

# CSE341: Programming Languages

## Installing and Using Ruby and irb

### Autumn 2019

## 1 Overview

The last two homeworks will be using the Ruby language. We recommend editing your files in Emacs and using irb, which is Ruby's REPL. (If you prefer a different editor that has support for Ruby, that is fine.) This document describes basic installation and usage steps sufficient for doing your homework. Unlike when we used SML, we recommend running the REPL from a terminal (shell) window, *not* from within Emacs. This is described below.

For installation purposes related to Homework 6, there are some key facts:

- **You need to have some version of Ruby 2, but in order to most easily use the Tk graphics library, which is necessary for Homework 6, we recommend Ruby 2.2.X for some X. We have had installation troubles for more recent versions.** We will grade with Ruby 2.2 but believe any version of Ruby 2.x.y (or even 1.9) should work fine if installed and working with the necessary graphics. Please do not use Ruby 1.8 since there are substantial language differences (though not relevant to any of our high-level concepts).
- You need to have a version of the Tk graphical toolkit.
- If you have Emacs version 23 or higher, you should not need to configure it in any special way for using Ruby: opening a file with extension `.rb` should use Ruby mode.
- If you have a version of Ruby that is not working, we recommend you uninstall it before installing a different version, though there are ways to avoid doing this if needed.

Because this document describes installation for various operating systems, you do not need to read it all. Just find the section that is most convenient for you. But also notice the last two sections of this document, which have information relevant to all operating systems.

## 2 Windows

### 2.1 Using Windows in the Department's Undergraduate Labs

- You do not need to install any software.
- Create a Ruby file (e.g., by downloading the provided code and renaming it) wherever is convenient (e.g., where you have been storing your other homework assignments). **Do not store it on the desktop or under C:\ because all changes to these directories get erased when you log out. Store your files under Z:\ (your directory on the department file system) or somewhere else (e.g., on a personal usb drive).**
- Open Emacs however is convenient. You should be able to find it by, e.g., typing `emacs` in the "Search Programs and Files" from the Start Menu.
- Open your Ruby file in Emacs, by dragging it from an Explorer window onto Emacs, or by C-x C-f and then entering the full path to the file, or by using the File menu. Edit and save the file as usual.

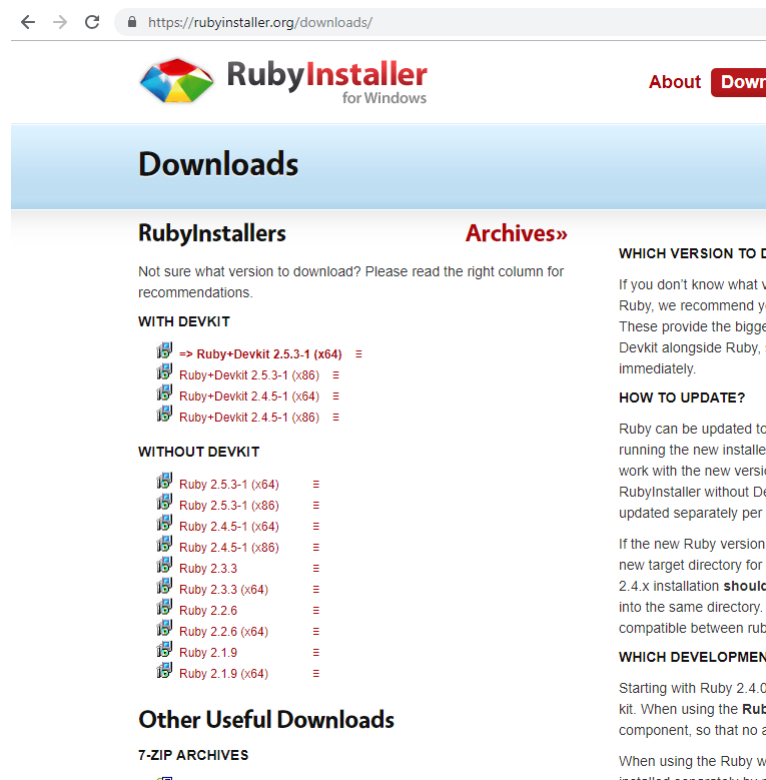
- In Windows, open a command shell, either by running `cmd` (a Windows shell) or a cygwin bash shell. Use the `cd` command to switch to the directory where your Ruby file is (using the Windows shell you separate directories with backward slashes (\) and with cygwin you use forward slashes (/)). *After* you are in the right directory, type `irb` to start the Ruby REPL. Alternately, you can use File Explorer to navigate to the folder holding your Ruby file, then hold down shift while right-clicking to bring up an option to “Open a command shell here” — this will open the shell already switched to that directory.

