

# Android



# Final Android Resizer 툴 이용하자.

목적 : 이미지를 각 폰 해상도에 맞게 자동 생성해준다.

Google

android final resizer

All Videos News Images Shopping More Settings Tools

About 9,560,000 results (0.29 seconds)

[GitHub - asystat/Final-Android-Resizer: A simple yet powerful resizer ...](https://github.com/asystat/Final-Android-Resizer)

<https://github.com/asystat/Final-Android-Resizer> ▾

Final Android Resizer. Select starting density. Select res directory, so you don't have to move the resized files. Resize several images at once (Drag & Drop) Select output densities (ldpi, mdpi, tvdpi, hdpi, xhdpi, xxhdpi) Preserve the image format (jpg,png) Option to select output directory (mipmap or ...)

[Executable Jar](#)  
A simple yet powerful resizer for  
Android Image resources ...

[Final-Android-Resizer.iml](#)  
Final-Android-Resizer/Final-Android-  
Resizer.iml. Fetching ...

[asystat/Final-Android-Resizer](#)

GitHub is where people build

1. 무료 이미지 다운로드
2. 안드로이드에서 사용하기 위해서 파일처리 **final android resizer**
3. 위 파일 존재하는 폴더에서 파워쉘 창열기
4. **java -jar Fin탭 (자동완성)**
5. **Final Android Resizer** 를 이용해서,  
이미지파일을 변환시킴
6. 안드로이드 res 폴더 안에다가,  
폴더 6개 붙여넣기

# res 폴더에는 이름 규칙이 엄격.

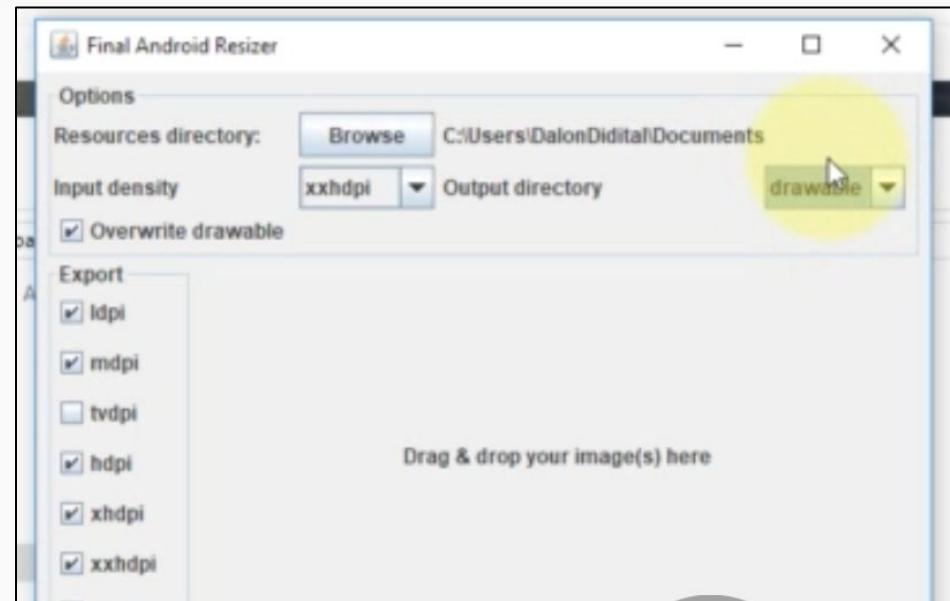
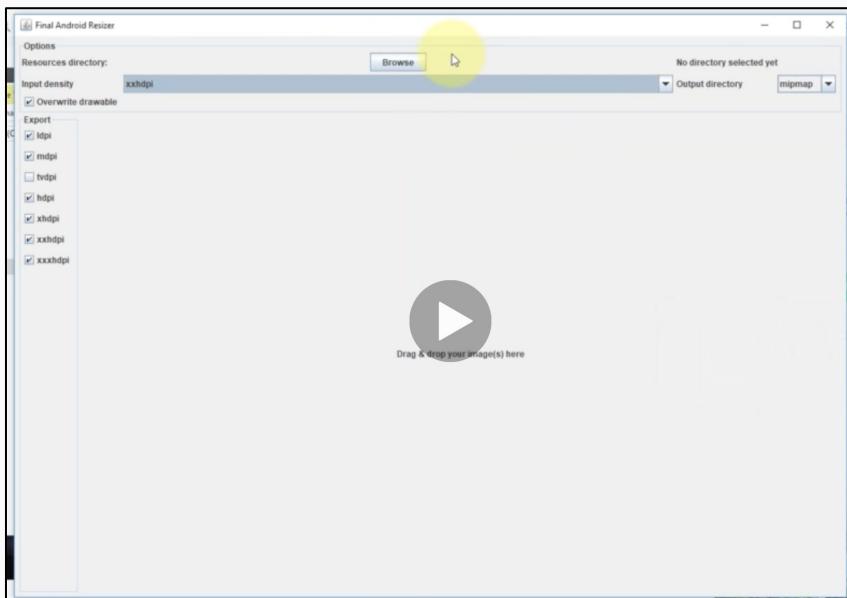
소문자로 되어있어야 함.

중간 공백 허용 X

중간에 - (하이픈) X

중간에 \_ (언더스코어) O

브라우즈 버튼 눌러서, 어느 폴더에 저장할지 먼저 선택하고,  
input density : xxhdpi, output directory : drawable  
로 설정



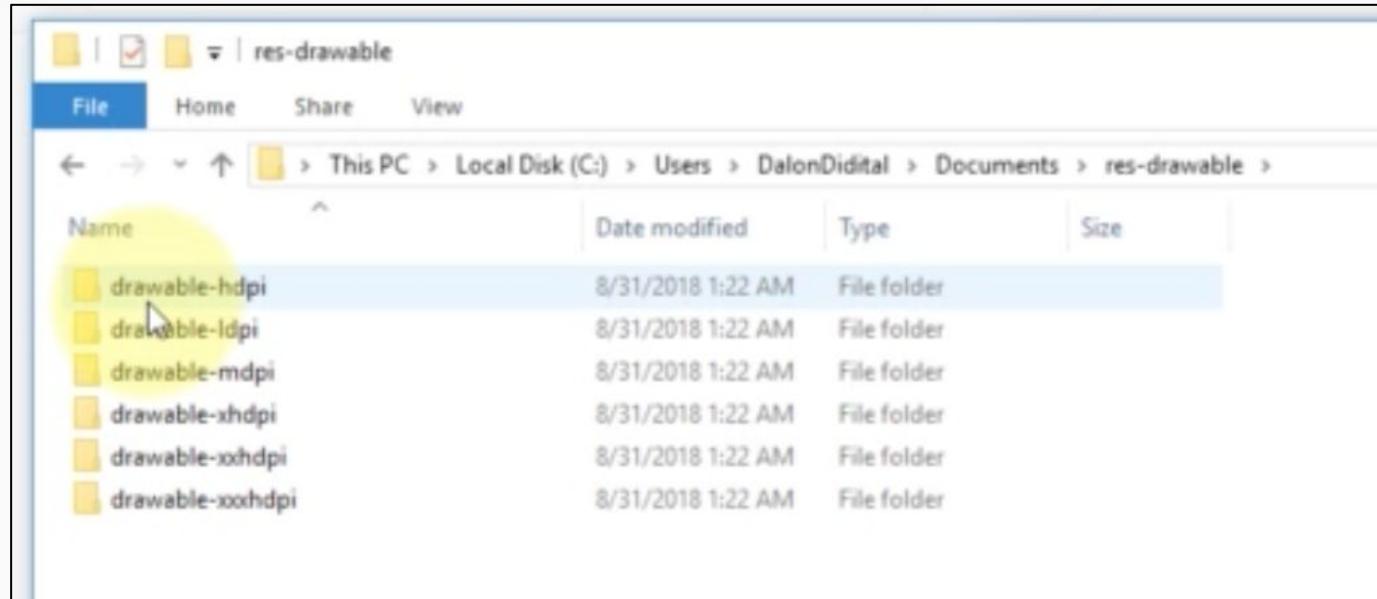
목적 : xxhdpi 란 그럼 무엇이냐? 안드로이드 공식페이지  
가보자

다음처럼 검색하면, 설명나온다.

A screenshot of a Google search results page. The search query "Support different pixel densities" is entered into the search bar. The results page shows the following information:

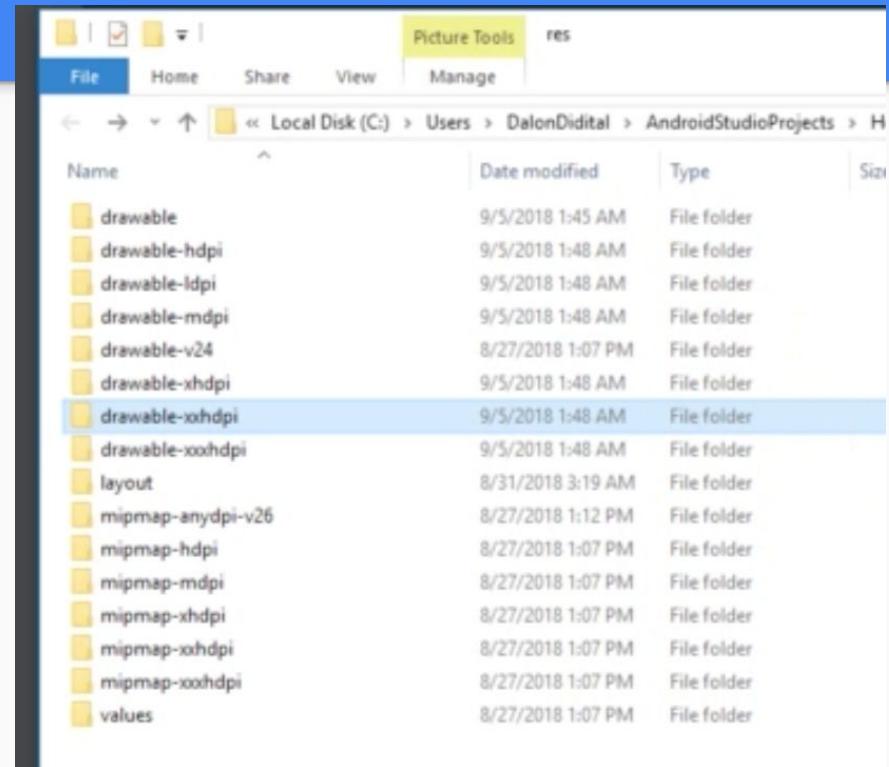
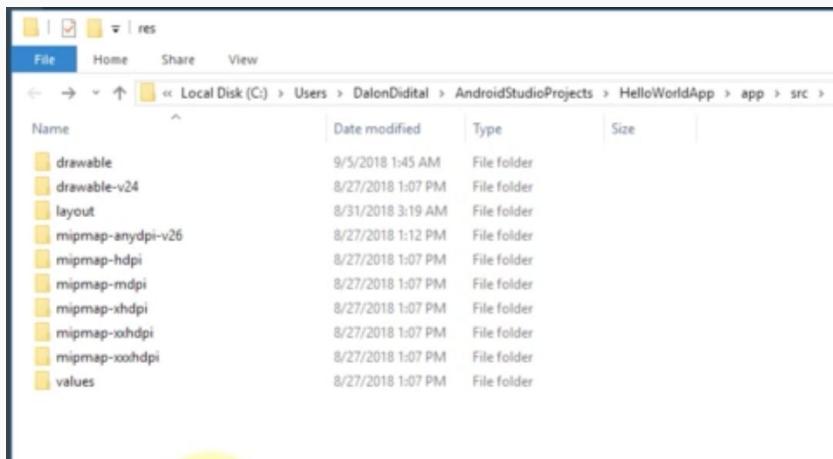
- The search term "Support different pixel densities" is displayed above the results.
- The "All" tab is selected, followed by "Images", "Shopping", "Videos", "News", and "More".
- The message "About 40,600,000 results (0.42 seconds)" is shown.
- The top result is a link titled "Support different pixel densities | Android Developers" with the URL <https://developer.android.com/training/multiscreen/screendensities>.
- The snippet for this result describes the density-independent bitmap system and provides alternative bitmap resources for different screen densities.

이제 픽사베이에서 가져온 이미지를 드래그 앤 드랍하여 드로어블 폴더 만든다.  
경로로 이동해 보자

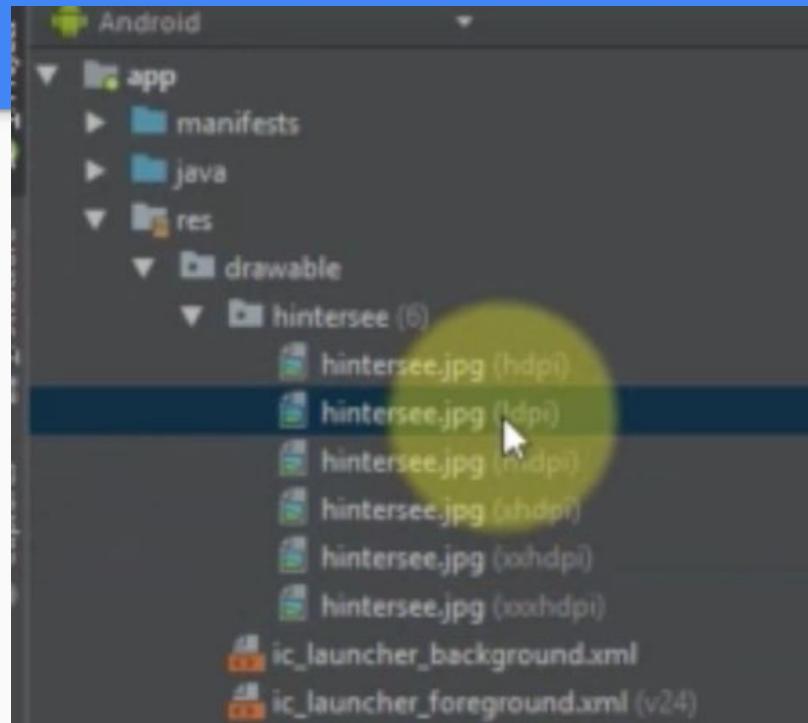


위의 모든 폴더를 다 선택하여 폴더 자체를 전부  
카피한후,  
안드로이드 스튜디오에서 Drawable 우클릭 하여, 경로  
폴더 연다.

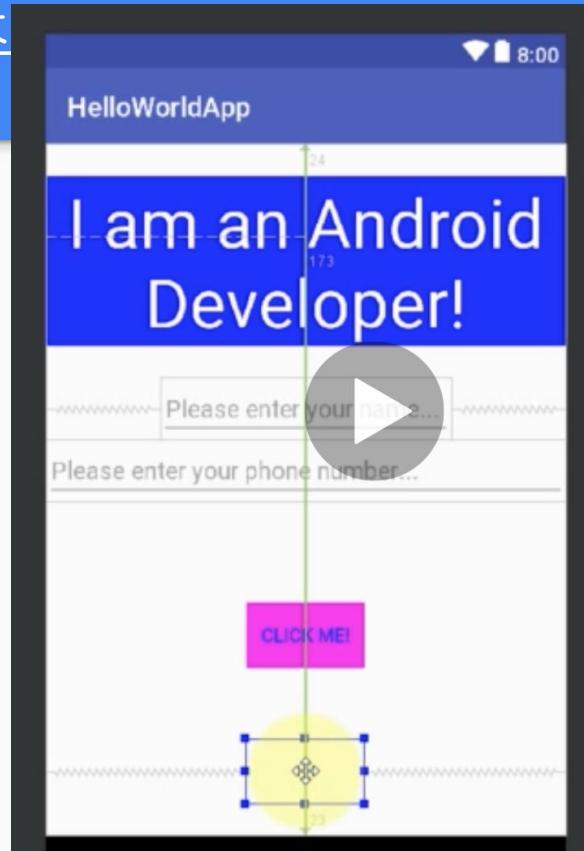
여기에 카피한거 복사한다.



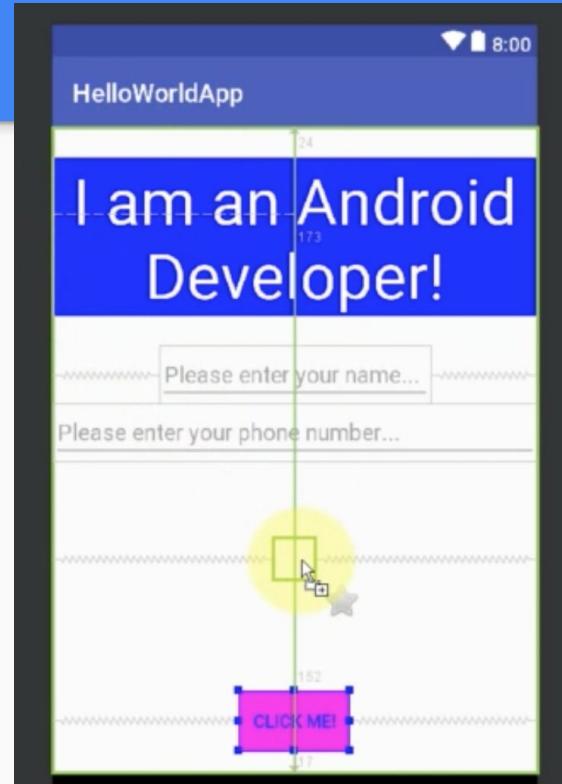
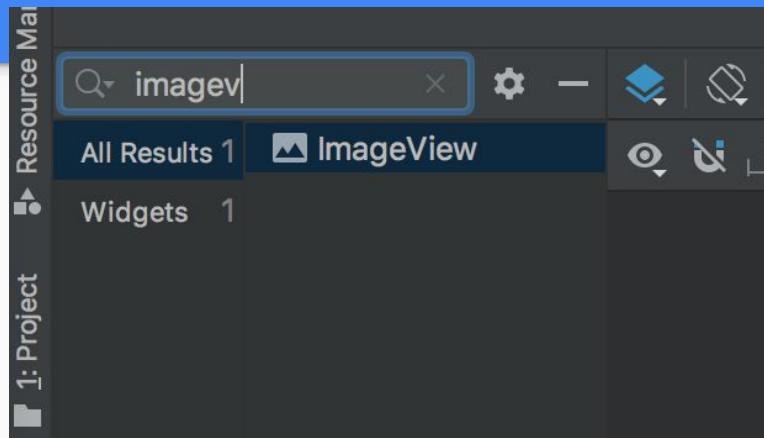
그러면, 안드로이드 스튜디오 왼쪽 res - drawables 확인



목적 : 이제 이미지뷰 만들어서 이미지 넣어보자  
원래 버튼을 클릭해서 아래로 드래그 하여 아래  
그림처럼 움직



이미지뷰 검색하여, 찾아서, 드래그해서 넣는다.



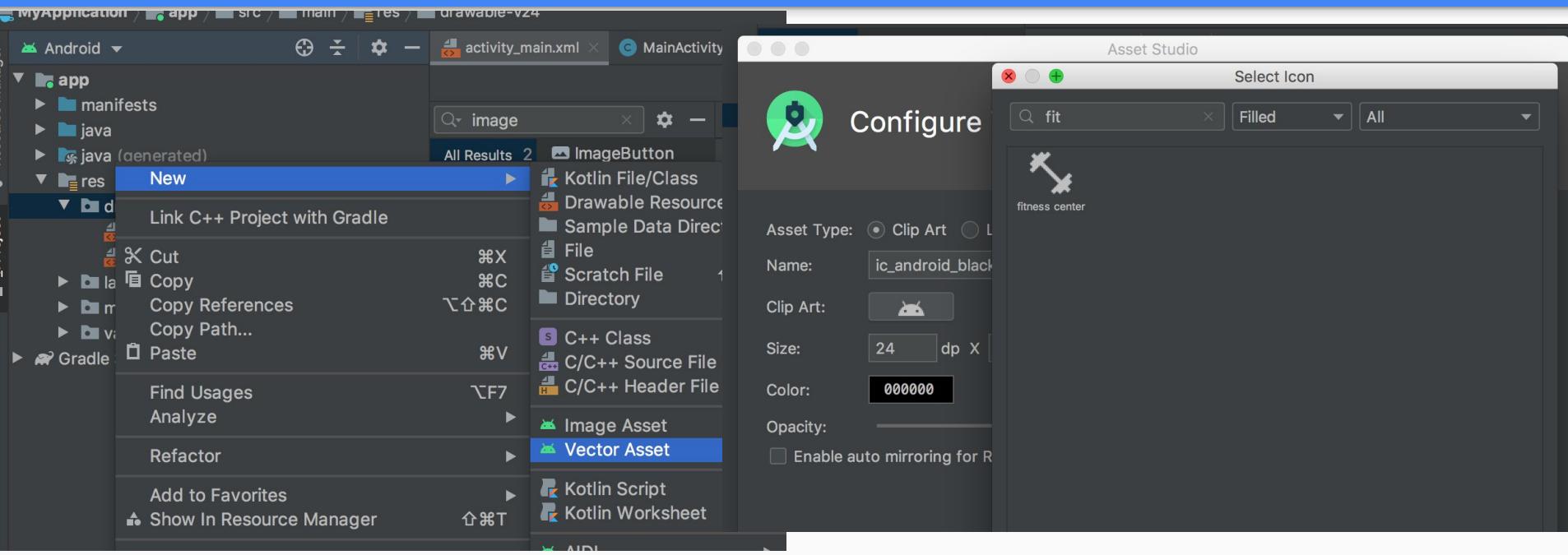
xml 확인해 본다.

src 부분을 바꾸면, 이미지가 바뀐다.

