

# android

~~8월~~ 6일 만에  
배워보기



<https://github.com/ye0jun/AndroidStudy>



## Android day #2

ye0jun@naver.com

### Introduce

### Learn

### Do it!

- Develop Language - JAVA
- Goal - 지도 기반 메모 어플리케이션 제작
- 수업 목표 교육 및 실습 / 어플리케이션에 적용
  
- #1 - 안드로이드 레이아웃의 이해 및 기본 컴포넌트 활용
- #2 - Custom View 제작
- #3 - 안드로이드 라이브러리 사용방법 / Activity 생성 및 전환 / +@Fragment
- #4 - 안드로이드 네트워크
- #5 - API 활용 및 데이터 처리
- #6 - 안드로이드 데이터베이스 활용



## Android day #2

ye0jun@naver.com

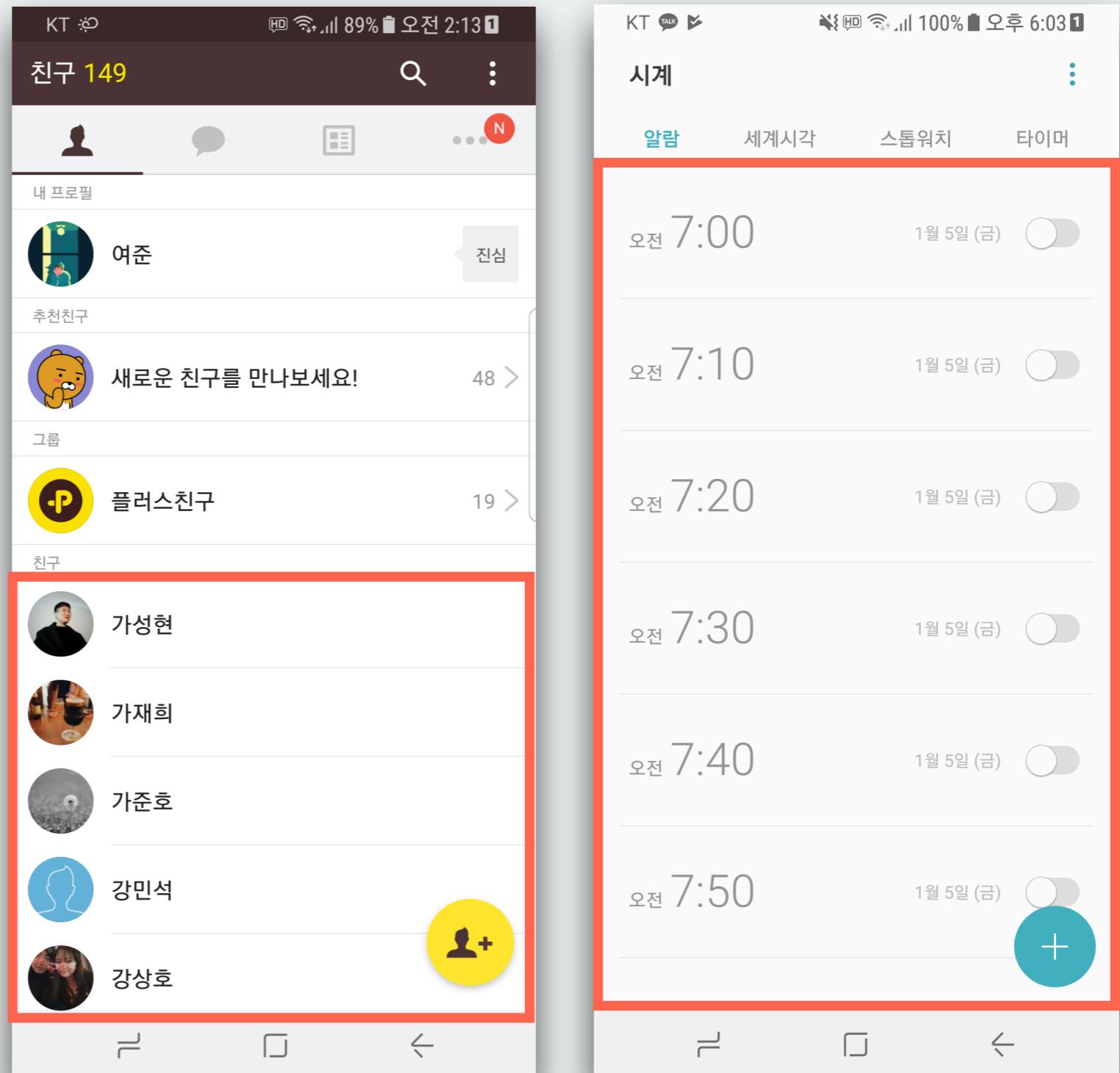
Introduce

Learn

Do it!

