



안드로이드 앱 프로그래밍

Chapter 08

선택 위젯 만들기



이번 장에서는 무엇을 다룰까요?



아이콘이 들어간 리스트가 포함된 화면을 만들 때는 어떻게 하나요?



- 나인패치 이미지에 대해 알아보까요?
- 뷰를 직접 정의해볼까요?
- 리사이클러뷰가 포함된 화면을 만들어 볼까요?
- 콤보박스처럼 사용되는 스피너가 포함된 화면을 만들어 볼까요?





이번 장에서는 무엇을 다룰까요?



예쁜 모양의 버튼을 디자인해서 레이아웃에 넣고 싶은데요?

- 나인 패치 이미지 이해하기



제공되는 뷰 말고 직접 뷰를 정의할 수도 있나요?

- 새로운 뷰 만들기



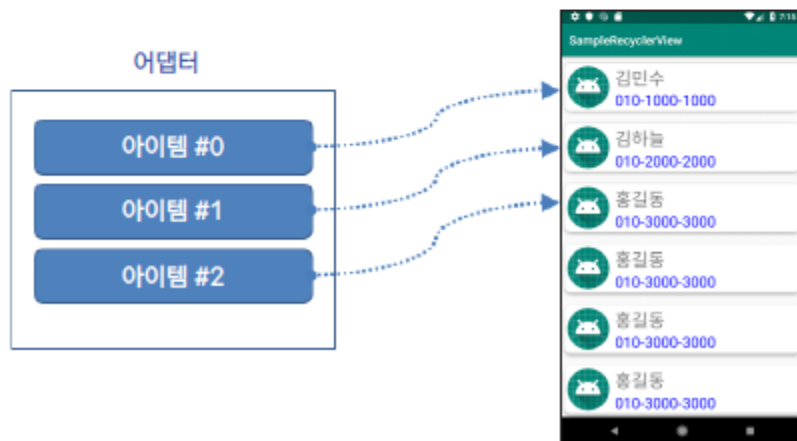
리스트 모양으로 보여주고 싶어요.

- 리사이클러뷰 사용하기



콤보박스 형태로 간단하게 보여줄 수 있나요?

- 스피너 사용하기





강의 주제

대표적인 선택 위젯의 이해와 실습



1

나인패치 이미지 알아보기

2

새로운 뷰 만들기

3

레이아웃 정의하고 카드뷰 넣기

4

리사이클러뷰 만들기

5

스피너 사용하기

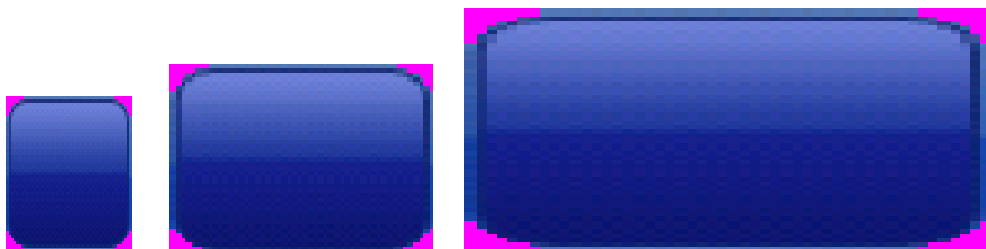
1.

나인패치 이미지 알아보기

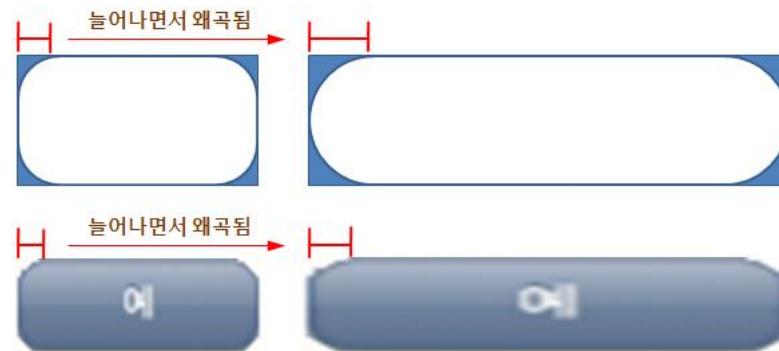


나인패치(Nine Patch) 이미지란?

- 이미지가 늘어나거나 줄어들 때 생기는 이미지 왜곡을 해결하는 방법을 정의한 것
- 서로 다른 해상도를 가진 여러 단말에 dp 단위로 뷰의 크기를 맞추다 보면 이미지 크기가 자동 조절되면서 왜곡되는 현상 발생 → 나인패치 이미지로 해결



[이미지의 크기에 따른 원본 이미지의 왜곡]



[이미지의 크기를 늘릴 때 왜곡되는 영역]



나인패치 이미지 예제

나인패치 이미지 예제

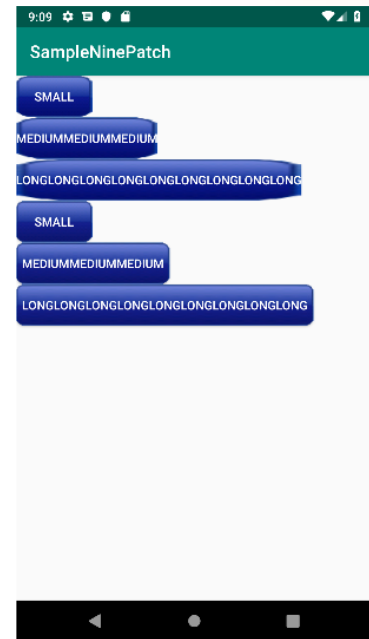
- 일반 이미지와 나인패치 이미지 비교
- 레이아웃에 여러 개의 버튼 추가

메인 액티비티의
XML 레이아웃

- 일반 버튼 추가

메인 액티비티의
XML 레이아웃

- 나인패치 버튼 추가





XML 레이아웃 조정

- 일반 이미지를 배경으로 사용하는 버튼과 나인패치 이미지를 배경으로 사용하는 버튼 추가

```
<Button
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="Small"
```

```
    android:textColor="#ffffffff"
```

```
    android:background="@drawable/button_image_01"
```

```
/>
```

1 일반 이미지를 배경으로 만든 작은 버튼 정의

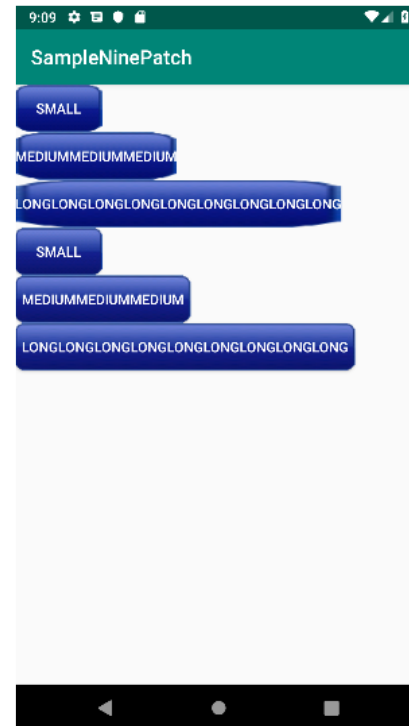


실행 화면

- 일반 이미지를 배경으로 사용하는 버튼은 모서리 부분의 이미지 깨짐 현상 발생
- 나인패치 이미지를 배경으로 사용하는 버튼은 크게 이상하게 보이지 않음

[Reference]

```
void setBackgroundColor (int color)
void setBackgroundDrawable (Drawable d)
void setBackgroundResource (int resid)
```



[일반 이미지와 나인패치 이미지를 적용한 버튼 모양 비교]

2.

