안드로이드 앱 프로그래밍

Chapter 04

레이아웃과 기본 위젯 사용하기

이번 장에서는 무엇을 다룰까요?

화면을 좀 더 잘 꾸미고 싶어요

- 텍스트뷰나 버튼과 같은 기본 위젯을 사용해 볼까요?
- 드로어블을 사용하는 방법에 대해 알아볼까요?

이번 장에서는 무엇을 다룰까요?

버튼이나 텍스트뷰를 다루는 방법을 좀 더 알고 싶어요
- 기본 위젯들 살펴보기

▼

드로어블은 어떻게 만들 수 있나요?
- 배경 설정하기
- 드로어블 만들기

▼

사용자에게 간단한 정보를 보여주고 싶어요
- 이벤트 이해하기
- 토스트, 스낵바 그리고 대화상자 사용하기
- 프로그레스바 사용하기

강의 주제 및 목차

강의 주제

기본 위젯과 드로어블에 대해 이해하기

1. 기본 위젯 다시 한 번 자세히 공부하기
2. 드로어블 만들기
3. 이벤트 처리 이해하기
4. 토스트, 스낵바 그리고 대화상자 사용하기
5. 프로그레스바 사용하기

1. 기본 위젯 다시 한 번 자세히 공부하기

기본 위젯 – 텍스트뷰의 속성

- **텍스트 뷰**

- **text** : 텍스트 뷰에 보이는 문자열을 설정할 수 있음
- **textColor** : 텍스트뷰에서 표시하는 문자열의 색상을 설정함
  : 색상 설정은 "#AARRGGBB" 포맷을 일반적으로 사용(Alpha, Red, Green, Blue)
  : 투명도를 나타내는 Alpha(색상만 표현할 때 - "FF", 투명 - "00", 반투명 - "88")
- **textSize** : 텍스트뷰에서 표시하는 문자열의 크기를 설정함
  ("dp"나 "sp" 또는 "px" 등의 단위 값을 사용함)
- **textStyle** : 텍스트뷰에서 표시하는 문자열의 스타일 속성을 설정함
  ("normal", "bold", "italic" 등의 값을 지정할 수 있음)
- **typeFace** : 텍스트뷰에서 표시하는 문자열의 폰트를 설정함
  ("normal", "sans", "serif", "monospace")
- **maxLines="1"** : 텍스트뷰에서 표시하는 문자열이 한 줄로만 표시되도록 설정함

기본 위젯 – 텍스트뷰의 속성 사용

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  >
  <TextView
    android:id="@+id/TextView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:background="#ff000055"
    android:padding="3px"
    android:text="사용자 이름을 입력하세요. 이름은 한 줄로 표시됩니다."
    android:textSize="22sp"
    android:textStyle="bold"
    android:textColor="#88ff8888"
    android:singleLine="true"
    android:gravity="center"
    />
</LinearLayout>
```

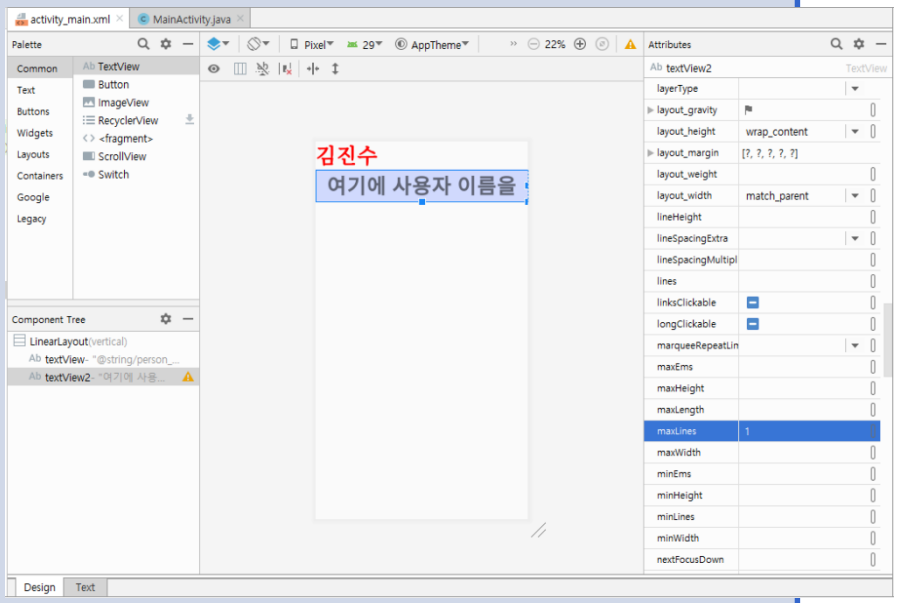
1 배경색 설정

2 크기 설정

3 스타일 설정

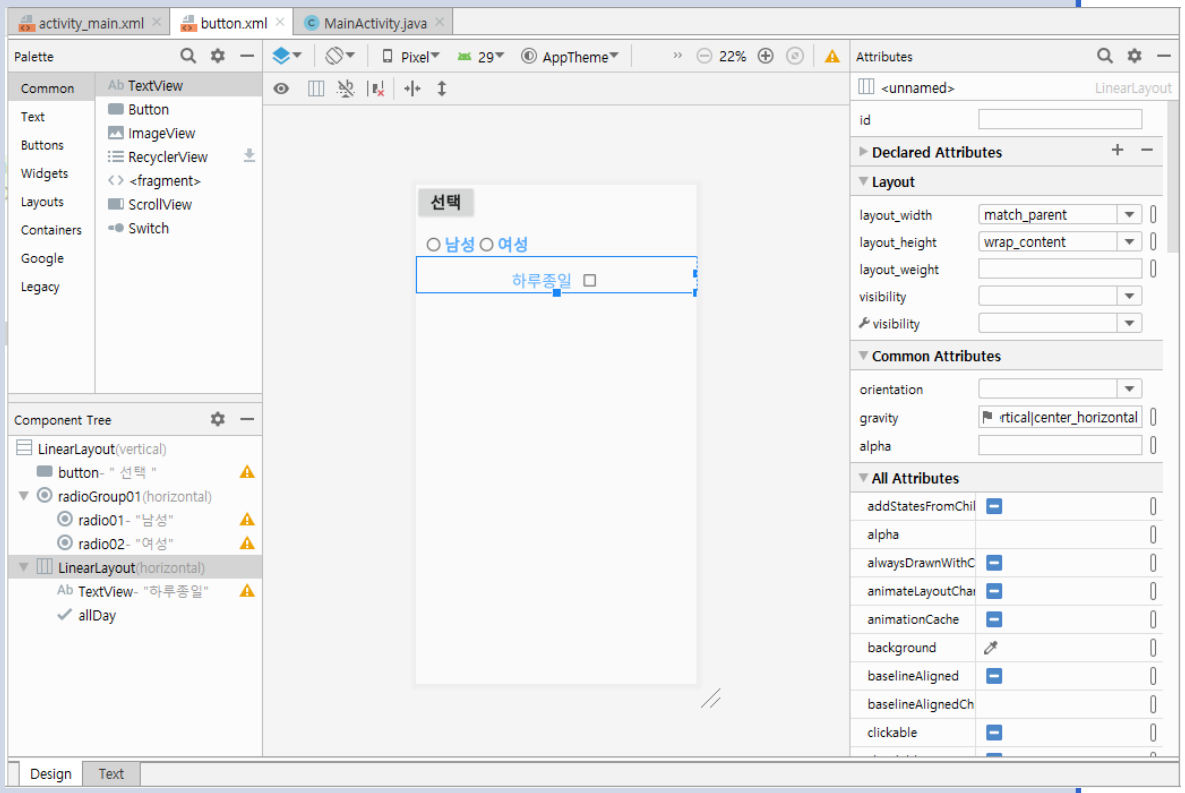
4 색상 설정

5 한 줄 설정



기본 위젯 – 버튼의 속성 사용

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
 android:orientation="vertical"
 android:layout_width="match_parent"
 android:layout_height="match_parent"
 >
 <Button
  android:id="@+id/btnExit"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text=" 선택 "
  android:textSize="24dp"
  android:textStyle="bold"
  android:gravity="center"
  />
```



Continued..

기본 위젯 – 버튼의 속성 사용 (계속)

```
<RadioGroup
 android:id="@+id/radioGroup01"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content"
 android:orientation="horizontal"
 android:paddingLeft="5dp"
 android:paddingRight="5dp"
 >
 <RadioButton
  android:id="@+id/radio01"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_weight="1"
  android:text="남성"
  android:textColor="#ffaaff10"
  android:textStyle="bold"
  android:textSize="24dp"
  />
```

Continued..

```
 <RadioButton
  android:id="@+id/radio02"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_weight="1"
  android:text="여성"
  android:textColor="#ffaaff10"
  android:textStyle="bold"
  android:textSize="24dp"
  />
</RadioGroup>
<LinearLayout
 android:layout_width="match_parent"
 android:layout_height="wrap_content"
 android:gravity="center_vertical|center_horizontal"
 android:paddingTop="10dp"
 >
```

