

02. 처음 만드는 안드로이드 애플리케이션

Android Studio를 활용한 안드로이드 프로그래밍

© 이 강의노트는 한빛아카데미 출판사에서 제공한 강의노트를 기초로 만들어졌습니다.

학습목표

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

차례

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

Page • 3

1. [Hello Android] 프로그램



그림 2-1 처음 만든 안드로이드 애플리케이션

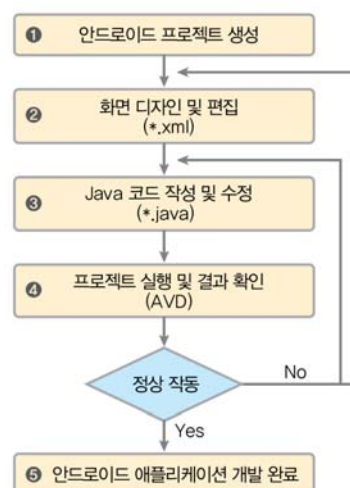


그림 2-2 안드로이드 프로젝트 개발 단계

1. [Hello Android] 프로그램 ▶ 프로젝트 생성[1/6]

실습 2-1 첫 번째 응용 프로그램 작성하기

❖ Android Studio를 실행한 후 [Start a new Android Studio project]를 클릭



Page • 5 그림 2-3 새로운 안드로이드 프로젝트

1. [Hello Android] 프로그램 ▶ 프로젝트 생성[2/6]

실습 2-1 첫 번째 응용 프로그램 작성하기

❖ [Create Android Project]에서 프로젝트 정보 입력



- Application Name : 프로젝트 이름은 자동으로 애플리케이션 이름과 동일하게 설정
- Company Domain : 회사의 도메인 (필자가 가정한 가상 주소)
- Project location : 적당한 폴더 지정(한글 안 됨)
- Package Name : 자동 생성되며 도메인 이름과 애플리케이션 이름이 반대로 이어짐

Page • 6 그림 2-4 프로젝트 정보 입력 1

1. [Hello Android] 프로그램 ▶ 프로젝트 생성[3/6]

실습 2-1 첫 번째 응용 프로그램 작성하기

- ❖ [Select the form factors and minimum SDK]에서 앱을 실행한 환경을 선택
 - ✓ 'Phone and Tablet' 체크



Page • 7

그림 2-5 프로젝트 정보 입력 2

1. [Hello Android] 프로그램 ▶ 프로젝트 생성[4/6]

실습 2-1 첫 번째 응용 프로그램 작성하기

- ❖ [Add an activity to Mobile] 창에서 디폴트인 'Empty Activity' 선택



Page • 8 그림 2-6 액티비티 선택 화면

1. [Hello Android] 프로그램 ▶ 프로젝트 생성[5/6]

실습 2-1 첫 번째 응용 프로그램 작성하기

❖ [Configure Activity] 창도 디폴트로 두고 <Finish>를 클릭

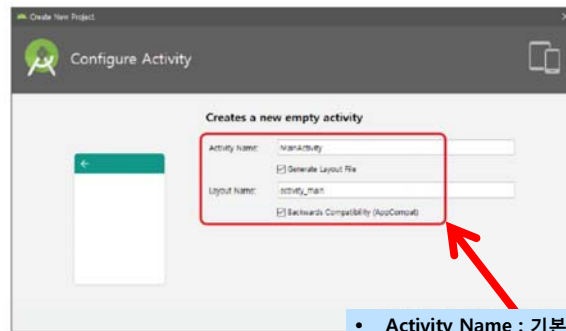


그림 2-7 관련 이름 지정

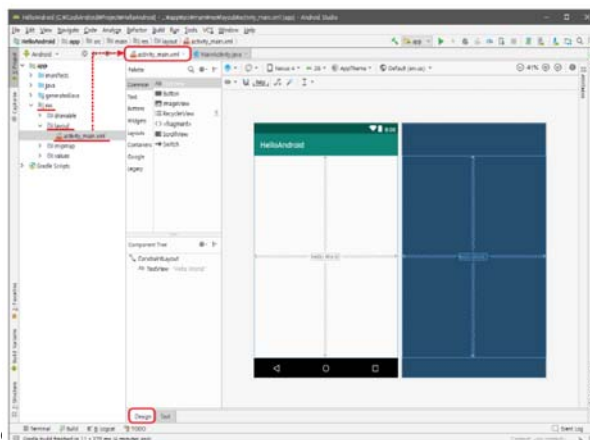
- Activity Name : 기본 소스인 Java 파일 이름으로 지정
- Layout Name : 기본 화면인 XML 파일 이름으로 지정

Page • 9

1. [Hello Android] 프로그램 ▶ 프로젝트 생성[6/6]

실습 2-1 첫 번째 응용 프로그램 작성하기

❖ Android Studio의 왼쪽 Project Tree에 'HelloAndroid' 프로젝트가 추가됨



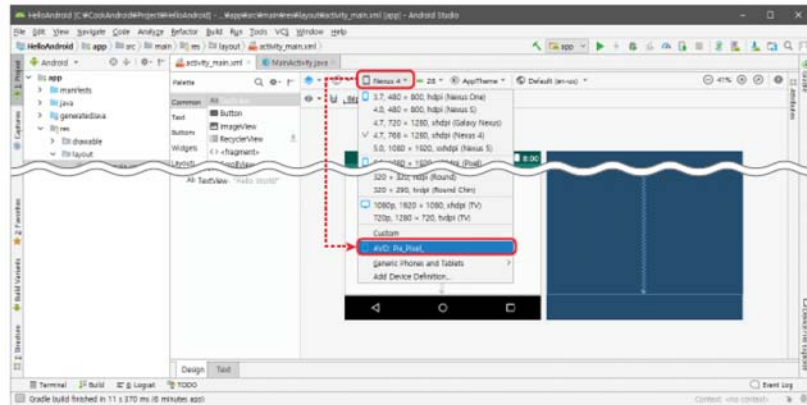
Page • 11

그림 2-8 안드로이드 프로젝트 생성 결과

1. [Hello Android] 프로그램 ▶ 화면 디자인 및 편집[1/5]

실습 2-1 첫 번째 응용 프로그램 작성하기

❖ 화면을 가상 AVD와 동일하게 설정



Page • 11 그림 2-9 디자인 환경 설정

1. [Hello Android] 프로그램 ▶ 화면 디자인 및 편집[2/5]

실습 2-1 첫 번째 응용 프로그램 작성하기

❖ 아래쪽의 'Text'를 클릭해서 android.support.constraint.ConstraintLayout을 RelativeLayout으로 변경

