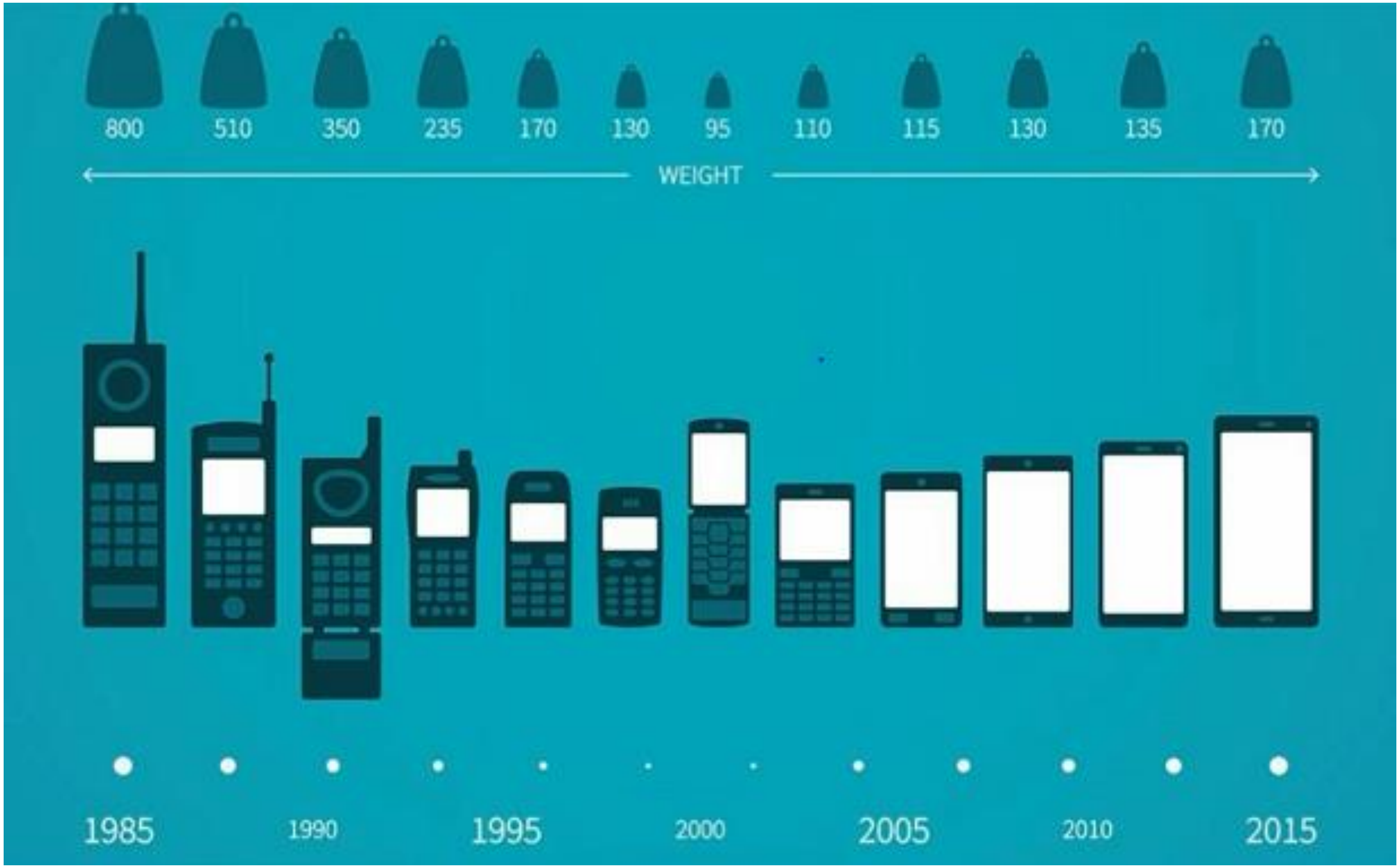
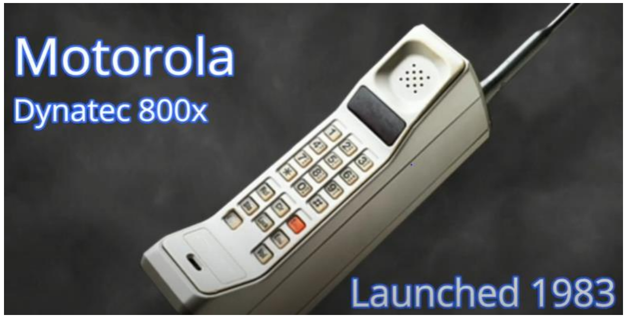
Basic of Android Programing

-Dileep Singh Rathore

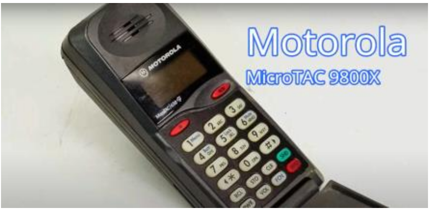
A brief history of THE MOBILE PHONE



History of The Mobile Phone…

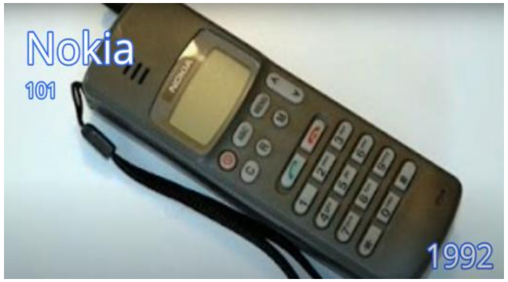


Motorola
Dynatec 800x
Launched 1983

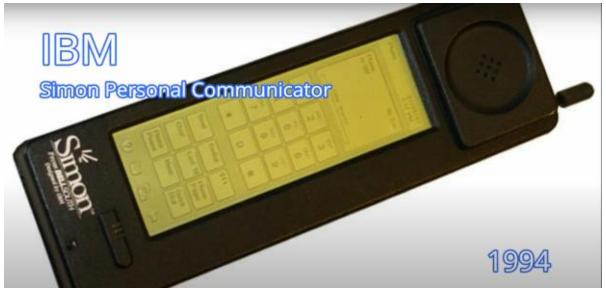


Motorola
MicroTAC 9800X

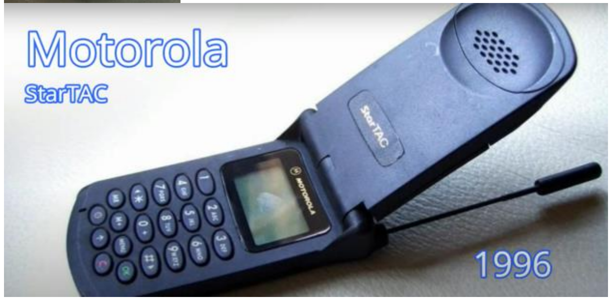
Launched 1989



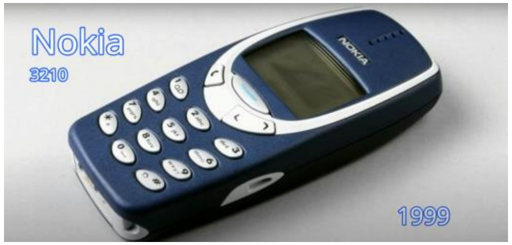
Nokia
101
1992



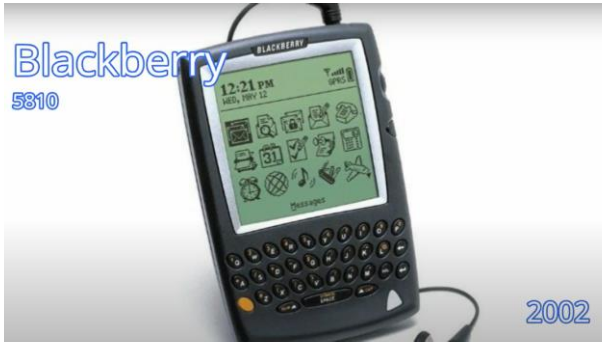
IBM
Simon Personal Communicator
1994



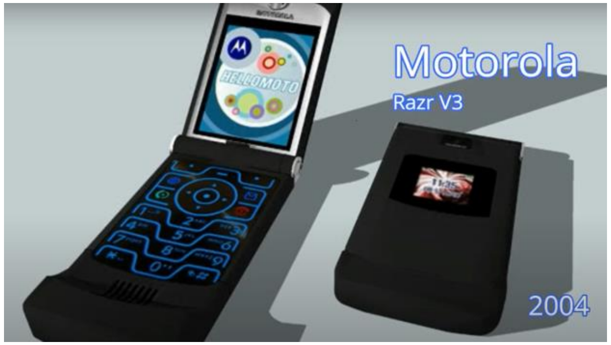
Motorola
StarTAC
1996



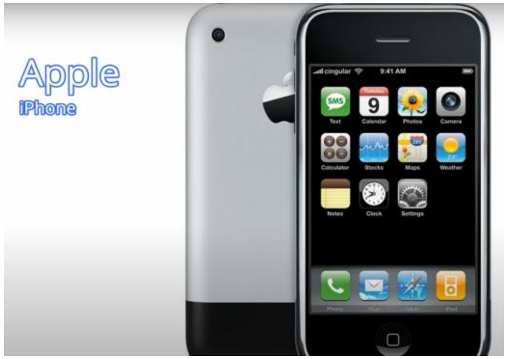
Nokia
3210
1999



Blackberry
5810
2002



Motorola
Razr V3
2004



Apple
iPhone

History of The Mobile Phone…

History of Android

1. Initially, **Andy Rubin** founded Android Incorporation in Palo Alto, California, United States in October, 2003.

2) In 17th August 2005, Google acquired android Incorporation. Since then, it is in the subsidiary of Google Incorporation.