새로운 뷰 만들기



뷰 만들기

- 새로운 뷰를 직접 만들 수 있음

- 뷰를 상속하면 새로운 뷰를 정의할 수 있음
- 뷰의 크기를 결정할 수도 있고 뷰 위에 그래픽을 그릴 수도 있음

[Reference]

```
public void onMeasure (int widthMeasureSpec, int heightMeasureSpec)  
public void onDraw(Canvas canvas)
```

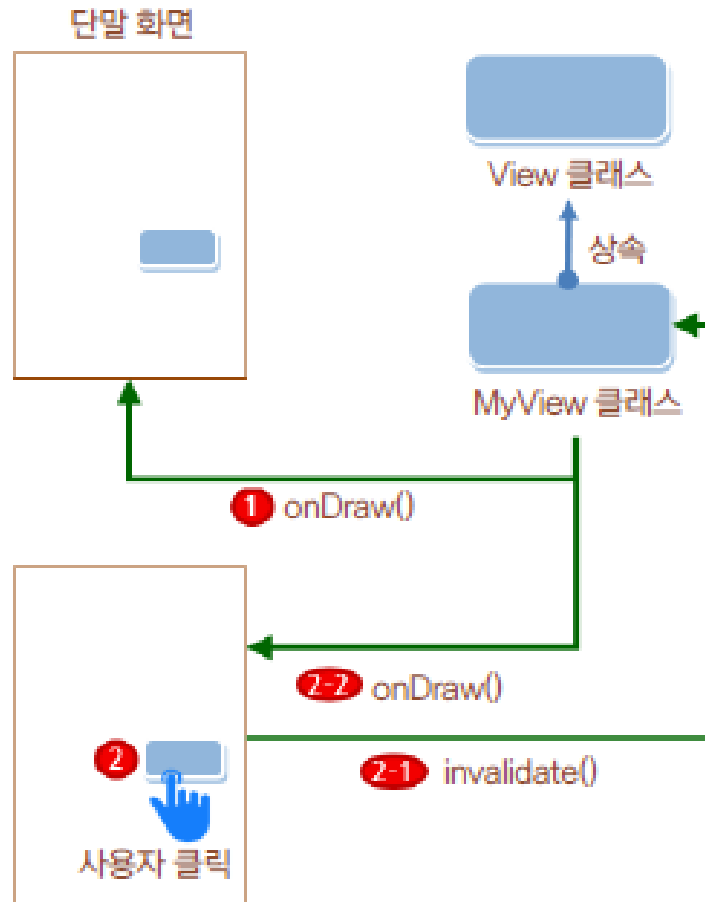
[Reference]

```
void setMeasuredDimension (int measuredWidth, int measuredHeight)
```



뷰 위에 그래픽을 그리는 과정

- 뷰에 그래픽이 그려질 때 onDraw() 메소드 호출됨
- 다시 그리기는 invalidate() 메소드 사용





버튼 만들기 예제

버튼 만들기 예제

-이미지를 보여주는 버튼

버튼을 상속한
새로운 클래스 정의

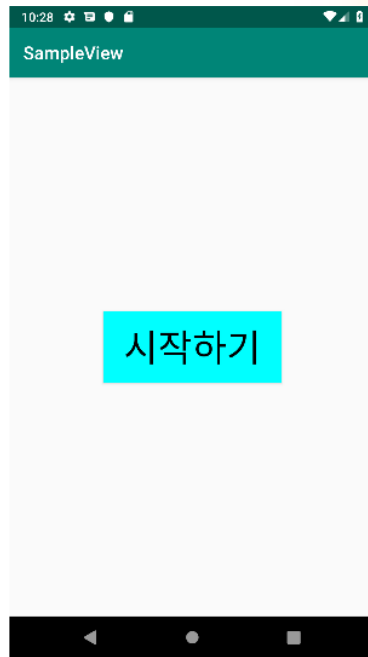
-새로운 버튼 클래스 정의

XML 레이아웃에 추가

-새로운 버튼 태그를 XML 레이아웃에 추가

메인 액티비티 코드 작성

-메인 액티비티 코드에서 참조하여 사용





새로운 버튼 클래스 정의

- 터치 이벤트에 따라 배경 이미지를 바꾸어주는 버튼 클래스 정의

참조파일 SampleView>/java/org.techtown.view/MyButton.java

```
public class MyButton extends AppCompatActivity { → ❶ AppCompatActivity 클래스 상속하여 새로운  
                                                    클래스 정의하기  
  
    public MyButton(Context context) {  
        super(context);  
        init(context);  
    }  
  
    public MyButton(Context context, AttributeSet attrs) {  
        super(context, attrs);  
        init(context);  
    }  
  
    private void init(Context context) {  
        setBackgroundColor(Color.CYAN);  
        setTextColor(Color.BLACK);  
  
        float textSize = getResources().getDimension(R.dimen.text_size);  
        setTextSize(textSize);  
    }  
}
```

❷ 초기화를 위한
메서드 정의하기



새로운 버튼 클래스 정의

- 터치 이벤트에 따라 배경 이미지를 바꾸어 줌

```
@Override
public boolean onTouchEvent(MotionEvent event) {
    Log.d("MyButton", "onTouchEvent 호출됨");

    int action = event.getAction();
    switch (action) {
        case MotionEvent.ACTION_DOWN:
            setBackgroundColor(Color.BLUE);
            setTextColor(Color.RED);

            break;

        case MotionEvent.ACTION_OUTSIDE:
        case MotionEvent.ACTION_CANCEL:
        case MotionEvent.ACTION_UP:
            setBackgroundColor(Color.CYAN);
            setTextColor(Color.BLACK);

            break;
    }

    invalidate();

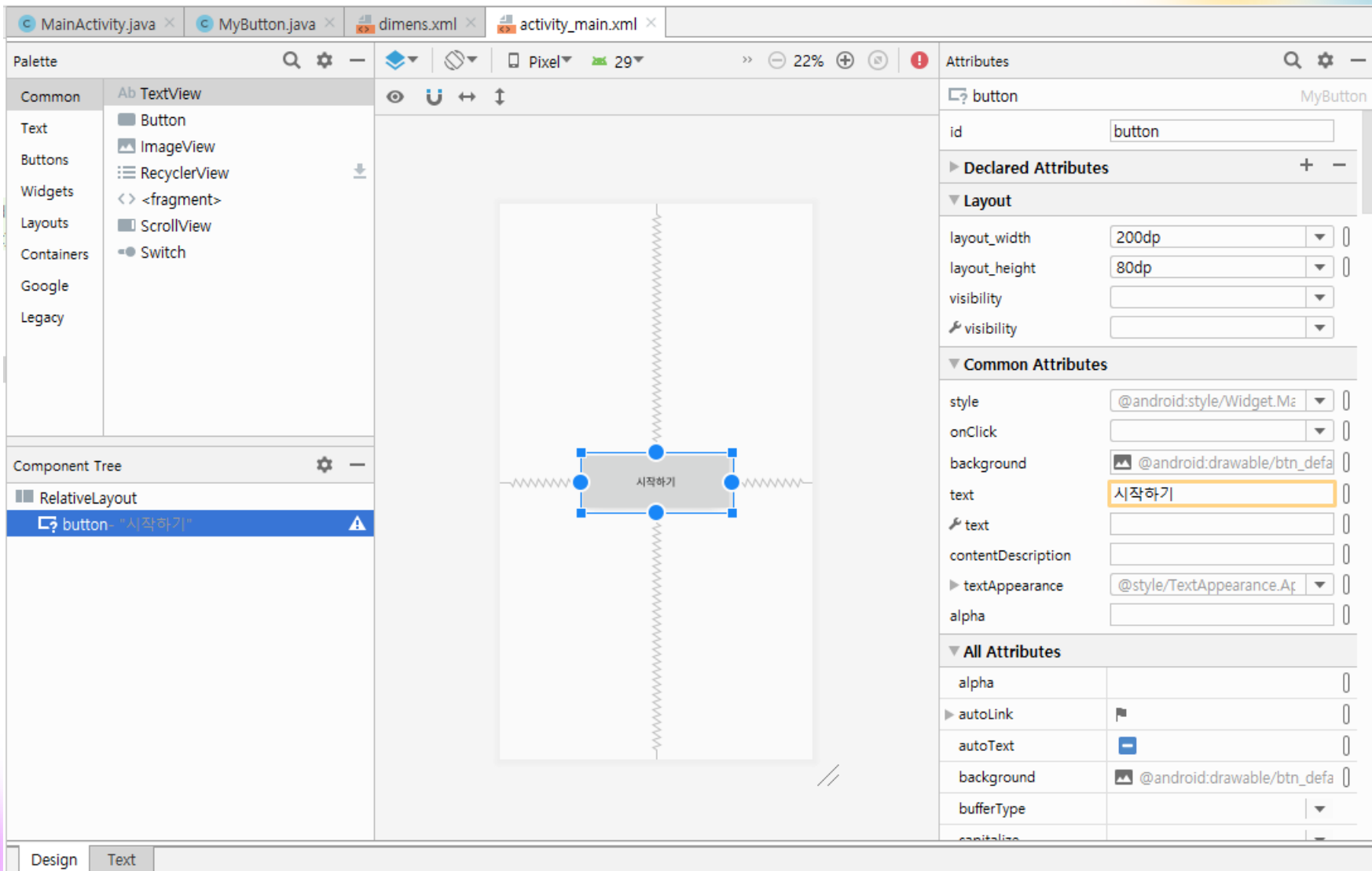
    return true;
}
```

2 뷰가 터치될 때 호출되는 함수에 기능 추가하기

중략...



