

Ch02-처음 만드는 안드로이드 애플리케이션



2025-01-01(분반)

AI 오픈소스 전문프로젝트

Fri 09:00~ (S4-1-201)

학습목표

- 안드로이드 애플리케이션의 기본적인 작성법을 익힌다.
- AVD의 명칭을 파악한다.
- 안드로이드 프로젝트의 구성을 파악한다.

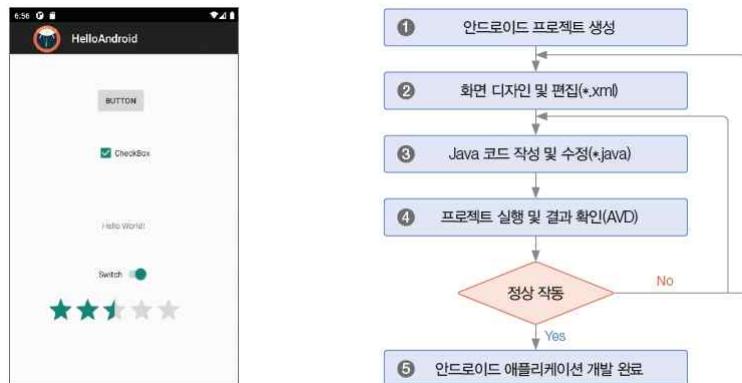
목차

- 01 처음 만드는 Hello Android 프로그램
- 02 AVD의 명칭과 사용법
- 03 완전한 기능의 안드로이드 애플리케이션 작성
- 04 안드로이드 프로젝트의 구성

01
처음 만드는
Hello Android 프로그램

1. 처음 만드는 Hello Android 프로그램

■ 첫 번째로 만들어볼 안드로이드 애플리케이션



5 / 80

1. 처음 만드는 Hello Android 프로그램

- 실습 2-1 첫 응용 프로그램 작성하기
- 1. 안드로이드 프로젝트 생성
 - (1) Android Studio 실행
 - (2) Android Studio 소개화면이 잠시 나온다.



그림 2-3 Android Studio 소개 화면

6 / 80

1. 처음 만드는 Hello Android 프로그램

- 실습 2-1 첫 응용 프로그램 작성하기
- 1. 안드로이드 프로젝트 생성
 - (3) 초기 창에서 왼쪽 [Projects] 선택 → <New Project> 클릭

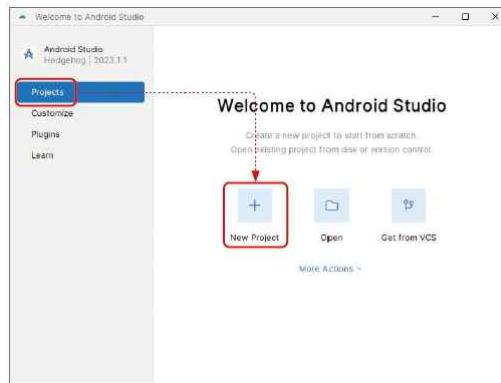


그림 2-4 새로운 안드로이드 프로젝트

7 / 80

1. 처음 만드는 Hello Android 프로그램

- 실습 2-1 첫 응용 프로그램 작성하기
- 1. 안드로이드 프로젝트 생성
 - (4) 먼저 생성할 프로젝트 종류 선택
 - [Phone and Tablet] → 'Empty Views Activity' 선택 → <Next> 선택

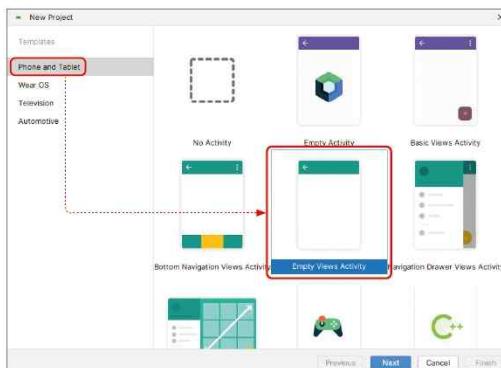


그림 2-5 프로젝트 선택

8 / 80

1. 처음 만드는 Hello Android 프로그램

■ 실습 2-1 첫 응용 프로그램 작성하기

■ 1. 안드로이드 프로젝트 생성

- (5) 프로젝트에 대한 정보를 입력하거나 선택
 - Name - 프로젝트 이름, 대문자로 시작하는 것이 좋다.
 - Package name - 도메인 이름과 애플리케이션 이름을 이어서 만듦
 - Save location - 프로젝트가 저장될 폴더를 지정(한글은 안 됨)
 - Language - 사용할 언어
 - Minimum SDK
 - 앱이 작동하는 최하 버전

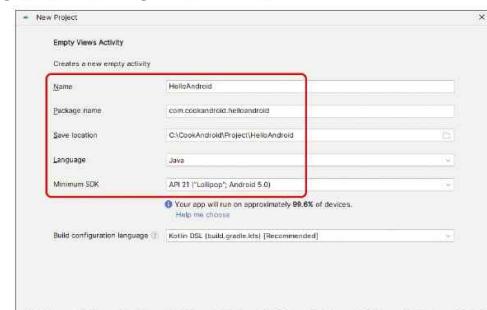


그림 2-6 프로젝트 설정

9 / 80

1. 처음 만드는 Hello Android 프로그램

■ 실습 2-1 첫 응용 프로그램 작성하기

■ 1. 안드로이드 프로젝트 생성

- (6) 2개 항목이 내부적으로 추가되어 있음(설정 중에는 생략)
 - Activity Name - 기본 소스인 Java 파일 이름으로 지정
 - Layout Name - 기본 화면인 XML 파일 이름으로 지정

■ Android 버전별 누적 점유율



그림 2-7 안드로이드 버전별 누적 점유율

10 / 80

1. 처음 만드는 Hello Android 프로그램

■ 실습 2-1 첫 응용 프로그램 작성하기

■ 1. 안드로이드 프로젝트 생성

- (7) 안드로이드 프로젝트 생성 화면

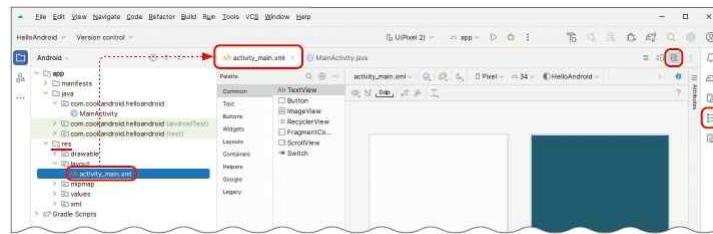


그림 2-8 안드로이드 프로젝트 생성 결과

11 / 80

1. 처음 만드는 Hello Android 프로그램

■ 실습 2-1 첫 응용 프로그램 작성하기

■ 2. 화면 디자인 및 편집

- (8) 가상 AVD와 화면을 동일하게 설정하기
 - ‘Pixel’ 클릭 – ‘AVD: U(Pixel 2) 선택 (생략 가능)

