**INSTANCE METHODS**

An instance method is declared inside a class definition and is intended for use by a specific object or ‘instance’ of the class, like this:

class MyClass

def instanceMethod

puts( "This is an instance method" )

end

end

ob = MyClass.new

ob.instanceMethod

### [**Defining Class Methods**](https://github.com/rubocop/ruby-style-guide#def-self-class-methods)

Use def self.method to define class methods. This makes the code easier to refactor since the class name is not repeated.

class TestClass

def TestClass.some\_method

puts "hellpo"

end

end

TestClass.some\_method

class TestClass

def self.some\_method

puts "hellpo"

end

end

TestClass.some\_method

class TestClass

def TestClass.some\_method

puts "testclass"

end

def self.some\_other\_method

pust "someother"

end

class << self

def first\_method

puts "firstmethod"

end

def second\_method\_etc

puts "secondmdthod"

end

end

def first\_method

puts "instacnemethod"

end

end

TestClass.first\_method

c=TestClass.new

c.first\_method

**SINGLETON METHOD**

Singleton methods are methods which are added to a single object and cannot be used by other objects. A singleton method may be defined by appending the method name to the object name followed by a dot or by placing a ‘normal’ method definition inside an << self block like this

class Animal

def self.method

puts "hai"

end

end

c=Animal.new

class << c

def method2

puts "method2"

end

end

c.method2

**Returning multiple values**

def ret\_things

greeting = "Hello world"

a = 1

b = 2.0

return a, b, 3, "four", greeting, 6 \* 10

end

def ret\_hash

return {'a'=>'hello', 'b'=>'goodbye', 'c'=>'fare thee well'}

end

puts ret\_hash

**DEFAULT AND MULTIPLE ARGUMENTS**

def aMethod( a=10, b=20 )

return a, b

end

p( aMethod )

p( aMethod( 1 ))

p( aMethod( 1, 2 ))

def aMethod( a=10, b=20, c=100, \*d )

return a, b, c, d

end

p( aMethod( 1,2,3,4,6 ,9) )

**PARALLEL ASSIGNMENT**

s1, s2, s3 = "Hickory", "Dickory", "Dock"

puts s1,s2,s3

i1 = 1

i2 = 2

i1, i2 = i2, i1

def returnArray( a, b, c )

a = "Hello, " + a

b = "Hi, " + b

c = "Good day, " + c

return a, b, c,"tyfb"

end

x, y, z = returnArray( "Fred", "Bert", "Mary" )

puts x,y,z

num=[1,2,4]

def arr(num)

num[0]=4

end

arr(num)

puts num