

Chapter12-2 어댑터뷰



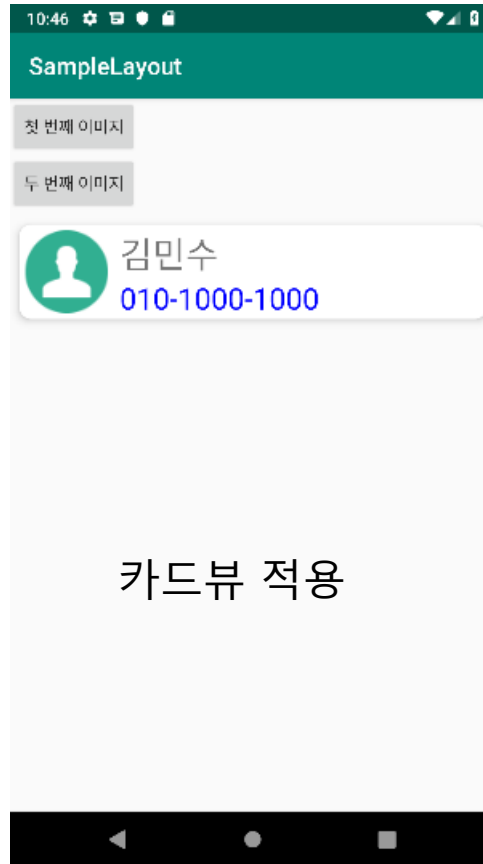
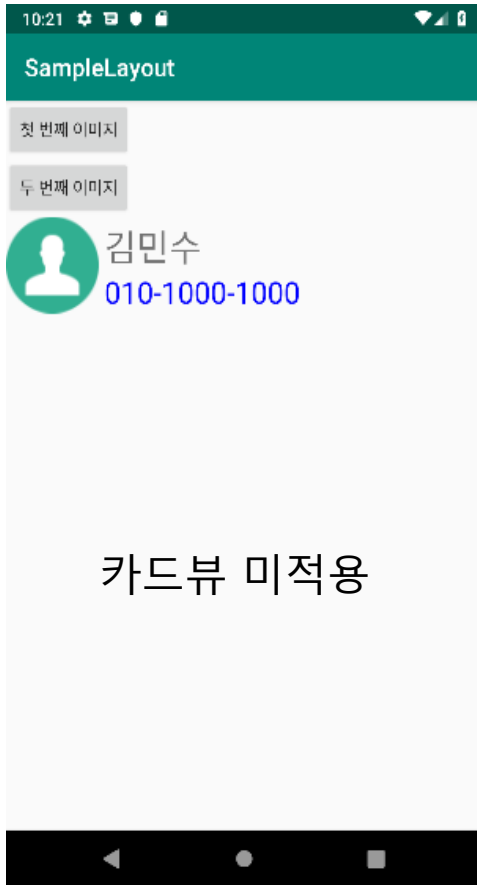
차례