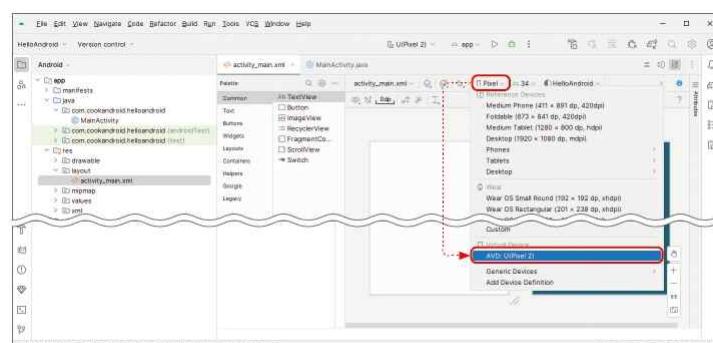


그림 2-9 디자인 환경 설정

12 / 80

1. 처음 만드는 Hello Android 프로그램

- 실습 2-1 첫 응용 프로그램 작성하기
- 2. 화면 디자인 및 편집
 - (9) 자동으로 생성되는 레이아웃은 ConstraintLayout임
 - 오른쪽 위의 [Code] 아이콘을 클릭하여 확인하기



그림 2-10 레이아웃 확인

13 / 80

1. 처음 만드는 Hello Android 프로그램

- 실습 2-1 첫 응용 프로그램 작성하기
- 2. 화면 디자인 및 편집
 - (10) 다시 [Design] 아이콘을 클릭하여 왼쪽 [Widgets]에서 몇 개를 오른쪽에 옮김
 - 각 위젯을 선택한 후 [Infer Constraints] 아이콘 클릭
 - 오른쪽 아래의 확대/축소 아이콘으로 화면 크기 조절 가능

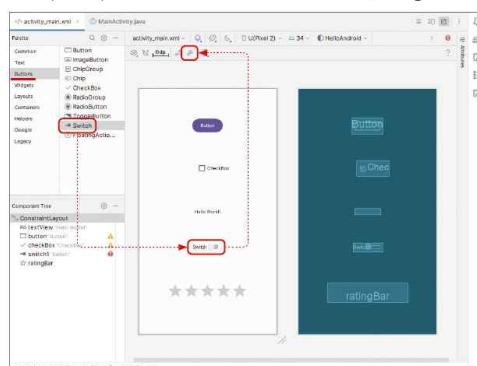


그림 2-11 그래픽 화면에서 위젯 끌어다 놓기

14 / 80

1. 처음 만드는 Hello Android 프로그램

■ 실습 2-1 첫 응용 프로그램 작성하기

■ 2. 화면 디자인 및 편집

- (11) 오른쪽 위의 [Split] 아이콘을 클릭하면 화면에 XML 코드와 미리보기가 표시됨
 - 추가된 적당한 위젯 코드 하나 삭제 (이 실습에서는 Button 삭제)



그림 2-12 XML 코드 변경

15 / 80

1. 처음 만드는 Hello Android 프로그램

■ 실습 2-1 첫 응용 프로그램 작성하기

■ 2. 화면 디자인 및 편집

- (12) 다시 [Design] 아이콘 클릭
 - 방금 삭제한 코드(버튼)가 화면에서도 삭제됨을 확인
 - [File]-[Save All] 클릭하여 변경사항 저장

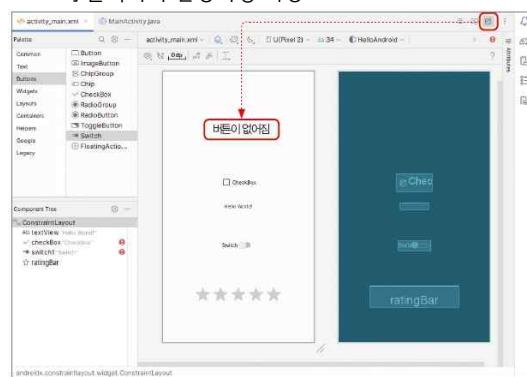


그림 2-13 그래픽 화면 확인

16 / 80

1. 처음 만드는 Hello Android 프로그램

■ 실습 2-1 첫 응용 프로그램 작성하기

■ 3. Java 코드 작성 및 수정

- (13) 원쪽 Project Tree의 [java]-[com.cookandroid.helloandroid]-[MainActivity] 더블 클릭 혹은 위쪽 [MainActivity.java] 탭 클릭

The screenshot shows the Android Studio interface. On the left is the Project Tree, which includes an Android folder containing app, res, and Java subfolders. Inside Java, there's a package com.cookandroid.helloandroid with a file named MainActivity. On the right is the code editor showing MainActivity.java:

```
activity_main.xml>MainActivity.java
1 package com.cookandroid.helloandroid;
2
3 import android.os.Bundle;
4 import android.app.Activity;
5 import android.widget.TextView;
6
7 public class MainActivity extends AppCompatActivity {
8
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13    }
14 }
```

그림 2-14 메인 Java 코드 확인

17 / 80

1. 처음 만드는 Hello Android 프로그램

■ 실습 2-1 첫 응용 프로그램 작성하기

■ 4. 프로젝트 실행 및 결과 확인

- (14) 메뉴의 [Run As]-[Run 'app']을 선택 혹은 [Run 'app'] 아이콘 클릭

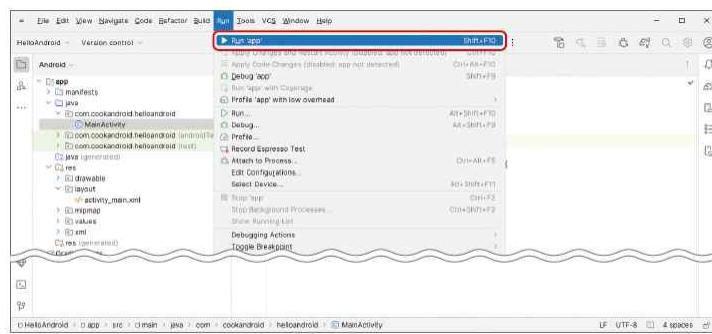


그림 2-15 애플리케이션 실행

18 / 80

1. 처음 만드는 Hello Android 프로그램

- 실습 2-1 첫 응용 프로그램 작성하기
- 4. 프로젝트 실행 및 결과 확인
 - (15) AVD가 여러 개이거나 스마트폰이 연결된 경우
 - 먼저 작동할 장치를 상단 중앙 툴바에서 선택

