

# Mobile Programming (Part 1)

Copyright © 2022 by  
Robert M. Dondero, Ph.D.  
Princeton University

# Objectives

- We will cover:
  - Mobile programming
  - Android mobile programming

## Using the Example Apps

- If you want to:
  - **Build** the example apps on your local computer, or
  - **Run** the example apps on an **Android emulator** on your local computer
- Then:
  - Download and install **Android Studio**
    - Browse to <https://developer.android.com/studio/index.html>
    - Follow the instructions

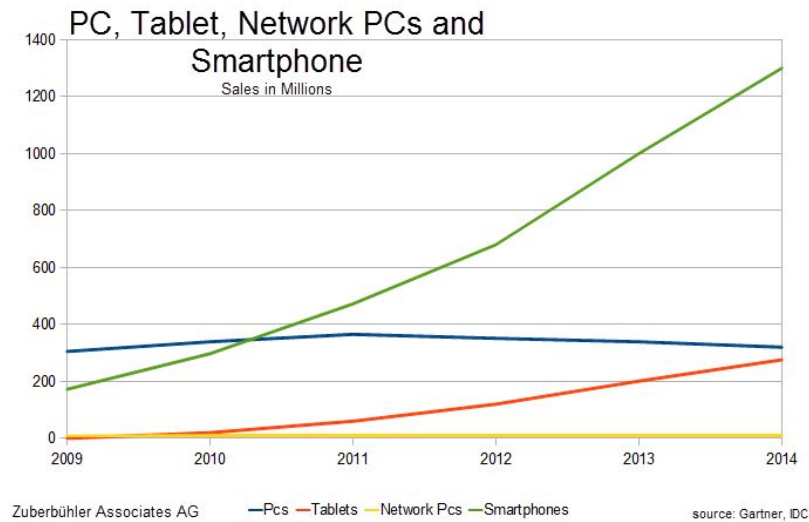
# Agenda

- **Mobile programming**
- Native mobile programming
- Android virtual devices
- HelloAndroid app

# Mobile Pgmming: Motivation

- **Question:**
  - Why study mobile programming?
- **Answer:** ...

# Mobile Pgmming: Motivation



## Mobile Pgmming: Options

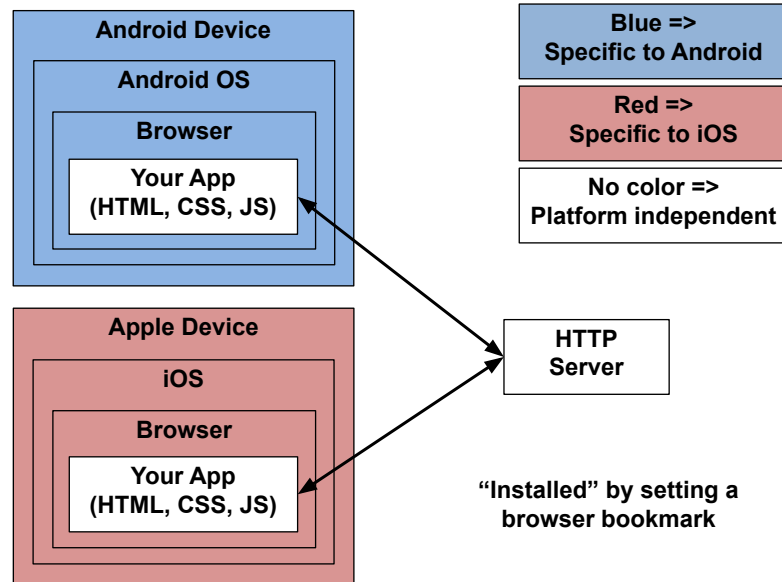
- **Question:**

- For mobile programming, what are my options?