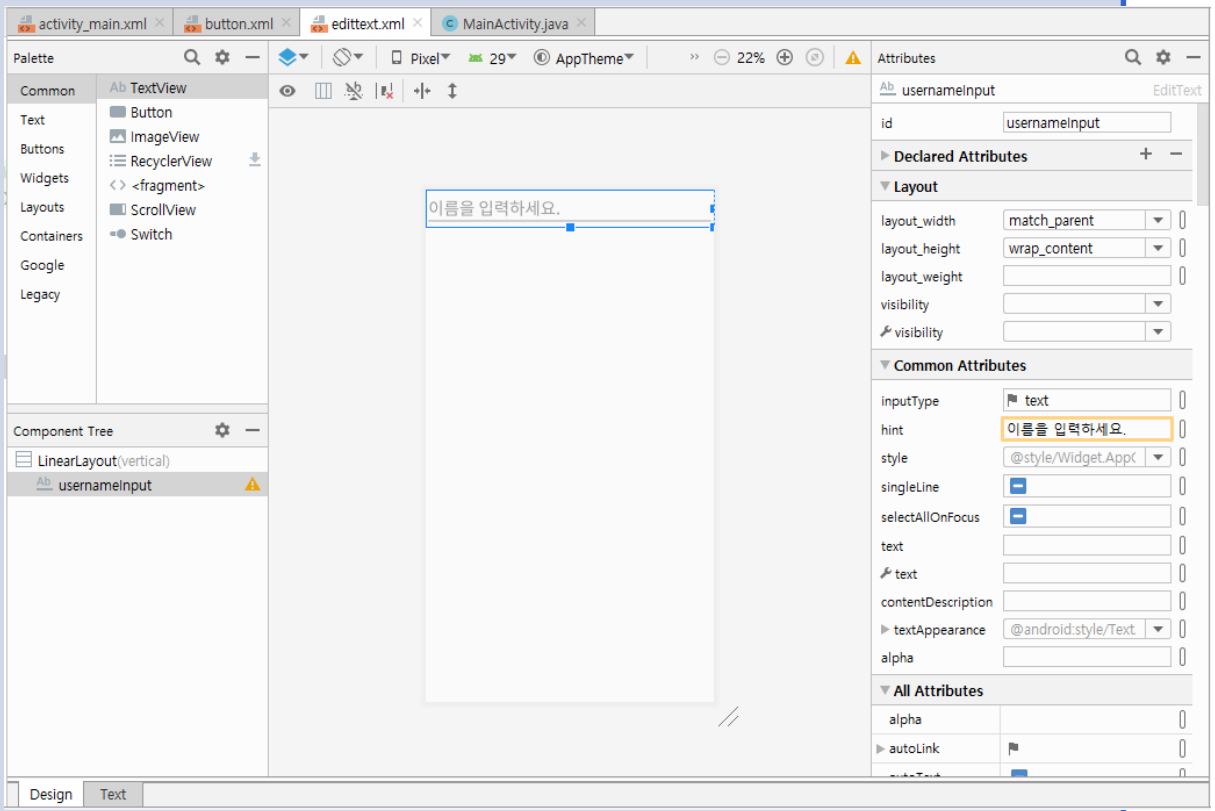
Continued..

기본 위젯 – 버튼의 속성 사용 (계속)

```
    <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="하루종일"
    android:textSize="24dp"
    android:paddingRight="10dp"
    android:textColor="#ffaaff10"
    />
    <CheckBox
    android:id="@+id/allDay"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    />
  </LinearLayout>
</LinearLayout>
```

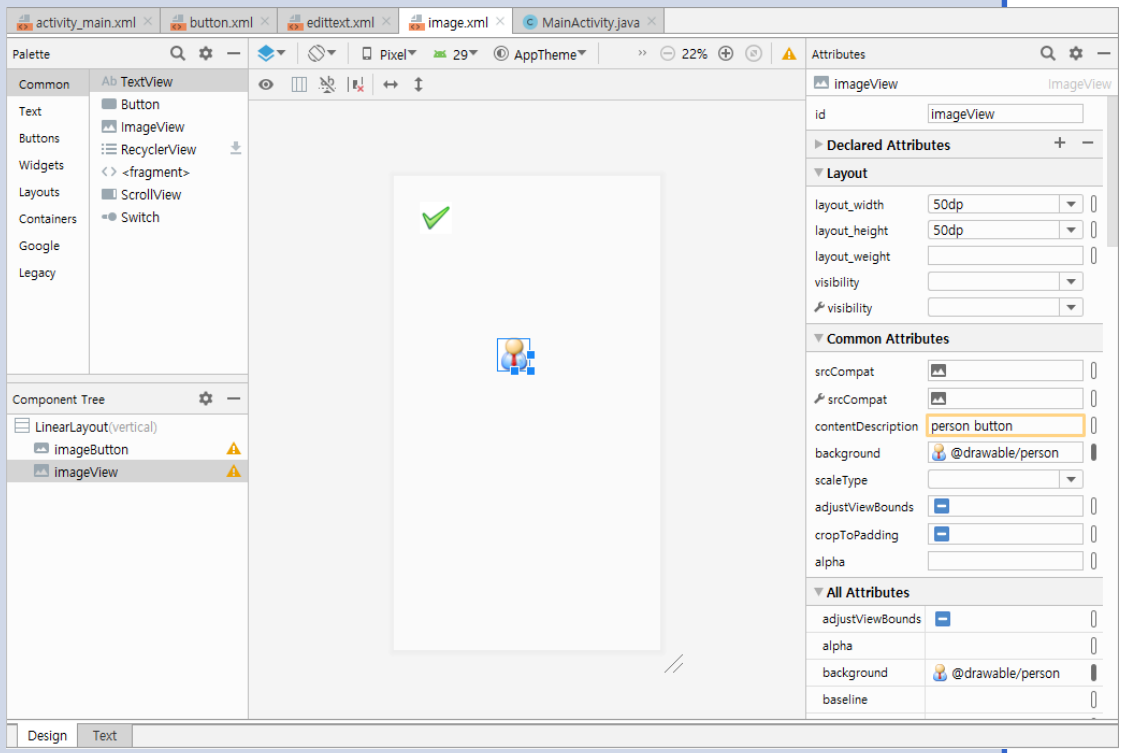
기본 위젯 – 입력상자의 속성 사용

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:orientation="vertical"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  >
  <EditText
    android:id="@+id/editText1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textSize="18sp"
    android:autoText="true"
    android:capitalize="words"
    android:hint="이름을 입력하세요."
    >
  </EditText>
</LinearLayout>
```



기본 위젯 – 이미지뷰의 속성 사용

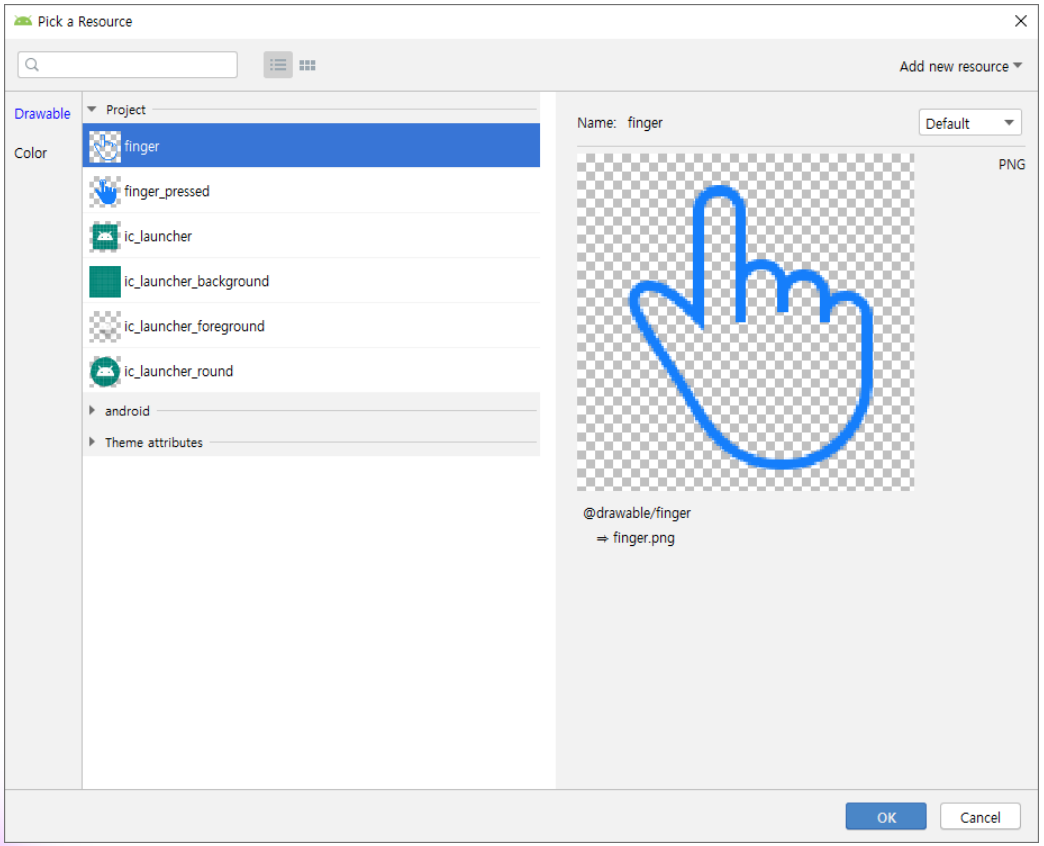
```
<ImageButton
    android:id="@+id/ImageButton01"
    android:background="@drawable/ok_btn"
    android:layout_width="24dp"
    android:layout_height="24dp"
    ...
    />

