

코틀린을 활용한 안드로이드 앱 개발

안드로이드 기본 구조

■ 학습내용

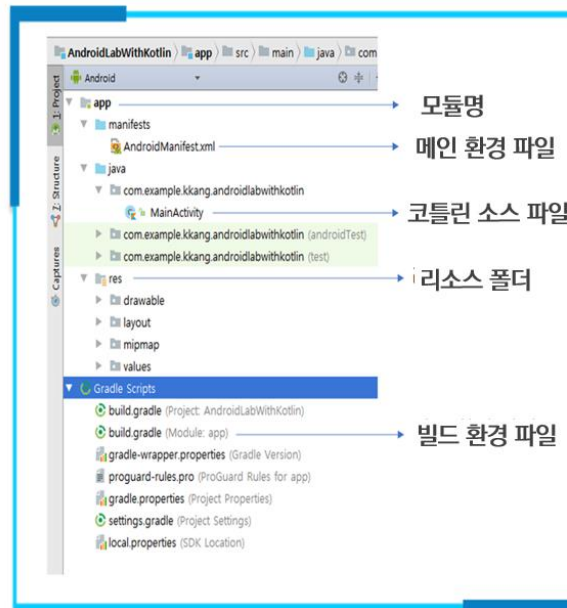
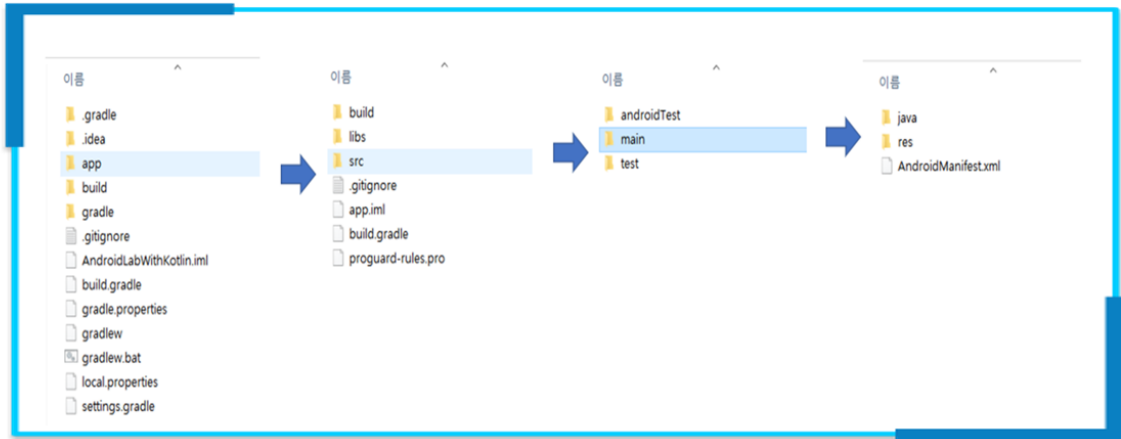
1. 안드로이드 프로젝트 구조 이해
2. 안드로이드 앱 동작 원리 이해

■ 학습목표

1. 안드로이드 프로젝트 구조에 대해 이해할 수 있다.
2. 안드로이드 앱 동작 원리에 대해 이해할 수 있다.

1. 안드로이드 프로젝트 구조 이해

1 앱의 디렉터리와 파일



1. 안드로이드 프로젝트 구조 이해

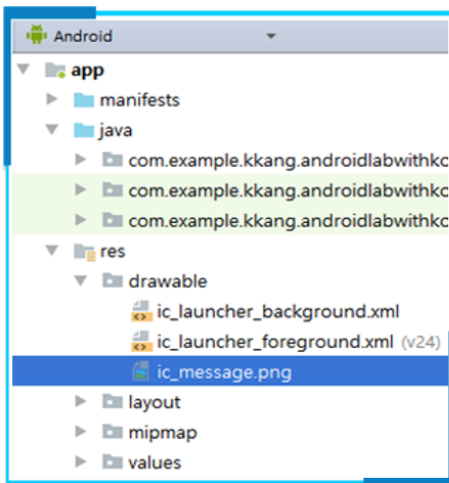
2 R.java의 이해

```
public final class R {
```

```
    public static final class anim {
        public static final int abc_fade_in=0x7f050000;
        public static final int abc_fade_out=0x7f050001;
        public static final int abc_grow_fade_in_from_bottom=0x7f050002;
        public static final int abc_popup_enter=0x7f050003;

    } //.....}
```

- ▶ res 폴더의 리소스들을 식별하기 위한 변수
- ▶ res 밑에 임의의 폴더 생성 불가
- ▶ 리소스 파일명은 자바 명명규칙을 위배할 수 없음
- ▶ 파일명에 대문자를 사용할 수 없음



```
public static final class drawable {
```

```
//.....
```

```
public static final int ic_message=0x7f020053;
```

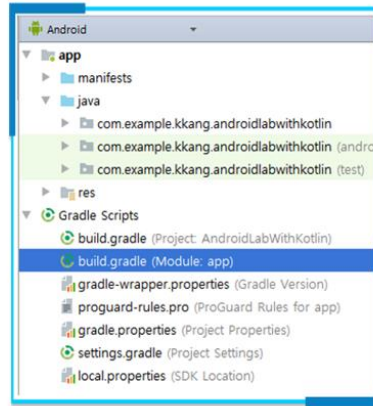
```
//.....
```

```
}
```

1. 안드로이드 프로젝트 구조 이해

3 그레이들(gradle) 파일

- ▶ ‘Gradle Scripts’ 영역의 최상위에 있는 build.gradle
- ▶ 모듈을 위한 최상위 설정을 목적



```

apply plugin: 'com.android.application'
apply plugin: 'kotlin-android'
apply plugin: 'kotlin-android-extensions'
android {
    compileSdkVersion 29
    buildToolsVersion "29.0.3"
    defaultConfig {
        applicationId "com.example.android_koreatech_20200518"
        minSdkVersion 16
        targetSdkVersion 29
        versionCode 1
        versionName "1.0"
        .....
    }
    .....
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk7:$kotlin_version"
    implementation 'androidx.appcompat:appcompat:1.1.0'
    implementation 'androidx.core:core-ktx:1.2.0'
    .....
}

```

2. 안드로이드 앱 동작 원리 이해

1 AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android_koreatech_20200518">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

2 MainActivity.kt

```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)
    }
}
```

2. 안드로이드 앱 동작 원리 이해

3 activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

■ 정리하기

1. 안드로이드 프로젝트 구조 이해

- 앱의 메인 환경 파일은 AndroidManifest.xml
- 앱의 리소스는 R.java 파일에 int 변수로 식별자로 등록되어 이용
- 빌더를 위한 환경파일은 build.gradle

2. 안드로이드 앱 동작 원리 이해

- AndroidManifest.xml의 설정에 의해 앱 시작 시 MainActivity가 실행
- MainActivity에서 layout_main.xml에 등록된 내용대로 화면을 구성