- **Answer:**

- (1) Mobile web app (hereafter: web app)
- (2) Native mobile app (hereafter: native app)
- (3) Hybrid mobile app (Appendix)

# Mobile Pgmming: Web Apps

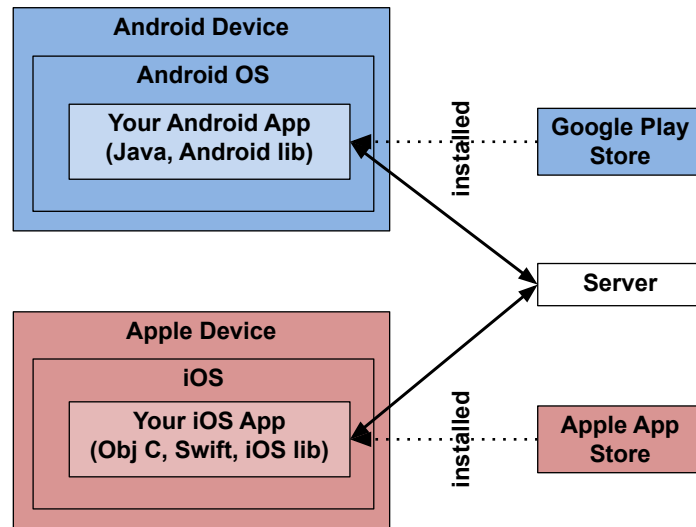




# Mobile Pgmming: Web Apps

- **Web app**
  - Runs in mobile device browser
  - Discovered by browsing/searching the web
  - “Installed” by setting browser bookmark
  - Device independent

# Mobile Pgmming: Native Apps



# Mobile Pgmming: Native Apps

- **Native app**

- Discovered by browsing/searching app store (Google Play Store, Apple App Store, ...)
- Installed through app store
- Device dependent
  - Android, iOS, Windows Phone, Blackberry, ...

# Mobile Pgmming: Native Apps

- Native apps use:
  - Native app library/framework
    - For Android: **Android SDK**
    - For iOS: **iOS SDK**
- Native apps typically are composed using:
  - Native app IDE
    - For Android: **Android Studio**
    - For iOS: **Xcode**

## Mobile Pgmming: Which Option?

- Which option is right for your app?
  - **Step 1:** Consider whether you have an option!

13

Mobile Pgmming: Which option?

Which option is right for your app?

**Step 1:** Consider whether you have an option!

A **native** app can support the required app features

Can a **web** app support the required app features?

## Mobile Pgmming: Device Features

- See <https://whatwebcando.today/>
- Examples:
  - **Offline mode:** yes
  - **Audio & video capture:** probably yes
  - **Proximity sensors:** probably no
  - **Contacts:** no

14

### Mobile Pgmming: Device Features

[see slide]

Examples:

Seamless experience | offline mode: yes  
Camera & microphone | audio & video capture: probably yes  
Location & position | proximity sensors: probably no  
Operating system | contacts: no

## Mobile Pgmming: Which Option?

- Which option is right for your app?
  - **Step 1:** Consider whether you have an option!
  - **Step 2:** Consider the broader context...

# Mobile Pgmming: Broader Context

Factor	Web App	(Multiple) Native Apps
Easy discoverability	✓	
Native look & feel		✓
Good performance (speed)		✓
Easy installation	✓	
Low development cost	✓	
Low maintenance cost	✓	
Few content restrictions, easy approval process, low/no fees	✓	

<https://www.nngroup.com/articles/mobile-native-apps/>

16

## Mobile Pgmming: Broader Context

[see slide]

To avoid the Apple approval process, some COS 333 teams have delivered native iOS applications via TestFlight, a Mac testing framework



## Mobile Pgmming: Commentary

- On **desktop** and **laptop** computers
  - Web apps and native apps are complementary
  - Web apps are becoming more attractive
- On **mobile** devices
  - Same!

# Agenda

- Mobile programming
- **Native mobile programming**
- Android virtual devices
- HelloAndroid app

# Native Mobile Pgmning

- Previously covered (CSS lecture, Asgt 4):
  - Responsive web apps, mobile web apps
- Covered now:
  - (Minimal) native apps
  - More specifically, Android apps...