[https://github.com/  
ye0jun/AndroidStudy](https://github.com/ye0jun/AndroidStudy)

# AdapterView





## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

[https://github.com/  
ye0jun/AndroidStudy](https://github.com/ye0jun/AndroidStudy)





## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

[https://github.com/  
ye0jun/AndroidStudy](https://github.com/ye0jun/AndroidStudy)

KT TALK 95% 오후 8:44

벅스차트  
전체 장르

곡 앨범 영상 Bside 곡 Bside 영상

실시간 일간 주간 2018.01.04 20:00

▶ 전체 듣기  전체 선택

1		칠해줘 마마무(Mamamoo)	▶	⋮
2		instagram DEAN(딘)	▶	⋮
3		그날처럼 장덕철	▶	⋮
4		겨울소리 박효신	▶	⋮
5		미안해 양다일	▶	⋮
6		Heart Shaker TWICE(트와이스)	▶	⋮
7		선물 멜로망스(MeloMance)	▶	⋮
		Parade 윤하(Younha/ Yunna)	▶	≡



## Android day #2

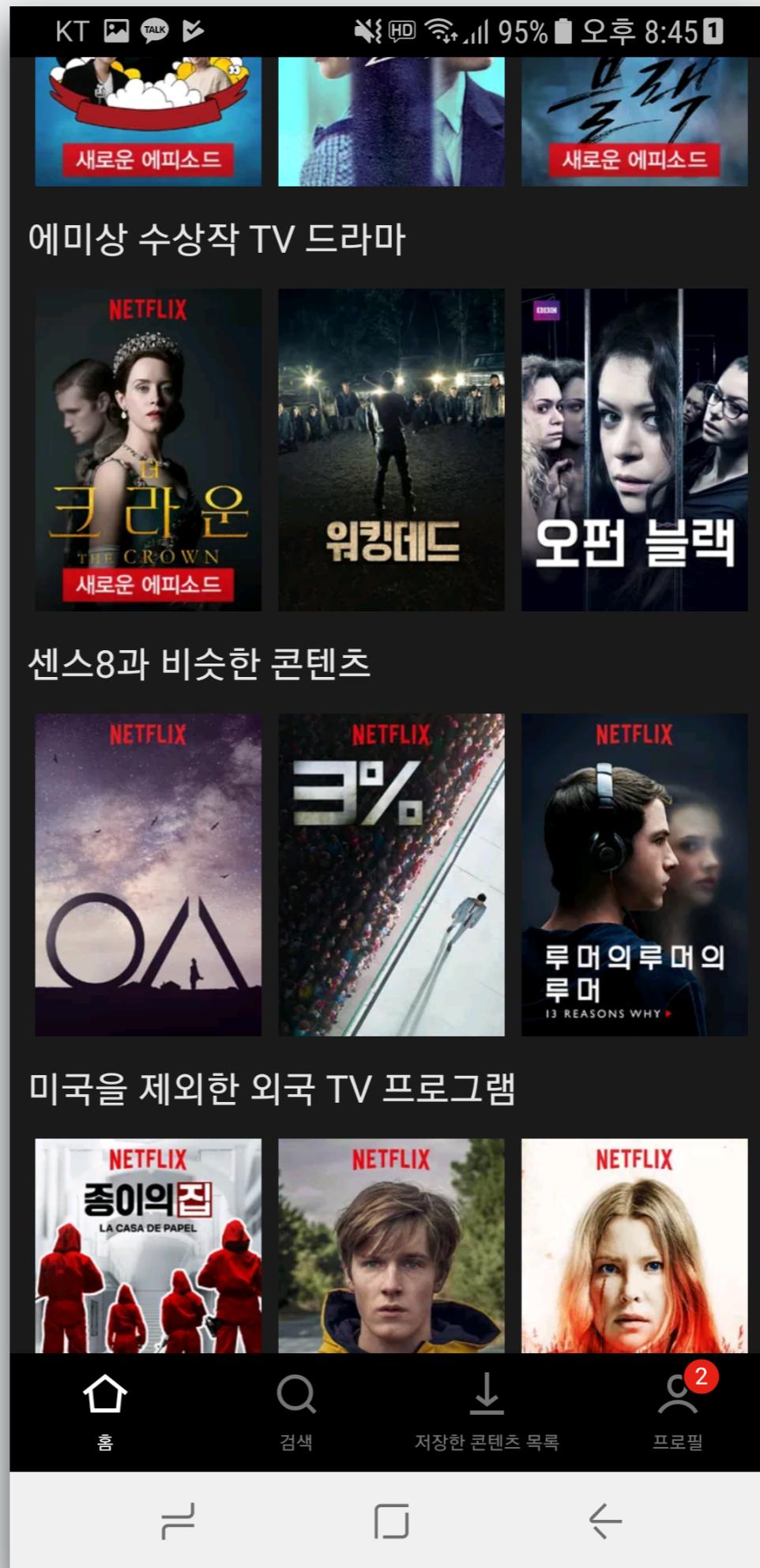
ye0jun@naver.com

Introduce

Learn

Do it!

[https://github.com/  
ye0jun/AndroidStudy](https://github.com/ye0jun/AndroidStudy)





## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

[https://github.com/  
ye0jun/AndroidStudy](https://github.com/ye0jun/AndroidStudy)

The image shows a screenshot of the YouTube app on an Android device. At the top, there's a navigation bar with icons for signal strength, battery level (94%), and time (오후 8:45). Below that is the YouTube logo and a search bar. There are four category icons: 음악 (Music), 실시간 (Live), 게임 (Games), and 뉴스 (News). The main content area features a video thumbnail with three men looking at large plates of fried chicken. The title of the video is "양념 vs 깐풍 vs 닭똥진" (Soy sauce vs. pepper vs. chicken feces) and it has a duration of 10:17. Below the thumbnail, the channel name is "영국남자" (Korean Englishman) and the description is "한국 치맥 왕중왕전: 외국인 입맛에 제일 잘 맞는 한국 치킨은?!" (King of Korean Fried Chicken: Which Korean chicken is the best for foreign taste?). It also says "영국남자 Korean Englishman" and "조회수 64.1만회" (View count 64.1 million). At the bottom of the screen, there are navigation icons for Home, Popular (highlighted in red), Subscriptions, Activity, and Library.



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

[https://github.com/  
ye0jun/AndroidStudy](https://github.com/ye0jun/AndroidStudy)

KT TALK 94% 오후 8:46

← 맛집랭킹 ↗

분식 돈까스,회,일식 치킨 피자 중식

★★★★★ 바다여행  
최근리뷰 1,000+ 최근사장님댓글 1,100+  
바로결제 만나서결제

★★★★★ NEW LA 수제 돈까스  
최근리뷰 70+ 최근사장님댓글 70+  
바로결제 만나서결제

★★★★★ 씨씨월드  
최근리뷰 1,000+ 최근사장님댓글 1,000+  
바로결제 만나서결제

★★★★★ 라운지앤  
최근리뷰 150+ 최근사장님댓글 150+  
바로결제 만나서결제

★★★★★ 후레쉬팩토리 스테이크  
최근리뷰 100+ 최근사장님댓글 100+  
바로결제 만나서결제

☰ ☐ ←



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# AdapterView

AdapterView만 사용해서는 화면에 아무것도 나오지 않는다.

