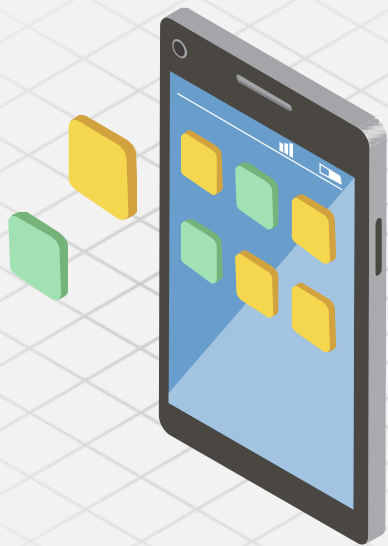


# Contents



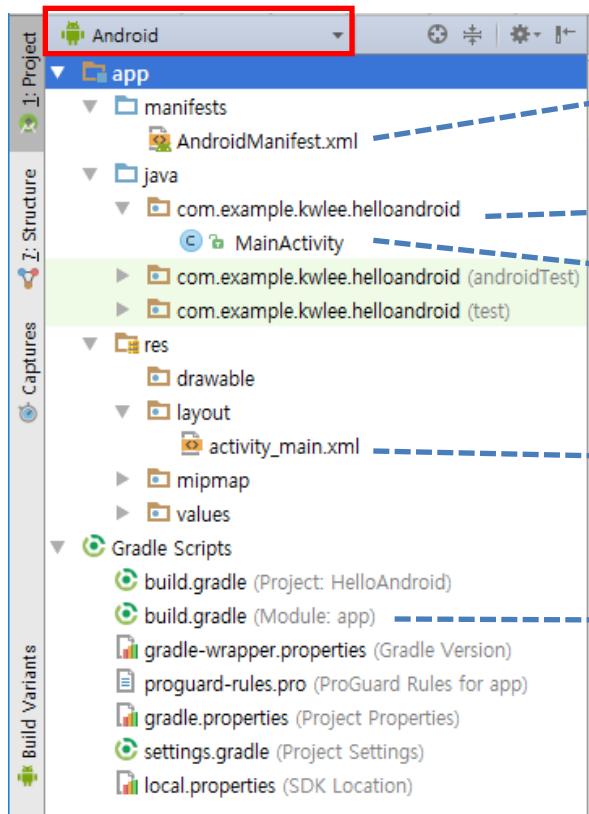
01 안드로이드 프로젝트 기본

02 안드로이드 프로젝트 관리

# 1 안드로이드 프로젝트 기본



## 가. HelloAndroid 프로젝트 구조 (Android 뷰)



앱의 기본 특징을 설명하고 앱의 각 구성 요소(예, 액티비티)를 정의

패키지 이름 (앱의 고유 식별자)

액티비티 컴포넌트의 자바 소스 파일

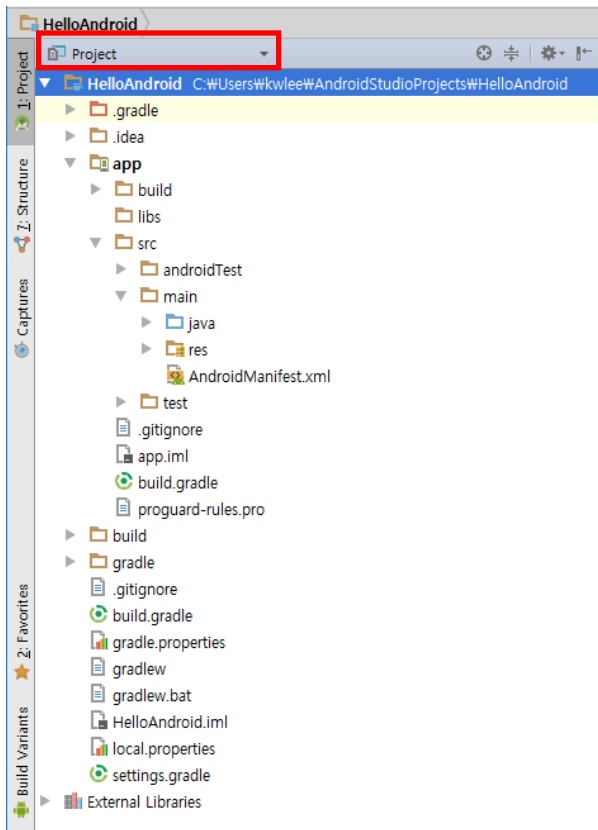
액티비티 컴포넌트의 화면 레이아웃을 정의하는 xml 파일

빌드 스크립트 파일 (컴파일 버전, 실행되는 최소 버전, 추가 컴파일 라이브러리 등)

