Having Fun with Strings

Paolo Perrotta

FREELANCE DEVELOPER

@nusco

Search

Ruby-Doc.org

Help and documentation for the Ruby programming language.

Core API

These are the API documents for base classes and modules in the current stable release of Ruby 2.7.2

- 2.7.2 core - Core API docs for Ruby 2.7.2 This is the current official release.
- 2.7.1 core - Core API docs for Ruby 2.7.1
- 2.7.0 core - Core API docs for Ruby 2.7.0
- more...

Standard Library API

These are the API documents for the standard library classes and modules in version 2.7.2

- The 2.7.2 standard library (this is the current official release)
  Additional libraries included with the standard Ruby distribution, such as CGI, OpenURI, and REXML
- The 2.7.1 standard library
- more...

Ruby 3.0.0

- Ruby 3.0 resources
  - Ruby 3.0.0 Preview 1 Released Overview of some new features in Ruby 3.0.0.
  - 3.0.0 preview1 core - Core API docs for Ruby 3.0.0 preview1 base classes and modules, such as String, Array, Symbol, etc.
  - The 3.0.0 preview1 standard library

Getting Started

A collection of resources for those just starting out with

Docs in Other Languages

French / Français

Bang Methods and Question Methods

Weird Method Names

Bang methods

Methods that end in *!* have side effects–for example, they change their object

Question methods

Methods that end in *?* are boolean–they return either *true* or *false*

Cleaning Up the Text

Unleashing Regular Expressions

"Some people, when confronted with a problem, think "I know, I'll use regular expressions." Now they have two problems."

Jamie Zawinski

Splitting the Text

Have a doubt?
Ask the REPL.

A Little Extra: Single and Double Quotes

Two Ways to Define Strings

With single quotes

They're taken literally as the exact sequence of characters that you typed

With double quotes

They're processed to respect escape sequences and string interpolation

What We Did

word_counter.rb ×

Users > nusco > Documents > Work > Content > Screencasts > Pluralsight > trainings > ruby getting started > production > m04 > code04 > word_counter.rb

```ruby
TEXT_FILE = "romeo-juliet.txt"

words = File.read(TEXT_FILE).downcase.gsub(/[^a-z]/, " ").split
```