AdapterView에 항목이 나열되어 화면에 데이터가 나오게 하려면  
**Adapter**라는 클래스를 이용해야 함





## Android day #2

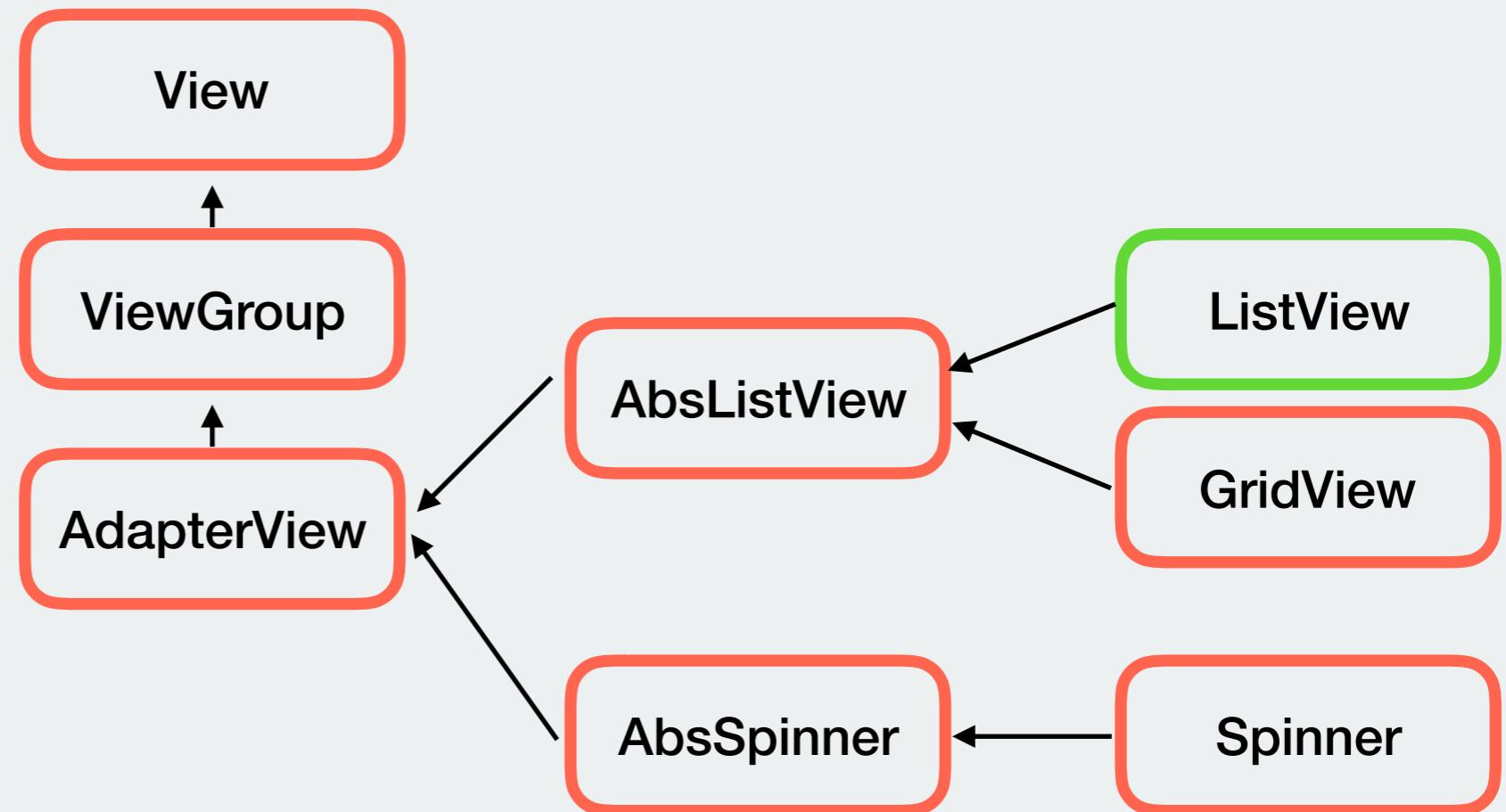
ye0jun@naver.com

Introduce

Learn

Do it!

# ListView



AdapterView들은 구조적으로 라이브러리  
AdapterView를 상속받아 작성된 뷰를 의미



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# ArrayAdapter

각 항목에 문자열 데이터를 순서대로 하나씩 나열할 때 사용



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# ArrayAdapter

햄버거

---

피자

---

치킨

---

짜장면

---

족발

---

...



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# ArrayAdapter

```
<ListView  
    android:id="@+id/mylist"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" />
```



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# ArrayList

```
ArrayList<자료형> 변수명;  
ArrayList<String> data;
```

```
data = new ArrayList<String>();
```



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# ArrayList

```
data.add(같은 자료형의 데이터);  
data.add(특정한 위치, 같은 자료형의 데이터);  
data.clear();  
data.get(특정한 위치);  
data.isEmpty();  
data.remove(특정한 위치);  
data.set(특정한 위치, 같은 자료형의 데이터);  
data.size();
```



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# ArrayAdapter

```
ArrayList<String> data = new ArrayList<String>();  
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,  
    android.R.layout.simple_list_item_1, data);  
  
ListView myListview = (ListView) findViewById(R.id myList);  
myListview.setAdapter(adapter);
```



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# ArrayAdapter

```
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,  
    android.R.layout.simple_list_item_1, data);
```