- 카드뷰
- 리사이클 뷰

1. 카드뷰

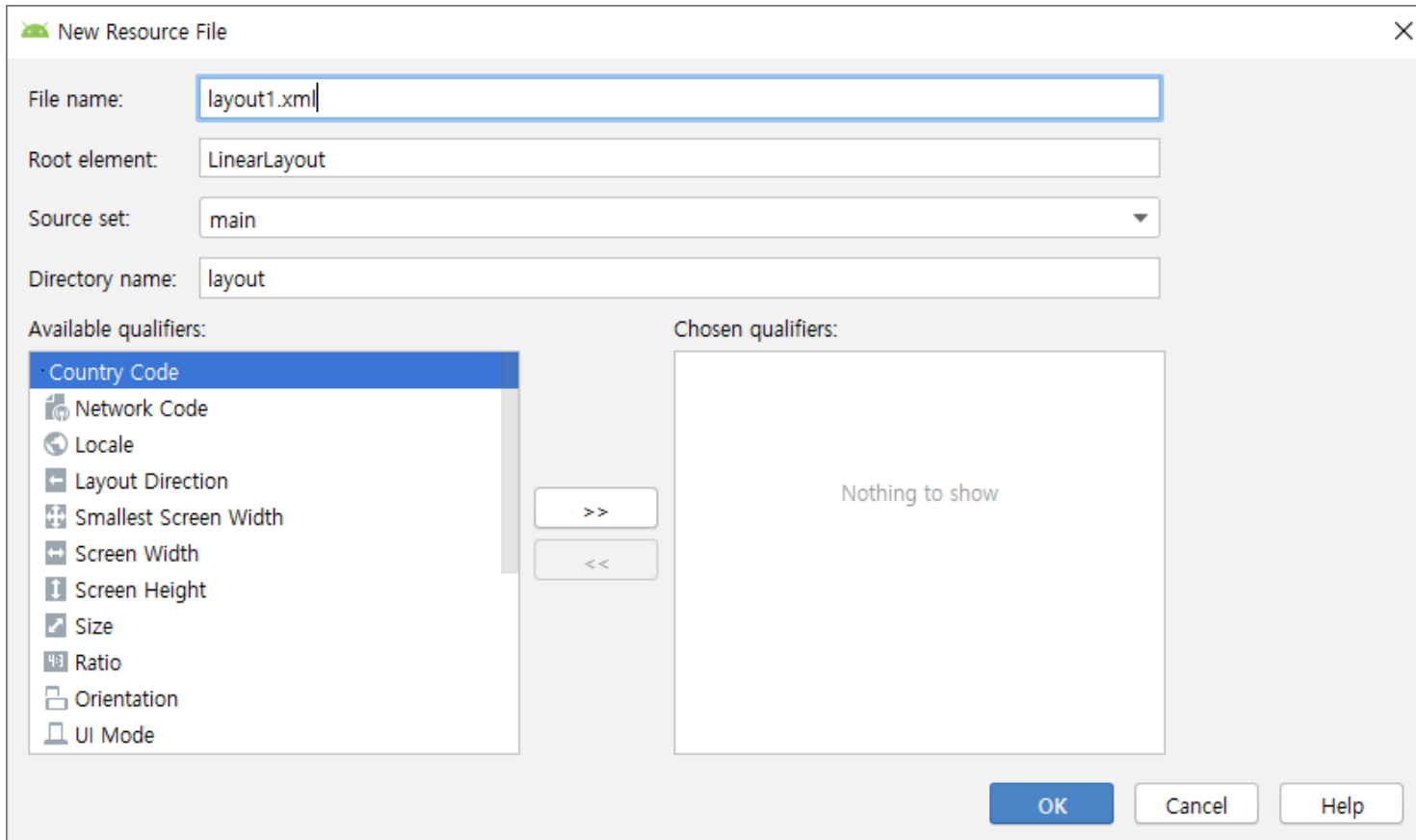
■ 카드뷰(CardView)

