Michelle McGrath

50 Ruby Methods

Examples:

String#rstrip

“Removes any whitespace found on the right side of the string”

Example:

my\_string\_with\_white\_space.rstrip

Array#reject

“Removes any objects from an array that do not match a given criteria”

Example:

states.reject{|s|s.land\_locked?}

**Arrays:**

#count

“To query the an array about the number of elements (a specific element to count can be put in parenthesis) it contains”

Example:

array\_to\_count\_items.count

array\_to\_count\_2\_items.count(2)

array\_to\_count\_even\_items{|x|x%2==0}

#include?

“Returns “true” if the object is present in self (if any object ==object) and otherwise it returns “false””

Example:

array\_test\_with\_letters\_in\_it.include?(“a”)

#select

“Returns a new array containing all of the elements of the original array for which the given block returns “true””

Example:

array\_test\_with\_numbers\_in\_it.select{|num|num.even?}

#each

“Calls the given block once for each element in the array, passing that element as a parameter”

Example:

array\_with\_letters\_in\_it.each{|x| print x, “—“}

#reverse

“Returns a new array containing the original array elements, but in reverse order”

Example:

array\_containing\_list\_of\_letters.reverse

#uniq

“Returns a new array that contains the elements of the original array, but it does not include any repeated items”

Example:

array\_with\_duplicates.uniq

array\_with\_duplicates\_in\_the\_first\_position.uniq{|s| s.first}

#clear

“Removes all elements from the array”

Example:

array\_with\_info.clear

#sort

“Creates a new array that is sorted from an initial array”

Example:

array\_with\_letters\_to\_sort\_alphabetically.sort

array\_with\_letters\_to\_sort\_reverse\_alphabetically.sort{|x, y| y <=> x}

#join

“Returns a string that is created from the elements of an array that are separated by the given separator (and defaults to an empty string to separate if nothing is declared)

Example:

array\_with\_letters.join(“-“)

array\_with\_letters.join

[“a”, “b”, “c”].join

#find\_index

“Returns the index (place) of the first object where the object is equal to the object we are looking for (true) and returns nil if nothing matches the object we are looking for”

Example:

array\_with\_abc\_letters.index(“b”)

array\_with\_abc\_letters.index{|x|x==”b”}

#empty?

“Returns “true” if the array is empty and returns “false” if it is not empty”

Example:

array\_that\_is\_empty.empty?

[].empty?

#first

“Returns either the first element or the first “n” elements, as specified, and returns nil if it is empty and did not declare “n” or returns empty array if original array is empty and did declare “n””

Example:

array\_with\_3\_elements.first

array\_with\_3\_elements\_and\_want\_first\_two.first(2)

#delete

“Deletes all objects from an array that are equal to the declared object and if the optional block is included, if an item is not found instead of returning nil, it returns the text that is in the block”

Example:

array\_with\_letters.delete(“a”)

array\_with\_letters.delete(“f”) {“not found”}

#last

“Returns the last elements of the array and if “n” is included, it returns the last “n” elements of the array”

Example:

array\_with\_letters.last

array\_with\_letters\_and\_want\_last\_and\_second\_to\_last.last(2)

#flatten

“Returns an array that is one-dimensional, for instance, if there is an array inside of an array, it will take all of the individual elements, extract them, and put them individually into a new array. If a level is declared, then it only flattens that many times down (for example if level =1 it will only flatten an array inside of the original array, but will not flatten anything inside of that)”

Example:

array\_with\_array\_inside\_of\_it\_and\_that\_array\_has\_an\_array\_inside\_of\_it.flatten

array\_with\_array\_inside\_of\_it\_and\_that\_array\_has\_an\_array\_inside\_of\_it\_and\_onl\_want\_the\_first\_array\_inside\_to\_be\_flattened.flatten(1)

#sort\_by

“Sorts each item in the array by the contents that are in the block / that the block is asking it”

array\_to\_sort\_length.sort\_by {|i| i.length}

array\_to\_sort\_reverse\_number\_order.sort\_by {|i| -i}

#sample

“Chooses either a random element or random “n” elements from the array”

Example:

array\_with\_numbers.sample

array\_with\_numbers.sample(3)

array\_with\_numbers.sample(random: random\_number\_generator)

array\_with\_numbers.sample(3, random: random\_number\_generator)

**Strings:**

#strip

“Returns a copy of the string, but without any leading or trailing whitespace”

Example:

My\_string\_in\_quotes\_with\_white\_space\_before\_and\_after\_a\_word.strip

#downcase

“Returns a copy of the string with all upper case letters as lower case letters; only works for A – Z”

Example:

My\_string\_in\_quotes\_with\_upper\_case\_letters\_throughout.downcase

#empty?

“Returns “true” if the string has length = 0, returns “false” if string has length >0”

Example:

String\_in\_quotes\_with\_characters\_so\_length\_is\_greater\_than\_0.empty?

