



# Variables

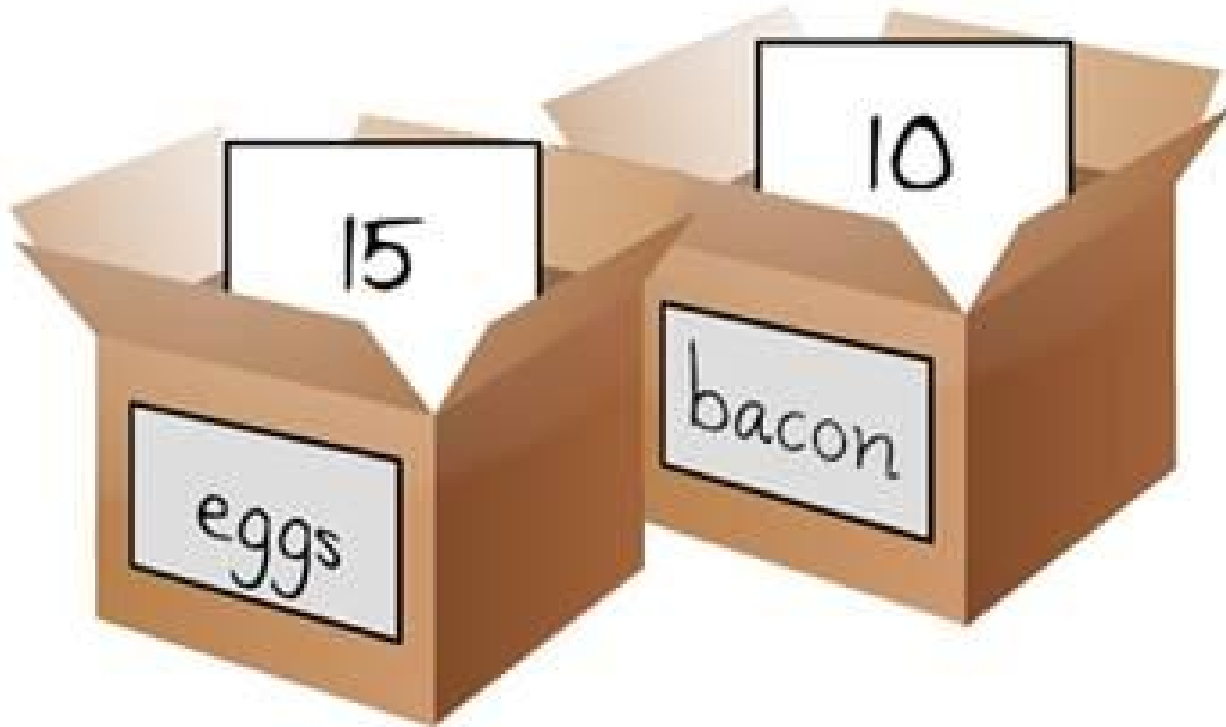


# Outline

1. What is variable?
2. Ruby variables
3. Ruby variable scope
4. Practice
5. Excercise



# 1. What is variable?





## 2. Ruby variables

- ❖ Global variable - \$sign

Available every where within your Ruby script

- ❖ Local variable - sign

It depends on the scope (create new scope with new class, new module, new method)

- ❖ Instance variable - @sign

Available only within a specific object, across all methods in a class instance. Not available directly from class definitions.

- ❖ Class variable - @@sign

Available from the class definition and any sub-classes. Not available from anywhere outside.

