

>1번

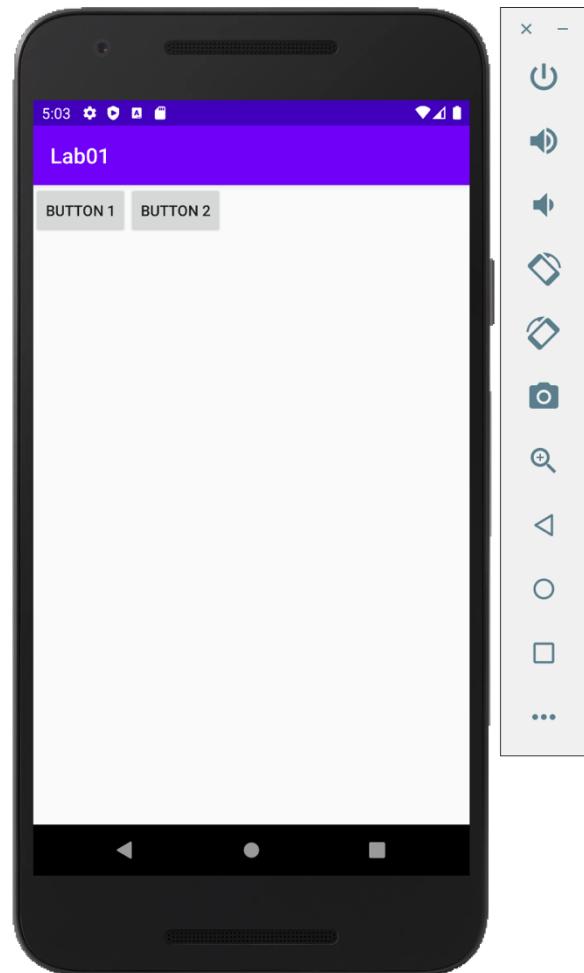
- 안드로이드 창 MainActivity.java

The screenshot shows the Android Studio interface with the project 'InterfaceCourse' open. The 'MainActivity.java' file is selected in the top navigation bar. The code editor displays the following Java code:

```
1 package com.example.lab01;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import android.os.Bundle;
5 import android.widget.Button;
6 import android.widget.LinearLayout;
7
8 public class MainActivity extends AppCompatActivity {
9
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13
14        LinearLayout linear = new LinearLayout( context: this );
15
16        Button bt=new Button( context: this );
17        bt.setText("Button 1");
18        linear.addView(bt);
19
20        Button bt2=new Button( context: this );
21        bt2.setText("Button 2");
22        linear.addView(bt2);
23
24        setContentView(linear);
25    }
26
27 }
```

The code creates a new `LinearLayout` and adds two `Button` instances to it. The buttons are labeled "Button 1" and "Button 2". The `setContentView` method is used to set the `linear` layout as the content view of the activity.

- AVD 실행 화면



- 학습한 내용

activity들은 이름 때문에 activity로 여겨지는 것이 아니라 AppCompatActivity를 상속하는 클래스이기 때문에 activity로 여겨진다.

onCreate는 Activity를 생성하면 자동적으로 생성되는 메소드로 Activity가 처음 실행될 때 제일 먼저 호출되는 메소드로 각종 초기화 작업을 해준다.

LinearLayout은 ViewGroup이다.

View의 배치를 담당하는 것을 ViewGroup이라고 한다.

그 중, LinearLayout은 수평, 수직으로 배치하는 기능을 가졌다.