  <ImageView
      android:id="@+id/ImageView01"
      android:background="@drawable/person"
      android:layout_width="32dp"
      android:layout_height="32dp"
      ...
     />
```

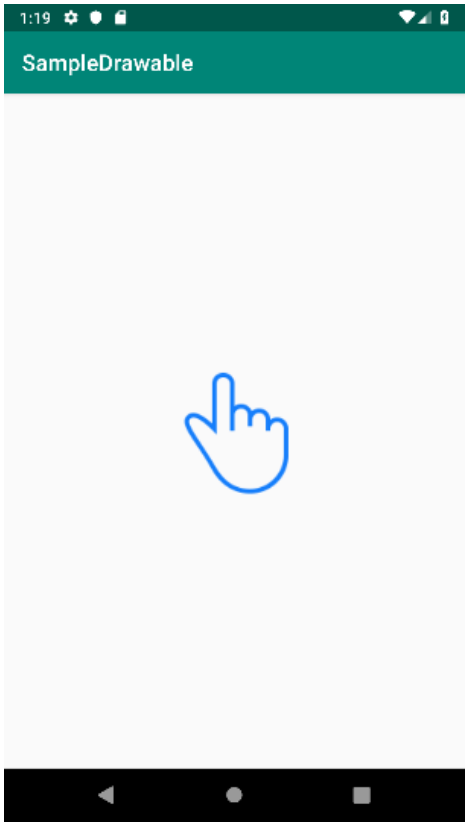


2. 드로어블 만들기

뷰의 배경으로 이미지 설정하기

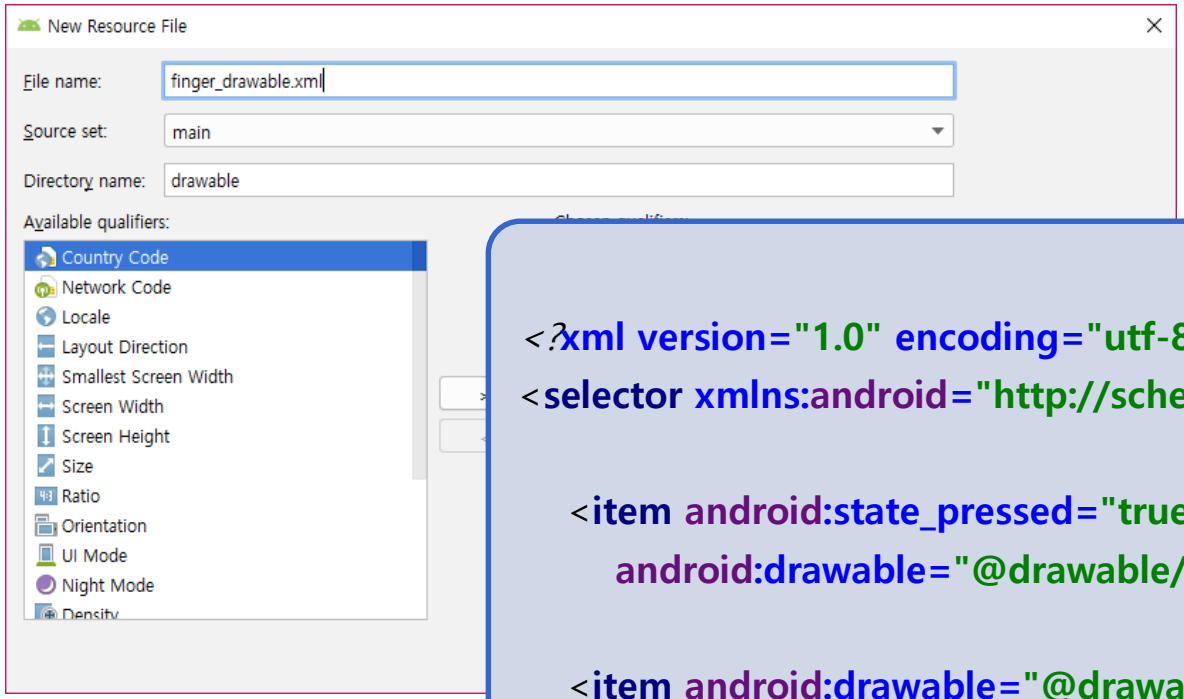
- 뷰의 background 속성으로 이미지 설정
- /app/res/drawable 폴더 안에 넣어둔 이미지를 설정할 수 있음





상태 드로어블 만들기

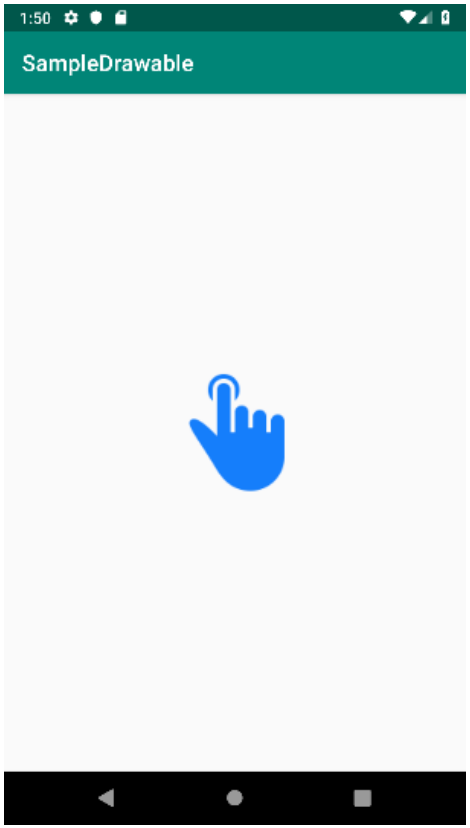
- 상태에 따라 다른 이미지를 보여줄 수 있음
- /app/res/drawable 폴더 안에 xml 파일을 만들고 그 안에 정의

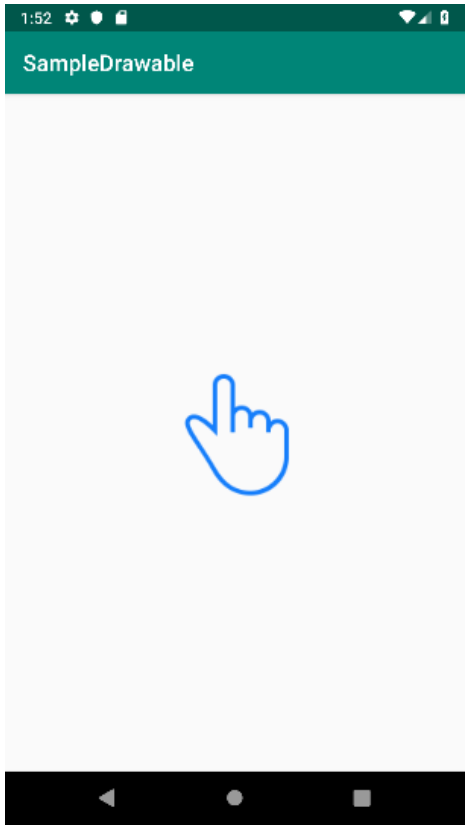



```xml
<?xml version="1.0" encoding="utf-8"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">

    <item android:state_pressed="true"
        android:drawable="@drawable/finger_pressed" />

    <item android:drawable="@drawable/finger" />

</selector>
```

상태 드로어블 만들기

- 뷰의 background 속성 값으로 xml 설정
- 앱 실행 후 뷰를 누르면 눌린 상태에서 다른 이미지로 변경됨



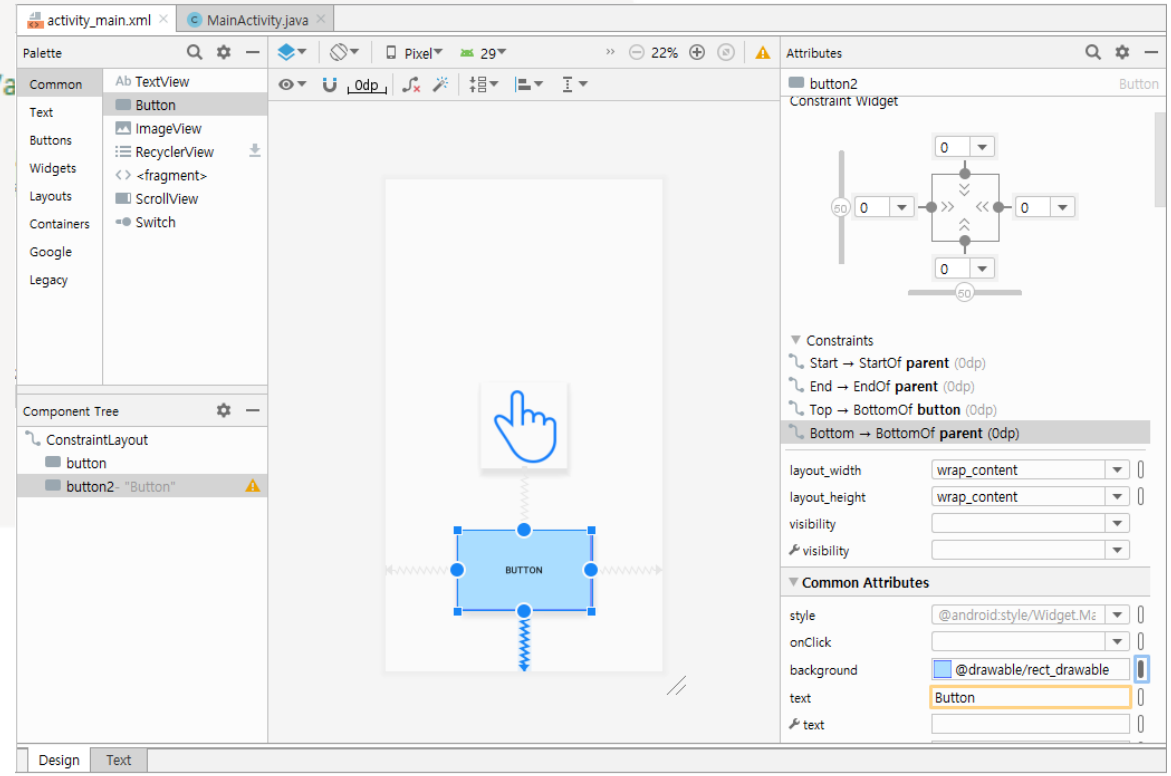


셰이프 드로어블 만들기

- 도형을 그릴 수 있음
- /app/res/drawable 폴더 안에 xml 파일을 만들고 그 안에 정의

참조파일 SampleDrawable>/app/res/drawable/rect_drawable.xml

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/a
    android:shape="rectangle">

  <size android:width="200dp" android:height="120dp"/>
  <stroke android:width="1dp" android:color="#0000ff"/>
  <solid android:color="#aaddff" />
  <padding android:bottom="1dp" />

</shape>
```