- **this** : Context 객체
- **android.R.layout.simple\_list\_item\_1** : 항목 하나를 구성하기 위한 XML
- **data** : 항목을 구성하는 데이터
  
- **simple\_list\_item\_1** : 항목에 문자열 데이터 하나
- **simple\_list\_item\_2** : 항목에 문자열 데이터 두 개 위아래 나열
- **simple\_list\_item\_multiple\_choice** : 문자열과 오른쪽 체크박스 제공
- **simple\_list\_item\_single\_choice** : 문자열과 오른쪽 라디오버튼 제공



## Android day #2

ye0jun@naver.com

Introduce

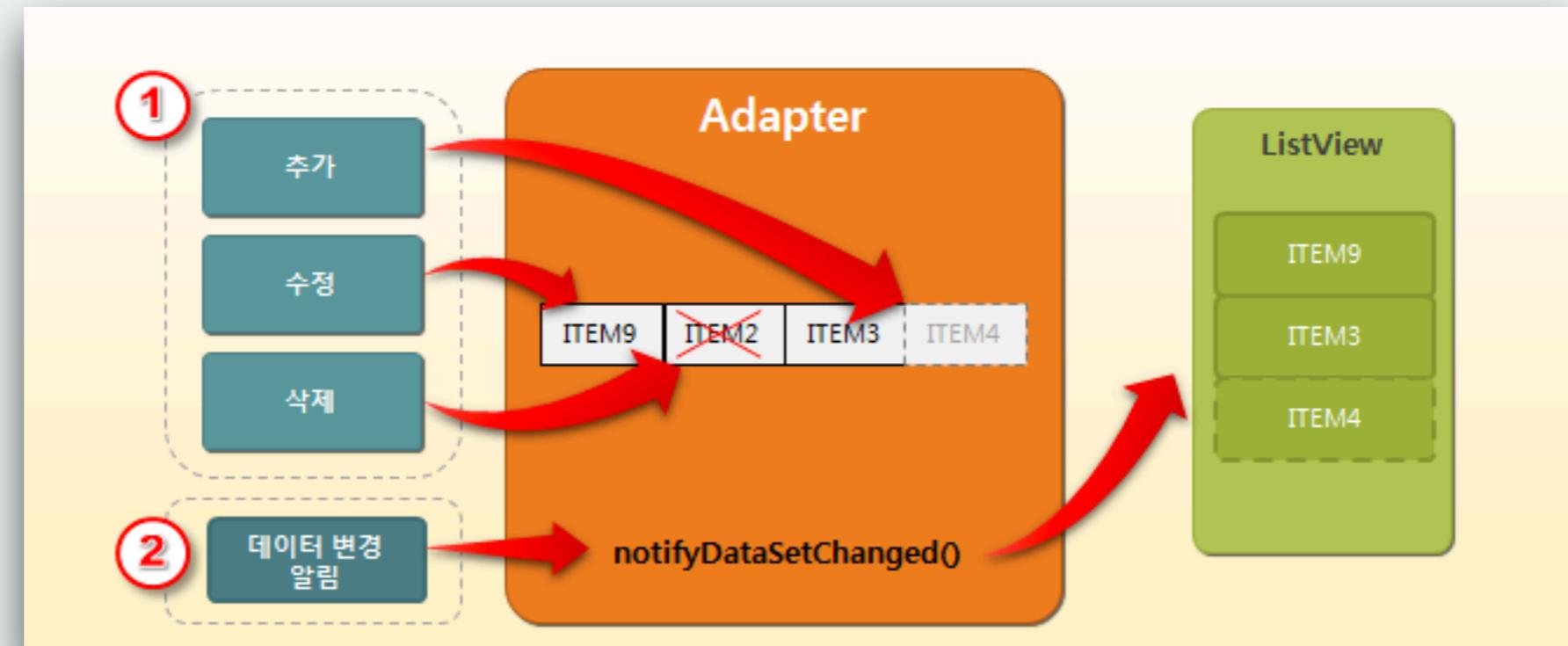
Learn

Do it!

