Some\_object.Some\_method.Some\_method.Some\_method.etc => output

String Predefined Methods - nothing but string instance\_methods(readymade methods)

0) Concatenating Strings

lang = "Ruby" + " programming" + " language"

lang1 = "Python" " programming" " language"

lang2 = "Perl" << " programming" << " language"

lang3 = "Java".concat(" programming").concat(" language")

2) s = "hello"

s.replace "world"

3) first\_name = "Michael"

last\_name = "Hartl"

first\_name + " " + last\_name # Concatenation, with a space in between

"Michael Hartl"

"#{first\_name} #{last\_name}"

"Michael Hartl"

4) puts "football".empty? # false

puts "".empty? #true

5) puts "hello world".nil?

puts "".nil?

puts nil.nil?

6) abc = "Ruby language"

puts abc[0]

puts abc[-1]

puts abc[0, 3]

puts abc[0..9]

puts abc.length

7)

puts "zetcode".upcase

puts "zetcode".size

puts "zetcode".reverse

puts "NIPUN".downcase

puts "zetcode".capitalize

puts "Hello".swapcase

puts 'man from the united states'.split.map(&:capitalize).join(" ")

8) word = "Determination"

puts "The word #{word} has #{word.size} characters"

puts word.include? "tion"

puts word.include? "tic"

puts word.empty?

word.clear

puts word.empty?

9)names = [ 1, 2 ] << "c" << "d" << [ 3, 4 ]  
 #=>names.inspect

10)

names1 = [ 'ann', 'richard', 'william', 'susan', 'pat' ]

puts names1[0]

puts names1[3]

# this is the same:

names2 = %w{ ann richard william susan pat }

puts names2[0]

puts names2[3]

11)

Multi Line String

address = <<-Doc

S/o M Durga Prasad

H:no: 50-5-4/1

weaver's colony

Hyderabad

Doc

12)

Change String

s = "Ruby rocks"

s["rock"] = "rule"

s #=> "Ruby rules"

13)

split method

"apple banana cherry".split

=> ["apple", "banana", "cherry"]

14)

"Koala".intern #=> :Koala  
s = 'cat'.to\_sym #=> :cat

" hello ".strip #=> "hello"  
" hello ".lstrip #=> "hello "

" hello ".rstrip #=> " hello"

a = "world"  
a.prepend("hello ") #=> "hello world"  
a #=> "hello world"

15)

[ "a", "b", "c" ].join #=> "abc"  
[ "a", "b", "c" ].join("-") #=> "a-b-c"

16)

"abcd".insert(0, 'X') #=> "Xabcd"  
"abcd".insert(3, 'X') #=> "abcXd"

17)

"abcd".class.instance\_methods.sort #type on command prompt which shows instance methods

Some Important List of String methods

concat, downcase , gsub, include?, replace, inspect, intern/to\_sys, strip, split, sub, upcase, size/length, empty?, capitalize, to\_i, clear, crypt

Arrays:

first, last, length, count, empty?, include?, push, unshift, insert, pop, shift, delete, delete\_at, compact, uniq, fetch, flatten, shuffle, find, find\_all, first(2), max, min, none?, sort, sort\_by, take(3), join, slice

<http://ruby-doc.org/core-2.2.2/Array.html>

Enumerable : Iterator

Collect, select, each, map, find, find\_all, first(2), max, min, minmax, none?, sort, sort\_by, take(3), count,

1)

integers = [1, 2, 3, 4, 5]

animals = %w( donkey dog cat dolphin eagle )

weights = []

weights << 4.55 << 3.22 << 3.55 << 8.55 << 3.23

puts integers.inspect

puts animals.inspect

puts weights.inspect

2)

clues = %w(vitamins minerals chocolates)

puts clues

3)

# Sum some numbers

(5..10).reduce(:+) #=> 45

# Same using a block and inject

(5..10).inject {|sum, n| sum + n } #=> 45

4) a = [ 1, 2, [3, [4, 5] ] ]  
a.flatten! #=> [1, 2, 3, 4, 5]

Hash predefined methods: keys, values, values\_at,

h = { "cat" => "feline", "dog" => "canine", "cow" => "bovine" }  
h.values\_at("cow", "cat") #=> ["bovine", "feline"]

Convert from One Object to another Object: to\_s, to\_sym, to\_a, to\_i, to\_f

Array to Hash

a = ["item 1", "item 2", "item 3", "item 4"]  
h = Hash[\*a]

{"item 1"=>"item 2", "item 3"=>"item 4"}

#partition

(1..6).partition { |v| v.even? } #=> [[2, 4, 6], [1, 3, 5]]

(1..10).reject { |i| i % 3 == 0 } #=> [1, 2, 4, 5, 7, 8, 10]  
[1, 2, 3, 4, 5].reject { |num| num.even? } #=> [1, 3, 5]

Other Instance methods:

1. "42".to\_f # => 42.0  
2. "foo".[chars](http://apidock.com/ruby/String/chars).to\_a #=> ["f","o","o"]

3. "Xnkl".chop

4. "srinivas".crypt("srinivas")

5. "hello".[each\_char](http://apidock.com/ruby/String/each_char) {|c| print c, ' ' }

6. "hello".[index](http://apidock.com/ruby/String/index)('e')

7. "hello".[partition](http://apidock.com/ruby/String/partition)("l")

8.

Powered By: srinivas