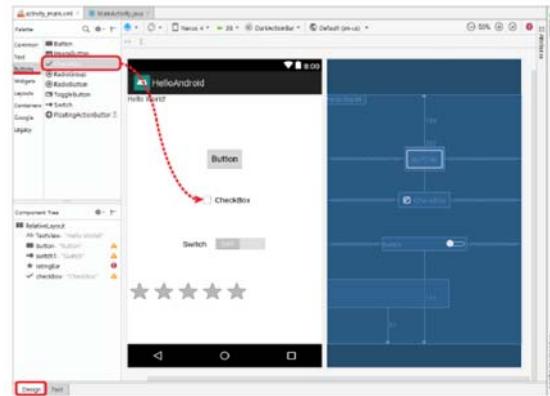


Page • 12 그림 2-10 레이아웃 변경

1. [Hello Android] 프로그램 ▶ 화면 디자인 및 편집[3/5]

실습 2-1 첫 번째 응용 프로그램 작성하기

- ❖ 왼쪽 [Widgets]에서 몇 개를 오른쪽에 옮김. 오른쪽 위의 확대/축소 아이콘으로 화면 크기 조절 가능



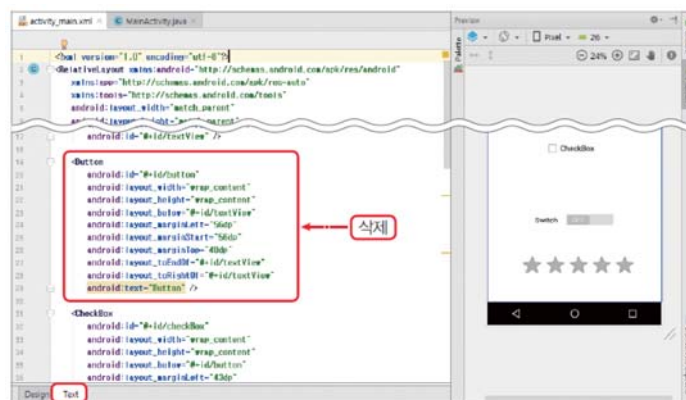
Page • 13

그림 2-11 그래픽 화면에서 위젯 끌어다 놓기

1. [Hello Android] 프로그램 ▶ 화면 디자인 및 편집[4/5]

실습 2-1 첫 번째 응용 프로그램 작성하기

- ❖ 왼쪽 아래 [Text] 탭을 클릭하면 화면에 xml 코드가 표시됨



Page • 14 그림 2-12 XML 코드 변경

1. [Hello Android] 프로그램 ▶ 화면 디자인 및 편집[5/5]

실습 2-1 첫 번째 응용 프로그램 작성하기

- ❖ [Design]을 클릭하면 삭제한 코드가 화면에도 삭제되어 있음
- ❖ 왼쪽 상단의 저장 아이콘을 클릭하거나 메뉴 [File]-[Save All]을 클릭하여 저장

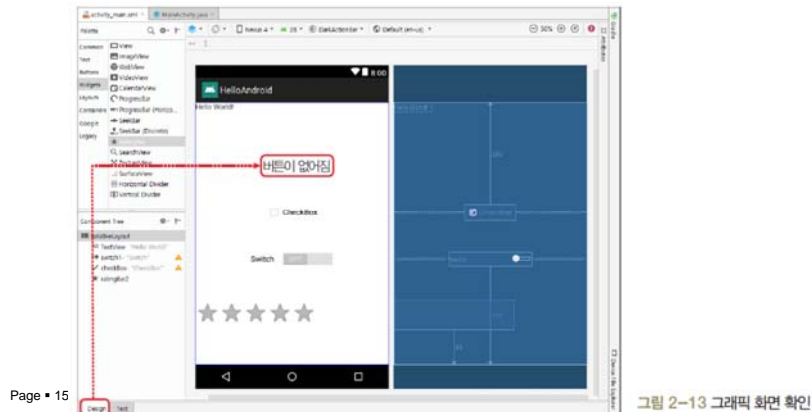


그림 2-13 그래픽 화면 확인

1. [Hello Android] 프로그램 ▶ JAVA 코드 작성 및 수정

실습 2-1 첫 번째 응용 프로그램 작성하기

- ❖ Project Tree의 [java]-[com.cookandroid.helloandroid]-[MainActivity] 선택

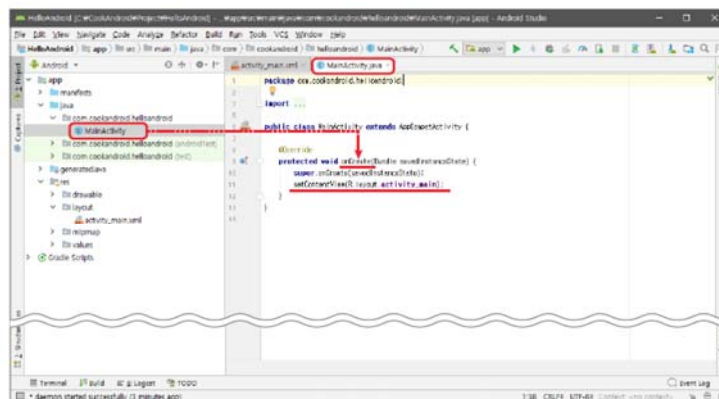


그림 2-14 메인 Java 코드 확인

1. [Hello Android] 프로그램 ▶ 프로젝트 실행, 확인[1/4]

실습 2-1 첫 번째 응용 프로그램 작성하기

❖ 메뉴의 [Run As]-[Run 'app']을 선택하거나 [Run 'app'] 아이콘을 클릭

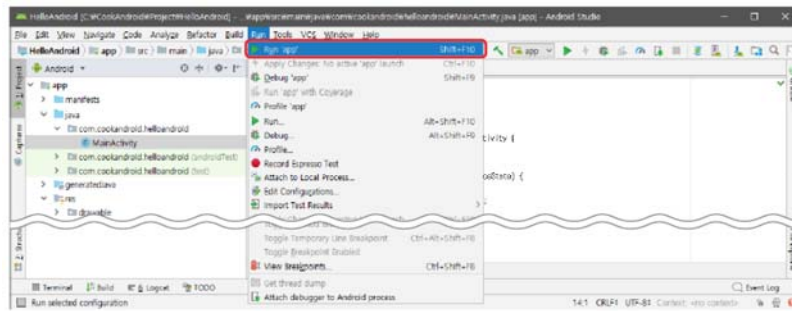


그림 2-15 애플리케이션 실행

1. [Hello Android] 프로그램 ▶ 프로젝트 실행, 확인[2/4]