## 2.2 Using the Department’s Windows Machines Remotely

- Follow the instructions for your operating system at <http://vdi.cs.washington.edu/> to connect.
- Once connected, follow the directions above for *Using Windows in the Department Undergraduate Labs*.
- Save your work often in case you lose your connection.

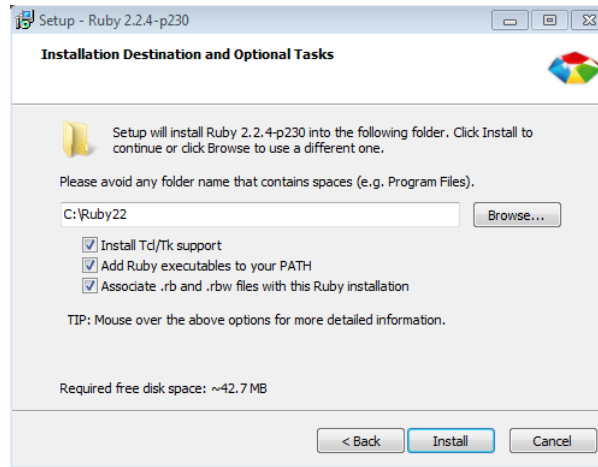
## 2.3 Using Your Own Windows Machine

- If you did not install Emacs earlier in the course, do so using the instructions for the SML portion of the course. Note you do not need SML mode — all you need to do is install Emacs.
- Go to <http://rubyinstaller.org/> and click on the giant red ‘Download’ button. You should see a list of many versions you can download:



- The easier and faster way is to click on the installer for **Ruby 2.2.6** that is down below all the more recent versions (no need for the x64 variant, though it’s probably fine too).
  - Download it.

- Run the installer.
- Accept the License.
- **On the next screen, click all three boxes: Install Tcl/Tk support, Add Ruby executables to your PATH, and Associate .rb and .rbw files with this Ruby installation:**



- Then click Install.
- If you did Ruby 2.2.6, skip to the next bullet. If you instead prefer a more recent version, here are the steps:
  - Download a more recent version with the Devkit (e.g., 2.5.5-1 (x64)).
  - Run the installer.
  - Accept the License.
  - Leave the check-boxes how they are.
  - At the end of installation, leave the check-box about running “ridk install” checked as you click Finish.
  - This bring up another Window.
  - In that window, type 1 and then Enter.
  - After this is done, type 3 then Enter.
  - After this is done, type Enter, which will close the window.
  - Open The Windows Command Prompt (you can hit the Windows button and then type `cmd`, the name of this program).
  - Type `gem install tk` at the Windows Command Prompt.
  - Whew, we’re done, that was all one-time installation, so no need to repeat any of these steps.
- Create a Ruby file (e.g., by downloading the provided code and renaming it) wherever is convenient (e.g., where you have been storing your other homework assignments).
- Open Emacs however is convenient, e.g., from the Start Menu.
- Open your Ruby file in Emacs, by dragging it from an Explorer window onto Emacs, or by C-x C-f and then entering the full path to the file, or by using the File menu. You can now edit and save the file as usual.

- In Windows, open a command shell by running `cmd` (a Windows shell). Use the `cd` command to switch to the directory where your Ruby file is (using the Windows shell you separate directories with backward slashes (`\`)). *After* you are in the right directory, type `irb` to start the Ruby REPL. Alternately, you can use File Explorer to navigate to the folder holding your Ruby file, then hold down shift while right-clicking to bring up an option to “Open a command shell here” — this will open the shell already switched to that directory.