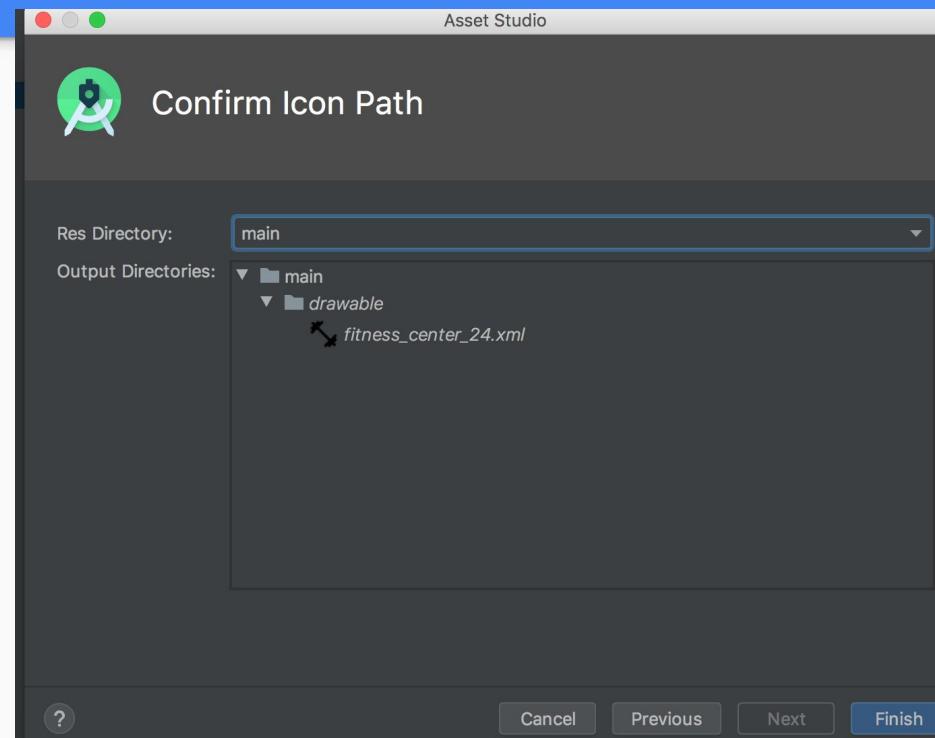
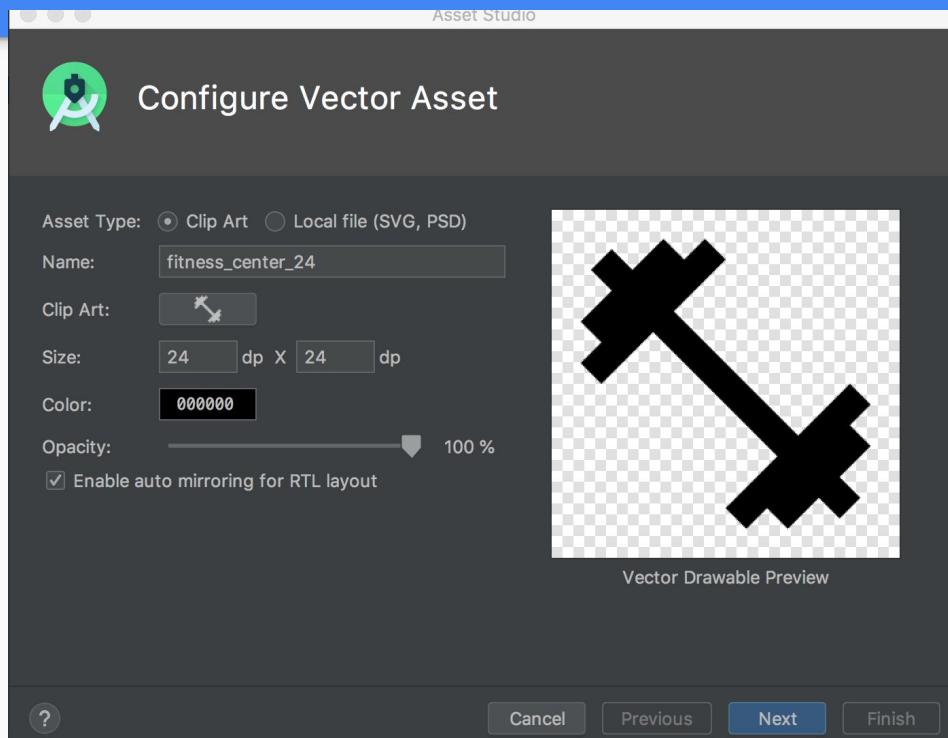
```
<ImageView  
    android:id="@+id/imageView"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_alignParentBottom="true"  
    android:layout_centerHorizontal="true"  
    android:layout_marginBottom="106dp"  
    android:scaleType="centerInside"  
    android:src="@drawable/orangejuice"  
/>
```

목적 : 아이콘 이용해 본다.

res - drawable 우클릭하여, new - vector asset 선택  
Clip Art 눌러서 아이콘 검색 버튼에, fit 검색

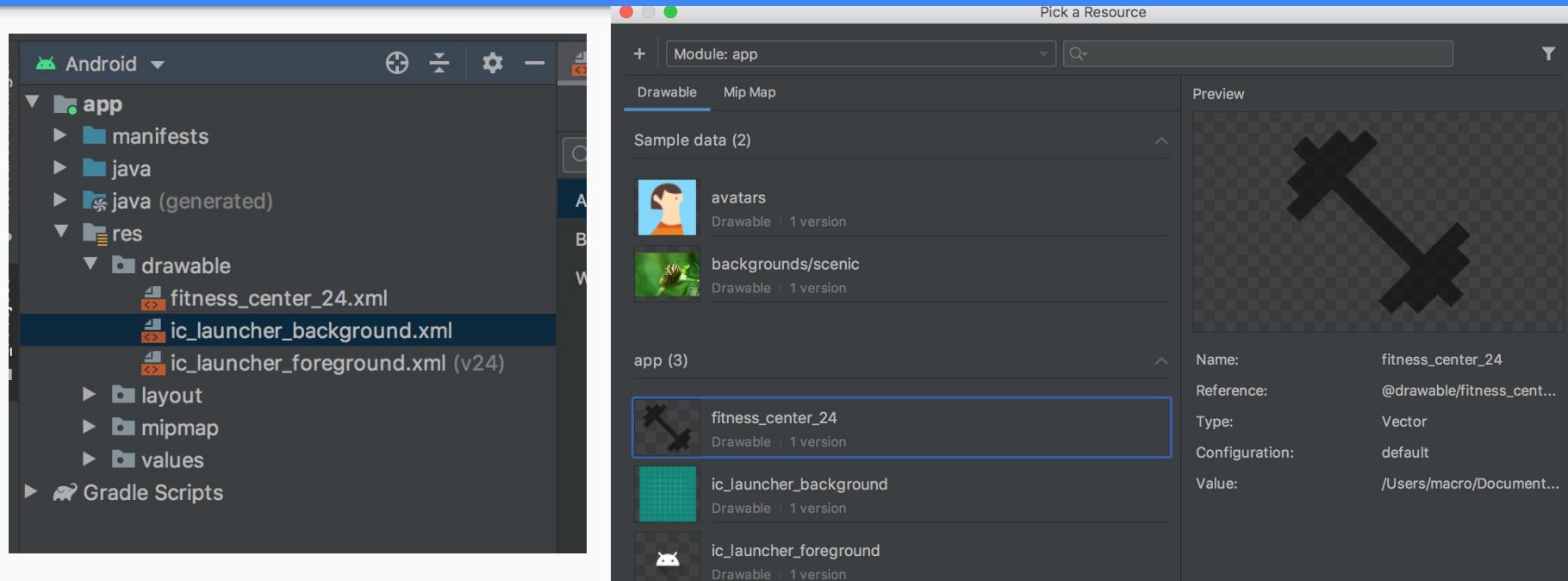


Name에 fitness라고 변경하고, 맨아래 체크한다.  
경로 그대로 두고 피니시



본격 `drawable`에 새도 생기고,  
새로 이미지뷰 끌어다 놓으면, app 부분에 이미지  
표시된다.

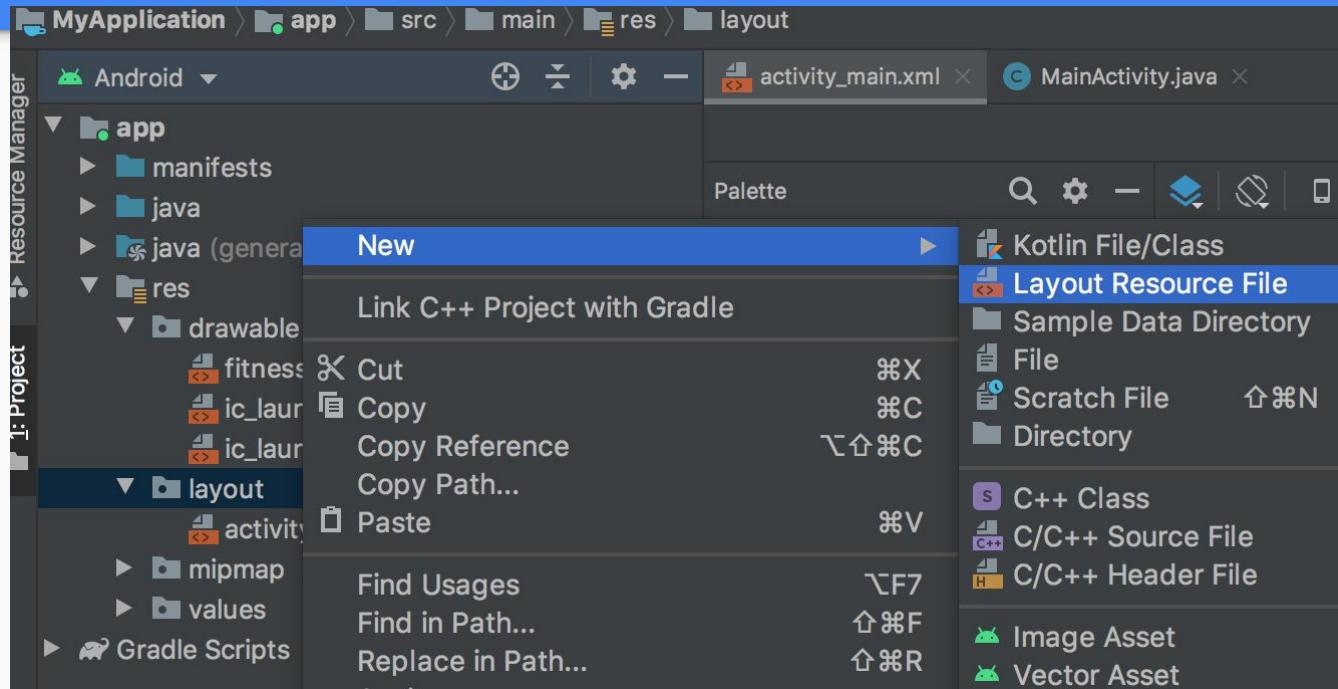
벡터이미지는? 자동으로 해상도 맞게 조절되는 이미지



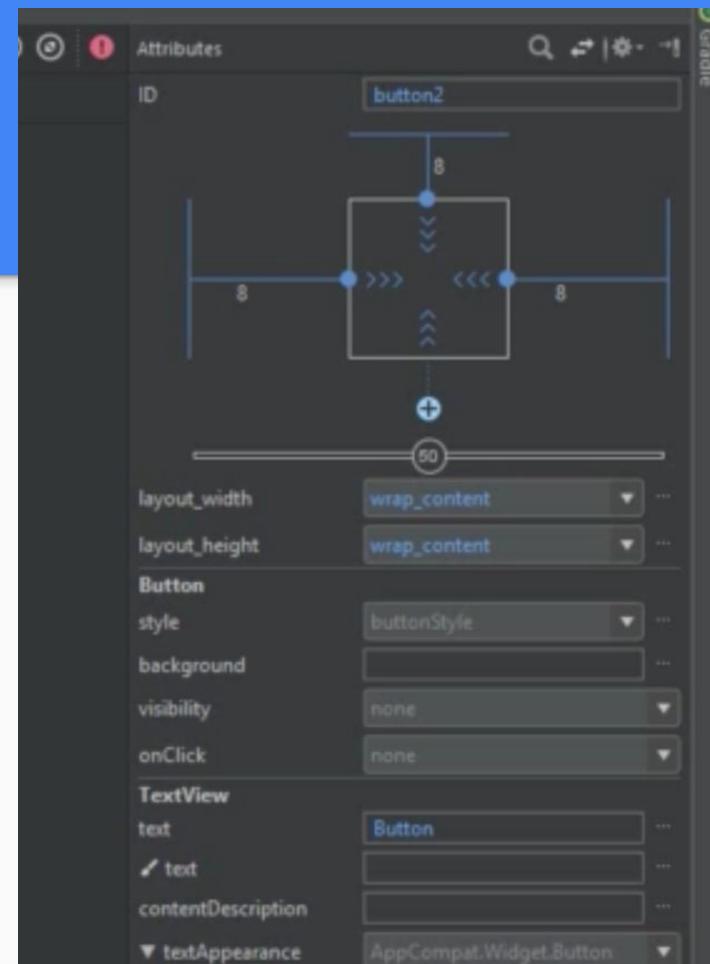
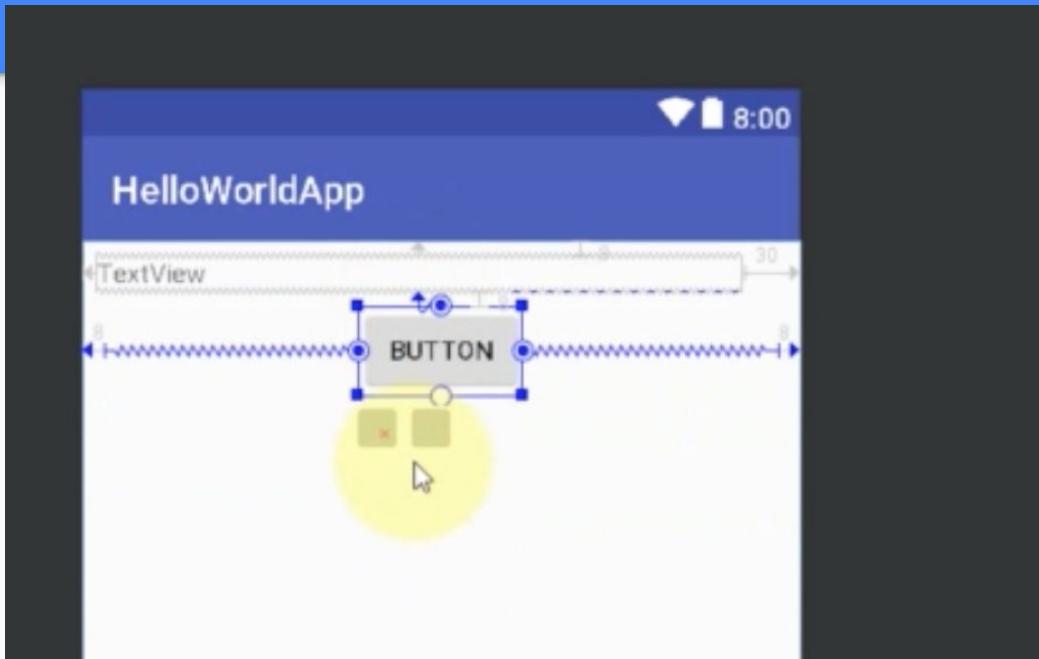
목적 : 버튼클릭하면, 이미지뷰 이미지 바꿔도록 해본다.

```
ImageView myImage = findViewById(R.id.img);  
myImage.setImageResource(R.drawable.hintersee);
```

res - layout 우클릭 - new - 레이아웃리소스파일  
파일이름적고, ConstraintLayout 으로 만든다.

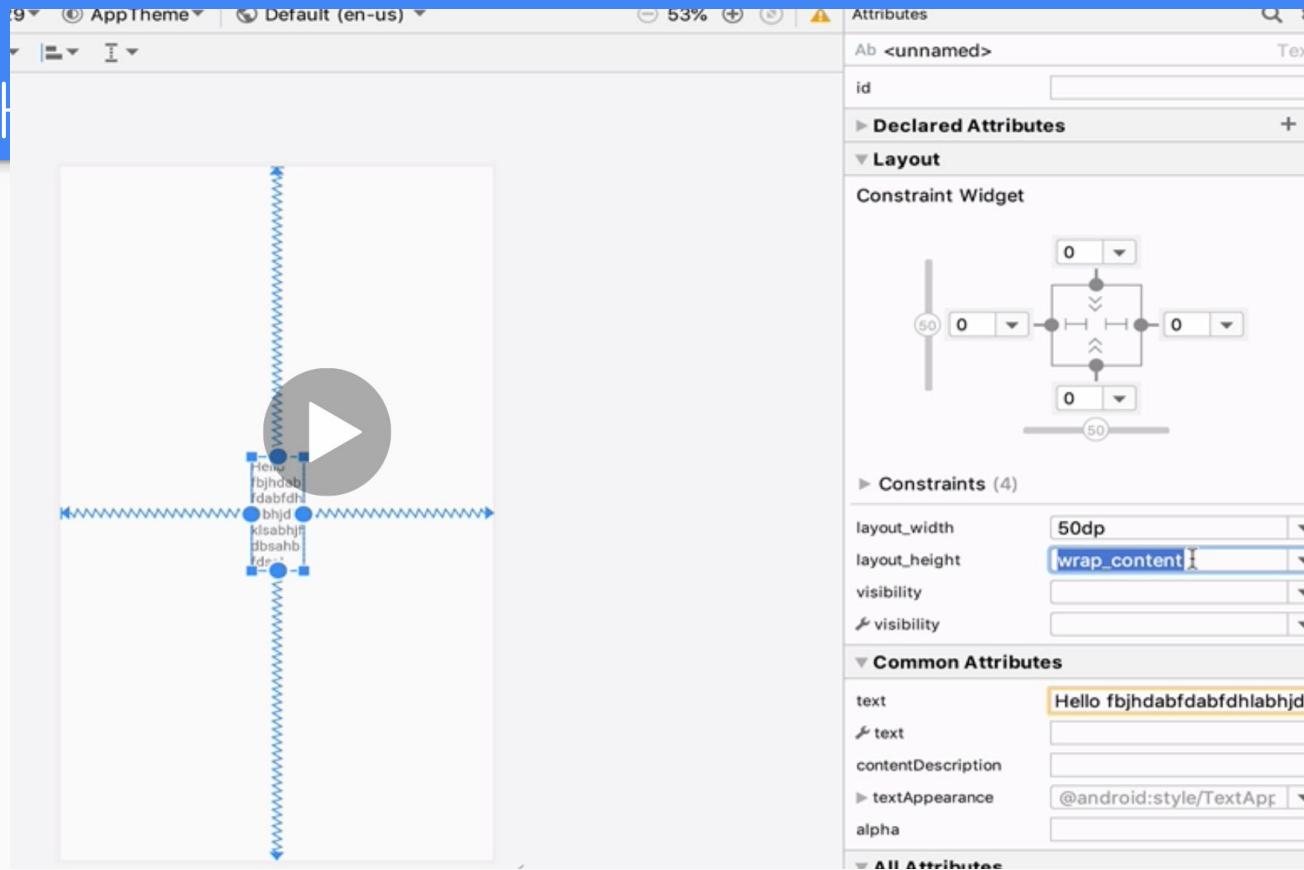


# 다음처럼 해본다.



컨스트레인 레이아웃 - 텍스트뷰 하나 만들고  
layout\_width 를 wrap\_content 로 하여 글자를 길게  
넓어본다.

50dp로 바꿔본다



이렇게 하면, 글자수 많은 적든, 페어런트에 맞게  
맞춰준다.

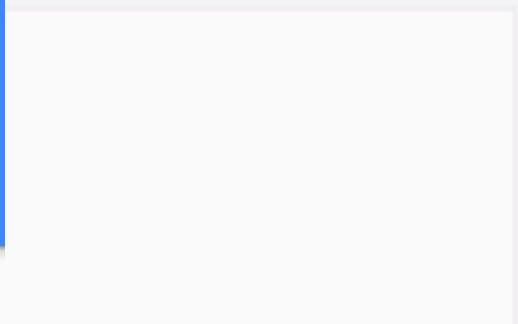
extView

```
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:text="Hello fbjhdbabfdabfdhlabhjdbjklsal"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
```

► Declared Attributes

▼ Layout

Constraint Widget



Wrap Content

16 16

24

50

▼ Constraints

- Start → StartOf parent (16dp)
- End → EndOf parent (16dp)
- Bottom → BottomOf parent (24dp)

layout_width	wrap_content
layout_height	wrap_content
visibility	
✓ visibility	

▼ Common Attributes

style	@android:style/Widget.Material
onClick	
background	@android:drawable/btn_def
text	Button
✓ text	

좌우 스크롤 하면, 옆으로 이동한다.

The screenshot shows the Android Studio interface with the Constraint Layout editor open. On the left, a preview of a button with horizontal constraints is visible. On the right, the constraint graph and properties panel are shown.

**Constraint Graph:**

- The top node is a square with a plus sign (+).
- It has four children: two "16" nodes (top-left and top-right), a "24" node (bottom), and a "30" node (bottom-right).
- Horizontal arrows point from the "16" nodes to the "24" node, and from the "24" node to the "30" node.
- Vertical arrows point from the "16" nodes to the "30" node.

