
Kotlin을 이용한 Android 프로그래밍

퀴즈 잠금화면 앱 만들기

Contents

- I. 사용자가 스마트폰의 잠금 화면을 해제하기 위하여 퀴즈의 정답을 맞춤
- II. 잠금화면 앱 작성 방법
 - I. Activity, Service, BroadcastReceiver 및 UI 애니메이션 등의 효과를 사용



퀴즈 잠금화면 앱 만들기

▶ 퀴즈 잠금화면 앱

▶ 안드로이드는 잠금화면을 직접 개발할 수 있음

- ▷ 이외에 키보드, 런처 등도 모두 앱으로 제작 가능

- ▷ 아이폰의 경우 키보드, 잠금화면, 런처 등은 시스템에서 제공하는 것만 사용

▶ 잠금화면을 이용하는 앱은 종류가 다양함

- ▷ 잠금화면에서 특정 캐릭터나 아름다운 배경을 보여주는 앱이나 광고를 보고 일정 금액을 정립해주는 앱 등

- ▷ 사용자가 스마트폰을 사용하기 전에 한 번은 꼭 확인하는 화면

▶ 본 프로젝트는 잠금화면에서 퀴즈를 보여주고 정답을 맞춰서 잠금을 해제하는 앱

▶ 화면은 2가지로 구성

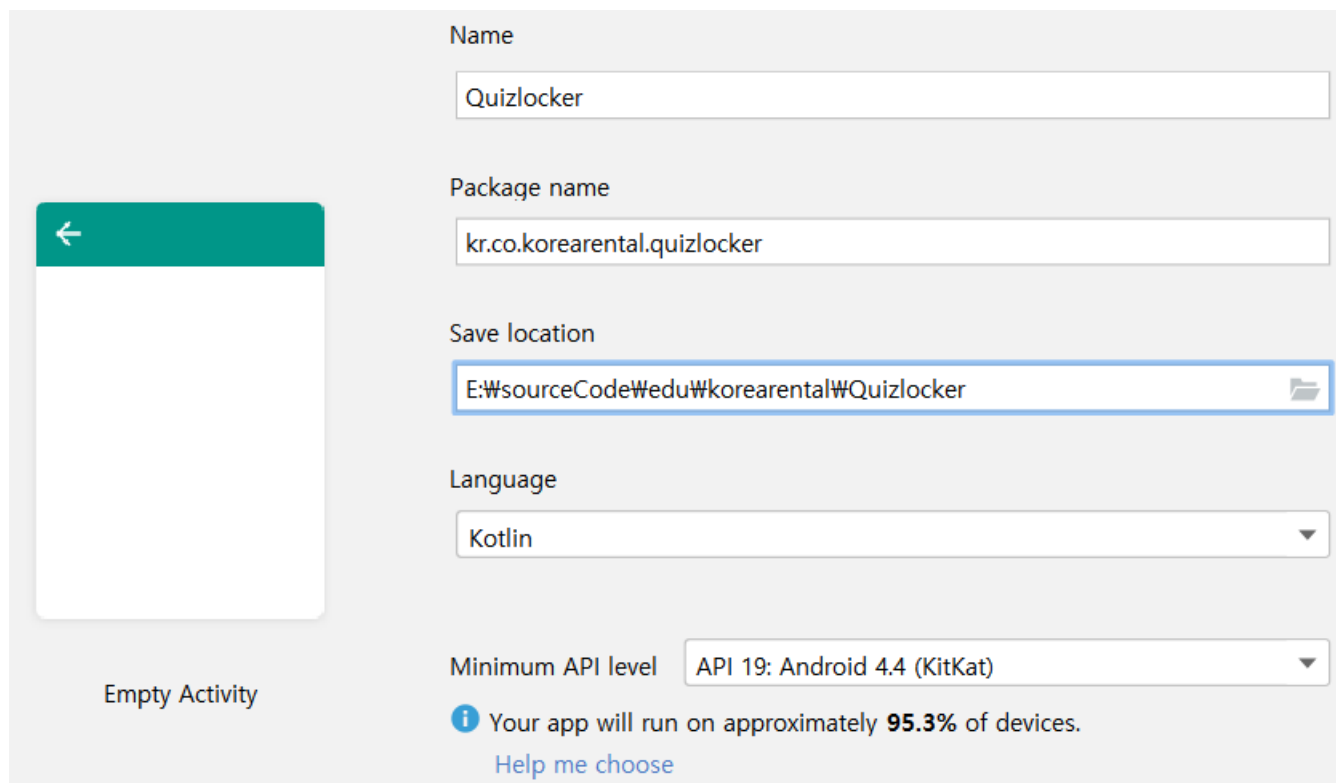
- ▷ 기존의 잠금화면을 대체하는 화면

- ▷ 잠금화면의 설정을 변경하는 화면

퀴즈 잠금화면 앱 만들기

▶ 프로젝트 생성

- ▶ 프로젝트 명 : Quizlocker
- ▶ minSdkVersion : 19(Android 4.4 KitKat)
- ▶ 기본 액티비티 : EmptyActivity



Name

Quizlocker

Package name

kr.co.korearental.quizlocker

Save location

E:\sourceCode\Wedu\Wkorearental\WQuizlocker

Language

Kotlin

Minimum API level

API 19: Android 4.4 (KitKat)

Empty Activity

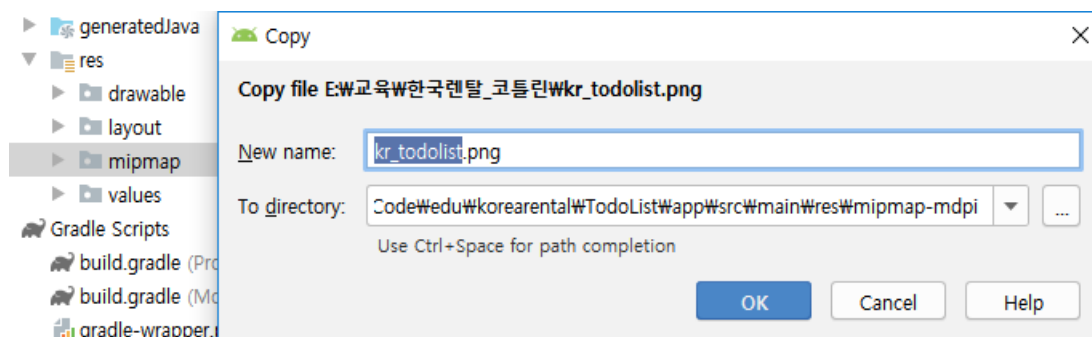
i Your app will run on approximately **95.3%** of devices.
[Help me choose](#)