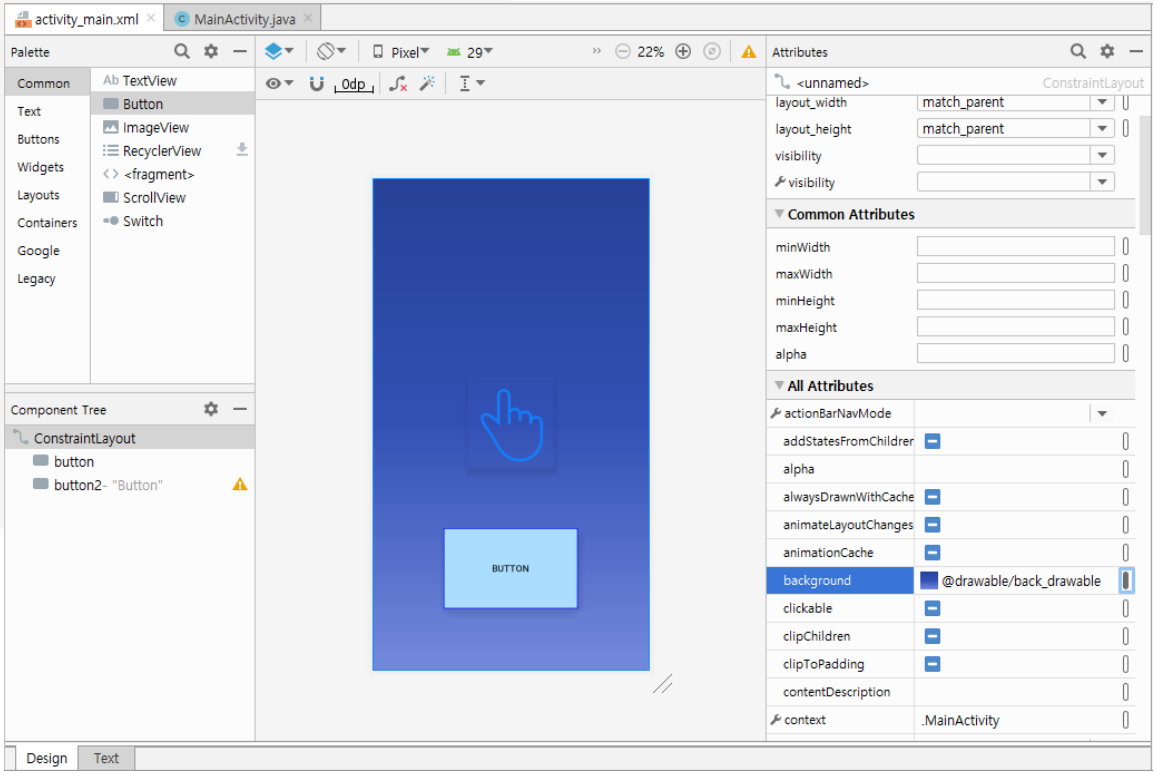


셰이프 드로어블 만들기

- 화면 전체의 배경으로 그러데이션 적용
- 최상위 레이아웃의 background 속성 값으로 설정

참조파일 SampleDrawable>/app/res/drawable/back_drawable.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">

  <gradient
    android:startColor="#7288DB"
    android:centerColor="#3250B4"
    android:endColor="#254095"
    android:angle="90"
    android:centerY="0.5"
    />

  <corners android:radius="2dp" />

</shape>
```

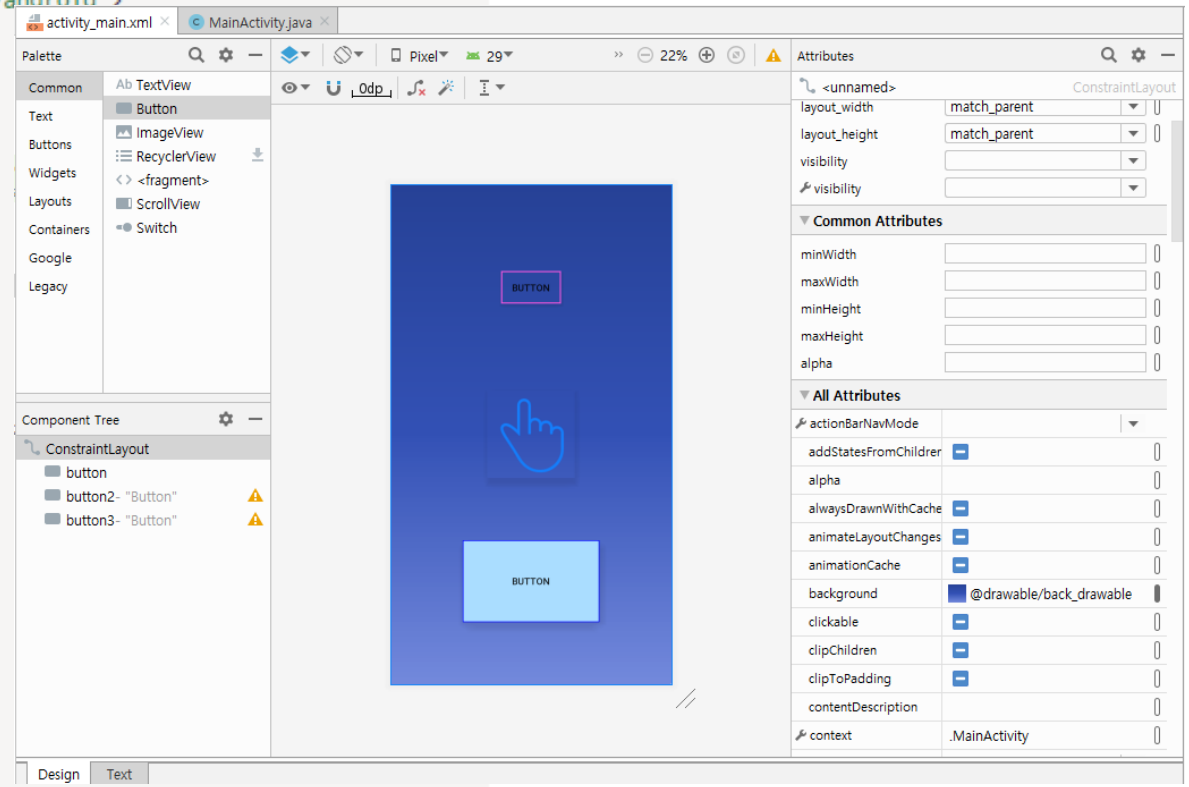


셰이프 드로어블 만들기

- 테두리만 있는 버튼 만들기
- 버튼의 background 속성 값으로 설정

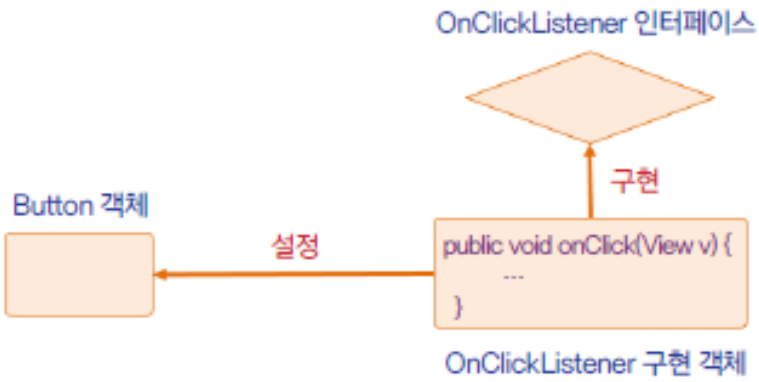
참조파일 SampleDrawable>/app/res/drawable/border_drawable.xml

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android">

  <item>
    <shape android:shape="rectangle">
      <stroke android:width="1dp" android:color="#BE55DA" />
      <solid android:color="#00000000" />
      <size android:width="200dp" android:height="100dp" />
    </shape>
  </item>

  <item android:top="1dp" android:bottom="1dp"
        android:right="1dp" android:left="1dp">
    <shape android:shape="rectangle">
      <stroke android:width="1dp" android:color="#FF55DA" />
      <solid android:color="#00000000" />
    </shape>
  </item>

</layer-list>
```



3. 이벤트 처리 이해하기

뷰의 이벤트 처리하기



[버튼에 OnClickListener를 설정할 때의 패턴]

뷰를 상속할 때 이벤트 처리를 위한 메소드 재정의

boolean onTouchEvent (MotionEvent event)

boolean onKeyDown (int keyCode, KeyEvent event)

boolean onKeyUp (int keyCode, KeyEvent event)

뷰 객체에 전달되는 이벤트를 처리하기 위한 리스너 설정

View.OnTouchListener : boolean onTouch (View v, MotionEvent event)

View.OnKeyListener : boolean onKey (View v, int keyCode, KeyEvent event)

View.OnClickListener : void onClick (View v)

View.OnFocusChangeListener : void onFocusChange (View v, boolean hasFocus)