실습 2-1 첫 번째 응용 프로그램 작성하기

❖ [Choose Device] 장치에서 실행할 안드로이드 기기나 AVD를 선택

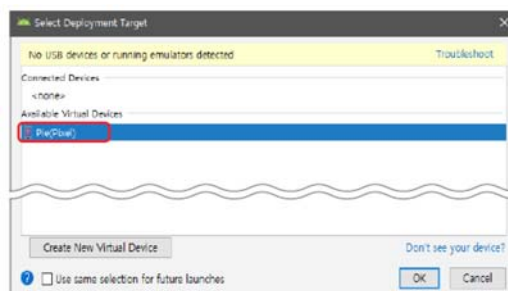


그림 2-16 실행할 안드로이드 장치 선택

1. [Hello Android] 프로그램 ▶ 프로젝트 실행, 확인[3/4]

실습 2-1 첫 번째 응용 프로그램 작성하기

❖ AVD가 부팅된 후 실행 결과 화면이 나타남

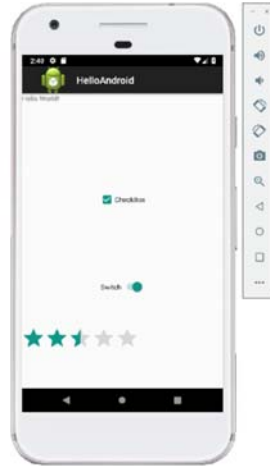


그림 2-17 프로젝트 실행 결과

Page • 19

1. [Hello Android] 프로그램 ▶ 프로젝트 실행, 확인[4/4]

실습 2-1 첫 번째 응용 프로그램 작성하기

❖ AVD의 초기화면으로 돌아가려면 오른쪽 키패드의 돌아가기 버튼이나 (◀)를 누름



그림 2-18 돌아가기 버튼

Page • 20

1. [Hello Android] 프로그램

▶ 직접 풀어보기 2-1

새로운 프로젝트 'HiAndroid'를 작성하고 실행한 뒤 화면을 꾸며보자.

Page • 21

1. [Hello Android] 프로그램

자주 환미디 - AVD에서 앱 아이콘을 출력하는 방법

① 먼저 아이콘으로 사용할 파일이 필요하다. 일반적으로 96×96 크기 정도의 *.png 파일을 사용하는 것이 좋다. 필자는 C:\WookAndroid\sdks\extras\google\google_play_services\samples\wallet\res\drawable-xhdpi 폴더의 ic_launcher.png 파일을 사용했다(적당한 다른 이미지를 사용해도 상관없다).

② 파일을 복사해서 Project Tree의 [app]-[res]-[drawable]에 붙여넣기 한다. 메시지 창이 나오면 모두 <OK>를 클릭한다.

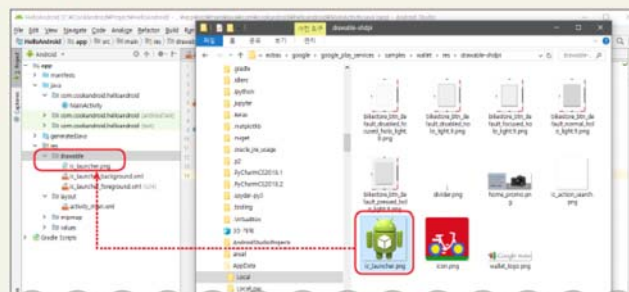


그림 2-19 아이콘 이미지를 drawable에 복사

1. [Hello Android] 프로그램

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

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

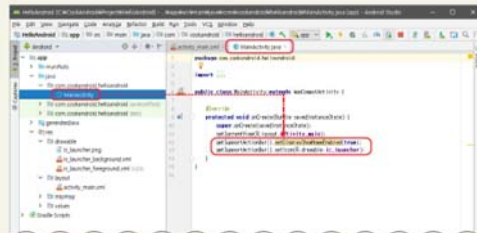


그림 2-20 예인 Java 코드 수정

● Project Tree에서 [app]-[manifests]-[AndroidManifest.xml]을 더블클릭해서 열고, 11행쯤의
android:theme를 다음과 같이 수정한다. (이 부분은 생략해도 된다.)

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

프로젝트를 실행하면 이제 AVD에도 완벽 상단에 앱 아이콘이 출력될 것이다.

2. AVD 명칭과 사용법 ▶ AVD 명칭



그림 2-21 AVD의 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

2. AVD 명칭과 사용법 ▶ AVD 사용법[1/7]

실습 2-2 AVD 사용법 알아보기

❖ 화면 전환



[Ctrl]+[←]
[Ctrl]+[→]



그림 2-23 가로로 전환한 화면(크롬 앱을 실행하고 자동 회전을 켜)