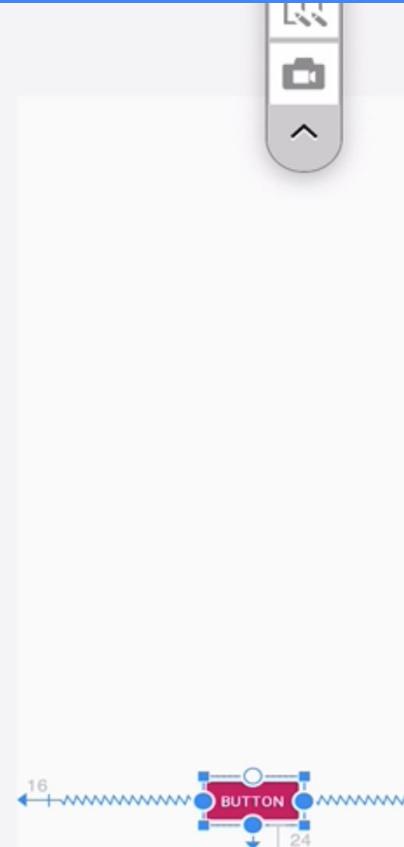
**Properties Panel:**

- Constraints:**
  - Start → StartOf **parent** (16dp)
  - End → EndOf **parent** (16dp)
  - Bottom → BottomOf **parent** (24dp)
  - Horizontal Bias (0.3)
- Common Attributes:**
  - style: @android:style/Widget.Mater...
  - onClick: (empty)
  - background: @android:drawable/btn\_de...

# 버튼 스타일 바꿔줘 본다.

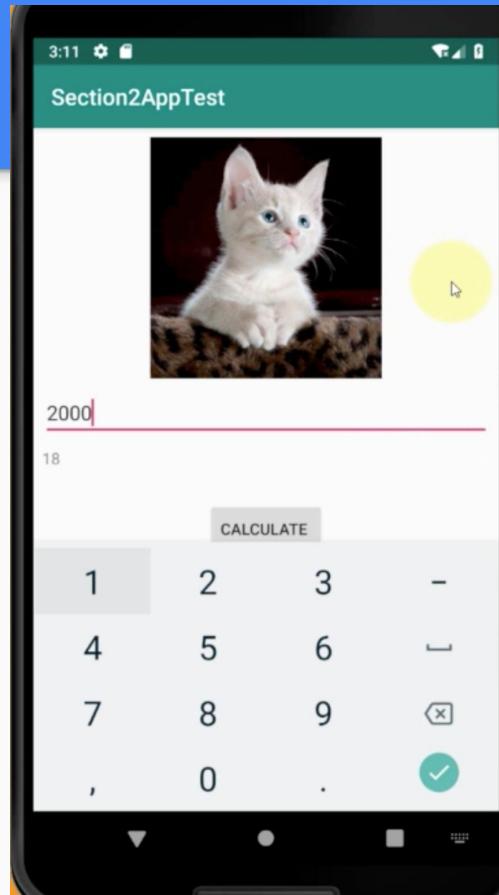
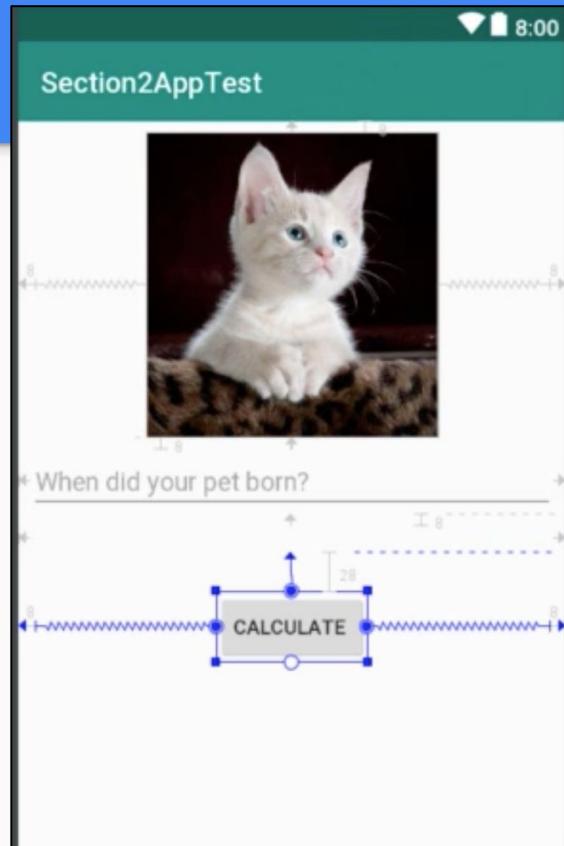
tton

```
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginEnd="16dp"
    android:layout_marginRight="16dp"
    android:layout_marginBottom="24dp"
    android:text="Button"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    style="@style/Widget.AppCompat.Button.Colored"/>
idx.constraintlayout.widget.ConstraintLayout>
```



실습 . 샘플앱 .  
태어난 년도 넣으면 나이 알려주는  
앱

다음과 같은 앱을 만들것. Calendar 클래스 Year 가져오기



캘린더 사용법 설명 먼저 한다.

```
public void calculateButtonPressed(View buttonView) {  
  
    EditText petBornTextField = findViewById(R.id.edtPetBorn);  
    int petAge = Calendar.getInstance().get(Calendar.YEAR) - Integer.parseInt(petBornTextField.getText().toString());  
    TextView petAgeTextView = findViewById(R.id.txtPetYearsOld);  
    petAgeTextView.setText(petAge + "");  
}
```

```
etBorn);  
YEAR) - Integer.parseInt(petBornTextField.getText().toString());  
YearsOld);
```

샘플앱 개발. 퀴즈앱

## Configure your project

Name

QuizApp

Package name

morteza.packag.com.quizapp

Save location

/Users/mortezasaadat/AndroidStudioProjects/QuizApp



Language

Java



Minimum API level

API 16: Android 4.1 (Jelly Bean)



Empty Activity

ⓘ Your app will run on approximately **99.6%** of devices.

[Help me choose](#) This project will support instant apps Use androidx.\* artifacts

Creates a new empty activity

Cancel

Previous

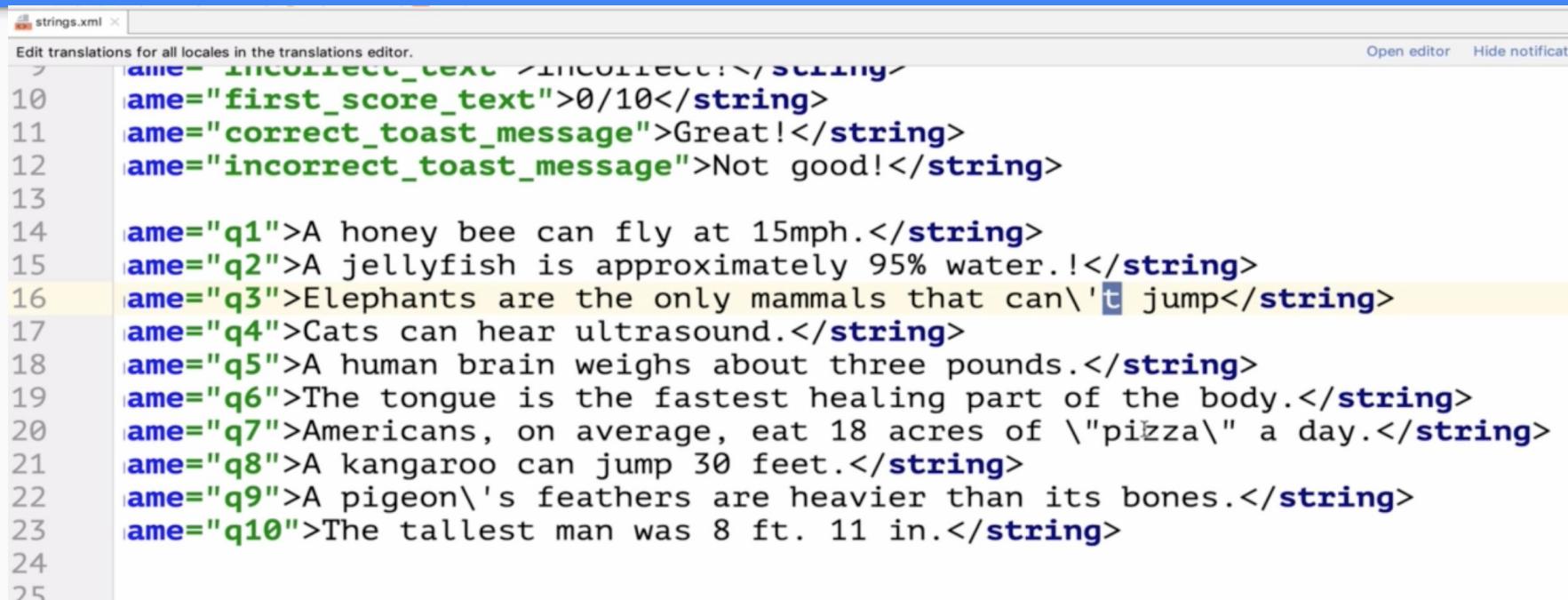
Next

Finish

# strings.xml 파일에 대해서 설명.

문자열 처리할 파일은, 우리 깃허브 사이트

02\_QuizApp\_Strings 폴더에 있음



The screenshot shows the Android Studio interface with the 'strings.xml' file open. The file contains 10 entries, each defining a string resource with a key and a value. The keys are 'incorrect\_text', 'first\_score\_text', 'correct\_toast\_message', 'incorrect\_toast\_message', and 'q1' through 'q10'. The values are descriptive sentences about various animals. The code is color-coded for readability, with tags in blue and strings in green. The editor has a toolbar at the top with buttons for 'Open editor' and 'Hide notifications'.

```
strings.xml
Edit translations for all locales in the translations editor.
10 <?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="incorrect_text">INCORRECT!</string>
    <string name="first_score_text">0/10</string>
    <string name="correct_toast_message">Great!</string>
    <string name="incorrect_toast_message">Not good!</string>
    <string name="q1">A honey bee can fly at 15mph.</string>
    <string name="q2">A jellyfish is approximately 95% water.!</string>
    <string name="q3">Elephants are the only mammals that can't jump</string>
    <string name="q4">Cats can hear ultrasound.</string>
    <string name="q5">A human brain weighs about three pounds.</string>
    <string name="q6">The tongue is the fastest healing part of the body.</string>
    <string name="q7">Americans, on average, eat 18 acres of "pizza" a day.</string>
    <string name="q8">A kangaroo can jump 30 feet.</string>
    <string name="q9">A pigeon's feathers are heavier than its bones.</string>
    <string name="q10">The tallest man was 8 ft. 11 in.</string>
</resources>
```

# 리니어 레이아웃으로 변경

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

</LinearLayout>
```

## 텍스트뷰 / 프로그레스바 설정

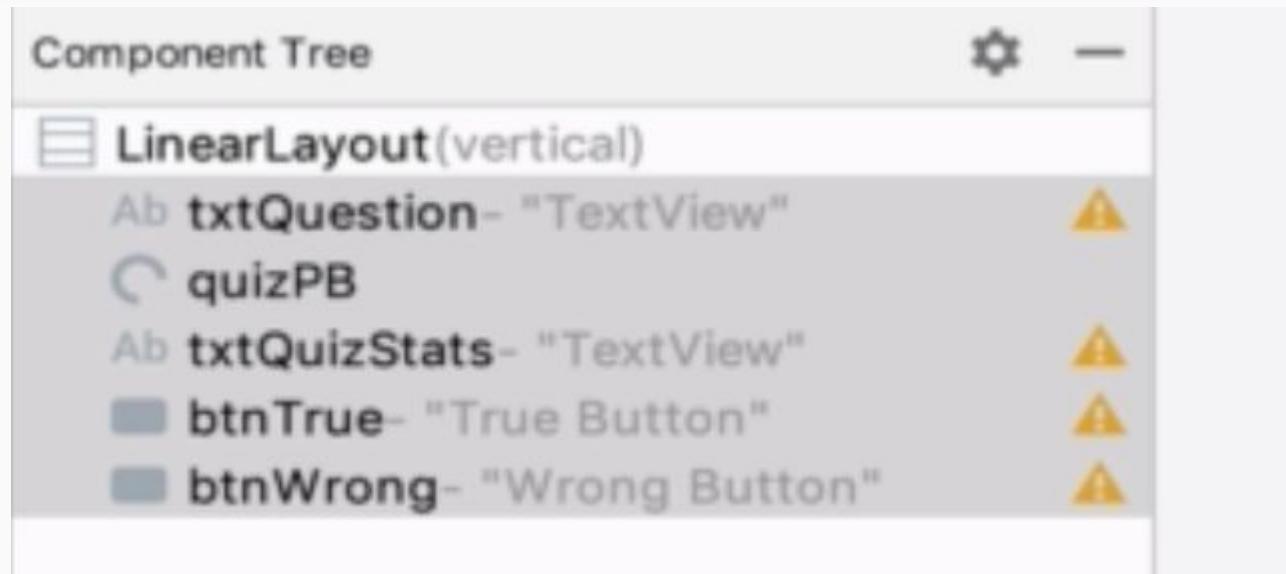
```
<TextView  
    android:id="@+id/txtQuestion"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="TextView"  
    android:gravity="center_horizontal"/>
```



```
<ProgressBar  
    android:id="@+id/quizPB"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```

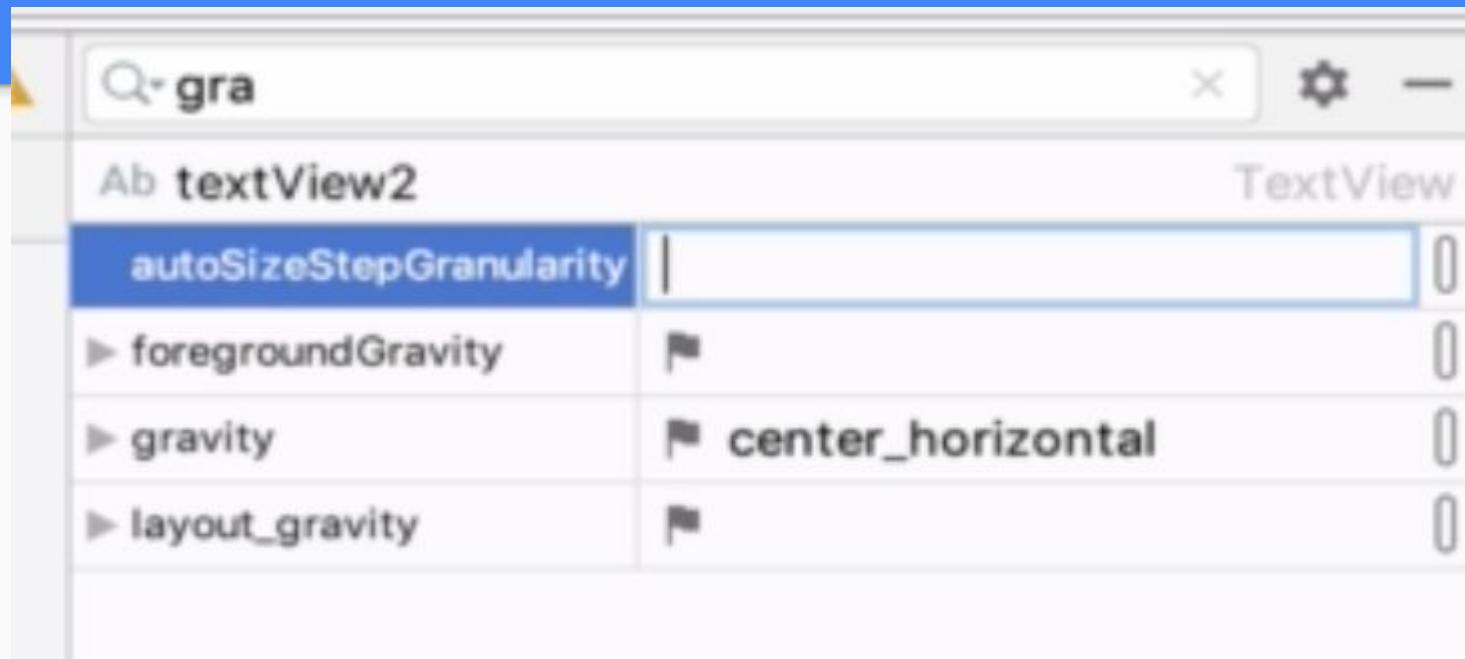
아이디는 다음처럼 한다.

아래 아이디를 참고해서 개발하자



두번째 텍스트뷰 추가.

아이디는 txtQuizStart, 그래비티 설정



번튼 2개 추가. 백그라운드 #0000ff, #ff0000  
텍스트칼라 #ffffff

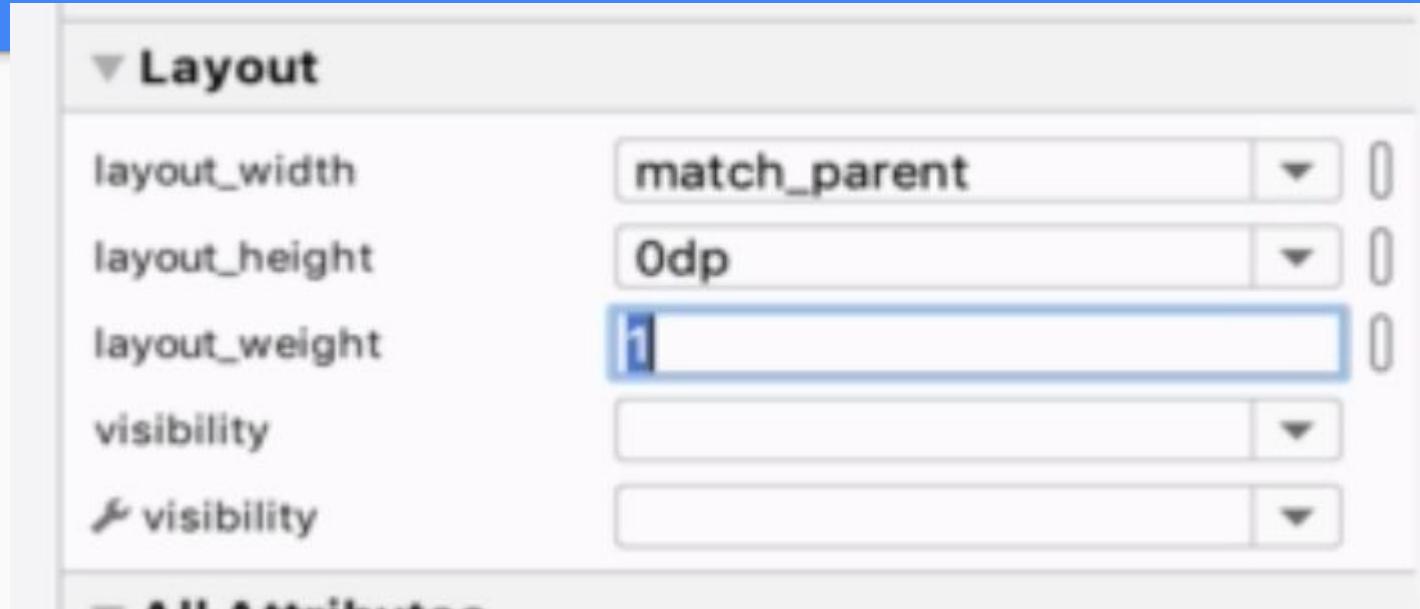


왼쪽에서 모두 선택

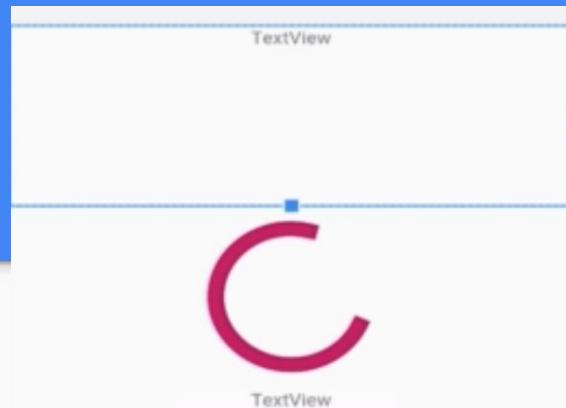
Component Tree

Component	Type	Status
LinearLayout (vertical)		
Ab txtQuestion - "TextView"	"TextView"	⚠️
C quizPB	"Button"	
Ab txtQuizStats - "TextView"	"TextView"	⚠️
B btnTrue - "True Button"	"Button"	⚠️
B btnWrong - "Wrong Button"	"Button"	⚠️

