Become a Better Ruby Developer

By Jesus Castello

1

Practice Makes Mastery

The key for getting better at programming is no secret: you need to practice a lot. This is probably not a surprise to you, but you have to keep going no matter what. I know you can do it!

Every day you should dedicate at least 20 minutes to work on a side project or a small program that solves a specific problem, remember to be consistent!

If you don't know what to work on check out the following sites:

- Exercism.io
- Codewars.com
- Projecteuler.net
- Adventofcode.com

2

Master your editor

It doesn't matter what it is: vim, atom, sublime, notepad (ok not notepad...)

You need to understand all the nice things your code editor offers. Does it have a plugin system? What are the best plugins you can use to make you more effective?

Does it have a snippet/macro system? Learn what snippets you can use to work faster, and add new ones as needed.

3

Read source code

Working on your own projects is great, but that can only take you so far. To learn from others, you should pick a few of your favorite Ruby projects on GitHub and take a look at their code.

Look for things that stand out to you and ask yourself the following questions:

- What is this code doing?
- What can I learn from it?
- Can it be improved?

You will probably end up with a few new tricks you can apply to your own projects.

4

Read the docs

I can't stress enough how important is reading and having easy access to the Ruby documentation. Instead of having to guess a method name or its arguments you can just look it up.

You can also learn about new methods & classes by reading the documentation, which is a great way to expand your Ruby knowledge (I show you another method to do this on my Ruby Deep Dive course).

For quick access a bookmark is good enough, but you can also use an off-line documentation browser like Zeal.

5

Improving your style

Your programming style can make a big difference on how hard it is to read your code. Remember that code is read way more often than it's written, so it's important to keep it as clean as possible.

Using the style guide that the community has developed can help you improve on this aspect.

You can use tools to help you enforce a coding style, a popular one is Rubocop.

Thanks for reading!

If you enjoyed this guide do me a favor & share it with your Ruby friends :)

Here is the sharing url:

http://www.blackbytes.info/free-gift/