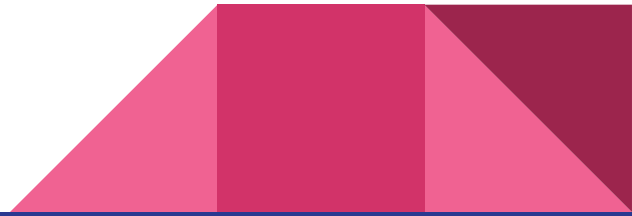
Four Ways To Create An App

For iOS or Android

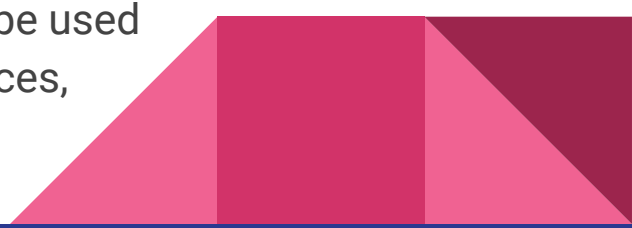
Overview: Four Ways to Create an App

1. Create a **web app** using HTML, CSS, and JavaScript.
2. Create a web app using HTML, CSS, and JavaScript, then use PhoneGap or similar technologies to **export** native code for iOS, Android, and other operating systems.
3. Create a **native app for iOS** using the Xcode tool and the Swift programming language.
4. Create a **native app for Android** using the Java programming language and the Android Developer Toolkit (ADT).



Native vs. Non-native

- **Native** apps are compiled to machine code for the particular processor in the device. Apple code cannot run on an Android device; they use different processors. Android code cannot run on an Apple device. They use different machine code. Native code is compiled by an environment like Xcode, Eclipse, or ADT.
- Examples of native code include Swift code compiled to run on Apple devices, and Java code compiled to run on Android devices.
- **Non-native** code is written in a generic language, and the code is interpreted or compiled on the device.
- Examples of non-native code include JavaScript, which can be used to create web pages and apps that can be run on Apple devices, Android devices, and other devices.



Advantages and Disadvantages

- **Native** code generally runs *more quickly* and creates the most pleasing user interface.
- But native code must be *re-written for each different device* you wish to deploy to. (That is, you need to write code for an Apple version using Xcode, then write completely different code for Android using Java. The code may do the same thing, but it looks entirely different, and the different versions may even need to be written by entirely separate teams.
- **Non-native** code can be written *just once,* then deployed to various kinds of devices, including Apple, Android, and others.
- Non-native code generally *runs a little slower,* and the user interface may be clunky. For example, the Apple version may look a little like Android, which may turn off Apple users; or the Android version may look a little like Apple, with similar reactions.

Method 1: Web Page

- **PRO.** A web page is easy to write using HTML, CSS, and JavaScript. We learn all these in this class.
- **PRO.** A web page can be viewed on any device: Apple, Android, Blackberry, Windows mobile, even laptop and desktop computers. So you can run the app anywhere.
- **CON.** Web apps require an internet connection in order to run them.
- **CON.** Web apps cannot be downloaded to your device, but they need to be run on the web.
- **CON.** A web page is a non-native solution, so it has some of the disadvantages of the non-native apps: slower, more generic user interface.

Method 2: Phone Gap

- Phone Gap is an Adobe technology that allows you to compile your web page code (HTML, CSS, and JavaScript) to native code targeted at each platform you wish to target. (“Platform” means a particular kind of device like Apple or Android.)
- You write your code in HTML, CSS, and JavaScript, which you probably already know.
- You upload your code to Adobe, and their computers compile to all the different platforms. It can compile many different platforms at the same time.
- People can run your application by connecting to an Adobe link that you can send to them in email. You don’t need to publish your app in any app store.
- If you wish to compile for Apple, you must have an Apple Developer account. I think you don’t have to pay for this account, but you do have to register for it.

Method 3: Xcode and Swift

- You can use Xcode and Swift to create apps for Mac, iPhone, iPad, iPod (touch), Apple Watch, Apple TV and other future devices.
- You cannot use the Xcode tool to create Android apps.
- The Xcode tool creates native apps that run quickly and look nice on Apple devices.
- You don’t need a developer account to write Swift code and view it running in a simulator on your computer.
- If you wish to download your app to run on your own iPhone or iPad, you need a (free) Apple developer account.

Method 4: Eclipse or ADT and Java

- You can use the various Android development tools to create Android native apps. These apps will not run on Apple devices.
- The resulting apps are native code and run quickly on Android devices. They also have the Android “look and feel.”
- You don’t need any developer license to write Android code, view it in an emulator, or even download the app to your phone.
- But you need a developer account to publish your app in the Google Play Store. This account does cost some money but it is not expensive.