

Mobile Application Development

Unit-1

What is Android?

- Mobile operating system based on [Linux kernel](#)
- User Interface for touch screens
- Used on [over 80%](#) of all smartphones
- Powers devices such as watches, TVs, and cars
- Over 2 Million Android apps in Google Play store
- Highly customizable for devices / by vendors
- Open source

Android user interaction

- Touch gestures: swiping, tapping, pinching
- Virtual keyboard for characters, numbers, and emoji
- Support for Bluetooth, USB controllers and peripherals

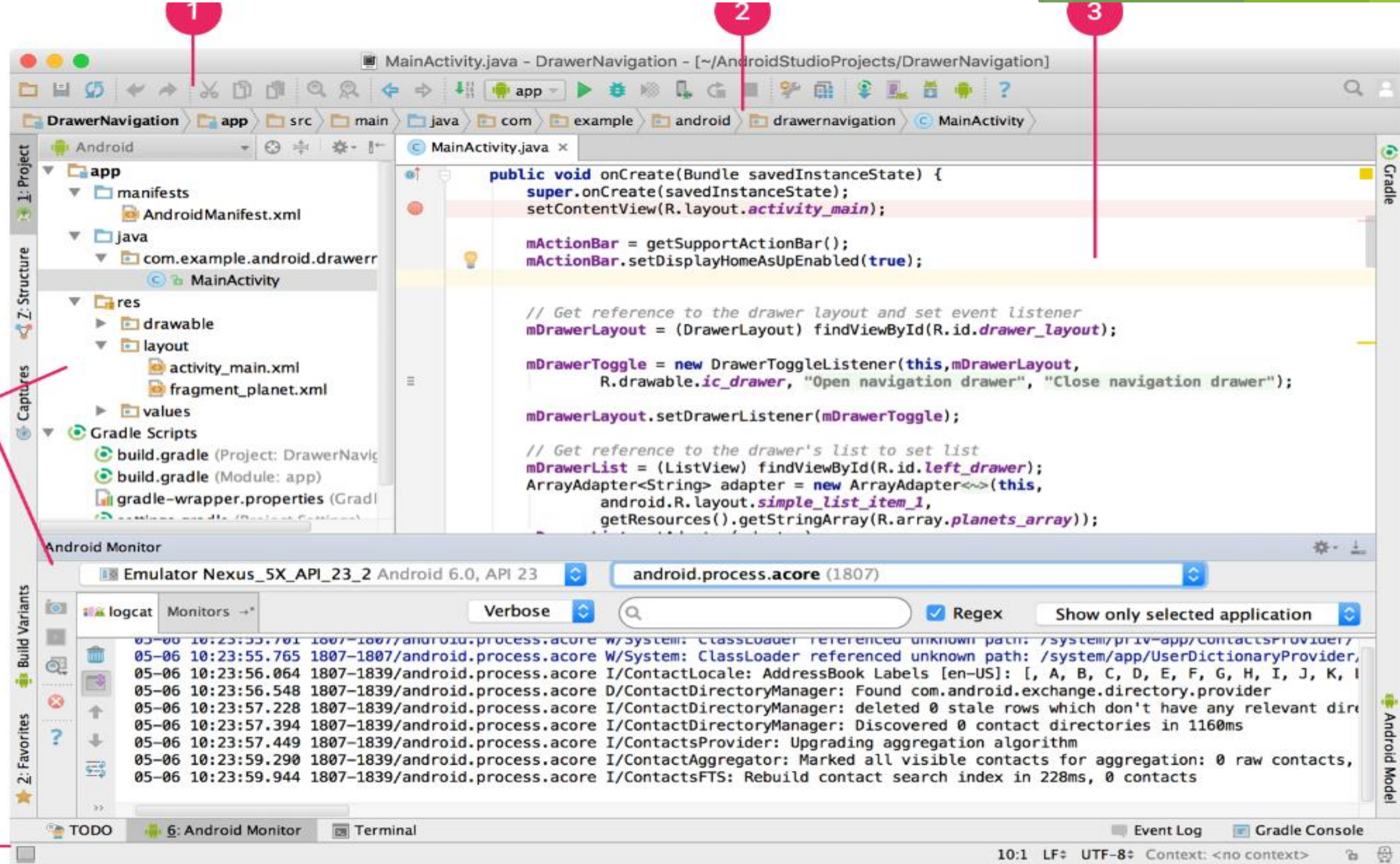
Android Software Developer Kit (SDK)

