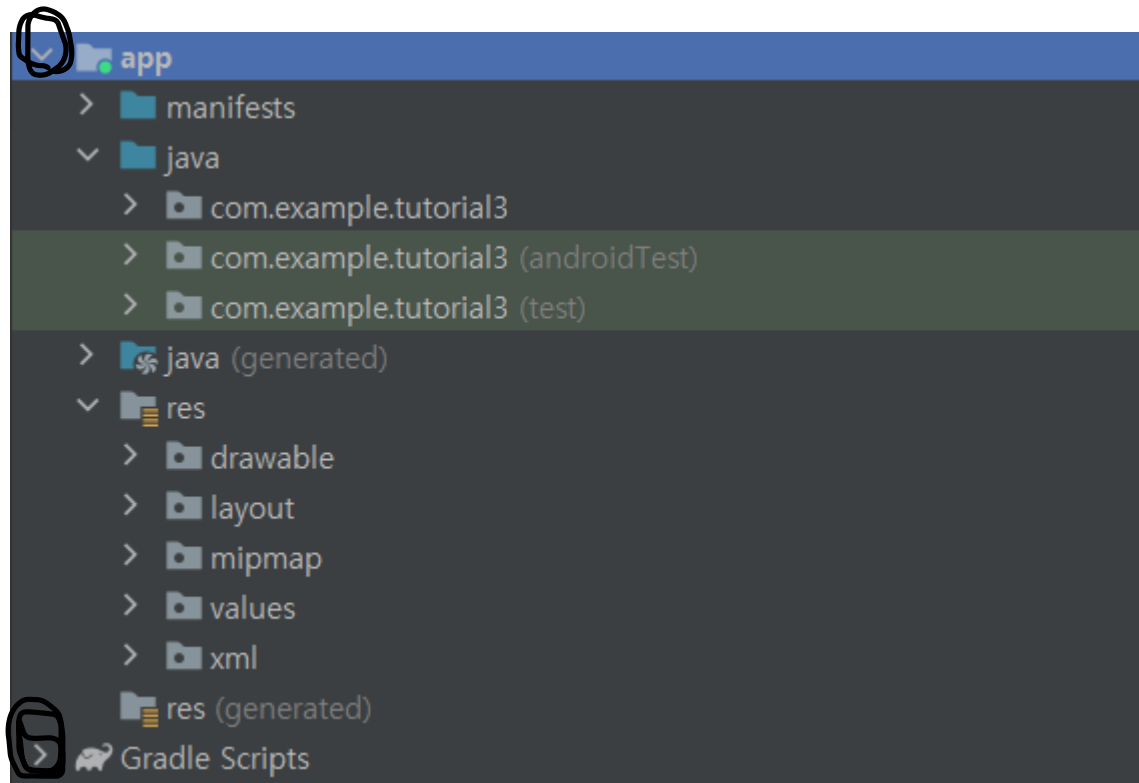


안드로이드 스튜디오 구조



프로젝트 파일

-app 폴더: 우리가 코딩해야 할 파일들

-Gradle Scripts 폴더: 어플리케이션 빌드에 필요한 설정 옵션, 라이브러리 정보들 저장

App 폴더

1.manifests: AndroidManifest.xml 파일, 컨트롤 타워, 어플리케이션 구동에 필요한 설정 값 관리

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="Tutorial3"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.Tutorial3"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
```

패키지명 표시

```
<application
```

안드로이드 프로젝트는 application 위에 activity가 실행되는 구조

```
android:icon="@mipmap/ic_launcher"
```

mipmap폴더 ic_launcher 파일을 앱 아이콘으로 설정,

아이콘 파일 이름 바뀌었을 때 여기서도 수정해주어야 함

```
android:label="Tutorial3"
```

아이콘에 표시될 앱 이름은 "Tutorial3"이다

```
android:theme="@style/Theme.Tutorial3"
```

어플리케이션에 Theme.Tutorial3라는 테마가 적용됨

```
<activity
```

사용자가 어떠한 활동 할 때 실행되는 컴포넌트

UI를 제공해 사용자와 상호작용을 하며, class에서 activity를 상속받아 사용.

여러 화면으로 구성되어 여러 액티비티로 구성됨

(로그인 화면, 메인화면 등이 각각의 액티비티이다.)

액티비티를 생성하면 kotlin or java 클래스 파일, xml파일이 같이 생성됨

New-activity로 추가 가능- xml, manifest 추가 등을 알아서 생성해줌

class파일 xml을 직접 추가해 사용 가능

2.Java 폴더 - java형식 파일 만들고 수정, xml파일에 만든 버튼 등의 UI에 기능을 붙여 여러 동작이 가능하게 함.

```
MainActivity.java
```

모든 프로그램에 있어야 하는 기본 파일(본체 느낌)

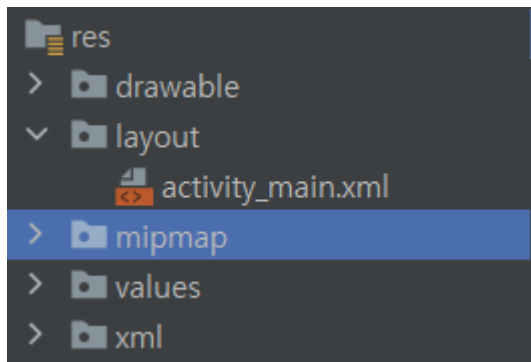
```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
}
```

setContentView파일 – layout(UI)패키지 아래 activity_main.xml파일을 View로 연결

따라서 Activity파일(Java)과 xml 파일은 한 쌍이다. 이 Activity파일에서 xml 파일을 연결해주고, xml파일은 UI를 구현하는 파일이다.

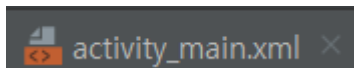
****새로운 화면 만들 때마다, xxxActivity.java 파일 만든 뒤 그 파일 안에서 activity_xxx.xml 파일을 연결해 활용한다.**

3.res 폴더 - resource폴더, UI관련 파일, 디자인이나 문자열 리소스 저장



drawable: 이미지 파일 저장 패키지, 이미지 파일 담음.

layout: 디자인의 뼈대를 담당하는 부분, 나머지 파일들(drawable, mipmap..)등은 부수적인 저장소일 뿐임.



여기서 design탭, text탭을 오가며 디자인을 구현

mipmap: 아이콘 이미지를 담음, 여기에 바꿀 아이콘 이미지를 담고, 파일명은 Manifest에서 mipmap부분 이름을 바꾸어 준다.

Values: 컬러 값 저장 파일, 색상 값, 수치 값, 스타일, 문자열 등을 저장