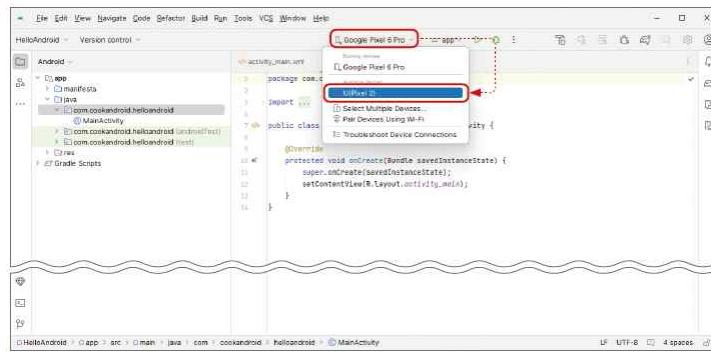


그림 2-16 실행할 안드로이드 장치 선택

19 / 80

1. 처음 만드는 Hello Android 프로그램

- 실습 2-1 첫 응용 프로그램 작성하기
- 4. 프로젝트 실행 및 결과 확인
 - (16) 실행 결과 화면 확인



그림 2-17 프로젝트 실행 결과

20 / 80

1. 처음 만드는 Hello Android 프로그램

- 실습 2-1 첫 응용 프로그램 작성하기
- 4. 프로젝트 실행 및 결과 확인
 - (17) AVD의 초기화면으로 돌아가기
 - 키패드의 돌아가기 버튼 클릭
 - AVD에서 [ESC] 누르기

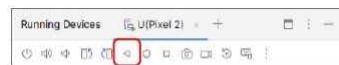


그림 2-18 돌아가기 버튼

▶ 직접 풀어보기 2-1

[새로운 프로젝트 'HiAndroid'를 작성하고 실행한 뒤 화면을 꾸며보라.]

21 / 80

1. 처음 만드는 Hello Android 프로그램

여기서 점검 AVD에서 앱 아이콘을 출력하는 방법

- ➊ 먼저 아이콘으로 사용할 파일이 필요하다. 일반적으로 128x128 정도의 *.png 파일을 사용하는 것이 좋다.
- ➋ 파일을 복사하여 Project Tree의 [app] -> [res] -> [drawable]에 붙여넣기 한다. [Copy] 창이 나오면 <OK>를 클릭 한다.

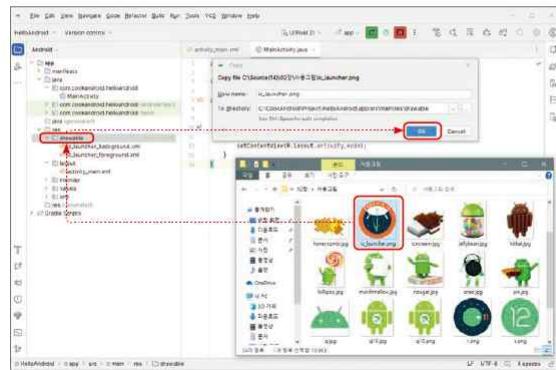


그림 2-19 아이콘 이미지를 drawable에 복사

22 / 80

1. 처음 만드는 Hello Android 프로그램

④ Project Tree에서 [app]→[java]→[com.cookandroid 앱 이름]→[MainActivity]를 더블클릭해서 열고 setContentView(R.layout.activity_main) 행 아래에 다음 두 행을 코딩한다.

```
getSupportActionBar().setDisplayHomeAsUpEnabled(true);  
getSupportActionBar().setIcon(R.drawable.ic_launcher);
```



그림 2-20 메인 Java 코드 수정

⑤ Project Tree에서 [app]→[manifests]→[AndroidManifest.xml]을 더블클릭해서 열고 13행쯤의 android:theme를 다음과 같이 수정한다.

```
android:theme = "@style/Theme_암이온"  
→ android:theme = "@style/Theme.AppCompat.Light.DarkActionBar"
```

이제 프로젝트를 실행하면 AVD 왼쪽 상단에 앱 아이콘이 출력될 것이다.

23 / 80

02

AVD의 명칭과 사용법

1. AVD의 명칭

■ AVD의 기본적인 명칭



그림 2-21 AVD의 명칭

25 / 80

1. AVD의 명칭

■ AVD의 기본적인 명칭

- 돌아가기(Back) 버튼 : 키보드 [Ctrl]+[Back Space]와 동일한 기능
- 홈(Home) 버튼 : 키보드 [Ctrl]+[H]와 동일한 기능
- 오버뷰(Overview) 버튼 : 실행 중인 앱이 보이고, 전환 혹은 종료 가능
- 상태 바(Status Bar) : 3G 수신율, 배터리 상태, 시간 등 나타냄
- 타이틀 바(Title Bar) : 안드로이드 프로젝트 이름 나타냄

26 / 80

2. AVD 사용법

■ 실습 2-2 AVD 사용법 알아보기

■ 화면 전환

- (1) AVD를 초기화면 상태로 두고 초기화면이 아니라면 홈버튼 클릭
- (2) [Zoom to Fit Screen] 아이콘 클릭 → 화면 크기에 딱 맞도록 AVD가 조절됨
- (3) AVD 위쪽의 화면 회전 아이콘을 클릭하면 화면이 좌우로 회전



그림 2-22 초기화면

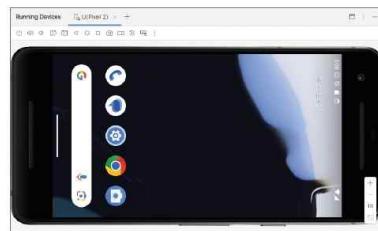


그림 2-23 회전된 화면

27 / 80

2. AVD 사용법

■ 실습 2-2 AVD 사용법 알아보기

■ 알람 추가

- (1) [애플리케이션] 화면 - 시계 버튼 클릭 클릭
- (2) 알람 추가 화면 생성
- (3) 아래쪽의 < + > 클릭하여 알람 시간 추가



그림 2-24 알람 시간 추가



28 / 80

2. AVD 사용법

■ 실습 2-2 AVD 사용법 알아보기

■ 카메라와 갤러리

▪ (1) 초기화면 - 카메라 버튼 클릭

- 사진 위치 기록 메시지 창 : <다음> 클릭
- 액세스 허용 여부 메시지 : <앱 사용 중에만 허용> 클릭
- 실제 카메라는 없지만 가상의 카메라 화면을 보여줌
- 셔터 버튼 클릭 : 가상 화면이 촬영됨