레이아웃 만들기





레이아웃 만들기

참조파일 SampleView>/app/res/layout/activity_main.xml

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
```

```
    <org.techtown.view.MyButton
        android:id="@+id/button"
        android:layout_width="200dp"
        android:layout_height="80dp"
        android:layout_centerInParent="true"
        android:text="시작하기"
    />
```

새로 만든 MyButton 클래스를 태그로 추가하기

```
</RelativeLayout>
```



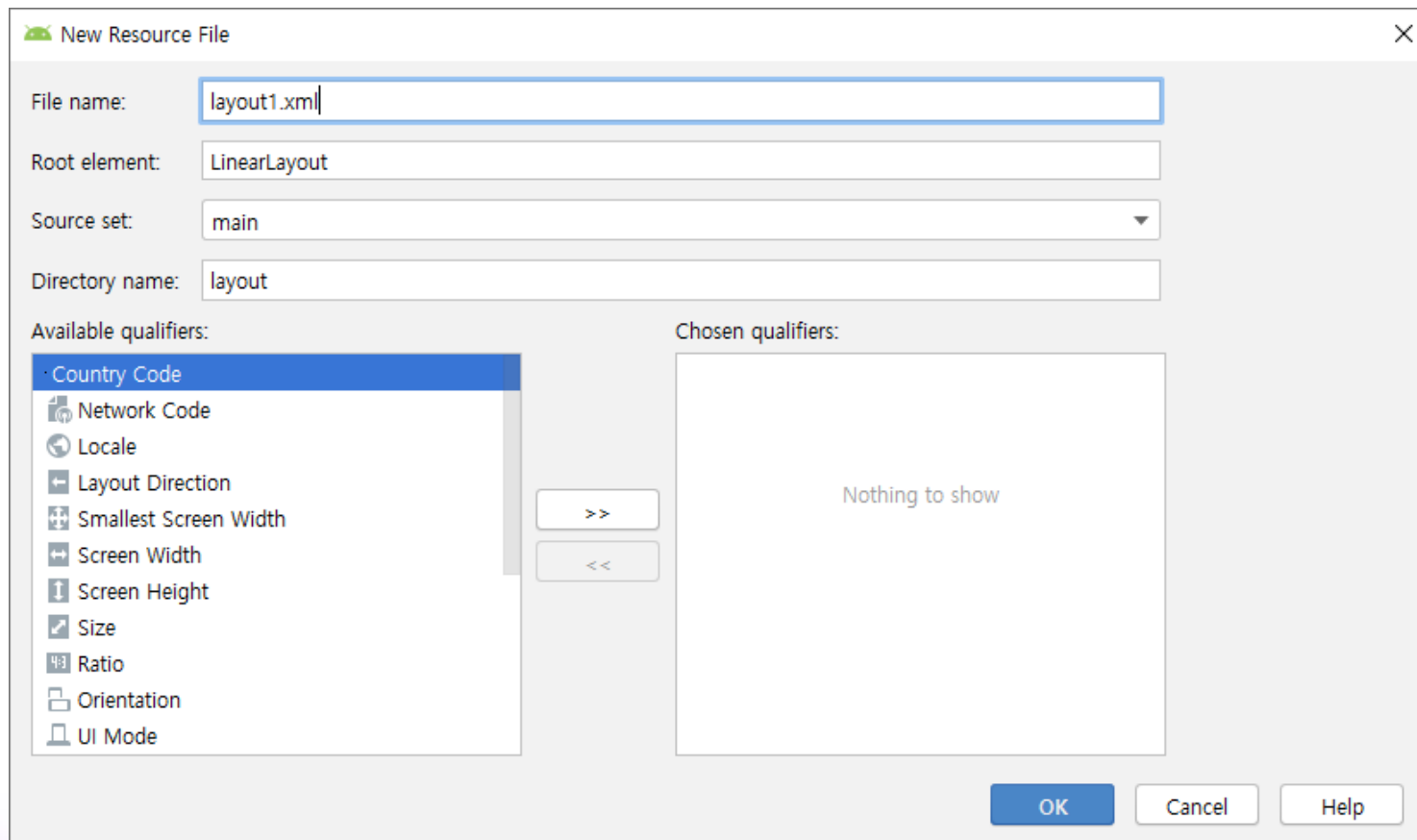
3.

레이아웃 정의하고 카드뷰 넣
기



레이아웃 파일 추가

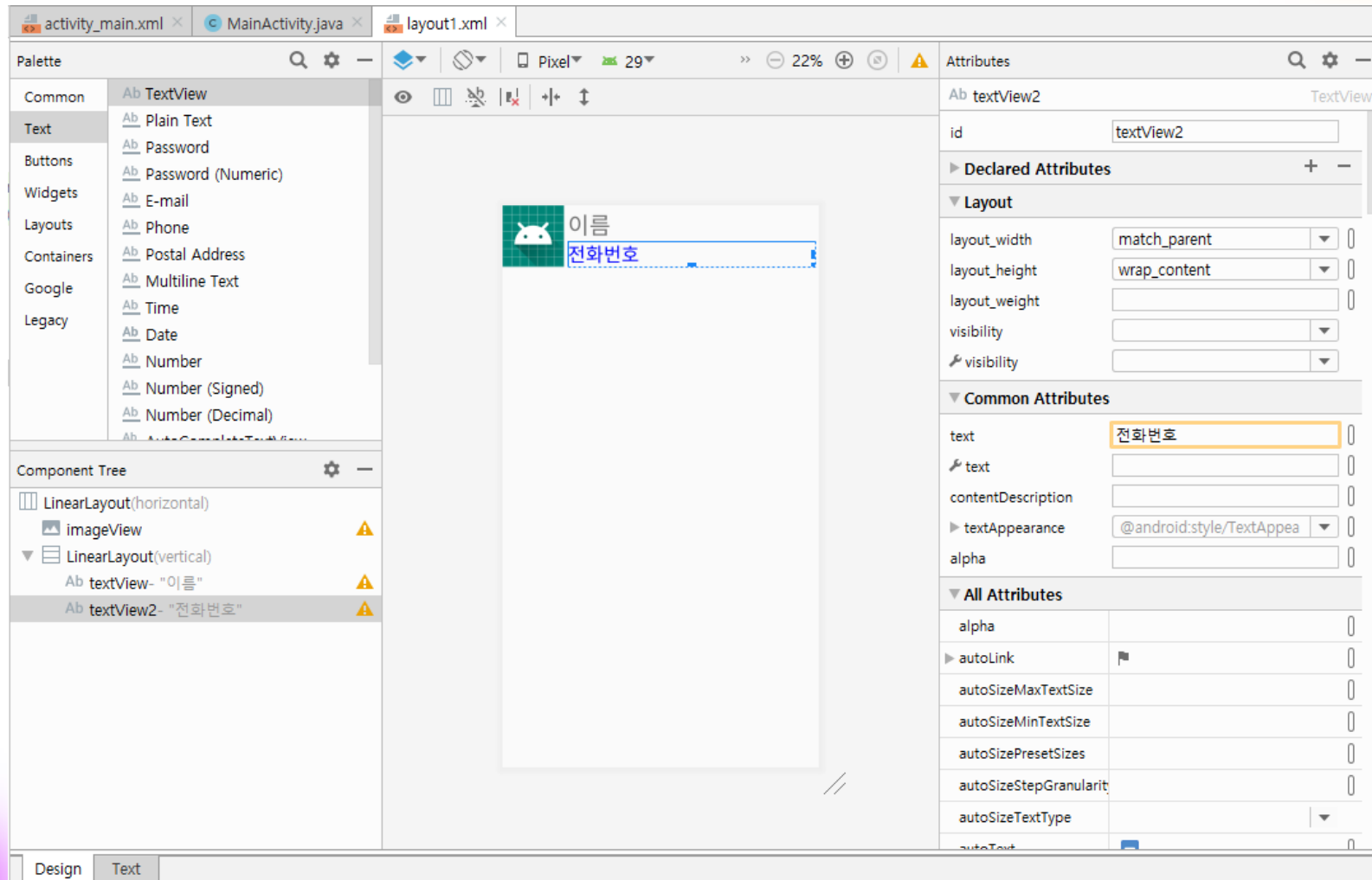
- /app/res/layout 폴더 안에 새로운 레이아웃 파일 만들기





레이아웃 파일 추가

- LinearLayout 안에 하나의 이미지뷰와 텍스트뷰 두 개 추가하기





레이아웃 파일 추가

- 새로운 클래스 파일 추가하기
- 생성자는 2개 재정의

Create New Class

Name:

Kind: ☒ Class

Superclass:

Interface(s):

Package:

Visibility: ☒ Public ☐ Package Private

Modifiers: ☒ None ☐ Abstract ☐ Final

☐ Show Select Overrides Dialog

OK Cancel Help

Choose Super Class Constructor

android.widget.LinearLayout

- ☒ LinearLayout(context:Context)
- ☐ LinearLayout(context:Context, attrs:AttributeSet)
- ☐ LinearLayout(context:Context, attrs:AttributeSet, defStyleAttr:int)
- ☐ LinearLayout(context:Context, attrs:AttributeSet, defStyleAttr:int, defStyleRes:int)

☐ Copy JavaDoc

OK Cancel



레이아웃 파일 추가

- 생성자 안에 인플레이션 진행하는 init 메서드 호출

참조파일 SampleLayout>/app/java/org.techtown.samplelayout/Layout1.java

