RuboCop Ruby linter and formatter

CONCEPTS

A cop is one rule; cops belong to departments such as Style, Layout, Lint, Metrics, Naming, Security.

Safe autocorrect (-a) keeps behaviour; unsafe (-A) may change it. Review the diff after -A.

RUNNING

`rubocop`
Check all Ruby files here

`rubocop` *`<path> ...`*
Check given files or dirs

`rubocop --force-exclusion` *`<path>`*
Honour Exclude for explicit paths

`rubocop -v / -V`
Show version / full versions

CHOOSING COPS

`rubocop --only` *`<Cop>,<Dept>`*
Run only these cops or departments

`rubocop --except` *`<Cop>`*
Run all but these

`rubocop -l`
Lint cops only

AUTOCORRECT

`rubocop -a`
Fix safe offenses

`rubocop -A`
Fix safe and unsafe offenses

`rubocop -x`
Fix layout offenses only

`rubocop -a --only` *`<Cop>`*
Fix one cop everywhere

OUTPUT

`rubocop -f` *`<name>`*
Pick an output formatter

OUTPUT (cont.)

`rubocop -f offenses`
Count offenses per cop

`rubocop -f json -o` *`<file>`*
Machine readable report

CACHE AND SERVER

`rubocop --server`
Run via a background server

CONFIG FILES

`rubocop --init`
Write a starter .rubocop.yml

`rubocop --auto-gen-config`
Write .rubocop_todo.yml

YAML KEYS

`inherit_from:` *`<file>`*
Include other configs

`plugins: -` *`<name>`*
Load extension plugins

`AllCops: NewCops: enable`
Opt into pending cops

`AllCops: TargetRubyVersion:` *`<x.y>`*
Set Ruby version to parse as

`AllCops: Exclude: -` *`<glob>`*
Skip paths everywhere

*`<Dept>`***`/`***`<Cop>`***`: Enabled: false`**
Disable a cop

*`<Dept>`***`/`***`<Cop>`***`: Exclude: -`** *`<glob>`*
Skip paths for one cop

*`<Dept>`***`/`***`<Cop>`***`: Max:`** *`<n>`*
Tune a numeric limit

INLINE COMMENTS

`# rubocop:disable` *`<Cop>`*
Turn off from here

`# rubocop:enable` *`<Cop>`*
Turn back on

`<code>` **`# rubocop:disable`** *`<Cop>`*
Disable for one line