그림 2-25 가상 카메라 촬영

29 / 80

2. AVD 사용법

■ 실습 2-2 AVD 사용법 알아보기

■ 카메라와 갤러리

▪ (2) 초기화면 - 위쪽 스와이프 - [포토] 클릭

- 돌아가기 버튼 클릭 : 조금 전에 촬영한 화면 보여줌
- 사진 편집/변경 및 삭제, 공유 등의 작업 가능



그림 2-26 촬영된 사진 확인

30 / 80

2. AVD 사용법

■ 실습 2-2 AVD 사용법 알아보기

■ 웹 브라우저

- (1) 초기화면에서 크롬 버튼 클릭
- (2) 구글 사이트에 접속하여 ‘웹 주소 검색 또는 입력’ 부분에 주소 입력하고 [Enter]입력

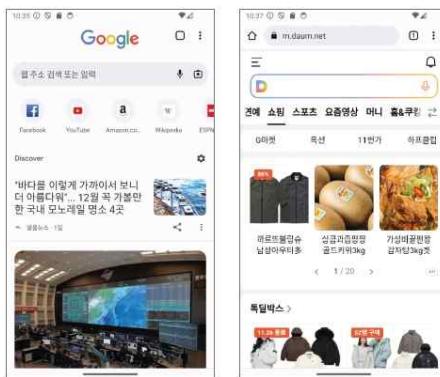


그림 2-27 웹 서핑

31 / 80

2. AVD 사용법

■ 실습 2-2 AVD 사용법 알아보기

■ 배경화면 변경

- (1) 바탕화면을 몇 초 동안 누르고 [배경화면 및 스타일] 클릭 - [배경화면 더 보기] 선택
- (2) 그림 선택 - <배경화면 설정> 클릭 - <홈 화면 및 잠금 화면> 클릭



그림 2-28 배경화면 변경

32 / 80

2. AVD 사용법

■ 실습 2-2 AVD 사용법 알아보기

■ 오버뷰

- 실행 중인 전체 앱을 보여주고 다시 앱을 화면 앞으로 가져오거나 종료하는 기능 수행

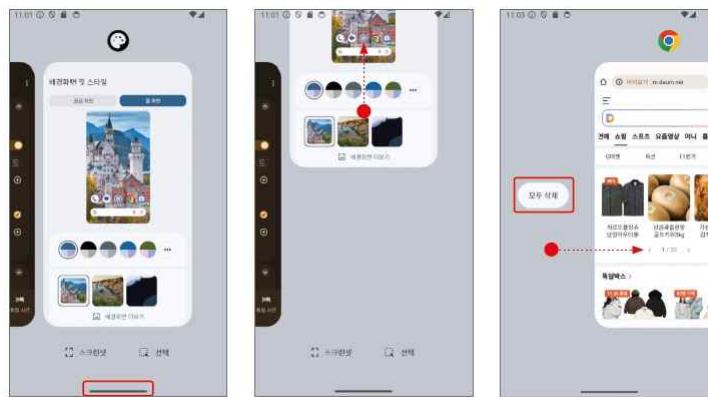


그림 2-29 실행 중인 프로그램 종료

33 / 80

2. AVD 사용법

■ 실습 2-2 AVD 사용법 알아보기

■ 자주 사용하는 애플리케이션 바탕화면에 복사

- (1) HelloAndroid 아이콘 몇 초 동안 누르기 - <앱 정보> 확인 가능
- (2) 아이콘을 몇 초 동안 누르고 위로 움직여 바탕화면으로 복사

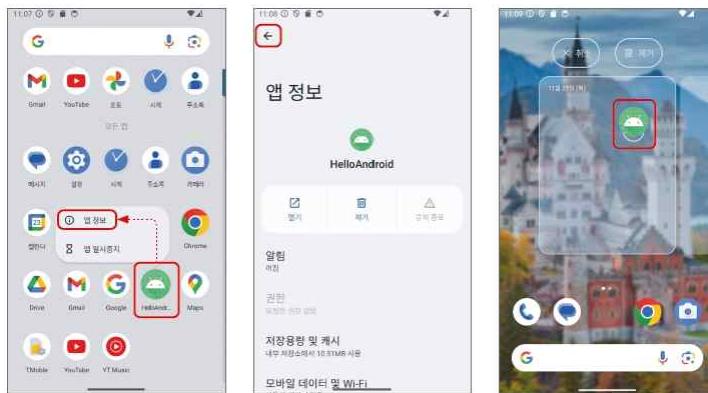


그림 2-30 바탕화면에 애플리케이션 복사

34 / 80

2. AVD 사용법

- 실습 2-2 AVD 사용법 알아보기
- 자주 사용하는 애플리케이션 바탕화면에 복사
 - (3) 바탕화면의 아이콘을 몇 초 동안 누르고- 상단의 <삭제>로 이동하여 아이콘 삭제
 - 애플리케이션이 삭제되는 것은 아님



그림 2-31 바탕화면의 아이콘 삭제

35 / 80

03 완전한 기능의 안드로이드 애플리케이션 작성

1. Android Studio 프로젝트 관리

- 실습 2-3 안드로이드 프로젝트 관리하기
- 프로젝트 닫기와 열기
 - (1) [File]-[Close Project]를 선택 : 프로젝트 닫음
 - 오른쪽 프로젝트 목록 : 지금까지 작업한 프로젝트 목록
 - 프로젝트 이름 클릭 : 다시 프로젝트 열림

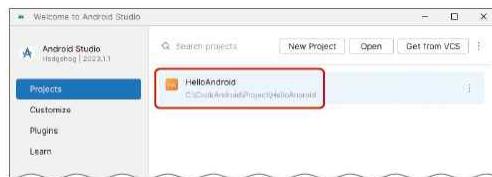


그림 2-32 최근에 사용한 프로젝트 열기 1

37 / 80

1. Android Studio 프로젝트 관리

- 실습 2-3 안드로이드 프로젝트 관리하기
- 프로젝트 닫기와 열기
 - (2) 여러 개의 프로젝트 열고 작업하기
 - [Open] 클릭 - 프로젝트 폴더 선택
 - <New Window> 클릭 : 새로운 창이 열려서 여러 개의 프로젝트 동시 작업 가능



그림 2-33 최근에 사용한 프로젝트 열기 2

- (3) [File]-[Close Project] : 열린 창 모두 닫기

38 / 80

1. Android Studio 프로젝트 관리

- 실습 2-3 안드로이드 프로젝트 관리하기
- 프로젝트 닫기와 열기
 - (4) 폴더 검색하여 프로젝트 열기
 - [Open an Existing Project] 클릭 - 프로젝트가 저장된 폴더 지정

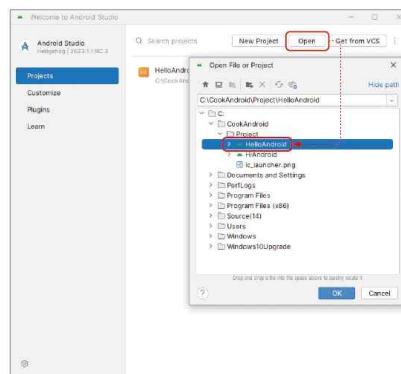


그림 2-34 폴더에서 직접 프로젝트 열기

39 / 80

1. Android Studio 프로젝트 관리

- 실습 2-3 안드로이드 프로젝트 관리하기
- Android Studio 프로젝트 내보내기/가져오기
 - 내보내기/가져오기 기능이 따로 없음
 - 프로젝트가 생성된 폴더를 통째로 복사하거나 압축하여 보내거나 가져오는 방식