대표적인 이벤트

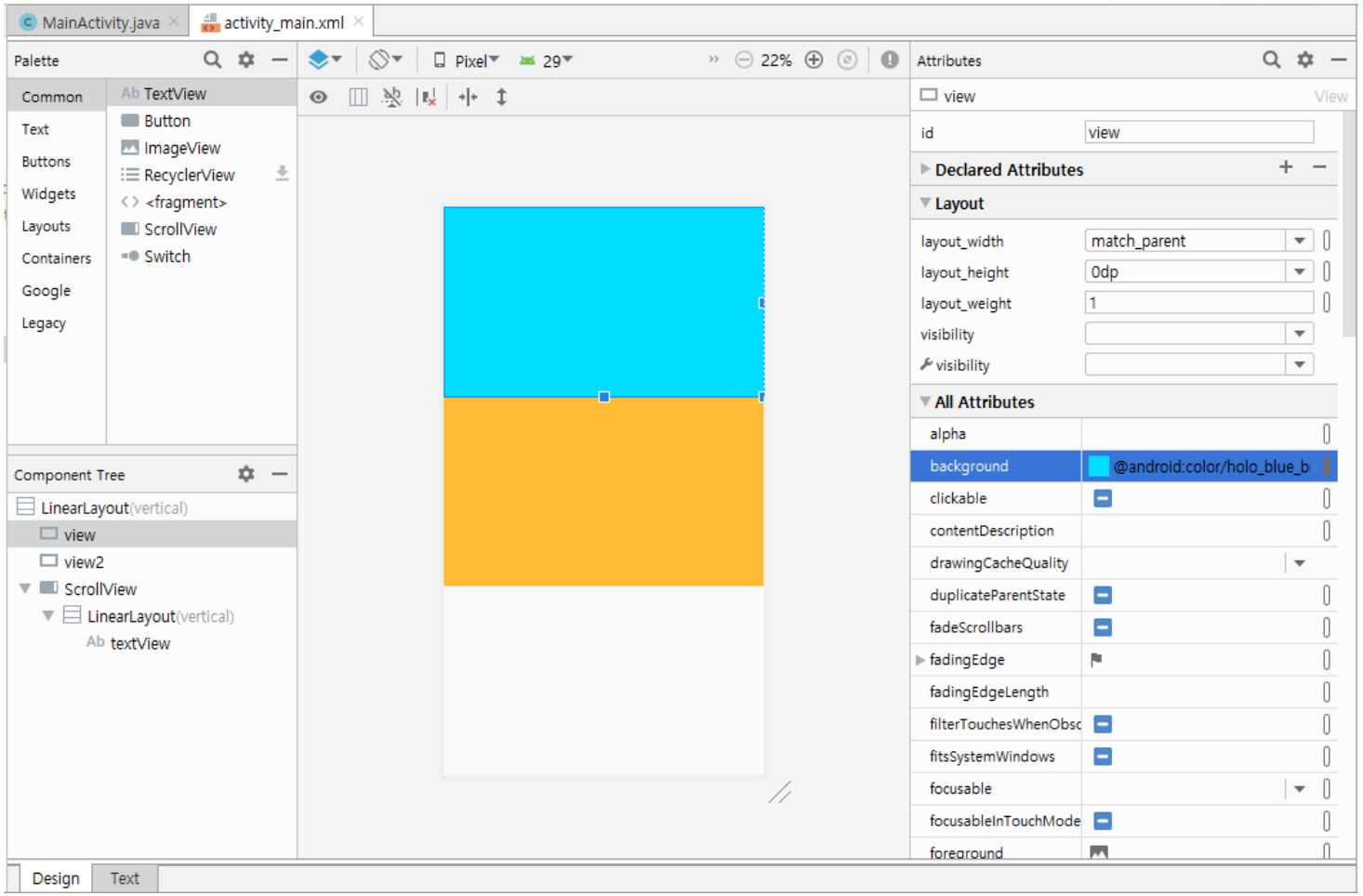
- **터치 이벤트**
  - 화면을 손가락으로 누를 때 발생하는 이벤트
- **키 이벤트**
  - 키패드나 하드웨어 버튼을 누를 때 발생하는 이벤트
- **제스처 이벤트**
  - 터치 이벤트 중에서 일정 패턴을 만들어 내는 이벤트
- **포커스**
  - 뷰마다 순서대로 주어지는 포커스
- **화면 방향 변경**
  - 화면의 방향이 가로/세로로 바뀜에 따라 발생하는 이벤트

제스처를 통해 처리할 수 있는 이벤트

메소드	이벤트 유형
onDown()	- 화면이 눌렸을 경우
onShowPress()	- 화면이 눌렸다 떼어지는 경우
onSingleTapUp()	- 화면이 한 손가락으로 눌렸다 떼어지는 경우
onSingleTapConfirmed()	- 화면이 한 손가락으로 눌려지는 경우
onDoubleTap()	- 화면이 두 손가락으로 눌려지는 경우
onDoubleTapEvent()	- 화면이 두 손가락으로 눌려진 상태에서 떼거나 이동하는 등 세부적인 액션을 취하는 경우
onScroll()	- 화면이 눌린 채 일정한 속도화 방향으로 움직였다 떼는 경우
onFling()	- 화면이 눌린 채 가속도를 붙여 손가락을 움직였다 떼는 경우
onLongPress()	- 화면을 손가락으로 오래 누르는 경우

터치 이벤트 처리하기

- SampleEvent 프로젝트를 만들고 XML 레이아웃 구성



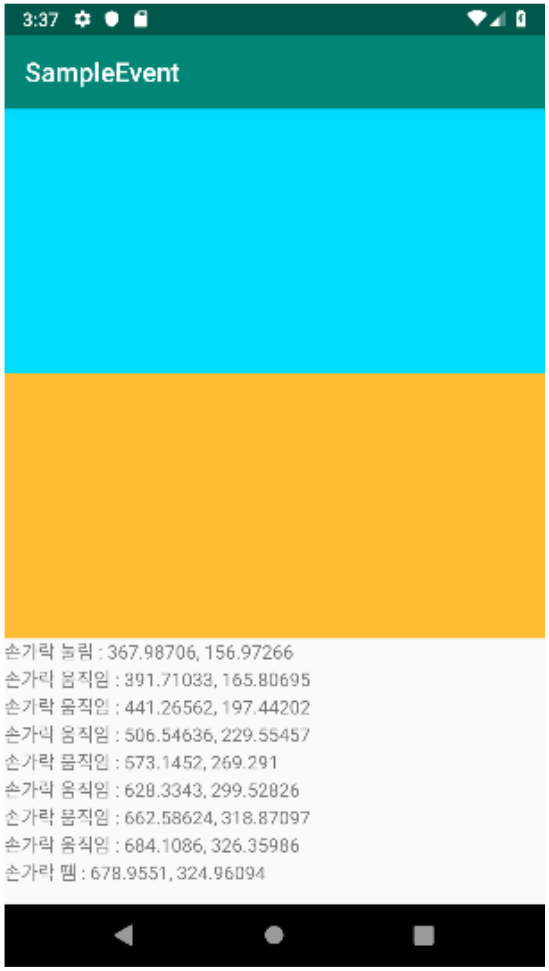
메인 액티비티 코드 만들기

```
View view = findViewById(R.id.view);
view.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View view, MotionEvent motionEvent) {
        int action = motionEvent.getAction();
        float curX = motionEvent.getX();
        float curY = motionEvent.getY();

        if (action == MotionEvent.ACTION_DOWN) {
            println("손가락 눌림 : " + curX + ", " + curY);
        } else if (action == MotionEvent.ACTION_MOVE) {
            println("손가락 움직임 : " + curX + ", " + curY);
        } else if (action == MotionEvent.ACTION_UP) {
            println("손가락 뗌 : " + curX + ", " + curY);
        }

        return true;
    }
});
```

Continued..

앱 실행 결과

- 가장 위쪽에 있는 부분을 터치했을 때 가장 아래쪽에 보이는 로그 메시지 확인



제스처 이벤트 처리하기

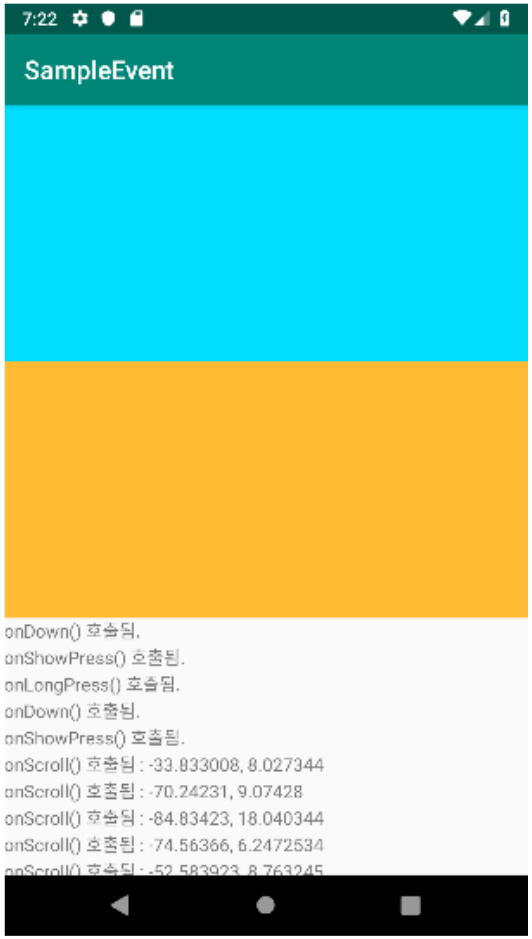
- GestureDetector 객체 만들고 터치 이벤트 발생 시 해당 객체 전달

```
public boolean onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY) {
    println("₩nonFling ₩n₩tx = " + velocityX + "₩n₩ty=" + velocityY);
    return super.onFling(e1, e2, velocityX, velocityY);
}

public boolean onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY) {
    println("₩nonScroll ₩n₩tx = " + distanceX + "₩n₩ty = " + distanceY);
    return super.onScroll(e1, e2, distanceX, distanceY);
}
```

Continued..

앱 실행 결과

- 중간 부분을 터치했을 때 가장 아래쪽에 보이는 로그 메시지 확인



키 입력 이벤트 처리하기

뷰를 상속할 때 키 이벤트 처리를 위한 메소드 재정의

```
boolean onKeyDown (int keyCode, KeyEvent event)
boolean onKey (View v, int keyCode, KeyEvent event)
```

[키를 눌렀을 때 전달되는 대표적인 키값]

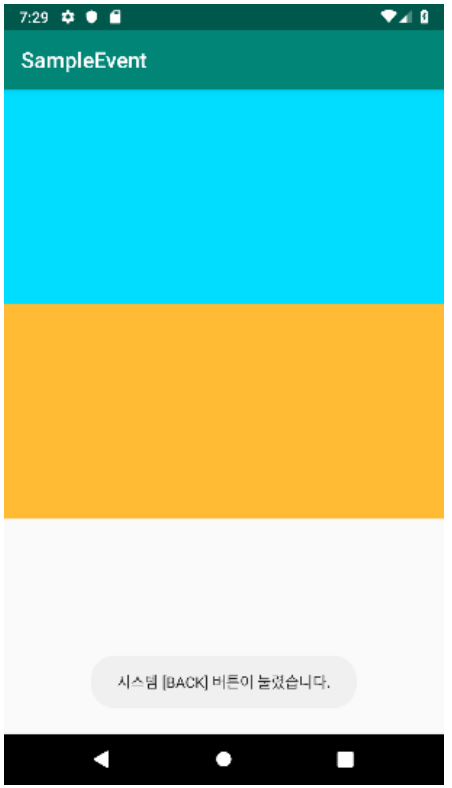
키 코드	설 명
KEYCODE_DPAD_LEFT	- 왼쪽 화살표
KEYCODE_DPAD_RIGHT	- 오른쪽 화살표
KEYCODE_DPAD_UP	- 위쪽 화살표
KEYCODE_DPAD_DOWN	- 아래쪽 화살표
KEYCODE_DPAD_CENTER	- [중앙] 버튼
KEYCODE_CALL	- [통화] 버튼
KEYCODE_ENDCALL	- [통화 종료] 버튼
KEYCODE_HOME	- [홈] 버튼
KEYCODE_BACK	- [뒤로 가기] 버튼
KEYCODE_VOLUME_UP	- [소리 크기 증가] 버튼
KEYCODE_VOLUME_DOWN	- [소리 크기 감소] 버튼
KEYCODE_0 ~ KEYCODE_9	- 숫자 0부터 9까지의 키값
KEYCODE_A ~ KEYCODE_Z	- 알파벳 A부터 Z까지의 키값