# ArrayAdapter

동적으로 UI를 변경했을 때에는 adapter에 알려줘야함

`adapter.notifyDataSetChanged();`





## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# ArrayAdapter

리스트뷰의 이벤트 처리

A screenshot of an Android application window. The title bar has three colored dots (red, yellow, green). The main area is a dark gray list view that is currently empty, showing only its header row.

```
myListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
        Toast.makeText(getApplicationContext(), data.get(i), Toast.LENGTH_SHORT).show();
    }
});
myListView.setAdapter(adapter);
```



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# CustomAdapter

직접 Adapter를 제작하여 ListView 화면을 구성하는 방법을 알아보자

단순 문자열 나열 이상의 ListView를 만들어야 한다면  
개발자가 직접 커스텀하여 Adapter를 만들어서 적용해야 한다

CustomAdapter 란 개발자가 직접 만든 Adapter를 의미한다



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# CustomAdapter

## Adapter를 직접 만들어 개발해야 하는 상황

- 개발자 알고리즘대로 항목의 데이터가 설정되어야 할 때
- 개발자 알고리즘대로 항목별 뷰의 이벤트를 다르게 처리해야 할 때
- 개발자 알고리즘대로 항목별 레이아웃을 다르게 적용해야 할 때

실제 개발을 하게 되면 ListView가 대부분 복잡할 것이므로  
Custom Adapter는 선택이 아니라 필수일 것



## Android day #2

ye0jun@naver.com

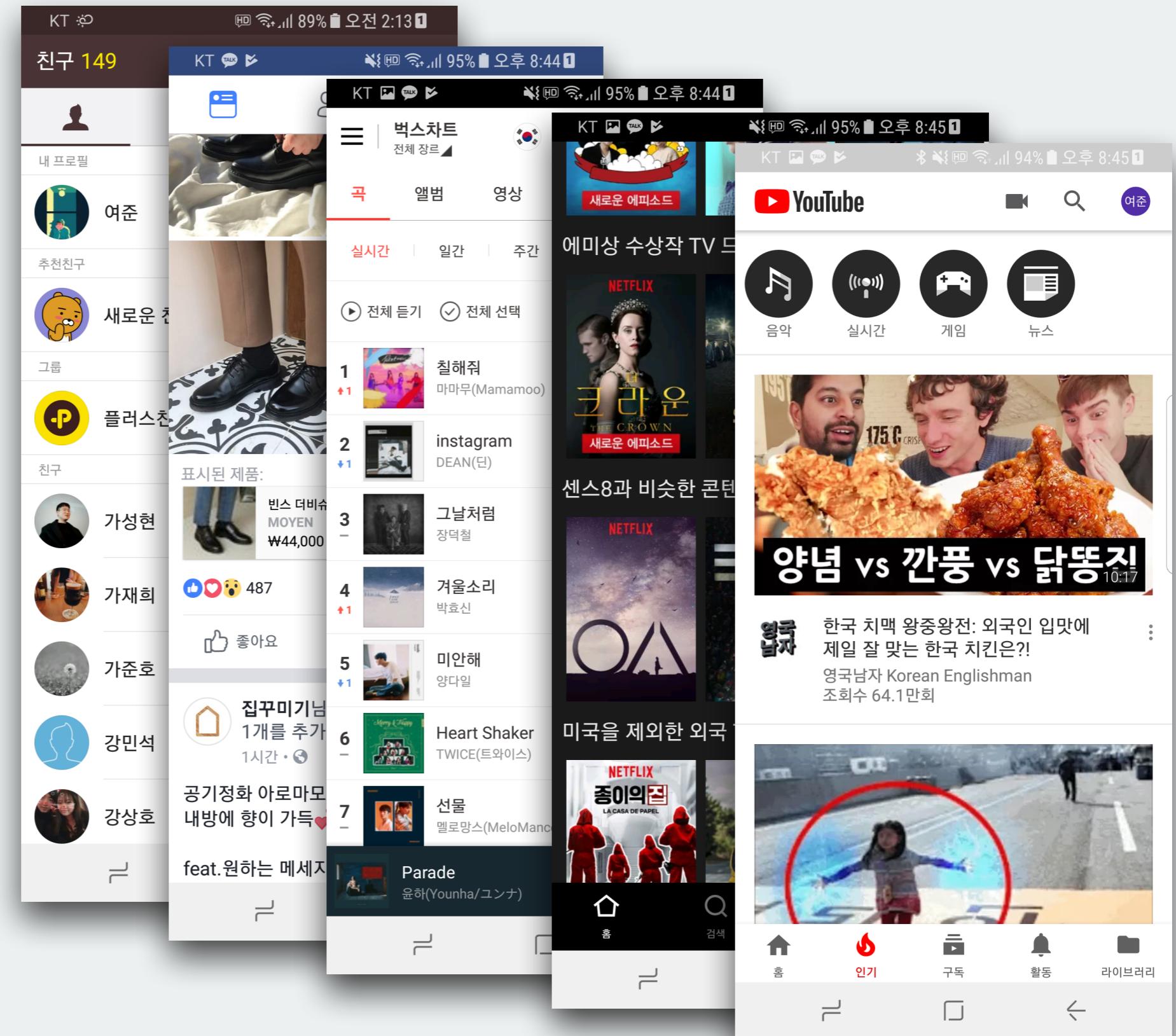
Introduce

Learn

Do it!

