Ruby Methods and Hashes Recap

Devpoint Labs - Jake Sorce / Dave Jungst

What is a method?

- breaks our code into manageable chunks
- should perform a single task
- should have a concise name
- should perform an action or return an object
- always returns last line executed
- if you have to use and / or to describe what your method does you may have 2 methods

What is a hash?

- A hash is a ‘key’, ‘value’ pair
  - Contained in { }
  - keys can be symbols, strings, ints, …, objects.
    - hash = { first_name: ‘Dave’, last_name: ‘Jungst’ }
    - hash = { 1 => ‘a’, 2 => ‘b’, 3 => ‘c’ }
    - hash = { ‘first_name’ => ‘Dave’, ‘last_name’ => ‘Jungst’ }

Ways to define keys and values of a hash

- With Symbols
  - hash = {first_name: ‘jake’, last_name: ‘sorce’}
  - hash[:first_name]
    - returns ‘jake’
- With Strings
  - hash = {‘first_name’ => ‘jake’, last_name => ‘sorce’}
    - hash[‘last_name’]
      - returns ‘sorce’
- With Ints
  - hash = {1 => ‘one’, 2 => ‘two’, 3 => ‘three’}
    - hash[2]
      - returns ‘two’

Ways to define a method

- With required params

```
def my_method(message)
	puts message
end
```

- With default params

```
def my_method(message, message_2 = ‘my message’)
	puts “#{message} #{message2}”
end
```

Method signature

```
def add_one
  puts 1 + 1
end


def add_one(number)
  puts 1 + number
end


def add_one(number, offset)
  puts number - offset + 1
end
```

```
add_one
     => 2

add_one(12)
     => 13

add_one(5, 2)
     => 4
```