퀴즈 잠금화면 앱 만들기

▶ 프로젝트 설정

▶ kr_quizlocker.png로 아이콘 변경

▶ res/mipmap 에 아이콘 이미지 복사



▶ 매니페스트 수정

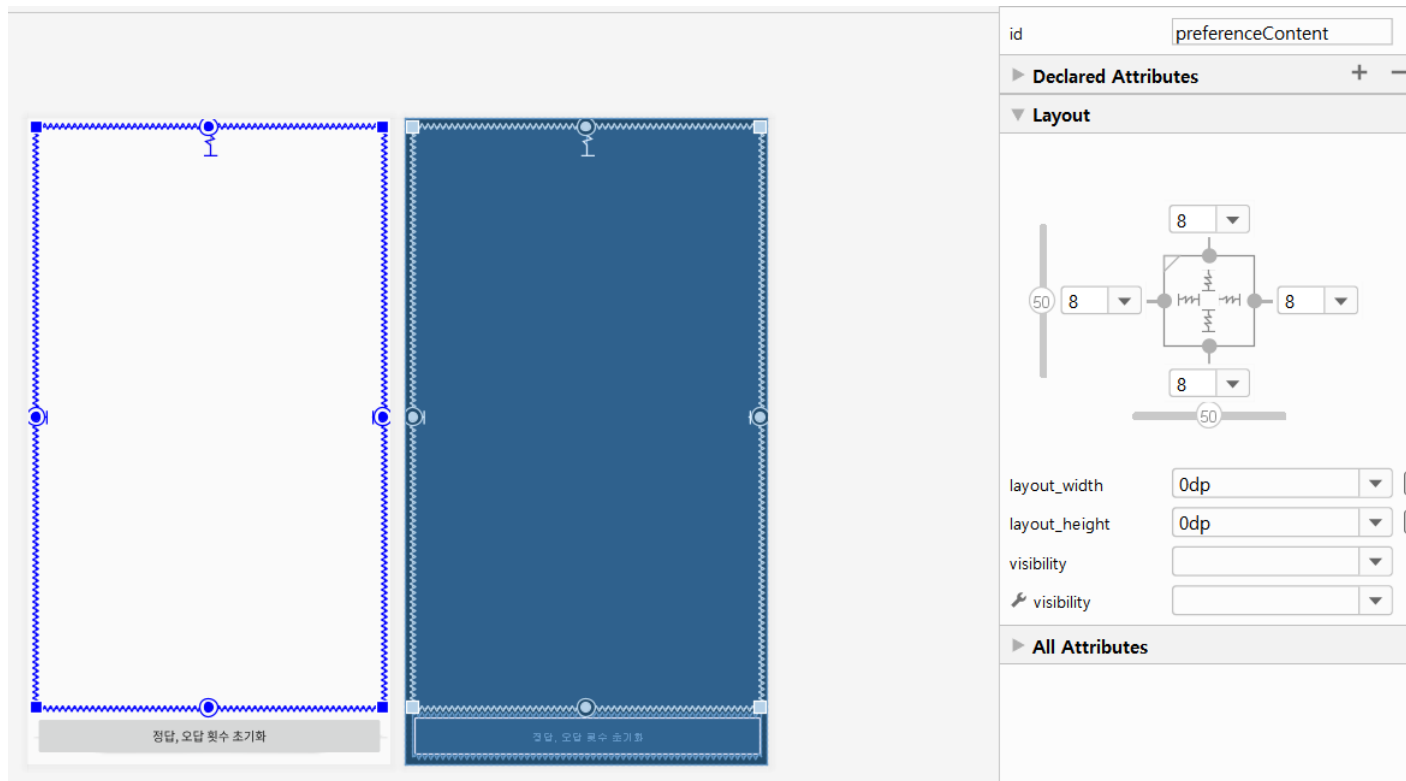
```
<application
    android:allowBackup="true"
    android:icon="@mipmap/kr_todoist"
    android:label="ToDoList"
    android:roundIcon="@mipmap/kr_todoist"
    android:supportRtl="true"
    android:theme="@style/AppTheme">
```

퀴즈 잠금화면 앱 만들기

▶ 화면 구성

▶ FrameLayout과 button추가

- ▶ FrameLayout 내부에 PreferenceFragment를 사용할 예정
- ▶ PreferenceFragment는 SharedPreferences처럼 저장된 데이터를 UI로 보여주는 기능