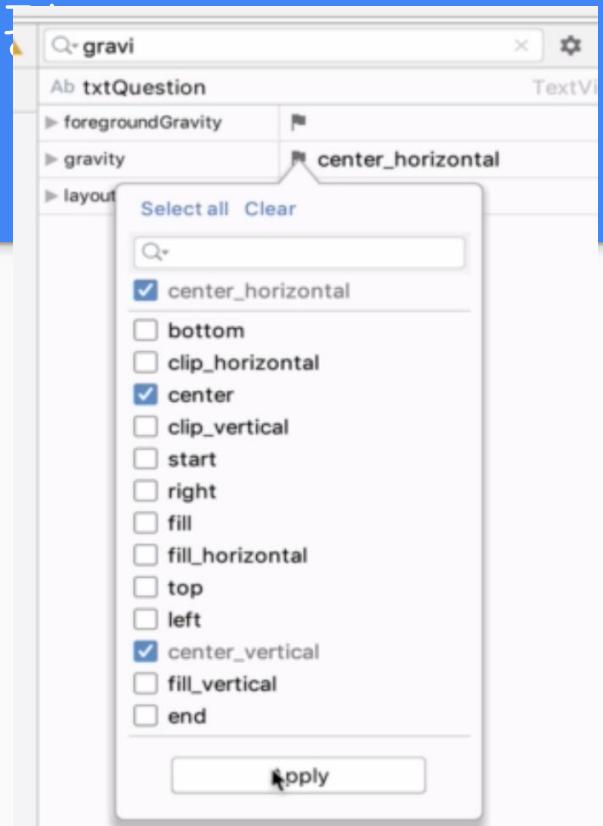
오른쪽에서 하이트와 웨이트 값 0 과 1로 변경



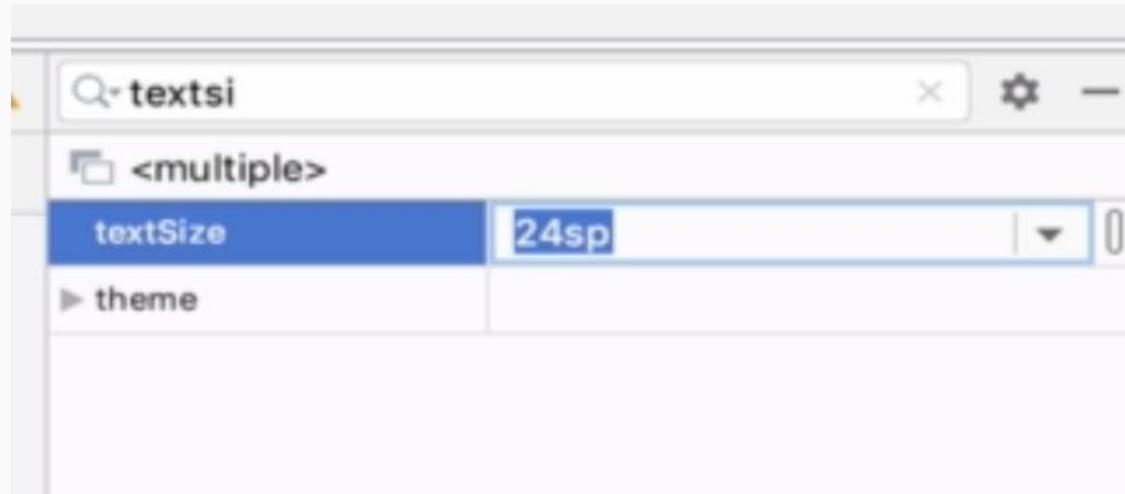
이런 그림 확인



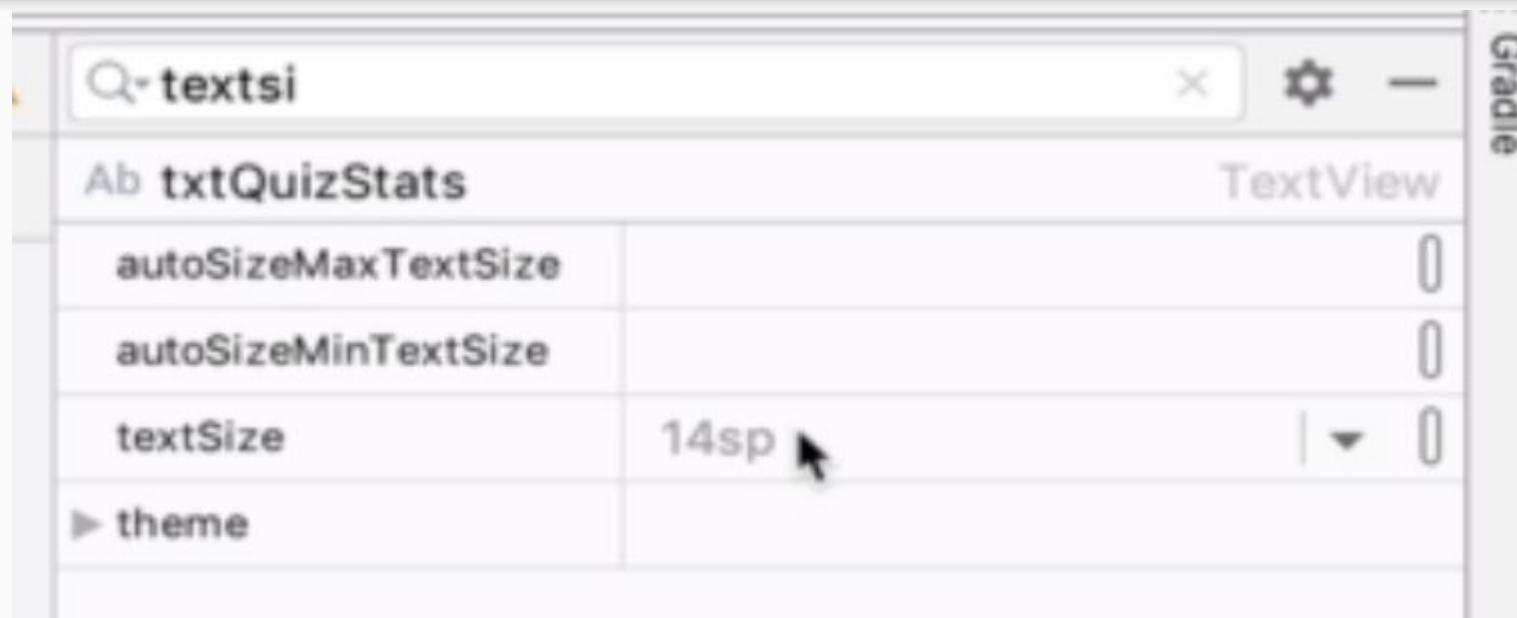
텍스트뷰의 그래비티 모두 이렇게 변환



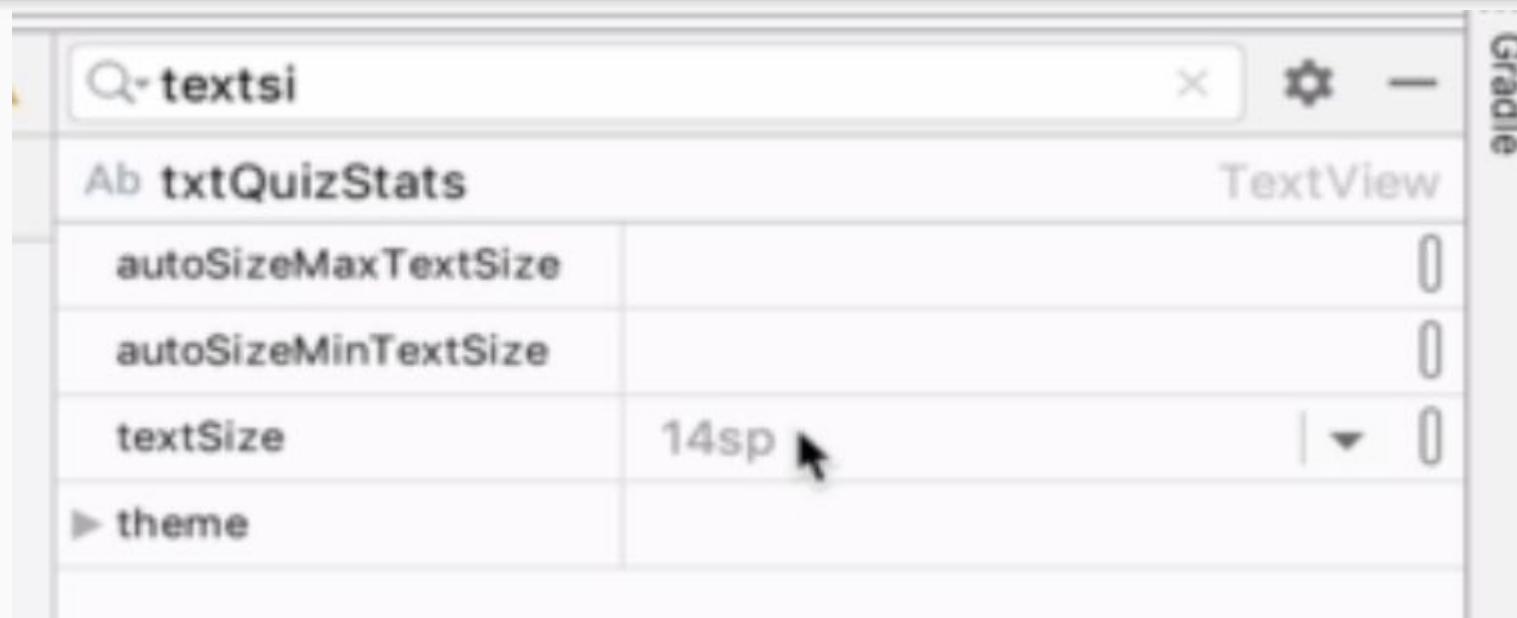
버튼의 텍스트 사이즈를 24sp로 변경.  
글자크기는 sp 사용함



텍스트뷰의 텍스트사이즈도 18로 모두 변경



텍스트뷰의 텍스트사이즈도 18로 모두 변경



# 버튼에 이벤트 발생시키려면? 공식 오피셜 레퍼런스 페이지 가보자

developer.android.com/reference/android/view/View.OnClickListener

The screenshot shows the Android Developers documentation website at developer.android.com. The top navigation bar includes links for 'developers', 'Platform', 'Docs' (which is underlined in green), 'More', a search bar, a language dropdown set to 'LANGUAGE', and a 'SIGN IN' button. Below the navigation is a 'Documentation' section with tabs for 'OVERVIEW', 'GUIDES', 'REFERENCE' (underlined in green), 'SAMPLES', and 'DESIGN & QUALITY'. A sidebar on the left lists various listener interfaces, with 'View.OnClickListener' highlighted in blue. The main content area displays the 'View.OnClickListener' interface documentation, which includes a breadcrumb trail ('Android Developers > Docs > Reference'), a star rating of five stars, and a table of contents with sections for 'Contents', 'Summary', 'Public methods', and 'onClick'. It also notes that the interface was 'Added in API level 1'. At the bottom, there are links for 'View.OnClickListener' in 'Kotlin' and 'Java'.

레퍼런스에서 버튼의 상속 관계도 확인한다.

## Button

---

```
public class Button  
extends TextView  
  
java.lang.Object  
↳ android.view.View  
    ↳ android.widget.TextView  
        ↳ android.widget.Button  
  
⌄ Known direct subclasses  
    CompoundButton  
  
⌄ Known indirect subclasses  
    CheckBox, RadioButton, Switch, ToggleButton
```

# 리스너 생성. logcat에서 확인

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    Button btnTrue = findViewById(R.id.btnTrue);  
  
    View.OnClickListener myClickListener = new View.OnClickListener() {  
        @Override  
        public void onClick(View view) {  
  
            Log.i(tag: "Listener", msg: "My Button is tapped");  
        }  
    };  
  
    btnTrue.setOnClickListener(myClickListener);  
}
```

목적 : (별로 안 좋은 방법) 버튼 2개 이벤트 처리  
버튼이 두 개일 때, 각각의 버튼마다 이벤트 처리하는  
방법 1

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

View.OnClickListener myClickListener = new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        if (view.getId() == R.id.btnTrue) {
            Log.i("MyApp", "btn True is tapped now!");
        } else if (view.getId() == R.id.btnWrong) {
            Log.i("MyApp", "btn Wrong is tapped now!");

        }
    }
};

btnTrue.setOnClickListener(myClickListener);

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

# 각각의 버튼에 리스너 바로 생성하는 (이너 클래스)

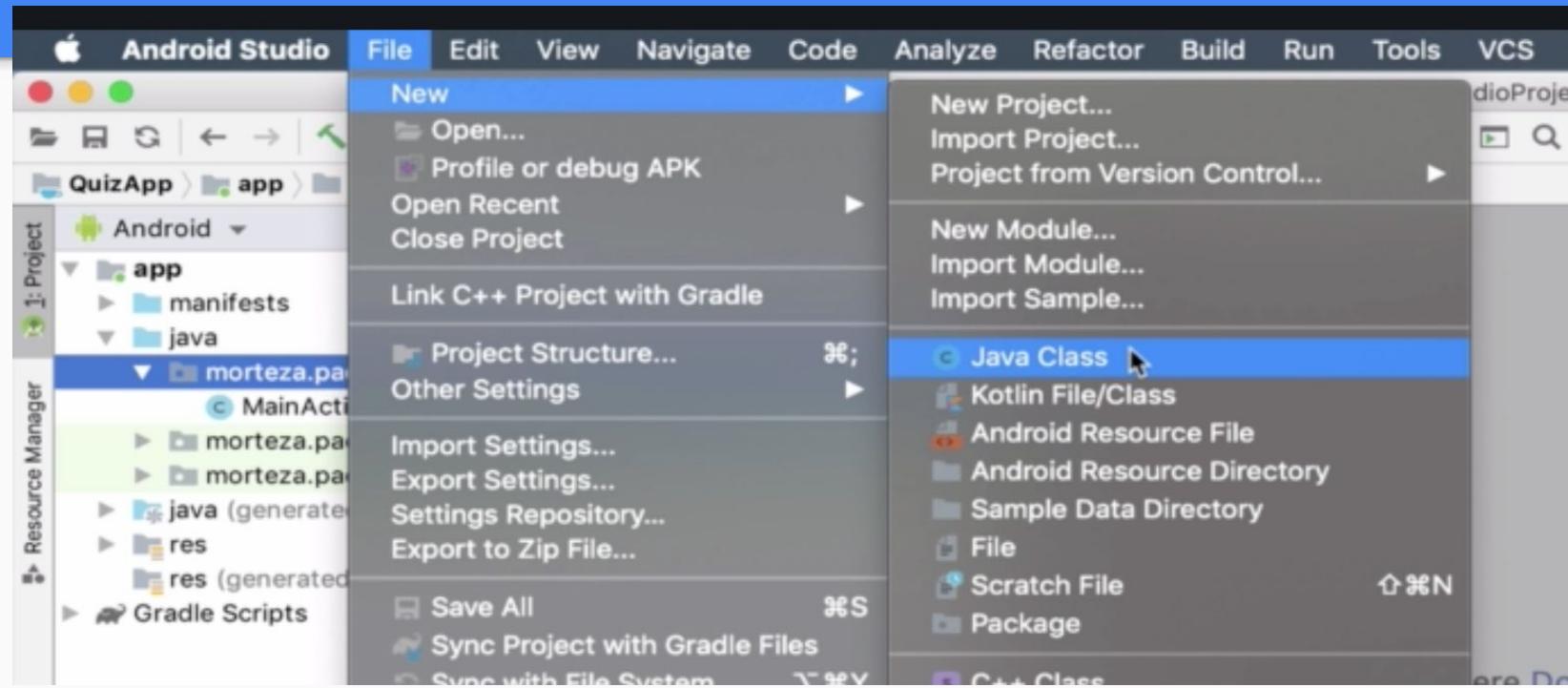
```
btnWrong.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Log.i("tag: " + "MyApp", "msg: " + "btn Wrong is tapped now!");
    }
});
```

목적 : log.i 대신 Toast 를 이용하자. 두개 버튼 전부 바꾼다.

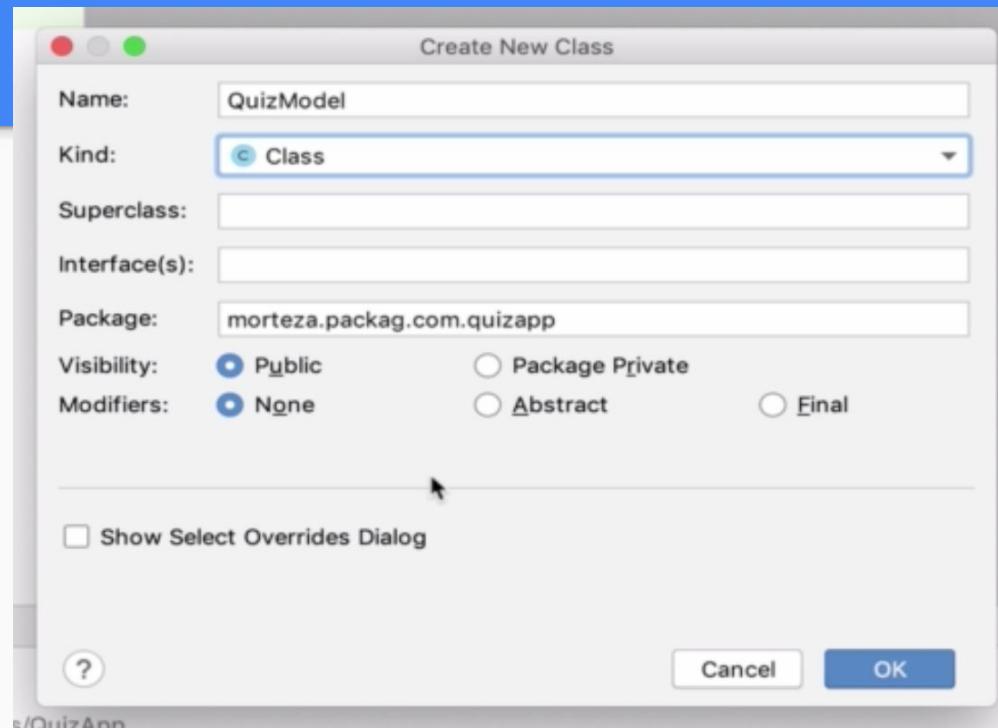
```
@Override  
public void onClick(View view) {  
  
    if (view.getId() == R.id.btnTrue) {  
        // Log.i("MyApp", "btn True is tapped now!");  
  
        Toast myToastObject = Toast.makeText(getApplicationContext(),  
        myToastObject.show();  
  
    }  
    else {  
        Toast myToastObject = Toast.makeText(getApplicationContext(),  
        "btn False is tapped now!", Toast.LENGTH_LONG);  
        myToastObject.show();  
  
    }  
  
    if (view.getId() == R.id.btnWrong) {  
        // Log.i("MyApp", "btn Wrong is tapped now!");  
  
        Toast myToastObject = Toast.makeText(getApplicationContext(),  
        "btn Wrong is tapped now!", Toast.LENGTH_SHORT).show();  
        myToastObject.show();  
  
    }  
}
```

목적 : 퀴즈를 처리할 클래스 만든다.

왼쪽의 패키지명 - 파일 - 뉴 - 자바클래스 선택



# QuizModel 클래스 생성



멤버변수와 생성자 만들기.

생성자 자동생성 방법 알려줌 : 마우스 오른버튼,  
generate - constructor - 2개 변수 모두 선택

```
package morteza.packag.com.quizapp;

public class QuizModel {

    private int mQuestion;
    private boolean mAnswer;

    public QuizModel(int question, boolean answer) {
        mQuestion = question;
        mAnswer = answer;
    }

}
```

getter / setter 도 만든다.

```
public int getQuestion() {  
    return mQuestion;  
}  
  
public void setQuestion(int question) {  
    mQuestion = question;  
}  
  
public boolean isAnswer() {  
    return mAnswer;  
}  
  
public void setAnswer(boolean answer) {  
    mAnswer = answer;  
}
```

메인액티비티로 이동해서, 객체 생성!

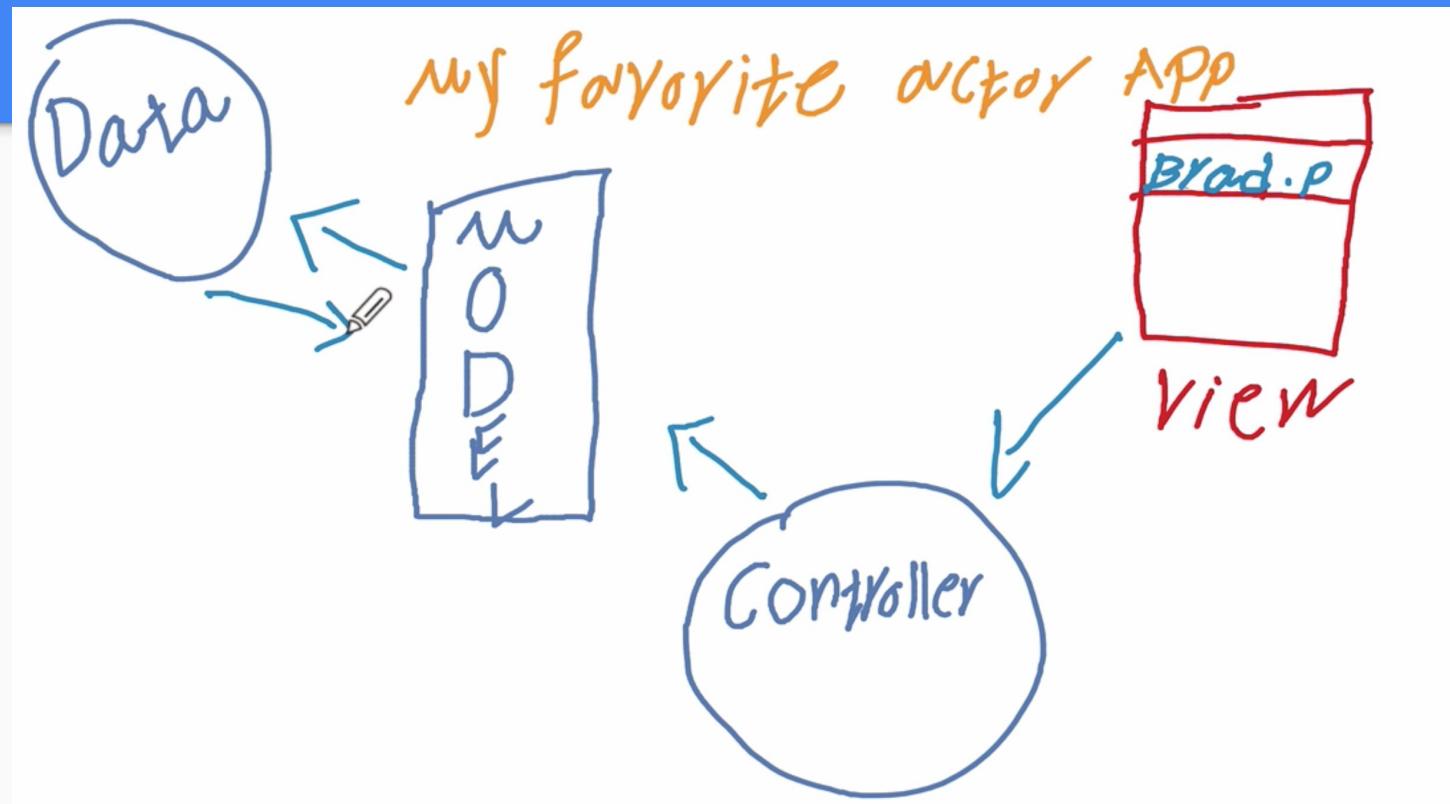
R.string.q1 설명 : strings.xml 파일에 있는것이, 아이디 정수로 바뀐다.

R.string.q1 설명 : strings.xml 파일에 있는것이, 아이디 정수로 바뀐다.

