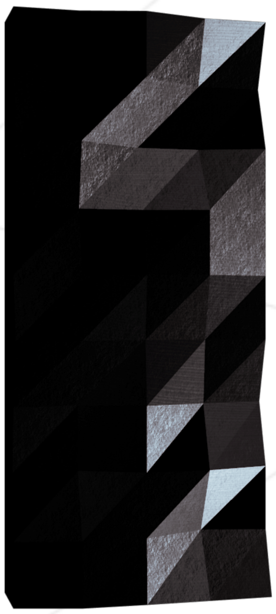

Intro to Coding Sequence

Scatter, learn things, and regroup

Kai kai@42.us.org

Summary: Keep a friend at your side, and study in the path of inspired ones who have given their knowledge out for free.

Contents

Chapter I

Before you Start!

- Download the `setup.sh` script included with this project. Then, in your terminal, run it by typing "sh ~/Downloads/setup.sh". If it gives you error messages, copy and paste the error message in the #setup_help Slack channel and ask your mentors.

- For each piece of this project, you can work with a partner or work alone.

- If you are on our AP Computer Science Principles track, take this practice AP exam for a baseline score: `quiz.42.us.org` (Set aside 1 hour and answer as many as you can. The real exam is 2 hours.)

- Join some useful Slack groups! Click on "Channels" and join `#pdf_questions`, `#setup_help`, `#ruby-help`, `#python-help`, and one of the AP channels if interested.

- Create your project folder:

  1. From your project page on intra, copy the git repository link. Now, in the terminal type "git clone " and paste the link. After the link and before pressing enter, write a name for the new folder. Cloning your git repository always creates a new folder.

  2. cd into the folder you just created and from now on, save your work there. Use the command "mkdir <name>" to create new folders. Put each puzzle from this project in a folder with the same name.

Chapter II

Coding Tools

Set up your programming environment by picking a program for each of three essential functions. Our favorites are listed below.

If you want to install one which is not already on your computer, simply download the program and then drag its icon to your desktop or to a folder of your choice (you won't have access to the Applications folder).

1. Terminal
   - Built-in: Terminal
   - Upgrade: iTerm2
2. Text Editor
   - In the terminal: Emacs or Vim
   - Free: Atom
   - Free: Sublime
   - Built-in: Xcode
   - Free: Brackets
3. Web browser
   - Built-in: Safari
   - Free: Google Chrome
   - Free: Opera
   - Free: Vivaldi

Chapter III

Study One of These

Since we offer a series of programming challenges, not a step by step walkthrough, you will need to search online for resources that can explain each piece to you.

I've pulled together a list of freely offered, noncommercial Intro to Programming courses to recommend.

Use your chosen curriculum to learn about programming. Feel free to either follow it from start to finish or skip around. You need to put together the puzzle pieces from what those tutorials teach you to solve our programming challenges below.

III.1 Ruby

- Chris Pine's Learn to Program: `https://pine.fm/LearnToProgram/`
- Bastards' Book of Ruby, sections from "Style, Conventions, and Debugging" to "File input/output". `http://ruby.bastardsbook.com/toc/` (Don't worry about the tweet fetching scripts, they no longer work with Twitter's new API.)
- Learn Ruby the Hard Way: `https://learnrubythehardway.org/book/`
- Test-first Ruby: `http://testfirst.org/learn_ruby` - best for someone with some programming experience, but really fun. The same author wrote `http://codelikethis.com/lessons/learn_to_code` which provides a beginner overview. Make sure you learn "puts" if you choose the test-first course.

III.2 Python

- IntroPython.org: `http://introtopython.org/hello_world.html`
- A Byte of Python: `https://python.swaroopch.com/`
- MIT Gentle Introduction to Programming using Python: `https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-189-a-gentle-introduction-` `lectures/`

- Google's open Python class: https://developers.google.com/edu/python/ - best for people with a bit of programming experience already.

III.3 Java

I'm accepting recommendations for intro courses on this language. We don't teach it here, but you should feel free to study for the AP Computer Science A exam by completing all programming challenges in Java - there is generally always a way to do the same thing in a different language! You will just need to explain the syntax while getting corrected by your peers.

`In addition to studying these resources and Googling your questions, each challenge chapter contains a "Notes" section which links you directly to the official documentation of Ruby and Python. It's a good idea to get familiar with the layout of official documentation - that means you can find answers directly from the source.`

III.4 Gemstones? Invertebrates?

"Which language should I learn?"

If you do not know either Ruby or Python, you may not have anything to base your preference on.

Programmers love to argue about the superiority of their favorite tools but the truth is that both of these are popular, useful, modern programming languages that have similar abilities. Python is used more widely and Ruby is a bit more "poetic" or aesthetically pleasing.

You can research opinions online if you would like to read something about their characteristics.

If you have no prejudice, I arbitrarily recommend that you go with Ruby =)

It's what Intra is written in and thus it is what we love.

III.5 Going Retro

Ruby and Python are updated on a regular basis with new features.

So, for example, hashtables are written using different symbols in Ruby 1.9.3 than they are in the modern version, Ruby 2.4.2.

Python 3 is a big leap forward from the features included in Python 2. However, it's not supported by all of the tools that interface with Python yet.

`Setup.sh` sets your computer to use Ruby 2.4.2 and Python 2.7 by default. It also installs the tools needed to easily switch between versions of both languages (so please run it regardless of which version you want)!

Some of the freely offered coding classes are written for a different version of the programming language than Ruby 2.4 & Python 2.7. If that's the case, post in the `#setup_help` Slack channel for help using `rbenv` and `pyenv` to switch the version used by your computer.

Chapter IV

Turn In

In each folder, please turn in the assigned project in a .rb or .py file named as instructed in the PDF. Place your scripts in the top level of your Vogsphere repository, not within your nested folder.

When you type "ls -a", you should see the .git folder alongside your .rb and .py files.

Each day, turn in your work so far by typing three commands in order:

- git add *
- git commit -m "<your comments here>"
- git push
- If you have an error during the git push, you may need to refresh your authentication ticket. Do this by typing "kinit <username>" and then typing your intra password.

Chapter V

Format your Code

Each 42 challenge you turn in must adhere to the following format:

```
#!/usr/bin/env ruby

# This is what my program does
# By <userid>

def function_a
 #code
end

def function_b
 #code
end

def main(ARGV)
 #main method
 function_a
 function_b
end

main(ARGV)
```

- Always begin with the "#!/usr/bin/env ruby" statement. This tells your terminal to run the program using Ruby. In python, the first line is "#!/usr/bin/env python".
- Always add a comment stating what this program is for, some hints to help others use or understand it, and your name or intra ID.
- Do not write any code outside of functions except for one line, at the end of your program, which calls the main() function.
- The (ARGV) parameter is not always needed. In Python it is sys.argv.

Reference your chosen intro to coding class to learn about functions/methods (The keyword "def" means "define function...").