

Programmierung

- Es ist in Ordnung, Competitive Programming zu betreiben, solange es dich motiviert.
- Programmieren ist wie Schreiben. Programmieren ist eine kreative Tätigkeit.
- Mach dein eigenes Projekt. Schreibe deinen technischen Blog. Programmiere für ein Projekt, das du über Jahre hinweg pflegen wirst, so wie das Führen eines 10-jährigen Blogs.
- Normalerweise musst du nicht dem hinterherjagen, was gerade in der Technologie ange sagt ist, da viele Trends nach ein paar Jahren verblassen werden.
- Folge deiner Neugier und programmiere für dich selbst.
- Versuche, Programme für dich selbst zu erstellen. Sie sind keine Arbeitsaufträge.
- Wenn du dich oft unglücklich fühlst, wenn du programmierst, dann machst du es falsch.
- iOS, Android, Backend, Frontend, AI – alles ist gut. Man kann zumindest versuchen, ein kleines Projekt damit zu machen oder sie ein paar Monate lang zu lernen.
- Debuggen bedeutet, misstrauisch zu sein. Vertraue nicht jeder Zeile deines Codes; du kannst dir einen besseren Weg ausdenken, es zu tun.
- Beim Programmieren ist sogar ein Zeichen oder eine Zeile im Log wichtig. Sie sagen dir etwas.
- Durch Programmieren entstehen Produkte, die andere nutzen können. Es ist interessant, Nutzer zu haben.
- Du musst nicht hart sein. Ein paar hundert Nutzer, die dein Produkt wirklich lieben, sind besser als zehntausende Nutzer, die dein Produkt nur freundlich mögen.
- Erinnere dich daran, warum du mit dem Programmieren angefangen hast, und vergiss es nie.
- Wende das Wissen aus dem Programmieren auf jeden Aspekt des Lebens an. Sie sind dasselbe. Dinge in Batches oder einzeln erledigen. Wie man Aufgaben in Einheiten unterteilt. Die zugrunde liegende Technik hinter jeder App. Die nuancierten Details hinter Netzwerkanfragen.
- Die Abstraktion und das logische Denken. Das detailorientierte Denken. Das Durchdenken jeder Lösung.
- Wahrheit ist Wahrheit. Normalerweise liegt der Fehler nicht beim Computer. Die elek trische Schaltung liegt nicht falsch. Der Compiler liegt nicht falsch. Sei nicht verärgert, wenn es einen Bug gibt.
- Strebe nach eleganten und einfachen Lösungen. Einfachheit ist die ultimative Raffinesse. Du musst hart nachdenken, um das Wesentliche zu behalten und das Überflüssige zu entfernen.
- Bei Programmiersprachen sind die Sprachen, die die Arbeit erledigen, in Ordnung. Ich

persönlich empfehle Java und Python.

- Folge Yin Wang auf <https://www.yinwang.org>. Er ist einer der wenigen Genies im Programmieren, obwohl er sagt, dass Genies nie existieren.
- Das Wissen und die Prinzipien des Programmierens können leicht auf das Sprachenlernen, die Hardware-Reparatur, das Life-Hacking und die wissenschaftliche Forschung angewendet werden.
- Für die meisten Programmieraufgaben brauchst du keine ausgefallene Mathematik außerhalb der Schulmathematik.
- Reflektiere deinen alten Code nach Jahren oder pflege ein Codeprojekt über lange Zeit. Es wird dir viel beibringen.
- Wenn du deine Leidenschaft für das Programmieren verlierst, mach einfach eine Zeit lang andere Dinge.
- Der Zeitpunkt des Testens ist wichtig. Mach es einfach natürlich. Oft musst du keine Tests für dein Projekt schreiben. Versuche, keine Tests zu schreiben, schreibe Unit-Tests, schreibe Integrationstests, schreibe API-Tests. Vergleiche sie weise.
- Probiere KI-Code-Editoren aus. Nutze ChatGPT oder andere Chatbots oft. Da KI-Tools jetzt einfach zu verwenden sind, kannst du dich auf kreativere oder wichtigere Teile konzentrieren.