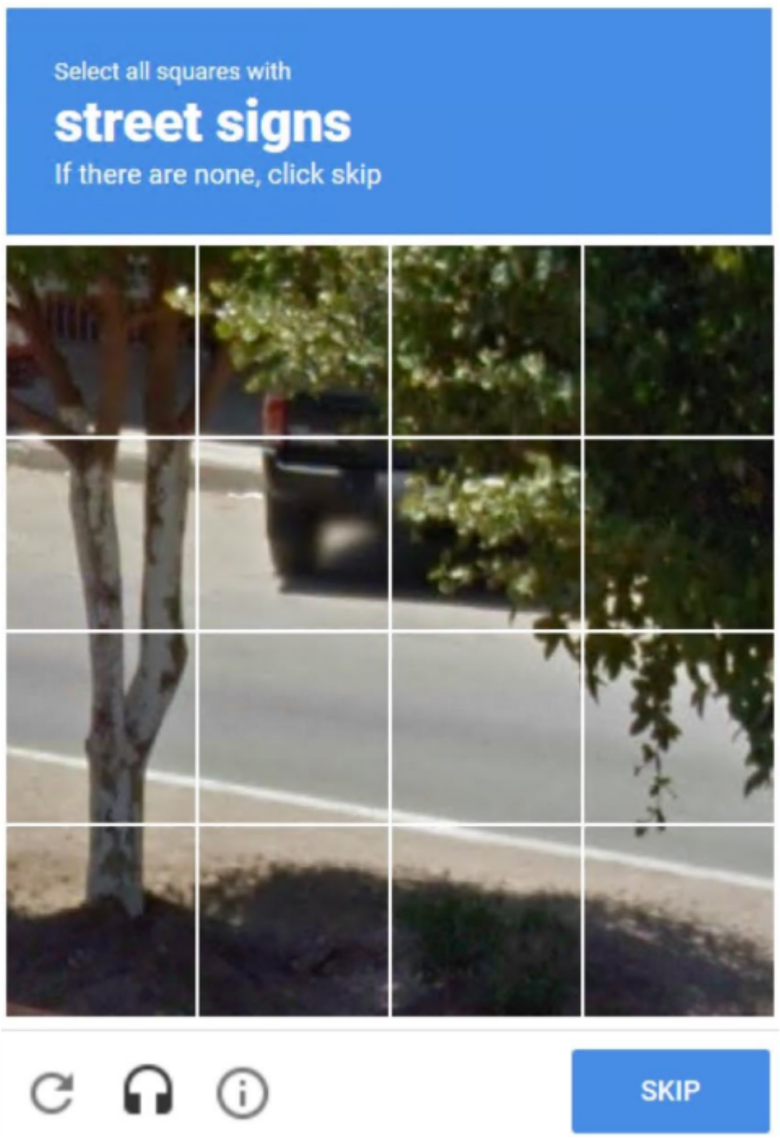

Best Mac Pro Apps 2014

How fast does your MacBook need to be to comfortably code iOS apps with Xcode? Is a MacBook Pro from 2-3 years ago good enough to learn Swift programming? Let’s find out!

Apple puts an immense focus on Education for students and it reflects on the apps they create and provide for Students. It is important for a student to keep up with the pace of the changing world of education in order to get good grades as well as proper adequate knowledge. With more and more things in the world becoming associated with the computer it is absolutely essential to have an idea. Explore the world of Mac. Check out MacBook Pro, iMac Pro, MacBook Air, iMac, and more. Visit the Apple site to learn, buy, and get support.

Here’s what we’ll get into:

- The minimum/recommended system requirements for Xcode 11
- Why you need – or don’t need – a fancy $3.000 MacBook Pro
- Which second-hand Macs can run Xcode OK, and how you can find out

I’ve answered a lot of *“Is my MacBook good enough for iOS development and/or Xcode?”*-type questions on Quora. A few of the most popular models include:

- The 3rd- and 4th-gen MacBook Pro, with 2.4+ GHz Intel Core i5, i7, i9 CPUs
- The 2nd-gen MacBook Air, with the 1.4+ GHz Intel Core i5 CPUs
- The 4th-generation iMac, with the 2.7+ GHz Intel Core i5 and i7 CPUs

These models aren’t the latest, that’s for sure. Are they good enough to code iOS apps? And what about learning how to code? We’ll find out in this article.

2012 Mac Pro For Sale

My Almost-Unbreakable 2013 MacBook Air

Since 2009 I’ve coded more than 50 apps for iOS, Android and the mobile web. Most of those apps, including all apps I’ve created between 2013 and 2018, were built on a 13” MacBook Air with 8 GB of RAM and a 1.3 GHz Intel i5 CPU.

My first MacBook was the gorgeous, then-new MacBook White unibody (2009), which I traded in for a faster but heavier MacBook Pro (2011), which I traded in for that nimble workhorse, the mighty MacBook Air (2013). In 2018 I upgraded to a tricked out 13” MacBook Pro, with much better specs.

Frankly, that MacBook Air from 2013 felt more sturdy and capable than my current MacBook Pro. After 5 years of daily intenstive use, the MacBook Air’s battery is only through 50% of its max. cycle count. It’s still going strong after 7 hours on battery power.