- Development tools (debugger, monitors, editors)
- Libraries (maps, wearables)
- Virtual devices (emulators)
- Documentation (developers.android.com)
- Sample code

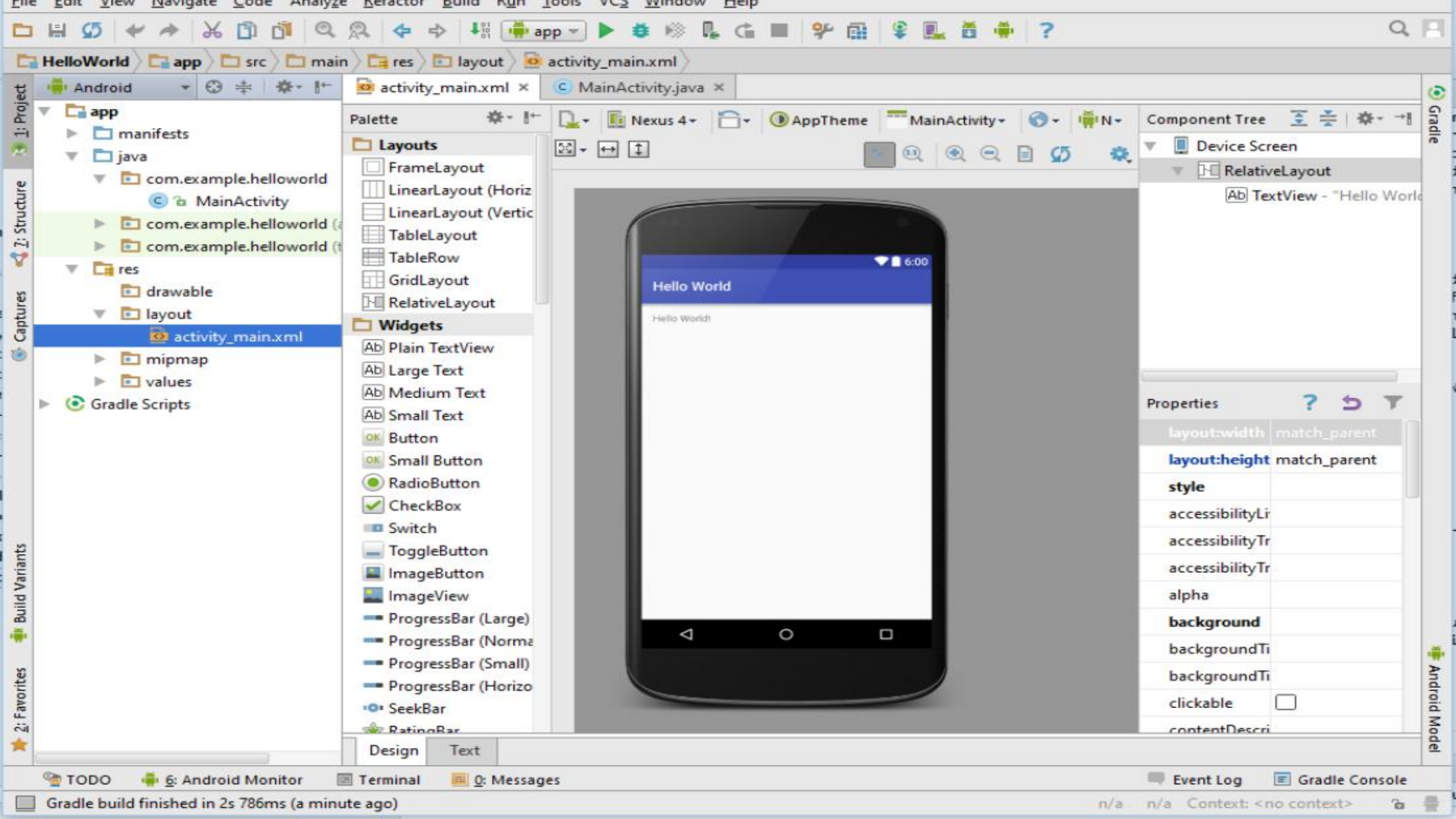
Android Studio



- [Official Android IDE](#)
- Develop, run, debug, test, and package apps
- Monitors and performance tools
- Virtual devices
- Project views
- Visual layout editor



1. The **toolbar** lets you carry out a wide range of actions, including running your app and launching Android tools.
2. The **navigation bar** helps you navigate through your project and open files for editing. It provides a more compact view of the structure visible in the Project tool window.
3. The **editor window** is where you create and modify code. Depending on the current file type, this window can change. For example, when viewing a layout file, the editor window displays the layout editor and offers the option to view the corresponding XML file.
4. **Tool windows** give you access to specific tasks like project management, search, version control, and more. You can expand them and collapse them.
5. The **status bar** displays the status of your project and the IDE itself, as well as any warnings or messages.



