

Chapter 4

4.1 Motivation

Start by creating a branch off master to work

```
$ git checkout -b ruby-flavored-rails
```

Helpers

in views (app/helpers/*.rb)

we can use it to simplify our code

changing

```
<title><%= yield(:title) %> | Ruby on Rails Tutorial Sample
```

```
App</title>
```

to

```
<title><%= full_title(yield(:title)) %></title>
```

4.2 Strings and methods

On cloud you can use nano (a text editor) to configure

```
$ nano ~/.irbrc
```

To insert in our case

```
IRB.conf[:PROMPT_MODE] = :SIMPLE
```

```
IRB.conf[:AUTO_INDENT_MODE] = false
```

then save and exit (Ctrl-X)

string manipulation

in our rails console

```
"" (empty string vs non empty string) "hola"
```

Concatenation using +

Or concatenate using #{var}

```
"Andy" + "AE"
```

Same as

```
#{fn} #{ln} (once fn defined as Andy and ln defined as AE)
```

Printing

Use puts or print and get a returned value of nil after it prints

Other commands

.length, .empty? .nil?

4.3 Other data structures

Arrays

.split('character splitting on')

creates an array with each string that is separated by the char

other commands empty? Include()? Sort, reverse and shuffle

all print but no change to initial array

mutable arrays

mutate with a bang (!)

or by push

Blocks

Ruby's anonymous functions

Applies function for each in a range

```
>> (1..5).each do |i|
```

```
?> puts 2 * i
```

```
>> end
```

Hashes

Our maps

```
>> user = { "fn" => "Andy", "ln" => "AE" }
```

Symbols

“Symbols look kind of like strings, but prefixed with a colon instead of surrounded by quotes... symbols are basically strings without all the extra baggage”

4.4 Classes

We should understand these and let them be