```
public class Layout1 extends LinearLayout { ————— ❶ LinearLayout 클래스 상속하여 새로운 클래스
                                                         정의하기

    public Layout1(Context context) {
        super(context);
        init(context);
    }

    public Layout1(Context context, AttributeSet attrs) {
        super(context, attrs);
        init(context);
    }

    private void init(Context context) {
        LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        inflater.inflate(R.layout.layout1, this, true); ❷ 인플레이션 진행하기
    }
}
```



레이아웃 파일 추가

- XML 레이아웃 안에 들어있는 뷰 객체들을 찾아 변수에 할당

참조파일 SampleLayout>/app/java/org.techtown.samplelayout/Layout1.java

```
public class Layout1 extends LinearLayout {
    ImageView imageView;
    TextView textView;
    TextView textView2;
    중략...
    private void init(Context context) {
        LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_
SERVICE);
        inflater.inflate(R.layout.layout1, this, true);

        imageView = findViewById(R.id.imageView);
        textView = findViewById(R.id.textView);
        textView2 = findViewById(R.id.textView2);
    }

    public void setImage(int resId) {
        imageView.setImageResource(resId);
    }

    public void setName(String name) {
        textView.setText(name);
    }

    public void setMobile(String mobile) {
        textView2.setText(mobile);
    }
}
```

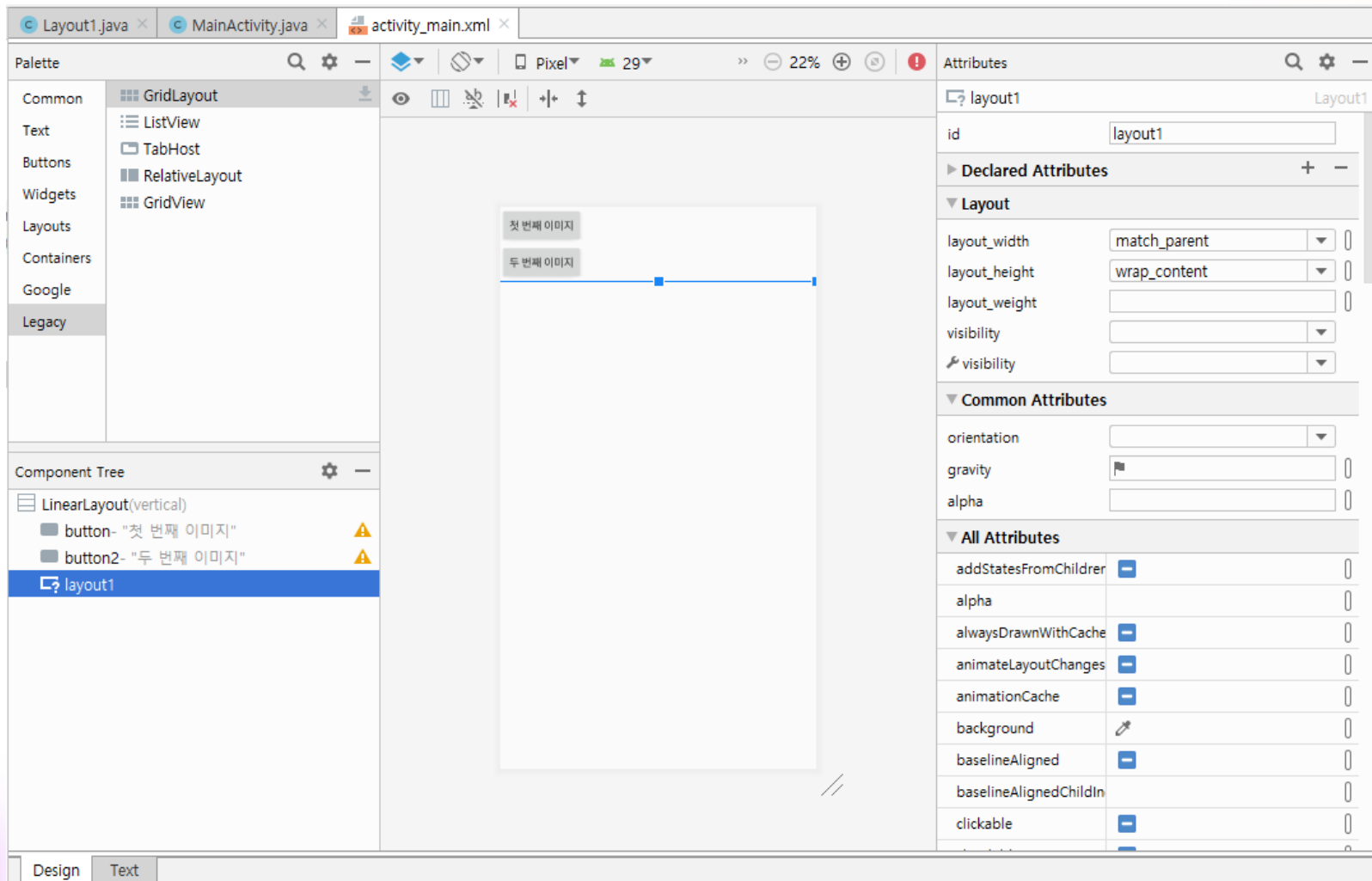
① XML 레이아웃에서 정의했던 뷰 참조하기

② 뷰에 데이터 설정하기



레이아웃 파일 추가

- activity_main.xml 파일에 새로 만든 뷰 추가하기





레이아웃 파일 추가

- 메인 액티비티 소스에서 데이터 설정

참조파일 SampleLayout>/app/java/org.techtown.samplelayout/MainActivity.java

```
public class MainActivity extends AppCompatActivity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

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

```
        Layout1 layout1 = findViewById(R.id.layout1);
```

→ ❶ XML 레이아웃에 추가한 뷰 참조하기

```
        layout1.setImage(R.drawable.ic_launcher_foreground);
```

```
        layout1.setName("김민수");
```

```
        layout1.setMobile("010-1000-1000");
```

```
    }
```

```
}
```

❷ 뷰의 메서드 호출하여 데이터 설정하기



레이아웃 파일 추가

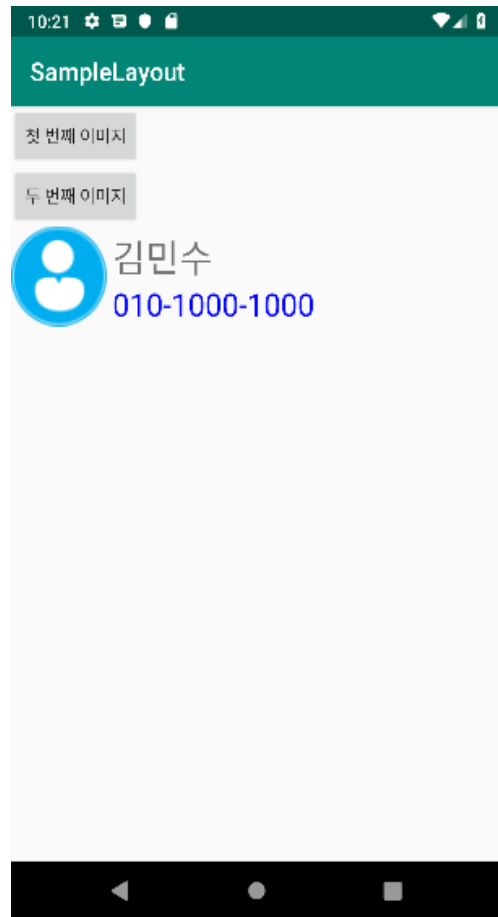
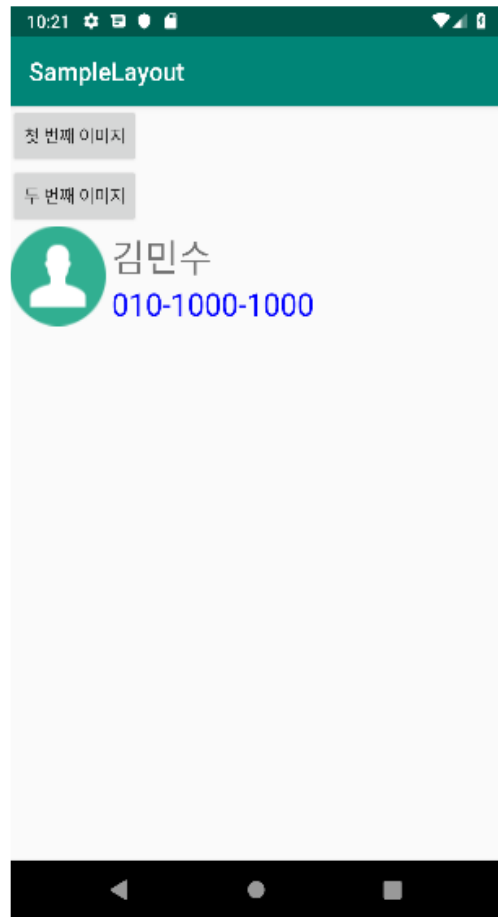
- 버튼 클릭 시의 이벤트 처리

```
Button button2 = findViewById(R.id.button2);
button2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        layout1.setImage(R.drawable.profile2);
    }
});
}
```



