Learning Coding Checklist

Websites to Learn: LeetCode, HackerRank

Math & Computer Problems: Project Euler

Code Review: Code Review, a Stack Overflow community. Get a review of your code. Take a look at the beginners section.

Read a lot of code: Go to GitHub and read a bunch of code. This is the only way you will learn to code idiomatically.

Install Linting Tools: Flags bugs, idiomatic offenses, style errors and more.

Build Something: Jump in and build. Quantity over quality when you’re first starting.

Don’t Give Up: Beat the learning curves.

Don’t Stop Studying: Spend time each day studyi

ng. Use notecards or phone apps to study on the toilet, in the bath, while in waiting rooms.

Learn Problem Solving: Improve skills at pattern recognition, algorithms, and abstractions.

Learn Theory: Learn the lingo and jargon that occur in every language (loops, variables, methods, etc.).

Free Online Courses: Check out Harvard’s CS50X Intro to Computer Science course. You can audit courses free via edX and watch lectures on YouTube.

Online Camps: Check out FreeCodeCamp.

Coding Games: Hour of Code, Minecraft Educational Edition, CodinGame.

Coding Video Channels: CS Dojo, TheNetNinja, Kudvenkat, CS50, LinkedIn Learning’s video library.

Google Error Messages: Your not the first person to usually have this problem. Play around with what you Google. Think about your true problem, the correct terminology, and the frameworks or context you are working in as you assemble your search queries. Leave out filler words.

Post Questions Online: Take advantage of Stack Overflow or Reddit’s Programming. Remember, there are sections just for beginners.

Reverse Engineer: Clone someone else’s code and try to add onto it or make changes.

Mentor: Get a mentor that can help you. Make sure to return the value to that person or pass it on.

20 Minute Rule: Take atleast 20 minutes to figure something out before you even think of asking for help. Strengthening your problem solving skills is more important than your knowledge of languages.

Debug: Step through code to see how it works. Open up the objects as the code is loading to understand what’s inside of them.

Coding Academies: Check out Codecademy, freeCodeCamp, edX, Codewars, GA Dash, MIT OpenCourseware, SoloLearn (mobile), The Code Player, Haskr.io, W3Schools, Microsoft Virtual Academy, Edabit, Launch School Open Book Shelf, LearnCode.academy, Css-Tricks, Lean Enough Command Line to Be Dangerous, Command Line Power User, Git Immersion, Try Git, Javascript for Cats, NodeSchool, Eloquent Javascript, Javascript.com, Cybrary, The Encyclopedia of Human Computer Interaction, UXPin, UX Beginner

Competitive Programming: SPOJ, Codechef, Codeforces, Topcoder, Hackerrank, Geeksforgeeks.

Determine Your Best Learning Method: Visual, Auditory, Reading/Writing, Kinesthetic.

Visual Learner: Use mind maps, videos, diagrams.

Lookup Projects: Create a project per language that has the fundamentals you can refer back to.

Technical Interview Preparation: Check out Cracking the Coding Interview, LeetCode, Gainlo, Glassdoor.

Create List of Questions: Find an expert to answer your list once compiled.

Diversity: Take projects that will allow you to learn different parts of a language.

Compare Languages: Look for things that look familiar and act the same, things that look familiar but do something unexpected, and things that look new.

Official Docs: Read the languages official docs and code while you are doing it so you can see how it works.

Treat yourself: Create positive reinforcements every time you hit a milestone.

Clone Popular Sites: Build imitation sites. Start simple.

Inspect Code: Right click on pages and inspect how the code is compiled.

Take a Breather: Don’t forget to step away from your computer and let your brain retain information you learned and think of more efficient solutions to a problem.