## 3 Linux

### 3.1 Using Linux in the Department’s Undergraduate Labs

The version of Ruby we recommend using is not installed directly on these machines, but that is okay. You can log-in to these machines and then follow the instructions below for, “Using the Department’s Remote Linux Server” on Linux or Mac.

### 3.2 Using Your Own Linux machine

Most Linux distributions should make it easy to install a recent version of Ruby and connect it to the Tk graphics library.

- When you install Ruby, explicitly get some 2.x.y version (e.g., 2.2.4). If you install the default “ruby” package, check the version.
- You can use your favorite package manager GUI, or on the command line. For example, on Debian (Ubuntu, Mint) systems, type: `sudo apt-get install ruby2.2.4`
- If `irb` is a separate package, install that too, again looking for an explicit use of the same Ruby version.
- After installation, you can check the version by running `ruby --version`.
- Create a Ruby file (e.g., by downloading the provided code and renaming it) wherever is convenient (e.g., where you have been storing your other homework assignments).
- Open Emacs or whatever text editor you are using.
- Open your Ruby file in Emacs by `C-x C-f` and then entering the full path to the file, or by using the File menu. You can now edit and save the file as usual.
- Open a command shell however is convenient.
- Use the `cd` command to switch to the directory where your Ruby file is (separating directories with forward slashes). *After* you are in the right directory, type `irb` to start the Ruby REPL.

### 3.3 Using the Department’s Remote Linux Server

*Using `attu` will not work; it does not have the version of Ruby we need. Instead use `ruby22.cs.washington.edu`, which is like `attu` but set up just for us.*

- On Linux or Mac
  - Open a terminal window

- Run `ssh -X uname@ruby22.cs.washington.edu`, replacing `uname` with your CSE id, and log in with your CSE Linux password.
- On Windows
  - Use a program like PuTTY (<http://www.putty.org>) to gain access to SSH.
  - Connect to `uname@ruby22.cs.washington.edu`, replacing `uname` with your CSE NetId, and log in with your CSE Linux password.
- On `ruby22`, open Emacs and your Ruby code.
- On `ruby22`, run `irb` or `ruby` from the command line as described below.

### 3.4 Using the Department’s Linux Virtual Machine on Your Machine

The department provides a Linux virtual machine that runs on your own computer no matter what operating system you have.

#### Installation

- The instructions for the current release are available here:  
<https://www.cs.washington.edu/lab/software/linuxhomevm>.
- For one-time set-up, run these three commands at a shell prompt:

```
sudo yum -y install sqlite-devel
curl -sSL https://get.rvm.io | bash
rvm reinstall ruby-2.2.3 --with-tcl --with-tk
```

- After installation, you can check the version by running `ruby --version`.
- Create a Ruby file (e.g., by downloading the provided code and renaming it) wherever is convenient.
- Open Emacs or whatever text editor you are using.
- Open your Ruby file in Emacs by C-x C-f and then entering the full path to the file, or by using the File menu. You can now edit and save the file as usual.
- Open a command shell however is convenient.
- Use the `cd` command to switch to the directory where your Ruby file is (separating directories with forward slashes). *After* you are in the right directory, type `irb` to start the Ruby REPL.

## 4 Mac

How to install what, if anything, you need, depends on your version of Mac OS X. After you’re done installing, see the last section for how to edit files and run Ruby.

**If installing Ruby, use Ruby Version 2.2.3 or similar, not Ruby Version 2.3.X, which appears to have a bug that prevents using our graphics code.**

If you find the instructions do not work, please let us know so we can update them.