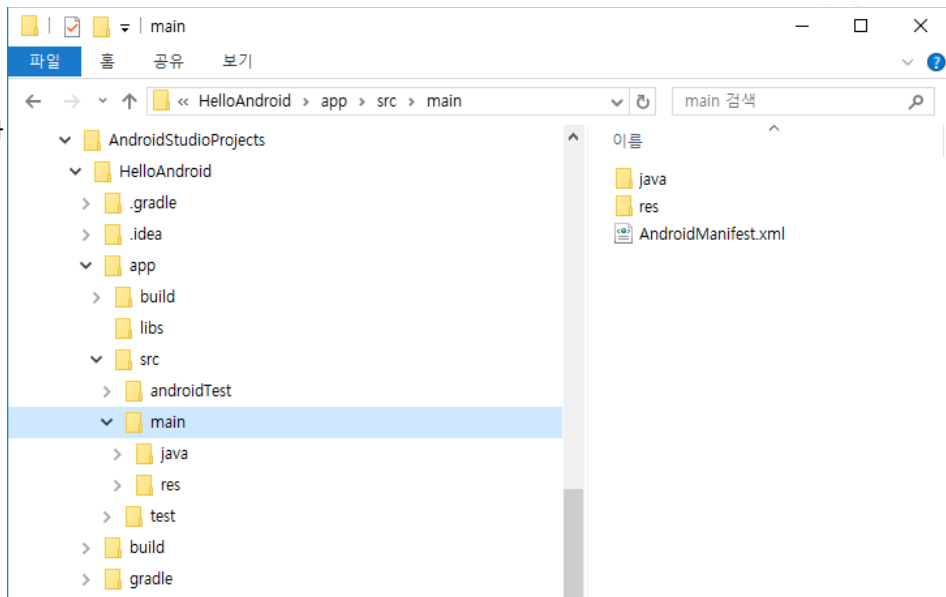
# 1 안드로이드 프로젝트 기본



## 가. HelloAndroid 프로젝트 구조 (Project 뷰)



파일 계층 구조와  
파일치



# 1 안드로이드 프로젝트 기본



## AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.kwlee.helloandroid">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="@string/app_name"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportRtl="true"
11        android:theme="@style/AppTheme">
12         <activity android:name=".MainActivity">
13             <intent-filter>
14                 <action android:name="android.intent.action.MAIN" />
15
16                 <category android:name="android.intent.category.LAUNCHER" />
17             </intent-filter>
18         </activity>
19     </application>
20
21 </manifest>
```

패키지 이름

application을 구성하는 컴포넌트 (액티비티, 서비스, BR, CP) 기술

액티비티 컴포넌트 이름

앱을 시작시킬 때, 처음 수행되는 메인 액티비티 컴포넌트를 나타냄

# 1 안드로이드 프로젝트 기본



## MainActivity.java

MainActivity

```
1 package com.example.kwlee.helloandroid;
```

```
2  
3 + import ...
```

```
5  
6 public class MainActivity extends AppCompatActivity {
```

```
7  
8 @Override
```

```
9 protected void onCreate(Bundle savedInstanceState) {
```

```
10     super.onCreate(savedInstanceState);
```

```
11     setContentView(R.layout.activity_main);
```

```
12 }
```

```
13  
14 }
```

패키지 이름

MainActivity 액티비티에 대한  
클래스 정의

MainActivity 액티비티가 생성될  
때, 수행되는 메소드 ( 안드로이드  
플랫폼이 호출하는 메소드)

activity\_main.xml에 정의된 화면  
레이아웃을 MainActivity의 컨텐  
츠뷰로 설정

# 1 안드로이드 프로젝트 기본



## activity\_main.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     tools:context="com.example.kwlee.helloandroid.MainActivity">
8
9     <TextView
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:text="Hello World!"
13         app:layout_constraintBottom_toBottomOf="parent"
14         app:layout_constraintLeft_toLeftOf="parent"
15         app:layout_constraintRight_toRightOf="parent"
16         app:layout_constraintTop_toTopOf="parent" />
17
18 </android.support.constraint.ConstraintLayout>
```