- 프로필 같은 간단한 정보를 넣기 위해 각 영역을 구분하는 역할



레이아웃 파일 추가

- /app/res/layout 폴더 안에 새로운 레이아웃 파일(layout1.xml) 만들기



레이아웃 파일 추가

- LinearLayout 안에 하나의 이미지뷰와 텍스트뷰 두 개 추가하기

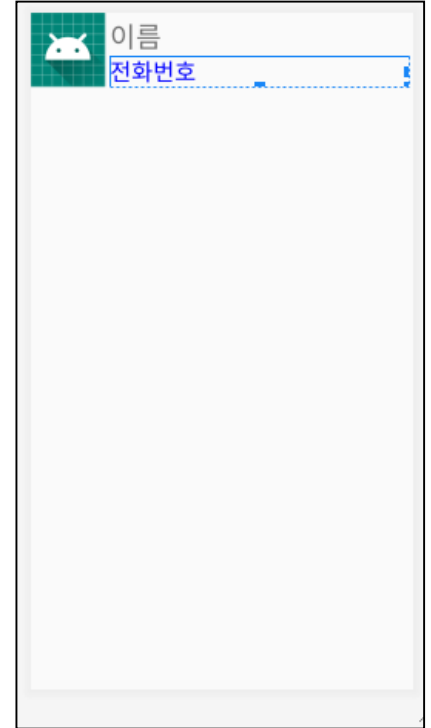
```
<LinearLayout>
  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal">

    <ImageView
      android:id="@+id/imageView"
      android:layout_width="80dp"
      android:layout_height="80dp"
      android:layout_margin="5dp"
      app:srcCompat="@drawable/profile1" />

    <LinearLayout
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:layout_margin="5dp"
      android:layout_weight="1"
      android:orientation="vertical">

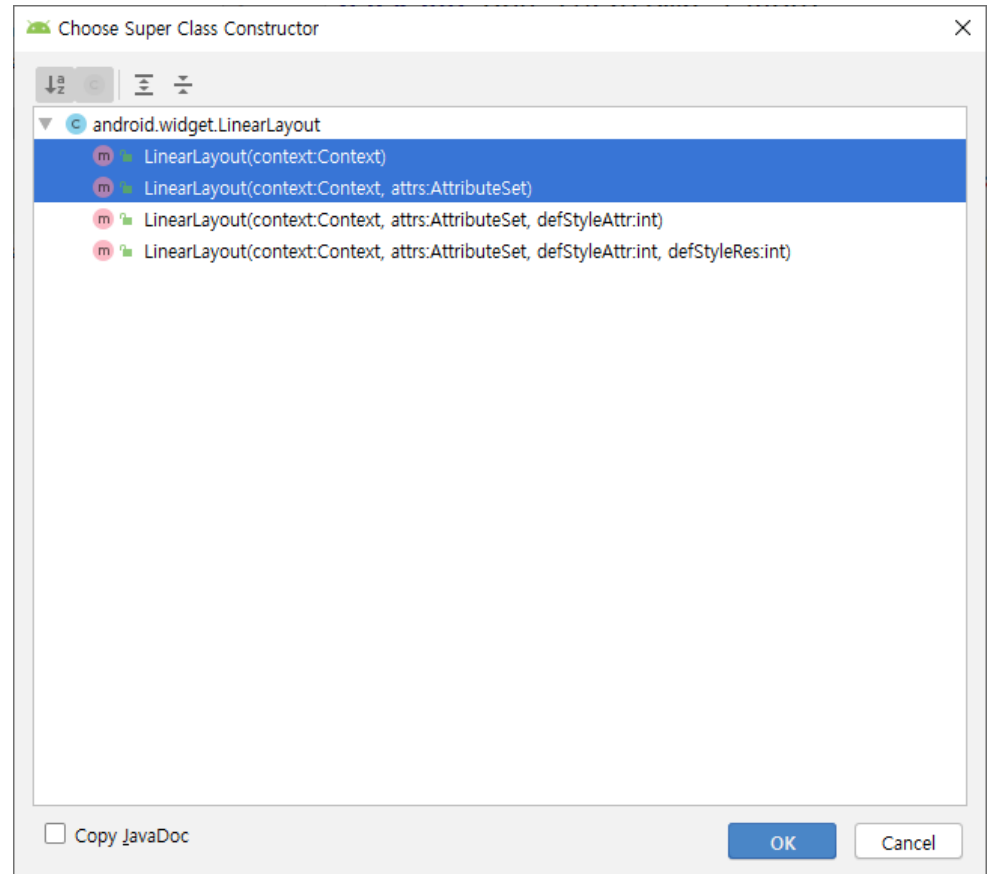
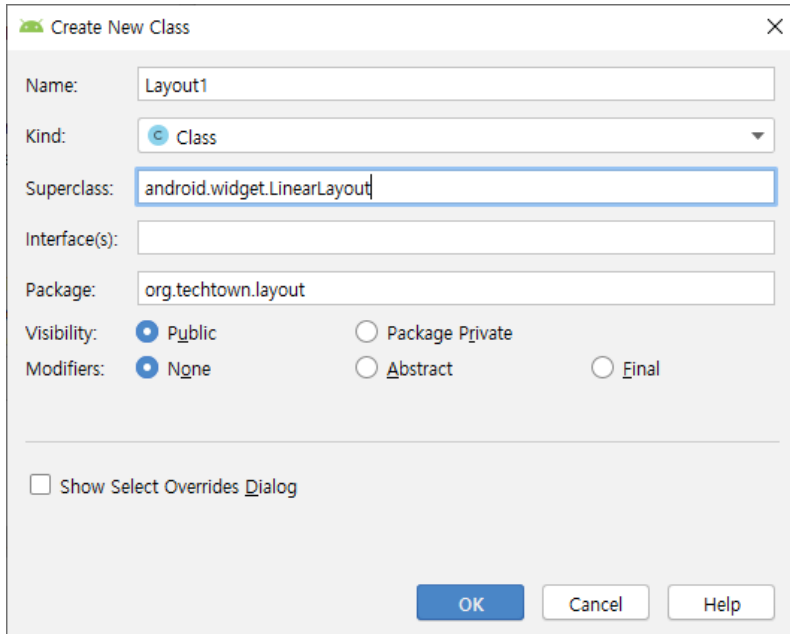
      <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="이름"
        android:textSize="30sp" />

      <TextView
        android:id="@+id/textView2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="전화번호"
        android:textColor="#FF0000FF"
        android:textSize="25sp" />
    </LinearLayout>
  </LinearLayout>
</LinearLayout>
```



레이아웃 파일 추가

- 새로운 클래스 파일 추가하기
- 생성자는 2개 재정의



레이아웃 파일 추가

■ 생성자 안에 인플레이션 진행하는 init 메서드 호출

```
public class Layout1 extends LinearLayout {
    ImageView imageView;
    TextView textView1;
    TextView textView2;

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

    public Layout1(Context context, @Nullable 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.layout, this, true);
    }
}
```