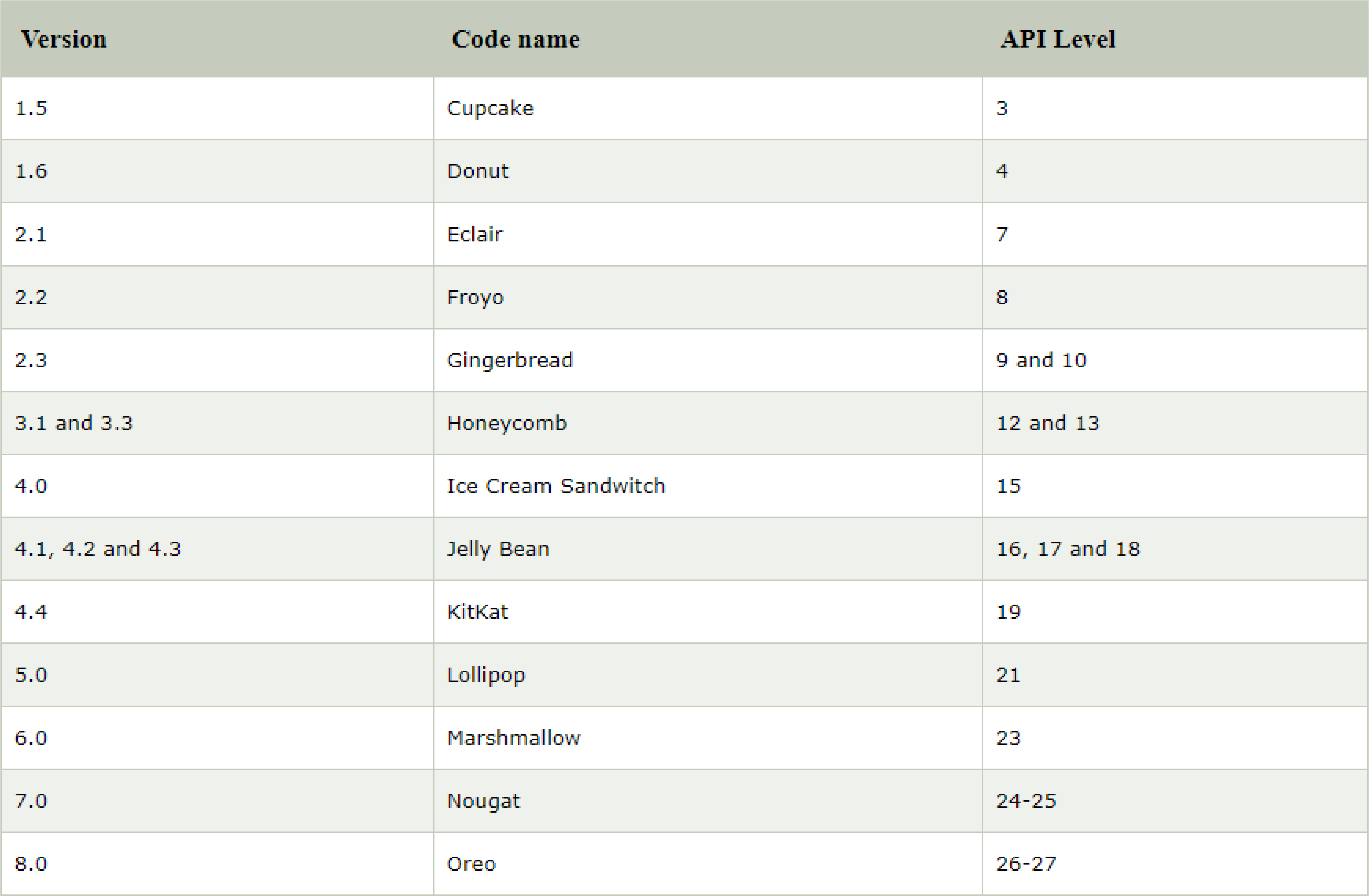
3) The key employees of Android Incorporation are **Andy Rubin**, **Rich Miner**, **Chris White** and **Nick Sears**.

4) Originally intended for camera but shifted to smart phones later because of low market for camera only.

5) Android is the nick name of Andy Rubin given by co workers because of his love to robots.

6) In 2007, Google announces the development of android OS.

7) In 2008, HTC launched the first android mobile.



Version	Code name	API Level
1.5	Cupcake	3
1.6	Donut	4
2.1	Eclair	7
2.2	Froyo	8
2.3	Gingerbread	9 and 10
3.1 and 3.3	Honeycomb	12 and 13
4.0	Ice Cream Sandwitch	15
4.1, 4.2 and 4.3	Jelly Bean	16, 17 and 18
4.4	KitKat	19
5.0	Lollipop	21
6.0	Marshmallow	23
7.0	Nougat	24-25
8.0	Oreo	26-27

Mobile Application Development Strategy

- Planning a Mobile App
- A "One-Size Fits All" model isn't adequate
  - Making a Decision Between Mobile, Hybrid and Native App Development
- Getting Approval for Your Mobile App
- Designing and Developing Your Mobile App
- Listing Your Mobile App

OHA (Open Handset Alliance)

The Open Handset Alliance (OHA) is a business alliance that was created for the purpose of developing open mobile device standards. The OHA has approximately 80 member companies, including HTC, Dell, Intel, Motorola, Qualcomm and Google. The OHA's main product is the Android platform - the world's most popular smartphone platform.

OHA members are primarily mobile operators, handset manufacturers, software development firms, semiconductor companies and commercialization companies. Members share a commitment to expanding the commercial viability of open platform development.

OHA member companies back the open platform concept for a number of reasons, as follows:

- **Lower overall handset costs:** Opens up resources, which facilitates the focus on creating innovative applications, solutions and services.
- **Developer-friendly environment:** In the open-source community, developers share notes to expedite application development.
- **Post-development:** Provides an ideal channel for application marketing and distribution.

Mobile Information Architecture

- Your information architecture (also known as IA), is the foundation of your mobile product. A well-engineered product with good visual design can still fail because of poor information architecture. The truly successful mobile products always have a well-thought-out information architecture.

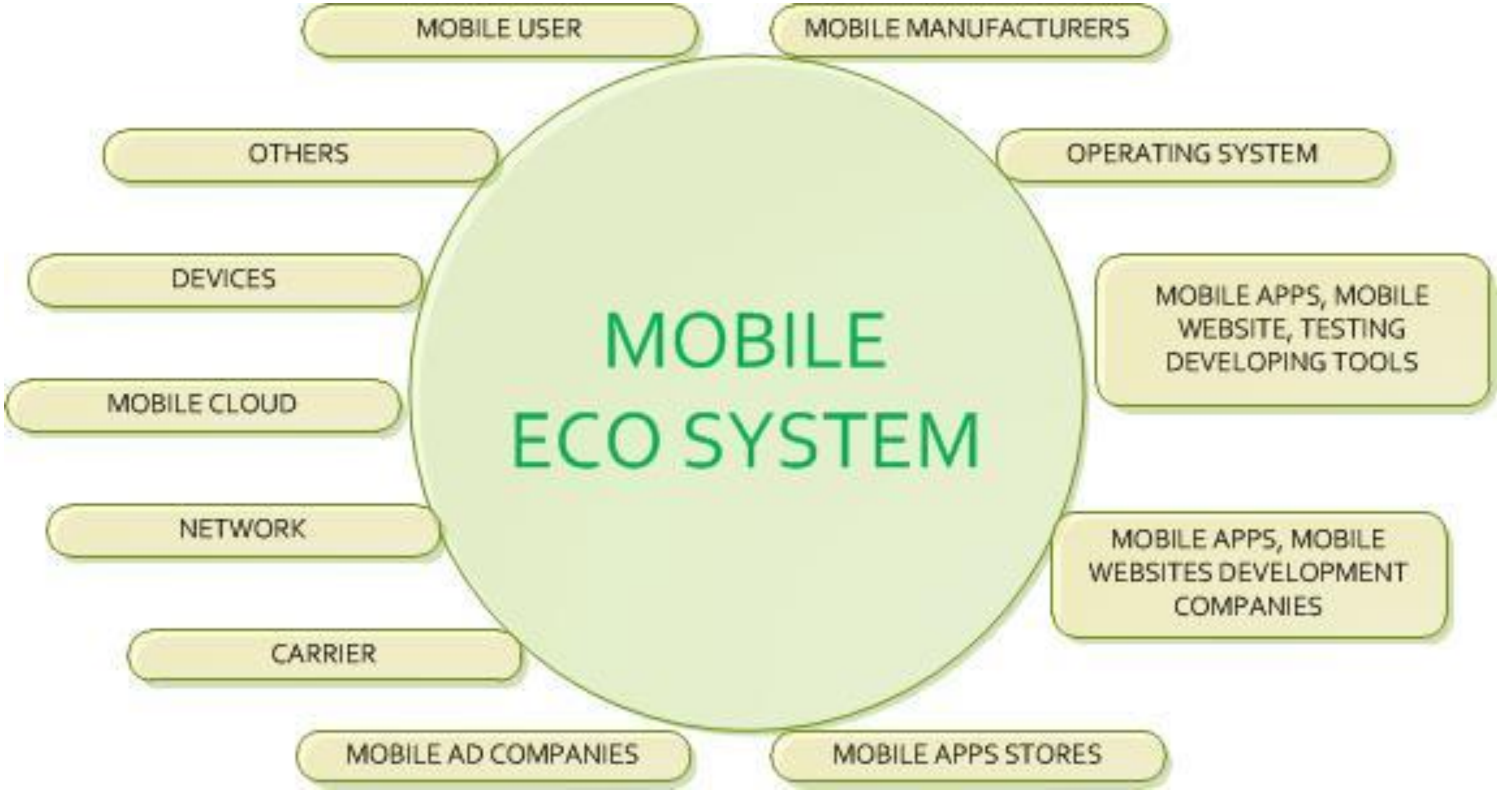
- From a simple mobile website to an iPhone application, the mobile information architecture defines not just how your information will be structured, but also how people will interact with it. This is made especially tricky when you consider that different devices have different capabilities and therefore different interaction models. Take the way people interact with their devices: for example, a touch device on which the user literally points and clicks, or a more basic device on which the user uses the directional pad to navigate to the desired location.