## 4.1 Installation: Mac OS X Mojave, High Sierra, Sierra, and El Capitan

Ruby is already installed, but for Homework 6 we will also need the Tk graphics library, so we will install Tk and a different version of Ruby (2.2.3) known to work as follows:

- Make sure you have the Homebrew package manager installed (check using `which brew` in Terminal) else install it. Also check that the directory `/usr/local/Cellar` exists and is owned by you (not root).
- Install ActiveTcl 8.5 instead of the latest version from <http://www.activestate.com/activetcl/downloads> or this more convenient direct link: [http://downloads.activestate.com/ActiveTcl/releases/8.5.18.0/ActiveTcl8.5.18.0.298892-macosx10.5-i386-x86\\_64-threaded.dmg](http://downloads.activestate.com/ActiveTcl/releases/8.5.18.0/ActiveTcl8.5.18.0.298892-macosx10.5-i386-x86_64-threaded.dmg).
- Install the Ruby version manager with the instructions to “Install RVM stable with Ruby” at <https://rvm.io/rvm/install> (basically run a couple commands on that web page in Terminal).
- Now in Terminal, run the command `rvm reinstall ruby-2.2.3 --with-tcl --with-tk` (see <http://stackoverflow.com/questions/20874900/ruby-require-tk-yields-loadererror-no-such-file-to-load-tk>).
- Now in Terminal, tell rvm to make the installation from the previous step the default version via `rvm use 2.2.3 --default`.

## 4.2 Installation: Mac OS X Yosemite and Mavericks

Ruby 2.0 is already included, which should work fine, and should already be connected to Tk graphics. So, there's nothing to do.

## 4.3 Installation: Mac OS X Mountain Lion and Older

By default, you have Ruby 1.8.7 installed if you have Ruby installed at all, so you will need a more recent version of Ruby. Try to follow the same instructions as for El Capitan above, but if you run into problems, email the course staff and we will try to help and, working with you, provide more detailed instructions by updating this document. (Or you can use a different approach, such as using the department labs, the department virtual machine, or the department's remote access.)

## 4.4 Editing and Running Code

- You should be able to use Emacs to edit files and your *Terminal* program to navigate directories and run `irb` and `ruby`. In other words, your Mac *Terminal* behaves like a Linux terminal. Use the `cd` command to switch to the directory where your Ruby file is (separating directories with forward slashes). *After* you are in the right directory, type `irb` to start the Ruby REPL.
- For a better terminal application, consider *iTerm2*, available free at <http://www.iterm2.com/>.
- For a native Emacs experience on Mac, consider *Aquamacs*, available free at <http://aquamacs.org/>.

## 5 General Information on Using (or Not Using) the REPL

- As described previously, run `irb` from the command line to start the REPL.
- To run the code in file `foo.rb`, do `load "foo.rb"` assuming the file is in the same directory where you started `irb`.

- As usual, it is least error-prone to restart the REPL after editing and resaving any files you are using. Reloading a file without restarting may work depending on what has changed.
- To quit, type `quit` or `exit` or *Control-d* (i.e., `d` while holding down the Control key). As usual in Ruby, there is more than one way to do things.
- You can cycle through previous input lines by using the up and down arrows.
- Unlike in SML, do *not* end a line of REPL input with a semicolon. This will *not* work because this indicates to the REPL exactly that you want to enter another expression before evaluating anything.

You can also run a Ruby program that is in file `foo.rb` by running `ruby "foo.rb"` from the shell command-line (the place where you ran `irb`, *not* from within `irb`). For this to be useful, your Ruby file should have some top-level expression like a call to a method outside of any class or method definition. Otherwise, “nothing will happen” since just defining methods has no effect until you use one of them.

On Windows and perhaps other operating systems, you can also just double-click on your `foo.rb` file to have the same effect as running `ruby foo.rb`. This approach may bring up another blank window, which you can ignore.

## 6 Main Links for Ruby Documentation, etc.

- The main Ruby home page is <http://www.ruby-lang.org/>.
- For documentation, you may find <http://ruby-doc.org/> at least as useful. See in particular the standard-library documentation. (The library differences among recent versions are probably very minor.)
- There are many excellent books on Ruby available, which you may find useful but not required for the material in this course. We particularly recommend Programming Ruby 1.9 & 2.0: The Pragmatic Programmers’ Guide by Dave Thomas et al; see <https://pragprog.com/book/ruby4/programming-ruby-1-9-2-0>.