레이아웃 파일 추가

- XML 레이아웃 안에 들어있는 뷰 객체들을 찾아 변수에 할당

```
private void init(Context context){
    ...

    imageView = findViewById(R.id.imageView);
    textView1 = findViewById(R.id.textView);
    textView2 = findViewById(R.id.textView2);
}

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

public void setName(String name) {
    textView1.setText(name);
}

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


레이아웃 파일 추가

■ activity_main.xml 파일에 새로 만든 뷰 추가하기

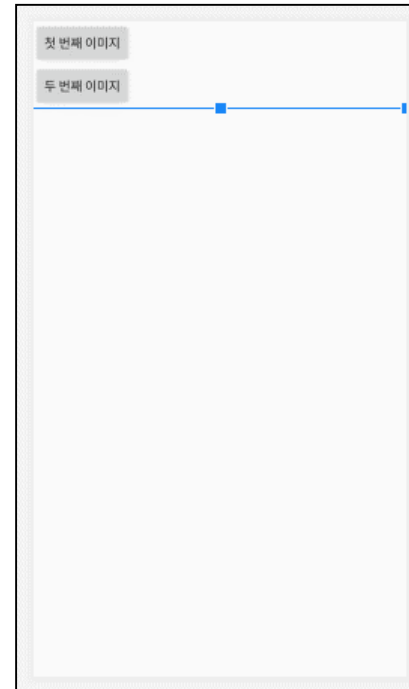
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity" >
```

```
    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="첫 번째 이미지" />
```

```
    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="두 번째 이미지" />
```

```
    <com.example.samplelayoutapplication.Layout1
        android:id="@+id/layout1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal" />
```

```
</LinearLayout>
```



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);  
  
        layout1.setImage(R.drawable.ic_launcher_foreground);  
        layout1.setName("김민수");  
        layout1.setMobile("010-1000-1000");  
    }  
}
```

① XML 레이아웃에 추가한 뷰 참조하기

② 뷰의 메서드 호출하여 데이터 설정하기

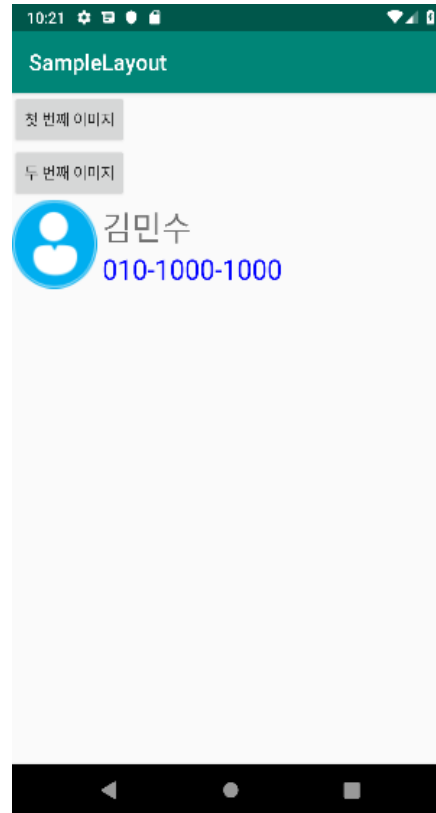
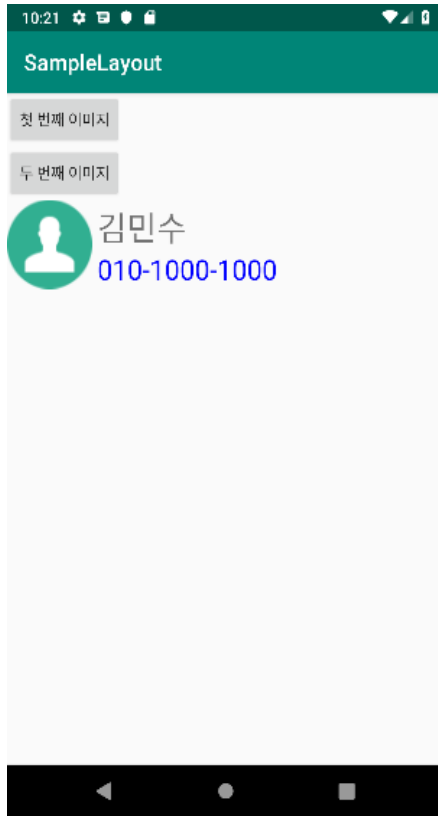
레이아웃 파일 추가

■ 버튼 클릭 시의 이벤트 처리

