RubyNumbers

```ruby
# Ruby Numbers
# Usual operators:
# + addition
# - subtraction
# * multiplication
# / division


puts "1 + 2 = #{1 + 2}"
puts "2 * 3 = #{2 * 3}"

# # Integer division
# # When you do arithmetic with integers, you'll get integer answers
puts "3 / 2 = #{3 / 2}"
puts "10 - 11 = #{10 - 11}"

# # Check to see what these are
puts "1.5 / 2.6 = #{1.5 / 2.6}"

puts "3.class = #{3.class}"
puts "11.class = #{11.class}"
puts "1.5.class = #{1.5.class}"
puts "2.6.class = #{2.6.class}"
```

```ruby
# # Let's see what modulo in Ruby does.
puts "5 % 3 = #{5 % 3}"    # prints  2
puts "-5 % 3 = #{-5 % 3}"   # prints  1
puts "5 % -3 = #{5 % -3}"   # prints -1
puts "-5 % -3 = #{-5 % -3}"  # prints -2

# # The boolean OR
puts "nil || 2008  = #{nil || 2008}"
puts "false || 2008 = #{false || 2008}"
puts '"ruby" || 2008 = ' "ruby" || 2008
puts "2008 || nil  = #{2008 || nil}"
puts "2008 || false = #{2008 || false}"
e = 2008 || "ruby"
puts '2008 || "ruby" = ' << "#{e}"

@variable = @variable || "default value"
puts "@variable = #{@variable}"
@variable ||= "default value"
puts "@variable = #{@variable}"
```

```ruby
def g *args # The splat here says accept 1 or more arguments
            # in the form of an Array
  args      # This returns an array
end

def f arg
  arg
end

x,y,z = [true, 'two', false] # parrallel assignment lets us do this
puts "x = #{x}, y = #{y}, z = #{z}"

if a = f(x) and b = f(y) and c = f(z) then d = g(a,b,c) end

if a = f(x) and b = f(y) and c = f(z) then
  d = g(a,b,c) # An array is returned, and stored in variable d
end
# #p d # using p to puts and inspect d
puts "x and y and z makes d = #{d.class}"

x,y,z = [true, 'two', nil] # parrallel assignment lets us do this
puts "x = #{x}, y = #{y}, z = nil"
if a = f(x) and b = f(y) and c = f(z) then
   d = g(a,b,c) # An array is returned, and stored in variable d
end
# #p d # using p to puts and inspect d
puts "x and y and z makes d = #{d.class}"

x,y,z = [true, 'two', 'ten'] # parrallel assignment lets us do this
puts "x = #{x}, y = #{y}, z = #{z}"
if a = f(x) and b = f(y) and c = f(z) then
  d = g(a,b,c) # An array is returned, and stored in variable d
end
# #p d # using p to puts and inspect d
puts "x and y and z makes d = #{d.class}"
puts "d = #{d}"
```