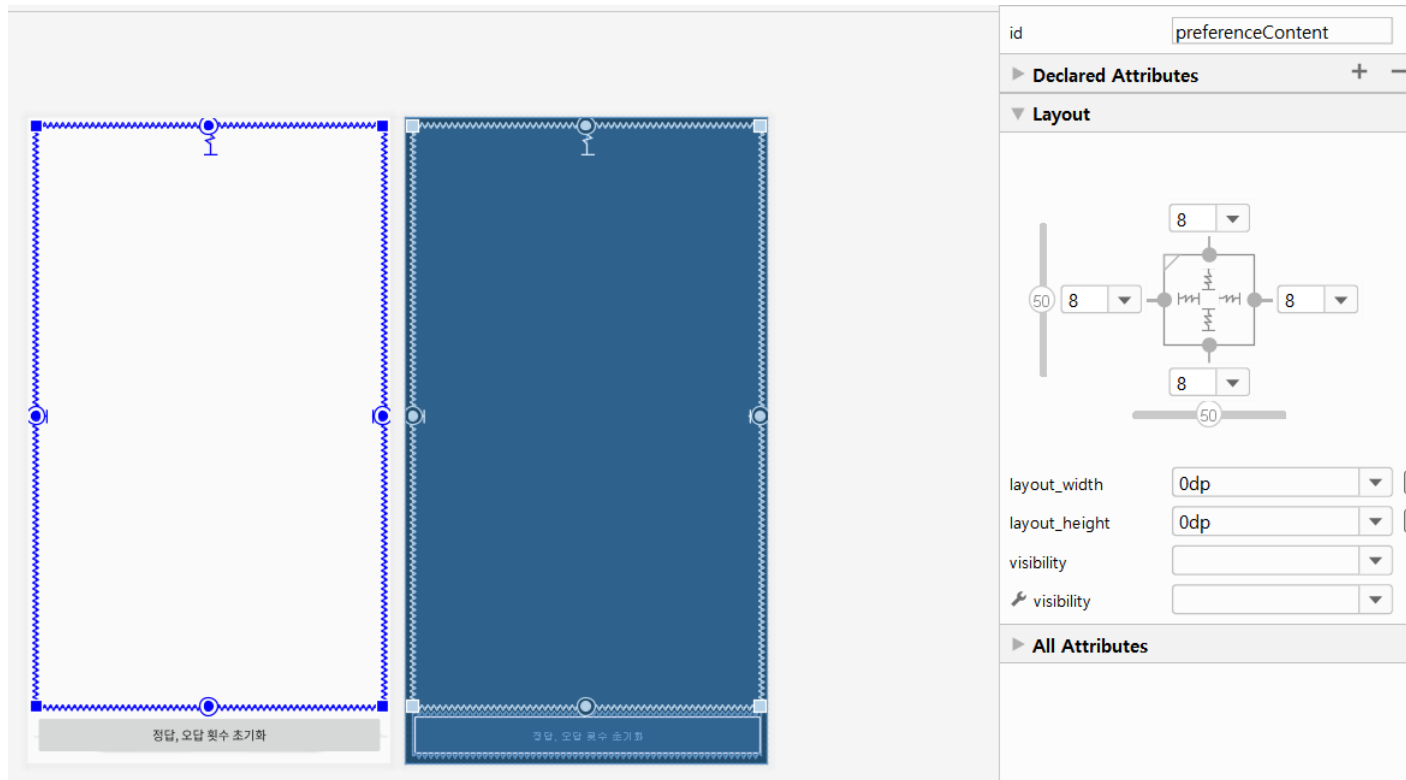
퀴즈 잠금화면 앱 만들기

▶ 화면 구성

▶ FrameLayout과 button추가

▶ FrameLayout 내부에 PreferenceFragment를 사용할 예정

▶ PreferenceFragment는 SharedPreferences처럼 저장된 데이터를 UI로 보여주는 기능



퀴즈 잠금화면 앱 만들기

▶ 화면 구성

▶ xml text 코드

```
<FrameLayout
    android:id="@+id/preferenceContent"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginBottom="8dp"
    app:layout_constraintBottom_toTopOf="@+id/initButton"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">

</FrameLayout>
```


퀴즈 잠금화면 앱 만들기

▶ 화면 구성

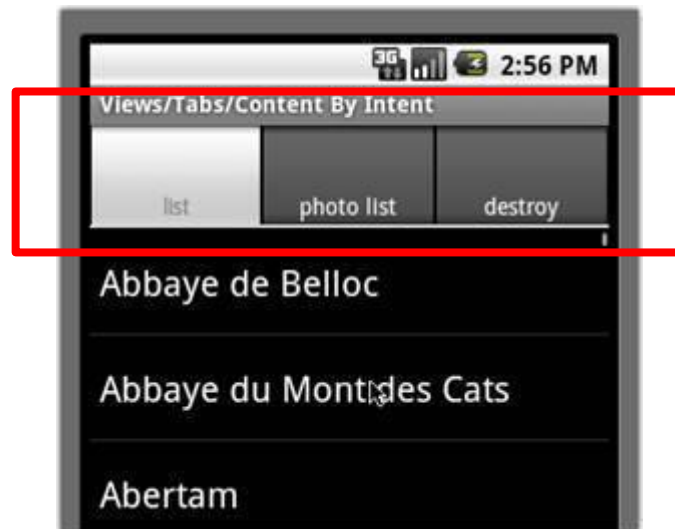
▶ xml text 코드

```
<Button
    android:id="@+id/initButton"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:text="정답, 오답 횟수 초기화"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" android:layout_marginTop="8dp"
    app:layout_constraintTop_toBottomOf="@+id/preferenceContent"/>
```

퀴즈 잠금화면 앱 만들기

▶ 프레임레이아웃(FrameLayout)

- ▶ 단순히 레이아웃 내의 위젯들은 왼쪽 상단부터 겹쳐서 출력
- ▶ 프레임레이아웃 자체로 사용하기보다는 탭 위젯 등과 혼용해서 사용할 때 유용
- ▶ 한 번에 하나의 뷰만 보여주며, 다른 뷰들은 그 아래에 중첩되어 쌓임
- ▶ 중첩되는 효과와 함께 뷰의 가시성(Visibility) 속성을 이용해 다양한 화면 구성이 가능함



퀴즈 잠금화면 앱 만들기

▶ 프레임레이아웃(FrameLayout)

- ▶ 본 프로젝트에서는 설정을 저장하기 위하여 PreferenceFragment를 사용
- ▶ 앞서 학습한 SharedPreferences는 단순히 데이터를 저장하기 위한 방법이고 PreferenceFragment는 Preference 데이터를 UI화하여 보여주는 Fragment
- ▶ 앱의 데이터를 영구적으로 저장하기 위한 방법으로 File과 SharedPreferences를 예제를 통하여 확인