```
btn1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        layout1.setName("김민수");  
        layout1.setMobile("010-1000-1000");  
        layout1.setImage(R.drawable.profile1);  
    }  
});  
btn2.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        layout1.setImage(R.drawable.profile2);  
        layout1.setName("박경미");  
        layout1.setMobile("010-1000-2000");  
    }  
});
```

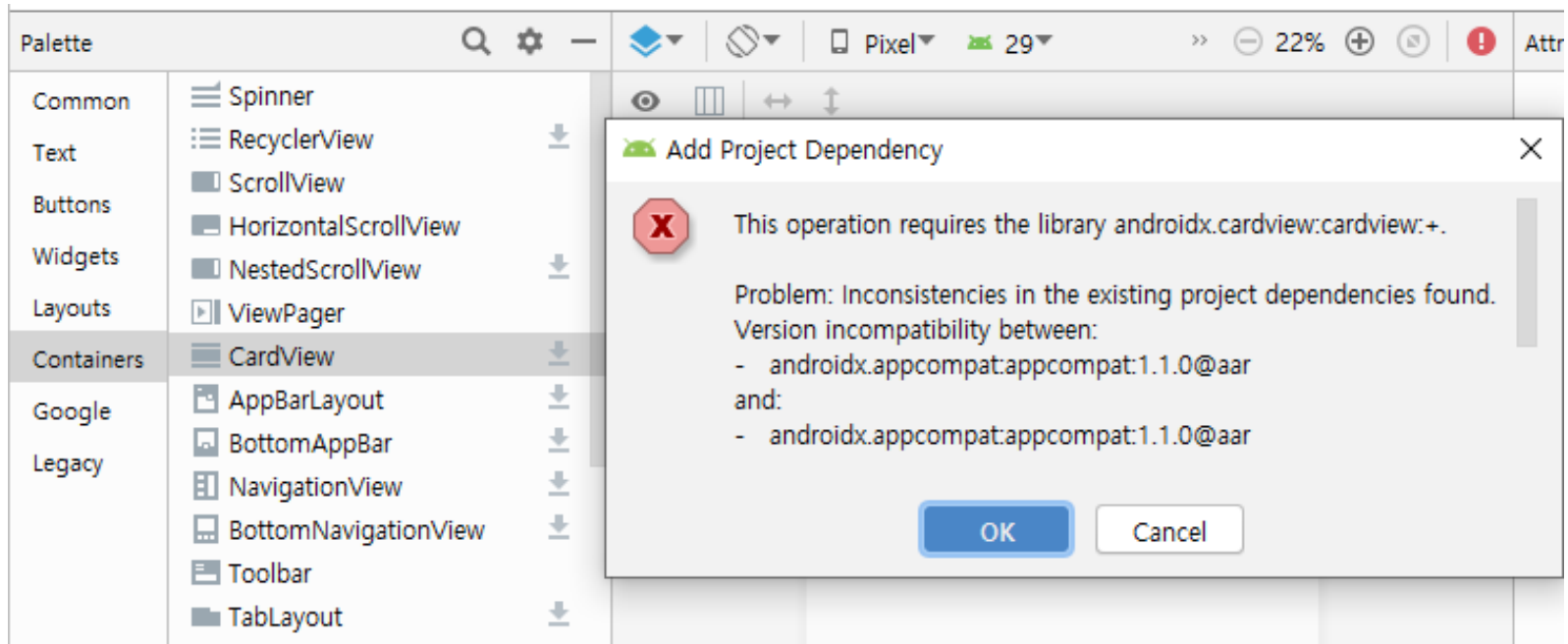
레이아웃 파일 추가

■ 앱 실행



카드뷰 모양으로 바꾸기

•카드뷰를 위한 외부 라이브러리 추가



3. 레이아웃 정의하고 카드뷰 넣기

카드뷰 모양으로 바꾸기

•카드뷰 태그 추가

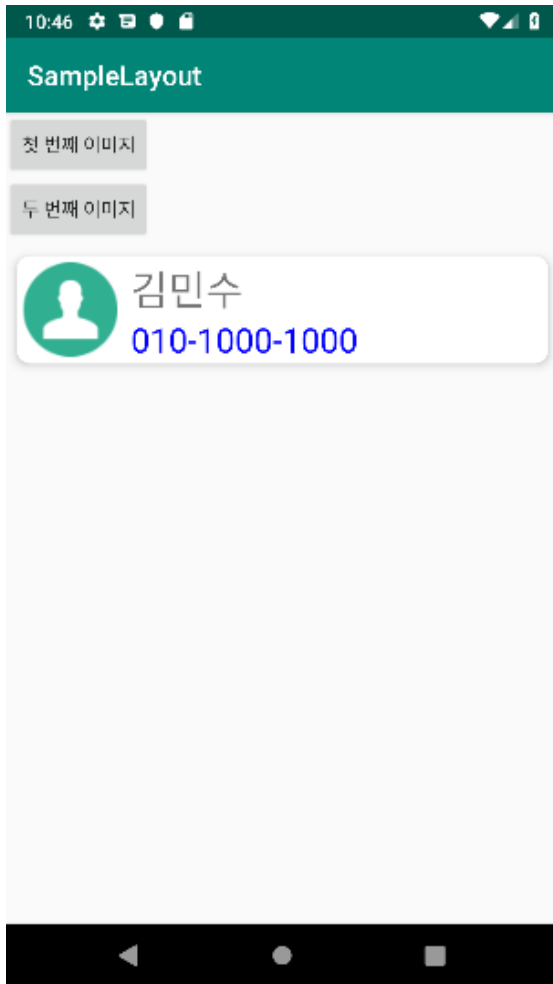
```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