Mobile Ecosystem:

Mobile Ecosystem is collection of multiple devices (mobile phones, Tablet, Phablet etc), software (operating system, development tools, testing tools etc.), companies (device manufacturers, carrier, apps stores, development/testing companies, etc.) etc., and the process by which data (sms, bank transactions etc.), is transferred/shared by a user from one device to another device or by the device itself based on some programs (Birthday, Wedding Messages, calendar).

Data (Text, MultiMedia, VOICE) sharing can be done between devices of the same operating system or different operating systems. Examples: IPhone (IOS) to Windows Phone or IPhone(IOS) to Nexus(Android) or Motorola(Android) to Nexus (Android).

Data can be also shared between multiple devices with the same operating system of the same manufacturer. Example: Apples: IOS: Iphone, Ipad, to Ipod, TV, Laptops.



MOBILE USER

MOBILE MANUFACTURERS

OTHERS

OPERATING SYSTEM

DEVICES

MOBILE APPS, MOBILE WEBSITE, TESTING DEVELOPING TOOLS

MOBILE ECO SYSTEM

MOBILE CLOUD

NETWORK

MOBILE APPS, MOBILE WEBSITES DEVELOPMENT COMPANIES

CARRIER

MOBILE AD COMPANIES

MOBILE APPS STORES

Mobile Manufacturers:

They manufacture mobiles.

- Example: Samsung, BlackBerry, Sony, Nokia, Motorola, Windows Phone, Nexus

Operating System:

This is the important component of a Mobile, which controls/operates all applications that are residing on the mobile phone. Android is open source and IOS is a closed source.

- Example: IOS, Android, BlackBerry OS, Symbian, Bada etc.

Mobile Apps Development Tools:

- Android Applications are developed with Eclipse, Intellij Idea etc.
- BlackBerry Applications are developed with Eclipse etc.
- IOS Applications are developed with XCode, MonoDevelop, AppCode etc.
- Windows Phone Applications are developed with Microsoft Visual Studio etc.

Please go to relevant websites to get more information on each Tool.

Mobile Website Development Tools:

- HTML5, CSS3, JavaScript etc. are used to create mobile websites.

Mobile Apps Testing Tools:

A. Emulators:

Actual device (mobile) operations and functions are simulated on to the computer. Instead of buying several devices (Motorola, Samsung, Micromax etc), we can use emulators for functional testing. Network Connectivity, resolution testing etc cannot be tested 100% with emulators. So for testing to be 100% perfect, devices are needed and again buying several devices will be very costly. For this, you can go to Mobile Cloud environment companies and rent those mobile phone.

B. Mobile Cloud:

Companies rent mobiles and other devices virtually at hourly or weekly basis or monthly basis etc. Applications can be tested by subscribing to those companies.

Example:

- DeviceAnywhere (http://www.deviceanywhere.com/mobile-application-testing-overview.html)
- Perfecto Mobile: www.perfectomobile.com

C. Mobile Testing Tools:

Jamo Solutions, Perfecto Mobile, Device Anywhere Pro.

Mobile Stores:

Application created can be uploaded in stores after getting approval from those stores and applications can be sold. Applications can also be downloaded at free of cost or on paid basis.

- Examples: play.google.com, store.apple.com

Mobile Ad Companies:

They display advertisements on the mobiles. Example: Google's AdMob.

Carrier/Network:

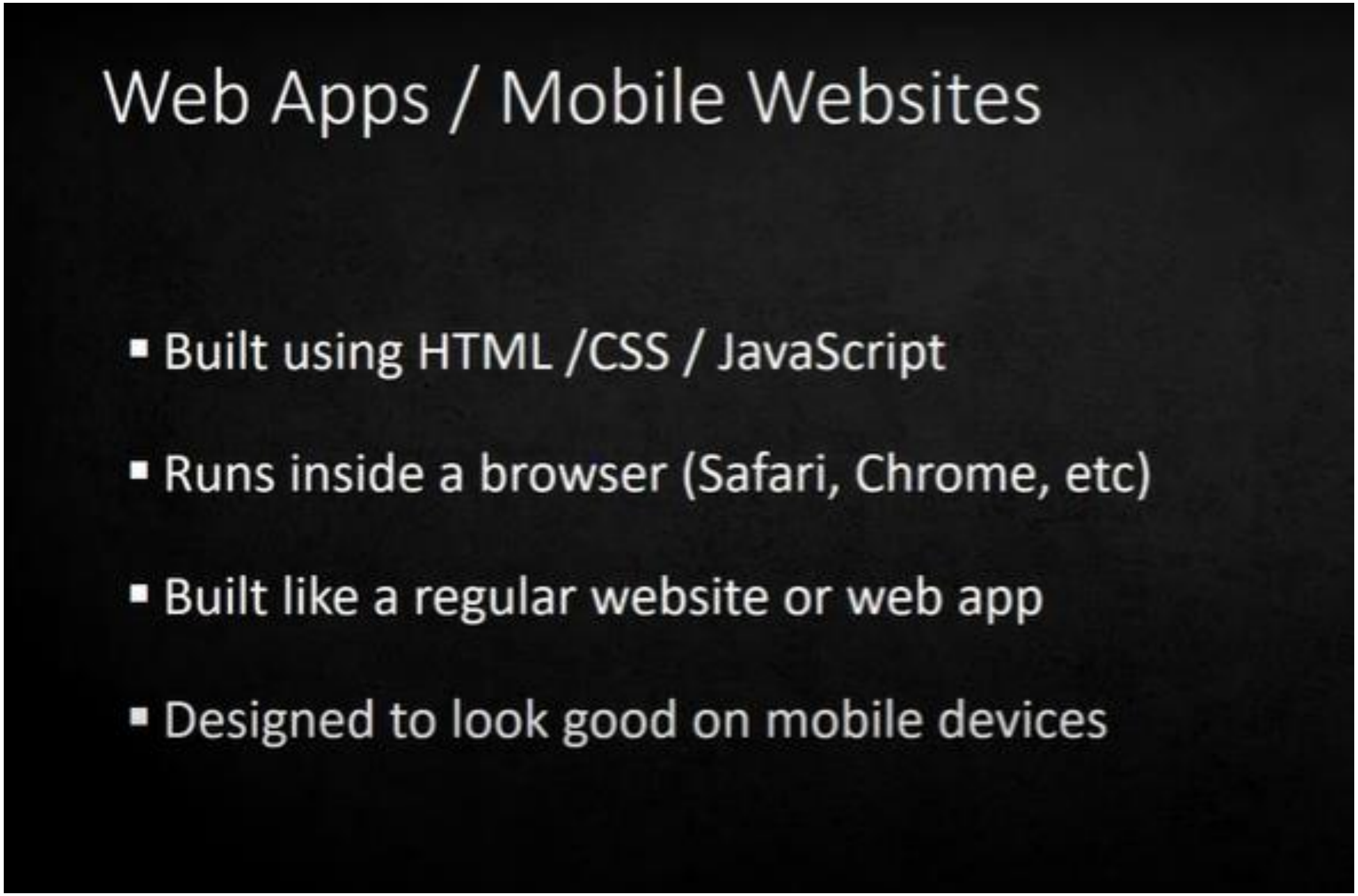
Carriers carry Voice data (calling), text data (sms), and multimedia data etc from one device to another device through Networks. Users subscribe to different plans available with Carriers.

Types of mobile application

- Mobile Web App Development
- Hybrid App Development
- Native Mobile App Development