In 2014, my trusty MacBook Air broke down on a beach in Thailand, 3 hours before a client deadline, with the next Apple Store 500 kilometer away. It turned out OK, of course. Guess what? My current MacBook Pro from 2018, its keyboard doesn’t even work OK, I’ve had sound recording glitches, and occasionally the T2 causes a kernel panic. Like many of us, I wish we had 2013-2015 MacBook Air’s and Pro’s with today’s specs. Oh, well…

Mac Pro 2014

That 100 Mhz i486 PC I Learned to Code With

When I was about 11 years old I taught myself to code in BASIC, on a 100 Mhz i486 PC that was given to me by friends. It had a luxurious 16 MB of RAM, initially only ran MS-DOS, and later ran Windows 3.1 and ’95.

A next upgrade came as a 400 Mhz AMD desktop, given again by friends, on which I ran a local *EasyPHP* webserver that I used to learn web development with PHP, MySQL and HTML/CSS. I coded a mod for Wolfenstein 3D on that machine, too.

We had no broadband internet at home back then, so I would download and print out coding tutorials at school. At the one library computer that had internet access, and I completed the tutorials at home. The source codes of turn-based web games, JavaScript

tidbits and HTML page snippets were carried around on a 3.5” floppy disk.

Later, when I started coding professionally around age 17, I finally bought my first laptop. My own! I still remember how happy I was. I got my first gig as a freelance coder: creating a PHP script that would aggregate RSS feeds, for which I earned about a hundred bucks. Those were the days!

Xcode, iOS, Swift and The MacBook Pro

The world is different today. Xcode simply doesn’t run on an i486 PC, and you can’t save your app’s source code on a 1.44 MB floppy disk anymore. Your Mac probably doesn’t have a CD drive, and you store your Swift code in a cloud-based Git repository somewhere.

Make no mistake: owning a MacBook is a luxury. Not because learning to code was harder 15 years ago, and not because computers were slower back then. It’s because kids these days learn Python programming on a $25 Raspberry Pi.

Best Mac Pro Apps 2014 Software

I recently had a conversation with a young aspiring coder, who complained he had no access to “decent” coding tutorials and mentoring, despite owning a MacBook Pro and having access to the internet. Among other things, I wrote the following:

You’re competing with a world of people that are smarter than you, and have better resources. You’re also competing against coders that have had it worse than you. They didn’t win despite adversity, but because of it. Do you give up? NO! You work harder. It’s the only thing you can do: work harder than the next person. When their conviction is wavering, you dig in your heels, you keep going, you persevere, and you’ll win.

Winning in this sense isn’t like winning a race, of course. You’re not competing with anyone else; you’re only really up against *yourself*. If you want to learn how to code, don’t dawdle over choosing a $3.000 or a $2.900 laptop. If anything, it’ll keep you from developing the grit you need to learn coding.

Great ideas can change the world, but only if they’re accompanied by deliberate action. Likewise, simply complaining about adversity isn’t going to create opportunities for growth – unless you take action. I leapfrogged my way from one hand-me-down computer to the next. I’m not saying you should too, but I do want to underscore how it helped me develop character.

If you want to learn how to code, welcome adversity. Be excellent because of it, or despite it, and never give up. Start coding today! Don’t wait until you’ve got all your ducks in a row.

Which MacBook is Fast Enough for Xcode 11?

The recommended system specs to run Xcode 11 are:

- A Mac with macOS Catalina (10.15.2) for Xcode 11.5 or macOS Mojave (10.14.4) for Xcode 11.0 (see alternatives for PC here)
- At least an Intel i5- or i7-equivalent CPU, so about 2.0 GHz should be enough
- At least 8 GB of RAM, but 16 GB lets you run more apps at the same time
- At least 256 GB disk storage, although 512 GB is more comfortable
- You'll need about 8 GB of disk space, but Xcode's intermediate files can take up to 10-30 GB of extra disk space

Looking for a second-hand Mac? The following models should be fast enough for Xcode, but YMMV!

- 4th-generation MacBook Pro (2016)
- 3rd-generation Mac Mini (2014)
- 2nd-generation MacBook Air (2017)
- 5th-generation iMac (2015)

When you're looking for a Mac or MacBook to purchase, make sure it runs the latest version of macOS. Xcode versions you can run are tied to macOS versions your hardware runs, and iOS versions you can build for are tied to Xcode versions. See how that works? This is especially true for SwiftUI, which is iOS 13.0 and up *only*. Make sure you can run the latest!

Pro tip: You can often find the latest macOS version a device model supports on their Wikipedia page (see above links, scroll down to *Supported macOS releases*). You can then cross-reference that with Xcode's minimum OS requirements (see here, scroll to *min macOS to run*), and see which iOS versions you'll be able to run.

Further Reading

Best Mac Pro Apps 2014 Pro

Awesome! We've discussed what you need to run Xcode on your Mac. You might not need as much as you think you do. Likewise, it's smart to invest in a future-proof development machine.

Best Mac Pro Apps 2014 Download

Whatever you do, don't ever think you need an expensive computer to learn how to code. Maybe the one thing you *really* want to invest in is *frustration tolerance*. You can make do, without the luxury of a MacBook Pro. A hand-me-down i486 is enough. Or… is it?

Want to learn more? Check out these resources:

Learn how to build iOS apps

Get started with iOS 14 and Swift 5

Sign up for my iOS development course, and learn how to build great iOS 14 apps with Swift 5 and Xcode 12.