

- [Android Studio Handbook](#)
 - [Table of Contents](#)
 - [1. Introduction to Mobile Apps](#)
 - [2. Android Operating System](#)
 - [3. Key Components of Android](#)
 - [Application Framework Components](#)
 - [Android Runtime \(ART\) Components](#)
 - [Platform Libraries](#)
 - [Linux Kernel Components](#)
 - [4. Getting Started with Android Development](#)
 - [5. Application Fundamentals](#)
 - [6. Important Files in Android Projects](#)
 - [7. App Components](#)
 - [8. The Manifest File](#)
 - [9. Android User Interface Fundamentals](#)
 - [10. Layouts in Android](#)
 - [11. Creating Your First Android Project](#)
 - [Resources](#)

Android Studio Handbook

Table of Contents

- [1. Introduction to Mobile Apps](#)
- [2. Android Operating System](#)
- [3. Key Components of Android](#)
- [4. Getting Started with Android Development](#)
- [5. Application Fundamentals](#)
- [6. Important Files in Android Projects](#)
- [7. App Components](#)
- [8. The Manifest File](#)
- [9. Android User Interface Fundamentals](#)
- [10. Layouts in Android](#)
- [11. Creating Your First Android Project](#)

1. Introduction to Mobile Apps

Mobile apps are software applications that run on portable devices such as smartphones and tablets. They are typically available for download through device-specific portals like Google Play Store.

Types of mobile applications:

1. **Native Apps:** Built specifically for one operating system.
2. **Web Apps:** Mobile-optimized websites that function like apps.
3. **Hybrid Apps:** A blend of native and web apps.

Native Android apps, which are the focus of this module, are written specifically for the Android Platform.

2. Android Operating System

- Largest installed base among mobile platforms globally.
- Conquered 71% of the global market share by the end of 2021.
- First Android-powered device launched in September 2008.
- Popular for various devices including tablets, wearables, smart TVs, etc.

3. Key Components of Android

1. **Applications Layer:** Where user-installed apps reside.
2. **Application Framework:** Provides APIs for developers.
3. **Android Runtime (ART):** Responsible for executing Android applications.
4. **Platform Libraries:** C/C++ libraries supporting system features.
5. **Linux Kernel:** Foundation of the Android OS.

Application Framework Components

- Activity Manager
- Content Providers
- Resource Manager

- View System

Android Runtime (ART) Components

- Core Libraries
- Ahead-of-Time (AOT) Compilation

Platform Libraries

- SQLite
- OpenGL ES
- WebKit

Linux Kernel Components

- Memory Management
- Process Management
- Networking
- Device Drivers

4. Getting Started with Android Development

1. Download the latest version of Android Studio from <https://developer.android.com/studio>
2. Android SDK includes compiler, debugger, and device emulator.

5. Application Fundamentals

- Android apps can be written using Kotlin and Java languages.
- Each app runs in its own sandbox environment for security.
- Apps operate within their own Linux process.
- Android actively manages app processes and lifecycles.

6. Important Files in Android Projects

1. **MainActivity** (app > java > com.example.myfirstapp > MainActivity)

- Entry point for the app
- Contains core logic and code for the main screen

2. **activity_main.xml** (app > res > layout > activity_main.xml)

- Defines the layout for the activity's user interface

3. **AndroidManifest.xml** (app > manifests > AndroidManifest.xml)

- Describes fundamental characteristics of the app
- Declares app components and permissions

4. **build.gradle** (Gradle Scripts > build.gradle)

- Project-level and module-level configuration files
- Manage dependencies and build settings

7. App Components

1. **Activities**: Represent single, focused screens with user interfaces.
2. **Services**: Background components for long-running tasks.
3. **Broadcast Receivers**: Listen for and respond to system-wide announcements.
4. **Content Providers**: Manage access to structured data.

Activities, Services, and Broadcast Receivers are activated by Intents.

8. The Manifest File

The AndroidManifest.xml file must declare all app components and:

- Identify user permissions required by the app
- Declare the minimum API Level required
- Declare hardware and software features used or required
- Declare API libraries the app needs to be linked against

Example of declaring an activity in the manifest:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest ... >
    <application android:icon="@drawable/app_icon.png" ... >
        <activity android:name="com.example.project.ExampleActivity"
            android:label="@string/example_label" ... >
        </activity>
        ...
    </application>
</manifest>
```

9. Android User Interface Fundamentals

- **Views:** Basic building blocks of the UI (e.g., TextView, Button, ImageView)
- **ViewGroups:** Containers for Views and other ViewGroups (e.g., LinearLayout, RelativeLayout)
- **Fragments:** Encapsulated UI components within an Activity
- **Activities:** Containers for the app's UI, acting as entry points for user interaction

10. Layouts in Android

Layouts are extensions of the ViewGroup class that control the positioning of child Views.

Types of layouts:

1. **FrameLayout:** Simplest layout, pins each child view within its frame.
2. **LinearLayout:** Aligns child Views in vertical or horizontal lines.
3. **TableLayout:** Arranges child Views into rows and columns.
4. **ConstraintLayout:** Creates complex layouts by defining constraints between Views.

Design considerations:

- Use layout properties like **wrap_content** and **match_parent** for adaptable UIs.
- Avoid deep nesting to optimize performance.

Defining layouts:

- Preferred method is using XML external resources.
- Each layout XML must contain a single root element.

Example of a simple layout:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Enter Text Below"
    />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Text Goes Here!"
    />
</LinearLayout>
```

11. Creating Your First Android Project

1. Open Android Studio
2. Click "Start a new Android project" or File > New > New Project
3. Select "Empty Activity" and click Next
4. Configure your project:
 - Enter "My First App" as the Name
 - Enter "com.example.myfirstapp" as the Package name
 - Choose Save location
 - Select Java as the Language
5. Click Finish

Resources

Official Android developer website: <https://developer.android.com/docs>