Types of Mobile Apps

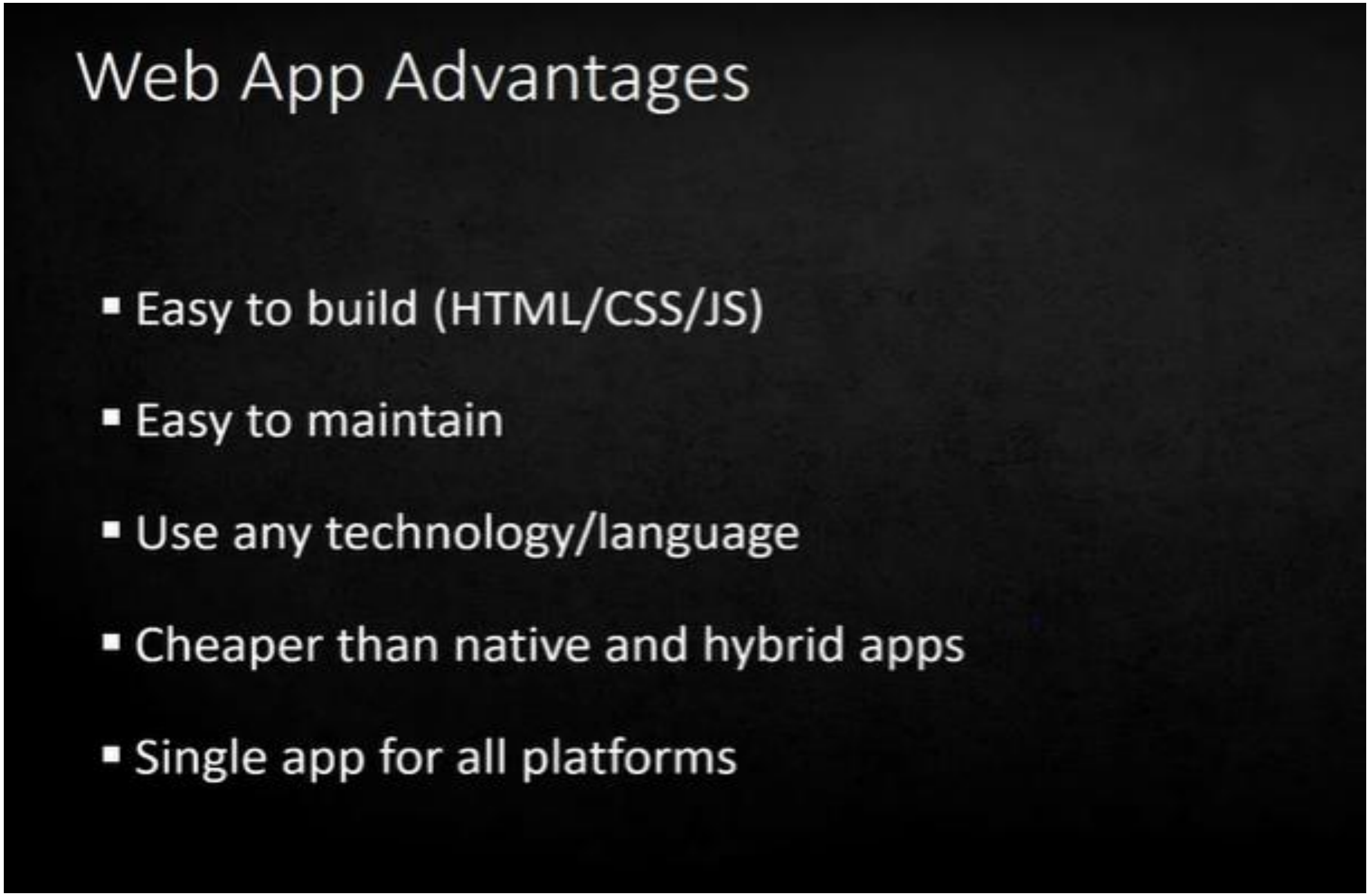
Web Apps



Web Apps / Mobile Websites

- Built using HTML /CSS / JavaScript
- Runs inside a browser (Safari, Chrome, etc)
- Built like a regular website or web app
- Designed to look good on mobile devices

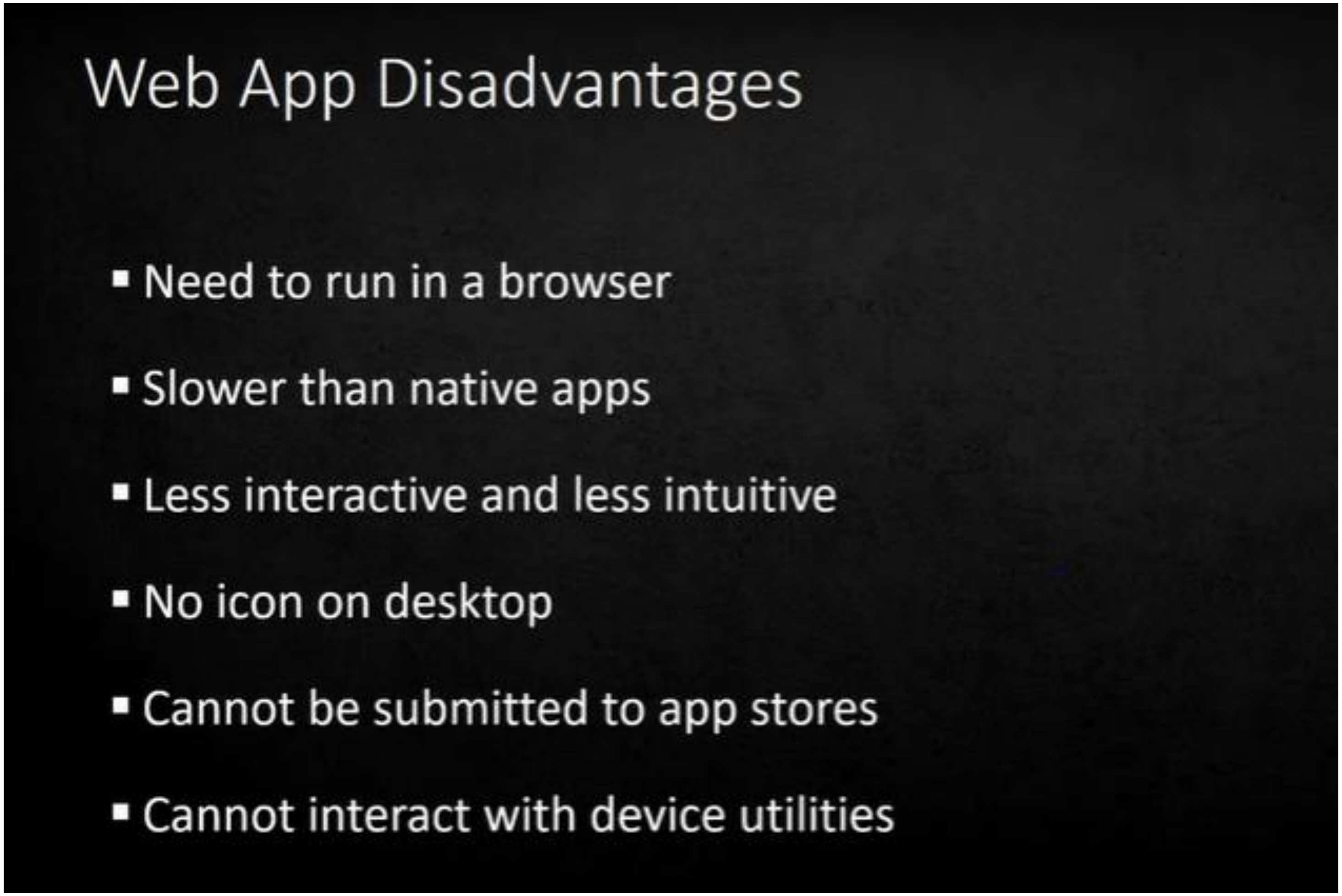
Web Apps…



Web App Advantages

- Easy to build (HTML/CSS/JS)
- Easy to maintain
- Use any technology/language
- Cheaper than native and hybrid apps
- Single app for all platforms

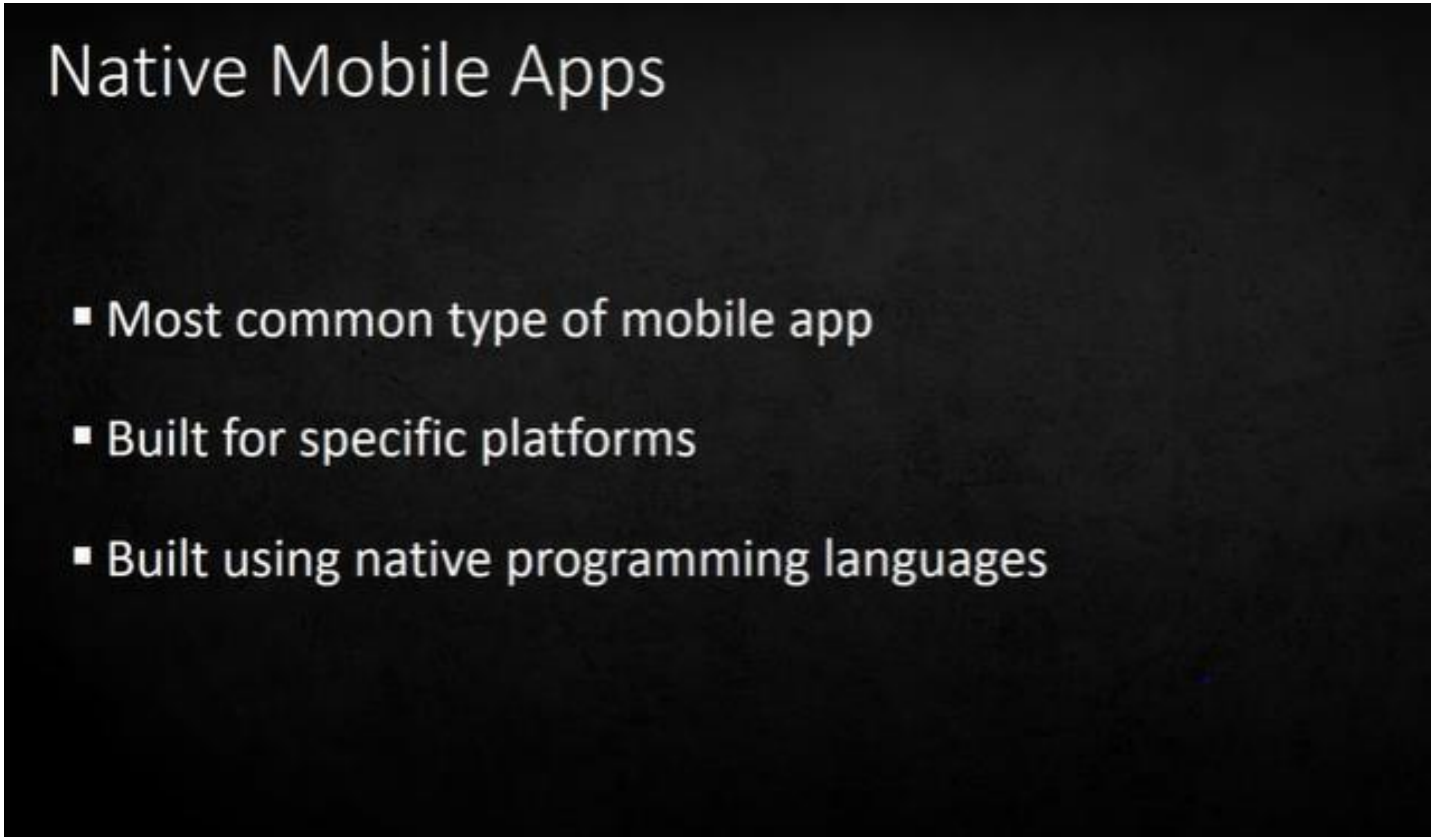
Web Apps…



Web App Disadvantages

- Need to run in a browser
- Slower than native apps
- Less interactive and less intuitive
- No icon on desktop
- Cannot be submitted to app stores
- Cannot interact with device utilities

