

- app/src/main: 개발자는 이곳의 파일들을 이용해서 앱을 개발한다.
- ▶ manifests/AndroidManifest.xml: 앱의 메인 환경 파일이며, 이곳에 정의된 대로 앱이 실행된다. 개발자는 이 파일을 열어 분석하거나 수정하면서 앱을 개발한다.
- ▶ java: 개발자가 작성하는 모든 자바 파일(컴포넌트 파일 포함)은 java폴더에 위치한다.
 - MainActivity.java: 화면 구성을 위한 액티비티 컴포넌트로 실제 이 파일이 수행되어 화면에 UI가 출력된다.
- ▶ res: 모든 리소스 파일은 res 하위폴더에 위치한다.
 - res/drawable: 리소스 중 이미지 파일을 저장하기 위한 폴더이다.
 - res/layout: 리소스 중 UI 구성을 위한 레이아웃 XML 파일을 위한 폴더이다(activity_main.xml)
 - res/mipmap: 리소스 중 앱 아이콘 이미지를 위한 폴더이다.
 - res/values: 리소스 중 문자열 값 등을 위한 폴더이다.
- R.java: 모든 리소스 파일은 res폴더 하위에 있어야 한다. res 하위의 폴더는 이름이 지정되어 있으며, 개발자가 임의로 이름을 추가할 수 없다. 즉, 이미지는 res/drawable, 레이아웃 XML파일은 res/layout 폴더에 있어야 한다. 또한 각 폴더의 서브 폴더를 만들 수 없다. 개발을 하다 보면 수백 개의 리소스 파일이 만들어지는데, 코드 영역에서 이를 식별하도록 도움을 주는 파일이 R.java이다. 안드로이드 스튜디오 최신 버전에선 R.java파일을 직접 열 수 있는 방법은 없고, 다만 app - build - intermediate - apk - debug - app - debug.apk -> classes.dex -> androidx -> appcompat -> R\$id 순으로 클릭하여 정보를 확인할 수는 있다. R.java파일은 툴이 자동으로 만들어주며, R.java에는 int 형 변수가 나열되어 있다. 이는 res폴더의 리소스들을 식별하기 위한 변수이다. 즉, 개발자가 res 밑에 하나의 리소스 파일을 추가하면 해당 파일명으로 된 int형 변수가 R.java에 자동으로 추가된다. 그리고 자바 코드에서 이 변수(리소스 파일의 이름)로 리소스를 지칭해서 사용하는 구조이다. R.java에서 각각의 리소스를 내부 클래스명인 string, drawable, layout등으로 구분하고 있고, 각각의 변수는 대부분 파일명을 이용한다. 이런 이유로 res밑에 임의의 폴더를 생성하면 안 되며, 리소스 파일명도 자바 명명규칙을 위배할 수 없고 리소스 파일명에 대문자를 사용할 수 없다. 예를 들어 res/drawable에 새로운 이미지 파일 하나를 image1.png 이름으로 추가하면, 이 리소스를 식별하기 위한 변수가 R.java 파일에 자동으로 추가되며, 변수명은 리소스 파일명이 된다.

```
public static final class drawable {
    //...
    public static final int image1=0x7f020053;
    //...
}
```

즉, 리소스 파일을 추가하면 R.java 파일에 해당 리소스 파일명으로 int형 변수가 자동으로 추가되고, 자바 코드에서는 R.drawable.image1의 변수로 리소스를 식별해서 사용한다.

Resources: 강샘의 안드로이드 프로그래밍