Coding Horror: We Are Typists First, Programmers Second

Coding Horror
programming and human factors
by Jeff Atwood

Nov 17, 2008

We Are Typists First, Programmers Second

Remember last week when I said coding was just writing?

I was wrong. As one commenter noted, it's even simpler than that.

> [This] reminds me of a true "Dilbert moment" a few years ago, when my (obviously non-technical) boss commented that he never understood why it took months to develop software. "After all", he said, "it's just typing."

Like broken clocks, even pointy-haired managers are right once a day. **Coding is just typing.**

So if you want to become a great programmer, start by becoming a great typist. Just ask Steve Yegge.

> I can't understand why professional programmers out there allow themselves to have a career without teaching themselves to type. It doesn't make any sense. It's like being, I dunno, an actor without knowing how to put your clothes on. It's showing up to the game unprepared. It's coming to a meeting without your slides. Going to class without your homework. Swimming in the Olympics wearing a pair of Eddie Bauer Adventurer Shorts.
>
> Let's face it: it's *lazy*.
>
> There's just no excuse for it. There are no excuses. I have a friend, John, who can only use one of his hands. He types 70 wpm. He invented his own technique for it. He's not making excuses; he's typing circles around people who are making excuses.

I had a brief email exchange with Steve back in March 2007, after I wrote Put Down The Mouse, where he laid that very same Reservoir Dogs quote on me. Steve's followup blog post was a very long time in coming. I hope Steve doesn't mind, but I'd like to pull two choice quotes directly from his email responses:

> I was trying to figure out which is the most important computer science course a CS student could ever take, and eventually realized it's Typing 101.
>
> The really great engineers I know, the ones who build great things, they can type.

Strong statements indeed. I concur. **We are typists first, and programmers second.** It's very difficult for me to take another programmer seriously when I see them using the hunt and peck typing techniques. Like Steve, I've seen this far too often.

First, a bit of honesty is in order. Unlike Steve, I am a completely self-taught typist. I didn't take any typing classes in high school. Before I wrote this blog post, I realized I should check to make sure I'm not a total hypocrite. So I went to the first search result for typing test and gave it a shot.

typing test speed (WPM) results

I am by no means the world's fastest typist, though I do play a mean game of Typing of the Dead. Let me emphasize that *this isn't a typing contest*. I just wanted to make sure I wasn't full of crap before I posted this. I know, there's a first time for everything. Maybe this'll be the start of a trend. Doubtful, but you never know.

Steve and I believe there is nothing more fundamental in programming than the ability to efficiently express yourself through typing. Note that I said "efficiently" not "perfectly". This is about **reasonable competency at a core programming discipline**.

Maybe you're not convinced that typing is a core programming discipline. I don't blame you, although I do reserve the right to wonder how you manage to program without using your keyboard.

Instead of answering directly, let me share one of my (many) personal foibles with you. At least four times a day, I walk into a room having *no idea* why I entered that room. I mean no idea whatsoever. It's as if I have somehow been teleported into that room by an alien civilization. Sadly, the truth is much less thrilling. Here's what happened: in the brief time it took for me to get up and move from point A to point B, I have totally forgetten whatever it was that motivated me to get up at all. Oh sure, I'll rack my brain for a bit, trying to remember what I needed to do in that room. Sometimes I remember, sometimes I don't. In the end, I usually end up making multiple trips back and forth, remembering something else I *should* have done while I was in that room after I've already left it.

It's all quite sad. Hopefully your brain has a more efficient task stack than mine. But I don't fault my brain -- I fault my body. It can't keep up. If I had arrived faster, I wouldn't have had time to forget.

What I'm trying to say is this: *speed matters*. **When you're a fast, efficient typist, you spend less time between thinking that thought and expressing it in code.** Which means, if you're me at least, that you might actually get *some* of your ideas committed to screen before you completely lose your train of thought. Again.

Yes, you should think about what you're doing, obviously. Don't just type random gibberish as fast as you can on the screen, unless you're a Perl programmer. But all other things being equal -- and they never are -- the touch typist *will* have an advantage. The best way to become a touch typist is through typing, and lots of it. A little research and structured practice couldn't hurt either. Here are some links that might be of interest to the aspiring touch typist:

- Type Racer
- Typer Shark
- Dvorak, Keyboard Layout of Champions
- Colemak keyboard layout
- TyperA
- Das Keyboard with blank keys
- The Typing of the Dead (for PC)
- Put Down That Mouse. Seriously. It's a crutch.
- Typingmania (warning, Japanophiles only)

(But this is a meager and incomplete list. What tools do *you* recommend for becoming a better typist?)

There's precious little a programmer can do without touching the keyboard; it is the primary tool of our trade. I believe in practicing the fundamentals, and **typing skills are as fundamental as it gets for programmers.**

Hail to the typists!