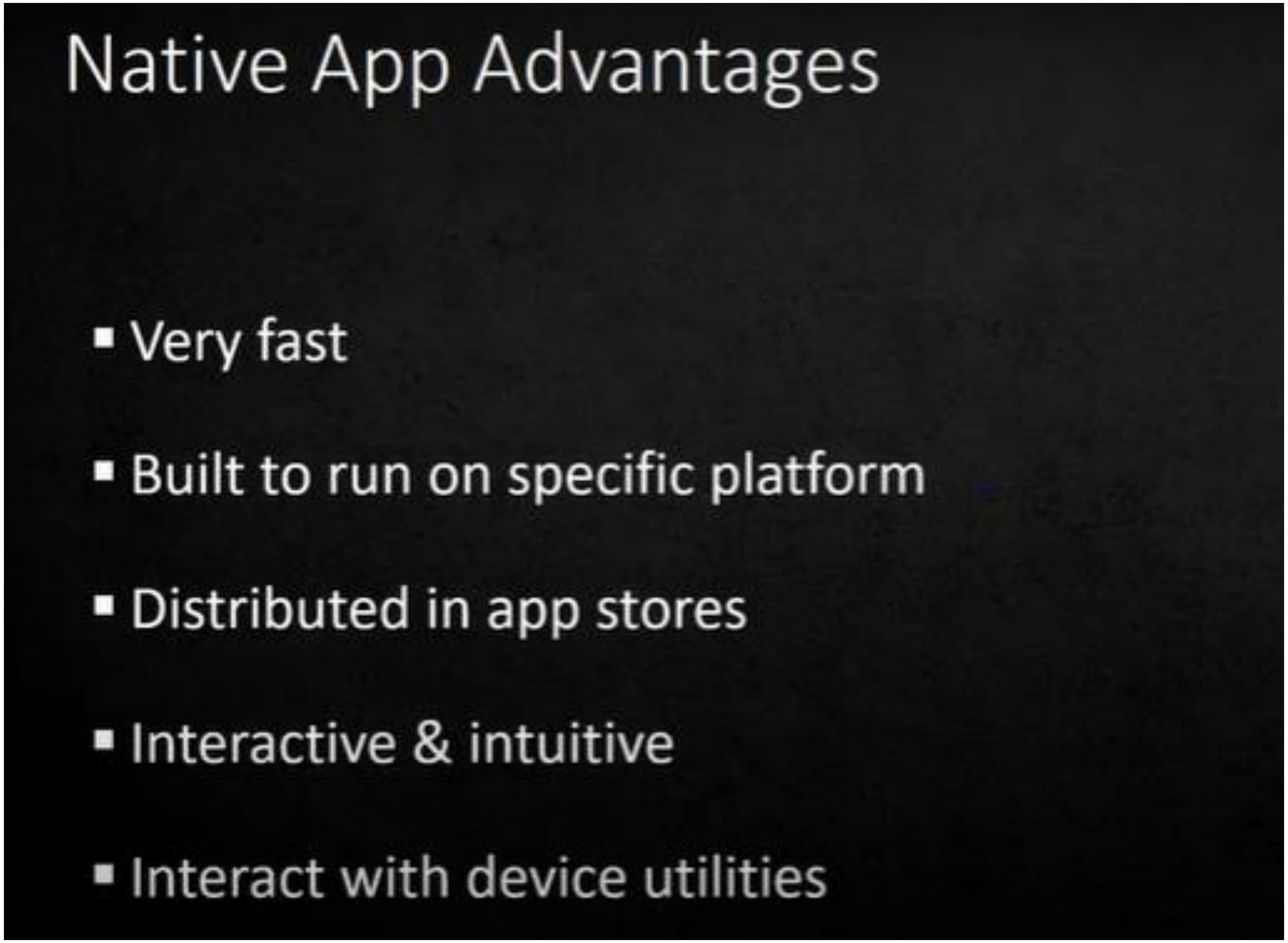
Native Apps



Native Mobile Apps

- Most common type of mobile app
- Built for specific platforms
- Built using native programming languages

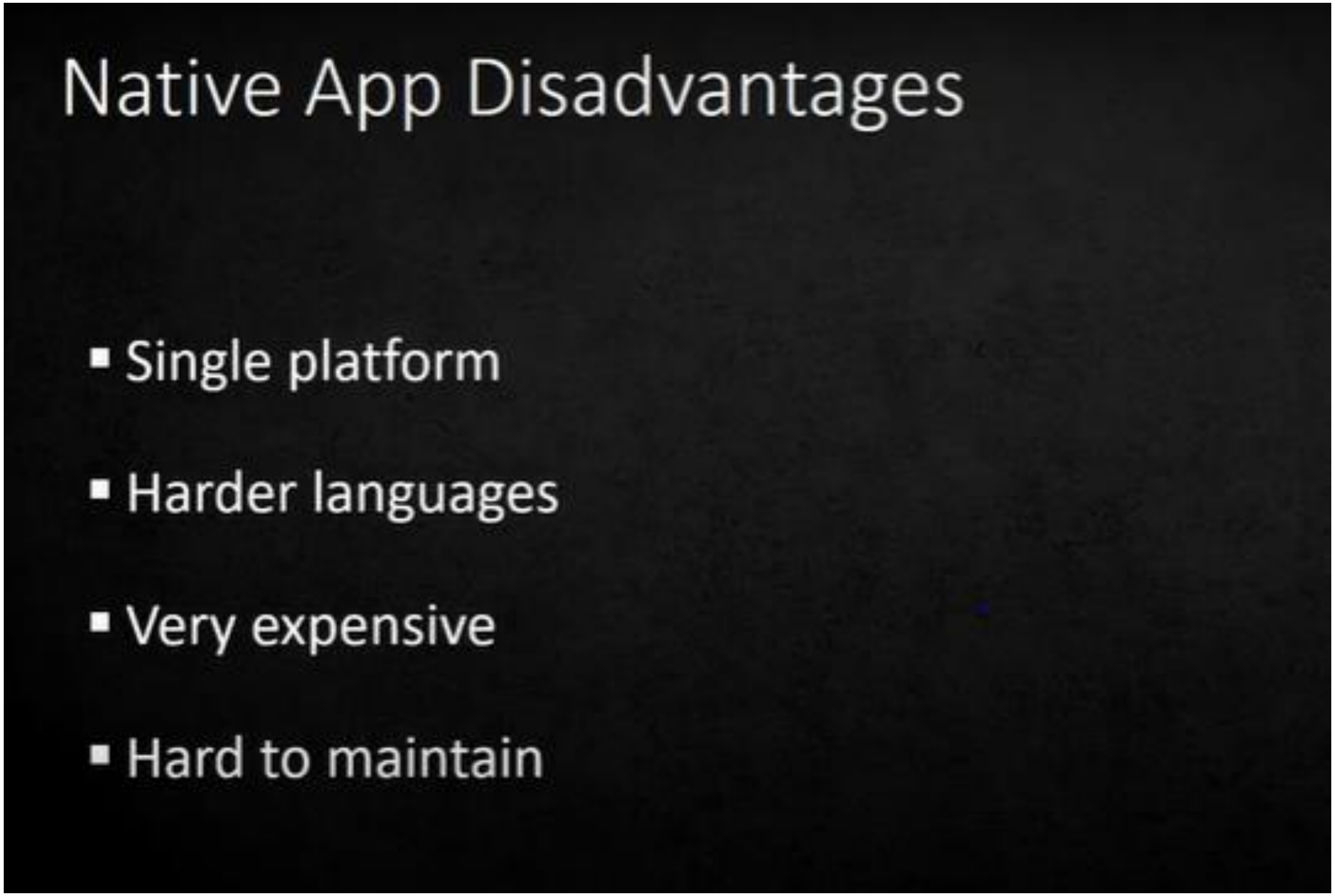
Native Apps…



Native App Advantages

- Very fast
- Built to run on specific platform
- Distributed in app stores
- Interactive & intuitive
- Interact with device utilities

