



04

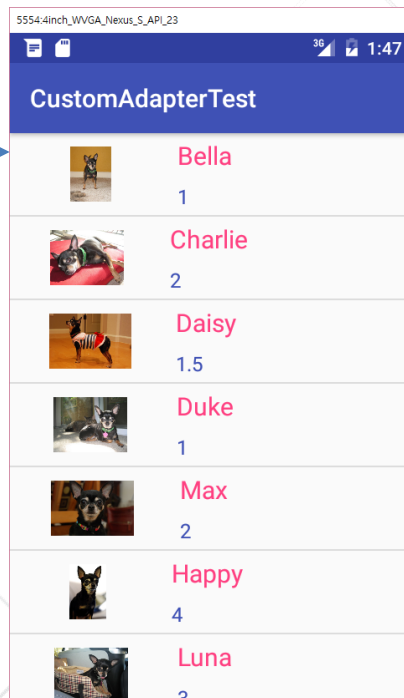
커스텀 어댑터 뷰

(Custom Adapter View)



커스텀 어댑터 뷰(Custom Adapter View)란?

- ◆ 어댑터 뷰의 항목 하나는 단순한 문자열 이나 이미지 뿐만 아니라, 임의의 뷰가 될 수 있음



- ◆ 커스텀 어댑터 뷰 설정 절차

1

항목을 위한 XML 레이아웃 정의

2

어댑터 정의

3

어댑터를 생성하고 어댑터 뷰 객체에 연결



1. 항목 XML 레이아웃 정의

```
<?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="wrap_content"
    android:orientation="horizontal"
    >
    <ImageView
        android:id="@+id/iconItem"
        android:layout_width="@dimen/icon_size"
        android:layout_height="@dimen/icon_size"
        android:scaleType="centerInside"
        android:padding="@dimen/icon_padding"
        android:layout_gravity="center_vertical"
        android:layout_weight="1"
        />
    ...
```

이미지뷰

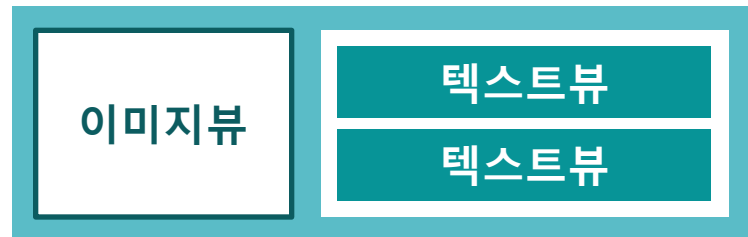
텍스트뷰

텍스트뷰



1. 항목 XML 레이아웃 정의

```
...  
<LinearLayout  
    android:orientation="vertical"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_weight="2">  
    <TextView  
        android:id="@+id/textItem1"  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:textColor="@color/colorAccent"  
        android:textSize="@dimen/list_item_text_size1"  
        android:padding="@dimen/list_item_padding"  
    />  
    <TextView  
        android:id="@+id/textItem2"  
        ...  
    />  
</LinearLayout>  
</LinearLayout>
```





2. 어댑터 정의

```
class MyItem {
    int mIcon; // image resource
    String nName; // text
    String nAge; // text
    MyItem(int aIcon, String aName, String aAge) { mIcon = aIcon; nName = aName; nAge = aAge; }
}
class MyAdapter extends BaseAdapter {
    private Context mContext;
    private int mResource;
    private > mItems = new ArrayList<MyItem>();
    ArrayList<MyItem>
    public MyAdapter(Context context, int resource, ArrayList<MyItem> items) {
        mContext = context;
        mItems = items;
        mResource = resource;
    }
    public int getCount() { return mItems.size(); }
    public Object getItem(int position) { return mItems.get(position); }
    public long getItemId(int position) { return position; }
    ...
}
```



2. 어댑터 정의

```
public View getView(int position, View convertView, ViewGroup parent) {
    if (convertView == null) {
        LayoutInflater inflater =
            (LayoutInflater) mContext.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        convertView = inflater.inflate(mResource, parent, false);
    }
    // Set Icon
    ImageView icon = (ImageView) convertView.findViewById(R.id.iconItem);
    icon.setImageResource(mItems.get(position).mIcon);

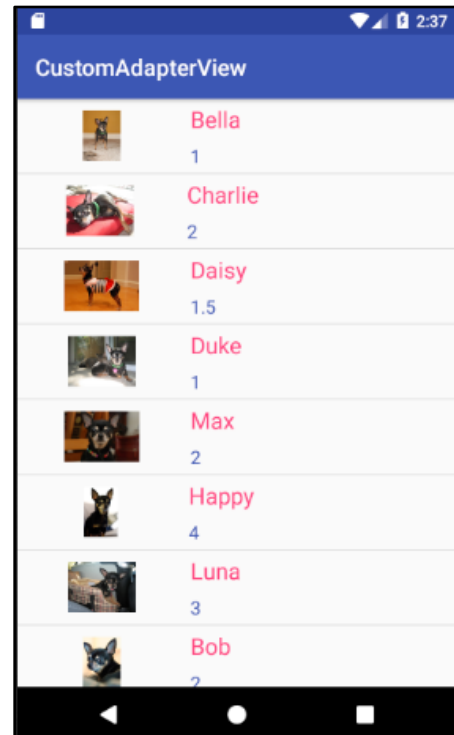
    // Set Text 01
    TextView name = (TextView) convertView.findViewById(R.id.textItem1);
    name.setText(mItems.get(position).nName);

    // Set Text 02
    TextView age = (TextView) convertView.findViewById(R.id.textItem2);
    age.setText(mItems.get(position).nAge);
    return convertView;
}
```



3. 어댑터 생성 및 연결

```
public class MainActivity extends AppCompatActivity {  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        // 데이터 원본 준비  
        ArrayList<MyItem> data = new ArrayList<MyItem>();  
        data.add(new MyItem(R.drawable.sample_0, "Bella", "1"));  
        data.add(new MyItem(R.drawable.sample_1, "Charlie", "2"));  
        ...  
  
        // 어댑터 생성  
        MyAdapter adapter = new MyAdapter(this, R.Layout.item, data);  
  
        // 어댑터 연결  
        ListView list = (ListView) findViewById(R.id.listView);  
        list.setAdapter(adapter);  
    }  
}
```





항목 클릭 이벤트 처리

- ◆ AdapterView의 항목이 클릭될 때, 호출되는 callback method의 인터페이스

```
public static interface AdapterView.OnItemClickListener {  
    abstract void onItemClick(AdapterView<?> parent,  
                               View view,  
                               int position,  
                               long id);  
}
```



파라미터	설명
parent	클릭 이벤트가 발생한 AdapterView
view	실제 클릭 된 AdapterView안의 View
position	어댑터 내에서 클릭 된 항목/뷰의 위치
id	클릭 된 항목의 id

4 커스텀 뷰(Custom View)



항목 클릭 이벤트 처리

```
MyAdapter adapter = new MyAdapter(this, R.layout.item, data);
...
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    public void onItemClick(AdapterView<?> parent, View vClicked, int position, long id) {

        // 클릭된 뷰(vClicked)안에서 id가 textItem1인 텍스트뷰의 레이블을 얻어 옴
        // String name = (String)((TextView)vClicked.findViewById(R.id.textItem1)).getText();

        // 어댑터(adapter) 내의 position 위치의 항목의 nName 값을 읽어 옴
        String name = ((MyItem)adapter.getItem(position)).nName;

        Toast.makeText(MainActivity.this, name + " selected", Toast.LENGTH_SHORT).show();
    }
});
```