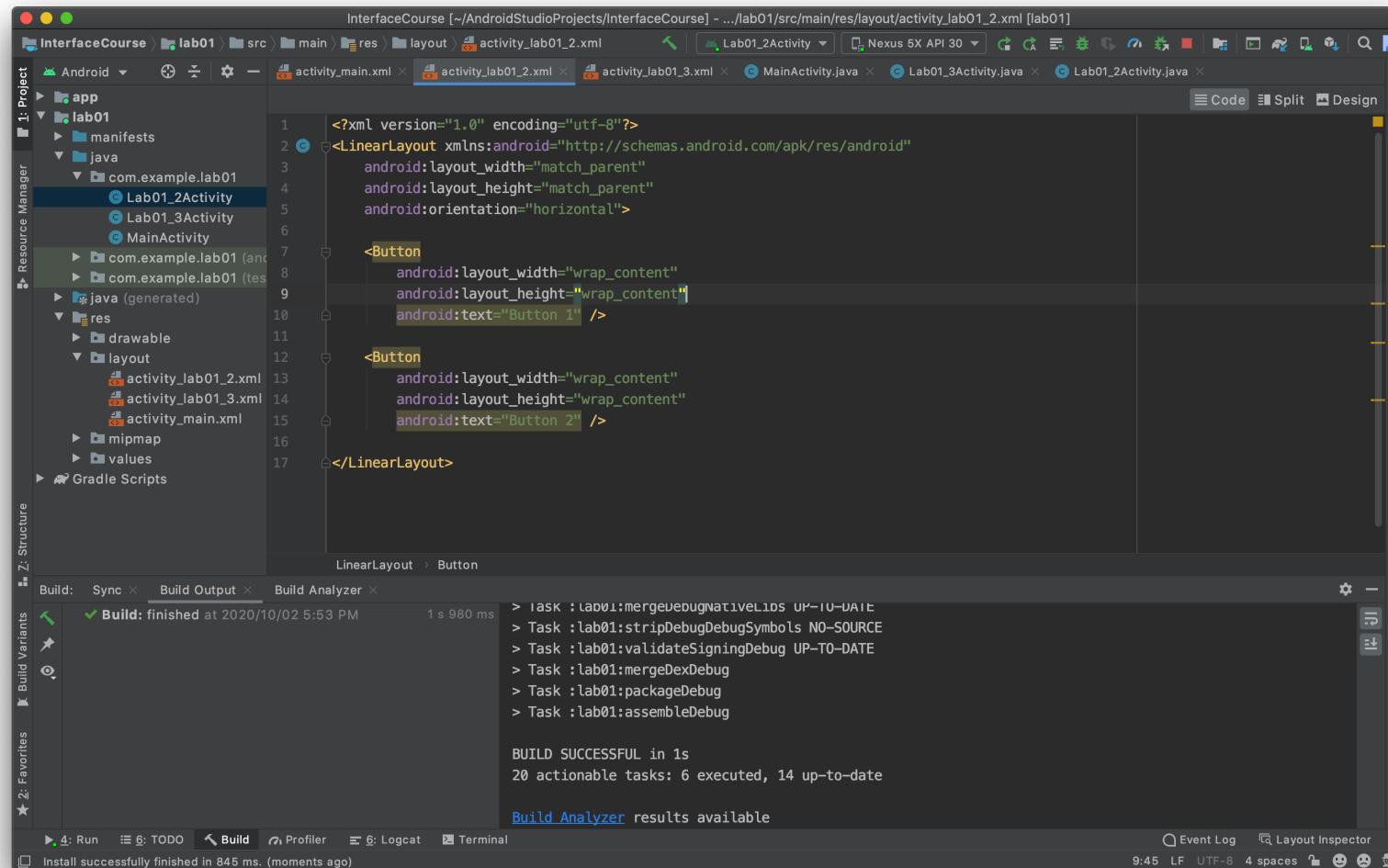
Button은 View를 상속하는 클래스이다.

생성한 button을 linear에 추가해준다.

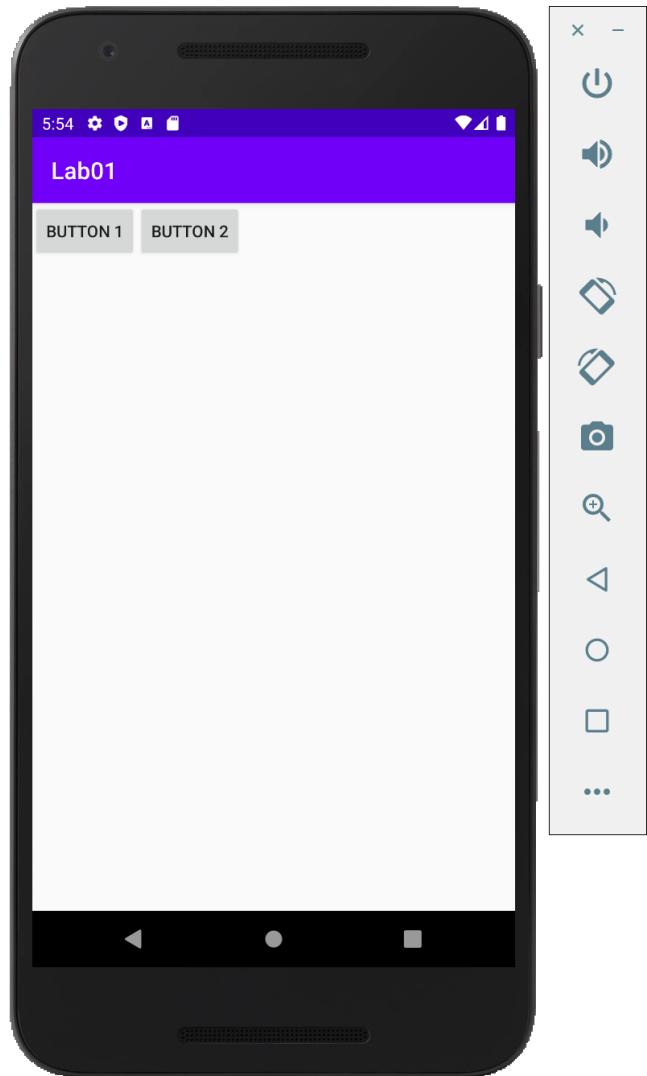
setContentView 메소드를 활용해 제작한 linear를 화면에 출력해준다.