[https://github.com/  
ye0jun/AndroidStudy](https://github.com/ye0jun/AndroidStudy)

# CustomAdapter





## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

[https://github.com/  
ye0jun/AndroidStudy](https://github.com/ye0jun/AndroidStudy)

# CustomAdapter

하나의 아이템에 해당하는 뷰를 제작

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="100dp"  
    android:orientation="vertical">  
  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Title"/>  
  
    <View  
        android:layout_width="match_parent"  
        android:layout_height="1dp"  
        android:background="#000000" />  
  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Content" />  
  
    <View  
        android:layout_width="match_parent"  
        android:layout_height="1dp"  
        android:background="#123456" />  
  
</LinearLayout>
```



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# CustomAdapter

CustomAdapter를 만들려면 우선 항목별 데이터를 추상화한 클래스를 정의

```
public class DataModel{  
    String title;  
    String content;  
  
    public DataModel(String title, String content){  
        this.title = title;  
        this.content = content;  
    }  
  
    public String getTitle(){  
        return title;  
    }  
  
    public String getContent(){  
        return content;  
    }  
}
```



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# CustomAdapter

```
public class myAdapter extends ArrayAdapter<DataModel>{
    Context context;
    private ArrayList<DataModel> datas;

    public myAdapter(Context context, ArrayList<DataModel> datas){
        //...
    }

    @Override
    public int getCount(){
        //...
    }

    @NonNull
    @Override
    public View getView(int position, View convertView, ViewGroup parent){
        //...
    }
}
```



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# CustomAdapter - `getView`

```
@Override  
public View getView(int position, View convertView, ViewGroup parent){  
    LayoutInflator inflater =  
        (LayoutInflator)context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);  
    convertView = inflater.inflate(R.layout.custom_item,null);  
}
```



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# CustomAdapter - getView