```
QuizModel model = new QuizModel(R.string.q1, answer: true);
```

```
|
```

# MVC 모델이란



# 메인액티비티에 배열로 추가

```
public class MainActivity extends AppCompatActivity {

    private QuizModel[] questionCollection = new QuizModel[] {

        new QuizModel(R.string.q1, answer: true),
        new QuizModel(R.string.q2, answer: false),
        new QuizModel(R.string.q3, answer: true),
        new QuizModel(R.string.q4, answer: false),
        new QuizModel(R.string.q5, answer: true),
        new QuizModel(R.string.q6, answer: false),
        new QuizModel(R.string.q7, answer: true),
        new QuizModel(R.string.q8, answer: false),
        new QuizModel(R.string.q9, answer: true),
        new QuizModel(R.string.q10, answer: false),
    };

}
```

텍스트뷰를 멤버변수로 바꾼다. 그래야 어느  
메소드에서든 값을 가져오고 변경할 수 있다.  
중요!! : 모든 텍스트뷰와 버튼도 다 처리하자.

```
private TextView mTxtQuestion;
```

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    mTxtQuestion = findViewById(R.id.txtQuestion);
```

# TesxView 의 setText 메소드를 이용해서, 글자 변경 가능 오피셜 레퍼런스 확인 먼저 하자.

The screenshot shows the Android Developers website with the search bar set to "setText". The results page displays the TextView class documentation, specifically the setText() methods. The first method listed is `final void setSpannableFactory(Spannable.Factory factory)`, which sets the Factory used to create new Spannable. The second method listed is `final void setText(int resid)`, which sets the text to be displayed using a string resource identifier. This method is highlighted with a yellow background. The third method listed is `final void setText(CharSequence text)`, which sets the text to be displayed. The fourth method listed is `void setText(CharSequence text, TextView.BufferType type)`, which sets the text to be displayed and the TextView.BufferType. The fifth method listed is `final void setText(int resid, TextView.BufferType type)`, which sets the text to be displayed using a string resource. The entire list of methods is shown in a table format with columns for the method signature and its description.

	line input.
<code>final void setSpannableFactory(Spannable.Factory factory)</code>	Sets the Factory used to create new <a href="#">Spannable</a> .
<code>final void setText(int resid)</code>	Sets the text to be displayed using a string resource identifier.
<code>final void setText(CharSequence text)</code>	Sets the text to be displayed.
<code>void setText(CharSequence text, TextView.BufferType type)</code>	Sets the text to be displayed and the <a href="#">TextView.BufferType</a> .
<code>final void setText(int resid, TextView.BufferType type)</code>	Sets the text to be displayed using a string resource.

목적 : setText 테스트 해보자. 진짜 바뀌나.

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    mTxtQuestion = findViewById(R.id.txtQuestion);  
  
    mTxtQuestion.setText("Apple is healthy");
```

멤버변수 하나 추가.

현재 어느 문제를 보여주고 있는지 저장할 변수  
필요하니까.

```
private TextView mTxtQuestion;  
private Button btnTrue;  
private Button btnWrong;  
  
private int mQuestionIndex;
```

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    mTxtQuestion = findViewById(R.id.txtQuestion);  
  
    QuizModel q1 = questionCollection[mQuestionIndex];  
  
    mTxtQuestion.setText(q1.getQuestion());
```

멤버변수 하나 추가.

현재의 퀴즈 문자열을 저장할 변수

```
private int mQuizQuestion;|
```

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    mTxtQuestion = findViewById(R.id.txtQuestion);  
  
    QuizModel q1 = questionCollection[mQuestionIndex];  
  
    mQuizQuestion = q1.getQuestion();  
  
    mTxtQuestion.setText(mQuizQuestion);
```

이제 이 코들은, 버튼 누르면 보여지도록 할 것이므로  
주석처리

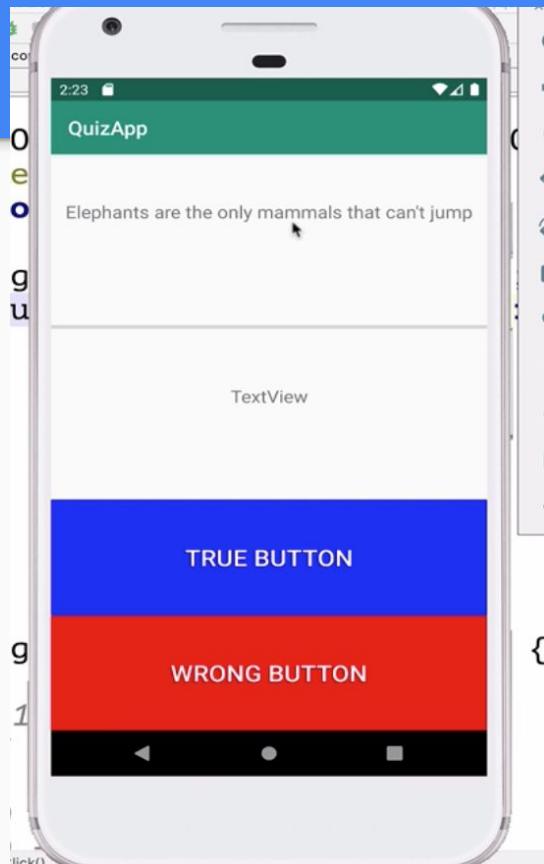
```
QuizModel q1 = questionCollection[mQuestionIndex];  
mQuizQuestion = q1.getQuestion();  
mTxtQuestion.setText(questionCollection[mQuestionIndex].getQuestion());
```

목적 : 문제를 바꿔주는 메소드 설계  
메인 액티비티에 우리가 메소드 하나 추가하여 만든다.  
문제 바꿔주는 코드를, 메소드로 만드는 것이다.

다음문제를 가져오도록 설계하여 개발한다.  
문제 가져오면서, 문자열도 같이 저장

```
private void changeQuestionOnButtonClick() {  
    mQuestionIndex = mQuestionIndex + 1;  
    mQuizQuestion = questionCollection[mQuestionIndex].getQuestion();  
    mTxtQuestion.setText(mQuizQuestion);  
}
```

실행 : 이제 앱 실행해서 토스트 뜨는지 확인하자



# 멤버변수

```
public class MainActivity extends AppCompatActivity {  
  
    private TextView mTxtQuestion;  
    private Button btnTrue;  
    private Button btnWrong;  
  
    private int mQuestionIndex;  
    private int mQuizQuestion;  
  
    private ProgressBar mProgressBar;  
    private TextView mQuizStatsTextView;|
```

I

목적 : 유저의 점수를 저장할 멤버변수 만든다.  
그리고 유저가 맞출때마다 프로그레스바 상태를 위한  
멤버변수 만든다. 10씩 증가하도록 설정.

```
private int mUserScore;
```

```
final int USER_PROGRESS = 10;
```

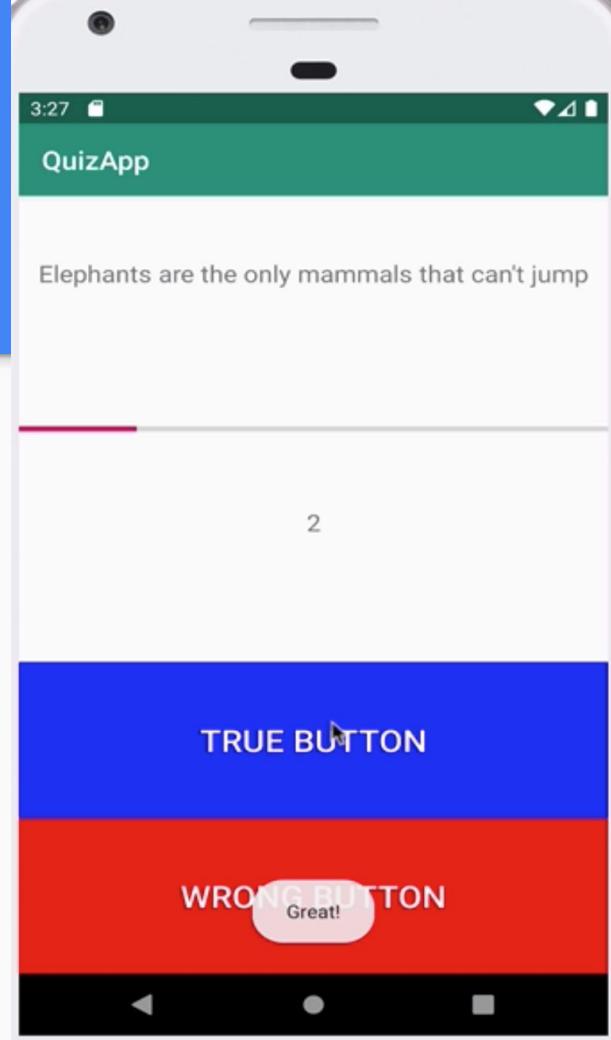
목적 : 점수를 1점씩 증가시키기 위해 이 함수에 점수 추가한다.

```
private void evaluateUsersAnswer(boolean userGuess) {  
    boolean currentQuestionAnswer = questionCollection[mQuestionIndex]  
  
    if (currentQuestionAnswer == userGuess) {  
        Toast.makeText(getApplicationContext(), "Great!", Toast.LENGTH_SHORT)  
        mUserScore = mUserScore + 1;  
    } else {  
        Toast.makeText(getApplicationContext(), "Not good!", Toast.LENGTH_SHORT)  
    }  
}
```

체인지케스쳔 함수에서, 텍스트뷰에 점수 보이도록 맨 아래에 추가 한다.

```
mQuizQuestion = questionCollection[mQuestionIndex].getQuestion();  
  
mTxtQuestion.setText(mQuizQuestion);  
mProgressBar.incrementProgressBy(USER_PROGRESS);  
mQuizStatsTextView.setText(mUserScore + "");
```

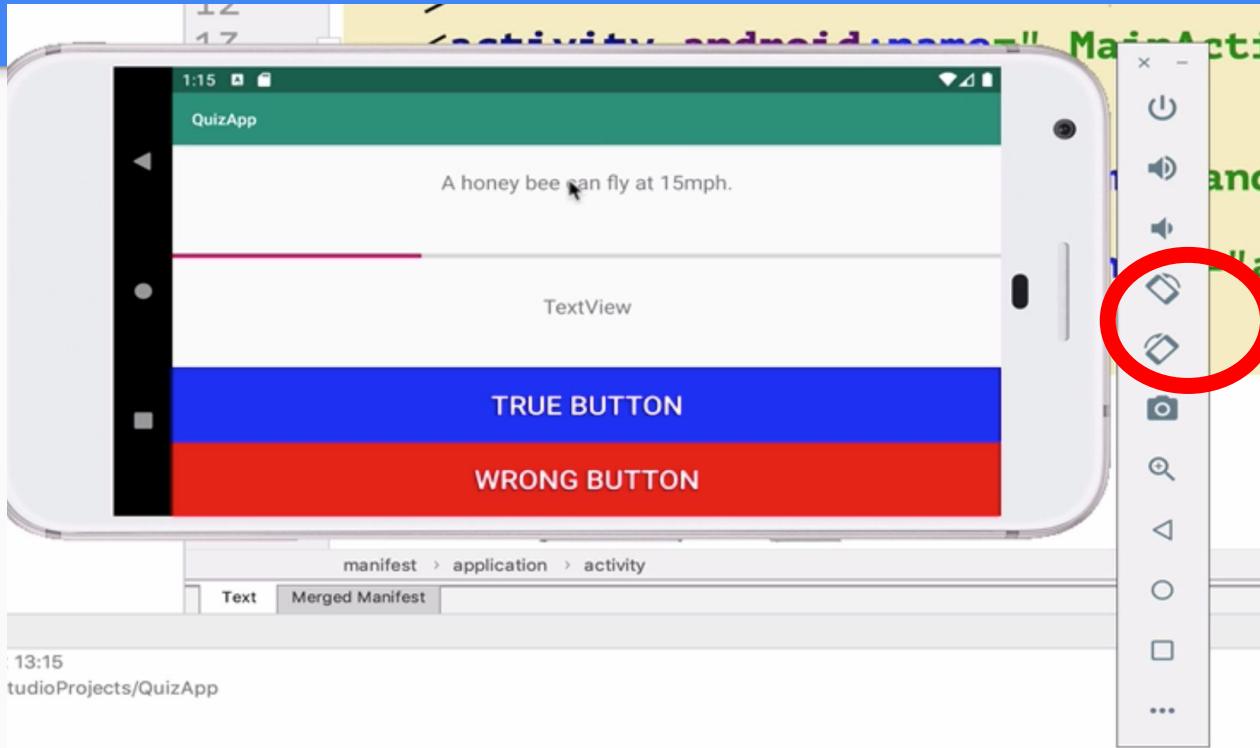
# 앱 실행해보자.



목적 : 질문 갯수에 맞게 프로그래스바 움직이도록  
아래처럼 나눈다. 단, 이문장은 멤버변수 맨 아래쪽에  
작성하자.

```
final int USER_PROGRESS = (int) Math.ceil(100.0 / questionCollection.length);
```

!! 목적 : 안드로이드 옆으로 기울이면 다 지워진다.  
이것을 어떻게 처리할 수 있는가

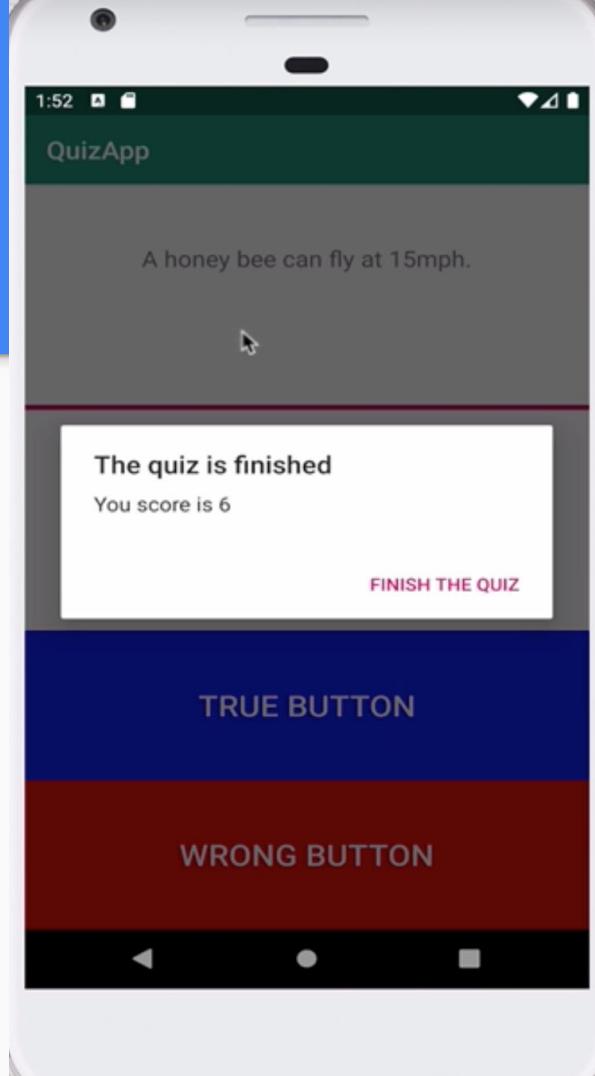


해결 : 앱의 오리엔테이션 고정시키는 법  
안드로이드 메니페스트 엑셈엘 파일로 이동

```
>
<activity android:name=".MainActivity"
    android:screenOrientation="portrait">
    <intent-filter>
        <action android:name="android.intent.action...
```

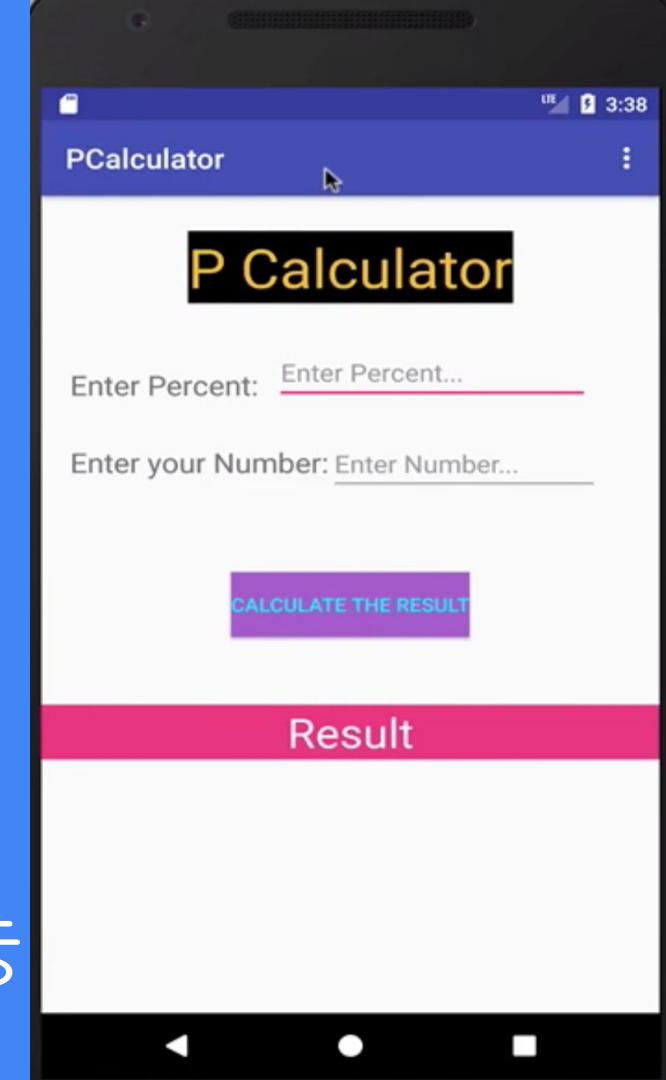
목적 : 앱 끝내는 알럿 만들어 보자.