Google Play store

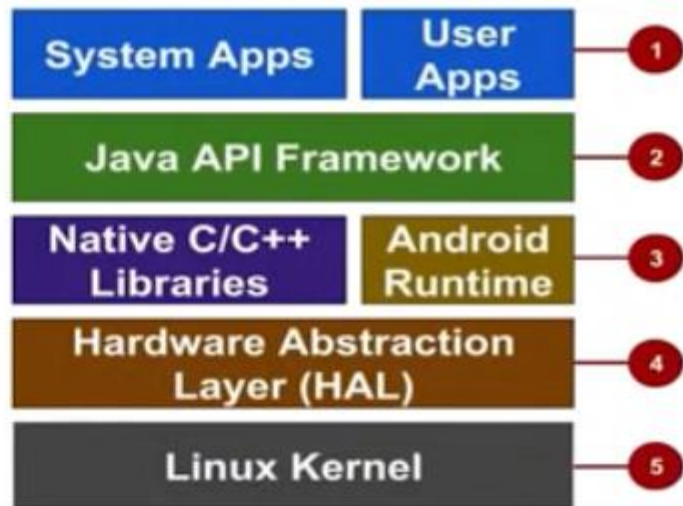
Publish apps through [Google Play](#) store:

- Official app store for Android
- Digital distribution service operated by Google



Android stack

1. System and user apps
2. Android OS API in Java framework
3. Expose native APIs; run apps
4. Expose device hardware capabilities
5. Linux Kernel



System and user apps

- System apps have no special status
- System apps provide key capabilities to app developers



Example:

Your app can use a system app to deliver a SMS message.

Java API Framework

The entire feature-set of the Android OS is available to you through APIs written in the Java language.

- View class hierarchy to create UI screens
- Notification manager
- Activity manager for life cycles and navigation
- Content providers to access data from other apps

Android runtime

Each app runs in its own process with its own instance of the Android Runtime.

C/C++ libraries

- Core C/C++ Libraries give access to core native Android system components and services.

Hardware Abstraction Layer (HAL)

- Standard interfaces that expose device hardware capabilities as libraries

Examples: Camera, bluetooth module

Linux Kernel

- Threading and low-level memory management
- Security features
- Drivers

Android versions



Codename	Version	Released	API Level	Android History and Platform Versions for more and earlier versions before 2011
<i>Honeycomb</i>	3.0 - 3.2.6	Feb 2011	11 - 13	
<i>Ice Cream Sandwich</i>	4.0 - 4.0.4	Oct 2011	14 - 15	
<i>Jelly Bean</i>	4.1 - 4.3.1	July 2012	16 - 18	
<i>KitKat</i>	4.4 - 4.4.4	Oct 2013	19 - 20	
<i>Lollipop</i>	5.0 - 5.1.1	Nov 2014	21 - 22	
<i>Marshmallow</i>	6.0 - 6.0.1	Oct 2015	23	
<i>Nougat</i>	7.0	Sept 2016	24	

What is an Android app?

- One or more interactive screens
- Written using [Java Programming Language](#) and [XML](#)
- Uses the Android Software Development Kit (SDK)
- Uses Android libraries and Android Application Framework
- Executed by Android Runtime Virtual machine (ART)

Challenges of Android development

- Multiple screen sizes and resolutions
- Performance: make your apps responsive and smooth
- Security: keep source code and user data safe
- Compatibility: run well on older platform versions
- Marketing: understand the market and your users
(Hint: It doesn't have to be expensive, but it can be.)

App building blocks

- Resources: layouts, images, strings, colors as XML and media files
- Components: activities, services, ..., and helper classes as Java code
- Manifest: information about app for the runtime
- Build configuration: APK versions in Gradle config files

Component types

- **Activity** is a single screen with a user interface
- **Service** performs long-running tasks in background
- **Content provider** manages shared set of data
- **Broadcast receiver** responds to system-wide announcements

Think of Android as a hotel

- Your app is the guest
- The Android System is the hotel manager
- Services are available when you request them (intents)
 - In the foreground (activities) such as registration
 - In the background (services) such as laundry
- Calls you when a package has arrived (broadcast receiver)
- Access the city's tour companies (content provider)