Comparison Operators

Comparison operators take simple values (numbers or strings) as arguments and used to check for equality between two values. Ruby provides following comparison operators :

|  |  |  |  |
| --- | --- | --- | --- |
| **Operator** | **Name** | **Example** | **Result** |
| == | Equal | x==y | True if x is exactly equal to y. |
| != | Not equal | x!=y | True if x is exactly not equal to y. |
| > | Greater than | x>y | Teue if x is greater than y. |
| < | Less than | x<y | True if x is less than y. |
| >= | Greater than or equal to | x>=y | True if x is greater than or equal to y. |
| <= | Less than or equal to | x<=y | True if x is less than or equal to y. |
| <=> | Combined comparison operator. | x<=>y | x <=> y : =  if x < y then return -1  if x =y then return 0  if x > y then return 1 if x and y are not comparable then return nil |
| === | Test equality | x===y | (10...20) === 9 return false. |
| .eql? | True if two values are equal and of the same type | x.eql? y | 1 == 1.0 #=> true  1.eql? 1.0 #=> false |
| equal? | True if two things are same object. | obj1.equal?obj2 | val = 10 => 10  val.equal?(10) => true |

**Example: Equality test**

puts ("Test two numbers for equality with ==, !=, or <=>")

puts 14 == 16

puts 14 != 16

puts 14 <=> 14

puts 14 <=> 12

puts 14 <=> 16

Copy

Output:

Test two numbers for equality with ==, !=, or <=>

false

true

0

1

-1

**Example: eql? and eqlity? operators**

irb(main):023:0> 1 == 1.0

=> true

irb(main):024:0> 1.eql?1.0

=> false

irb(main):025:0> obj1 = "123"

=> "123"

irb(main):026:0> obj2 = obj1.dup

=> "123"

irb(main):027:0> obj1 == obj2

=> true

irb(main):028:0> obj1.equal?obj2

=> false

irb(main):029:0> obj1.equal?obj1

=> true

irb(main):030:0>

Copy

**Example: Equal, less than, or greater than each other**

puts ("Test if two numbers are equal, less than, or greater than each other")

puts 14 < 16

puts 14 < 14

puts 14 <= 14

puts 14.0 > 12.5

puts 14.0 >= 14

Copy

Output:

Test if two numbers are equal, less than, or greater than each other

true

false

true

true

true

**Example: Spaceship operator returns -1, 0, or 1**

puts ("the <=> (spaceship operator) returns -1, 0, or 1,")

puts 2 <=> 3

puts 2 <=> 2

puts 3 <=> 2

Copy

Output:

the <=> (spaceship operator) returns -1, 0, or 1,

-1

0

1

**Example: Test the value in a range**

puts ("test if a value is in a range")

puts (12...16) === 8

puts (12...16) === 14

puts (12...16) === 16

puts (12...14) === 12

puts (12...16) === 14

Copy

Output:

test if a value is in a range

false

true

false

true

true