레이아웃 파일 추가

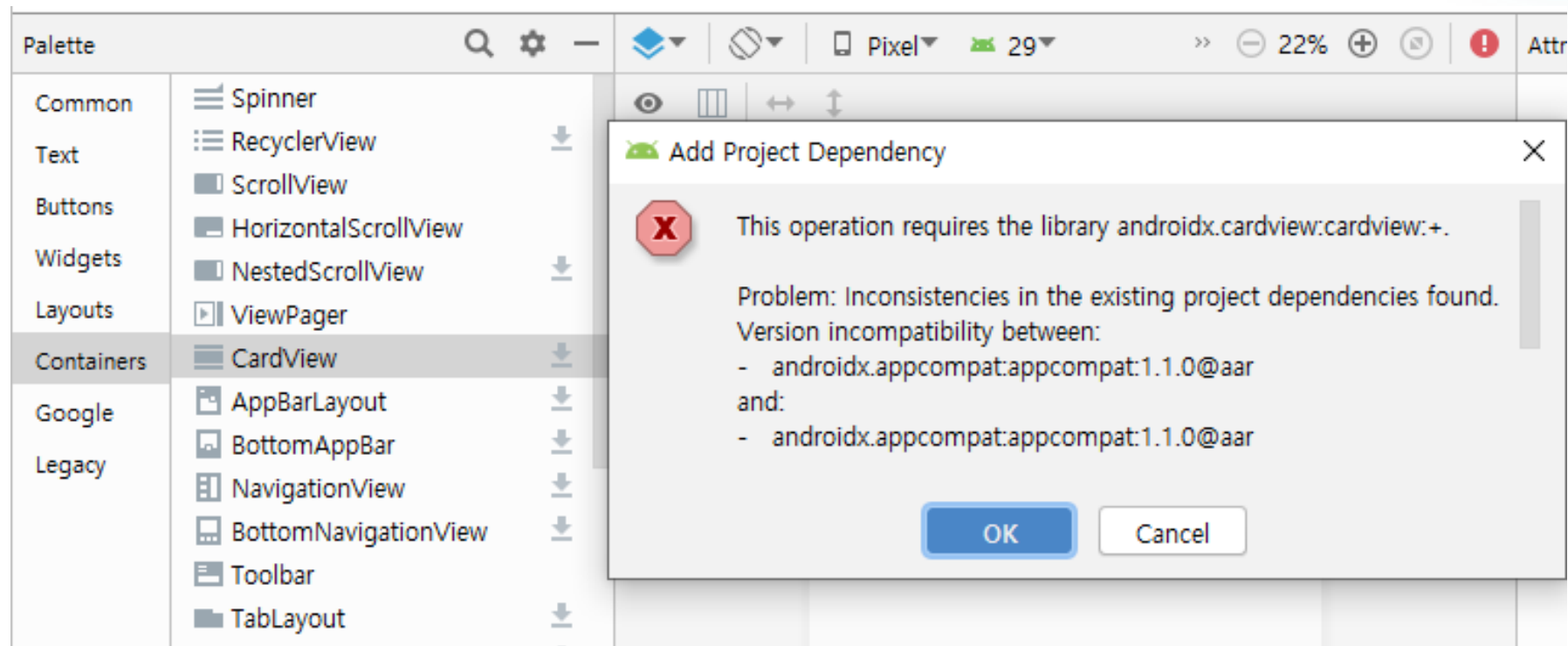
•앱 실행





카드뷰 모양으로 바꾸기

•카드뷰를 위한 외부 라이브러리 추가





카드뷰 모양으로 바꾸기

•카드뷰 태그 추가

참조파일 SampleLayout>/app/res/layout/layout1.xml

```
<?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"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

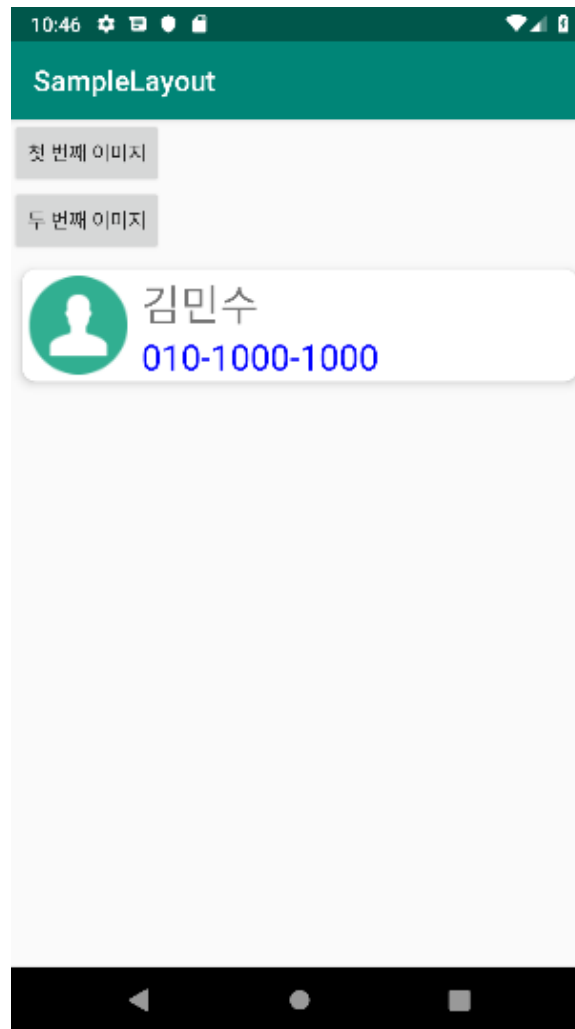
    <android.support.v7.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardBackgroundColor="#FFFFFFFF"
        app:cardCornerRadius="10dp"
        app:cardElevation="5dp"
        app:cardUseCompatPadding="true" >
```

CardView 태그 추가하기



카드뷰 모양으로 바꾸기

•카드뷰 모양으로 바꾼 결과



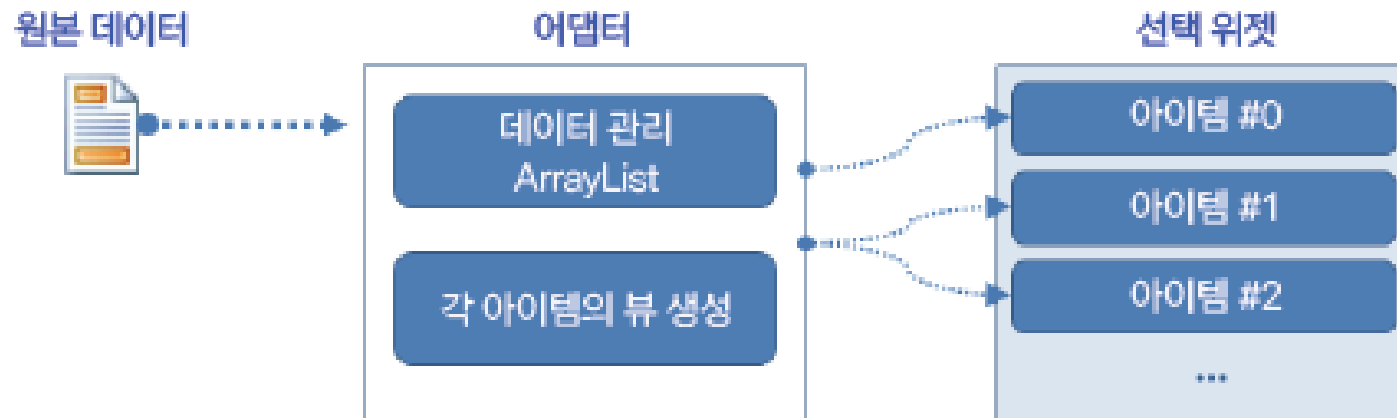
4.

리싸이클러뷰 만들기



왜 굳이 선택위젯이라는 이름으로 구분할까?