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

        ....
    </androidx.cardview.widget.CardView>
</LinearLayout>
```

카드뷰 모양으로 바꾸기

- 카드뷰 모양으로 바꾼 결과



어댑터 뷰(선택위젯)이라는 이름으로 구분할까?

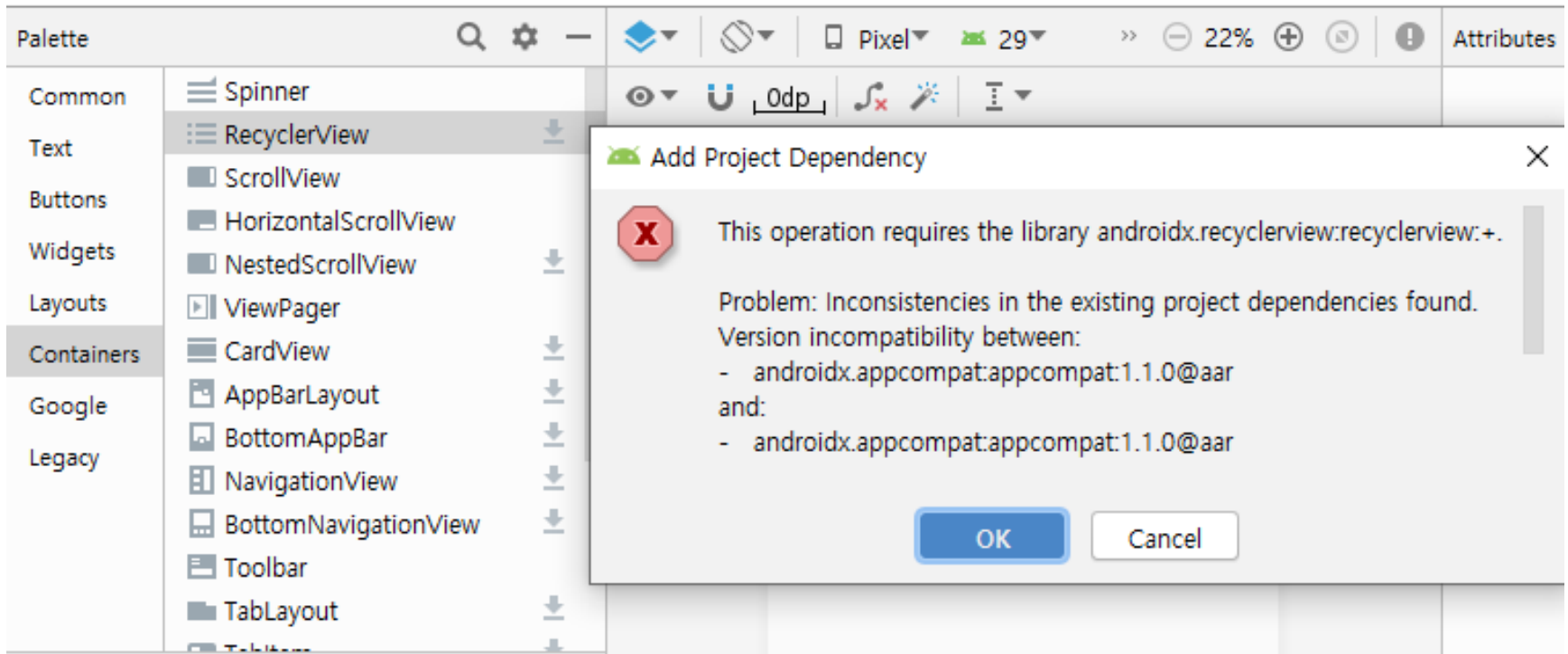
- 안드로이드에서는 여러 아이템 중의 하나를 선택하는 선택위젯은 별도의 패턴을 사용함
 - 여러 개의 아이템 중에서 하나를 선택하는 방식의 선택 위젯은 어댑터를 사용하여야 함
 - 이 어댑터에서 데이터를 관리하도록 해야 할 뿐만 아니라 화면에 보여지는 뷰도 어댑터의 getView() 메소드에서 결정함
 - 선택위젯의 가장 큰 특징은 원본 데이터를 위젯에 직접 설정하지 않고 어댑터라는 클래스를 사용하도록 되어 있다는 점으로 이 패턴을 잘 기억해 두어야 함



[선택 위젯과 어댑터]

리사이클러뷰 만들기

- 팔레트에서 RecyclerView 오른쪽에 있는 아이콘 눌러 외부 라이브러리 추가



4. 리사이클러뷰 만들기


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" />
```