>2번

- 안드로이드 창 activity_lab01_2.xml



- AVD 실행 화면



- 학습한 내용

java 코드로 화면을 구성할 수도 있고, xml로도 화면을 구성할 수 있다.

xml은 태그로 구성된 언어이다.

'android'는 네임스페이스를 선언한 것이다.

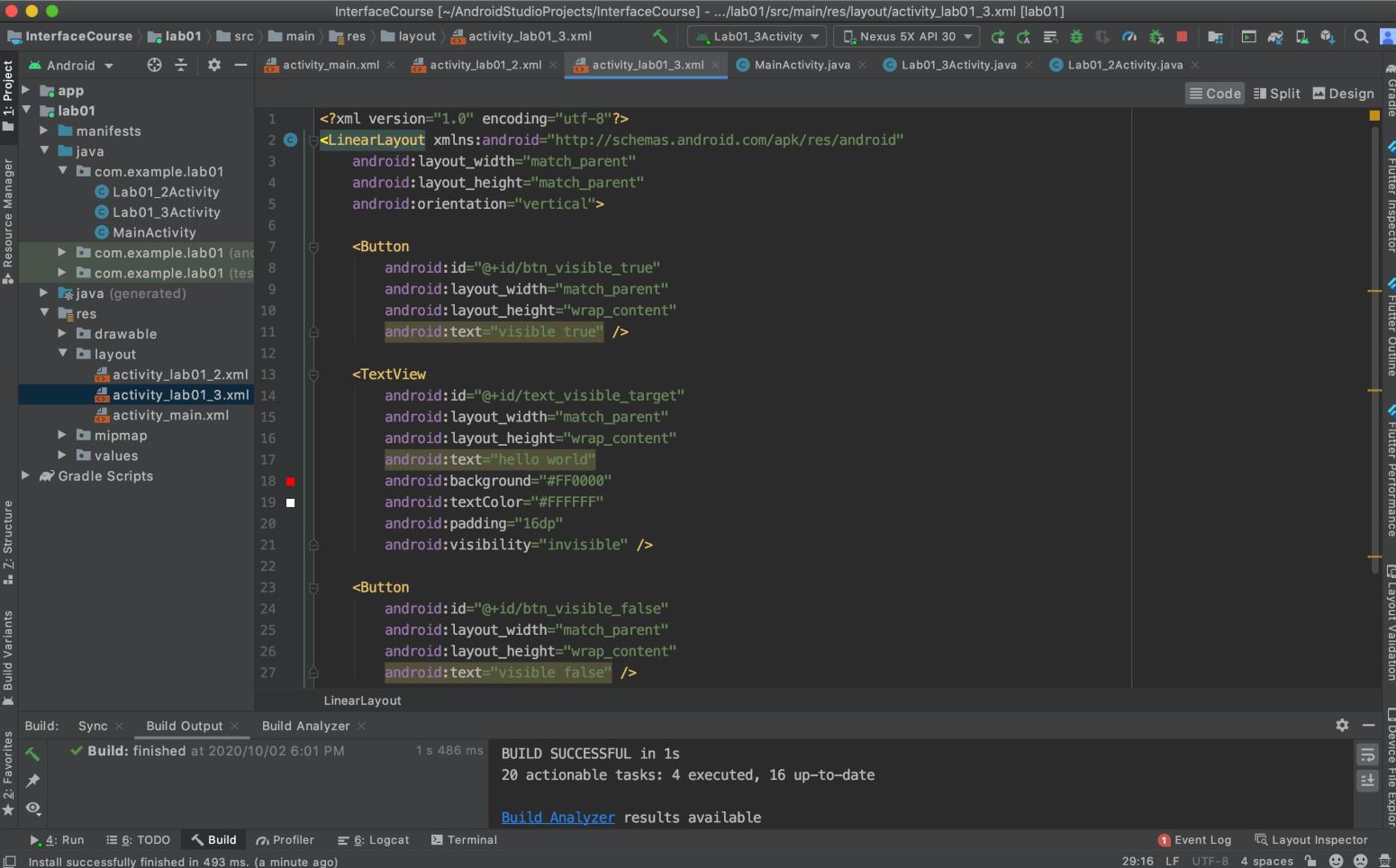
시작 태그가 <aaa>이면, 종료 태그는 </aaa>이다.

