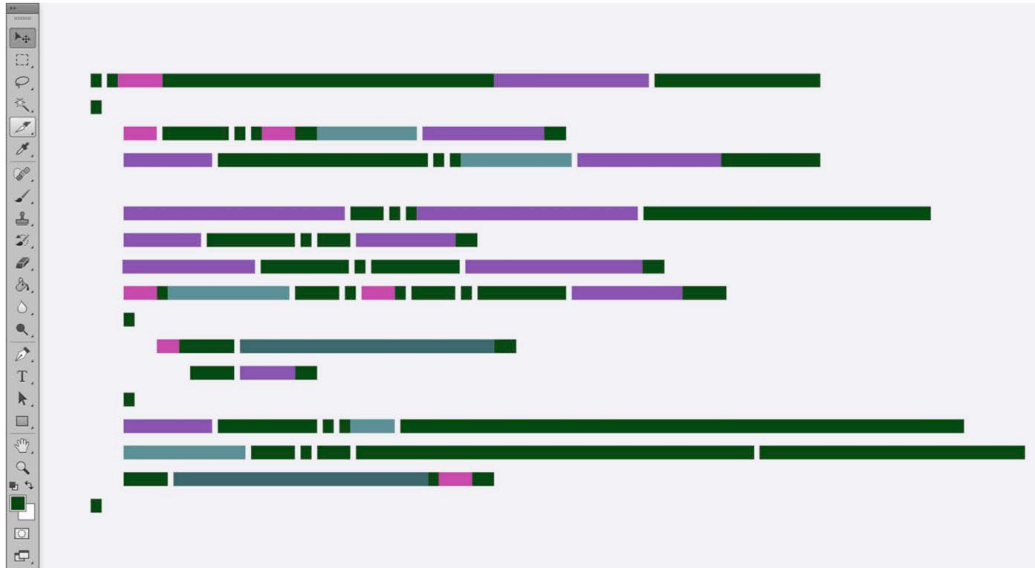


Designers & Code

<https://medium.com/@bleeharper/designers-code-c7077ee50780#.9oeyvdkgk>



Brian Harper [Follow](#)
Designer at Workday
Oct 30, 2013 · 3 min read



xdobe Photocode cc

The idea that everyone can benefit from learning to code has gained mainstream popularity over the past couple years. Politicians and billionaires support [Code.org](#). Online courses are [popping up everywhere](#). Somebody even tried to teach a homeless man to make a living by coding and [wrote about it](#). However, all this buzz has caused backlash in the form of [blog posts warning people](#) to hold their horses and think about what they're getting into. As a designer interested in learning to code, I still believe learning more about code will be a key to my success or failure as a mobile app designer in the coming years.

The common argument made against learning to code is one of opportunity cost. "Why learn to code when you could spend time on literally anything else?" That argument can be applied to any investment of time and doesn't carry a whole lot of weight for me. The bottom line behind all these counter-opinions is that learning to code is an enormous endeavor that won't necessarily lead to fame and fortune. That's a fair point for weeding out the get-rich-quick crowd, but it shouldn't deter anyone who wants to design and create interactive experiences.

Traditionally, mobile app designers deliver documentation (comps & specs) to developers, who take care of implementation and iteration. This process can work well but often results in an end product that diverges from the complete vision in a designer's head. As designers, we can certainly improve our designs by learning more about usability, copywriting, research, marketing, and psychology instead of learning to code. But no matter how good those initial comps and specs are, they need to be implemented correctly and iterated on continuously to be useful to the end user. I think the best way to make sure that happens is by empowering designers to implement and iterate on design decisions alongside developers. That doesn't necessarily mean writing production-quality code from scratch. Being able to understand the relative costs of different design executions, tweak design parameters (colors, type styles, and coordinates), and push changes using version control is enough to empower a designer to take total responsibility for their work.

Web design as a discipline seems to be years ahead of native app design in this respect. Most web designers today are expected to implement or edit their designs in HTML/CSS. I think native iOS app designers will be expected to use UIStoryboards/Interface Builder to implement their designs in the very near future.

We don't need to spend a lifetime learning to code at the expense of getting better at our own discipline, but I think Marc Edwards put it best [when he said](#), "I have never, ever known a designer to be worse off for knowing more about code."

I'm presenting on this topic and more at [CocoaConf Atlanta on November 15, 2013](#). Hope to see you there or on [twitter](#).