BACK 버튼 이벤트 처리하기

- 시스템 BACK 키를 눌렀을 때 동작할 코드 정의
- onBackPressed 메소드 재정의 또는 onKeyDown 메소드 안에서 정의

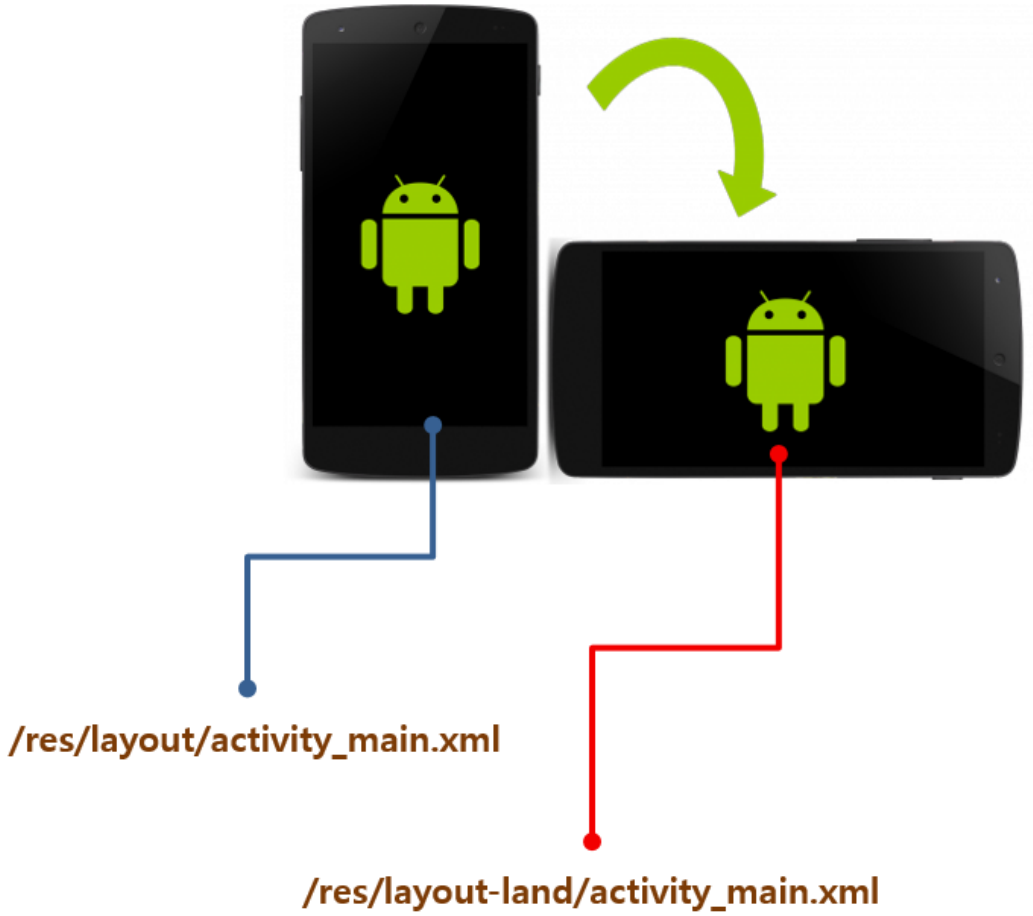
참조파일 SampleEvent>/app/java/org.techtown.sampleevent/MainActivity.java

```
중략…

@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {

    if(keyCode == KeyEvent.KEYCODE_BACK) {
        Toast.makeText(this, "시스템 [BACK] 버튼이 눌렸습니다.",
                Toast.LENGTH_LONG).show();

        return true;
    }

    return false;
}

중략…
```



단말 방향 전환

- 병렬 리소스 로딩 방식 사용
- [res] 폴더 안에 [layout] 폴더와 [layout-land] 폴더 생성

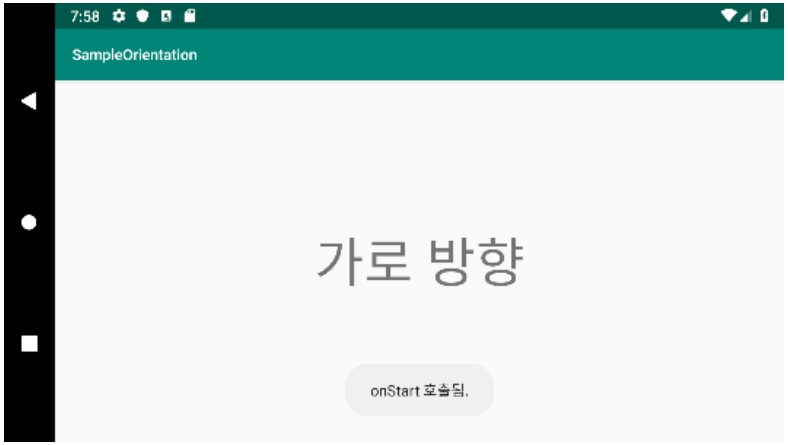


소스 코드에서 토스트 메시지 표시

```
public class MainActivity extends AppCompatActivity {
@Override
protected void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  setContentView(R.layout.activity_main);
  showToast("onCreate 호출됨.");
}

@Override
protected void onStart() {
  super.onStart();
  showToast("onStart 호출됨.");
}

@Override
protected void onStop() {
  super.onStop();
  showToast("onStop 호출됨.");
}
```

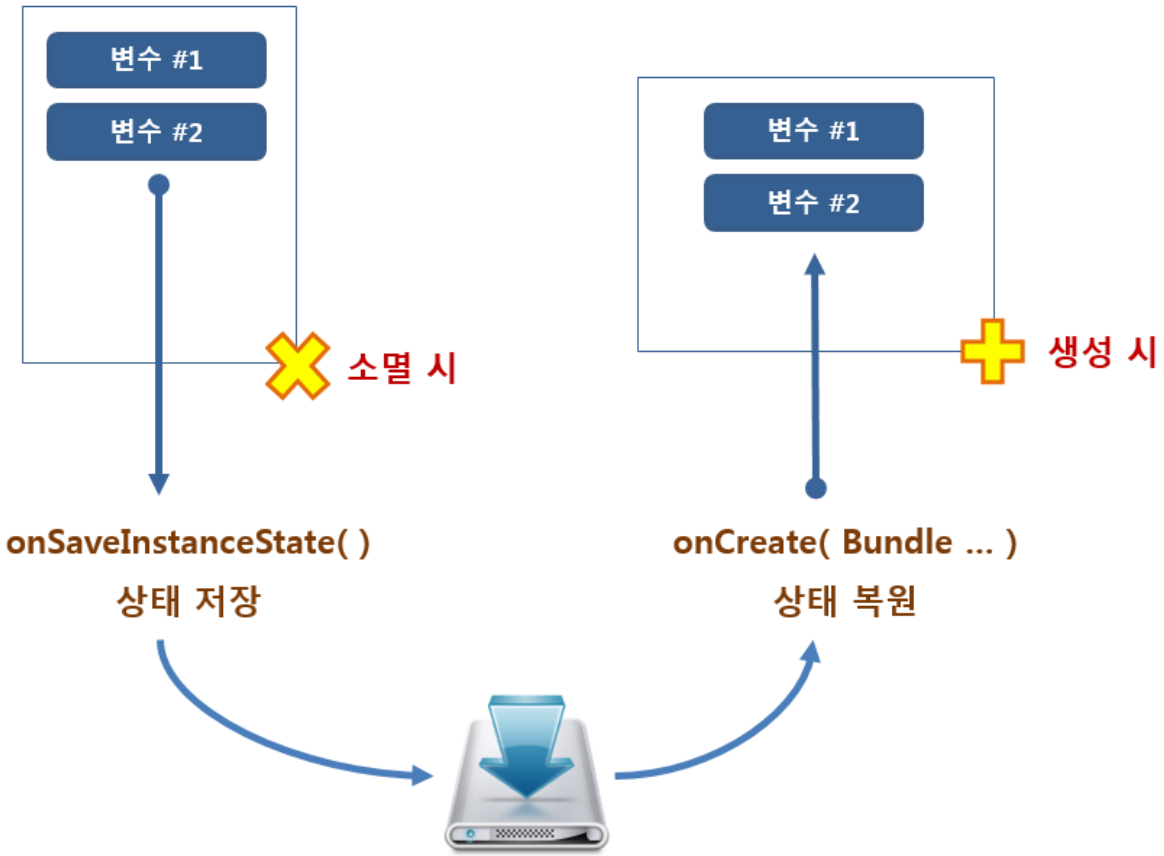
앱 실행 결과

- 단말 방향 바꾸었을 때 액티비티가 새로 생성되므로 토스트 메시지 표시됨



단말 방향 전환 시 상태 저장과 복원

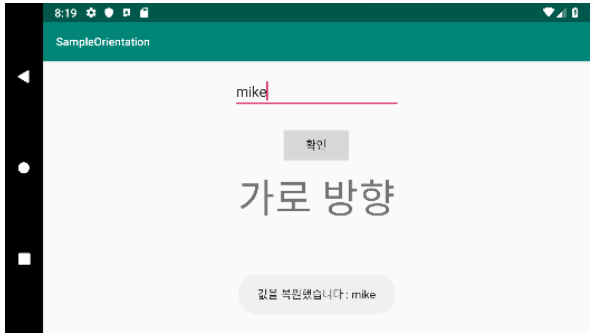
- onSaveInstanceState 메소드에서 상태 저장했다가 onCreate의 파라미터를 이용해 복원