Page • 25 그림 2-22 초기화면

2. AVD 명칭과 사용법 ▶ AVD 사용법[2/7]

실습 2-2 AVD 사용법 알아보기

❖ 알람 추가

- ✓ 초기화면을 위쪽으로 스와이프하여 [애플리케이션 화면에서 [시계 버튼] 선택
- ✓ 아래쪽의 <+>를 클릭해서 알람 시간 추가 가능



Page • 26

그림 2-24 알람 시간 추가

2. AVD 명칭과 사용법 ▶ AVD 사용법[3/7]

실습 2-2 AVD 사용법 알아보기

❖ 카메라

- ✓ 론처 버튼을 누른 후 카메라 버튼 클릭
- ✓ 셔터 버튼을 클릭하면 가상 화면이



Page • 21

그림 2-25 가상 카메라 촬영

❖ 포토

- ✓ 론처 버튼을 누른 후 [포토]를 클릭하면 백업 내용이 나옴. 사진의 편집/변경 및 삭제, 공유 등의 작업이



그림 2-26 촬영된 사진 확인

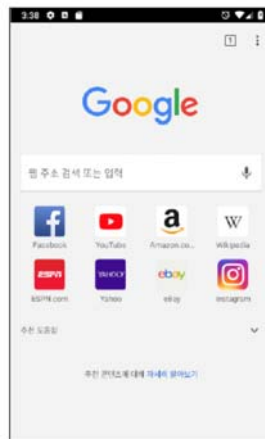


2. AVD 명칭과 사용법 ▶ AVD 사용법[4/7]

실습 2-2 AVD 사용법 알아보기

❖ 웹브라우저

- ✓ 크롬 버튼 클릭



Page • 28

그림 2-27 웹 서핑

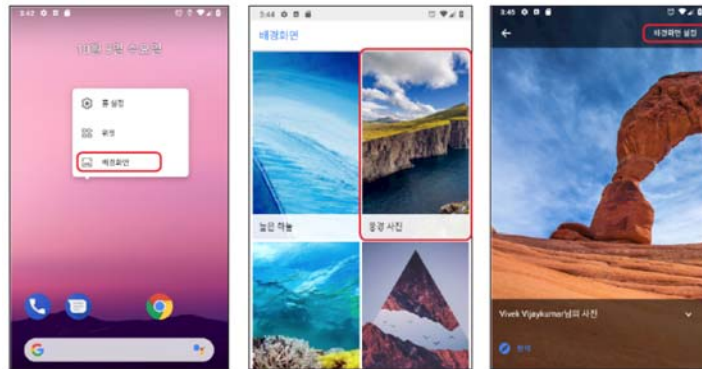


2. AVD 명칭과 사용법 ▶ AVD 사용법[5/7]

실습 2-2 AVD 사용법 알아보기

❖ 배경화면 변경

- ✓ 바탕화면을 몇 초 동안 누른 후 [배경화면] 선택. <배경화면 설정>을 클릭한 후 <둘다> 클릭



Page • 29

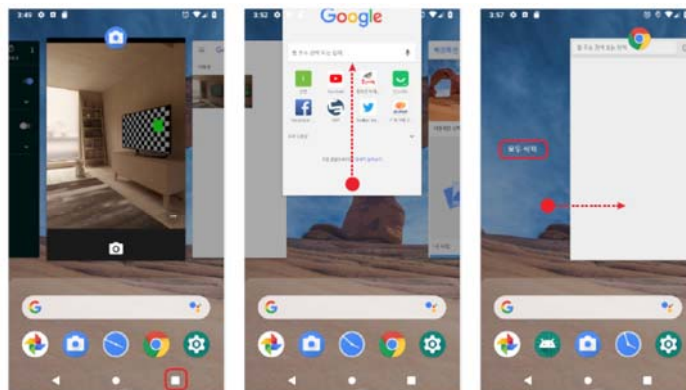
그림 2-28 배경화면 설정

2. AVD 명칭과 사용법 ▶ AVD 사용법[6/7]

실습 2-2 AVD 사용법 알아보기

❖ 오버뷰 버튼

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



Page • 30

그림 2-29 실행 중인 프로그램 종료

2. AVD 명칭과 사용법 ▶ AVD 사용법[7/7]

실습 2-2 AVD 사용법 알아보기

❖ 바탕화면에 자주 사용하는 애플리케이션 복사

- ✓ HelloAndroid 아이콘을 몇 초간 누르면 <앱정보>가 나옴. 몇초간 누르면 바탕화면으로 복사
- ✓ 아이콘을 삭제하려면 아이콘을 몇 초간 누르고 상단의 <삭제>로 가져다 놓으면 됨



Page • 31

그림 2-30 바탕화면에 애플리케이션 복사

그림 2-31 바탕화면의 아이콘 삭제

3. 안드로이드 애플리케이션 작성 ▶ 프로젝트 관리[1/6]

실습 2-3 안드로이드 프로젝트 관리하기

❖ 프로젝트 닫기와 열기

- ✓ [File]-[Close Project]를 선택해서 프로젝트 닫음
- ✓ 프로젝트 이름을 클릭하면 다시 프로젝트가 열림, 폴더 검색해서 프로젝트를 열어도 됨

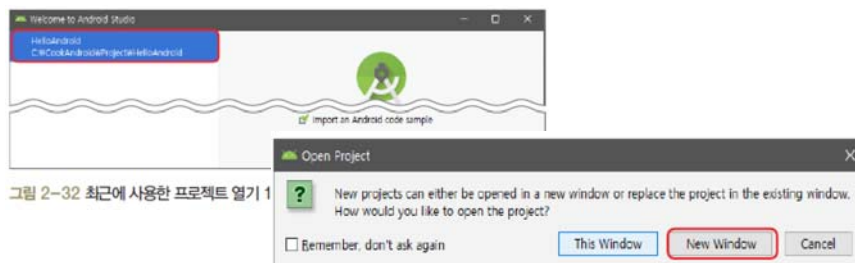


그림 2-32 최근에 사용한 프로젝트 열기 1

그림 2-33 최근에 사용한 프로젝트 열기 2

- ✓ 여러 개의 프로젝트를 열려면 [File]-[open Recent]-[프로젝트 이름] 선택 <New Window>를 클릭하면 새로운 창이 열려서 여러 개의 프로젝트 동시 작업 가능

Page • 32

3. 안드로이드 애플리케이션 작성 ▶ 프로젝트 관리[2/6]

실습 2-3 안드로이드 프로젝트 관리하기

- ✓ 폴더를 검색해서 프로젝트를 열어도 됨
- ✓ [Open an existing Android Studio project]를 클릭하고 프로젝트가 저장된 폴더를 지정

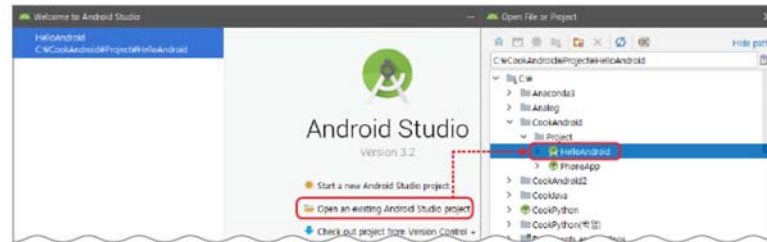


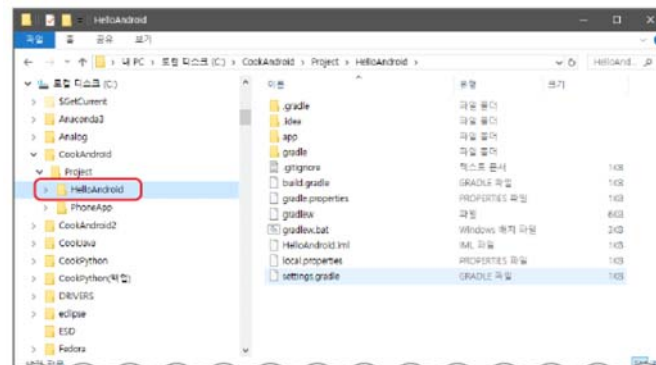
그림 2-34 폴더에서 직접 프로젝트 열기

Page • 33

3. 안드로이드 애플리케이션 작성 ▶ 프로젝트 관리[3/6]

