

● AndroidManifest.xml: 앱의 메인 환경 파일. 앱이 스마트폰에서 동작할 때 이 파일에 정의된 대로 동작한다. 개발자가 분석하고 수정하는 작업이 빈번하게 이루어진다.

→ package="com.example.user.androidlab" : 앱의 식별자 정보. 그레이들 파일에 설정된 applicationId의 값과 같다. 값은 유일무이해야 한다.

→ <application> : 앱의 구성요소를 등록하기 위한 태그. 안드로이드 컴포넌트 클래스는 생명주기를 안드로이드 시스템이 관리하기 때문에 AndroidManifest.xml에 <application> 태그의 하위 태그로 등록한다.

→ android:icon="@mipmap/ic_launcher", android:label="@string/app_name" : <application> 태그에 등록된 속성으로, 사용자 스마트폰이 앱에 설치되었을 때 앱의 아이콘 이미지와 앱의 이름을 명시하기 위한 속성이다. XML속성값이 @로 지정되었다는 건, res하위 폴더의 리소스들을 지칭한다.

→ <activity android:name=".MainActivity"> : 앱에 액티비티 컴포넌트를 등록하기 위한 태그. 하나만 있다면 앱이 하나의 액티비티로 구성된 앱이다. 만약 액티비티가 3개라면 AndroidManifest.xml파일에 3개의 <activity>태그가 정의되어야 한다. 또한, 서비스, 프로바이더, 리시버 컴포넌트가 있다면 <service>, <provider>, <receiver> 태그로 등록해 주어야 한다. <activity>태그에 액티비티를 등록할 때 name 속성은 생략할 수 없다. name 속성은 등록하고자 하는 클래스명을 명시하기 위한 속성으로 위의 코드는 MainActivity라는 클래스를 액티비티로 등록하는 구문이다.

→ <intent-filter> : 사용자가 앱의 아이콘을 클릭했을 때 실행되는 액티비티를 설정하는 용도다. 이것의 설정에 따라 사용자가 아이콘 클릭 시 MainActivity가 실행된다.

● MainActivity.java: 사용자가 앱을 클릭했을 때 실행되는 액티비티 클래스(화면 구성을 주목적으로 하는 클래스). 액티비티 클래스들은 Activity를 상속받아서 작성하는데, 디폴트로는 AppCompatActivity(Activity의 서브 클래스)를 상속받아 작성한다(public class MainActivity extends AppCompatActivity). 액티비티가 실행되면 자동으로 onCreate()함수가 호출되는데, 이 함수의 setContentView()함수가 화면 출력 함수이다. R.layout.activity_main을 매개변수로 지정하였으므로, res/layout/activity_main.xml의 구성대로 액티비티 화면이 출력된다.

● activity_main.xml: 이곳 XML에 정의된 대로 액티비티 화면이 출력된다.