퀴즈 잠금화면 앱 만들기

▶ 파일로 앱 데이터 저장

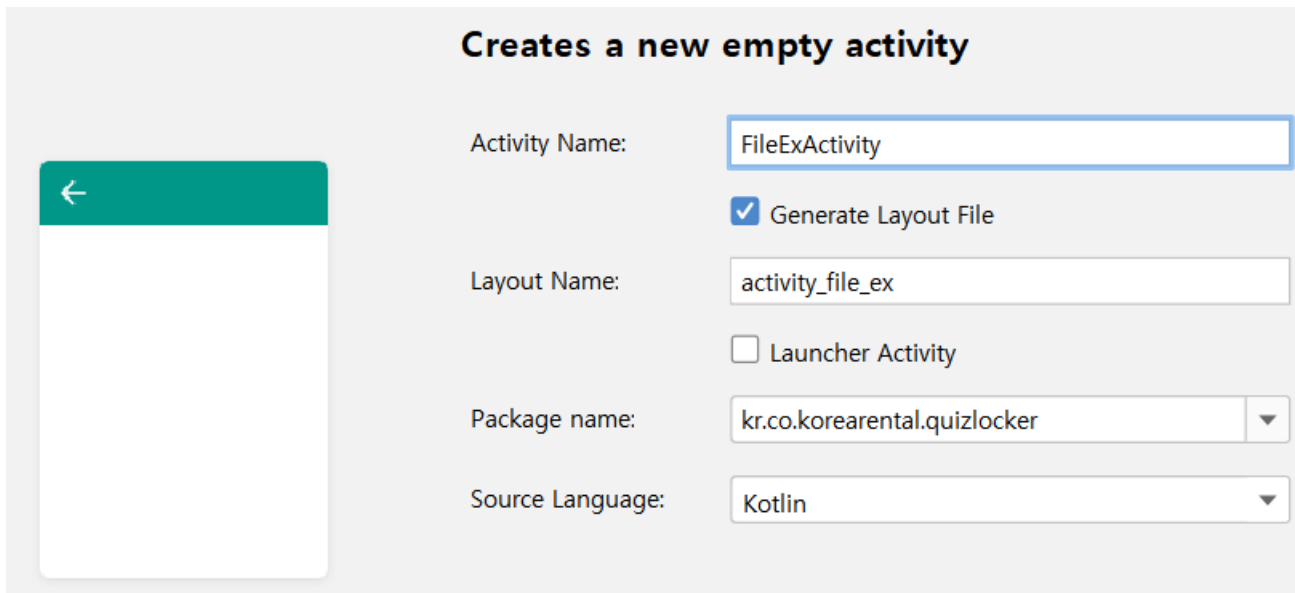
- ▶ File은 대부분의 운영체제에서 사용하고 있는 시스템
- ▶ 안드로이드에서는 파일로 데이터를 저장할 때, 주로 앱 전용 디렉터리를 사용
 - ▷ 앱마다 생성되는 별도의 디렉터리
 - ▷ 다른 앱 간에 서로 접근이 불가능

퀴즈 잠금화면 앱 만들기

▶ 파일로 앱 데이터 저장하는 예제

▶ FileExActivity 추가

▶ File - New - Activity - Empty Activity



Creates a new empty activity

Activity Name:

☒ Generate Layout File

Layout Name:

☐ Launcher Activity

Package name:

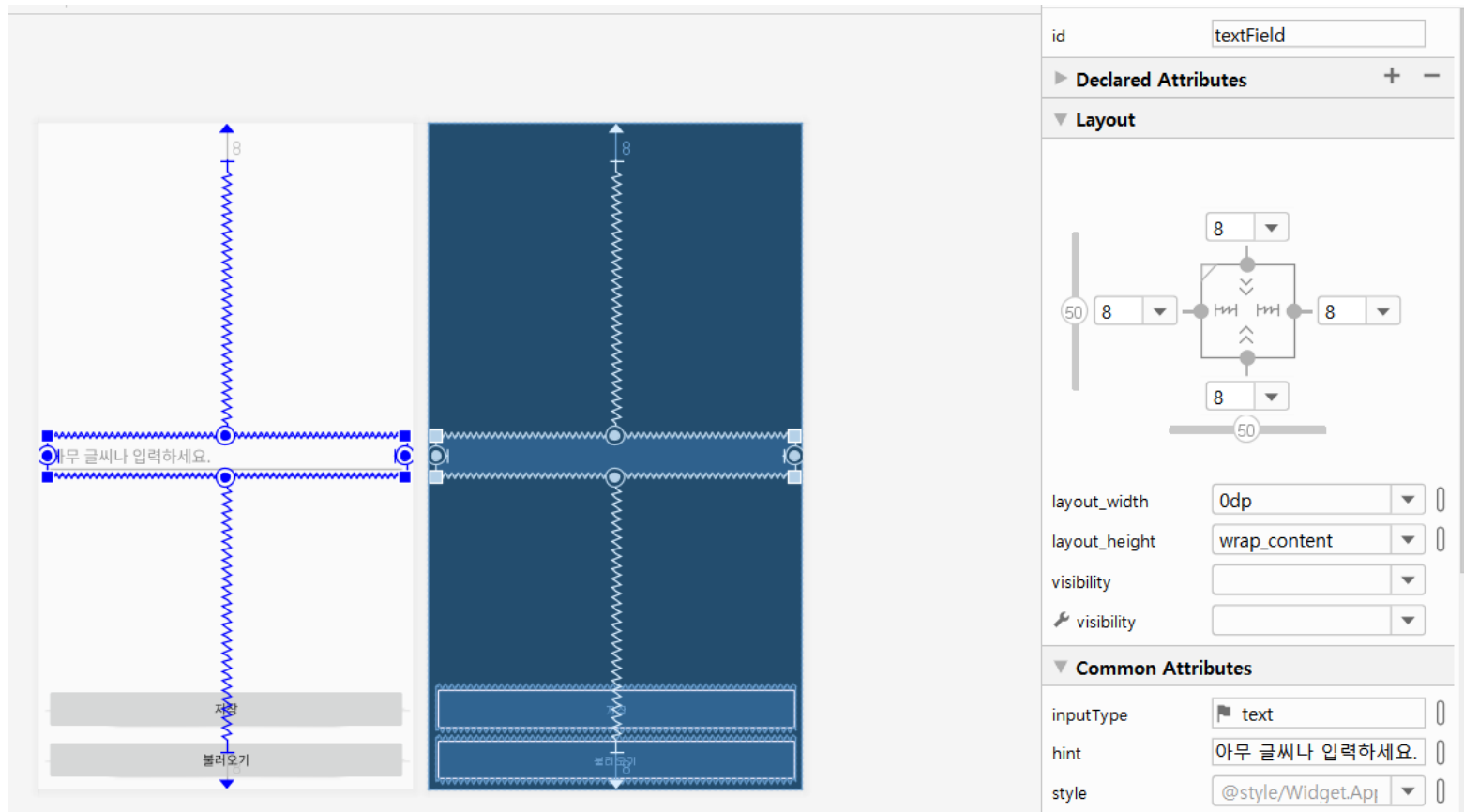
Source Language:

퀴즈 잠금화면 앱 만들기

▶ 파일로 앱 데이터 저장하는 예제

▶ activity_file_ex.xml 파일 편집

▶ edittext 1개, button 2개

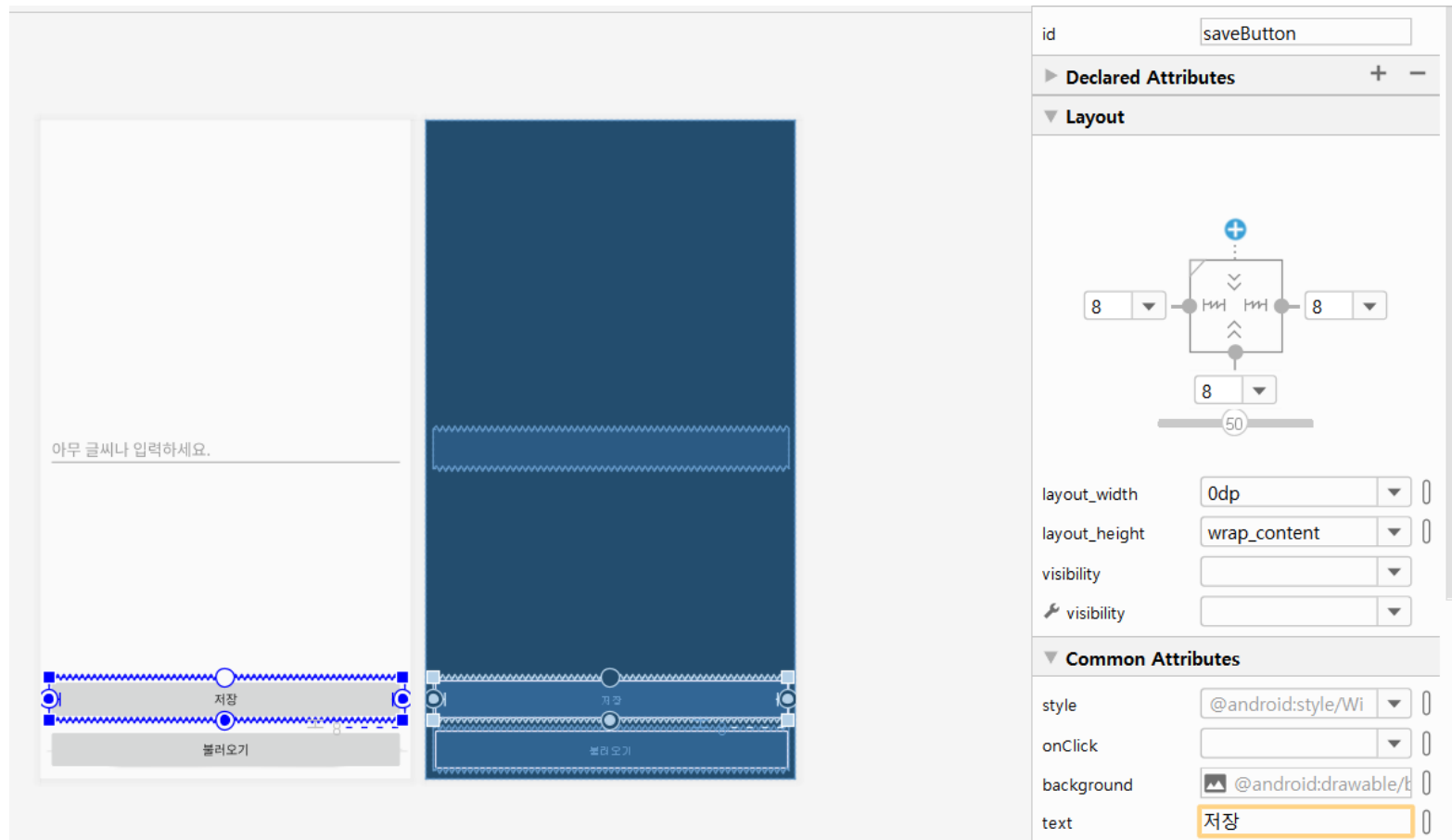


퀴즈 잠금화면 앱 만들기

▶ 파일로 앱 데이터 저장하는 예제

▶ activity_file_ex.xml 파일 편집

▶ edittext 1개, button 2개

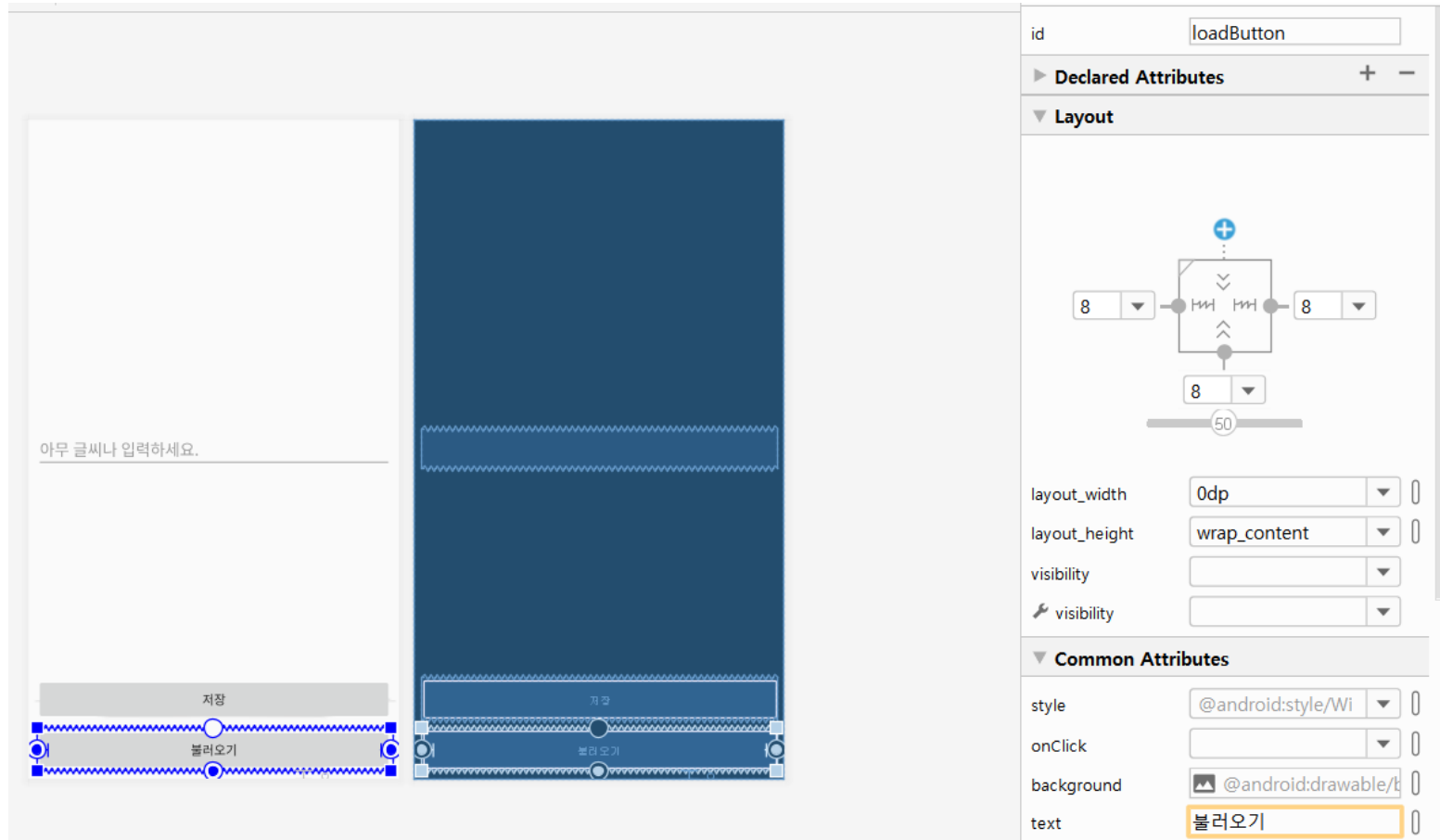


퀴즈 잠금화면 앱 만들기

▶ 파일로 앱 데이터 저장하는 예제

▶ activity_file_ex.xml 파일 편집

▶ edittext 1개, button 2개



퀴즈 잠금화면 앱 만들기

▶ 화면 구성

▶ xml text 코드

```
<EditText
    android:id="@+id/textField"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginBottom="8dp"
    android:ems="10"
    android:hint="아무 글씨나 입력하세요."
    android:inputType="text"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"/>
```

퀴즈 잠금화면 앱 만들기

▶ 화면 구성

▶ xml text 코드

```
<Button
    android:id="@+id/loadButton"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginBottom="8dp"
    android:text="불러오기"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
```

퀴즈 잠금화면 앱 만들기

▶ 화면 구성

▶ xml text 코드

```
<Button
    android:id="@+id/saveButton"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginEnd="8dp"
    android:layout_marginRight="8dp"
    android:layout_marginBottom="8dp"
    android:text="저장"
    app:layout_constraintBottom_toTopOf="@+id/loadButton"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent" />