실습 2-3 안드로이드 프로젝트 관리하기

- ❖ Android Studio 프로젝트 내보내기/가져오기
 - ✓ 프로젝트가 생성된 폴더를 통째로 복사하거나 압축해서 보내거나 가져오면 됨



Page • 34

그림 2-35 프로젝트 폴더를 통째로 복사

3. 안드로이드 애플리케이션 작성 ▶ 프로젝트 관리[4/6]

실습 2-3 안드로이드 프로젝트 관리하기

❖ 이클립스용 프로젝트 가져오기

- ✓ [Import project (Gradle, Eclipse ADT, etc.)] 선택 후 작성된 프로젝트 폴더 선택

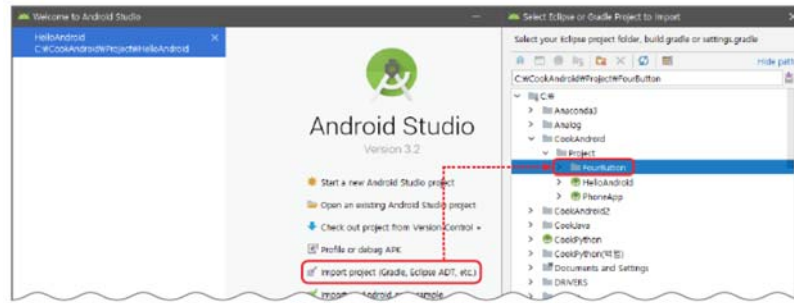


그림 2-36 Eclipse용 프로젝트 가져오기 1

Page • 35

3. 안드로이드 애플리케이션 작성 ▶ 프로젝트 관리[5/6]

실습 2-3 안드로이드 프로젝트 관리하기

❖ 이클립스용 프로젝트 가져오기

- ✓ 가져올 폴더 지정 후 새로운 폴더가 생성되면 <Next> 클릭
- ✓ 프로젝트 변환하는 옵션이다. 모두 체크된 상태에서 <Finish> 클릭

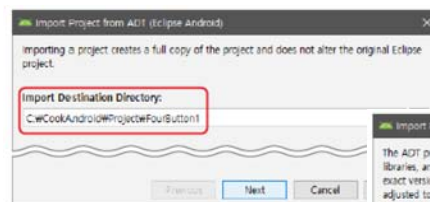


그림 2-37 Eclipse용 프로젝트 가져오기 2

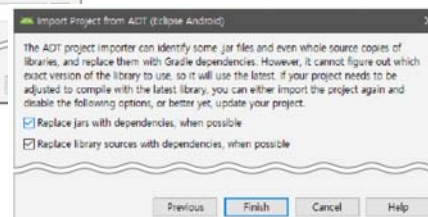


그림 2-38 Eclipse용 프로젝트 가져오기 3

Page • 36

3. 안드로이드 애플리케이션 작성 ▶ 프로젝트 관리[6/6]

실습 2-3 안드로이드 프로젝트 관리하기

❖ 이클립스용 프로젝트 가져오기

- ✓ 변환 완료된 이클립스 프로젝트는 Android Studio 프로젝트와 같은 방법으로 사용

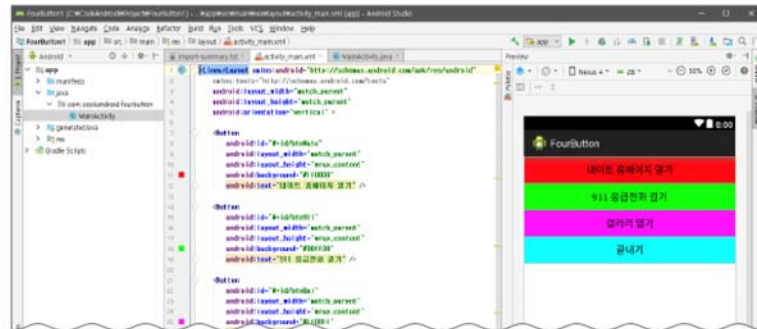


그림 2-39 변환 완료된 Eclipse용 프로젝트

Page • 37

3. 안드로이드 애플리케이션 작성

▶ 직접 풀어보기 2-2

이클립스에서 작성한 프로젝트를 Android Studio로 가져와보자.(이클립스용 프로젝트가 없다면 생략해도 좋다.)

❖ 안드로이드 프로젝트를 통해 다음을 파악할 수

- ✓ 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

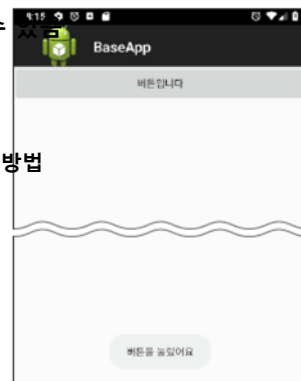


그림 2-40 안드로이드 프로젝트 표준 틀 결과

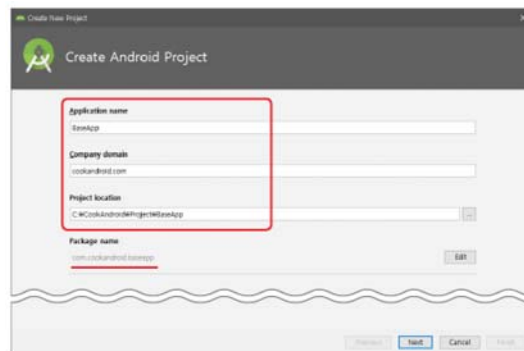
Page • 38

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[1/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ 안드로이드 프로젝트 생성

- ✓ 메뉴 [File]-[New Project]를 선택
- ✓ 프로젝트 이름은 'BaseApp'이라고 지정



Page • 39

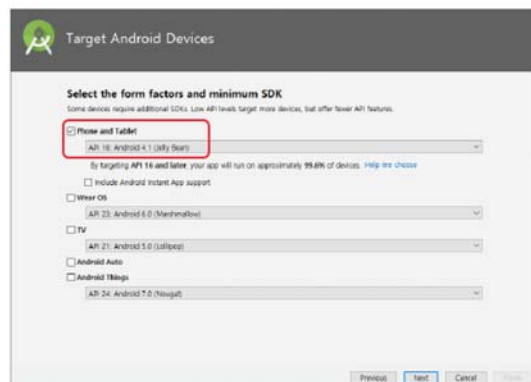
그림 2-41 애플리케이션 정보 입력

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[2/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ 안드로이드 프로젝트 생성

