

プログラミングを始めたばかりの過去の自分へ

あなたは数年後にこんな感じで遊び始めます

動画 OneDrive link; 40 分

1 基本編

1. Google 検索を極めなさい
 - あなたが直面している課題は、誰かがすでに解決しているかもしれません
2. AI を積極的に活用しなさい
3. キーボードショートカットを学びなさい
 - 壁を印刷したチートシートで埋めなさい
 - Ctrl + C の便利さを拡張しましょう

2 マニアック編 (プログラミングを仕事にしたい場合)

2.1 心がけ編

1. コンピュータの操作を英語で行いなさい
 - コンピュータは第一に日本人のために作られているわけではなさそうです
 - IELTS で国際的な機会を得ることができます 私の IELTS 学習アプリ
 - 現在カナダとオーストラリアは博士号と IELTS 7.0 以上ほどで永住権が取得できるかもしれません
2. マウスの使用を最小限に抑えなさい
 - なるべくキーボードのホームポジションから手を離さないでください
3. 技術書、ドキュメント、ブログ記事を読みなさい
 - 特に初期段階では、信頼のある技術書で大枠を把握することが役立ちます

2.2 ソフトウェア編

1. プログラミングには Linux を活用しなさい
 - この世界では避けて通ることはできません
 - Linux 教科書 LinuC レベル 1 の 3 冊を取り敢えず読めば良いと思います
 - Linux の設定を継続的に改善しましょう 私の dotfiles リポジトリ
2. 関数をたくさん作りなさい
 - 自動販売機のようなものです、ボタンを押せばコーラが出る、毎回毎回コーラが出る、自販機の中がどうなっていたか気にしなくて済みます
3. Emacs を学び、使用しなさい
 - 私が知らないだけで Emacs (宗教です) でなくてもよいです、キーボードから手を離したくない、引きこもりコードに集中したい、という精神です
 - チュートリアルから始めなさい M-x help-with-tutorial
 - Emacs を起動し、M-x help-with-tutorial RET (Meta (Alt or Esc) キー + x、その後 'help-with-tutorial' と入力し Enter)
 - SystemCrafters YouTube チャンネル
 - 流して見るだけで楽しいです
 - 私の Emacs 設定
 - 自分の設定を一から作ると何が起きているのか、どのようにプログラミングしたら良いのか、理解出来ると思います
4. 個人的なツールボックスを更新し続けなさい
 - 動く関数が引き出しのどこかにあるとは役に立ちます 私の Python ツールキット例
5. 独自のショートカットを作成し、最適化しなさい
 - 私の bash ショートカット
 - 私の Emacs ショートカット
6. GitHub を活用しなさい
 - 他人のために書きなさい。他人とは将来のあなたであり、他のマシン上でのあなたです。私の GitHub レポジトリ

2.3 道具編

1. 無刻印の HHKB キーボードを使用しなさい
 - 決してキーボードを見ることなくするので (意味がないので) 、どんなに細かい記号や修飾キーの組み合わせでも、感覚で覚えるようになります
 - 例えば左の手の小指の付け根で左下の Fn キーを押したりします
2. 一度はパソコンを組みなさい

- パーツを理解するためにとても有効です
- 電源、マザーボード、CPU、CPU クーラ、ストレージ、GPU、キーボード、マウスぐらいです

3 おわりに

コーディングを楽しんでください