```
mQuestionIndex = (mQuestionIndex + 1) % 10 ;  
  
if (mQuestionIndex == 0) {  
  
    AlertDialog.Builder quizAlert = new AlertDialog.Builder(context: this)  
    quizAlert.setTitle("The quiz is finished");  
    quizAlert.setMessage("Your score is " + mUserScore);  
    quizAlert.setPositiveButton(text: "Finish the quiz", new DialogInterface  
        @Override  
        public void onClick(DialogInterface dialogInterface, int i) {  
  
            finish();  
  
        }  
    };  
    quizAlert.show();  
}
```



# 실습앱. Calculator

30, 200 입력하면  
 $200 \times 30 / 100 = 60$   
리절트에 나오게 함.



숫자는 소수점도 가능

## 화면 개발 : 릴레이티브 레이아웃으로 변경

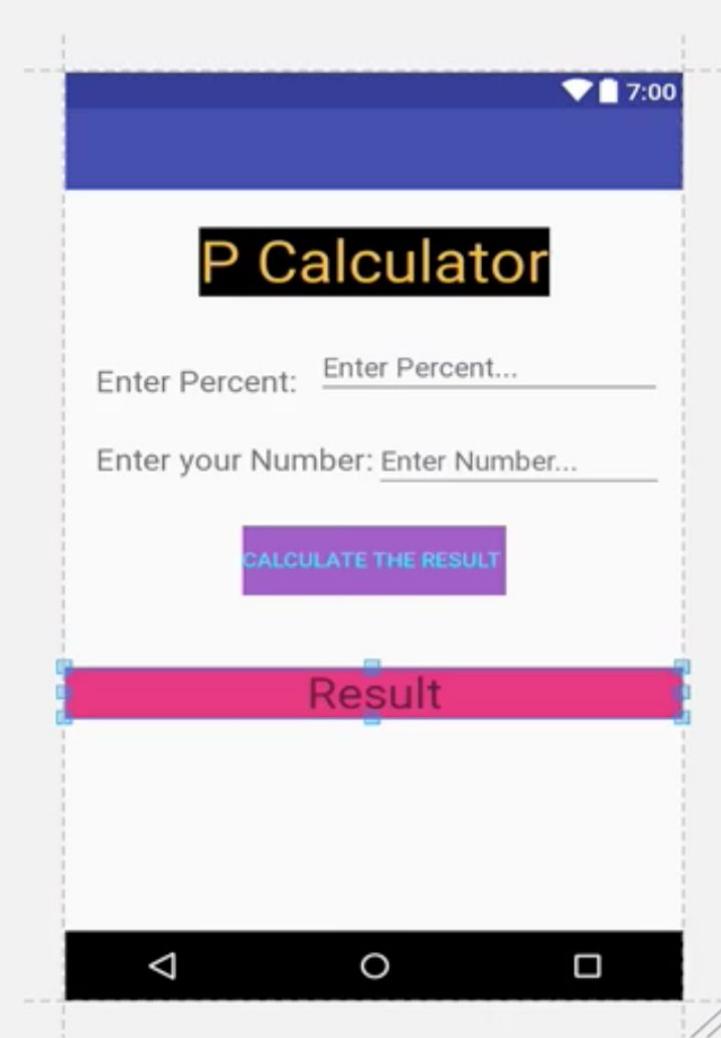
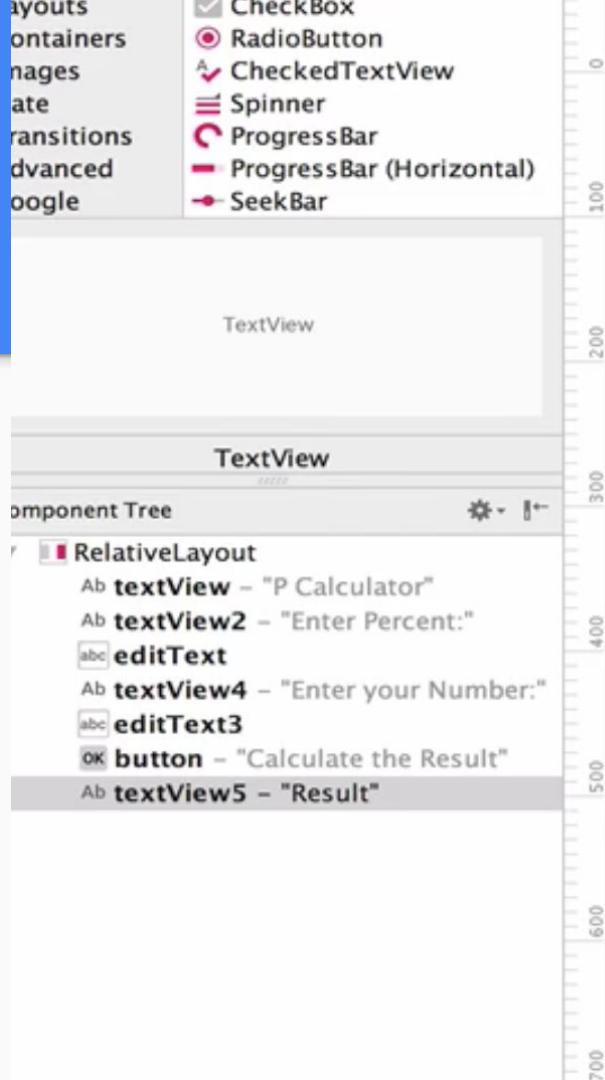
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    app:layout_behavior="android.support.design.widget.AppBarLayout$ScrollingViewBehavior"
    tools:context="com.example.mortezaasadat.pcalculator.MainActivity"
    tools:showIn="@layout/activity_main">

</RelativeLayout>
```

# 화면 개발

텍스트뷰

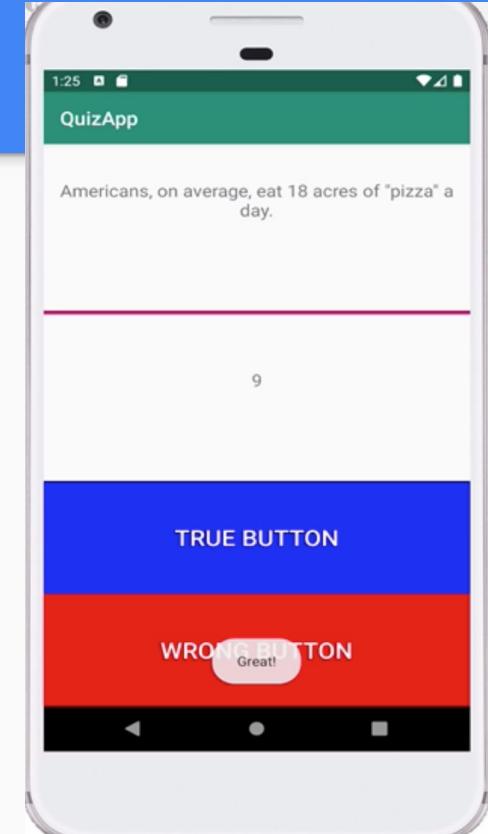
에디트텍스트



목적 : 라이프 사이클을 알아야 하는 이유!

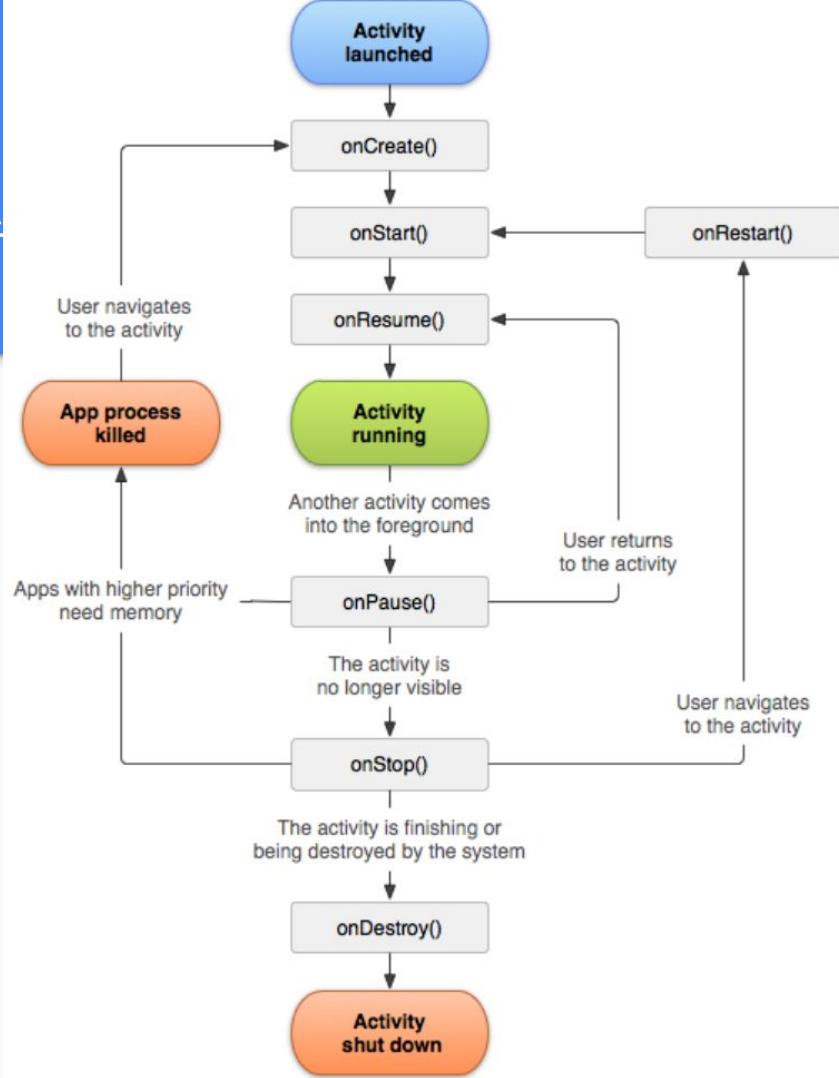
해결책 2: 상태값 저장해서, 돌린 화면에서 다시  
불러오는 방법

텍스트뷰의 9숫자와 상태를 저장하는 법



# 액티비티 라이프 사이클

이그림 설명 + 다음장부터  
메소드 오버라이딩 해서 로그찍



onCreate() 함수부터 록를 다 넣어보자.

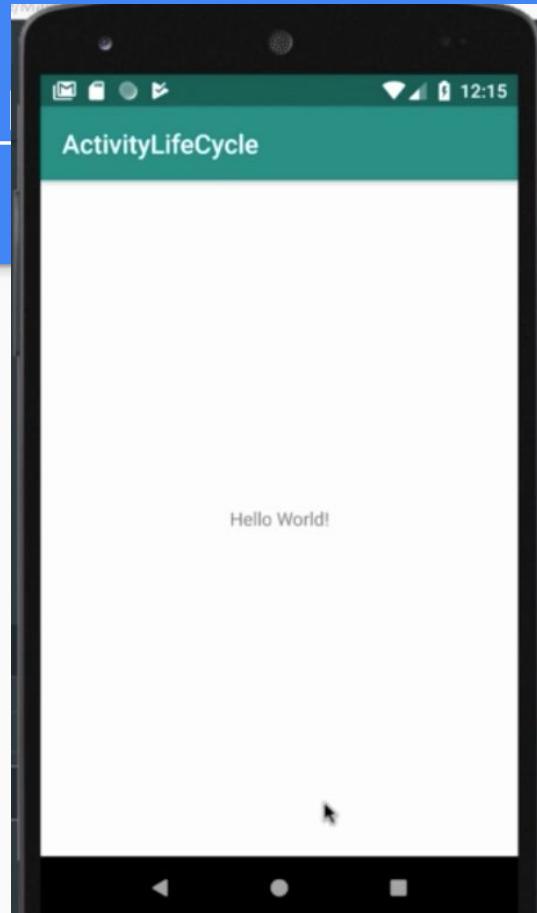
오버라이딩 함수 확인 위해서, 마우스 오른쪽 눌러서  
제너레이션 - 오버라이딩 메뉴 눌러서, 메소드 확인

```
        }  
  
    }  
  
    @Override  
    protected void onStart() {  
        super.onStart();  
    }  
  
    @Override  
    protected void onResume() {  
        super.onResume();  
        Log.d(tag: "Cycle", msg: "onCreate")  
        Toast.makeText(context: MainAct  
    }  
  
    @Override  
    protected void onPause() {  
        super.onPause();  
    }  
  
    @Override  
    protected void onStop() {  
        super.onStop();  
    }  
  
    @Override  
    protected void onDestroy() {  
        super.onDestroy();  
    }  
  
}
```

앞에서 만든 퀴즈앱에서 라이프 사이클용 로그  
넣어보자.

```
        }  
  
    }  
  
    @Override  
    protected void onStart() {  
        super.onStart();  
    }  
  
    @Override  
    protected void onResume() {  
        super.onResume();  
        Log.d(tag: "Cycle", msg: "onCreate")  
        Toast.makeText(context: MainAct.  
    }  
  
    @Override  
    protected void onPause() {  
        super.onPause();  
    }  
  
    @Override  
    protected void onStop() {  
        super.onStop();  
    }  
  
    @Override  
    protected void onDestroy() {  
        super.onDestroy();  
    }  
}
```

# 에뮬레이터 띄어서, 로그캣 확인

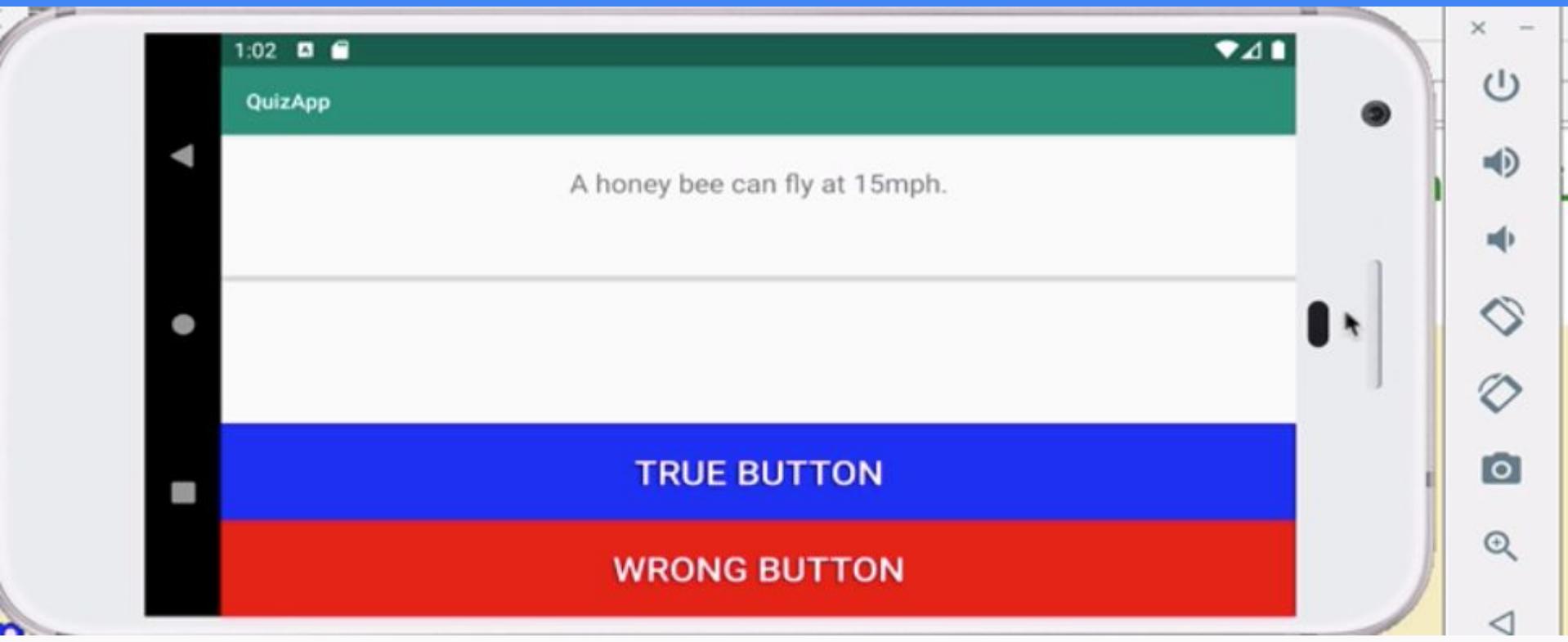


목적 : 라이프사이클 배웠으니, 앱 화면 옆으로 했을때,  
세로 가로화면 전환시 정보 저장 및 다시 실행시키기

아래 삭제하라!

```
    android:theme="@style/AppTheme"
    >
<activity android:name=".MainActivity"
    android:screenOrientation="portrait">
    <intent-filter>
        <action android:name="android.intent.action.
```

화면 돌리면 정보가 사라짐. 즉 라이프사이클 새로 시작하는 것임.



목적 : 정보 살리기 위해 onSaveInstanceState 이용하자.  
이거 이용하면, 정보를 저장했다가, 다시 시작할 때  
넘겨주는거다

키, 밸류 쌍으로 저장함.

```
@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);

    // computer - a robot
    outState.putInt("SCORE", mUserScore);
}

}
```

저장해야 하는 정보는 더 있다. 다음과 같은 정보도  
저장.

}

```
@Override
protected void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);

    // computer - a robot
    outState.putInt(SCORE_KEY, mUserScore);
    outState.putInt(INDEX_KEY, mQuestionIndex);

}
```

이제, 온크리트에서 저장된 정보 불러오자.  
저장된게 없다면 새로 시작했다는 뜻이다.

```
setContentView(R.layout.activity_main);

if (savedInstanceState != null) {

    mUserScore = savedInstanceState.getInt(SCORE_KEY);
    mQuestionIndex = savedInstanceState.getInt(INDEX_KEY);

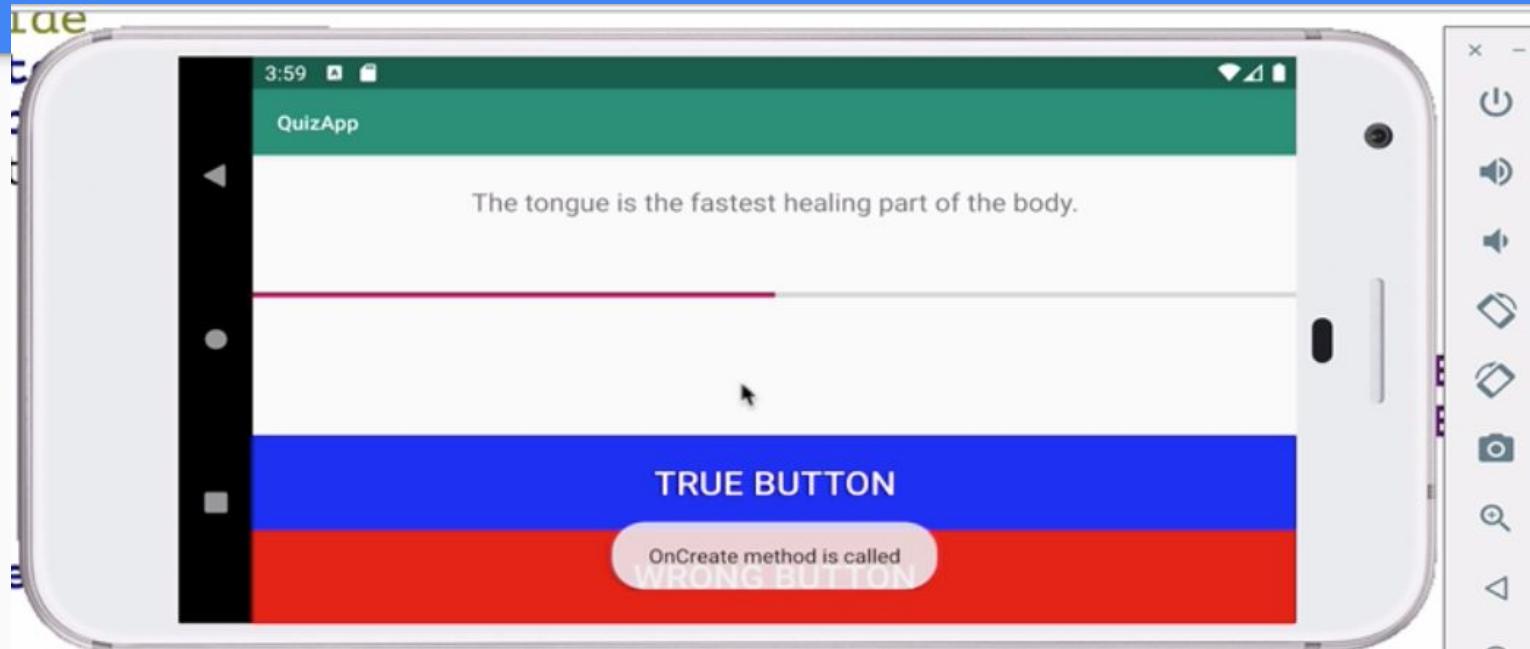
} else {

    mUserScore = 0;
    mQuestionIndex = 0;      }

}

// first lifecycle method
Toast.makeText(getApplicationContext(), text: "OnCreate method
```

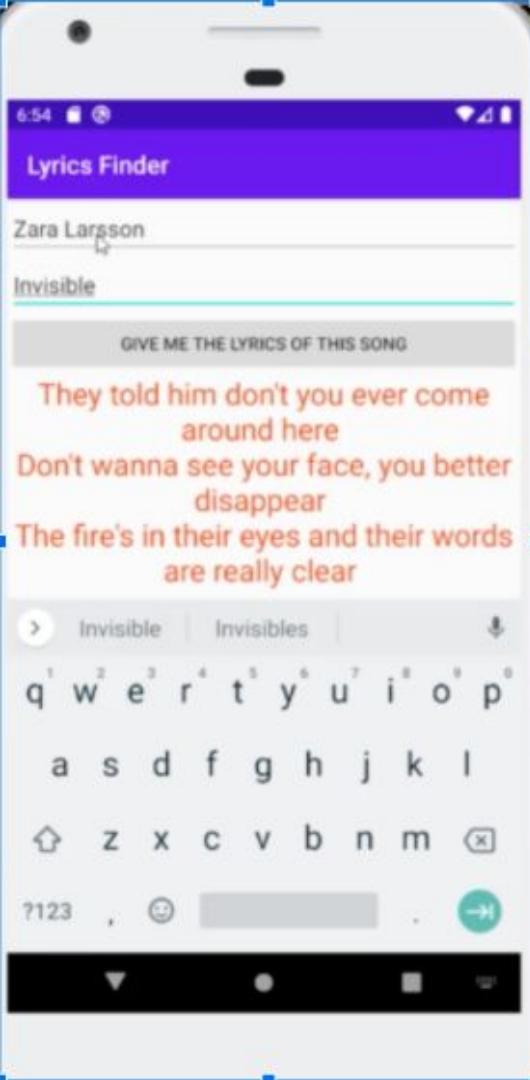
실습 : 앱 실행해본다. 화면에는 반영 안된다. 반영되게 해봐라



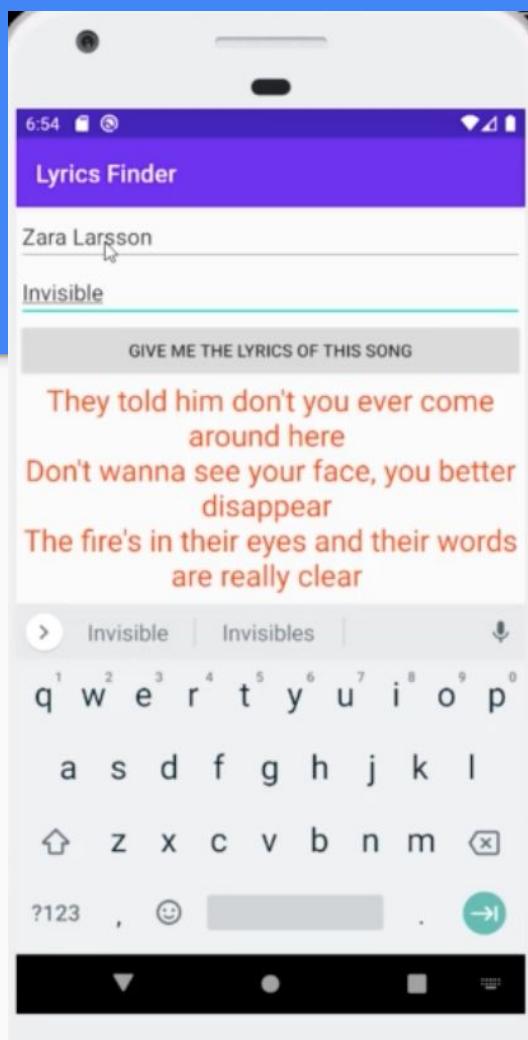
실습 정답 : 텍스트뷰의 findViewById 아래에 넣어준다.

```
mQuizStatsTextView = findViewById(R.id.txtQuizStats);  
mQuizStatsTextView.setText(mUserScore + "");
```