**ConstraintLayout** : 포함되는 위젯의 위치와 크기를 **RelativeLayout** 보다 유연한 방식으로 결정하는 뷰그룹

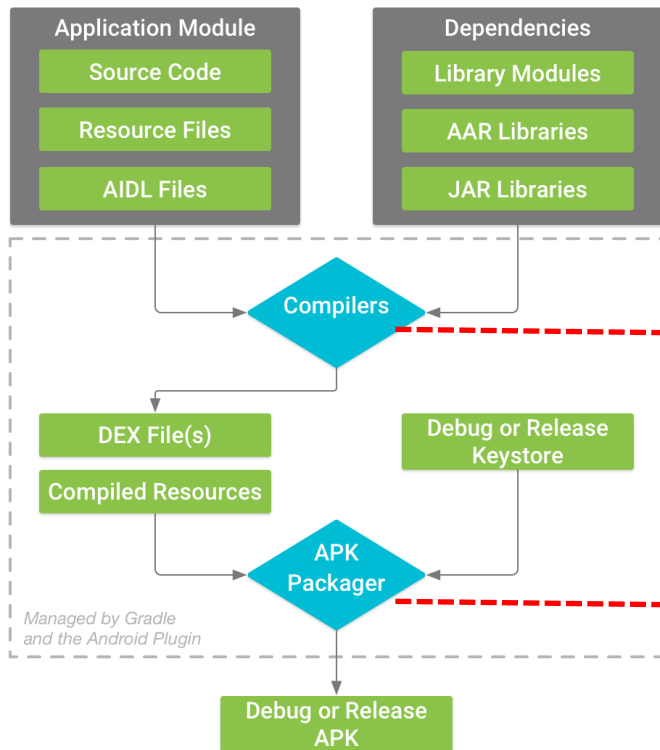
**TextView** : 문자열을 표시하는 위젯

**Hello World!** 문자열 출력

# 1 안드로이드 프로젝트 기본



## 나. 프로젝트 빌드 프로세스



**Compilers**는 소스 코드를 **DEX(Dalvik Executable)** 파일로 변환하고 그 외 모든 것을 **컴파일된 리소스**로 변환

- **APK Packager**는 DEX 파일과 컴파일된 리소스를 **단일 APK**에 결합
- **APK Packager**는 디버그 또는 릴리스 키스토어를 사용하여 **APK**를 서명

# 1 안드로이드 프로젝트 기본



## build.gradle (Module:app)

```
1  apply plugin: 'com.android.application'
2
3  android {
4      compileSdkVersion 26
5      buildToolsVersion "26.0.1"
6      defaultConfig {
7          applicationId "com.example.kwlee.helloandroid"
8          minSdkVersion 15
9          targetSdkVersion 26
10         versionCode 1
11         versionName "1.0"
12         testInstrumentationRunner "android.support.test.runner.AndroidJUnit4"
13     }
14     buildTypes {
15         release {
16             minifyEnabled false
17             proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
18         }
19     }
20 }
21
22 dependencies {
23     compile fileTree(dir: 'libs', include: ['*.jar'])
24     androidTestCompile('com.android.support.test.espresso:espresso-core:2.2.2', {
25         exclude group: 'com.android.support', module: 'support-annotations'
26     })
27     compile 'com.android.support:appcompat-v7:26.+'
28     compile 'com.android.support.constraint:constraint-layout:1.0.2'
29     testCompile 'junit:junit:4.12'
30 }
```

컴파일 시에 사용할 SDK 버전

빌드 도구의 버전

앱이 지원하는 최소 API 버전

앱을 테스트하기 위해 사용되었던 API 버전

외부 저장소(jcenter)에서 해당 라이브러리를 받아와 빌드