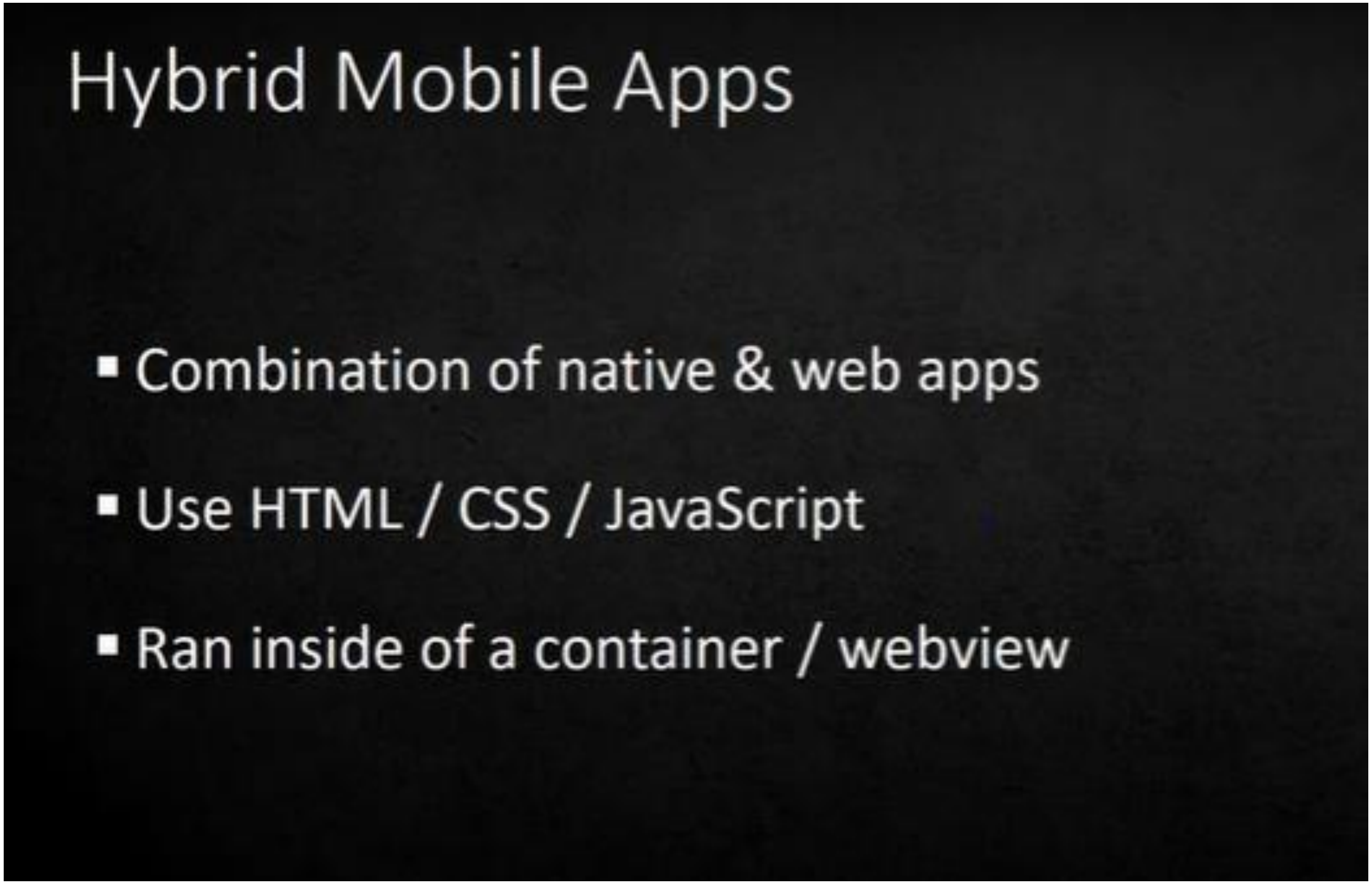
Native Apps…



Native App Disadvantages

- Single platform
- Harder languages
- Very expensive
- Hard to maintain

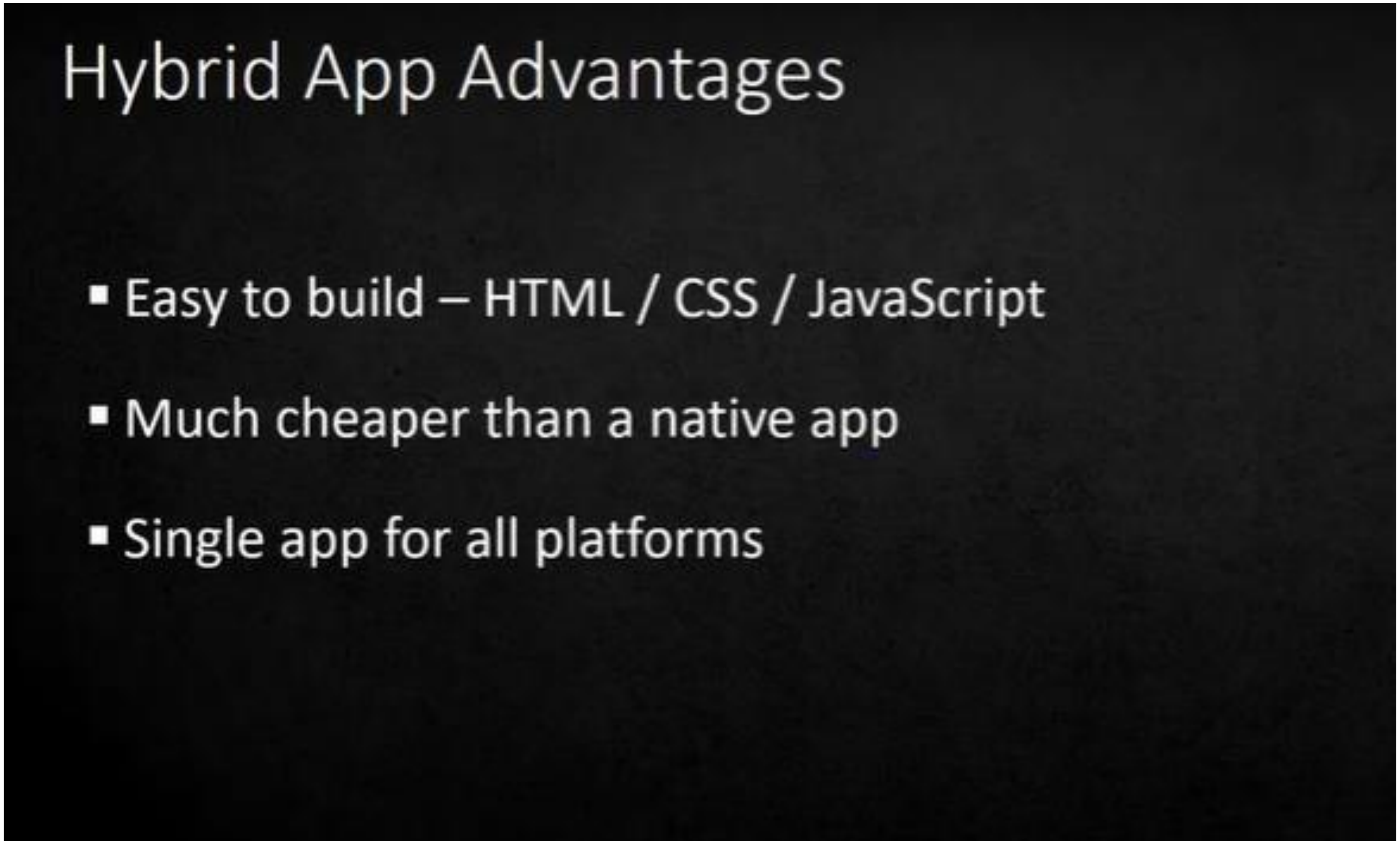
Hybrid Apps



Hybrid Mobile Apps

- Combination of native & web apps
- Use HTML / CSS / JavaScript
- Ran inside of a container / webview

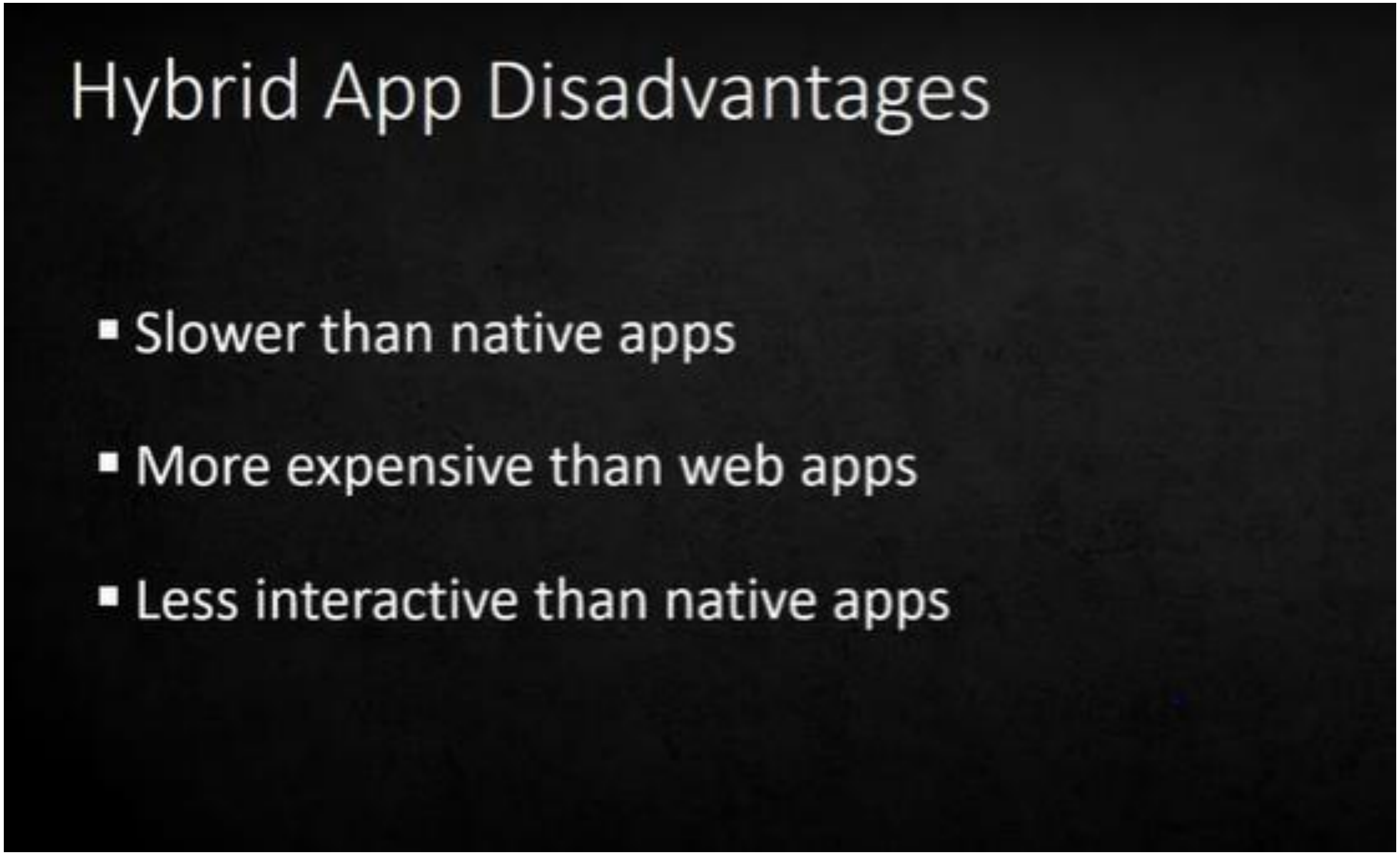
Hybrid Apps…



Hybrid App Advantages

- Easy to build – HTML / CSS / JavaScript
- Much cheaper than a native app
- Single app for all platforms