# Native Mobile Pgmming

- **Question:**

- Why study **Android** programming?
- ... Instead of iOS programming?

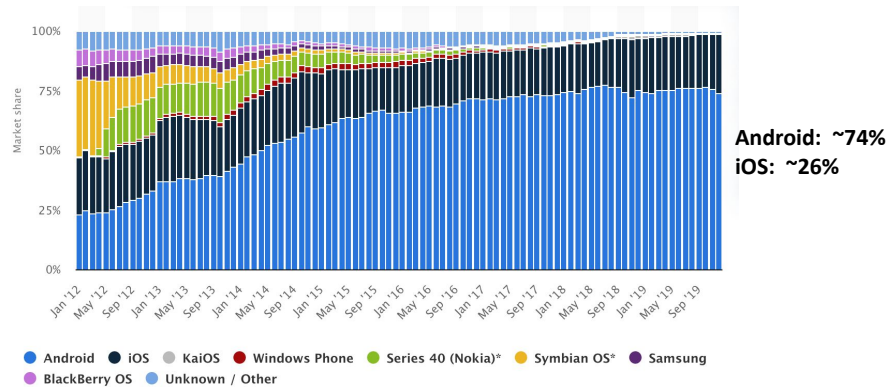
- **Answer 1:**

- Android programming is in **Java**
  - Which we know thoroughly
  - So we need not cover another pgmming lang (Objective-C, Swift, ...)

# Native Mobile Pgmming

## Answer 2:

Mobile operating systems' market share worldwide



<https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/>

21

## Native Mobile Pgmming

[see slide]

In many parts of the world, Android phones are the only computer

# Agenda

- Mobile programming
- Native mobile pgmming
- **Android virtual devices**
- HelloAndroid app

## Android Virtual Devices

- To test an Android app...
- Option 1:
  - Use an Android **device**
  - Pro: Fast
- Option 2:
  - Create an Android **virtual device**
  - Run it (on any computer) using the Android **emulator**
  - Pro: Convenient

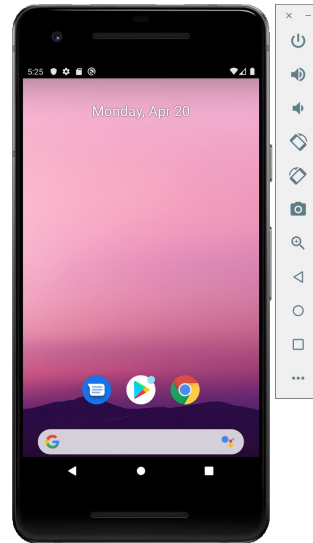
## Android Virtual Devices: Creating

- Creating an Android virtual device
  - Browse to  
<https://developer.android.com/training/basics/firstapp/running-app.html>
  - Follow the instructions
    - Performed within Android Studio
    - I created a device named Pixel\_2\_API\_29



## Android Virtual Devices: Running

- See **AndroidSetupMac**
- See **AndroidSetupLinux**
- See **AndroidEmulate**



# Agenda

- Mobile programming
- Native mobile programming
- Android virtual devices
- **HelloAndroid app**

26

Agenda

My Android programming story.

# HelloAndroid App: Prelim

- **Resource**
  - A data (non-code) asset of an app
- **Resource file**
  - A file defining resources of a specific kind

# HelloAndroid App: Prelim

- Resource file advantages
  - Factors data out of code
  - Can update data without modifying code
  - Can provide multiple resource files for each kind of resource

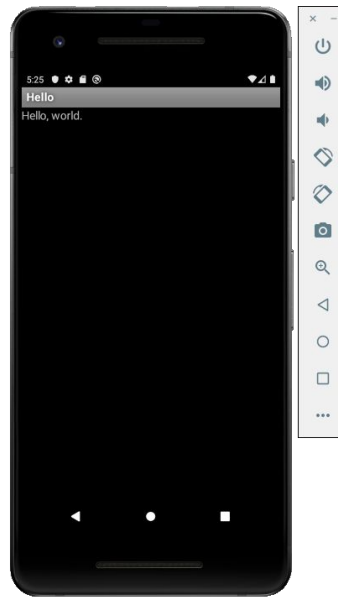
# HelloAndroid App: Prelim

- Resource file examples
  - For strings:
    - English phrases used in your code
    - French phrases used in your code
  - For screen layouts
    - Screen layouts for laptop browser
    - Screen layouts for smartphone browser
- Resource concept is not specific to Android!