```

퀴즈 잠금화면 앱 만들기

▶ 파일로 앱 데이터 저장하는 예제

▶ 저장 버튼을 클릭하면 입력한 텍스트를 파일로 저장

▶ 불러오기 버튼을 클릭하면 기존의 파일에서 텍스트를 읽어서 EditText에 출력

▶ FileExActivity를 편집

▶ 내부 저장소에 파일의 텍스트를 저장하고 불러오는 함수를 클래스 내부에 작성

```
// 내부저장소 파일의 텍스트를 저장한다.
fun saveToInnerStorage(text: String, filename: String) {
    // 내부 저장소의 전달된 파일이름의 파일 출력 스트림을 가져온다.
    val fileOutputStream = openFileOutput(filename, Context.MODE_PRIVATE)
    // 파일 출력 스트림에 text 를 바이트로 변환하여 write 한다
    fileOutputStream.write(text.toByteArray())
    // 파일 출력 스트림을 닫는다
    fileOutputStream.close()
}
// 내부저장소 파일의 텍스트를 불러온다
fun loadFromInnerStorage(filename: String): String {
    // 내부저장소의 전달된 파일이름의 파일 입력 스트림을 가져온다
    val fileInputStream = openFileInput(filename)
    // 파일의 저장된 내용을 읽어 String 형태로 불러온다.
    return fileInputStream.reader().readText()
}
```

퀴즈 잠금화면 앱 만들기

▶ 파일로 앱 데이터 저장하는 예제

- ▶ `openFileOutput()`은 앱 전용 내부 저장소 경로에 파일명으로 생성된 파일 출력 스트림 객체를 반환(`FileOutputStream`)
 - ▷ 스트림(stream)은 시간 경과에 따른 연속적인 데이터 흐름을 의미
 - ▷ 스트림을 사용하면 연속적으로 데이터를 저장하는 것이 가능
 - 한번에 10줄이나 1줄 씩 10번에 걸쳐서 저장 가능
- ▶ `openFileOutput()`은 앱의 전용 디렉터리에서 파일 출력스트림을 가져옴
 - ▷ 기본적으로 안드로이드 내부에는 `/data/data` 디렉터리 하위에 패키지 명으로 각 앱의 저장공간이 생성됨
 - ▷ `MODE_PRIVATE`은 파일을 앱 전용으로 만들고 이미 파일이 있는 경우 덮어씀
 - 다른 앱에서 해당 파일에 접근 불가
 - `_APPEND` : 기존 파일에 이어 붙임