- 안드로이드에서는 여러 아이템 중의 하나를 선택하는 선택위젯은 별도의 패턴을 사용함
- 여러 개의 아이템 중에서 하나를 선택하는 방식의 선택 위젯은 어댑터를 사용하여야 함
- 이 어댑터에서 데이터를 관리하도록 해야 할 뿐만 아니라 화면에 보여지는 뷰도 어댑터의 getView() 메소드에서 결정함
- 선택위젯의 가장 큰 특징은 원본 데이터를 위젯에 직접 설정하지 않고 어댑터라는 클래스를 사용하도록 되어 있다는 점으로 이 패턴을 잘 기억해 두어야 함

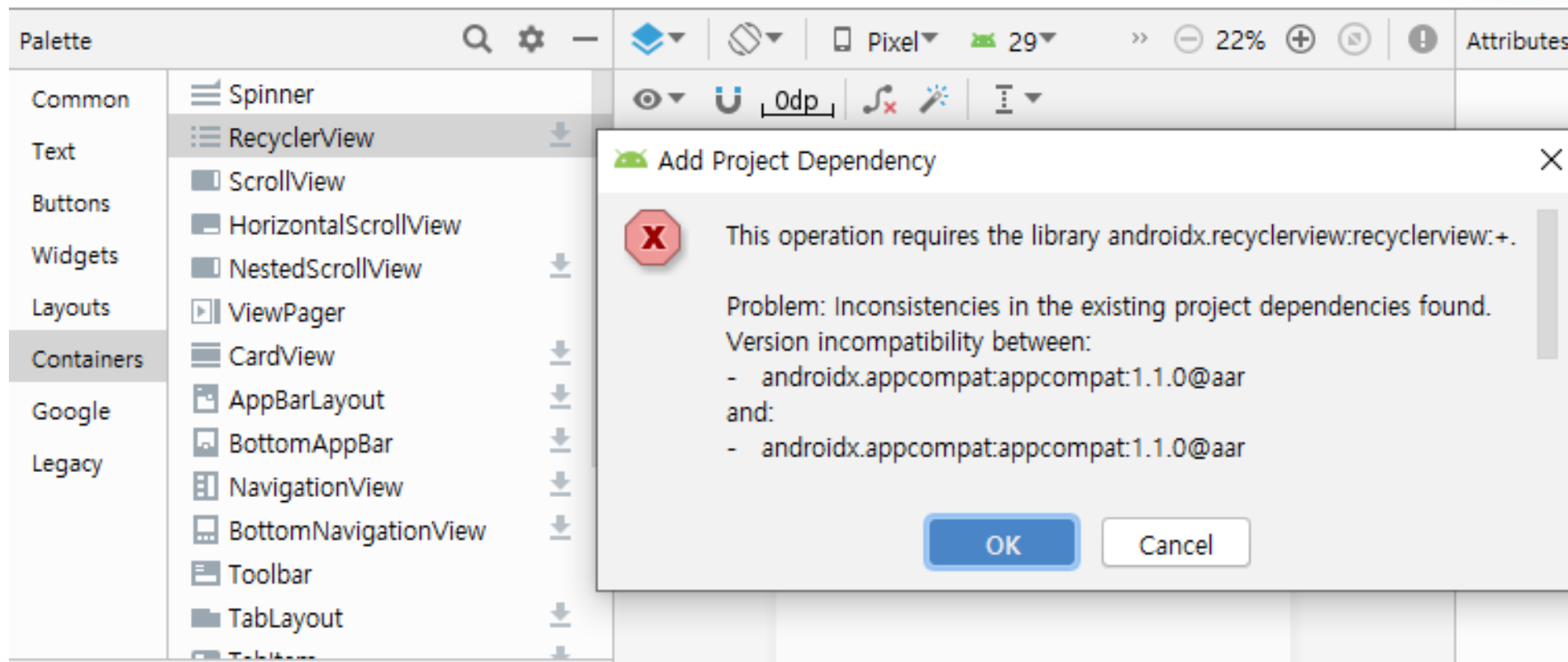


[선택 위젯과 어댑터]



리사이클러뷰 만들기

- 팔레트에서 RecyclerView 오른쪽에 있는 아이콘 눌러 외부 라이브러리 추가





리사이클러뷰 만들기

- activity_main.xml 에 RecyclerView 태그 추가

참조파일 SampleRecyclerView>/app/res/layout/activity_main.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"

    <android.support.v7.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
```

리사이클러뷰 태그 추가하기

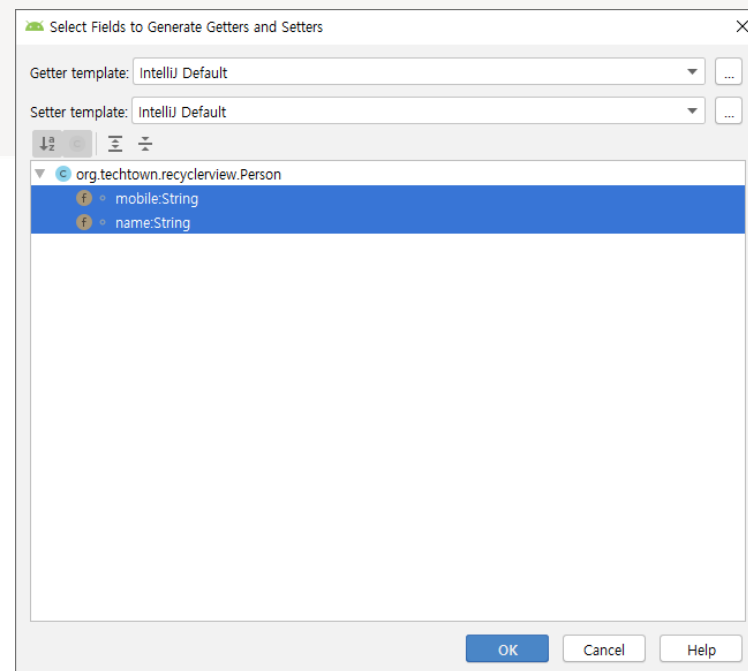
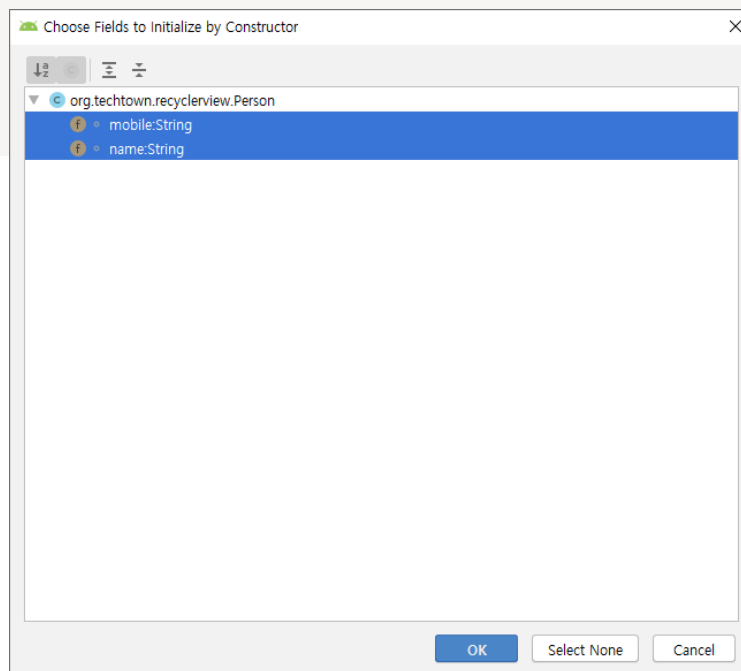


리사이클러뷰 만들기

- 각 아이템을 위한 데이터를 담아두기 위해 Person 클래스 추가
- 생성자와 get, set 메서드 추가

참조파일 SampleRecyclerView>/app/java/org.techtown.samplerectyclerview/Person.java

```
public class Person {  
    String name;  
    String mobile;  
}
```





리사이클러뷰 만들기

•어댑터 소스 추가하고 ViewHolder 정의

참조파일 SampleRecyclerView>/app/java/org.techtown.samplerectyclerview/PersonAdapter.java