# HelloAndroid App

- HelloAndroid app
  - The job
    - Simplest possible Android app
    - One screen
    - Displays “hello, world”

# HelloAndroid App



# HelloAndroid App

- General approach:
  - Develop without using Android studio
  - Develop from command line



# HelloAndroid App: Defining

```
HelloAndroid/  
  AndroidManifest.xml  
  res/  
    layout/  
      activity_main.xml  
    values/  
      values.xml  
  src/  
    edu/  
      princeton/  
        hello/  
          MainActivity.java
```

33

## HelloAndroidApp: Defining

Code notes: values.xml

- A resource file

- Factors strings out of Java code

- Facilitates internationalization

- Relates key “appLabel” to string “Hello”

- Relates key “textViewHint” to string “Hello, world.”

Code notes: activity\_main.xml

- A resource file

- Factors widget creation and layout out of Java code

- Facilitates changing layouts

- The layout for the MainActivity page is a vertical LinearLayout

  - “A layout that arranges other views either horizontally in a single column or vertically in a single row.”

  - Other ViewGroup classes: AbsoluteLayout, FrameLayout, GridLayout, RelativeLayout, SlidingDrawer, Toolbar, TextView, ...

- The MainActivity page contains a TextView object

  - “A user interface element that displays text to the user.”

  - Other View classes: Button, CheckBox, RadioButton, Switch, ToggleButton, AnalogClock, CalendarView, ...

Code notes: MainActivity.java

- Defines class MainActivity to extend Activity

“An activity is a single, focused thing that the user can do. Almost all activities interact with the user, so the Activity class takes care of creating a window for you in which you can place your UI with setContentView(View).”

Code notes: AndroidManifest.xml

- Describes the app as a whole

- The app's name is “Hello”

- Note use of resource

- Name is displayed in the mobile device window

- The app has one Activity

- Its name is edu.princeton.hello.MainActivity

## HelloAndroid App: Building

- See AndroidSetupLinux
- See AndroidSetupMac
- See **AndroidBuild**
  - **Dalvik** is the bytecode format that the **ART** (Android Runtime) understands
  - **APK** (Android Package Kit) is the package file format used for distribution and installation of Android apps

# HelloAndroid App: Building

- Resulting directory structure:

```
HelloAndroid/  
  AndroidManifest.xml  
  app.apk  
  app.apk.unaligned  
  classes.dex  
  res/  
    layout/  
      activity_main.xml  
    values/  
      values.xml
```

## Via command-line:

8 directories  
14 files  
~86 KB

## Via Android Studio:

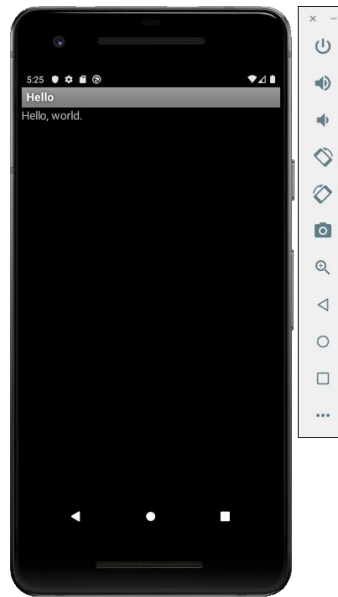
392 directories  
1270 files  
~22.4 MB

```
src/  
  edu/  
    princeton/  
      hello/  
        MainActivity.java  
        MainActivity.class  
        R.class  
        R.java  
        R$attr.class  
        R$id.class  
        R$layout.class  
        R$string.class
```