- ✓ [Target Android Devices]에서 앱을 실행한 환경을 선택



Page • 40

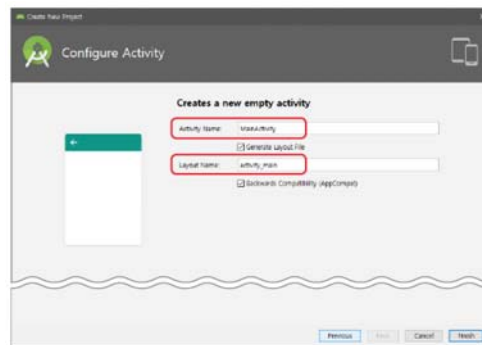
그림 2-42 최소 버전 선택

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[3/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ 안드로이드 프로젝트 생성

- ✓ [Add an Activity to Mobile]에서 디폴트인 'Empty Activity' 선택
- ✓ [Configure Activity] 창도 디폴트로 두고 <Finish>를 클릭



Page • 41

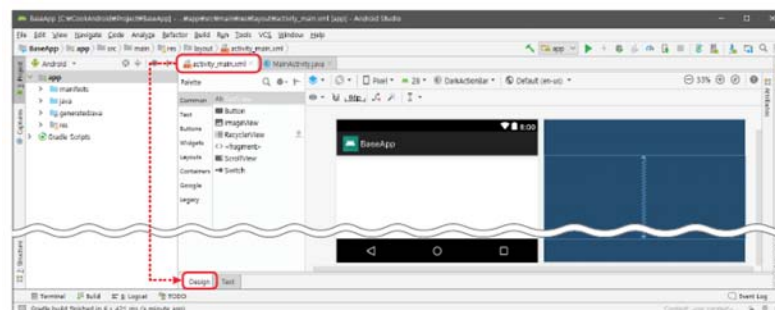
그림 2-43 액티비티 및 레이아웃 이름 입력

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[4/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ 화면 디자인 및 편집

- ✓ Project Tree에서 [java]-[com.cookandroid.baseapp]-[MainActivity]가 기본적으로 열려있을 것. 상단의 [activity_main.xml]을 클릭하고 아래쪽의 [Design] 탭을 클릭



Page • 42

그림 2-44 activity_main.xml 확인

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[5/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ 화면 디자인 및 편집

- ✓ 화면 아래쪽의 [Text] 탭을 클릭해서 XML 코드를 확인



그림 2-45 XML 코드 변경

Page • 43

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[6/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ 화면 디자인 및 편집

- ✓ <LinearLayout>과 </LinearLayout> 사이에 버튼을 추가

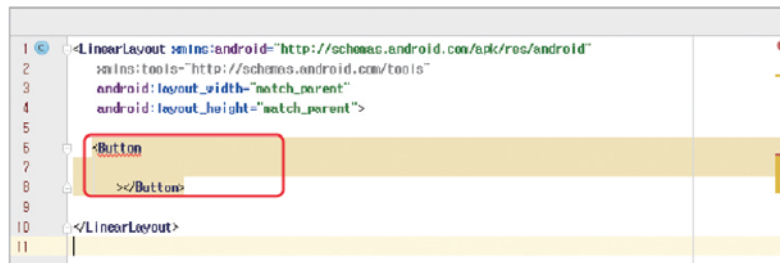


그림 2-46 Button 추가

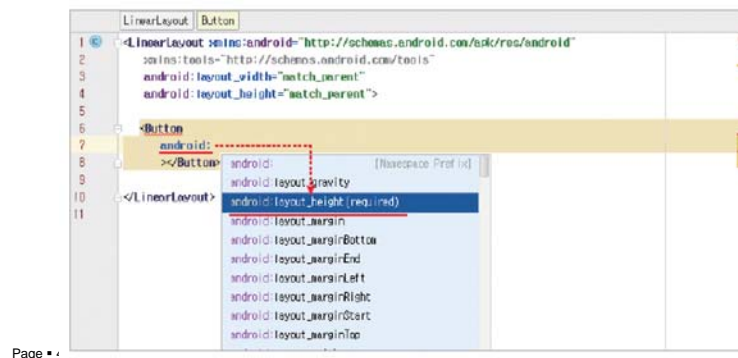
Page • 44

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[7/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ 화면 디자인 및 편집

- ✓ “android:”을 입력하면 자동으로 여러 개를 선택할 수 있는 목록이 나옴
- ✓ 그 중에서 layout_height를 더블클릭해서 선택하면 자동 완성됨



Page •

그림 2-47 버튼 속성 편집

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[8/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ 화면 디자인 및 편집

- ✓ 큰따옴표(“ ”) 안에 커서를 가져다 놓고 [Ctrl]+[Space Bar]를 누르면 내용을 쓸 수 있음



그림 2-48 버튼 속성 중 layout_height 편집

Page • 46

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[9/25]