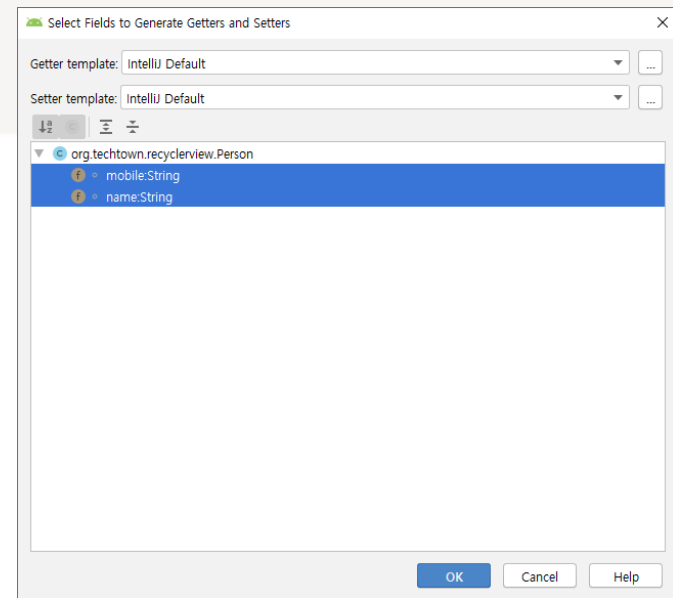
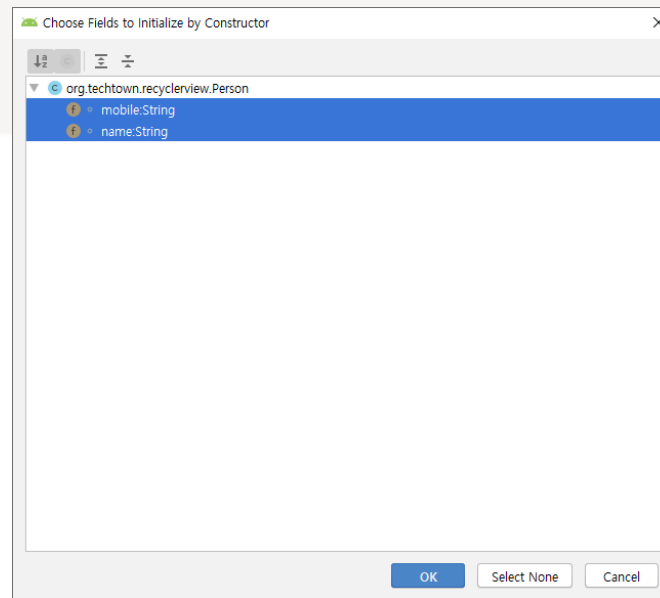
리사이클러뷰 태그 추가하기

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;  
}
```



4. 리사이클러뷰 만들기

리사이클러뷰 만들기

- 어댑터 소스 추가하고 ViewHolder 정의

참조파일 SampleRecyclerView>/app/java/org.techtown.sampleryclerview/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

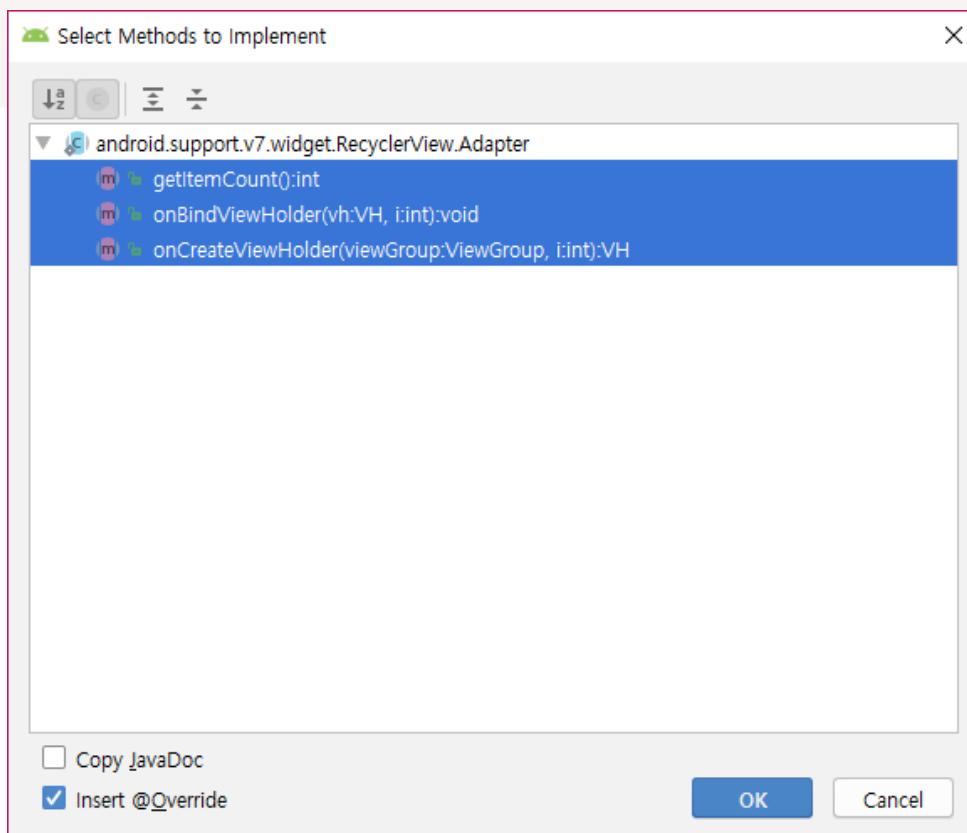
리사이클러뷰 만들기

- 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.samplerecyclerview/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  
                                                    통해 뷰 객체  
                                                    만들기  
  
        return new ViewHolder(itemView); —→ ❷ 뷰홀더 객체를 생성하면서 뷰 객체를  
                                                    전달하고 그 뷰홀더 객체를 반환하기  
    }  
  
    @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" />
```