Hybrid Apps…



Hybrid App Disadvantages

- Slower than native apps
- More expensive than web apps
- Less interactive than native apps

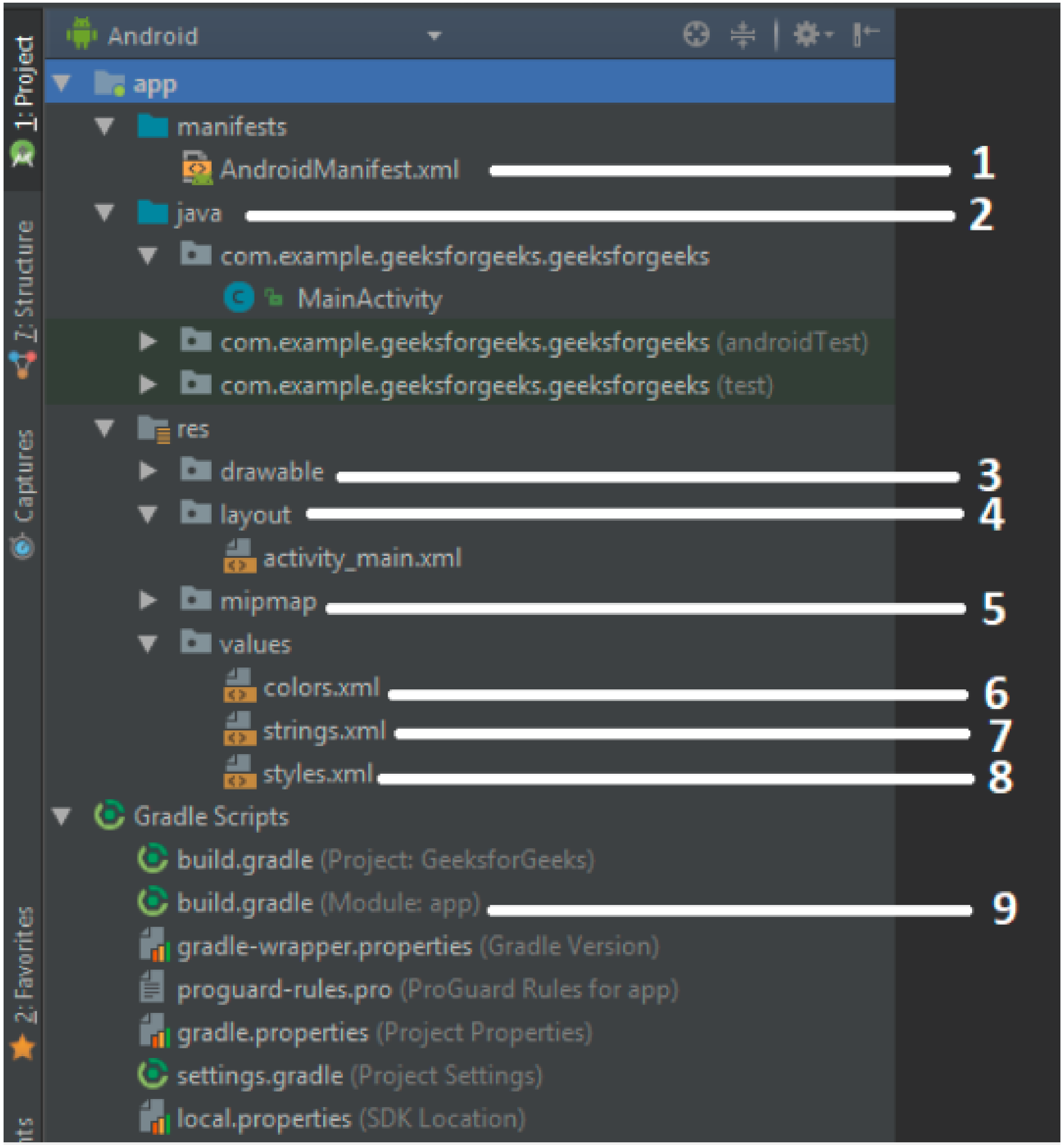
Which one?

- Which mobile platforms do you wish to target?
- Do you want to distribute your application via app stores?
- Are you looking to utilize the capabilities of the mobile device?
- What are the technical abilities of your development team?
- Does the one-size-fit-all approach of hybrid really live up to its promise?

NATIVE vs. WEB vs. HYBRID: 7 FACTORS OF COMPARISON

KEY 🟥 CON 🟩 PRO ⬜ NEUTRAL

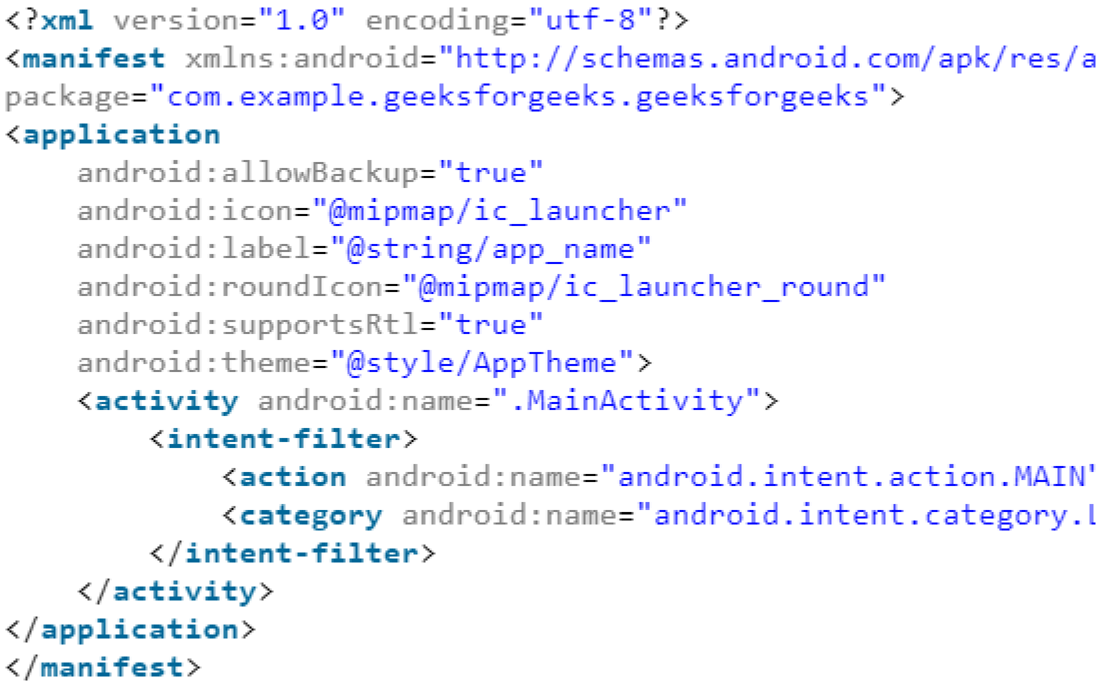
	NATIVE	HYBRID	WEB
COST	🟥 Commonly the highest of the three choices if developing for multiple platforms	🟩 Similar to pure web costs, but extra skills are required for hybrid tools	🟩 Lowest cost due to single codebase and common skillset
CODE REUSABILITY/ PORTABILITY	🟥 Code for one platform only works for that platform	🟩 Most hybrid tools will enable portability of a single codebase to the major mobile platforms	🟩 Browser compatibility and performance are the only concerns
DEVICE ACCESS	🟩 Platform SDK enables access to all device APIs	⬜ Many device APIs closed to web apps can be accessed, depending on the tool	🟥 Only a few device APIs like geolocation can be accessed, but the number is growing
UI CONSISTENCY	🟩 Platform comes with familiar, original UI components	⬜ UI frameworks can achieve a fairly native look	⬜ UI frameworks can achieve a fairly native look
DISTRIBUTION	⬜ App stores provide marketing benefits, but also have requirements and restrictions	⬜ App stores provide marketing benefits, but also have requirements and restrictions	⬜ No restrictions to launch, but there are no app store benefits
PERFORMANCE	🟩 Native code has direct access to platform functionality, resulting in better performance	🟥 For complex apps, the abstraction layers often prevent native-like performance	🟥 Performance is based on browser and network connection
MONETIZATION	🟩 More monetization opportunities, but stores take a percentage	🟩 More monetization opportunities, but stores take a percentage	🟥 No store commissions or setup costs, but there are few monetization methods



```
Android
▼ app
    ▼ manifests
        AndroidManifest.xml
    ▼ java
        ▼ com.example.geeksforgeeks.geeksforgeeks
            MainActivity
        ▶ com.example.geeksforgeeks.geeksforgeeks (androidTest)
        ▶ com.example.geeksforgeeks.geeksforgeeks (test)
    ▼ res
        ▶ drawable
        ▼ layout
            activity_main.xml
        ▶ mipmap
        ▼ values
            colors.xml
            strings.xml
            styles.xml
▼ Gradle Scripts
    build.gradle (Project: GeeksforGeeks)
    build.gradle (Module: app)
    gradle-wrapper.properties (Gradle Version)
    proguard-rules.pro (ProGuard Rules for app)
    gradle.properties (Project Properties)
    settings.gradle (Project Settings)
    local.properties (SDK Location)
```

1. AndroidManifest.xml: Every project in Android includes a manifest file, which is AndroidManifest.xml, stored in the root directory of its project hierarchy. The manifest file is an important part of our app because it defines the structure and metadata of our application, its components, and its requirements. This file includes nodes for each of the Activities, Services, Content Providers and Broadcast Receiver that make the application and using Intent Filters and Permissions, determines how they co-ordinate with each other and other applications.

