Programming Advice to My Past Self Who Just Started

1 You'll start playing like this in a few years

Video OneDrive link; 40 minutes

2 Basics

1. Master Google Search
    - The challenges you face may have already been solved by someone
2. Actively utilize AI
3. Learn keyboard shortcuts
    - Cover your walls with printed cheat sheets
    - Extend the convenience of Ctrl + C

3 Advanced (If you want to make programming your career)

3.1 Mindset

1. Conduct all computer operations in English
    - Computers don't seem to be primarily designed for Japanese users
    - Gain international opportunities through IELTS My IELTS learning app
        - Currently, Canada and Australia may offer permanent residency with a Ph.D. and IELTS 7.0 or higher
2. Minimize mouse usage
    - Try not to move your hands from the keyboard's home position
3. Read technical books, documentation, and blog posts

- Especially in the early stages, grasping the big picture from reliable technical books is helpful

3.2 Software

1. Utilize Linux for programming

    - It's unavoidable in this world
    - I think reading the 3 books of Linux Textbook LinuC Level 1 is a good start
    - Continuously improve your Linux configuration My dotfiles repository

2. Create lots of functions

    - They're like vending machines: press a button, get a cola, every time, without worrying about what's inside

3. Learn and use Emacs

    - It doesn't have to be Emacs (it's a religion) as far as I know, it's about not wanting to take your hands off the keyboard and focusing on code
    - Start with the tutorial `M-x help-with-tutorial`
        - Launch Emacs and enter `M-x help-with-tutorial RET` (Meta (Alt or Esc) key + x, then type 'help-with-tutorial' and press Enter)
    - SystemCrafters YouTube channel
        - It's fun just to watch
    - My Emacs configuration
        - Creating your own configuration from scratch helps you understand what's happening and how to program

4. Continuously update your personal toolbox

    - Having working functions somewhere in your drawer is useful Example of my Python toolkit

5. Create and optimize your own shortcuts

    - My bash shortcuts
    - My Emacs shortcuts

6. Make full use of GitHub

    - Write for others. Others include your future self and yourself on other machines. My GitHub repository

3.3 Tools

1. Use a blank HHKB keyboard

   - You'll never look at the keyboard (as it's meaningless), so you'll learn to feel even the most intricate symbol and modifier key combinations
   - For example, you'll press the Fn key in the bottom left with the base of your left pinky finger

2. Build a PC at least once

   - It's very effective for understanding the components
   - It's mainly about power supply, motherboard, CPU, CPU cooler, storage, GPU, keyboard, and mouse

4 In Conclusion

Enjoy coding!