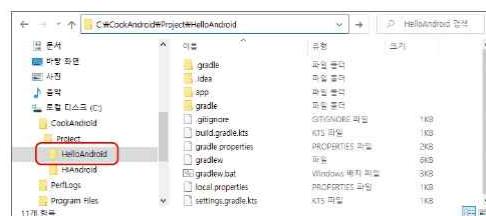


그림 2-35 프로젝트 폴더를 통째로 복사

40 / 80

1. Android Studio 프로젝트 관리

- 실습 2-3 안드로이드 프로젝트 관리하기
- 이전 버전의 프로젝트 가져오기
 - (1) Android Studio 초기화면의 [Open]을 선택
→ 이전 버전의 Android Studio에서 작성된 프로젝트 폴더 선택

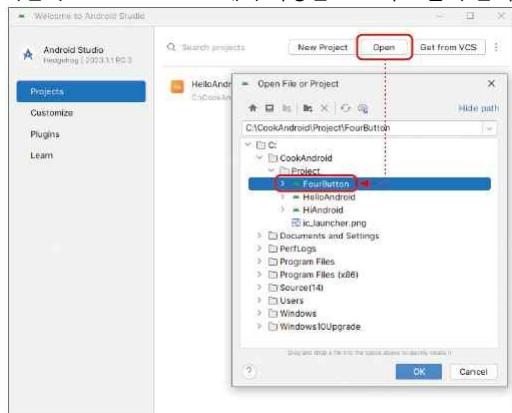


그림 2-36 이전 버전의 프로젝트 가져오기 1

41 / 80

1. Android Studio 프로젝트 관리

- 실습 2-3 안드로이드 프로젝트 관리하기
- Eclipse용 프로젝트 가져오기
 - (2) [Trust and Open Project '프로젝트이름'?] 창에서 <Trust Project> 클릭



그림 2-37 이전 버전의 프로젝트 가져오기 2

- (3) 오류 메시지가 나오면 일단 <OK>를 클릭 진행
→ 오류 메시지의 내용을 보고 프로젝트 수정



그림 2-38 이전 버전의 프로젝트 가져오기 3

42 / 80

1. Android Studio 프로젝트 관리

- 실습 2-3 안드로이드 프로젝트 관리하기
- Eclipse용 프로젝트 가져오기
 - (4) Android Studio에서 생성한 프로젝트와 같은 방법으로 사용

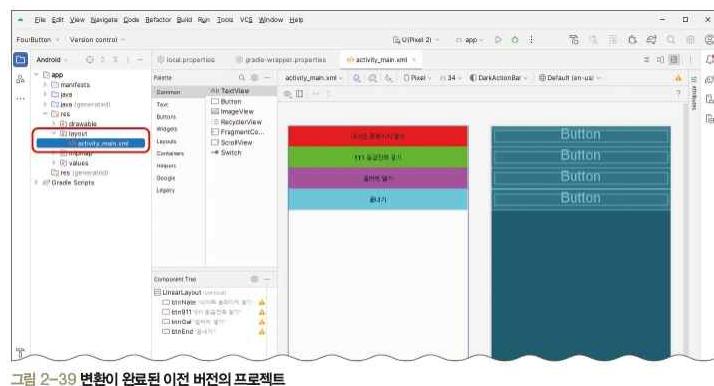


그림 2-39 변환된 원래의 이전 버전의 프로젝트

43 / 80

1. Android Studio 프로젝트 관리

▶ 직접 들어보기 2-2
[이전 버전의 Android Studio에서 작성한 프로젝트를 Android Studio로 가져오라. (이전 버전의 프로젝트가 없다면 생략해도 된다.)]

44 / 80

2. 안드로이드 프로젝트의 표준 틀

- Android Project 사용법과 자동 완성 기능
- 화면 구성을 위한 XML 파일의 문법
- Java 코드 코딩 방법
 - activity_main.xml 파일에 포함된 위젯의 접근 방법
 - 위젯의 이벤트 발생 시 작동하는 코드 작성법
- R.java의 내용



그림 2-40 안드로이드 프로젝트 표준 틀 실행 결과

45 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 1 안드로이드 프로젝트 생성
 - (1) [File] - [New Project] 선택 or [초기화면]의 [New Project] 클릭

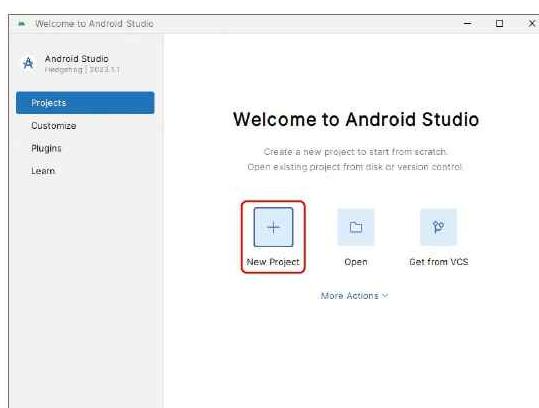


그림 2-41 새 프로젝트 생성

46 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 1 안드로이드 프로젝트 생성
 - (2) [Template] - [Phone and Tablet] - [Empty Views Activity] 선택 - [Next] 클릭
 - Activity란 스마트폰 화면을 의미함

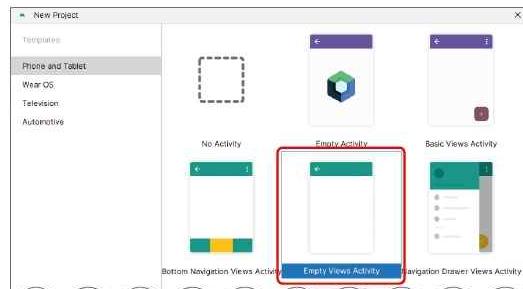


그림 2-42 프로젝트 형태 선택

47 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 1 안드로이드 프로젝트 생성
 - (3) [Empty Activity]에서 정보 입력
 - Name만 입력해도 나머지 자동 완성

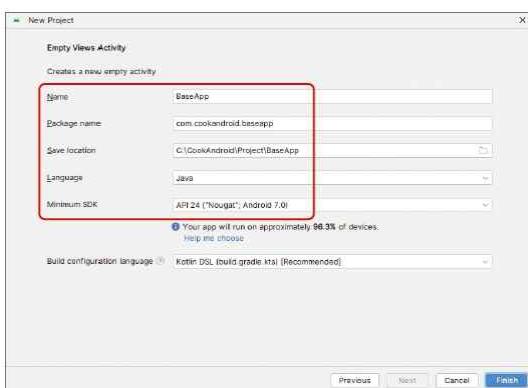


그림 2-43 프로젝트 환경 설정

48 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 2 화면 디자인 및 편집
 - (1) Project Tree에서 [java]-[com.cookandroid.baseapp]-[MainActivity]가 기본적으로 열린 상태
 - (2) 상단의 [activity_main.xml] 클릭 - 오른쪽 위의 [Design] 아이콘 클릭