## HelloAndroid App: Installing

- **Option 1:** Plug mobile device into your development computer's USB port
- **Option 2:** Run virtual device on emulator on your development computer
- See **AndroidInstall**
  - Use *adb* (Android Debug Bridge) to install the app on the mobile device or emulator and run the app

# HelloAndroid App: Result



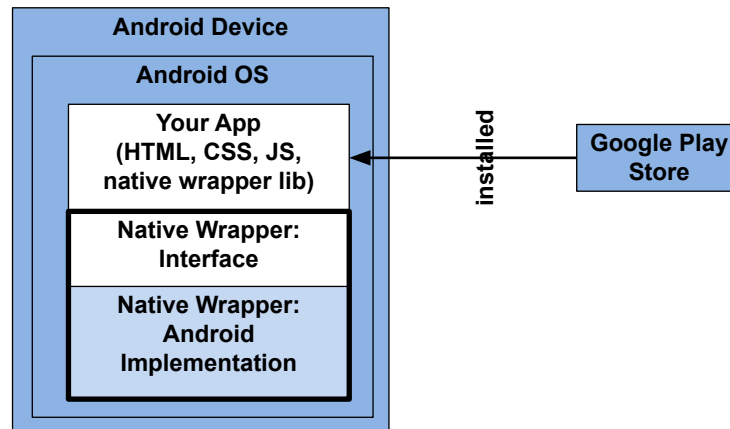
# Summary

- We have covered:
  - Mobile programming
  - Android mobile programming
    - Resources
- See also:
  - **Appendix:** Hybrid Mobile Apps

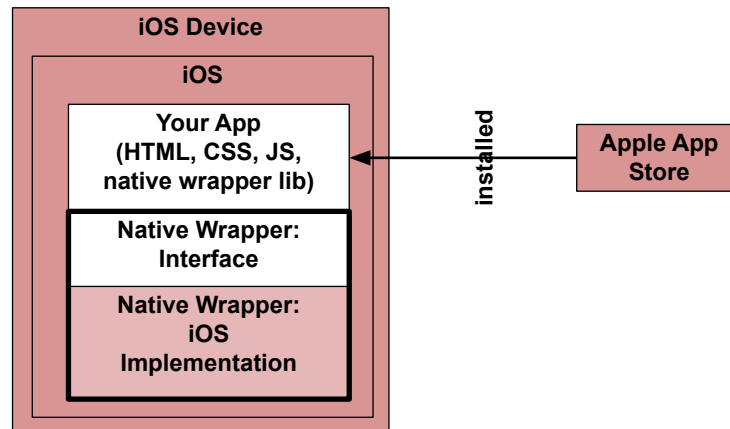
## Appendix: Hybrid Mobile Apps



# Hybrid Apps



# Hybrid Apps



# Hybrid Apps

- **Hybrid app**
  - Your app + native wrapper
    - Your app: runs in a **WebView**, a minimal web browser
    - Native wrapper: allows your app to use native features
  - Discovered by searching app store (Google Play Store, Apple App Store)
  - Installed via app store
  - [Your app + native wrapper] is platform dependent
  - [Your app] is platform independent

# Hybrid Apps

- Hybrid apps are typically composed using:
  - HTML5, CSS, JavaScript
  - Native wrapper
    - Examples: *Apache Cordova*, *PhoneGap*
  - Hybrid app framework
    - Examples: *Ionic* (used with AngularJS), *NativeScript* (used with AngularJS or Vue), *Framework7*

# Hybrid Apps

- Hybrid apps are sometimes composed using:
  - ***React Native***
    - App written in HTML, CSS, JavaScript, React
    - App does not render to WebView
    - App renders directly to Android/iOS components
    - Native web app???
      - Commentary: Not really; a JavaScript interpreter is involved