(main):004:0> 1 != 2
=> true
irb(main):005:0> 2 > 3
=> false
irb(main):006:0> 2 < 3
=> true
irb(main):007:0> 2 >= 3
=> false
irb(main):008:0> 2 <= 3
=> true
irb(main):009:0> 5 <= 5
=> true
irb(main):010:0> 4 <=> 8
=> -1
irb(main):011:0> 10 <=> 2
=> 1
irb(main):012:0> 12 <=> 12
=> 0

```

Observe que o operador `<=>` é o único que retorna um valor inteiro, e não um valor booleano. Observe os exemplos a seguir para ver como os operadores relacionais também podem ser usados com variáveis.

```

1  a = 2
2  b = 4
3  puts a == b
4  puts a != b
5  puts a < b
6  puts a > b
7  puts a <= b
8  puts a >= b
9  puts a <=> b
10 gets

```

Exemplo de código 1 - Usando operadores relacionais com variáveis

Atividade 6.1

Crie um *script* Ruby e digite o código ilustrado no Exemplo de código 1. Por fim, execute o código e verifique o resultado apresentado pelo *script*.

Após executar o script apresentado no Exemplo de código 1, você deve observar o seguinte resultado.

```
false
true
true
false
true
false
-1
```

Atividade 6.2

Modifique o *script* Ruby apresentado no Exemplo de código 1 para permitir que o usuário digite os valores das duas variáveis, e melhore a apresentação dos resultados usando a substituição de expressões. Ao concluir essa atividade, seu *script* deverá apresentar algo parecido com o ilustrado abaixo.

```
Digite um número inteiro:
2
Digite outro número inteiro:
3
2 é igual que 3? false
2 é diferente que 3? true
2 é menor que 3? true
2 é maior que 3? false
2 é menor ou igual que 3? true
2 é maior ou igual que 3? false
```

Resumindo

Nessa aula aprendemos o segundo conjunto de operadores: os relacionais, que nos permitem realizar comparações entre valores numéricos e variáveis. Aprendemos que os operadores relacionais, em sua maioria, retornam valores booleanos, que só podem assumir dois possíveis valores: *true* ou *false*.

Referências

POINT, T. Ruby Strings. **Tutorials Point**, 2015. Disponível em: <http://www.tutorialspoint.com/ruby/ruby_strings.htm>. Acesso em: 04 nov. 2015.

SOUZA, L. **Ruby - Aprenda a programar na linguagem mais divertida**. 1ª. ed. São Paulo: Casa do Código, v. I, 2012.

TECMUNDO. Lógica booleana? Saiba um pouco mais sobre esta lógica e como ela funciona. **Tecmundo**, 2015. Disponível em: <<http://www.tecmundo.com.br/programacao/1527-logica-booleana-saiba-um-pouco-mais-sobre-esta-logica-e-como-ela-funciona.htm>>. Acesso em: 13 nov. 2015.