시작 태그와 종료 태그 중간에 있는 요소를 element라고 지칭한다.
element가 없을 수도 있다. Button이 그 예시이다.

match_parent는 상위 view의 크기에 맞는 사이즈를 뜻하고,
wrap_parent는 해당 view를 보여줄 수 있는 크기를 뜻한다.
orientation은 수평, 수직을 선택할 수 있다.

〉 3번

- 안드로이드 창 activity_lab01_3.xml



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

    <Button
        android:id="@+id	btn_visible_true"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="visible true" />

    <TextView
        android:id="@+id/text_visible_target"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="hello world"
        android:background="#FF0000"
        android:textColor="#FFFFFF"
        android:padding="16dp"
        android:visibility="invisible" />

    <Button
        android:id="@+id	btn_visible_false"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="visible false" />

```

LinearLayout

Build: Sync Build Output Build Analyzer

Build: finished at 2020/10/02 6:01 PM 1 s 486 ms BUILD SUCCESSFUL in 1s
20 actionable tasks: 4 executed, 16 up-to-date

Build Analyzer results available

4: Run 6: TODO 8: Build 10: Profiler 6: Logcat Terminal

Event Log Layout Inspector

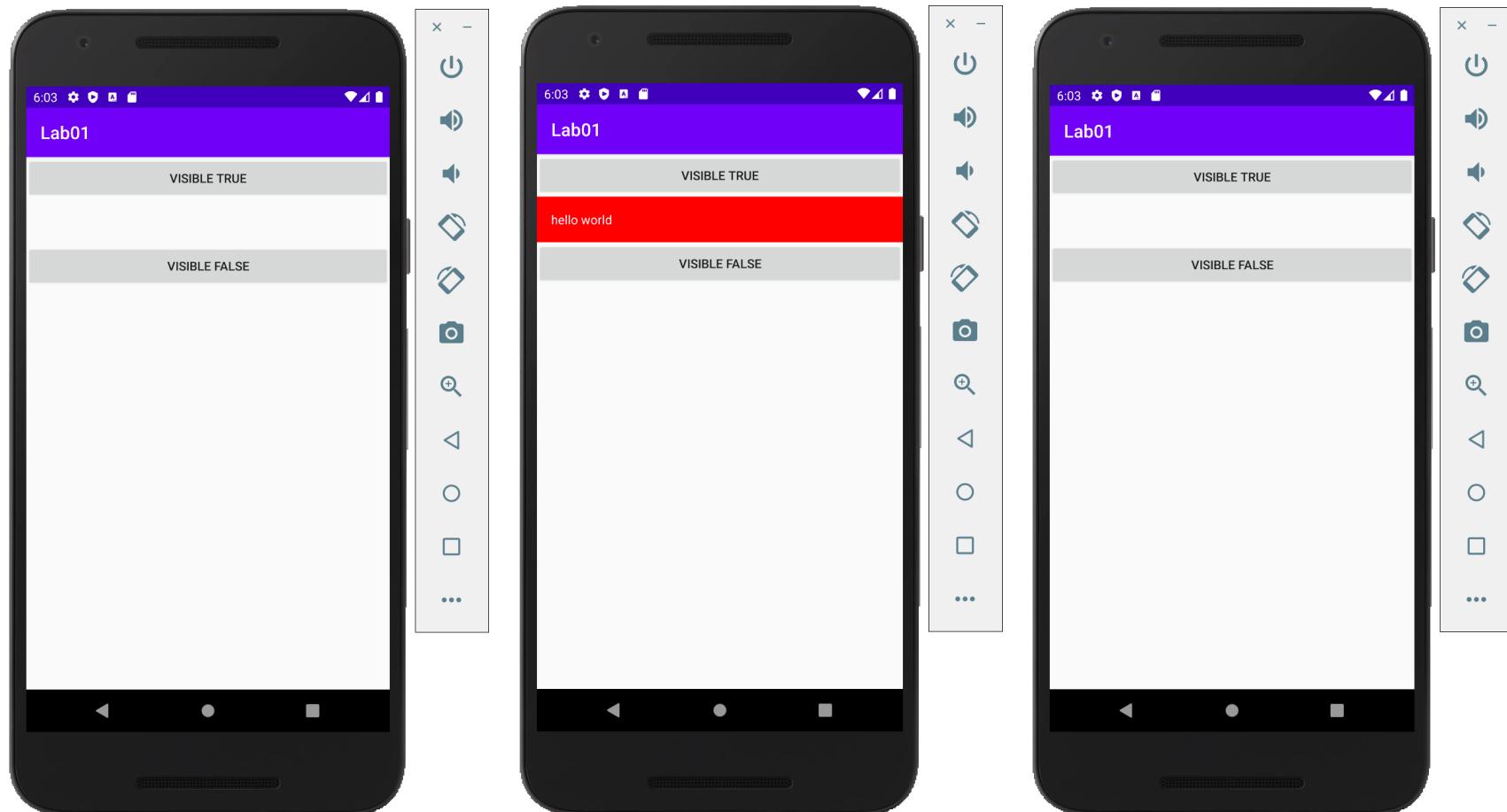
Install successfully finished in 493 ms. (a minute ago)

- 안드로이드 창 Lab01_3Activity.java

The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The project is named "InterfaceCourse". The "lab01" module contains an "app" folder, a "manifests" folder, and a "java" folder. Inside "java", there are three packages: "com.example.lab01", "com.example.lab01.lab01_2Activity", and "com.example.lab01.lab01_3Activity". The "Lab01_3Activity" class is selected in the code editor.
