

# Programming

- It is okay to do competitive programming as long as it motivates you.
- Programming is like writing. Programming is a creative activity.
- Do your own project. Write your technical blog. Program for a project that you will maintain for years, just like maintaining a 10-year-long blog.
- Usually, you do not need to pursue what's hot in technology now, as many trends will fade after a few years.
- Pursue your curiosity and program for your own sake.
- Try to create programs for yourself. They are not work assignments.
- If you feel unhappy often when programming, then you are doing it the wrong way.
- iOS, Android, Backend, Frontend, AI are all good. One can at least try to make a small project using them or learn about them for a few months.
- Debugging is about being suspicious. Do not trust every line of your code; you can think of a better way to do it.
- In programming, even a character or a line of log is important. They tell you something.
- Using programming makes products for others to use. It is interesting to have users.
- You do not need to be harsh. A few hundred users really loving your product is better than tens of thousands of users who just kindly like your product.
- Remember why you got into programming and never forget it.
- Apply knowledge in programming to every aspect of life. They are the same. Doing things in batches or one by one. How to separate jobs into units. The underlying tech behind every app. The nuanced details behind network requests.
- The abstraction and logical thinking. The detail-oriented thinking. The thinking out of every solution.
- Truth is truth. Usually, the computer won't be wrong. The electric circuit won't be wrong. The compiler won't be wrong. Don't feel upset when there is a bug.
- Pursue elegant and simple solutions. Simplicity is the ultimate sophistication. You need to think hard to leave what's essential and remove what's extra.
- For programming languages, the languages that get the work done are okay. I personally recommend Java and Python.
- Follow Yin Wang at <https://www.yinwang.org>. He is one of the few geniuses in programming, though he says geniuses never exist.
- The knowledge and principles of programming can be easily applied to language learning, hardware repairing, life hacking, and scientific researching.
- For most programming tasks, you don't need fancy math besides high school math.
- Reflect on your old code after years or maintain a code project for a long time. It will teach you a lot.
- If you lose your passion for programming, just do other things for some time.

- The timing of testing is important. Just do it naturally. You often don't need to write tests for your project. Try not to write tests, write unit tests, write integration tests, write API tests. Compare them wisely.
- Try AI code editors. Use ChatGPT or other chatbots often. As AI tools are easy to use now, you can focus on more creative or important parts.