

Name	Last commit message	Last commit date
..		
.github/workflows	Update checkout and setup-node version (#235)	2 months ago
.rubocop.yml	Remove typo	2 months ago
README.md	Fix code snippets for better readability	2 years ago

# Ruby Course

If you are not familiar with linters and GitHub Actions, read [root level README](#).


## Set-up Rubocop GitHub Action


[Rubocop](#) is a Ruby static code analyzer (a.k.a. linter) and code formatter. It will enforce many of the guidelines outlined in the community [Ruby Style Guide](#).

This GitHub Action is going to run [Rubocop](#) to help you find style issues.

Please do the following **steps in this order**:



- In the first commit of your feature branch create a `.github/workflows` folder and add a copy of `.github/workflows/linters.yml` to that folder.
  - Remember to use the file linked above
  - Remember that `.github` folder starts with a dot.
- Do not make any changes in config files - they represent style guidelines that you share with your team - which is a group of all Microverse students.
  - If you think that change is necessary - open a [Pull Request in this repository](#) and let your code reviewer know about it.
- When you open your first pull request you should see the result of the GitHub Actions:



 **All checks have failed**


2 failing checks

[Hide all checks](#)

  Linters / Rubocop (pull\_request)

Failing after 28s — Rubocop


[Details](#)



**This branch has no conflicts with the base branch**

Merging can be performed automatically.

Merge pull request








You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Click on the `Details` link to see the full output and the errors that need to be fixed:



**Linters / Rubocop**

failed 23 hours ago in 28s



  

  **Set up job**



3s

  **Run actions/checkout@v2**



0s

  **Run actions/setup-ruby@v1**

1s

  **Setup Rubocop**

23s

  **Rubocop**

1s

1 ▶ Run rubocop --color

4 Inspecting 1 file

5 W

6

7 Offenses:

8

9 bin/main.rb:1:1: W: Lint/ScriptPermission: Script file main.rb doesn't have execute permission.

10 #!/usr/bin/env ruby

11 ~~~~~

12 bin/main.rb:3:19: C: Layout/TrailingEmptyLines: Final newline missing.



13 puts 'Hello world'

14



15

16 1 file inspected, 2 offenses detected

17 ##[error]Process completed with exit code 1.

  **Post Run actions/checkout@v2**

0s

  **Complete job**


0s


## [OPTIONAL]Set-up RSpec GitHub Action

You can run your tests with GitHub Actions to ensure that they are passing before merging a PR.

To use the GitHub Action to run your tests, please do the following **steps in this order**:



- Add a copy of `.github/workflows/tests.yml` to your `.github/workflows` folder.
  - Remember to use the file linked above
  - Do not modify or delete the `.github/workflows/linters.yml` file that should already be in that folder.
  - RSpec by default will try to run any file ending in `_spec.rb` inside the `spec` folder. Make sure to follow this convention for your tests files so `rspec` can run your spec files.
  - You can modify the `.github/workflows/tests.yml` file to better fit your custom needs.
- When you open your pull request you should see the result of the GitHub Action:



 **Some checks were not successful**



1 failing and 1 successful checks

[Hide all checks](#)

  Tests / RSpec (pull\_request)


Failing after 11s — RSpec

[Details](#)

  Linters / Rubocop (pull\_request)

Successful in 8s


[Details](#)



**This branch has no conflicts with the base branch**

Merging can be performed automatically.

Merge pull request



You can also [open this in GitHub Desktop](#) or view [command line instructions](#).

Click on the `Details` link of the test action to check the results of your tests.

## Set-up linters in your local env

### RuboCop

- Add this line to the `Gemfile`

```
gem 'rubocop', '>= 1.0', '< 2.0'
```

*not sure how to use Gemfile? Read [this](#).*
- Run `bundle install`.
- Copy `.rubocop.yml` to the root directory of your project
- Do not make any changes in config files - they represent style guidelines that you share with your team - which is a group of all Microverse students.
  - If you think that change is necessary - open a [Pull Request in this repository](#) and let your code reviewer know about it.
- Run `rubocop`.
- Fix linter errors.
- IMPORTANT NOTE:** feel free to research [auto-correct options for Rubocop](#) if you get a flood of errors but keep in mind that correcting style errors manually will help you to make a habit of writing a clean code!

## Troubleshooting

- While using Colorize gem, if you are facing errors with Rspec related to

```
LoadError:
cannot load such file -- colorize
```

please remove `--deployment` from line no. [26](#) of test.yml file.