❖ 1차 코드 완성

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>
```

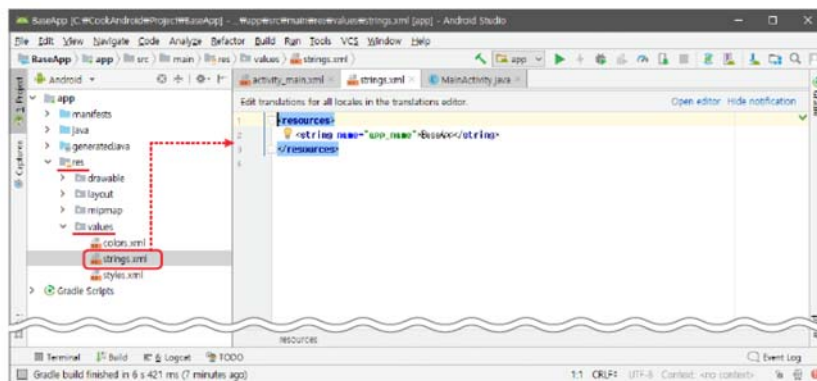
Page • 4

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[10/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ 오류 수정

- ✓ Project Tree의 [res]-[values]-[strings.xml]을 더블클릭해서 코드 확인



Page •

그림 2-49 strings.xml 파일

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[11/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ 오류 수정

- ✓ strBtn1 문자열을 추가한 후 저장. 빨간색 오류가 사라짐.

예제 2-2 strings.xml

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

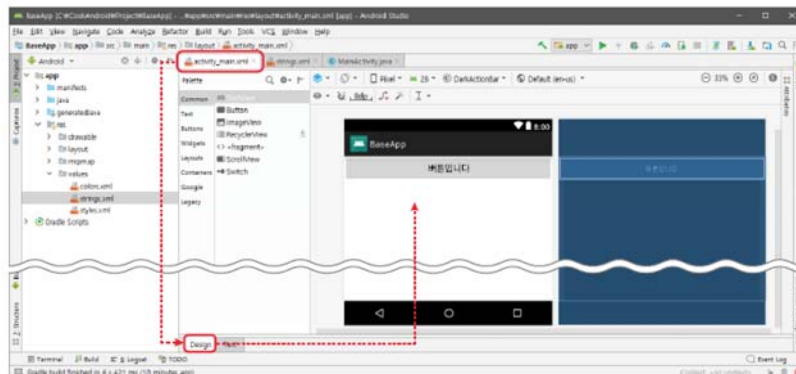
Page • 49

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[12/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ 확인

- ✓ activity_main.xml에서 하단의 [Design] 탭을 클릭해서 그래픽 화면으로 확인



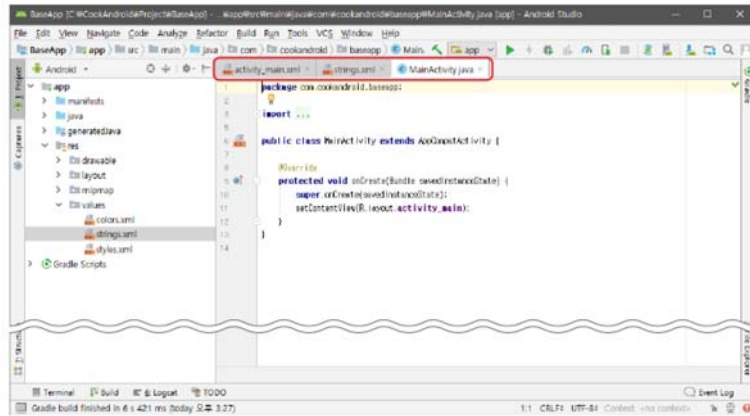
Page • 5

그림 2-50 activity_main.xml 코드의 그래픽 화면

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[12/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ 확인



Page • 51

그림 2-51 열린 파일을 탭으로 전환

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[13/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ 실행

- ✓ Android Studio 메뉴의 [Run As]-[Run 'app']을 선택하거나 [Run 'app'] 아이콘 클릭

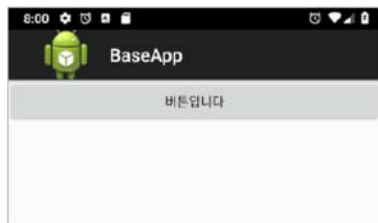


그림 2-52 프로젝트 실행

Page • 52

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[14/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ JAVA 코드 작성 및 수정

- ✓ Project Tree의 [java]-[패키지 이름]-[MainActivity]를 더블클릭하면 Java 코드가 열림

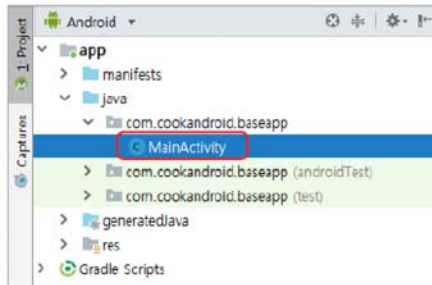


그림 2-53 Java 파일 선택

Page • 53

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[15/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ JAVA 코드 작성 및 수정

- ✓ 기본적인 코드의 구성



그림 2-54 MainActivity.java 기본 코드

Page • ...

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[16/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ JAVA 코드 작성 및 수정

- ✓ import 앞의 작은 '+' 아이콘을 클릭하면 행이 확장됨
- ✓ [Ctrl]+[Alt]+[O]를 누르면 불필요하게 임포트된 문장 제거

```
1 package com.cookandroid.basapp;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
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
```

Page • 55

그림 2-55 필요 없는 import문 삭제

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[17/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ JAVA 코드 작성 및 수정

- ✓ Button 변수 추가
- ✓ 멤버변수 위치인 @Override 바로 위에 다음을 입력

Button button1;

```
1 package com.cookandroid.basapp;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5
6 public class MainActivity extends AppCompatActivity {
7     Button button1;
8     @Override
9     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13 }
14
```

Page • 56

그림 2-56 Button 변수 추가

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[18/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ JAVA 코드 작성 및 수정

- ✓ [Alt]+[Enter] 누르면 Button과 관련된 클래스가 자동으로 import문에 추가됨



```
1 package com.cookandroid.baseapp;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5 import android.widget.Button;
6
7 public class MainActivity extends AppCompatActivity {
8     Button button1;
9     @Override
10    protected void onCreate(Bundle savedInstanceState) {
11        super.onCreate(savedInstanceState);
12        setContentView(R.layout.activity_main);
13    }
14 }
15
```

Page • 57

그림 2-57 자동 import 확인

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[19/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ JAVA 코드 작성 및 수정

- ✓ findViewById() 메소드를 사용하여 activity_main.xml 파일에서 만든 객체에 접근
- ✓ setContentView() 메소드 바로 아래에 다음 코드를 추가

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

Page • 58

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[20/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ JAVA 코드 작성 및 수정

- ✓ “button1.setOncl”까지 입력 후 setOnClickListener()를 선택



Page • 59

그림 2-58 Java 코드 자동 완성 1

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[21/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ JAVA 코드 작성 및 수정

- ✓ setOnClickListener()에 “new View”를 입력 후 OnClickListener()선택

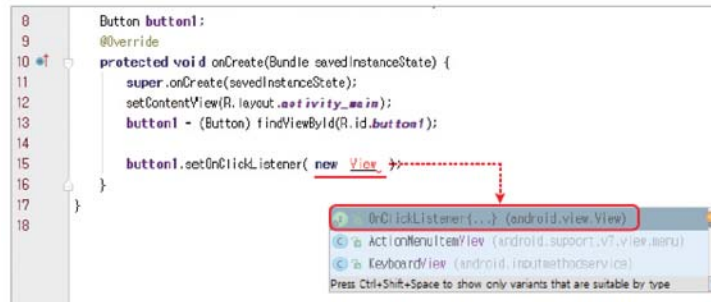


그림 2-59 Java 코드 자동 완성 2

Page • 60

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[22/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ JAVA 코드 작성 및 수정

- ✓ `onClick()` 메소드가 자동으로 추가

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

그림 2-60 자동 완성된 코드

Page • 61

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[23/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ JAVA 코드 작성 및 수정

- ✓ 버튼을 클릭했을 때 작동하기 원하는 모든 코드를 `onClick()` 메소드 안에 입력