단말 방향전환 상태 저장 예제

단말 방향전환 상태저장 예제

-단말 방향이 가로와 세로로 바뀌었을 때 상태 저장과 복원



매니페스트 속성 추가

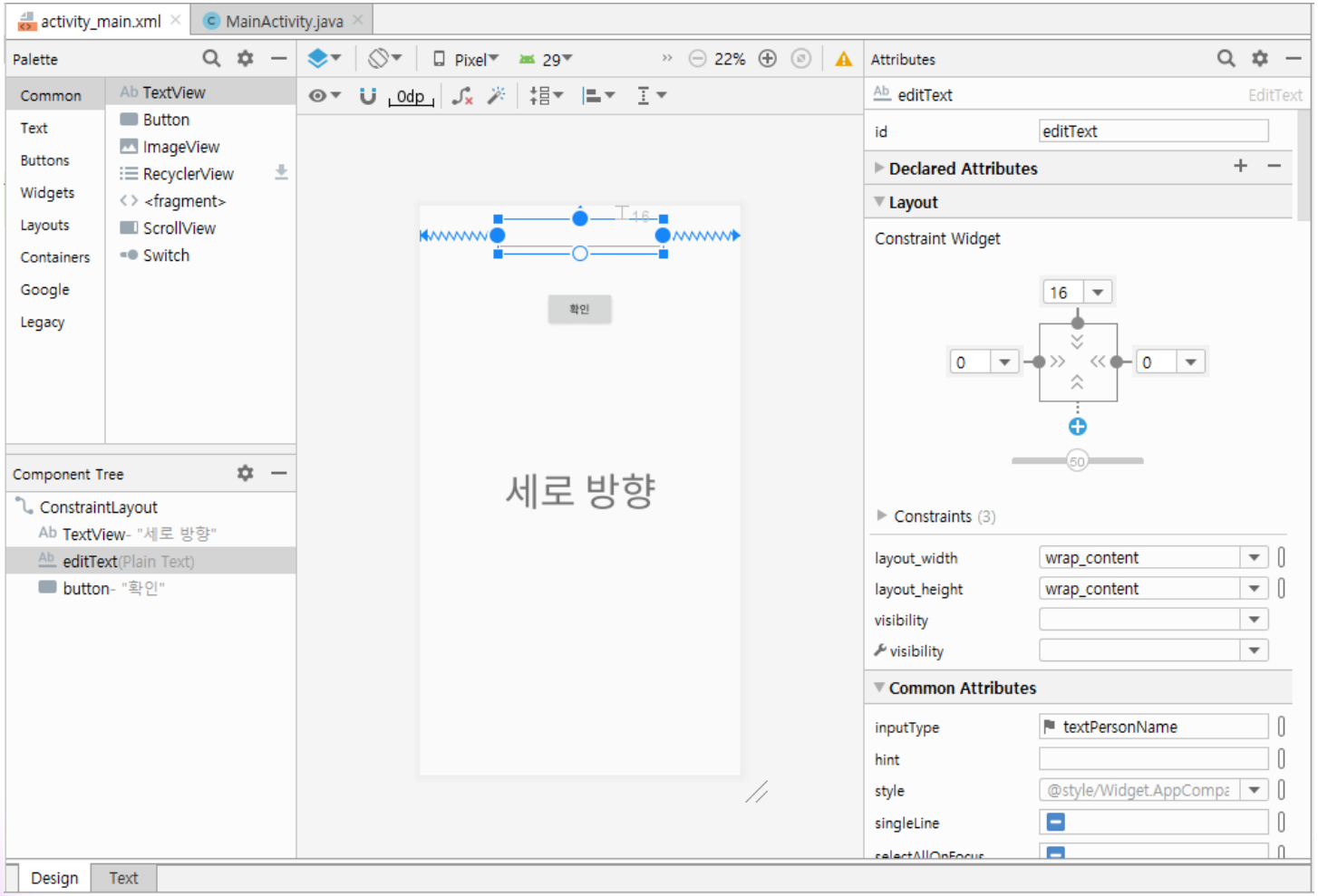
-매니페스트의 액티비티 속성 추가

메인 액티비티 코드 작성

-가로와 세로 방향으로 바뀌었을 때 처리 코드 작성

XML 레이아웃 구성

- 입력상자와 버튼 추가



소스 코드에서 값의 저장과 복원

```
protected void onCreate(Bundle savedInstanceState) {
...
  if (savedInstanceState != null) {
    name = savedInstanceState.getString("name");

    Toast.makeText(getApplicationContext(), "값을 복원했습니다 : " + name, Toast.LENGTH_LONG).show();
  }
}

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

  outState.putString("name", name);
}
...
```

단말 방향 전환 시 액티비티 유지

- AndroidManifest.xml 파일에 configChanges 속성 설정

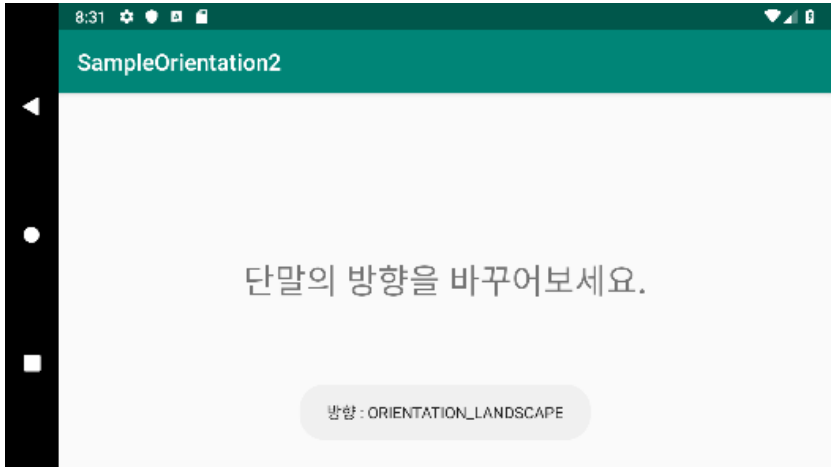
```
...
<activity android:name=".MainActivity"
    android:configChanges="orientation|screenSize|keyboardHidden"
    >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
...
```

소스 코드에서 방향 전환 이벤트 전달받음

```
public class MainActivity extends AppCompatActivity {
...
    public void onConfigurationChanged(Configuration newConfig) {
        super.onConfigurationChanged(newConfig);

        if (newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE) {
            showToast("방향 : ORIENTATION_LANDSCAPE");
        } else if (newConfig.orientation == Configuration.ORIENTATION_PORTRAIT) {
            showToast("Orientation : ORIENTATION_PORTRAIT");
        }
    }
}

...
```

앱 실행 결과

- 단말 방향 바꾸었을 때 값 액티비티는 유지되고 이벤트 전달받음



4. 토스트, 스낵바 그리고 대화상자 사용하기

토스트와 대화상자

- 토스트

- 간단한 메시지를 잠깐 보여주었다가 없어지는 뷰로 애플리케이션 위에 떠 있는 뷰라 할 수 있음

[Code]

```
Toast.makeText(Context context, String message, int duration)
```

[Code]

```
public void setGravity(int gravity, int xOffset, int yOffset)
public void setMargin(float horizontalMargin, float verticalMargin)
```

토스트 만들기 예제

토스트 만들기 예제

-토스트의 색상이나 모양을 직접 구성

-새로운 레이아웃 정의

메인 액티비티
XML 레이아웃 정의

-메인 액티비티의 레이아웃 정의

메인 액티비티 코드 작성

-메인 액티비티에서 위치 설정

토스트를 위한
XML 레이아웃 정의

-토스트의 모양을 XML 레이아웃으로 정의

메인 액티비티 코드 작성

-메인 액티비티에서 모양 설정

위치가 바뀐 토스트

```
    public void onButton1Clicked(View v) {
        try {
            Toast toastView = Toast.makeText(this, "위치가 바뀐 토스트 메시지입니다.",
                                             Toast.LENGTH_LONG);
            int xOffset = Integer.parseInt(editText.getText().toString());
            int yOffset = Integer.parseInt(editText2.getText().toString());

            toastView.setGravity(Gravity.TOP|Gravity.TOP, xOffset, yOffset);
            toastView.show();
        } catch (NumberFormatException e) {
            e.printStackTrace();
        }
    }
}
```

토스트 모양 바꾸기 – 메인 액티비티 코드 만들기

참조파일 SampleToast>/app/java/org.techtown.sampletoast/MainActivity.java

```
중략…
public void onButton2Clicked(View v) {
  LayoutInflater inflater = getLayoutInflater();  ❶ 레이아웃 인플레이터 객체 참조

  View layout = inflater.inflate(  ❷ 토스트를 위한 레이아웃 인플레이션
                R.layout.toastborder,
                (ViewGroup) findViewById(R.id.toast_layout_root));

  TextView text = layout.findViewById(R.id.text);

  Toast toast = new Toast(this);  ❸ 토스트 객체 생성
  text.setText("모양 바꾼 토스트");
  toast.setGravity(Gravity.CENTER, 0, -100);
  toast.setDuration(Toast.LENGTH_SHORT);
```