#length

“Returns the length of the string in number of characters”

Example:

String\_with\_length\_to\_test.length

#each\_line

“Splits the string using the supplied parameter as the record separator, passing each substring into the block; calls each line from the file and then you can loop through it”

Example:

String\_with\_letter\_a\_included\_multiple\_times.each\_line(‘a’) {|s| p s}

#capitalize

“Returns a copy of the string with the first letter capitalized and the rest of the letters lowercase (if the first characters in the string are numbers, then the following characters are all lowercase if they are letters)”

Example:

String\_to\_capitalize\_first\_letter\_and\_rest\_are\_lowercase.capitalize

#strip

“Returns a copy of the string, but without any leading or trailing whitespace”

Example:

My\_string\_in\_quotes\_with\_white\_space\_before\_and\_after\_a\_word.strip

#upcase

“Returns a copy of the string with all lowercase letters returned as uppercase letters; only works on a – z”

Example:

String\_containing\_lowercase\_letters.upcase

#split

“Divides a string into substrings based on a delimiter, and these are returned in an array; and the limit will make it so that at most that number of entries in the array are returned”

Example:

String\_with\_list\_of\_5\_names\_separated\_by\_comas.split(‘,’)

String\_with\_list\_of\_5\_names\_separated\_by\_comas.split(‘,’, 3)

Do “” vs ‘’ matter?? -> no!

**Integers:**

#to\_s

“Returns the number as a string (puts the number in quotes); makes the item a string; often used when have a variable you think is a string, so you do this to make sure it is a string”

Example:

Integer\_to\_make\_into\_a\_string.to\_s

#downto

“It iterates the given block from the given block down to and including the limit that is given”

Example:

Integer\_to\_start.downto(integer\_to\_end) {|n| print n, “… ”}

#upto

“It iterates the given block from the given block up to and including the limit that is given”

Example:

Integer\_to\_start.upto(integer\_to\_end) {|n| print n, “ “}

#even?

“Returns “true” if the integer is an even number, or else returns “false””

Example:

Integer\_to\_check\_if\_even.even?

#odd?

“Returns “true” if the integer is an odd number, or else returns “true””

Example:

Integer\_to\_check\_if\_odd.odd?

#round

“Rounds the integer to the given number of decimal places, and if none are specified, rounds to 0 decimal places/closes whole number; if decimal places is negative then it will round negative decimal places (ie -1, rounds to the nearest 10s place)”

Example:

Integer\_to\_round\_to\_1\_decimal\_place.round(1)

**Time:**

#now

“Creates a new time object for the current time in the format: yyyy-mm-dd hr:min:sec +0000”

Example:

Time.now

#year

“Returns the year for the time, including the century”

Example:

Time\_to\_check.year

#month/mon

“Returns the month of the year for the time to check, in its numeric value”

Example:

Time\_to\_check.mon

Time\_to\_check.month

#wday

“Returns a number that represents the weekday for a given time, with Sunday=0 to Saturday=6”

Example:

Time\_to\_check.wday

Weekday? Will check if the time is true or false for that weekday ex) time.wednesday?

#yday

“Returns an integer representing the day of the year from 1 to 366”

Example:

Time\_to\_check.yday

#strftime

“Formats time according to the directives in the given string format, and these begin with %character”

Example:

Time\_to\_format\_with\_word\_at\_and\_date\_in\_format.strftime(“at %m/%d/%Y %I:%M%p”)

~~Do we need to know what all of these abbreviations mean???~~

**Hashes:**

#to\_a

“Converts a hash to a nested array of [key,value] arrays”

Example:

Hash\_with\_multiple\_values\_with\_rockets\_assigning\_them\_to\_keys\_that\_you\_want\_to\_make\_into\_an\_array\_that\_are\_separated\_by\_comas.to\_a

#each\_key

“Calls the block for each key in the hash and passes the key as the parameter”

Example:

Hash\_with\_keys\_and\_values.each\_key{|key| puts key}

#keys

“Returns a new array that is populated with all of the keys from the given hash”

Example:

Hash\_with\_lots\_of\_keys\_to\_put\_in\_an\_array.keys

#has\_key?

“Returns “true” if the given key is present in the hash, or else returns “false””

Example:

Hash\_from\_which\_to\_test\_if\_”a”\_key\_is\_present.has\_key?(“a”)

#has\_value?

“Returns “true” if the given value is present in the hash, or else returns “false””

Example:

Hash\_with\_value\_100\_included.has\_value?(100)

#fetch

“Returns the value from the given key and will say KeyError if no default is given and the key cannot be found”

Example:

Hash\_to\_check\_value\_of\_key.fetch(“a”)

Hash\_to\_check\_value\_of\_key.fetch(“a”, “key not found”)

Hash\_to\_check\_value\_of\_key.fetch(“a”) {|m| “key not found, #{m}”}

#merge

“Returns a hash with the values of the original hash and the other hash, and if no block is specified then duplicates from the original has will be overridden with those of the other hash, rather than following the directions given in the block”

Example:

Original\_hash.merge(other\_hash)

Original\_hash.merge(other\_hash) {|key, oldval, newval | newval-oldval}

**Objects:**

#inspect

“Returns a string containing a human-readable representation of the object; the default shows the object’s class name, encoding of object id, and list of the instance variables and their values”

Example:

Array\_to\_inspect.inspect

Time.new.inspect

#instance\_of?

“Returns “true” if the object is an instance of the given class, else return “false””

Example:

Object\_like\_a\_specific\_array\_to\_check.instance\_of?(class\_like\_array\_to\_check)

#kind\_of?

“Returns “true” if the class specified is the class of the object, or if the class specified is one of the superclasses of the object, or if the modules included the object, else returns “false”; can call on a class itself and not only an instance of a class”

Example:

Object\_to\_check.kind\_of?(class\_to\_check\_against)

#nil?

“Responds “true” if the object is nil, else responds “false””

Example:

Object\_to\_check.nil?

#respond\_to?

“Returns “true” if the object responds to a given method, else returns “false”; private and protected methods are included in the search if the optional second parameter evaluates to true”

Example:

Object\_to\_check\_like\_array.respond\_to?(method\_to\_check\_like\_sort, include\_all=false)