

プログラミング

- ・競技プログラミングは、あなたのモチベーションを高める限り、やっても構いません。
- ・プログラミングは執筆のようなものです。プログラミングは創造的な活動です。
- ・自分のプロジェクトをやりましょう。技術ブログを書きましょう。10年間続けるブログのように、何年もメンテナンスするプロジェクトのためにプログラムを書いてください。
- ・通常、今ホットな技術を追いかける必要はありません。多くのトレンドは数年後に消え去ります。
- ・好奇心を追求し、自分のためにプログラミングをしてください。
- ・自分のためにプログラムを作ってみてください。それらは仕事の課題ではありません。
- ・プログラミングをしていると頻繁に不幸せを感じるなら、それは間違ったやり方でやっているということです。
- ・iOS、Android、バックエンド、フロントエンド、AIはすべて良いものです。少なくともそれらを使って小さなプロジェクトを作ったり、数ヶ月間学んだりすることができます。
- ・デバッグは疑うことです。コードの各行を信頼しないでください。もっと良い方法を考えることができます。
- ・プログラミングでは、1文字や1行のログも重要です。それらは何かを伝えています。
- ・プログラミングを使って他の人が使う製品を作ります。ユーザーがいることは面白いことです。
- ・厳しくする必要はありません。あなたの製品を本当に愛してくれる数百人のユーザーは、ただ優しく好きなだけの数万人のユーザーよりも良いです。
- ・プログラミングを始めた理由を覚えておいて、それを忘れないでください。
- ・プログラミングの知識を生活のあらゆる側面に適用してください。それらは同じです。物事をバッチでやるか一つずつやるか。仕事を単位に分ける方法。すべてのアプリの背後にある技術。ネットワーククリクエストの背後にある微妙な詳細。
- ・抽象化と論理的思考。細部にこだわる思考。すべての解決策を考えること。
- ・真実は真実です。通常、コンピュータは間違いません。電気回路は間違いません。コンパイラは間違いません。バグがあっても落ち込まないでください。
- ・エレガントでシンプルな解決策を追求してください。シンプルさは究極の洗練です。本質的なものを残し、余分なものを取り除くために一生懸命考える必要があります。
- ・プログラミング言語については、仕事をこなす言語なら何でも構いません。個人的にはJavaとPythonをお勧めします。
- ・<https://www.yinwang.org>のYin Wangをフォローしてください。彼はプログラミングの数少ない天才の一人ですが、天才は存在しないと言っています。
- ・プログラミングの知識と原則は、言語学習、ハードウェア修理、ライフハック、科学研究に簡単に適用できます。

- ・ほとんどのプログラミングタスクでは、高校数学以外の高度な数学は必要ありません。
- ・数年後に古いコードを振り返ったり、長期間コードプロジェクトをメンテナンスしたりしてください。それはあなたに多くのことを教えてくれます。
- ・プログラミングへの情熱を失ったら、しばらく他のことをしてください。
- ・テストのタイミングは重要です。自然にやってください。プロジェクトのためにテストを書く必要はありません。テストを書かないようにし、ユニットテストを書き、統合テストを書き、API テストを書いてみてください。それらを賢く比較してください。
- ・AI コードエディタを試してください。ChatGPT や他のチャットボットを頻繁に使ってください。AI ツールが使いやすくなった今、より創造的で重要な部分に集中できます。