- Code Editor:** The code for `Lab01_3Activity` is displayed. It imports `AppCompatActivity`, `Bundle`, `View`, `Button`, and `TextView`. The class extends `AppCompatActivity` and implements `View.OnClickListener`. It overrides `onCreate` to set the content view and initialize buttons. It also overrides `onClick` to handle button clicks.
- Build Output:** The build log shows a successful build: "BUILD SUCCESSFUL in 1s" with 20 actionable tasks.
- Bottom Bar:** The toolbar includes icons for Run, Build, Profiler, Logcat, Terminal, Event Log, Layout Inspector, and other developer tools.

- AVD 실행 화면 (초기 화면 → visible true 버튼 누른 후 → visible false 버튼 누른 후)



- 학습한 내용

xml 파일에서는 화면에 보여지는 요소를 작업한다.

java 파일에서는 해당 요소들을 바탕으로 이벤트를 작업한다.

먼저, xml 파일을 보면 각각의 뷰들은 고유하게 식별할 수 있는 id가 연결될 수 있다.

우리는 문자열로 id 값을 입력하지만 컴파일 시에는 정수로 참조된다.

TextView에서 visibility를 invisible로 설정해준 것은 공간은 차지하지만 화면상에만 보여지지 않는 상태를 의미한다.

java 파일에서 findViewById를 통해 뷰들의 id를 받아온다.

View.OnClickListener 인터페이스를 상속하기 때문에 onClick 메소드를 구현해야 한다.

클릭 이벤트를 처리하기 위한 리스너는 View.OnClickListener를 구현한 클래스이면 된다.

즉, 해당 클래스 또한 OnClickListener를 구현하고 있기 때문에 리스너가 될 수 있다.

trueBtn과 falseBtn의 리스너로 설정해준다.

클릭 버튼 이벤트는 onClick 메소드에 해준다.

trueBtn이 눌리면 targetTextView의 상태를 visible로 변경해 textView의 글씨가 보이게 한다.

반대로 falseBtn이 눌리면 targetTextView의 상태를 invisible로 변경해 textView의 글씨가 보이지 않게 한다.

》4번

- 안드로이드 창 Lab01_4Activity.java

The screenshot shows the Android Studio interface with the project 'InterfaceCourse' open. The code editor displays the Java file 'Lab01_4Activity.java'. The code implements an activity that sets a custom font and handles a checkbox's checked state change.