```
public class PersonAdapter {  
  
    static class ViewHolder extends RecyclerView.ViewHolder {  
        TextView textView;  
        TextView textView2;  
  
        public ViewHolder(View itemView) {  
            super(itemView);  
  
            textView = itemView.findViewById(R.id.textView);  
            textView2 = itemView.findViewById(R.id.textView2);  
        }  
  
        public void setItem(Person item) {  
            textView.setText(item.getName());  
            textView2.setText(item.getMobile());  
        }  
    }  
}
```

1 뷰홀더 생성자로 전달되는 뷰 객체 참조하기

2 뷰 객체에 들어 있는 텍스트뷰 참조하기



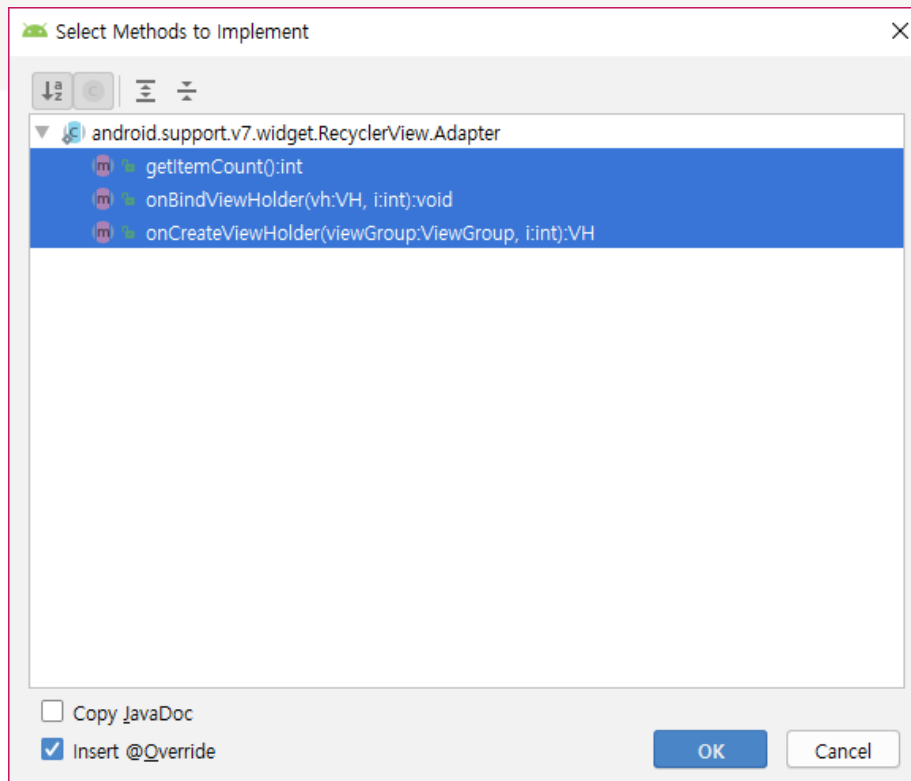
리사이클러뷰 만들기

- PersonAdapter가 상속할 클래스 지정

참조파일 SampleRecyclerView>/app/java/org.techtown.samplerectyclerview/PersonAdapter.java

```
public class PersonAdapter extends RecyclerView.Adapter<PersonAdapter.ViewHolder> {
```

중략...





리사이클러뷰 만들기

•onCreateViewHolder와 onBindViewHolder 메서드 재정의

참조파일 SampleRecyclerView>/app/java/org.techtown.samplerectyclerview/PersonAdapter.java

```
public class PersonAdapter extends RecyclerView.Adapter<PersonAdapter.ViewHolder> {
    ArrayList<Person> items = new ArrayList<Person>();

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int viewType) {
        LayoutInflater inflater = LayoutInflater.from(viewGroup.getContext());
        View itemView = inflater.inflate(R.layout.person_item, viewGroup, false); —→ ❶ 인플레이션을
        return new ViewHolder(itemView); —→ ❷ 뷰홀더 객체를 생성하면서 뷰 객체를 통해 뷰 객체
    }                                     전달하고 그 뷰홀더 객체를 반환하기 만들기

    @Override
    public void onBindViewHolder(@NonNull ViewHolder viewHolder, int position) {
        Person item = items.get(position);
        viewHolder.setItem(item);
    }

    @Override
    public int getItemCount() {
        return items.size();
    }
}
```

중략...



리사이클러뷰 만들기

•person_item.xml 파일 정의

참조파일 SampleRecyclerView>/app/res/layout/person_item.xml

```
<?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"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <android.support.v7.widget.CardView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:cardBackgroundColor="#FFFFFF"
        app:cardCornerRadius="10dp"
        app:cardElevation="5dp"
        app:cardUseCompatPadding="true" >

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:orientation="horizontal">

            <ImageView
                android:id="@+id/imageView"
                android:layout_width="80dp"
                android:layout_height="80dp"
                android:padding="5dp"
                app:srcCompat="@mipmap/ic_launcher" />
```




리사이클러뷰 만들기

- add, set, get 등의 메서드 추가

```
public void addItem(Person item) {  
    items.add(item);  
}  
  
public void setItems(ArrayList<Person> items) {  
    this.items = items;  
}  
  
public Person getItem(int position) {  
    return items.get(position);  
}  
  
public void setItem(int position, Person item) {  
    return items.set(position, item);  
}
```

중략...



리사이클러뷰 만들기

- MainActivity의 onCreate 메서드 안에 리사이클러뷰와 어댑터를 위한 코드 추가

참조파일 SampleRecyclerView>/app/java/org.techtown.samplerectyclerview/MainActivity.java

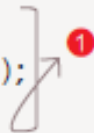
```
public class MainActivity extends AppCompatActivity {

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

        RecyclerView recyclerView = findViewById(R.id.recyclerView);

        LinearLayoutManager layoutManager =
            new LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false);
        recyclerView.setLayoutManager(layoutManager);
        PersonAdapter adapter = new PersonAdapter();

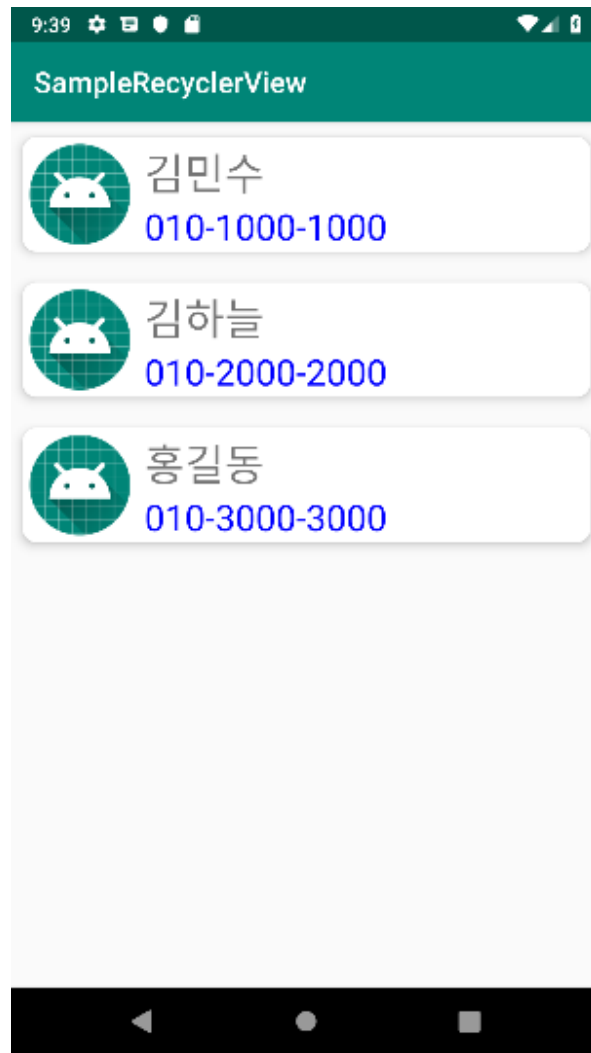
        adapter.addItem(new Person("김민수", "010-1000-1000"));
        adapter.addItem(new Person("김하늘", "010-2000-2000"));
        adapter.addItem(new Person("홍길동", "010-3000-3000"));
```





리싸이클러뷰 만들기

•앱 실행





리사이클러뷰 만들기

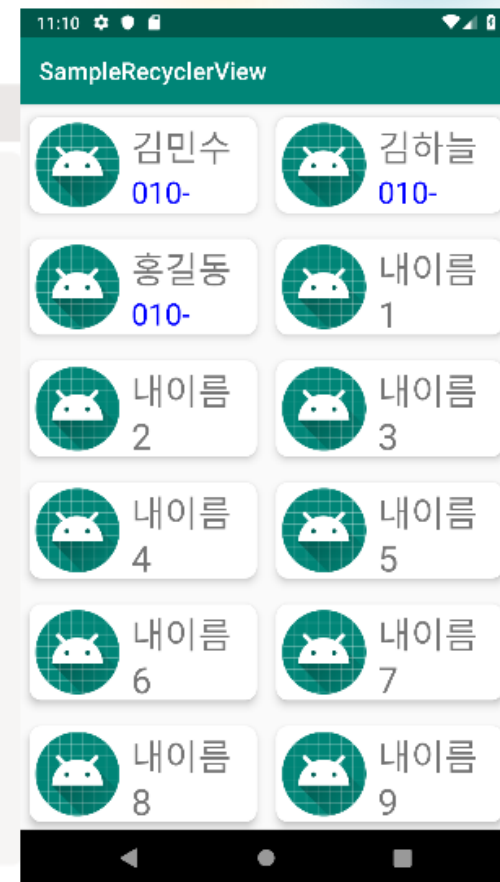
•여러 칼럼의 뷰로 보여주기

참조파일 SampleRecyclerView2>/app/java/org.techtown.recyclerview/MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        RecyclerView recyclerView = findViewById(R.id.recyclerView);  
  
        GridLayoutManager layoutManager = new GridLayoutManager(this, 2);  
        recyclerView.setLayoutManager(layoutManager);  
    }  
}
```

중략...