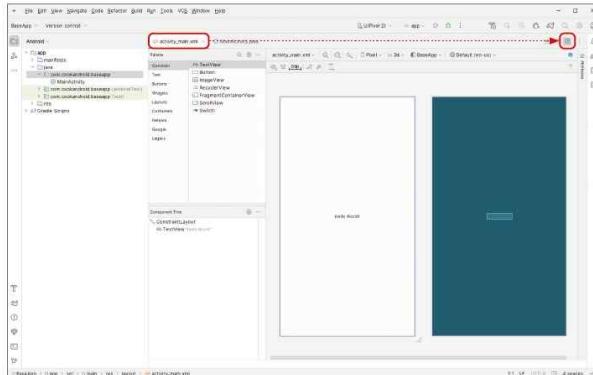


그림 2-44 activity_main.xml 확인

49 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 2 화면 디자인 및 편집
 - (3) 화면 오른쪽 위의 [Code] 아이콘 클릭 : XML 코드 확인
 - 주로 XML 코드로 편집
 - 프로젝트의 기본 레이아웃 : <LinearLayout>



그림 2-45 XML 코드 변경

50 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 2 화면 디자인 및 편집
 - (4) <LinearLayout> … </LinearLayout> 사이에 버튼 추가
 - <Button> 입력



The screenshot shows the Android Studio interface with two tabs: 'activity_main.xml' and 'MainActivity.java'. The XML tab displays the following code:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Button" />

</LinearLayout>
```

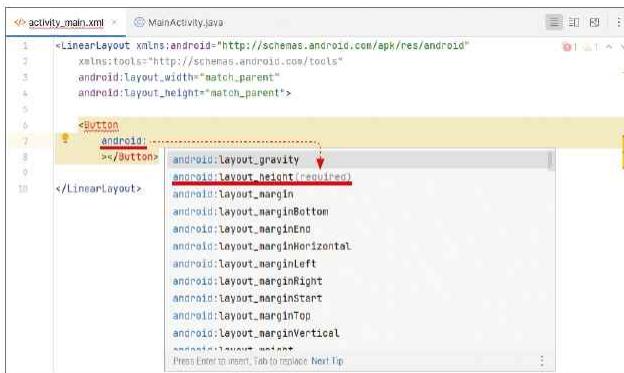
A red box highlights the opening tag of the `<Button>` element.

그림 2-46 버튼 추가

51 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 2 화면 디자인 및 편집
 - (5) “android:” 입력 – 목록에서 ‘layout_height’ 더블클릭



The screenshot shows the Android Studio code editor with the XML tab selected. The code is identical to the previous screenshot, but the cursor is positioned on the `layout_height` attribute of the `<Button>` element. A tooltip or auto-completion dropdown is open, showing the following options:

- android:layout_gravity
- android:layout_height (required)
- android:layout_margin
- android:layout_marginBottom
- android:layout_marginEnd
- android:layout_marginHorizontal
- android:layout_marginLeft
- android:layout_marginRight
- android:layout_marginStart
- android:layout_marginTop
- android:layout_marginVertical
- android:layout_weight

The option `android:layout_height (required)` is highlighted with a red box.

그림 2-47 버튼 속성 편집

52 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 2 화면 디자인 및 편집
 - (6) 자동 팝업 창 - 'wrap_content' 선택 - [Enter]
 - 자동 팝업 창이 뜨지 않으면, [Ctrl]-[Space Bar]

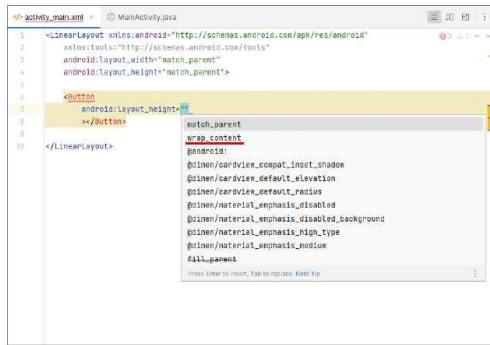


그림 2-48 버튼 속성 중 layout_height 편집

53 / 80

2. 안드로이드 프로젝트의 표준 틀

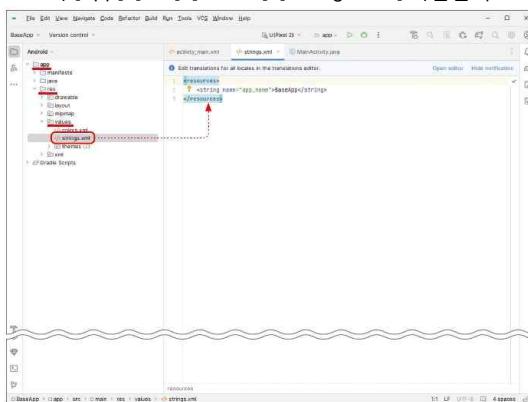
- 실습 2-4 기본적인 애플리케이션 작성하기
- 2 화면 디자인 및 편집
 - (7) 나머지 코드 입력하고 저장
 - 최종 코드 : [예제 2-1]

```
예제 2-1 activity_main.xml
1 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
2     xmlns:tools="http://schemas.android.com/tools"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent" >
5
6
7     <Button
8         android:layout_width="match_parent"
9         android:layout_height="wrap_content"
10        android:id="@+id/button1"
11        android:text="@string/strBtn1"
12    ></Button>
13
14 </LinearLayout>
```

54 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 2 화면 디자인 및 편집
 - (8) activity_main.xml의 11행에 오류 발생
 - Project Tree의 [app]-[res]-[values]-[strings.xml] 더블클릭: xml코드 확인



2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 2 화면 디자인 및 편집
 - (9) strBtn1 문자열 추가하고 저장

예제 2-2 strings.xml

