



Ruby 101

Ruby Basics: Iterators

Iterators:

Operate on:

each member of a data collection

```
first_five_integers = [1,2,3,4,5]
```

Run specific code on each member:

```
puts ...
```



How would I do this?

```
puts first_five_integers[0]  
puts first_five_integers[1]  
...
```

```
first_five_integers.each do |my_integer|  
  puts my_integer  
end
```

Ruby Basics:



'each' iterator:

- a method called on a collection object

```
first_five_integers.each
```

- takes a program block as an argument

```
first_five_integers.each do  
  ...  
end
```



'each' iterator: continued

- supplies a variable to the block
- variable is a member of the data collection

```
first_five_integers.each do |my_integer|  
  puts my_integer  
end
```



Ruby Basics:

Iterators

Why useful?

act as program loops for collections

DRY

Do not Repeat Yourself



'each' with a hash

```
my_hash = {"orange" => true, "banana" => false}
```

```
my_hash.each do | key, value |
```

```
  ...
```

```
end
```



Try it yourself:

Write a program that asks the user whether they need a specific grocery item or not, then print out the grocery list at the end of the program. Use iterators so that code is not repeated unnecessarily.

Ruby Basics:



Ruby Basics:

```
grocery_items = {"oranges" => false, "bananas" => false}
puts "Do you need:"
grocery_items.each do | item, need_for_item |
  puts item + "? (Y/N)"
  case gets
    when "Y\n"
      grocery_items[item] = true
    when "N\n"
      grocery_items[item] = false
  end
end
puts "Here's your list:"
grocery_items.each do | item, need_for_item |
  puts item if need_for_item
end
```

