Collections, Loops, and APIs

Week 2 / Lesson 2

Agenda

- Recap
    - Collections, Hashes, and Symbols
    - Your and my worst nightmare - FizzBuzz
- Practical Application Of Collections & Loops (APIs)
- Lab Time

Collections

Hashes

- AKA dictionaries
- Each entry has a key and a value
- Accessing a hash’s key returns its value (Like indexes in arrays)
- Useful for holding related data (User records, database results, etc.)

Hashes

Find by key

```
ga_markets = {"NYC" => "New York
City", "LA" => "Los Angeles", "SYD"
=> "Sydney", "LDN" => "London"}
```

```
ga_markets["NYC"]
ga_markets["LA"]
ga_markets[“SYD”]
```

Hashes

Setting Values

```
user_hash = {}
user_hash["name"] = "Salman"
user_hash["favorite_color"] = "Green"
user_hash
```

```
>> {"name"=>"Salman",
"favorite_color"=>"Green"}
```

Symbols

New Ruby type

- A symbol is a special type of object in ruby, used extensively
- It is always preceded by a colon
- Cannot contain **spaces** ~~or numbers~~
- Used a lot in configuration files and in places where you'd find strings in other languages
- Symbols are used because:
    - they are immutable and **take less memory**
    - they are easier to compare to other objects
    - they are cleaner in syntax
- Examples:
    - `:hello`
    - `:this_is_a_symbol`

Symbols

Primarily used as keys for hashes

```
ga_markets = {}
ga_markets = {:NYC => "New York City"}
ga_markets[:LA] = "Los Angeles"
ga_markets

>> {:NYC => "New York City", :LA => "Los Angeles"}
```

Hash

Methods

```
user = {:user_name => "SalmanAnsari",
        :email => "salman.ansari@gmail.com"}
user.has_key? :email #true
user.key? :email #true
user.include? :email #true
user.has_value? "SalmanAnsari" #true
(note: extremely inefficient!)
```

Hash

Ruby 1.9+ Alternate Syntax

```
user = {:user_name => "SalmanAnsari", :email => 
"salman.ansari@gmail.com"}
# becomes
user = {user: "SalmanAnsari", email: "salman.ansari@gmail.com"}
# a little bit more concise
# more closely matches JSON format
# considered an 'alternate' syntax, not a replacement
```

Collections

Array of Hashes

```
users = [
    {:user => "Salman Ansari", :role => "Instructor"},
    {:user => "Brooks Swinnerton", :role=> "TA"}
    {:user => "Brian Fountain", :role => "TA"]
]

# Alternate syntax for Ruby 1.9+

users = [
    {user: "Salman Ansari", role: "Instructor"},
    {user: "Brooks Swinnerton", role: "TA"},
    {user: "Brian Fountain", role: "TA"}
]
```

Iterating over Collections

.each

```
ga_markets = ["NYC", "LA", "SYD", "LDN"]
```

```
ga_markets.each {|market| puts market}
```

Let me show hash iteration

See Week2/Lesson1/instructor_notes/hashes.rb to follow along

Lab Time

Collections

See Week2/Lesson1/Examples/hashes.rb

Recap

Iterating Over Collections

Question time?

Fizz. Buzz.

FizzBuzz.

TIME : 30 min

- Write a program that prints numbers from 1 to 100.
- But for multiples of three print “Fizz” instead of the number and for the multiples of five print “Buzz”.
- For numbers which are multiples of both three and five print “FizzBuzz”.

Recap

- Variables
- Conditionals
- Iteration
- Collections

Working with Collections

Let’s get practical…

APIs

Application Programming Interface

APIs

Working with APIs

1 Ask for some data

2 Convert it into a format we can work with

3 Do cool stuff

APIs

Ask for Data

- We need to use a gem to ask for data.

```
gem install rest-client
```

API

Convert into a format we can work with

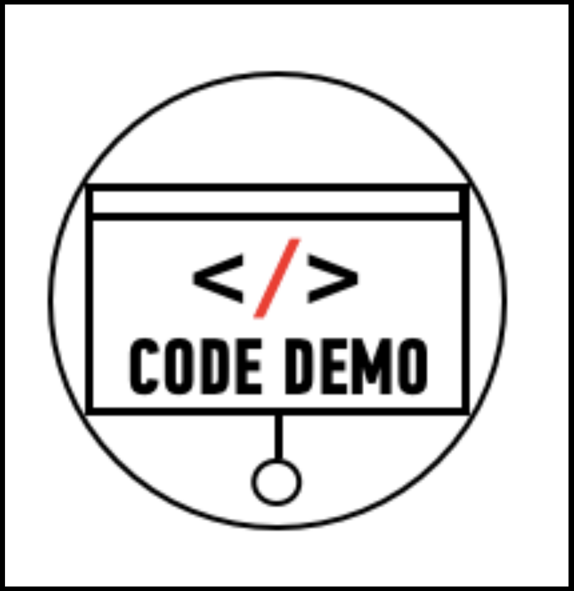
- JSON - JavaScript Object Notation

```
gem install json
#
'{"NY":"New York","LA":"Los Angeles","SYD":"Sydney","LDN":"London"}'
```

Do Cool Stuff

Getting Stories

I will get the latest stories from Mashable and output some of the data I get back from their API.



~~Teddit API~~ News API

Homework

- Continue Reviewing and Practicing
- Finish today’s lab
- All due next Thursday

Resources: Collections, Loops & APIs

Cheet Sheet

Array of Hashes

```
    super_heros =[{ hero: 'Batman', secret_identity: 
'Bruce Wayne'},
    {hero: 'Superman', secret_identity: 'Clark Kent'},
    {hero: 'Spiderman', secret_identity: 'Peter 
Parker'}]
```

.each & .map

```
 super_heros.map do |super_hero|
        puts "#{super_hero[:hero]} is 
#{super_hero[:secret_identity]}"
    end
```

```
  super_heros.each do |super_hero|
        puts "#{super_hero[:hero]} is 
#{super_hero[:secret_identity]}"
    end
```

- .each returns the original array.
- Map returns a new array with the results you added.

Tips, Tricks & Motivation

See this lesson’s readme file for helpful links and more information