Enumerable...

(of a set) having a finite number of elements;
able to be counted.

"You can iterate over many different types of objects in Ruby using the same set of methods. For example, you can use the **includes?** method to iterate over Arrays and Hashes to see if they include a specific object, and similarly use the **map** method to transform them. You might think each method implements these methods themselves, **but you'd be wrong.**

In Ruby, these types of methods are implemented in the **Enumerable** module. If your class implements the **each** method and **includes** the **Enumerable** module, your class will respond to a whole host of messages involving collections, including iterating, mapping, searching, etc."

Using the Enumerable Module
By Michael Morin, About.com Guide

Enumerable has many helpful methods!
Here are some we will explore today:

- all?
- any?
- collect
- count
- cycle
- find_all
- select
- reduce
- sort
- sort!
- to_a
- zip

Follow along in irb

- Open your terminal and type "irb"
- Now you can follow these examples live
- Stop me if you have questions

.sort & .sort!

- trees = ["maple", "elm", "walnut", "oak", "dogwood", "pine", "sweet-gum"]
- trees.sort = ["dogwood", "elm", "maple", "oak", "pine", "sweet-gum", "walnut"]
- Add a ! (bang) to make this indefinite

.collect & .collect!

- trees = ["maple", "elm", "walnut", "oak", "dogwood", "pine", "sweet-gum"]
- trees.collect {|tree| tree.capitalize} = ["Maple", "Elm", "Walnut", "Oak", "Dogwood", "Pine", "Sweet-gum"]
- Add a ! (bang) to make this indefinite

.count

- trees = ["maple", "elm", "walnut", "oak", "dogwood", "pine", "sweet-gum"]
- trees.count = 7
- trees.count("elm") = 1
- nums = [1, 2, 4, 8, 9]
- nums.count {|n| n%3 == 0}

all? & any?

- trees = ["maple", "elm", "walnut", "oak", "dogwood", "pine", "sweet-gum"]
- trees.all? {|tree| tree.length >= 4} = false
- trees.all? {|tree| tree.length >= 3} = true
- trees.any? {|tree| tree.length >= 6} = true

.cycle

- trees = ["maple", "elm", "walnut", "oak", "dogwood", "pine", "sweet-gum"]
- trees.cycle {|tree| print tree}
- trees.cycle(3) {|tree| puts tree}

.reduce

- nums = [45, 72, 12, 109]
- nums.reduce(:+) = 238
- nums.reduce(2, :*) = 8475840

.find_all & .select

- nums = [1, 2, 4, 6, 7, 9, 12, 15]
- nums.find_all {|n| n%3 == 0} = [6, 9, 12, 15]
- nums.select {|n| n.even?} = [2, 4, 6, 12]

.to_a

- { 'a' => 1, 'b' => 2, 'c' => 3 }.to_a
  = [["a", 1], ["b", 2], ["c", 3]]

- (1..7).to_a = [1, 2, 3, 4, 5, 6, 7]

.zip

- a = [4, 5, 6]
- b = [7, 8, 9]
- [1, 2, 3].zip(a, b)
  = [[1, 4, 7], [2, 5, 8], [3, 6, 9]]
- [1, 2].zip(a, b) = [[1, 4, 7], [2, 5, 8]]

Challenge

- letters = ["a", "t", "e", "c", "s", "d", "h", "o", "r", "d", "e"]
- Make this into a string that says: "Read the docs!"