퀴즈 잠금화면 앱 만들기

▶ 파일로 앱 데이터 저장하는 예제

▶ 리스너 등록(뒤에 계속)

```
val filename = "data.txt"
// 권한이 있는지 저장하는 변수
override fun onCreate(savedInstanceState: Bundle?) {
    super.onCreate(savedInstanceState)
    setContentView(R.layout.activity_file_ex)

    // 저장 버튼이 클릭된 경우
    saveButton.setOnClickListener {
        // textField 의 현재 텍스트를 가져온다.
        val text = textField.text.toString()
        when {
            // 텍스트가 비어있는 경우 오류 메시지를 보여준다.
            TextUtils.isEmpty(text) -> {
                Toast.makeText(applicationContext, "텍스트가 비어있습니다.",
                    Toast.LENGTH_LONG).show()
            }
            else -> {
                // 내부 저장소 파일에 저장하는 함수 호출
                saveToInnerStorage(text, filename)
            }
        }
    }
}
```

퀴즈 잠금화면 앱 만들기

▶ 파일로 앱 데이터 저장하는 예제

▶ 리스너 등록(계속)

```
// 불러오기 버튼이 클릭된 경우
loadButton.setOnClickListener {
    try {
        // textField 의 텍스트를 불러온 텍스트로 설정한다.
        textField.setText(loadFromInnerStorage(filename))
    } catch (e: FileNotFoundException) {
        // 파일이 없는 경우 에러메세지 보여줌
        Toast.makeText(applicationContext, "저장된 텍스트가 없습니다.",
            Toast.LENGTH_LONG).show()
    }
}
```

▶ `openFileInput()` 함수를 사용할 때 파일이 없으면 `FileNotFoundException`이 발생하므로 try - catch문 사용

퀴즈 잠금화면 앱 만들기

▶ 파일로 앱 데이터 저장하는 예제

▶ 테스트를 위하여 매니페이스 수정

▷ 런처 액티비티를 테스트용 액티비티로 변경

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Quizlocker"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportRtl="true"
    android:theme="@style/AppTheme">
    <activity android:name=".PreFragmentActivity">
    </activity>
    <activity android:name=".FileExActivity">
        <intent-filter>
            <action android:name="android.intent.action.MAIN"/>

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


퀴즈 잠금화면 앱 만들기

▶ 파일로 앱 데이터 저장하는 예제

▶ 실행하기

▷ 특정 문자를 저장한 후 삭제하고 불러오기 클릭

퀴즈 잠금화면 앱 만들기

▶ 파일로 앱 데이터 저장하는 예제

▶ 외부 저장소 사용 및 권한 요청

- ▶ 앱의 외부 저장소는 탈착이 가능한 SD카드 저장소일 수 있고 탈착이 불가능한 메모리일 수도 있으나 대부분 지원
 - 내부 저장소의 용량이 부족한 경우가 많기 때문에
- ▶ 최근에 SD카드의 추가가 불가능한 스마트 폰이 많지만 논리적으로 외부저장소를 분리하여 제공
 - 윈도우에서 c,d 드라이브를 구분하는 것과 같음
- ▶ 외부저장소에 파일을 저장할 경우 사용 권한 요청과 외부 저장소가 없을 경우에 대한 예외처리가 필요

퀴즈 잠금화면 앱 만들기

▶ 파일로 앱 데이터 저장하는 예제

▶ 외부 저장소 사용 및 권한 요청

▶ 매니페스트에 권한요청 관련 편집

```
■ <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
1<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="kr.co.korearental.quizlocker">

    <!-- 외부저장소에 읽기 쓰기 권한 요청, 앱 전용 데이터로 저장하는 경우 SDK 18 이후부터는 권한 요청이 필요 없음. -->
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

1    <application
        android:allowBackup="true"
```

▶ 권한 요청을 명시하면 사용자에게 사용자에게 앱이 사용하는 기능을 예측하게 할 수 있음

▶ maxSdkVersion 속성으로 특정 버전이후로 권한을 요청하지 않도록 할 수 있음

- <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" android:maxSdkVersion="18"/>

퀴즈 잠금화면 앱 만들기

▶ 파일로 앱 데이터 저장하는 예제

▶ 외부 저장소 앱 전용 디렉토리에 파일 저장

▷ FileExActivity파일에 아래 함수를 추가

```
// 외부 저장장치를 사용할 수 있고 쓸수 있는지 체크하는 함수
fun isExternalStorageWritable(): Boolean {
    when {
        // 외부저장장치 상태가 MEDIA_MOUNTED 면 사용 가능
        Environment.getExternalStorageState() == Environment.MEDIA_MOUNTED -> return true
        else -> return false
    }
}

// 외부저장장치에서 앱 전용데이터로 사용할 파일 객체를 반환하는 함수
fun getAppDataFileFromExternalStorage(filename: String): File {
    // KITKAT 버전 부터는 앱전용 디렉토리의 디렉토리 타입 상수인 Environment.DIRECTORY_DOCUMENTS 를
    val dir = if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT) {
        getExternalFilesDir(Environment.DIRECTORY_DOCUMENTS)
    } else {
        // 하위 버전에서는 직접 디렉토리 이름 입력
        File(Environment.getExternalStorageDirectory().absolutePath + "/Documents")
    }
    // 디렉토리의 경로중 없는 디렉토리가 있다면 생성한다.
    dir?.mkdirs()
    return File("${dir.absolutePath}${File.separator}${filename}")
}
```

