Before entering Phase 0, you will need to be able to set up and use your environment, know some programming basics, and have an understanding of basic ruby concepts. This checklist outlines the **minimum** level of knowledge you will need to enter Phase 0. This phase is called Prep.

The list has an implied order and was designed for beginners. If you are completely new to programming, this prep work will probably take between 50-80 hours to do well. If you have programming experience, review each concept before checking it off.

For any technical support, please feel free to scope out and post in the [Dev Bootcamp Help Facebook group](https://www.facebook.com/groups/devbootcamp.help/).

**You should feel confident with each objective BEFORE entering Phase 0.**

|  |  |  |
| --- | --- | --- |
| **Topic** | **Objectives: You are able to** | **Resources** |
| Computer set-up | ▢ Install Ruby version 2.0 (check the version by running “ruby –v” from the command line). You may need to update.  ▢ Install and expand your understanding and skill with our preferred text editor - [Sublime Text](http://www.sublimetext.com/).  ▢ Run a ruby program from Sublime in your terminal | * [RailsBridge Installfest Instructions (only do the parts before setting up a Heroku account)](http://docs.railsbridge.org/installfest/) * [Chris Pine, Learn to Program, chapter 1](https://drive.google.com/file/d/0B7YM0hCMn56weUdWTFRZVzBQTDA/edit?usp=sharing) * [Perfect Workflow in Sublime Text](http://net.tutsplus.com/articles/news/perfect-workflow-in-sublime-text-free-course/) |
| Command Line | ▢ Navigate between directories  ▢ Create a new directory  ▢ Delete a directory  ▢ List files in the folder  ▢ Identify the folder you are in  ▢ Move a file between folders  ▢ Copy a file or directory  ▢ Open a file in a text editor  ▢ Run a ruby file | Required:   * [The Command Line Crash Course](http://cli.learncodethehardway.org/book/) |
| Counting and basic math | ▢ Explain how to count in decimal  ▢ Describe how a computer uses binary  ▢ Describe why counting in hexadecimal is useful  ▢ Compare bits to bytes  ▢ Explain “floating point” and its challenges  ▢ Describe the difference between integer division and division using floats | Required:   * [“How to Count” by Steven Frank ($2.99)](http://www.amazon.com/Count-Programming-Mere-Mortals-ebook/dp/B005DPIKPE/ref=sr_1_1?ie=UTF8&qid=1380317094&sr=8-1&keywords=how+to+count)   Optional:   * [Chris Pine, Learn to Program, chapter 2](https://drive.google.com/file/d/0B7YM0hCMn56weUdWTFRZVzBQTDA/edit?usp=sharing) * [Learn Ruby the Hard Way, exercise 3](http://ruby.learncodethehardway.org/book/ex3.html) |
| Ruby | ▢ Identify strings, integers, arrays, hashes, floats  ▢ Define a variable  ▢ Interact with a user via the command line  ▢ Describe flow control using if, elsif, else, and unless  ▢ Convert a method using if/else to case using “when”  ▢ Compare “=” and “==”  ▢ Describe loops and explain how the following loops work (#times, #each, #while, #until)  ▢ Compare arrays and hashes  ▢ Describe methods, blocks, and classes  ▢ Summarize the objectives of object-oriented programming | Required:   * [Learn to Program, chapters 3-15 by Chris Pine](https://drive.google.com/file/d/0B7YM0hCMn56weUdWTFRZVzBQTDA/edit?usp=sharing) (read-only) * [Codecademy](http://www.codecademy.com/tracks/ruby) (chapters 1-14, and 16)   Optional:   * [Try Ruby](http://tryruby.org/levels/1/challenges/0) * [Learn Ruby the Hard Way](http://ruby.learncodethehardway.org/book/) * [Treehouse Ruby Foundations](http://teamtreehouse.com/library/programming/ruby-foundations) (subscription required) * [CodeSchool’s Ruby Bits](http://www.codeschool.com/courses/ruby-bits) (if you feel confident with these objectives and want more (subscription required) |