```
package com.example.lab01;
import ...;
public class Lab01_4Activity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_lab01_4);

        TextView textView = findViewById(R.id.fontView);
        Typeface typeface = Typeface.createFromAsset(getAssets(), path: "xmas.ttf");
        textView.setTypeface(typeface);

        CheckBox checkBox = findViewById(R.id.checkbox);
        checkBox.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {
            @Override
            public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
                if(isChecked) buttonView.setText("is checked");
                else buttonView.setText("is unchecked");
            }
        });
    }
}
```

The code editor highlights several parts of the code, including the package declaration, imports, class definition, and the checkbox's text change logic. The Android Studio interface includes toolbars, a navigation bar, and various inspection and performance analysis tools on the right side.

- 안드로이드 창 activity_lab01_4.xml

The screenshot shows the Android Studio interface with the project 'InterfaceCourse' open. The 'activity_lab01_4.xml' file is selected in the navigation bar. The code editor displays the XML layout for this activity.

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

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="가나다라 http://www.google.com 마바사 a@a.com 아자차카"
        android:autoLink="web|email|phone" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="@string/long_text"
        android:layout_marginTop="16dp"
        android:ellipsize="end"
        android:maxLines="3" />

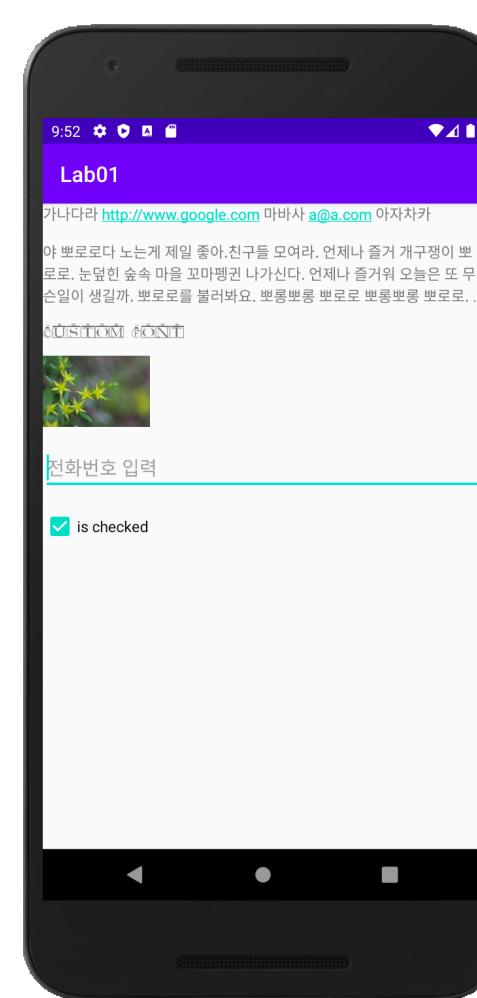
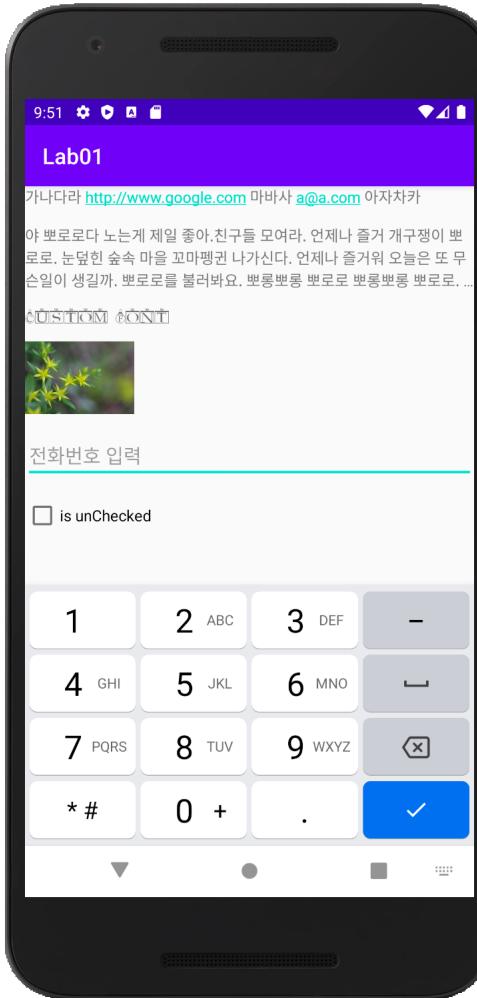
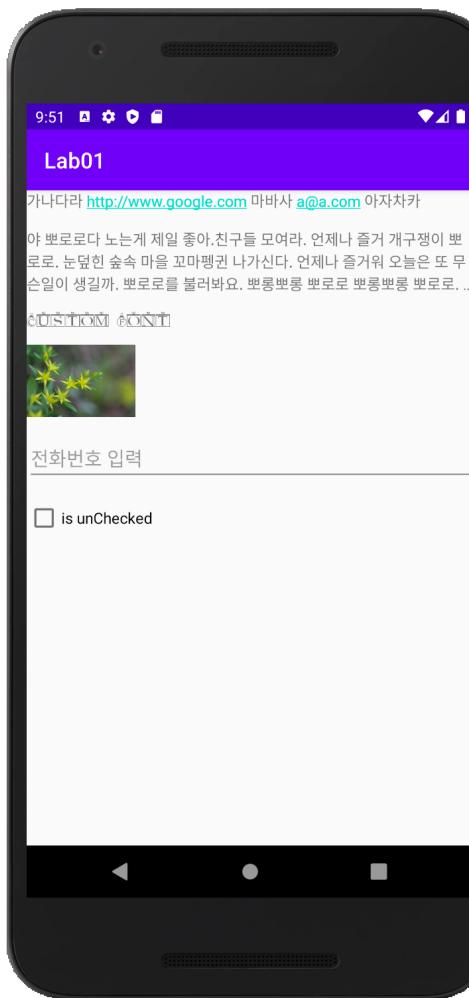
    <TextView
        android:id="@+id/fontView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Custom Font"
        android:layout_marginTop="16dp" />

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/sample"
        android:maxLength="100dp"
        android:maxLength="100dp" />