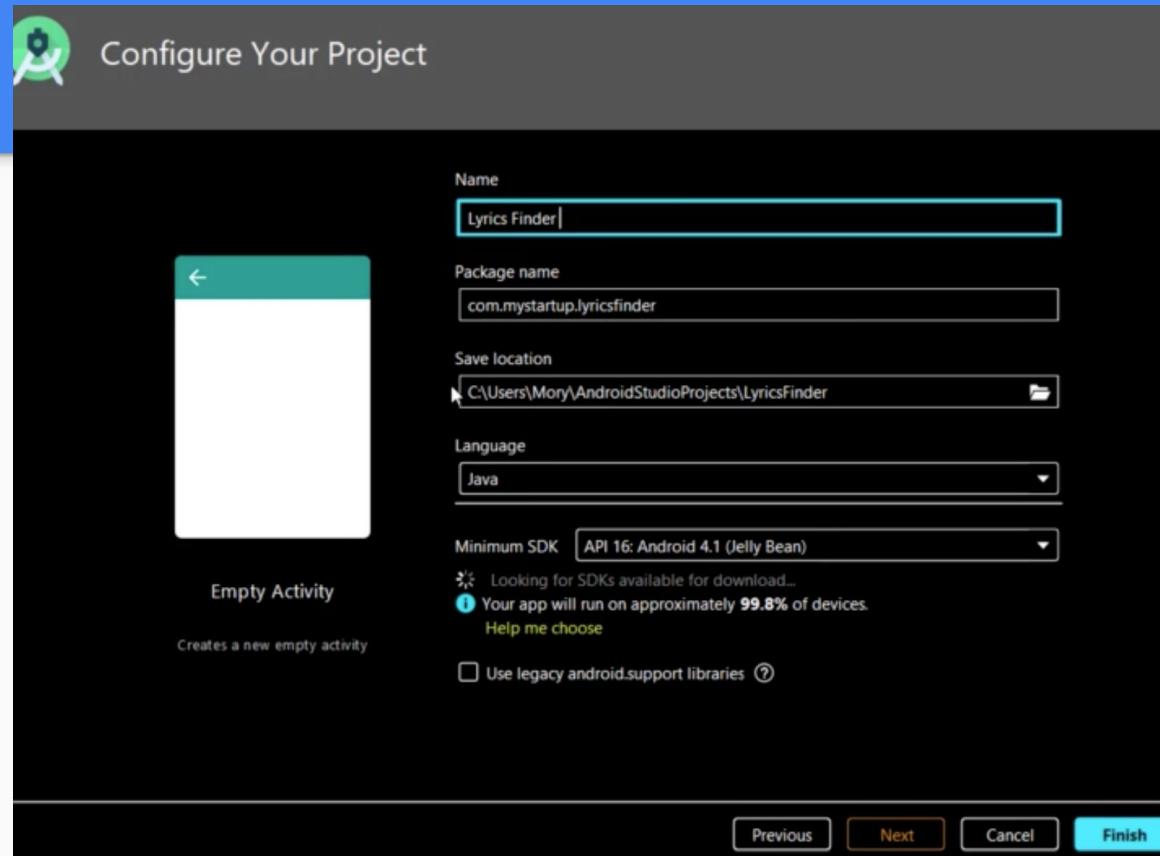
# 가수이름과 노래제목으로 가사 보여주는 앱 개발 (네트워크 통신)



# 완성앱

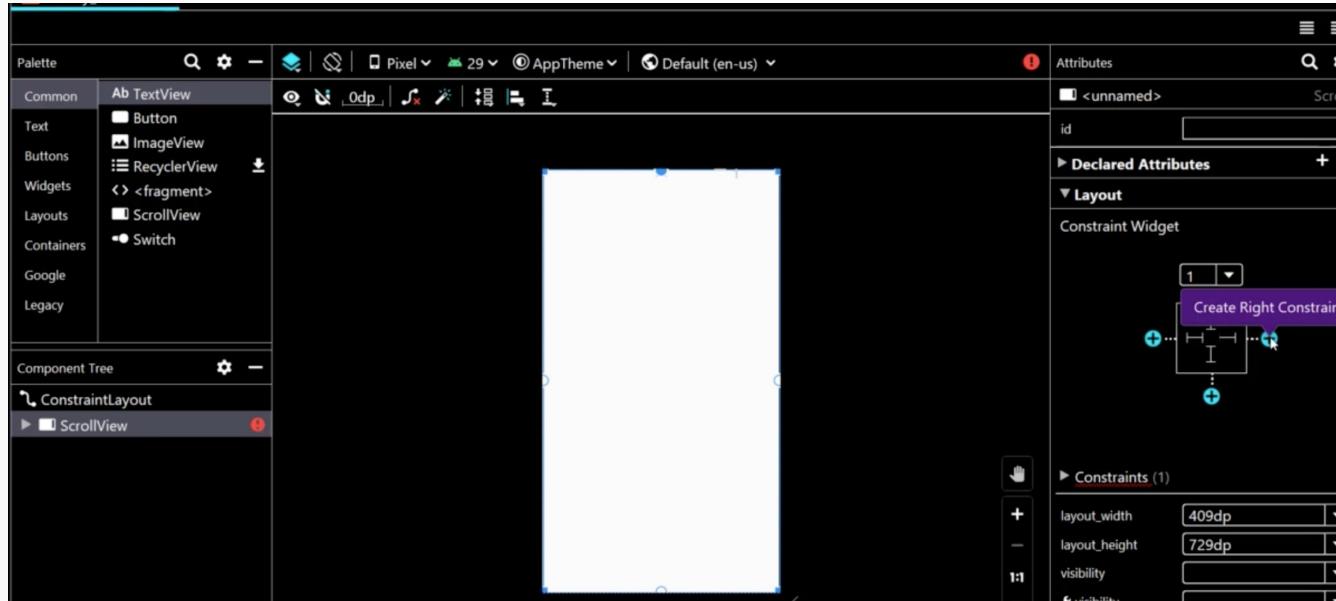


# 새 프로젝트 생성 : 앱 이름 : Lyrics

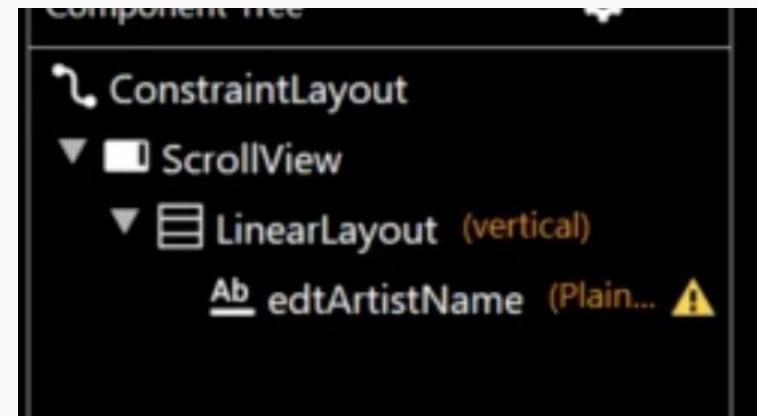
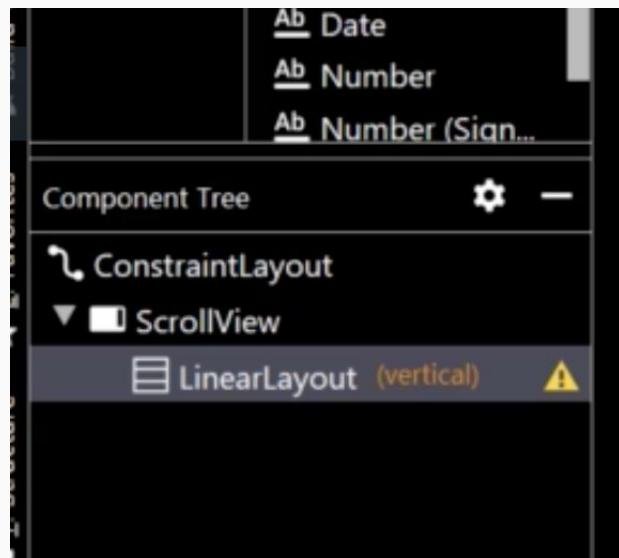


목적 : 컨스트레인 레이아웃 먼저 설명, 이것은 썸씽 투 썸씽엘스 연결시키는 레이아웃임은 설명.

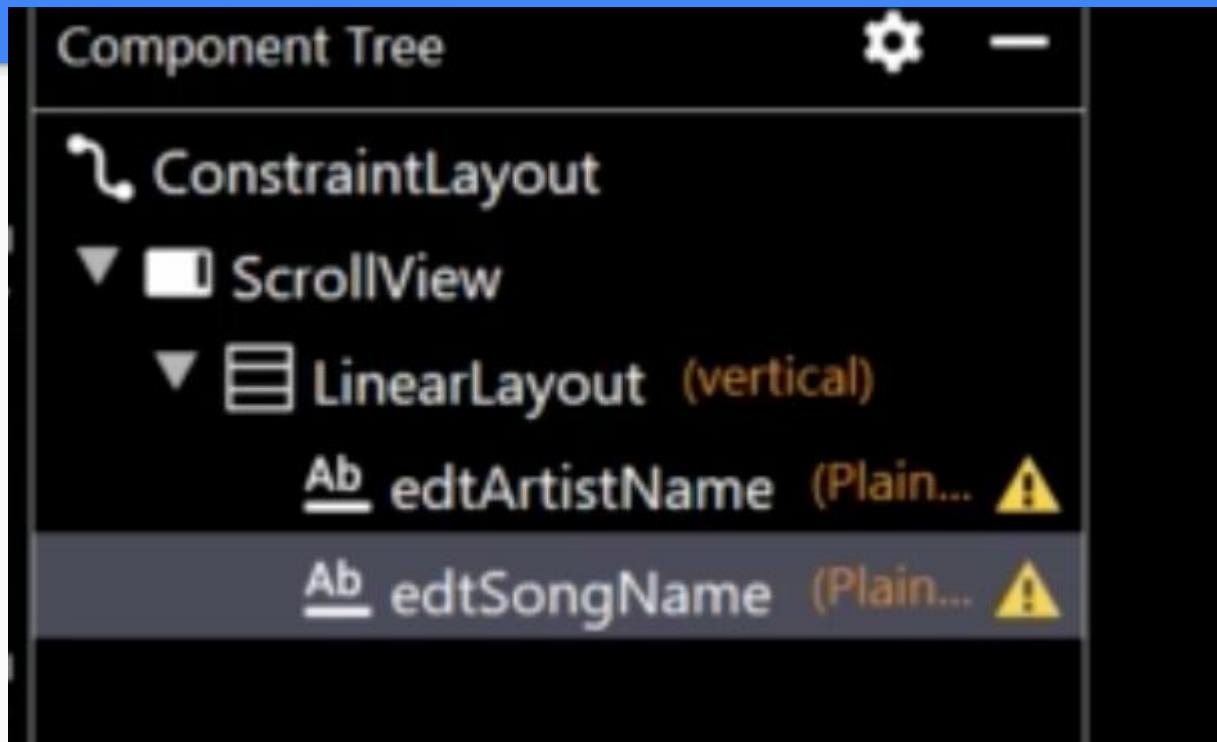
스크롤뷰 하나 가져오는데, 이유는? 가사 기니까 스크롤 하려고 가져온다, 오른쪽 화면에서 각각 4개 플러스 버튼 누른다.



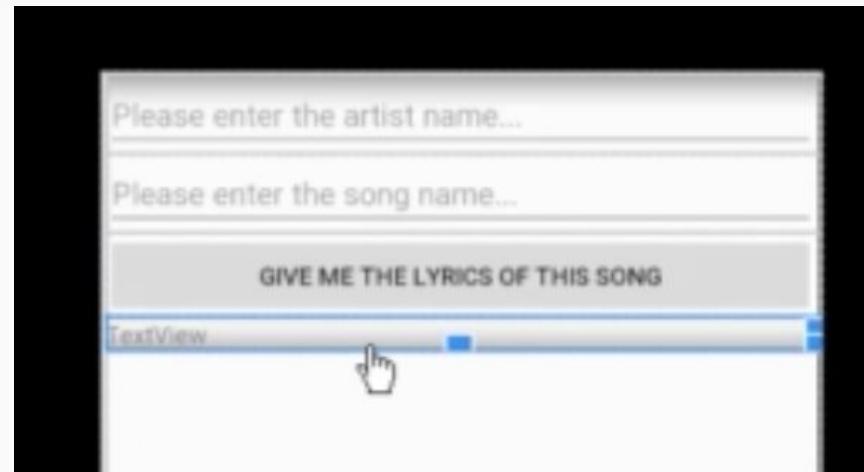
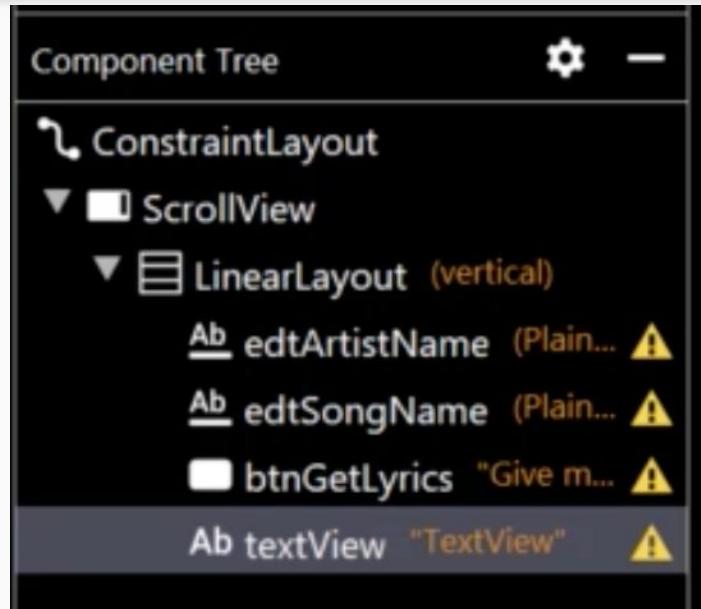
리니어레이아웃을 스크롤뷰 밑에 넣는다.  
리니어 아래에 에디트텍스트 넣고, 힌트와 모양 등을  
맞춘다.



노래 이름 넣으라는 에디트텍스트 만든다.



버튼(텍스트 바꾸고)과 텍스트뷰 넣는다.  
텍스트뷰는 가사 표시할 것이다.  
(텍스트 사이즈, 텍스트 칼라 등 수정, 텍스는 아무것도  
안넣음)



스크롤뷰 너비 높이 0, 0 으로 맞춘다.



목적 : 네트워크 통신을 위한, 빨리 라이브러리 배우자

구글에 검색 : android volley

<https://developer.android.com/training/volley>

중간에 사용법 나온다. 우리 앱에 라이브러리 사용  
추가

```
dependencies {  
    ...  
    implementation 'com.android.volley:volley:1.1.1'  
}
```

## 그레이들에 추가

```
dependencies {  
    implementation fileTree(dir: 'libs', include: ['*.jar'])  
  
    implementation 'androidx.appcompat:appcompat:1.1.0'  
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'  
    testImplementation 'junit:junit:4.12'  
    androidTestImplementation 'androidx.test.ext:junit:1.1.1'  
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.2.0'  
  
    implementation 'com.android.volley:volley:1.1.1'  
}
```

# 이제 자바 코드에 추가하자

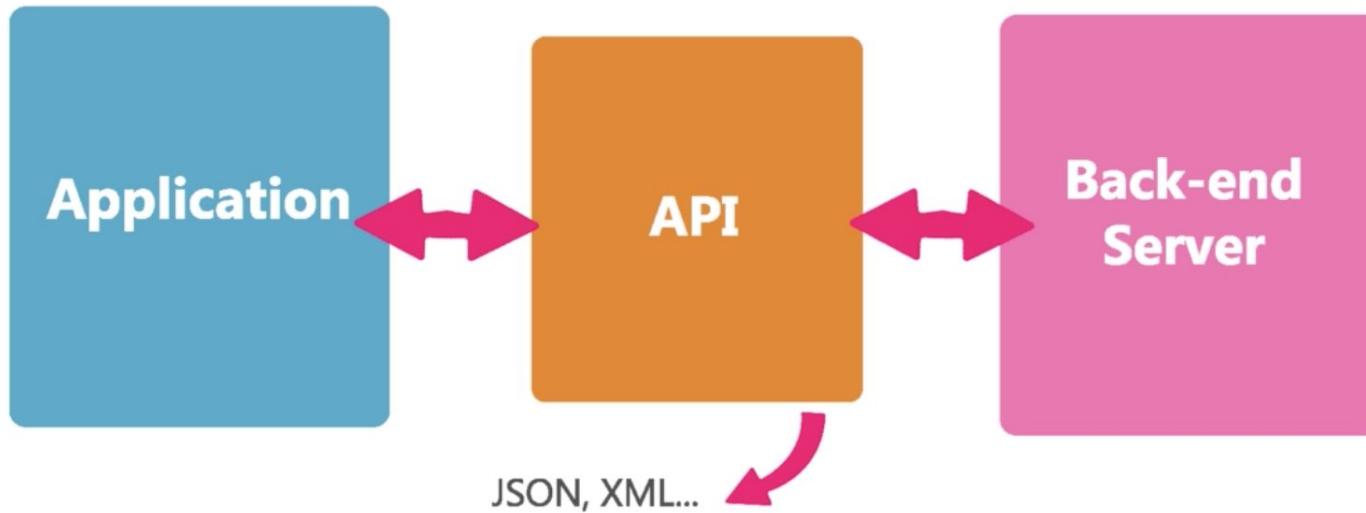
```
public class MainActivity extends AppCompatActivity {

    EditText edtArtistName, edtSongName;
    Button btnGetLyrics;
    TextView txtLyrics;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        edtArtistName = findViewById(R.id.edtArtistName);
        edtSongName = findViewById(R.id.edtSongName);
        btnGetLyrics = findViewById(R.id.btnGetLyrics);
        txtLyrics = findViewById(R.id.txtLyrics);
    }
}
```

# API - Application Programming Interface



목적 : 네트워크에 있는 노래 가사 api 연동

1. 먼저 사이트 가보자 <https://lyrics.ovh/>

The screenshot shows the LYRICS.OVH homepage with a dark background and orange text. The logo 'LYRICS.OVH' is at the top, followed by the tagline 'ONLY THE LYRICS'. Below the logo, there is a search bar containing the text 'Heal the world'. Underneath the search bar, a list of results is displayed, each separated by a horizontal line:

- Heal the World - Michael Jackson
- Heal the World - Spa
- Heal the World - Deep Sleep Music Experience
- Heal the World - Relaxing Music Therapy
- Heal the World - Spa Music Relaxation

목적 : 네트워크에 있는 노래 가사 api 연동

2. api 가 있다. 이것을 이용해서 정보 불러 올 것이다.

Google

lyrics.ovh api

All News Images Videos Maps More Settings Tools

About 158,000,000 results (0.46 seconds)

[lyrics.ovh](#) ▾

**lyrics.ovh: Only the lyrics**

[lyrics.ovh](#). Only the lyrics. Chrome extension · Github API Made by NTag.

[lyricsovh.docs.apiary.io](#) ▾

**lyrics.ovh API - Apiary**

[ovh](#). Introduction. Simple API to retrieve the lyrics of a song. Reference ...

People also search for

- apiary cors
- apiary documentation tutorial
- apiary mock
- musicbrainz api rest
- lyrics com api
- how to use last fm api

목적 : 이용하기 위해선, api 레퍼런스를 확인하는 것이다.

# lyrics.ovh

## INTRODUCTION

Simple API to retrieve the lyrics of a song.

## REFERENCE

### Lyrics of a song

Search >

Switch to Console

### Lyrics of a song / Search

GET <https://api.lyrics.ovh/v1/artist/title>

#### Parameters

**artist** ● Name of the artist Example: `Coldplay`. String

**title** ● Title of the song Example: `Adventure of a Lifetime`. String

#### Request

Production ▾ Raw ▾ Try

결국 api 이용해서 정보 불러 올 것이다.

다음 URL 을 이용하면, 가수이름 / 노래제목으로 조회가 가능하다.

이렇게 가사 가져올 수 있도록, 제공해준다. 이게 API다.

<https://api.lyrics.ovh/v1/Rihanna/Diamonds>

# JSON 설명

<https://en.wikipedia.org/wiki/JSON>

## Example [ edit ]

The following example shows a possible JSON representation describing a person.

```
{  
    "firstName": "John",  
    "lastName": "Smith",  
    "isAlive": true,  
    "age": 27,  
    "address": {  
        "streetAddress": "21 2nd Street",  
        "city": "New York",  
        "state": "NY",  
        "postalCode": "10021-3100"  
    },  
    "phoneNumbers": [  
        {  
            "type": "home",  
            "number": "212 555-1234"  
        },  
        {  
            "type": "office",  
            "number": "646 555-4567"  
        }  
    ],  
    "children": [],  
    "spouse": null  
}
```

많이한다.