| IRB | ▢ Load IRB  ▢ Explain what IRB is and how it can be used  ▢ Solve basic mathematical problems (+, -, \*, /, \*\*, %)  ▢ Create and manipulate variables  ▢ Create and add items to an array  ▢ Compare enumerable#collect and  enumerable#map  ▢ Compare #map and #map! methods | Suggested   * [Ruby in Twenty Minutes](https://www.ruby-lang.org/en/documentation/quickstart/)   Note: Get comfortable with IRB. Open your terminal, type “irb” and play with the skills you learned in Pine’s *Learn to Program* andCodecademy’s Ruby Course to complete these objectives. |
| Ruby Syntax | ▢ Indent using two-spaces  ▢ Find basic syntax errors from error messages  ▢ Use proper indentation | * Follow formatting styles you saw in Codecademy and Chris Pine’s book. Use their formatting in all of your code. |
| HTML and CSS  HTML and CSS (cont.) | ▢ Identify an opening and closing tag  ▢ Determine when to use various HTML tags  ▢ Indent HTML according to established guidelines  ▢ Identify CSS selectors  ▢ Explain how CSS formats HTML documents  ▢ Identify the parent/child elements in HTML  ▢ Compare classes and IDs in HTML  ▢ Describe the CSS box model  ▢ Explain basic CSS positioning (inline, inline-block, float, padding, margins, borders, absolute, relative, fixed) | Suggested   * [Codecademy: Web Fundamentals](http://www.codecademy.com/tracks/web) * [Conceptual intro to web development](http://milanlandaverde.com/blog/2013/10/20/conceptual-introduction-to-web-development/)   Optional   * [CodeSchool: CSS Cross Country](http://www.codeschool.com/courses/css-cross-country) (subscription needed) |
| JavaScript | ▢ Create variables using proper naming conventions  ▢ Evaluate Boolean statements  ▢ Write basic functions to solve mathematical problems (5 + 4)  ▢ Describe flow control using if , else if, and else  ▢ Compare for, while, and do loops  ▢ Define &&, ||, and !  ▢ Compare literal and constructor notation  ▢ Identify the similarities between OOP in Ruby and JavaScript | Suggested   * [Codecademy: JavaScript](http://www.codecademy.com/tracks/javascript)   Optional   * [Mozilla's Coding Style for JavaScript](https://developer.mozilla.org/en-US/docs/Developer_Guide/Coding_Style#JavaScript_practices) * [Code Combat (JavaScript game)](http://www.codecombat.com)   [CodeSchool’s JavaScript Roadtrip](http://www.codeschool.com/courses/javascript-road-trip-part-1) (pt. 1 is free) |
| Personal Preparation | ▢ Explain the difference between growth and fixed mindset  ▢ Describe how you learn best  ▢ Explain basic programming concepts to another person | * [Growth vs. Fixed Mindset](http://michaelgr.com/2007/04/15/fixed-mindset-vs-growth-mindset-which-one-are-you/) * [VARK questionnaire on learning style](http://www.vark-learn.com/english/page.asp?p=questionnaire) * [Gregorc Thinking Styles](http://www.thelearningweb.net/personalthink.html) |
| Additional (Optional) Resources:  *Note: Even though these are optional, they are very useful in orienting your skills and learning. If you’ve completed the previous objectives please make sure and check these out, all of them if you find the time.* | [Ruby Docs (1.9.3)](http://ruby-doc.org/core-1.9.3/) (official Ruby documentation)  [Ruby Monk](http://rubymonk.com/) (online activities and tutor)  [The Poignant Guide to Ruby](http://mislav.uniqpath.com/poignant-guide/) (online book)  [Practical Object Oriented Design in Ruby](http://www.poodr.com/) (Book)  [The Well-Grounded Rubyist](http://www.amazon.com/The-Well-Grounded-Rubyist-David-Black/dp/1933988657) (Book)  [JavaScript: The Good Parts](http://www.amazon.com/JavaScript-Good-Parts-Douglas-Crockford/dp/0596517742/ref=sr_1_4?s=books&ie=UTF8&qid=1380675525&sr=1-4&keywords=eloquent+JavaScript) (Book)  [Pragmatic Thinking & Learning](http://www.amazon.com/Pragmatic-Thinking-Learning-Refactor-Programmers/dp/1934356050) (Book)  [CS50: Harvard Intro Computer Science Course](https://www.cs50.net/) (Awesome lectures)  [Michael Hartl’s RoR Tutorial](http://ruby.railstutorial.org/) |  |