As we previously mentioned, the first step is to install RBENV on your machine. I’m writing this tutorial on a Macbook running macOS, and I’ll be using macOS-specific installation instructions and commands throughout.

**Even if you’re already an RBENV user**, you’ll want to read this chapter, since there’s a good chance you’ll need to re-install RBENV from source (as opposed to via Homebrew or similar).

## Ensuring you have no other version managers installed

Admittedly, the best candidates for reading this book are people who already use RBENV as their Ruby version manager. As you’ll soon see, users of other version managers (such as rvm, asdf, chruby, etc.) will face more and bigger hurdles than users of RBENV (more on why in a minute).

With that in mind, we’ll need to make sure you don’t have a different Ruby version manager installed, such as RVM. If you do, that will represent a blocker to your continuing this guide, since you’ll have different version managers competing to manage your Ruby version. This could introduce [unexpected behavior](https://web.archive.org/web/20160715135727/https://stackoverflow.com/questions/35808103/rvm-and-rbenv-on-the-same-machine){:target=“\_blank” rel=“noopener”} and negatively impact your usage of Ruby.

We can check for the most popular Ruby version managers by using the which command:

If you see anything other than “not found” for which asdf, which rvm, and which chruby, you likely have another version manager on your machine. In that case, you’ll need to make a decision about which version manager you want to use. [RBENV’s Github page has a guide](https://github.com/rbenv/rbenv/wiki/Comparison-of-version-managers){:target=“\_blank” rel=“noopener”} on differentiating between the various version managers out there.

If you don’t have other version managers on your machine, feel free to move on to the next section.

## Making sure RBENV is installed correctly

Installing RBENV via brew install rbenv would be a perfectly fine option for a normal user. But it won’t work for our purposes, because it would leave us without access to RBENV’s .git directory, and therefore its git history. That means we couldn’t roll back to the specific version of RBENV that I’ll be using for this walk-through. It’s important that we work off the same codebase, so let’s try another technique- installing from source.

To do this, we’ll follow the instructions on [this version of the RBENV Readme file](https://github.com/rbenv/rbenv/tree/e8b7a27ee67a5751b899215b4d35fd86ab552dae#basic-git-checkout){:target=“\_blank” rel=“noopener”}. First, open your terminal and check whether you already have a directory called ~/.rbenv, by running ls -la ~/.rbenv:



some alt text

If this directory does exist, and the command output has a .git directory in it, you should be good to go.

Another possibility is that the output exists but looks like this:

Specifically, you run the ls -la command and you don’t see a .git folder included in the output.

That likely means you previously installed RBENV using a package manager like Homebrew. If this is the case, one option is to simply rename your /.rbenv directory as /.rbenv-old or something similar:

Then we can install RBENV again via source, by following the instructions on [this version of the RBENV Readme file](https://github.com/rbenv/rbenv/tree/e8b7a27ee67a5751b899215b4d35fd86ab552dae#basic-git-checkout){:target=“\_blank” rel=“noopener”}. Run the following command:

$ git clone https://github.com/rbenv/rbenv.git ~/.rbenv

Once this is done, you should have a fresh installation of RBENV inside ~/.rbenv, which includes a .git directory:

## Ensuring your new RBENV install has your old data

Next, we’ll copy your currently-installed Ruby versions, gems, etc. to this new installation, ensuring that you can continue using RBENV as you have up until now:

$ sudo cp -r ~/.rbenv-old/versions \   
> ~/.rbenv-old/version \   
> ~/.rbenv-old/shims \   
> ~/.rbenv-old/rbenv.d \   
> ~/.rbenv-old/completions \   
> ~/.rbenv

Now, the version of RBENV which you installed from source should have any and all gems, installed Ruby versions, selected Ruby version, completions, and hooks that you may have previously installed in your old RBENV version.

If you ever encounter version-related problems with Ruby or RBENV on your machine from now on, you can simply delete the version of RBENV that we just installed, and rename your ~/.rbenv-old directory back to ~/.rbenv, and you should be good-to-go.

## Rolling back to the correct git commit

Next, let’s navigate into this directory via cd ~/.rbenv.

We want to make sure that we’re all looking at the same code, which means looking at the same git commit. So we’ll create a new branch, separate from your master or main branch, and point that branch to a specific commit, i.e. the one that I used when I started writing this guide (that commit SHA is c4395e58201966d9f90c12bd6b7342e389e7a4cb).

I called my new branch impostorsguides, but you can call it whatever is easiest for you to remember:

For future reference, the Github link to this specific version of the RBENV codebase can be found [here](https://github.com/rbenv/rbenv/tree/c4395e58201966d9f90c12bd6b7342e389e7a4cb){:target=“\_blank” rel=“noopener”}.

## Enabling RBENV’s shell function

We’re getting close to the end, but we still have 2 more steps in [the installation instructions](https://github.com/rbenv/rbenv/tree/e8b7a27ee67a5751b899215b4d35fd86ab552dae#basic-git-checkout){:target=“\_blank” rel=“noopener”}. Next we have to add some text to our shell’s startup script. This script will create a shell function called rbenv, which (for our purposes) will do the same job as if we were running the command from a file.

The RBENV Readme file tells you how to add the text to your script. The command you’ll copy/paste into your terminal depends on which shell program (bash, zsh, etc.) you’re running:

To find out which shell you’re running, run the following command:

$ echo $SHELL

In my case, since I’m on a new Mac, my default terminal is Zsh:

Therefore, I’d paste the following into my terminal:

$ echo 'eval "$(~/.rbenv/bin/rbenv init - zsh)"' >> ~/.zshrc

This will add the text eval "$(~/.rbenv/bin/rbenv init - zsh)" into a file called ~/.zshrc, which is run every time I open a new terminal tab. If your terminal is different from mine (for example, bash), the filename will be different from ~/.zshrc (for example, ~/.bashrc), but the gist of what’s happening is still the same.

Lastly, in order for these changes to take effect, I’ll need to do exactly that- open a new terminal tab. When I do so, and I run which rbenv, I see the following shell command definition:

If we see the above output, we know RBENV has been successfully installed.

## Summary

Now that we’ve installed RBENV and are pointing to the right version, let’s move on.