```
1 <resources>
2     <string name="app_name">BaseApp</string>
3     <string name="strBtn1">버튼입니다</string>
4 </resources>
```

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 2 화면 디자인 및 편집
 - (10) activity_main.xml에서 오른쪽 위의 [Design] 아이콘 클릭 : 추가된 버튼 확인

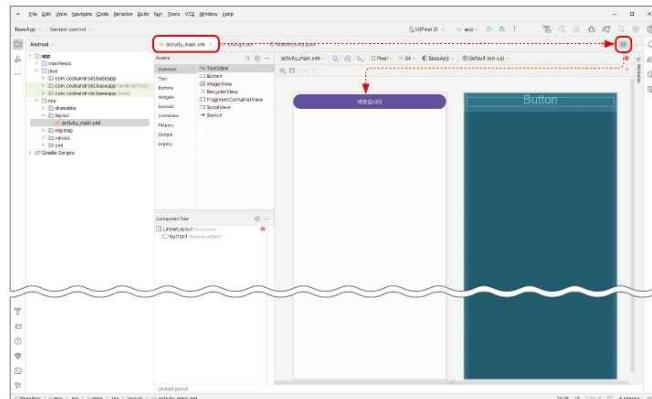


그림 2-50 activity_main.xml 코드의 그래픽화면

57 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 2 화면 디자인 및 편집
 - (10) activity_main.xml에서 오른쪽 위의 [Design] 탭 클릭 : 추가된 버튼 확인

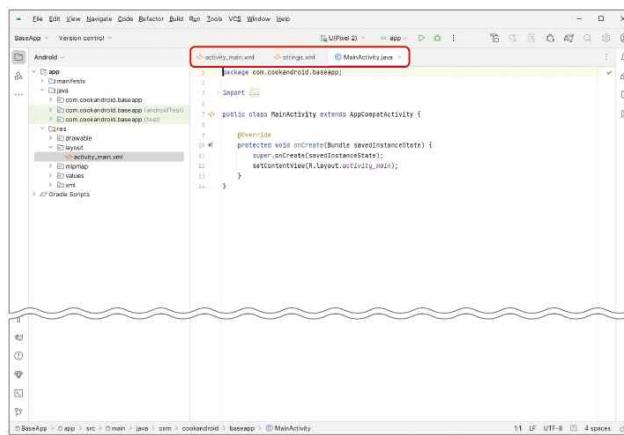


그림 2-51 엘린 파일을 탭으로 전환

58 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 2 화면 디자인 및 편집
 - (11) [Run] - [Run 'app'] 선택 혹은 [Run 'app'] 아이콘 클릭

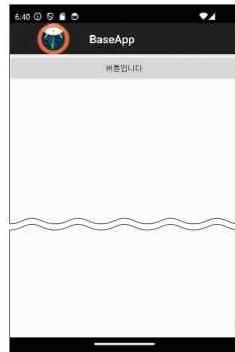


그림 2-52 프로젝트 실행

59 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 3 Java 코드 작성 및 수정
 - (1) 버튼을 눌렀을 때 작동이 일어나도록 코드 작성
 - (2) Project Tree의 [java]-[패키지 이름]-[MainActivity] 더블클릭

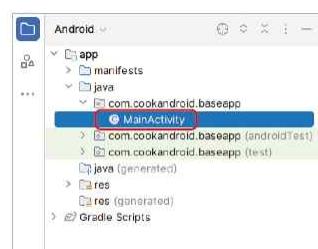


그림 2-53 Java 파일 선택

60 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 3 Java 코드 작성 및 수정
 - (3) 기본적인 코드 구성



```
activity_main.xml strings.xml MainActivity.java
1 package com.cookandroid.baseapp;
2
3 > import ...
4
5 <public class MainActivity extends AppCompatActivity {
6
7     @Override
8     protected void onCreate(Bundle savedInstanceState) {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.activity_main);
11    }
12}
13
14}
```

그림 2-54 MainActivity.java의 기본 코드

61 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 3 Java 코드 작성 및 수정
 - (4) 3행의 import 앞에 작은 ‘>’ 기호 클릭 : 행 확장
 - (5) [ctrl]+[alt]+[o] (알파벳 ‘o’) : 불필요하게 임포트된 문장 제거



```
activity_main.xml strings.xml MainActivity.java
1 package com.cookandroid.baseapp;
2
3 <import androidx.appcompat.app.AppCompatActivity;
4 <import android.os.Bundle;
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}
```

그림 2-55 필요 없는 import문 삭제

62 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
 - 3 Java 코드 작성 및 수정
 - (6) Button 변수 추가
 - 멤버변수 위치인 @Override 바로 위에 다음을 입력

```
Button button1;
```

```
activity_main.xml    strings.xml    MainActivity.java    E

1 package com.cookandroid.baseapp;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.os.Bundle;
6
7 android.widget.Button? Alt+Enter AppCompatActivity {
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }
```

그림 2-56 Button 변수 추가

63 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
 - 3 Java 코드 작성 및 수정
 - (7) [Alt]+[Enter] : Button과 관련된 클래스가 자동으로 import문에 추가된다.

```
activity_main.xml    strings.xml    MainActivity.java    A C
```

```
1 package com.cookandroid.baseapp;
2
3 import androidx.appcompat.app.AppCompatActivity;
4
5 import android.os.Bundle;
6 import android.widget.Button;
7
8 public class MainActivity extends AppCompatActivity {
9     no images
10     Button button;
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15     }
16 }
```

그림 2-57 자동 import 확인

64 / 80

2. 안드로이드 프로젝트의 표준 틀

■ 실습 2-4 기본적인 애플리케이션 작성하기

■ 3 Java 코드 작성 및 수정

- (8) activity_main.xml 파일에서 만든 객체에 접근 : findViewById() 메소드 사용
 - setContentView() 메소드 바로 아래에 다음 코드 추가

```
button1 = (Button) findViewById(R.id.button1);
```

- (9) 이벤트 메소드 정의

- 'button1.setOnClickListener()' 까지만 입력 - 팝업 창 맨 뒤의 'setOnClickListener()' 선택



그림 2-58 Java 코드 자동 완성 1

65 / 80

2. 안드로이드 프로젝트의 표준 틀

■ 실습 2-4 기본적인 애플리케이션 작성하기

■ 3 Java 코드 작성 및 수정

- (10) setOnClickListener() 괄호 안에 "new View" 입력 - OnClickListener()선택



그림 2-59 Java 코드 자동 완성 2

66 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 3 Java 코드 작성 및 수정
 - (11) onClick() 메소드 자동 완성

```
14     setContentView(R.layout.activity_main);
15     button1 = (Button) findViewById(R.id.button1);
16
17     button1.setOnClickListener(new View.OnClickListener() {
18         @Override
19         public void onClick(View view) {
20             ...
21         }
22     });
23 }
24 }
```

그림 2-60 자동 완성된 코드

67 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 3 Java 코드 작성 및 수정
 - (12) 버튼 클릭 시 작동하길 원하는 이벤트를 onClick() 메소드 안에 입력

```
예제 2-3 MainActivity.java
1 package com.cookandroid.baseapp;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import android.os.Bundle;
5 import android.view.View;
6 import android.widget.Button;
7 import android.widget.Toast;
8
9 public class MainActivity extends AppCompatActivity {
10     Button button1;
11
12     @Override
13     public void onCreate(Bundle savedInstanceState) {
14         super.onCreate(savedInstanceState);
15         setContentView(R.layout.activity_main);
16
17         button1 = (Button) findViewById(R.id.button1);
18         button1.setOnClickListener( new View.OnClickListener() {
```

68 / 80

2. 안드로이드 프로젝트의 표준 틀