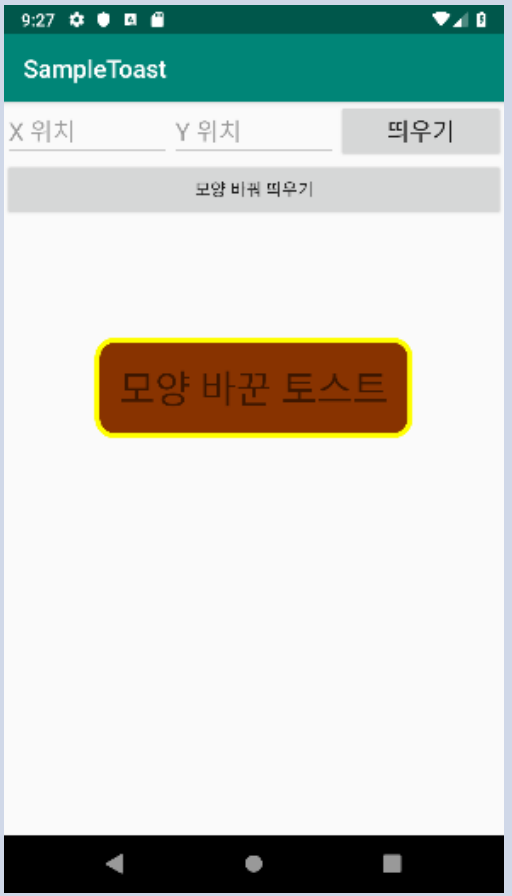
토스트 모양 바꾸기 – 토스트의 XML 레이아웃

참조파일 SampleToast>/app/res/layout/toastborder.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
  android:id="@+id/toast_layout_root"
  android:orientation="horizontal"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  android:padding="10dp"
  >
  <TextView
    android:id="@+id/text"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="20dp"
    android:textSize="32sp"
    android:background="@drawable/toast"
    />
</LinearLayout>
```

Shape 객체를 위한 XML 정의

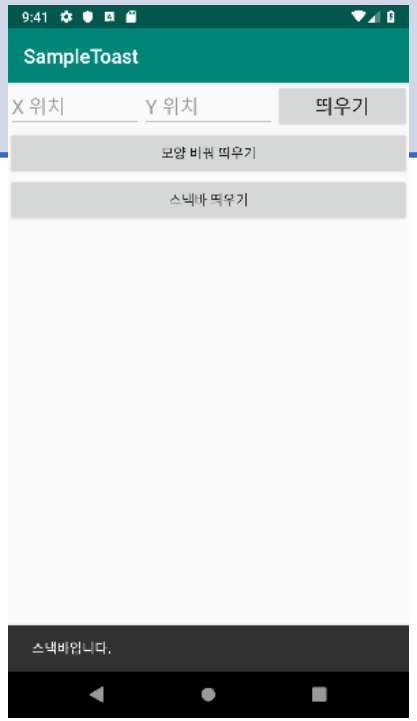
```
<?xml version="1.0" encoding="UTF-8" ?>
<shape xmlns:android="http://schemas.android.com/apk/res/android"
 android:shape="rectangle"
 >
 <stroke
  android:width="4dp"
  android:color="#ffffff00"
  />
 <solid
  android:color="#ff883300"
  />
 <padding
  android:left="20dp"
  android:top="20dp"
  android:right="20dp"
  android:bottom="20dp"
  />
 <corners
  android:radius="15dp"
 />
</shape>
```



스낵바 보여주기

- 사용 코드는 토스트 메시지를 띄울 때와 유사함

```
...
public void onButton3Clicked(View v) {
    Snackbar.make(v, "스낵바입니다.", Snackbar.LENGTH_LONG).show();
}
...
```



알림 대화상자 보여주기 예제

대화상자 만들기 예제

-대화상자 보여주기

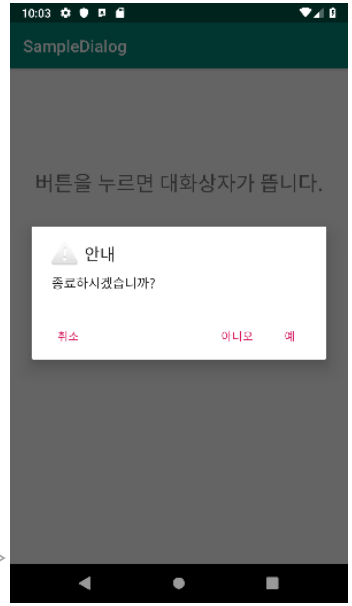
-이벤트 처리

메인 액티비티 XML 레이아웃 정의

-메인 액티비티의 레이아웃 정의

메인 액티비티 코드 작성

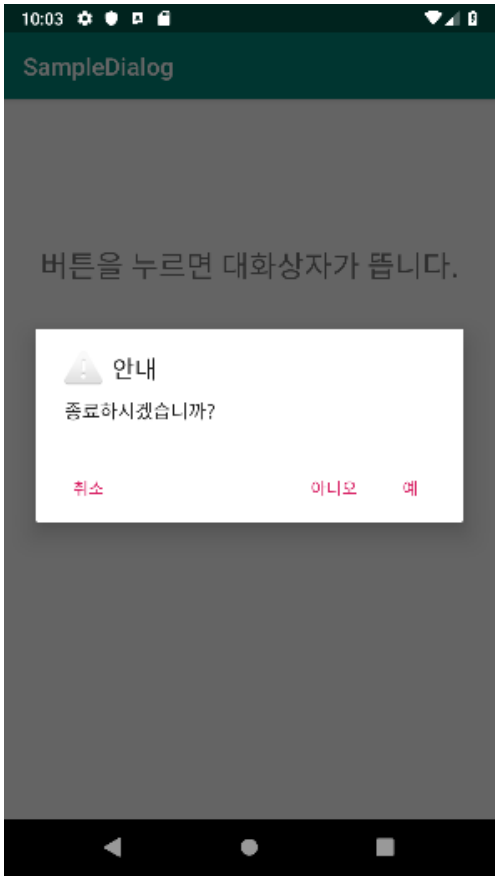
-메인 액티비티에서 대화상자 보여주기



메인 액티비티 코드 만들기

```
private void showMessage() {
    AlertDialog.Builder builder = new AlertDialog.Builder(this); ---> ❶
    builder.setTitle("안내");
    builder.setMessage("종료하시겠습니까?");
    builder.setIcon(android.R.drawable.ic_dialog_alert);

    builder.setPositiveButton("예", new DialogInterface.OnClickListener() { ---> ❷
         public void onClick(DialogInterface dialog, int which) {
            String message = "예 버튼이 눌렸습니다. ";
            textView.setText(message);
        }
    });

    builder.setNeutralButton("취소", new DialogInterface.OnClickListener() { ---> ❸
         public void onClick(DialogInterface dialog, int which) {
            String message = "취소 버튼이 눌렸습니다. ";
            textView.setText(message);
        }
    });

    builder.setNegativeButton("아니오", new DialogInterface.OnClickListener() { ---> ❹
         public void onClick(DialogInterface dialog, int which) {
            String message = "아니오 버튼이 눌렸습니다. ";
            textView.setText(message);
        }
    });
    AlertDialog dialog = builder.create(); ---> ❺
     dialog.show();
  }
}
```

알림 대화상자 실행 화면


10:03
SampleDialog
버튼을 누르면 대화상자가 뜹니다.
안내
종료하시겠습니까?
취소
아니오
예


10:03
SampleDialog
예 버튼이 눌렸습니다.
띄우기

5. 프로그레스바 사용하기

프로그레스바 사용하기

- 여러 가지 화면을 구성하고 그 안에 다양한 위젯을 사용하는데 있어서 대화상자처럼 중간 중간 상태 정보를 보여주는 가장 좋은 방법 중 하나임
- 최대값을 설정하는 max, 현재 값을 설정하는 progress 속성이 중요함

- **막대 모양**

  - 작업의 진행 정도를 알려줄 수 있도록 막대 모양으로 표시함
  - style 속성의 값을 "?android:attr/progressBarStyleHorizontal"로 설정함

- **원 모양**

  - 작업이 진행 중임을 알려줌
  - 원 모양으로 된 프로그레스바가 반복적으로 표시됨

프로그레스바 사용 메소드

[Code]

void setProgress (int progress)
void incrementProgressBy (int diff)

- 진행률이 변경되면 progress 속성으로 설정되었던 값을 바꿈

[Code]

requestWindowFeature(Window.FEATURE_PROGRESS);

프로그레스바 사용하기 예제

프로그레스바 사용하기 예제

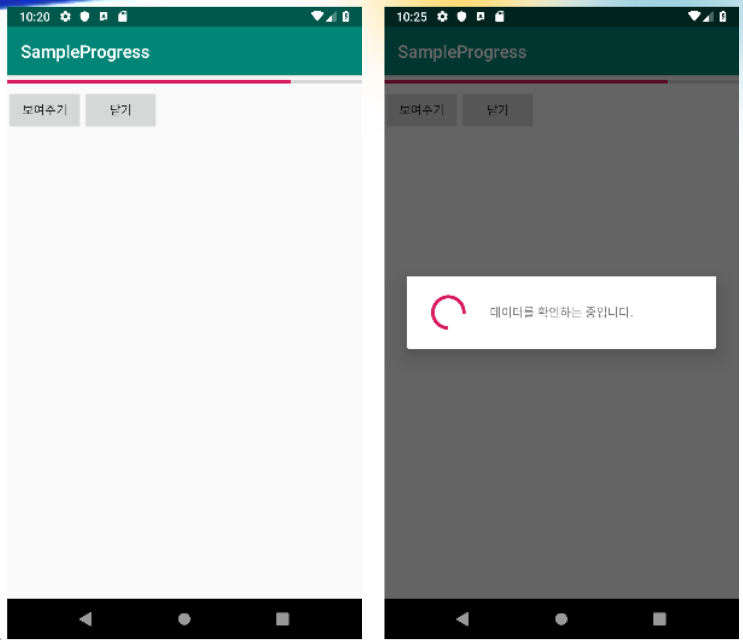
-프로그레스바로 진행상태 보여주기

메인 액티비티의 XML 레이아웃 정의

-메인 액티비티 레이아웃 정의

메인 액티비티 코드 작성

-프로그레스바 사용 코드 넣기



```
<ProgressBar
    style="?android:attr/progressBarStyleHorizontal"
    android:id="@+id/progressBar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:indeterminateOnly="false"
    android:minHeight="20dp"
    android:maxHeight="20dp"
    android:max="100"
    />
```

1 프로그레스바 정의

```
ProgressBar progressBar = (ProgressBar) findViewById(R.id.progressBar);
progressBar.setIndeterminate(false);
progressBar.setMax(100);
progressBar.setProgress(80);
```

```
progressDialog = new ProgressDialog(this);
progressDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
progressDialog.setMessage("데이터를 확인하는 중입니다.");
```