

# Ruby Operators *Cheatsheet*

---

## What Are Operators?

---

Is 2 equal to 4? Is your name Matilda? Is the sky blue?! Ruby operators are used to compare the values and types of properties and objects. They especially come in handy when you want to make a decision based on how two objects compare.

---

## Assignment Operators

---

**=**

Assigns a value to a variable name.

```
x = 'my_variable'
```

You can also do parallel assignments:

```
x, y, z = 100, 200, 300 # x equals 100, y equals 200, z equals 300
```

---

**+=**

Adds the two variables together and assigns the result to the first.

```
x = 3
y = 6
x += y # x is now equal to 9
```

---

**-=**

Subtracts the second variable from the first and assigns the result to the first variable.

```
x = 5
y = 2
x -= y # x is now equal to 3
```

---

**\*=**

Multiplies the second variable with the first and assigns the result to the first variable.

```
x = 4
y = 4
x *= y # x is now equal to 16
```

---

# Ruby Operators *Cheatsheet*

---

## **/=**

Divides the first variable by the second and assigns the result to the first variable

```
x = 10  
y = 2  
x /= y # x is now equal to 5
```

---

## **%=**

Divides the first variable by the second and assigns the remainder to the first variable.

```
x = 6  
y = 4  
x %= y # x is now equal to 2
```

---

## **\*\*=**

Performs an exponential (power) calculation and assigns result to the first variable.

```
x = 5  
y = 5  
x **= y # x is now equal to 3125
```

# Ruby Operators *Cheatsheet*

---

## Comparison Operators

---

**==**

Is equal to, compares values only.

```
x = 2 + 2
if x == 4
  puts 'good!'
end
```

---

**!=**

Is not equal to.

```
weather = 'sunny'
if weather != 'rain'
  puts "Looks like it'll be sunnier today"
end
```

---

**>**

Is greater than.

```
num_dogs = 7
if num_dogs > 1
  puts "#{num_dogs} dogs are better than 1 dog!"
end
```

---

**<**

Is less than.

```
num_dogs = 0
if num_dogs < 1
  puts 'You should get more dogs!'
end
```

---

# Ruby Operators *Cheatsheet*

---

**>=**

Is greater than or equal to.

```
num_dogs = 4
if num_dogs >= 2
  puts "I hope you have lots of Kibbles 'n Bits!"
end
```

---

**<=**

Is less than or equal to.

```
num_dogs = 1
if num_dogs <= 2
  puts "2 dogs are better than #{num_dogs}"
end
```

# Ruby Operators *Cheatsheet*

---

## Logical Operators

---

### && or and

Checks if ALL conditions are met.

```
num_cats = 1
num_dogs = 1
if num_cats == 1 && num_dogs == 1
  puts "Are they enemies or best friends?"
end
```

---

### || or or

Checks if ANY condition is met.

```
season = 'winter'
if season == 'fall' || season == 'winter'
  puts "Make sure you bring a coat!"
end
```

---

### ! or not

Reverses the result of a statement.

```
season = 'summer'
if !(season == 'winter')
  puts "We probably won't be getting any snow."
end
```

*Note that you could accomplish the same thing with ruby's [unless](#) while removing the need for the negation.*

# Ruby Operators *Cheatsheet*

---

## Math Operators

---

+

Adds.

```
x = 5
y = x + 7 # y is equal to 12
```

---

-

Subtracts.

```
x = 10
y = x - 2 # y is equal to 8
```

---

\*

Multiplies.

```
x = 10
y = x * 2 # y is equal to 20
```

---

/

Divides.

```
x = 10
y = x / 2 # y is equal to 5
```

---

\*\*

Exponent. Performs exponential (power) calculation.

```
x = 3
y = 2
x**y # x to the power of y
      # => 9
```

---

# Ruby Operators *Cheatsheet*

---

## %

Modulus. Returns the remainder of a division operation.

```
x = 9
x % 3 # remainder is equal to 0
x % 2 # remainder is equal to 1
```

---

## More Information About Operators

---

For more information about operators and all things Ruby, check out the [Ruby Documentation](#).