```
19     @Override  
20     public void onClick(View view) {  
21  
22         Toast.makeText(getApplicationContext(), "버튼을 눌렀어요",  
23             Toast.LENGTH_SHORT).show();  
24     }  
25 }  
26 }
```

69 / 80

2. 안드로이드 프로젝트의 표준 틀

- 실습 2-4 기본적인 애플리케이션 작성하기
- 4 프로젝트 실행 및 결과 확인
 - (1) [File]-[Save All] 선택 : 현재까지의 내용 모두 저장
 - (2) [Run]-[Run 'app']을 선택 혹은 [Run 'app'] 아이콘을 클릭 - 버튼 클릭

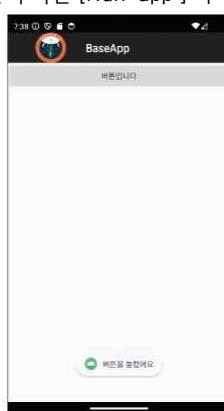


그림 2-61 실행 결과

70 / 80

2. 안드로이드 프로젝트의 표준 틀

여기서 참조 R 클래스

레이아웃 파일(C activity_main.xml), strings.xml에 들어 있는 문자열, 버튼 등의 아이디, 이미지 파일의 아이디 등은 모두 R 클래스에 자동으로 등록된다. R 클래스는 C:\CookAndroid\Project\BaseApp\app\build\intermediates\compile_and_runtime_not_namespaced_class.jar\debug 폴더에 R.jar 파일로 존재한다. 컴파일된 파일이라서 열어볼 수는 없지만, 이 파일에 등록된 위젯은 다음 그림과 같이 계층형으로 저장되어 있다. [예제 2-3]의 17행에서 버튼에 접근하는 데 R.id.button1을 사용했다.

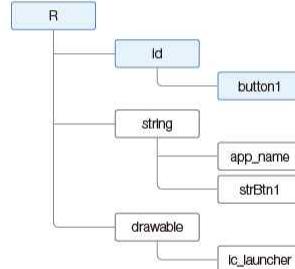


그림 2-62 R.jar 파일의 구조

그 외에 strings.xml 파일에 추가했던 문자열도 R.string.strBtn1로 접근이 가능하다. 나중에는 그림 파일도 사용할 텐데, 그림 파일은 'R.drawable.' 그림 파일 아이디('id') 형식으로 사용할 수 있다. [예제 2-1]에서 activity_main.xml의 10 행에 나온 android:id="@+id/button1"과 같은 형식이 등록되면 이것을 Java 코드에서 R.id.button1로 접근할 수 있다고 생각하면 된다.

71 / 80

2. 안드로이드 프로젝트의 표준 틀

■ 안드로이드 애플리케이션의 큰 윤곽과 작성법

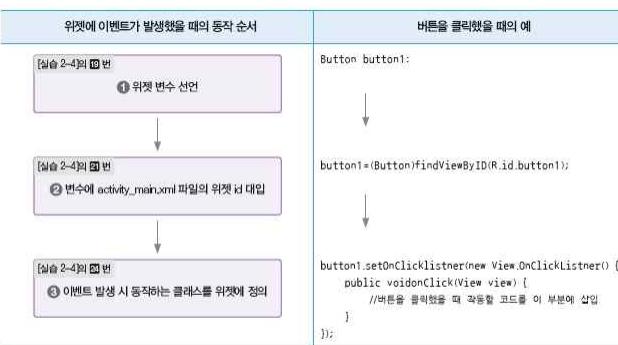


그림 2-63 위젯에 이벤트를 적용하기 위한 코딩 요약

72 / 80

2. 안드로이드 프로젝트의 표준 틀

■ 안드로이드 애플리케이션의 큰 윤곽과 작성법

- 코드 응용
- activity_main.xml 파일에 체크박스를 추가하고 체크박스를 클릭할 때마다 작동하기

```
// ① 체크박스 변수 선언  
CheckBox check1;  
  
// ② 변수에 activity_main.xml 파일의 체크박스 대입  
check1 = (CheckBox) findViewById(R.id.check1);  
  
// ③ 체크박스를 클릭했을 때 작동하는 클래스 정의  
check1.setOnClickListener( new View.OnClickListener() {  
    public void onClick(View view) {  
        // 체크박스를 클릭했을 때 작동할 코드를 이 부분에 삽입  
    }  
});
```

73 / 80

2. 안드로이드 프로젝트의 표준 틀

▶ 직접 풀어보기 2-3
다음 그림과 같이 버튼 4개를 만들고 각 버튼을 클릭하면 필요한 내용이 작동하는 FourButton 프로젝트를 작성하라. 각 버튼은 다른 색상으로 변경한다.



그림 2-64 FourButton 프로젝트 실행 결과

74 / 80

04

안드로이드 프로젝트의 구성

1 BaseApp 프로젝트의 구성

■ Project Tree의 구성 살펴보기



그림 2-65 안드로이드 프로젝트의 구성

2 프로젝트에서 사용되는 폴더 및 파일의 용도

■ java 폴더

- 하위에 패키지명의 하위 폴더가 존재
 - 안드로이드 프로젝트를 생성할 때 입력한 패키지 이름과 동일
- 패키지 이름 아래에 MainActivity.java로 메인 Java 소스가 들어 있음
 - 주로 액티비티에서 할 일을 프로그래밍

■ java (generated) 폴더

- Android Studio 3.2부터 제공되는 폴더
 - 내부적으로 사용

77 / 80

2 프로젝트에서 사용되는 폴더 및 파일의 용도

■ res 폴더

- 앱 개발에 사용되는 이미지, 레이아웃, 문자열 등이 들어가는 폴더
- drawable 폴더 : 이미지 파일 저장
- layout 폴더 : 액티비티(화면)을 구성하는 xml 파일을 넣으면 됨
- values 폴더 : 문자열을 저장하는 string.xml, 색상표를 저장하는 colors.xml, 스타일을 저장하는 themes.xml 등이 들어 있음
- menu 폴더 : 메뉴 XML 파일이 저장되어 있음

■ res(generated) 폴더

- Android Studio 3.5부터 제공
 - 내부적으로 사용

78 / 80

2 프로젝트에서 사용되는 폴더 및 파일의 용도

■ manifests 폴더

- AndroidManifest.xml 파일이 들어 있음
- 앱의 여러 가지 정보를 담고 있음

■ Gradle Scripts 폴더

- build.gradle (Module: app) : 빌드 스크립트 핵심 파일
- local.properties : 컴파일되는 SDK의 경로가 들어 있음
- gradle.properties: JVM 관련 메모리가 설정되어 있음

79 / 80



감사합니다.