리싸이클러뷰 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.samplerecyclerview/MainActivity.java

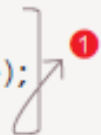
```
public class MainActivity extends AppCompatActivity {

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

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

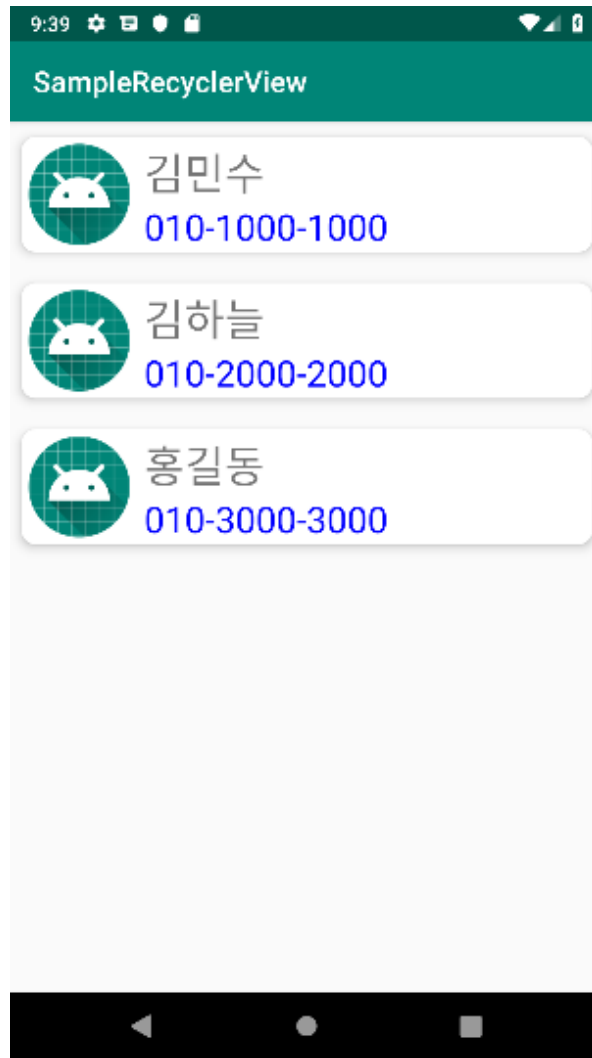
        LinearLayoutManager layoutManager =
            new LinearLayoutManager(this, LinearLayoutManager.VERTICAL, false);
        recyclerView.setLayoutManager(layoutManager);
        PersonAdapter adapter = new PersonAdapter();

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



리사이클러뷰 만들기

■ 앱 실행



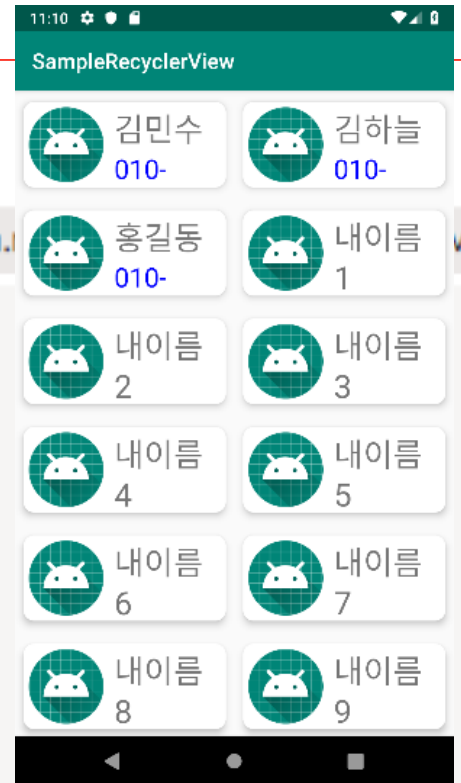
리사이클러뷰 만들기

- 여러 칼럼의 뷰로 보여주기

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

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

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

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

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
미리 정의한 다른 리스너의
메서드 호출하기

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);  
    }  
}
```

중략...

리사이클러뷰 만들기

■ 메인 액티비티에 추가

```
recyclerView.setAdapter(adapter);

adapter.setOnItemClickListener(new OnItemClickListener() { → 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();

    }
});
}
```

