

## Rails: Common Errors

Part of learning how to code includes learning how to debug errors. Both Ruby and Rails provide useful error messages that help you with debugging. There is even **byebug** to help you work on what went wrong. If you have not started using **byebug**, we strongly encourage you to start using it from now on.

Error messages can also appear in your terminal, for example, if a `bundle exec rake db:migrate` task was not performed successfully, an error message will appear in your terminal. It's always good to read the error message to know what went wrong.

### Challenge

1. List down the errors that arose during the lecture on "Introduction to Rails" and jot down how they were solved.
2. From now on, as you build your app, whenever you encounter an error, write them down and jot down how you solved it. Once you start logging these errors down, you will have a better understanding of why such errors occur and will know how to avoid it the next time. Or if it does happen again (we are humans after all), we will know how to approach it. It also helps you have a better understanding of what you are doing as well.

You can list your errors in a .txt file, a Github [README.md](#), in an Excel spreadsheet, in your own journal, or anywhere that you find convenient for you to refer to.

### Things to ask yourself when encountering errors:

1. Have I checked for spelling mistakes, especially for plural and singular nouns?
2. Do I really understand when to use `@` and when not to use `@` for a variable? How does it affect my code?
3. Have I checked if I have missed out an `end`, or conversely, have an extra `end`?
4. Have I used **byebug** to test my code? If it is a front-end related issue, have I open my developer tool and do the necessary checks (for example, inspect element, check the console output, check the sources, etc).
5. Have I read the error messages that appear on the browser and/or the terminal?
6. Did I miss out a `[""]` when I need to access a hash's value?

7. Am I clear about the difference between `<% %>` and `<%= %>`?
8. Have I named my files correctly? For example, if I was implementing Bootstrap, have I renamed my `css` files to have the extension `scss`?
9. Have I declared a variable before calling it?
10. Have I read and researched on the error message before asking for help?

In the blob on your right, we have also listed down some common errors faced while building a Rails app. Add to the list as you work on your app. Also, whenever you ask your mentor for help, pay attention to the mentor's thought process as he/she work through the problem with you. It's time that we learn to cycle on two wheels :) Let's learn the art of debugging and become better at it each day!