```
@Override  
public View getView(int position, View convertView, ViewGroup parent){  
    ...  
    TextView tv = (TextView)convertView.findViewById(R.id.tv);  
    final DataModel dm = datas.get(position);  
    tv.setText(dm.title);  
}
```



## Android day #2

ye0jun@naver.com

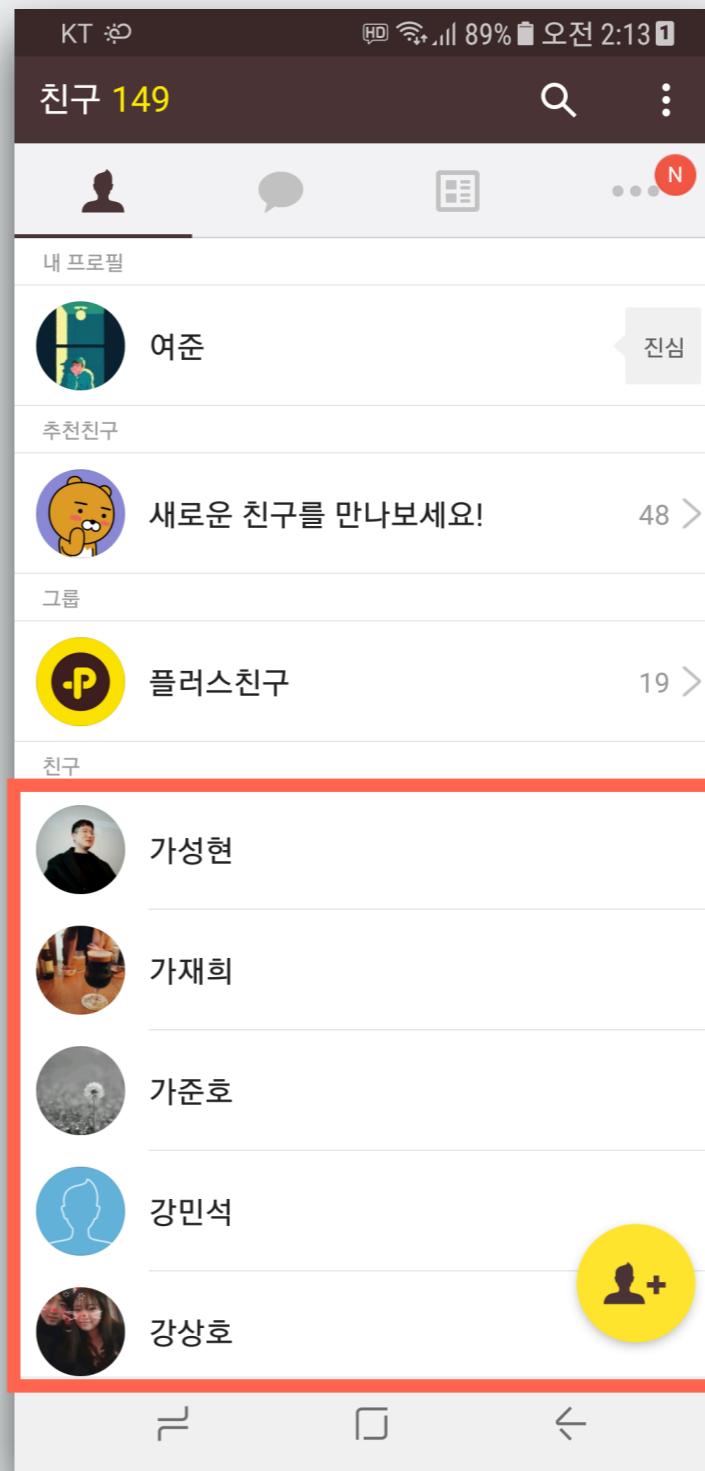
Introduce

Learn

Do it!

[https://github.com/  
ye0jun/AndroidStudy](https://github.com/ye0jun/AndroidStudy)

# AdapterView





## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# CustomAdapter - 고려사항

레이아웃 초기화 성능 이슈 : LayoutInflater

Adapter가 항목을 구성하려면 레이아웃 XML을 초기화  
=> LayoutInflater 클래스의 inflate() 함수 이용

레이아웃 초기화는 내부적으로 많은 코드 실행  
=> 성능상 부담

한 항목이 화면에 보일 때마다 반복해서 할 필요가 있을까?  
=> 바뀌는건 데이터, 레이아웃이 바뀌는 경우는 거의 없음

성능상 부담되는 레이아웃 XML을 항목이 최초로 나오는 순간  
한 번만 이루어지게 해줌



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# CustomAdapter - 고려사항

레이아웃 초기화 성능 이슈 : LayoutInflater

```
public View getView(int position, View convertView, ViewGroup parent){  
    if(convertView == null){  
        LayoutInflater inflater = LayoutInflater.from(context);  
        convertView = inflater.inflate(R.layout.custom_item,null);  
    }  
    ...  
    return convertView;  
}
```



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# CustomAdapter - 고려사항

뷰 획득 시 성능 이슈 : `findViewById`

뷰를 획득하기 위해서는 `findViewById()` 함수 이용  
=> `findViewById()` 내부적으로 많은 코드 실행 (부담)

처음 획득한 뷰를 저장했다가 그다음 이용 시 `findViewById()` 함수를 호출하지 않고 `저장된 뷰를 그대로 이용`



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# CustomAdapter - 고려사항

뷰 획득 시 성능 이슈 : findViewById

```
public class ViewHolder{  
    public TextView tv_Name;  
  
    public ViewHolder(View root){  
        tv_Name = (TextView)root.findViewById(R.id.custom_name);  
    }  
}
```

```
@Override  
public View getView(int position, View convertView, ViewGroup parent){  
    if(convertView == null){  
        LayoutInflator inflater = LayoutInflator.from(context);  
        convertView = inflater.inflate(R.layout.custom_item,null);  
  
        ViewHolder holder = new ViewHolder(convertView);  
        convertView.setTag(holder);  
    }  
}
```



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

# CustomAdapter - 고려사항

뷰 획득 시 성능 이슈 : findViewById

```
ViewHolder holder = (ViewHolder)convertView.getTag();
TextView tv = holder.tv_Name;
final DataModel dm = datas.get(position);
tv.setText(dm.title);
```



## Android day #2

ye0jun@naver.com

Introduce

Learn

Do it!

[https://github.com/  
ye0jun/AndroidStudy](https://github.com/ye0jun/AndroidStudy)