리사이클러뷰에 GridLayoutManager를 레이아웃 매니저로 설정하기





리사이클러뷰 만들기

•클릭했을 때 이벤트 처리

참조파일 SampleRecyclerView

중략...

```
static class ViewHolder extends RecyclerView.ViewHolder {
    TextView textView;
    TextView textView2;

    public ViewHolder(View itemView, final OnPersonItemClickListener listener) {
        super(itemView);

        textView = itemView.findViewById(R.id.textView);
        textView2 = itemView.findViewById(R.id.textView2);

        itemView.setOnClickListener(new View.OnClickListener() { → ❶ 아이템 뷰에 OnClickListener
            @Override
            public void onClick(View view) {
                int position = getAdapterPosition();

                if (listener != null) {
                    listener.onItemClick(ViewHolder.this, view, position); } ❷ 아이템 뷰 클릭 시
                }
            }
        });
    }
```

참조파일 SampleRecyclerView2>/app/java/org.techtown.recyclerview/OnPersonItemClickListener.java

```
public interface OnPersonItemClickListener {
    public void onItemClick(PersonAdapter.ViewHolder holder, View view, int position);
}
```



리사이클러뷰 만들기

- 어댑터에 리스너를 위한 변수 선언

참조파일 SampleRecyclerView2>/app/java/org.techtown.recyclerview/PersonAdapter.java

중략...

```
public class PersonAdapter extends RecyclerView.Adapter<PersonAdapter.ViewHolder>
    implements OnPersonItemClickListener {
    ArrayList<Person> items = new ArrayList<Person>();
    OnPersonItemClickListener listener;

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int viewType) {
        LayoutInflater inflater = LayoutInflater.from(viewGroup.getContext());
        View itemView = inflater.inflate(R.layout.person_item, viewGroup, false);

        return new ViewHolder(itemView, this);
    }
}
```

중략...



리사이클러뷰 만들기

- 어댑터에 리스너를 위한 변수 선언

```
public void setOnItemClickListener(OnPersonItemClickListener listener) {  
    this.listener = listener;  
}  
  
@Override  
public void onItemClick(ViewHolder holder, View view, int position) {  
    if (listener != null) {  
        listener.onItemClick(holder, view, position);  
    }  
}
```

종락...



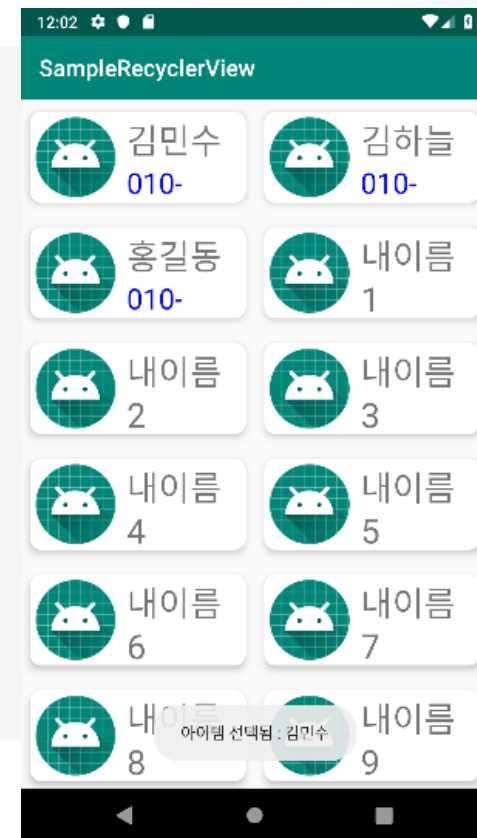
리사이클러뷰 만들기

•메인 액티비티에 추가

```
recyclerView.setAdapter(adapter);

adapter.setOnItemClickListener(new OnPersonItemClickListener() { → ❶
    @Override
    public void onItemClick(PersonAdapter.ViewHolder holder, View view, int position) {

        Person item = adapter.getItem(position); → ❷
        Toast.makeText(getApplicationContext(), "아이템 선택됨: " + item.getName(),
            Toast.LENGTH_LONG).show();
    }
});
}
```



5.

스피너 사용하기



스피너 사용하기 예제

스피너 사용하기 예제

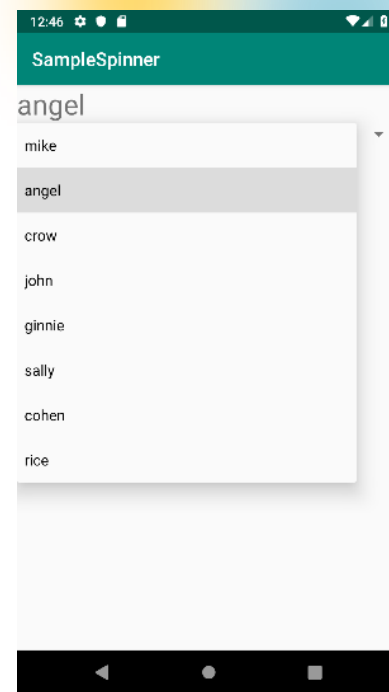
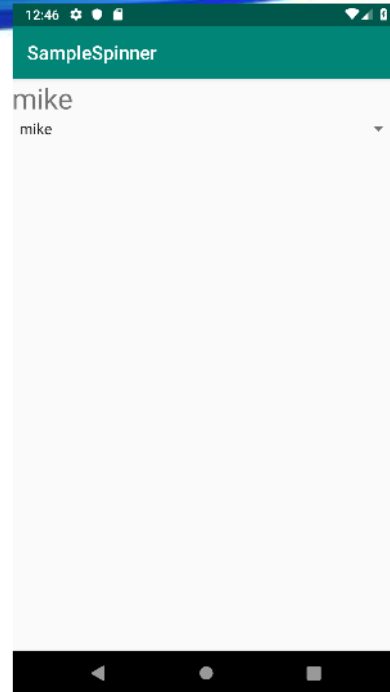
- 콤보박스처럼 선택할 수 있는 스피너 사용
- XML 레이아웃에 정의한 스피너 참조

메인 액티비티의 XML 레이아웃

- 스피너를 포함하는 메인 화면의 XML 레이아웃 정의

메인 액티비티 코드 작성

- 스피너 객체를 참조하여 설정





XML 레이아웃 만들기

- <Spinner> 태그를 사용하여 레이아웃에 추가

```
<Spinner  
    android:id="@+id/spinner"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
>
```

1

스피너 정의



메인 액티비티 코드 만들기

- 안드로이드에서 미리 만들어 제공하는 어댑터를 사용할 수 있음

```
String[] items = { "mike", "angel", "crow", "john", "ginnie", "sally", "cohen", "rice" };
```

1 스피너에 표시될 아이템들의 데이터를 배열로 정의

```
selection = (TextView) findViewById(R.id.selection);  
Spinner spinner = (Spinner) findViewById(R.id.spinner);  
spinner.setOnItemSelectedListener(this);
```

Continued..



메인 액티비티 코드 만들기 (계속)

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(  
    this, android.R.layout.simple_spinner_item, items);  
adapter.setDropDownViewResource(  
    android.R.layout.simple_spinner_dropdown_item);
```

2 ArrayAdapter를 이용해 어댑터 객체 생성

```
spinner.setAdapter(adapter);  
}
```

3 스피너에 어댑터 설정

```
public void onItemSelected(AdapterView<?> parent, View v, int position, long id) {  
    selection.setText(items[position]);  
}  
public void onNothingSelected(AdapterView<?> parent) {  
    selection.setText("");  
}  
}
```

4 스피너의 아이템이 선택되었을 때 처리하는 메소드 정의



ArrayAdapter의 사용

[Reference]

```
public ArrayAdapter (Context context, int textViewResourceId, T[] objects)
```

- 첫 번째 파라미터는 Context 객체이므로 액티비티인 this를 전달하면 됨
- 두 번째 파라미터는 뷰를 초기화할 때 사용되는 XML 레이아웃의 리소스 ID 값으로 이 코드에서는 android.R.layout.simple_spinner_item을 전달하였음
- 세 번째 파라미터는 아이템으로 보일 문자열 데이터들의 배열임

