**Ruby Arithmetic Operators:**

**A=10 B=20**

|  |  |
| --- | --- |
| Operator | Example |
| + | a + b will give 30 |
| - | a - b will give -10 |
| \* | a \* b will give 200 |
| / | b / a will give 2 |
| % | b % a will give 0 |
| \*\* | a\*\*b will give 10 to the power 20 |

**Ruby Comparative Operators: use with conditional statements**

|  |  |
| --- | --- |
| Operator | Example |
| == | (a == b) is not true. |
| != | (a != b) is true. |
| > | (a > b) is not true. |
| < | (a < b) is true. |
| >= | (a >= b) is not true. |
| <= | (a <= b) is true. |
|  |  |
| === | (1...10) === 5 returns true. |
| **.eql?** | 1**.eql?**(1.0) is true. |
| **.equal?** | aObj**.equal?**bObj is true. |

**Ruby Assignment Operators:**

|  |  |
| --- | --- |
| Operator | Example |
| = | c = a + b will assign value of a + b into c |
| += | c += a is equivalent to c = c + a |
| -= | c -= a is equivalent to c = c - a |
| \*= | c \*= a is equivalent to c = c \* a |
| /= | c /= a is equivalent to c = c / a |
| %= | c %= a is equivalent to c = c % a |
| \*\*= | c \*\*= a is equivalent to c = c \*\* a |

**Ruby Logical Operators: use with conditional statements**

|  |  |
| --- | --- |
| Operator | Example |
| and | (a and b) is true. |
| or | (a or b) is true. |
| && | (a && b) is true. |
| || | (a || b) is true. |
| ! | !(a && b) is false. |
| not | not(a && b) is false. |

**Ruby Ternary operator:**

|  |  |  |
| --- | --- | --- |
| Operator | Description | Example |
| ? : | Conditional Expression | If Condition is true ? Then value a : Otherwise value b |

Example: puts a == true ? "a" : "b"

**Ruby Parallel Assignment:**

a = 10  
b = 20  
c = 30

a, b, c = 10, 20, 30

$a, $b, $c = 100, 200, 300

**Ruby Range operators:**

|  |  |
| --- | --- |
| Operator | Example |
| .. | 1..10 Creates a range from 1 to 10 inclusive  Example: print (1990..2015).to\_a |
| ... | 1...10 Creates a range from 1 to 9  Example: print (1990...2015).to\_a |

**Ruby *if...else* Statement:**

Syntax:

if (condition && condition)  
 puts ruby code...  
elsif condition

puts ruby code...  
else  
 puts ruby code...  
end

x = 0  
if x > 5  
 puts "x is greater than 5"  
elsif (x <= 5) && (x != 0)  
 puts "x is 4"  
else  
 puts "I can't guess the number"  
end

**Ruby *if* modifier:**

Syntax:

code if condition

$name = "srinivas"

puts "My name is #{$name}" if $name == "srinivas"

**Ruby *unless* Statement:**

Syntax:

unless conditional   
 code  
else  
 code   
end

Example:

x = 8  
unless x > 7  
 puts "x is less than 5"  
 else  
 puts "x is greater than 5"  
end

**Ruby *case* Statement**

Syntax:

case expression  
when expression  
 code …...  
else  
 code   
end

Example:

$age = 12

case $age  
when 0 .. 2  
 puts "baby"  
when 3 .. 6  
 puts "little child"  
when 7 .. 12  
 puts "child"  
when 13 .. 18  
 puts "youth"  
else  
 puts "adult"  
end