- A typical AndroidManifest.xml file looks like:

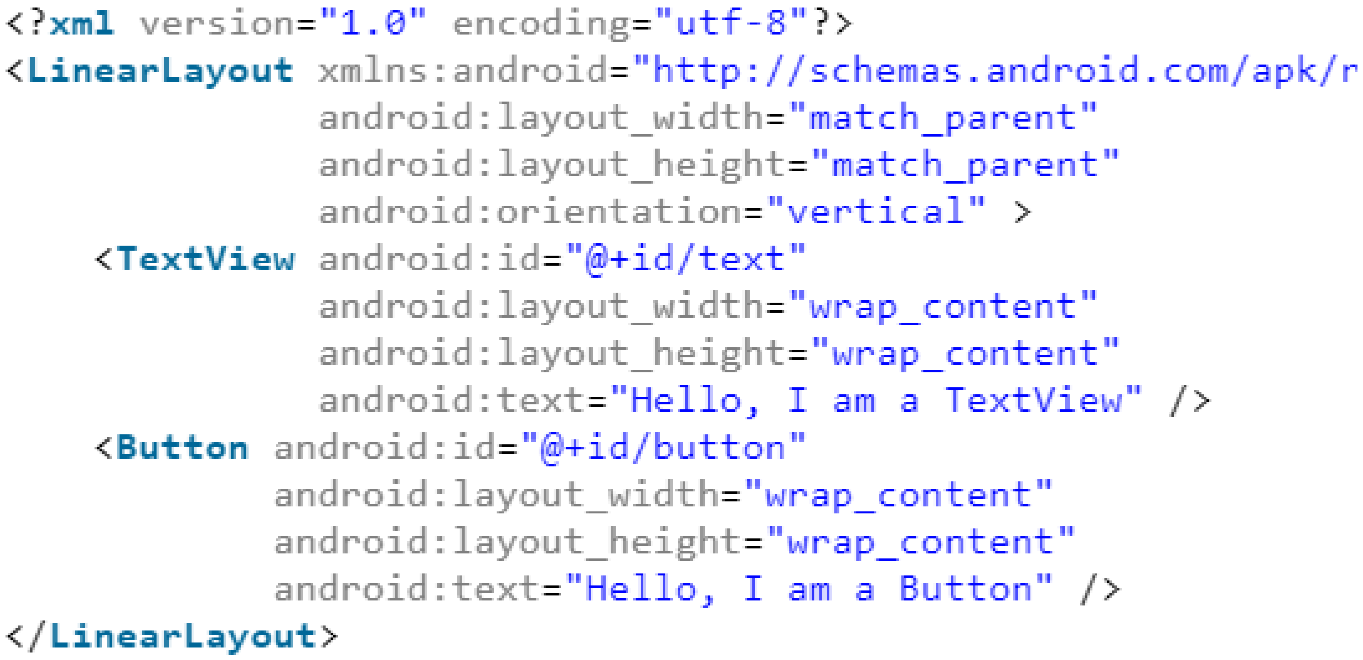


```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/a
package="com.example.geeksforgeeks.geeksforgeeks">
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".MainActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN'
            <category android:name="android.intent.category.l
        </intent-filter>
    </activity>
</application>
</manifest>
```

2. Java: The Java folder contains the Java source code files. These files are used as a controller for controlled UI (Layout file). It gets the data from the Layout file and after processing that data output will be shown in the UI layout. It works on the backend of an Android application.

3. Drawable: A Drawable folder contains resource type file (something that can be drawn). Drawables may take a variety of file like Bitmap (PNG, JPEG), Nine Patch, Vector (XML), Shape, Layers, States, Levels, and Scale.

4. layout: A layout defines the visual structure for a user interface, such as the UI for an Android application. This folder stores Layout files that are written in XML language. You can add additional layout objects or widgets as child elements to gradually build a View hierarchy that defines your layout file. Below is a sample layout file:



```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/r
              android:layout_width="match_parent"
              android:layout_height="match_parent"
              android:orientation="vertical" >
    <TextView android:id="@+id/text"
              android:layout_width="wrap_content"
              android:layout_height="wrap_content"
              android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Hello, I am a Button" />
</LinearLayout>
```

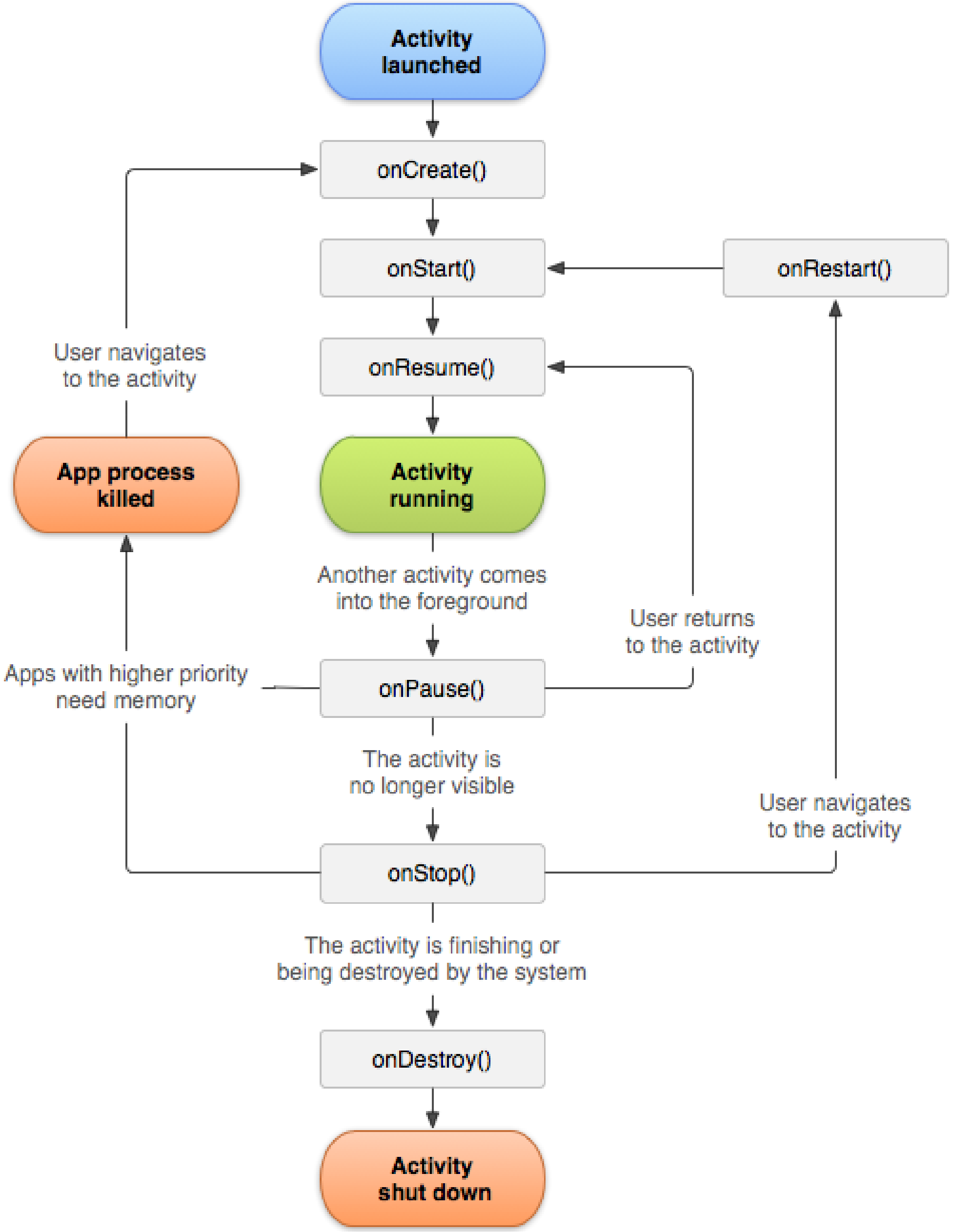
5. mipmap: Mipmap folder contains the Image Asset file that can be used in Android Studio application. You can generate the icon types like Launcher icons, Action bar and tab icons, and Notification icons.

6. colors.xml: colors.xml file contains color resources of the Android application. Different color values are identified by a unique name that can be used in the Android application program.

7. strings.xml: The strings.xml file contains string resources of the Android application. The different string value is identified by a unique name that can be used in the Android application program. This file also stores string array by using XML language.

```
<resources>
        <string name="app_name">SANDIP</string>
</resources>
```

Activity Life Cycle



Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

Practical

- Write a program to trace life cycle events for different situations.

- Log.d
- Sleep()
- 7 methods in life cycle of one activity
- screenOrientation-landscape/portrait
- How to open other activity other than activity_main fron mainActivity.

Features of Android

The important features of android are given below:

1) It is open-source.

2) Anyone can customize the Android Platform.

3) There are a lot of mobile applications that can be chosen by the consumer.

4) It provides support for messaging services(SMS and MMS), web browser, storage (SQLite), connectivity (GSM, CDMA, Blue Tooth, Wi-Fi etc.), media, handset layout etc.

Categories of Android applications

There are many android applications in the market. The top categories are:

- Entertainment
- Tools
- Communication
- Educational
- Music and Audio
- Social
- Media and Video
- Travel and Local etc.