

# Rubocop

Colin Dean -> @colindean  
Lightning Talks!

# Notices.

My words are not necessarily my employers'. Duh.

Tweet with [#rubocop](#) and it'll display up here.

# Introduction

Rubocop is a Ruby static code analyzer, based on the community Ruby style guide.

# Rubocop finds style errors.

139 Styles, 36 Lints, 8 Rails-specific, 7 Metrics

- ✓ Line length
- ✓ Unless With Else
- ✓ Trailing characters
- ✓ Hash syntax

# Run it.

```
gem install rubocop
```

```
rubocop # warning: hard mode
```

# Read it.

Inspecting 2924 files  
WCCC.C.C.W

Offences:

Library/brew.rb:4:17: C: StringLiterals: Prefer single-quoted strings when you don't need string interpolation or special symbols.  
std\_trap = trap("INT") { exit! 130 } # no backtrace thanks  
          ^^^^

Library/brew.rb:15:1: C: SpecialGlobalVars: Prefer \$LOAD\_PATH over \$:.  
\$.unshift(HOMEBREW\_LIBRARY\_PATH + '/vendor')  
^^

# Checks = “cops”

You can write them yourself!

# Automatically fix some offenses

```
ibm-registration.rb:8:13: C: [Corrected] Use the new Ruby 1.9 hash syntax.  
  uninstall :pkgutil => "com.ibm.IBM-Registration"  
      ^^^^^^^^^^^^^^^
```

9 files inspected, 31 offenses detected, 12 offenses corrected



# Output formats

Regular output is good enough for regular, daily use.

JSON for analytics. Dashboards.

# Offenses List

rubocop -f offenses

15664 StringLiterals

3805 LineLength

3039 Documentation

2363 AlignParameters

**taken from Homebrew rev cfc7042**

# Many more

`-f, --format FORMATTER` Choose an output formatter. This option can be specified multiple times to enable multiple formatters at the same time.

`[p]rogress` (default)

`[s]imple`

`[c]lang`

`[e]macs`

`[j]son`

`[f]iles`

`[o]ffences`

# Automatically generated configuration

- ✓ Analyzes and builds config from least number of violations to greatest
- ✓ Disables everything it finds
- ✓ Uses sane defaults for others

# TODO workflow

- ✓ Re-enable one cop
- ✓ Fix
- ✓ Commit
- ✓ Repeat

# Guard

```
group :development do  
  gem 'guard-rubocop'  
end
```

```
guard init rubocop
```

# Guardfile

```
guard :rubocop do
  watch(/.+\.rb$/)
  watch(/(?:.+V)?\.rubocop\.yaml$/) { |m| File.dirname(m[0]) }
end
```

(guard init rubocop does this for you!s)

# Rake

```
require 'rubocop/rake_task'
```

```
RuboCop::RakeTask.new
```



# New sheriff in town

- ✓ Cleanup is hard
- ✓ Focus on one area

# Cui bono?

- ✓ Give new contributors a TODO list guide
- ✓ TODO list is in logical order of least work to most work.

Small wins to big, low-hanging fruit to orchards.

**What's the  
benefit of  
this effort?**

# At the bark level,

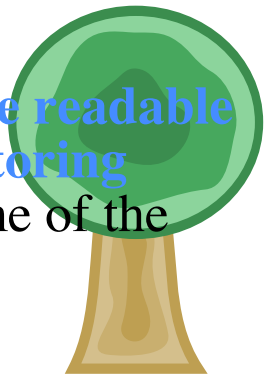
Little style violations are removed.

This isn't much more than bragging rights, but it's something to feel good about.



# At the tree level,

Code may become **more readable**  
because of trivial **refactoring**  
necessary to satisfy some of the  
easier validations.



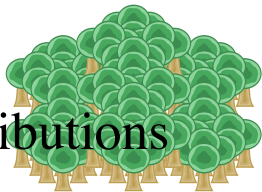
# At the grove level,

Code may **execute more efficiently** because of **refactoring** necessary to satisfy some of the harder validations.



# At the forest level,

- ✓ higher quality code
- ✓ guide for better contributions



# **This applies to all style checkers!**

- ✓ **ScalaCheck**
- ✓ **CheckStyle for Java**
- ✓ **pep8 for Python**
- ✓ **scan for Haskell**
- ✓ **??? for Rust, Go, et al.**



**gem  
install  
rubocop**

**Thank you for your  
cooperation. Good night.**



**RuboCop**

# Yell at me.

@colindean

cad.cx

[github.com/colindean/talks](https://github.com/colindean/talks)