```

The code includes several TextViews and an ImageView. The first TextView contains a URL and email links. The second TextView has a fixed height and margin. The third TextView uses a custom font and has a fixed height and margin. The fourth TextView is a placeholder for a custom font. The ImageView has a fixed width and height and uses a sample drawable.

- AVD 실행 화면 (초기 화면 -> 전화번호 EditText를 누른 화면 -> CheckBox를 체크한 화면)



- 학습한 내용

문자열 등록은 string.xml 파일에, ttf 파일은 assets 폴더를 생성해 저장한다.

먼저 xml 파일에서, 첫번째 TextView에 autolink를 등록해 웹사이트 주소, 이메일 주고, 핸드폰 번호가 나오면 링크를 연결해준다.

string.xml 파일에 등록해두었던 long_text는 '@string/lone_text'로 사용한다.

layout_marginTop은 뷰의 위쪽 여백을 설정해준다.

최대 라인수를 3으로 설정해주고, ellipsize="end" 옵션을 주면 초과했을 시 문장 끝에는 ...이 오게 한다.

이미지 사이즈를 지정해준 뒤, adjustViewBounds를 true로 설정해주면 지정한 사이즈보다 이미지 크기가 작을 경우, 여백을 남겨두지 않도록 한다.

EditText에서 inputType을 phone으로 설정해 키보드 입력 모양이 숫자가 되게 한다.

hint는 입력을 받는 텍스트 상자에 흐릿한 색상으로 표기되는 글자를 말한다.

java 파일에서

```
Typeface typeface = Typeface.createFromAsset(getAssets(), "xmas.ttf");
textView.setTypeface(typeface);
```

로 글꼴을 설정한다.

CheckBox 이벤트를 설정해주기 위해 리스너를 생성한다.

checkBox 이벤트는 CheckBox 클래스 안의 setOnCheckedChangeListener 메소드에서 설정한다.

checkbox를 check 하면, 'is checked'라는 문자가 나오고,

check 하지 않으면, 'is unchecked'라는 문자가 나온다.