String methods

"hello"

`.reverse`	`=> "olleh"`	reverses the string
`.upcase`	`=> "HELLO"`	makes a new string that is all uppercase
`.length`	`=> 5`	counts number of characters in the string
`.empty?`	`=> false`	`true` if string is empty, `false` if there's something there
`.index("e")`	`=> 1`	if the argument exists in the string, returns the position of the first occurrence (counting starts from 0), otherwise `nil`
`.count("l")`	`=> 2`	counts the occurrences of the argument
`+ "!"`	`=> "hello!"`	makes a new string from the two strings
`.concat("!")`	`=> "hello!"`	alters the original string by adding the argument
`.chars`	`=> ["h", "e", "l", "l", "o"]`	converts to an array, one character per element
`"hi\n"``.strip`	`=> "hi"`	removes spaces and newlines from the start and end

Array methods

["yellow", "green", "blue"]

`.length`	`=> 3`	counts the number of elements in the array
`.member?("red")`	`=> false`	`true` if argument is in the array, `false` otherwise
`.at(1)`	`=> "green"`	returns a specified element (counting starts at 0)
`[1]`	`=> "green"`	returns a specified element (counting starts at 0)
`.delete("blue")`	`=> "blue"`	deletes the argument from the existing array
`.sample`	`=> "green"`	gives you a randomly chosen element from the array
`+ ["red"]`	`=> ["yellow", "green", "blue", "red"]`	makes a new array from the two arrays
`.push("red")`	`=> ["yellow", "green", "blue", "red"]`	pushes the argument on to the existing array
`.sort`	`=> ["blue", "green", "yellow"]`	sorts the array's elements alphabetically
`.join("-")`	`=> "yellow-green-blue"`	converts array to string using argument to join elements

Other methods

`"hello" == "hello"`	`=> true`	compares two objects, returns `true` if they are the same
`"hello" != "hello"`	`=> false`	compares two objects, returns `true` if they are different
`1 < 5`	`=> true`	compares two objects, returns `true` if 1st is less than 2nd
`Kernel.puts("hello")`	`=> nil`	prints "hello" to the screen followed by a newline
`Kernel.gets`	`=> "hello!\n"`	gets a line of input from the keyboard
`Random.rand(10)`	`=> 4`	returns a random number between 0 and 9
`File.read("words.txt")`	`=> "abacus\n..."`	reads the entire file into a string

Iterator methods

cool_things = ["ruby", "rails", "iteration"]

`cool_things.each do |thing|`
`  Kernel.puts(thing + " is awesome!")`
`end`
	`ruby is awesome!`
	`rails is awesome!`
	`iteration is awesome!`
	`=> ["ruby", "rails", "iteration"]`

`cool_things.map do |thing|`
`  thing.upcase`
`end`
	`=> ["RUBY", "RAILS", "ITERATION"]`

`cool_things.select do |thing|`
`  thing.index("ra") != nil`
`end`
	`=> ["rails", "iteration"]`

`cool_things.detect do |thing|`
`  thing.index("ra") != nil`
`end`
	`=> "rails"`

`3.times do`
`  Kernel.puts("hello!")`
`end`
	`hello!`
	`hello!`
	`hello!`
	`=> 3`

`Kernel.loop do`
`  Kernel.puts("endless loop!")`
`end`
	`endless loop!`
	`endless loop!`
	`...`

Flow control

```
Kernel.loop do
  Kernel.print("Guess the word to win: ")
  word = Kernel.gets.strip
  if word == "cats"
    Kernel.puts("You win!")
    break
  else
    Kernel.puts("Sorry, that's not it.")
  end
end
```

Resources

For full reference documentation of all of Ruby's classes, visit http://ruby-doc.org/core-2.0.0. Googling "ruby array" or "ruby string" will take you to this site with heaps of information about these classes.

If you're the kind of person that learns from books, the "pickaxe" is the bible of the Ruby language, available from http://pragprog.com/book/ruby4/programming-ruby-1-9-2-0 or various other places.