```
예제 2-3 MainActivity.java
1 package com.cookandroid.baseapp;
2
3 import android.support.v7.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
19         button1.setOnClickListener(new View.OnClickListener() {
20             @Override
21             public void onClick(View v) {
22
23                 Toast.makeText(getApplicationContext(), "버튼을 눌렀어요",
24                     Toast.LENGTH_SHORT).show();
25             }
26         });
27     }
28 }
```

Page • 62

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[24/25]

실습 2-4 기본적인 애플리케이션 작성하기

❖ 프로젝트 실행 및 결과 확인

- ✓ [Run As]-[Run 'app']을 선택하거나 [Run 'app']을 클릭해서 프로젝트를 실행



그림 2-61 실행 결과

Page • 63

3. 안드로이드 애플리케이션 작성 ▶ 표준 틀[25/25]

제외 한다

R 클래스

레이아웃 파일에 activity_main.xml, strings.xml에 들어 있는 문자열, 버튼 등의 id, 이미지 파일의 id 등은 모두 R 클래스에 자동으로 등록된다. Package Tree 위쪽을 [Android]에서 [Packages]로 변경한 후 [app]-[?]지 이름]-[R]을 더블클릭하면 R 클래스가 나타난다. [예제 2-3]의 17행에서 버튼에 접근하는 데 R.id.button1을 사용했다. (참조는 @+id를 누르고 'button1'을 검색한 상태이다.)

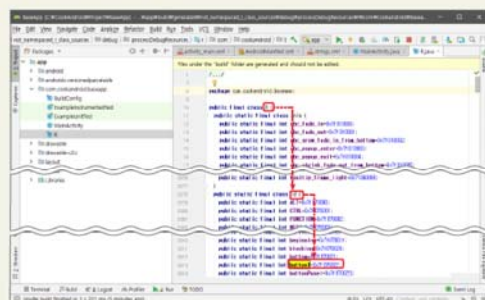
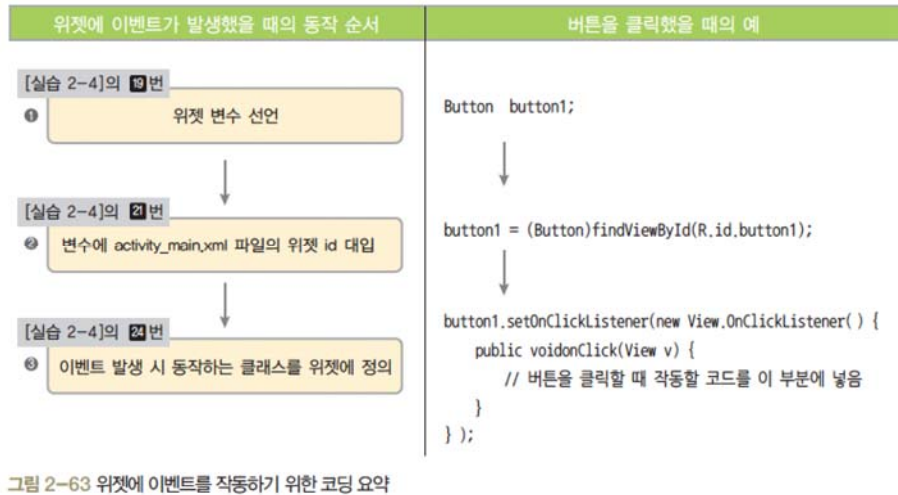


그림 2-62 R.java 파일

그 외에 strings.xml 파일에 추가했던 문자열도 R.string.string1로 접근이 가능하다. 나중에는 그림 파일도 사용할 텐데, 그림 파일은 R.drawable.그림 파일 id 형식으로 사용할 수 있다. R.java에 등록된 것은 모두 자동으로 등록되며, 각각 고유한 16진수 상수를 정의하는데 절대 직접 수정하면 안 된다. 그러므로 실제로 R.java 파일을 열어볼 필요도 없을 것이다. [예제 2-1] activity_main.xml의 10행에 나온 android:id="@+id/button1"과 같은 형식이 등록되면 이것을 Java 코드에서 R.id.button1로 접근할 수 있다고 생각하면 된다.

3. 안드로이드 애플리케이션 작성



Page • 65

3. 안드로이드 애플리케이션 작성

▶ 작업 풀이보기 2-3

다음 그림과 같이 버튼 4개를 만들고 각 버튼을 클릭하면 필요한 내용이 작동되는 FourButton 프로젝트를 작성하라. 각 버튼은 다른 색상으로 변경한다.



그림 2-64 실행 결과

4. 프로젝트의 구성 ▶ BaseApp 프로젝트 구성[1/3]

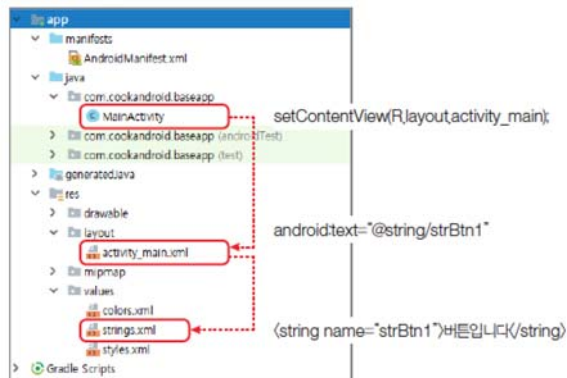


그림 2-65 안드로이드 프로젝트의 구성

Page • 67

4. 프로젝트의 구성 ▶ BaseApp 프로젝트 구성[2/3]

❖ java 폴더

- ✓ 하위에 패키지명의 하위 폴더가 있는데, 이는 안드로이드 프로젝트를 생성할 때 입력한 패키지 이름과 동일
- ✓ 패키지 이름 아래에 `MainActivity.java`로 메인 Java 소스가 들어 있음

❖ res 폴더

- ✓ 앱 개발에 사용되는 이미지, 레이아웃, 문자열 등이 들어가는 폴더
- ✓ 이미지 파일은 `drawable` 폴더에 넣음
- ✓ `layout` 폴더는 액티비티(화면)을 구성하는 xml 파일을 넣으면 됨
- ✓ `values` 폴더는 문자열을 저장하는 `string.xml` 등이 들어 있음
- ✓ `menu` 폴더는 메뉴 XML 파일이 저장되어 있음

Page • 68

4. 프로젝트의 구성 ▶ BaseApp 프로젝트 구성[3/3]

❖ generatedJava 폴더

- ✓ Android Studio 3.2부터 제공되는 폴더로 시스템 내부적으로 사용됨

❖ manifests 폴더

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

❖ Gradle Scripts 폴더

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

Page • 69

Android Studio를 활용한 안드로이드 프로그래밍