- ❖ Constants - Capital Letter



### 3. Ruby variable scope

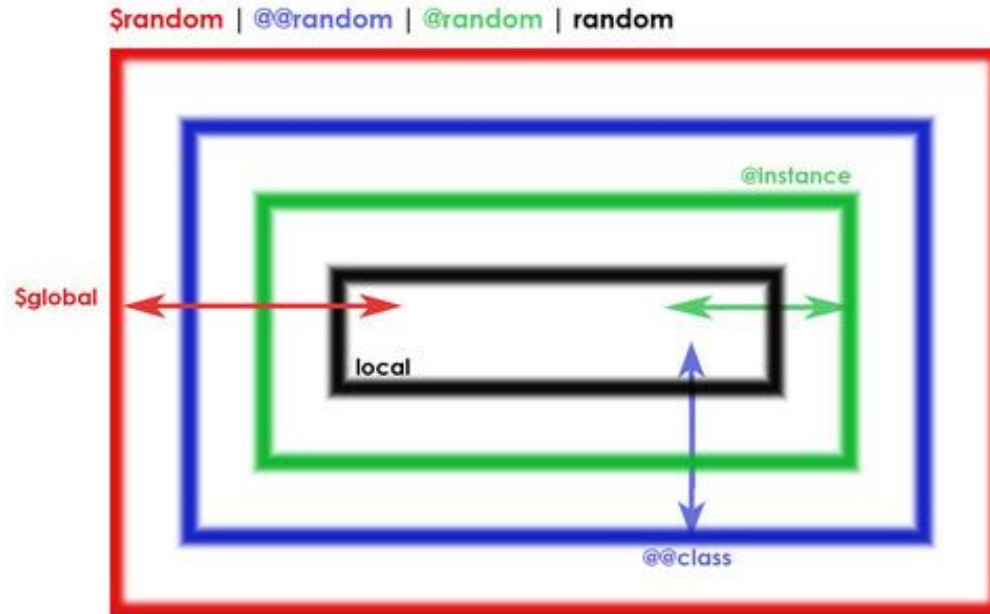


Image reference: <https://www.sitepoint.com/understanding-scope-in-ruby/>



## 4. Practice - Global variables

```
$global = 0
```

```
class C
  puts "in a class: #{$global}"

  def my_method
    puts "in a method: #{$global}"

    $global = $global + 1
    $other_global = 3
  end
end
```

```
C.new.my_method
```

```
puts "at top-level, $global: #{$global}, $other_global: #{$other_global}"
```

in a class: 0

in a method: 0

at top-level, \$global: 1, \$other\_global: 3



## 4. Practice - Local variable

```
color = "Red"
def method1
  color = 192
  puts("Color Value in method1 : ", color)
end

def method2
  color = 255
  puts("Color Value method2: ", color)
end

method1
method2
method1
puts("Color Value outside methods : " + color)
```

```
H:\>ruby abc.rb
Color Value in method1 :
192
Color Value method2:
255
Color Value in method1 :
192
Color Value outside methods : Red
```



## 4. Practice - Instance variable

```
class Student
  def initialize student_id, student_name
    @student_id = student_id
    @student_name = student_name
  end

  def show
    puts "Student Name and ID : "
    puts @student_id, @student_name
  end
end

Student.new(1, "Sara").show
Student.new(2, "Raju").show
```

```
Student Name and ID :
1
Sara
Student Name and ID :
2
Raju
```





## 4. Practice - Class variables

```
class School                                     80
  @@no_off_students = 650                       80
end                                              80
```

```
class V < School
  @@no_off_students = 75
end
```

```
class VI < School
  @@no_off_students = 80
end
```

```
puts School.class_eval("@@no_off_students")
puts V.class_eval("@@no_off_students")
puts VI.class_eval("@@no_off_students")
```



## 5. Exercise

Look at the following programs...

```
x = 0
3.times do
  x += 1
end
puts x
```

```
y = 0
3.times do
  y += 1
  x = y
end
puts x
```

What does x print to the screen in each case? Do they both give errors? Are the errors different? Why?



## 5. Exercise

The first prints 3 to the screen. The second throws an error undefined local variable or method because x is not available as it is created within the scope of the do/end block.



# References

1. <http://ruby-doc.org/>
2. <http://annaershova.github.io/blog/2015/10/26/variable-in-ruby/>
3. [http://rosettacode.org/wiki/99\\_Bottles\\_of\\_Beer#Ruby](http://rosettacode.org/wiki/99_Bottles_of_Beer#Ruby)
4. <https://www.w3resource.com/ruby/ruby-variables-constants.php>



Thank you!