지원

퀴즈 잠금화면 앱 만들기

▶ 파일로 앱 데이터 저장하는 예제

▶ 외부 저장소 앱 전용 디렉토리에 파일 저장

▷ isExternalStorageWritable()

- 외부 저장장치가 현재 사용 가능한지 여부를 확인하는 함수
- 외부 저장장치가 탈착이 가능할수도 있기 때문

▷ getAppDataFileFromExternalStorage()

- 외부 저장장치에 위한 앱 전용 디렉터리로부터 파라미터로 전달 받은 이름(filename)의 파일 객체를 반환
- Environment.DIRECTORY_DOCUMENTS 상수가 킷캣 버전 이후로만 지원되기 때문에 버전에 따라 다른 코드를 사용
- 제조사마다 디렉터리 경로나 이름이 다를 수 있으므로 API(Environment.DIRECTORY_DOCUMENTS) 상수를 사용
 - 대부분 /sdcard/Android/data/앱 패키지명

퀴즈 잠금화면 앱 만들기

▶ 파일로 앱 데이터 저장하는 예제

▶ 외부 저장소 앱 전용 디렉토리에 파일 저장

▷ FileExActivity파일에 아래 함수를 추가

```
// 외부저장소 앱 전용 디렉토리에 파일로 저장하는 함수
fun saveToExternalStorage(text: String, filename: String) {
    val fileOutputStream = FileOutputStream(getAppDataFileFromExternalStorage(filename))
    fileOutputStream.write(text.toByteArray())
    fileOutputStream.close()
}

// 외부저장소 앱 전용 디렉토리에서 파일 데이터를 불러오는 함수
fun loadFromExternalStorage(filename: String): String {
    return FileInputStream(getAppDataFileFromExternalStorage(filename)).reader().readText()
}
```

퀴즈 잠금화면 앱 만들기

▶ 파일로 앱 데이터 저장하는 예제

▶ 외부 저장소 앱 전용 디렉토리에 파일 저장

▶ 외부에 저장할 수 있도록 리스너 수정

```
// 저장 버튼이 클릭된 경우
saveButton.setOnClickListener {
    // textField 의 현재 텍스트를 가져온다.
    val text = textField.text.toString()
    when {
        // 텍스트가 비어있는 경우 오류 메시지를 보여준다.
        TextUtils.isEmpty(text) -> {
            Toast.makeText(applicationContext, "텍스트가 비어있습니다.",
                Toast.LENGTH_LONG).show()
        }
        !isExternalStorageWritable() -> {
            Toast.makeText(applicationContext, "외부 저장장치가 없습니다.",
                Toast.LENGTH_LONG).show()
        }
        else -> {
            // 내부 저장소 파일에 저장하는 함수 호출
            //saveToInnerStorage(text, filename)
            // 외부 저장소 파일에 저장하는 함수 호출
            saveToExternalStorage(text, filename)
        }
    }
}
```

퀴즈 잠금화면 앱 만들기

▶ 파일로 앱 데이터 저장하는 예제

▶ 외부 저장소 앱 전용 디렉토리에 파일 저장

▶ 외부에 저장할 수 있도록 리스너 수정

```
// 불러오기 버튼이 클릭된 경우
loadButton.setOnClickListener {
    try {
        // textField 의 텍스트를 불러온 텍스트로 설정한다.
        // textField.setText(loadFromInnerStorage(filename))
        // 외부저장소 앱전용 디렉토리의 파일에서 읽어온 데이터로 textField 의 텍스트를 설정
        textField.setText(loadFromExternalStorage(filename))
    } catch (e: FileNotFoundException) {
        // 파일이 없는 경우 에러메세지 보여줌
        Toast.makeText(applicationContext, "저장된 텍스트가 없습니다.",
            Toast.LENGTH_LONG).show()
    }
}
```


실습 예제 - 일기장 만들기

▶ 캘린더로 날짜를 지정하고 에디트 텍스트에 일기를 작성하고 저장하는 앱

▶ 새 프로젝트 생성 - DiaryKR

▶ 디자인은 오른쪽 화면과 유사하게 작성

▶ 캘린더는 스피너 타입이며 실행시 오늘 날짜로 초기화

▶ 로또 프로젝트 참고

▶ 작성된 이후에 버튼을 누르면 내부 저장소에 년_월_일.txt로 저장

▶ 2019_04_27.txt

■ 로또 프로젝트의 seed값 만들기 참고

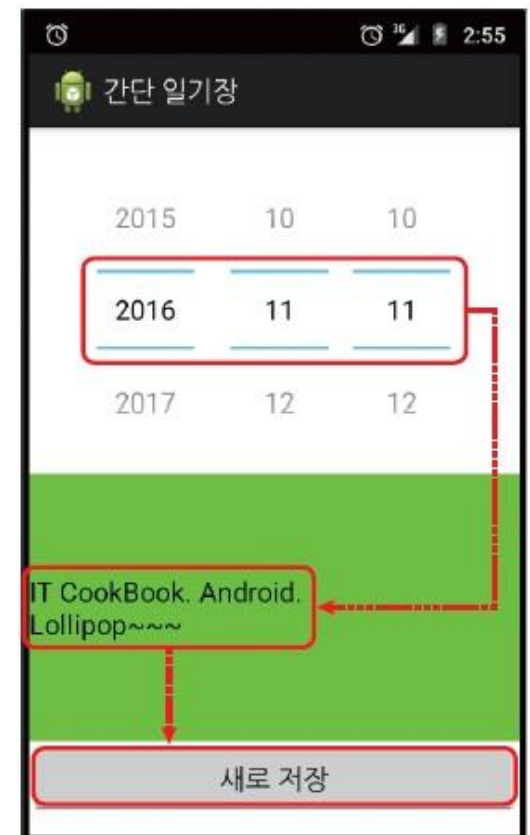
▶ 토스트 메시지로 “파일명 저장됨” 출력

■ 2019_04_27.txt 저장됨

▶ 날짜를 변경하면 해당 날짜의 일기가 출력 - 버튼 변경(수정하기)

▶ 없다면 " 일기 없음 " 으로 hint 가 출력

■ editText.setHint(string)



Q & A