난, 에디트텍스트에서 문자열 가져와서 각각 변수에 저장하자.

```
btnGetLyrics.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        // Toast.makeText(getApplicationContext(), "This button is tapped", Toast.LENGTH_SHORT)
        String url = "https://api.lyrics.ovh/v1/" + edtArilstName.getText().toString() + "/" +
        RequestQueue requestQueue = Volley.newRequestQueue(context: MainActivity.this);
        JsonObjectRequest jsonObjectRequest = new JsonObjectRequest(Request.Method.GET, url,
            @Override
            public void onResponse(JSONObject response) {
                txtLyrics.setText(response.toString());
            }
        },
```

url, listener: null, new Response.Listener<...> { com.a

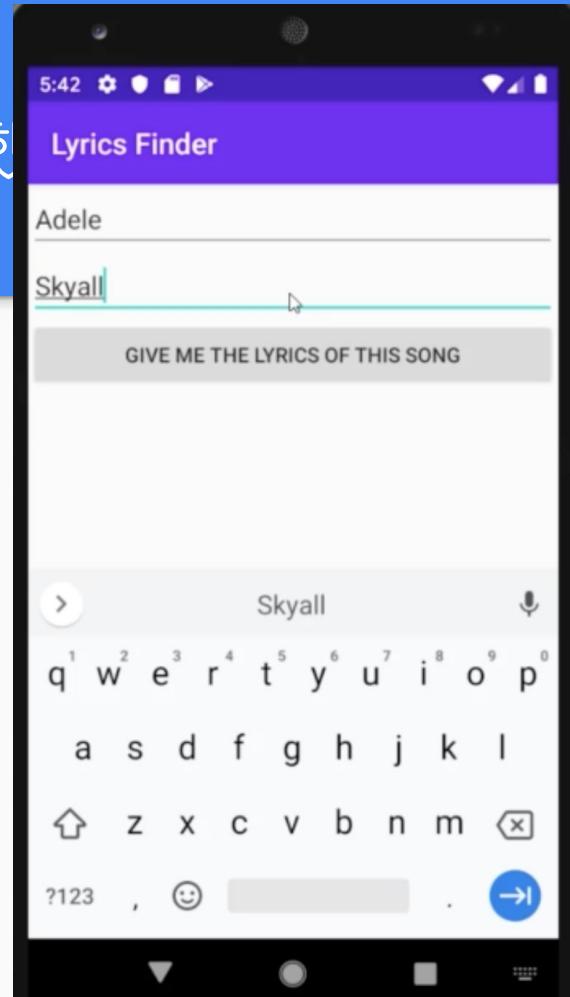
에러리스너도 작성. 로그 찍도록 하자.

그리고 중요, 이제 네트워크로 요청하는 코드를 마지막에 작성

```
        }
    },
    new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            ...
        }
    );
}

requestQueue.add(jsonObjectRequest);
});
```

앱 실행해보라. 안된다. 권한 설정 안호



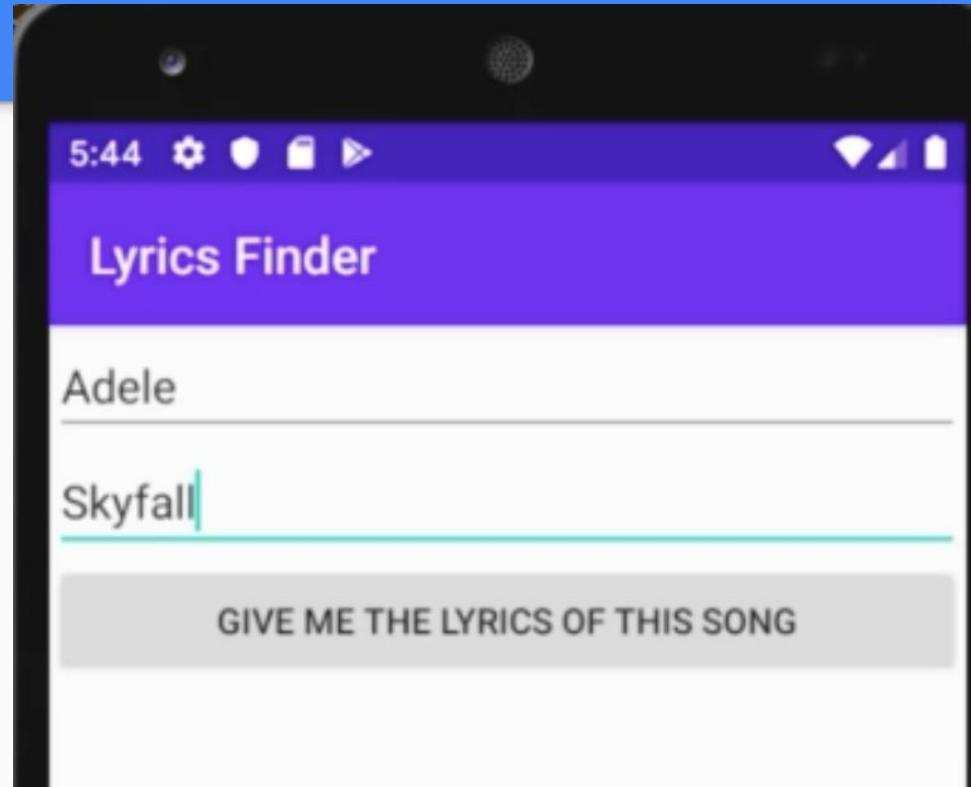
매니페스트에 권한 설정하고, 다음장에서 앱 실행

```
<?xml version='1.0' encoding='UTF-8'?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.mystartup.lyricsfinder">

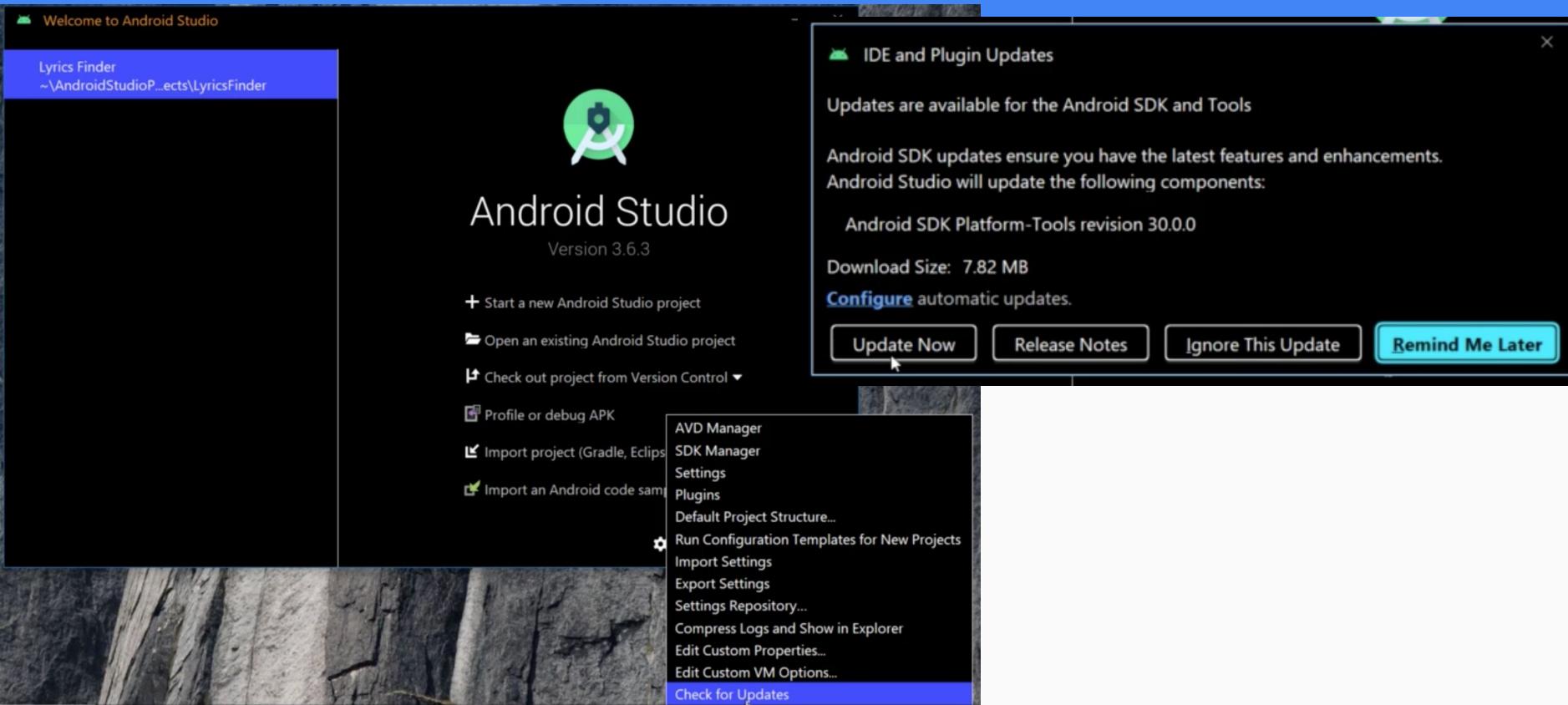
    <uses-permission android:name="android.permission.INTERNET"/>

    <application>
```

앱 실행 : 앱 실행 안되면, 다음장에서 스튜디오  
업데이트 하자



# 프로젝트 닫고, 업데이트 하자



sdk매니저로 이동해서, sdk tools 도 업데이트 하라.  
에뮬레이터도 설치하라.

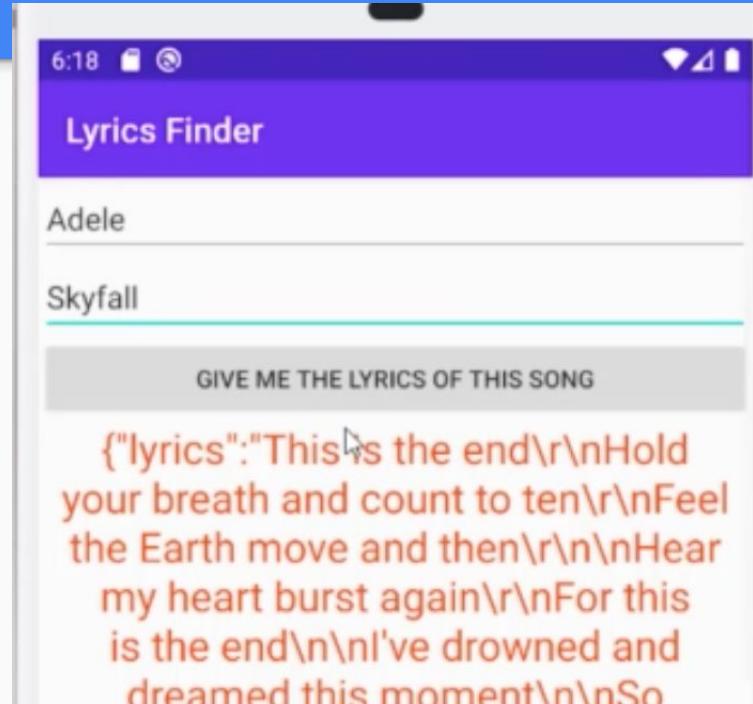
Android SDK Location: C:\Users\Mory\AppData\Local\Android\Sdk      [Edit](#) [Optimize disk space](#)

SDK Platforms    **SDK Tools**    SDK Update Sites

Below are the available SDK developer tools. Once installed, Android Studio will automatically check for updates. Check "show package details" to display available versions of an SDK Tool.

Name	Version	Status
<input checked="" type="checkbox"/> <a href="#">Android SDK Build-Tools 30-rc2</a>		Update Available: 30.0.0 rc2
<input type="checkbox"/> GPU Debugging tools		Not Installed
<input type="checkbox"/> LLDB		Not Installed
<input type="checkbox"/> NDK (Side by side)		Not Installed
<input type="checkbox"/> Android SDK Command-line Tools (latest)		Not Installed
<input type="checkbox"/> CMake		Not Installed
<input type="checkbox"/> Android Auto API Simulators	1	Not installed
<input type="checkbox"/> Android Auto Desktop Head Unit emulator	1.1	Not installed
<input checked="" type="checkbox"/> <a href="#">Android Emulator</a>	30.0.5	Installed
<input type="checkbox"/> <a href="#">Android Emulator Hypervisor Driver for AMD Processors (installer)</a>	1.4.0	Not installed
<input checked="" type="checkbox"/> <a href="#">Android SDK Platform-Tools</a>	30.0.0	Installed
<input checked="" type="checkbox"/> <a href="#">Android SDK Tools</a>	26.1.1	Installed

스크롤뷰가 있어서, 위아래 스크롤된다.  
문제점 : 가사만 나오도록 수정해야 겠다.



다음처럼 제이슨 가져오는것 설명하고,  
익셉션 처리함을 설명한다.

```
txtLyrics.setText(response.getString(name: "lyrics"));
```

Unhandled exception: org.json.JSONException

:

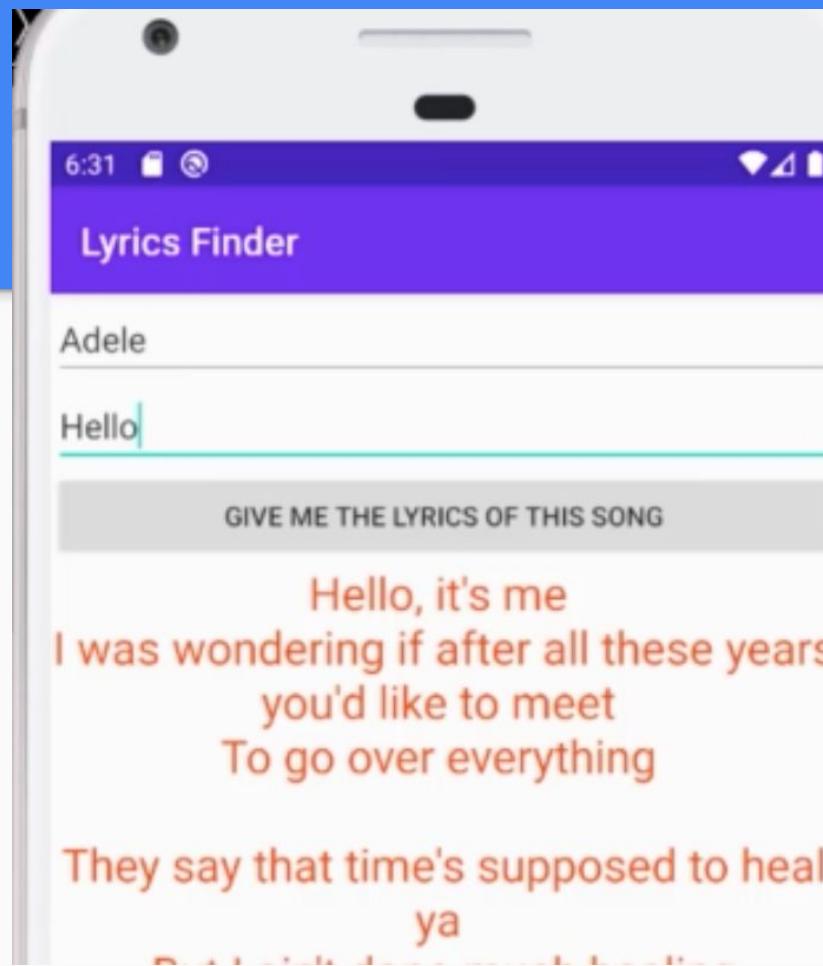
Surround with try/catch Alt+Shift+Enter More actions... Alt+Enter

}

익셉션 처리한다.

```
@Override  
public void onResponse(JSONObject response) {  
  
    try {  
        txtLyrics.setText(response.getString("name: lyrics"));  
  
    } catch (JSONException e) {  
        e.printStackTrace();  
    }  
  
}
```

앱 실행하면 된다.



api 를 조금 더 파보자

# 새 앱 만들기 : 앱 이름 Parsing

# 다음 api 테스트 할 수 있는 사이트 가보자.

The screenshot shows a search results page from a web browser. The search bar at the top contains the query "api test". Below the search bar, a snippet of text reads: "provides endpoints for 'GET', 'POST', 'PUT', various auth ...". The main result is a link to "jsonplaceholder.typicode.com" with the title "JSONPlaceholder - Fake online REST API for developers". A brief description below the title states: "Intro. JSONPlaceholder is a free online REST API that you can use whenever you need some fake data. It's great for tutorials, testing new libraries, ..." A "People also ask" section is visible on the right, listing four questions with dropdown arrows: "What does API testing mean?", "How do I test an API?", "What is API testing with example?", and "Why do we test API?". At the bottom right of the page, there is a "Feedback" link.

api test

provides endpoints for 'GET', 'POST', 'PUT', various auth ...

jsonplaceholder.typicode.com ▾

**JSONPlaceholder - Fake online REST API for developers**

Intro. JSONPlaceholder is a free online REST API that you can use whenever you need some fake data. It's great for tutorials, testing new libraries, ...

People also ask

What does API testing mean? ▾

How do I test an API? ▾

What is API testing with example? ▾

Why do we test API? ▾

Feedback

# 여기 url 카피해서, 메인 액티비티에 다음장처럼 넣는다

## Example

Run this code in a console or from any site:

```
fetch('https://jsonplaceholder.typicode.com/todos/1')
  .then(response => response.json())
  .then(json => console.log(json))
```

Try it

```
{
  "userId": 1,
  "id": 1,
  "title": "delectus aut autem",
  "completed": false
}
```

Congrats you've made your first call to JSONPlaceholder! 😊 🎉

# test url 주석 처리

```
package com.bawp.parsing;

import ...

/> public class MainActivity extends AppCompatActivity {

    //https://jsonplaceholder.typicode.com/todos/1
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

# 사이트 가서 포스트 눌러본다

## Resources

JSONPlaceholder comes with a set of 6 common resources:

/posts	100 posts
/comments	500 comments
/albums	100 albums
/photos	5000 photos
/todos	200 todos
/users	10 users

**Note:** resources have relations. For example: **posts** have many **comments**, **albums** have many **photos**, ... see below for routes examples.

# 데이터는 json 이다.

```
[  
  {  
    "userId": 1,  
    "id": 1,  
    "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",  
    "body": "quia et suscipit\\nsuscipit recusandae consequuntur expedita et cum\\nreprehenderit molestiae ut ut quas \\naturae\\ntem et iure\\nvoluptatem occaecati omnis eligendi aut ad\\nvoluptatem doloribus vel accusantium  
  },  
  {  
    "userId": 1,  
    "id": 2,  
    "title": "qui est esse",  
    "body": "est rerum tempore vitae\\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\\nfugiat blanditii\\ndebitis possimus qui neque nisi nulla"  
  },  
  {  
    "userId": 1,  
    "id": 3,  
    "title": "ea molestias quasi exercitationem repellat qui ipsa sit aut",  
    "body": "et iusto sed quo iure\\nvoluptatem occaecati omnis eligendi aut ad\\nvoluptatem doloribus vel accusantium  
  },  
  {  
    "userId": 1,  
    "id": 4,  
    "title": "eum et est occaecati",  
    "body": "ullam et saepe reiciendis voluptatem adipisci\\nsit amet autem assumenda provident rerum culpa\\nquis hic\\nvolutatem rerum illo velit"  
  },  
]
```

# volley android 검색. 공식 레퍼런스 페이지의 아래쪽에 예제

## Lessons ↗

### [Send a simple request](#)

Learn how to send a simple request using the default behaviors of Volley, and how to cancel a request.

### [Set up RequestQueue](#)

Learn how to set up a `RequestQueue`, and how to implement a singleton pattern to create a `RequestQueue` that lasts the lifetime of your app.

### [Make a standard request](#)

Learn how to send a request using one of Volley's out-of-the-box request types (raw strings, images, and JSON).

### [Implement a custom request](#)

Learn how to implement a custom request.

이제, 이거 요청해서 결과 가져오는거 코드 짜보자.

<https://jsonplaceholder.typicode.com/>

## Example

Run this code in a console or from any site:

```
fetch('https://jsonplaceholder.typicode.com/todos/1')
  .then(response => response.json())
  .then(json => console.log(json))
```

Try it

```
{
  "userId": 1,
  "id": 1,
  "title": "delectus aut autem",
  "completed": false
}
```

메인액티비티, 셋컨텐츠뷰 아래에 다음 코드 작성

```
JsonObjectRequest jsonObjectRequest = new JsonObjectRequest(Request.Method.GET,
    url: "https://jsonplaceholder.typicode.com/todos/1", jsonRequest: null,
    new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {
            Log.d(tag: "JSON:", msg: "onResponse: " + response);
        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
        }
    });
    
```

## 메인 액티비티에 멤버변수

```
//https://jsonplaceholder.typicode.com/todos/1  
private RequestQueue requestQueue;
```

## 온크리트에 추가

```
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
requestQueue = Volley.newRequestQueue(context: this);
```

맨 아래에 요청하도록 코드 작성

```
});  
requestQueue.add(jsonObjectRequest);  
}
```

앱 실행하여 로그캣 보자  
안된다. 메니페스트에 인터넷 권한 추가하자.

```
💡 <uses-permission android:name="android.permission.INTERNET" />
```

아래 응답에서, title 정보 가져오는거 해보자.

<https://jsonplaceholder.typicode.com/>

## Example

Run this code in a console or from any site:

```
fetch('https://jsonplaceholder.typicode.com/todos/1')
  .then(response => response.json())
  .then(json => console.log(json))
```

Try it

```
{
  "userId": 1,
  "id": 1,
  "title": "delectus aut autem",
  "completed": false
}
```

이제 파싱해보자. 다음 코드 추가

```
    ...
}

@Override
public void onResponse(JSONObject response) {
    try {
        Log.d( tag: "JSON:", msg: "onResponse: " + response.getString( name: "title"));
    } catch (JSONException e) {
        e.printStackTrace();
    }
}
```