

# 리스트뷰

# ListView

- 데이터를 리스트 모양으로 보여주며 리스트 중 하나를 선택하는 용도로 사용



# 기본 ListView 구현 – xml

---

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

# xml을 이용한 리스트뷰 적용하기

---

## ➤ 리스트뷰를 만드는 형식

1. 리스트뷰에 나열할 내용을 String 배열로 미리 만듦
2. 리스트뷰 변수를 생성하고 XML의 <ListView>에 대응시킴
3. ArrayAdapter<String> 형의 변수를 선언하고, 리스트뷰의 모양과 내용을 ①번에서 생성한 배열로 채움
4. ③번에서 생성한 어레이어댑터를 ②번의 리스트뷰 변수에 적용시킴
5. 리스트뷰의 항목을 클릭했을 때 동작하는 리스너 정의

# 리스트뷰

---

## 1. 리스트뷰에 나열할 내용을 String 배열로 미리 만듦

```
final ArrayList<String> actors = new ArrayList<>();  
actors.add("황정민");  
actors.add("정우성");  
.....  
actors.add("여진구");
```

## 2. 리스트뷰 변수를 생성하고 XML의 <ListView>에 대응시킴

```
ListView list = (ListView) findViewById(R.id.list);
```

# 리스트뷰

---

3. ArrayAdapter<String> 형의 변수를 선언하고, 리스트 뷰의 모양과 내용을  
①번에서 생성한 배열로 채움

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

4. ArrayAdapter<String> 형의 변수를 선언하고, 리스트 뷰의 모양과 내용을  
①번에서 생성한 배열로 채움

```
list.setAdapter(adapter);
```

# 리스트뷰

---

## 5. 리스트뷰의 항목을 클릭했을 때 동작하는 리스너 정의

```
list.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    public void onItemClick(  
        AdapterView<?> parent, View view, int position, long id) {  
        Toast.makeText(  
            getApplicationContext(),  
            actors.get(position),  
            Toast.LENGTH_SHORT).show();  
        }  
    });
```

# 기본 ListView 구현 – java

---

```
final ArrayList<String> actors = new ArrayList<>();
actors.add("황정민");
actors.add("정우성");
.....
actors.add("여진구");
ListView list = (ListView) findViewById(R.id.list);
ArrayAdapter<String> adapter = new ArrayAdapter<String>(
    this,
    android.R.layout.simple_list_item_1,
    actors
);
list.setAdapter(adapter);
```



# 기본 ListView 구현 – event

---

```
list.setOnItemClickListener(new AdapterView.OnItemClickListener() {  
    public void onItemClick(  
        AdapterView<?> parent, View view, int position, long id) {  
        Toast.makeText(  
            getApplicationContext(),  
            actors.get(position),  
            Toast.LENGTH_SHORT).show();  
        }  
    });
```

# 기본 ListView 구현 – 라디오 버튼(싱글 초이스)

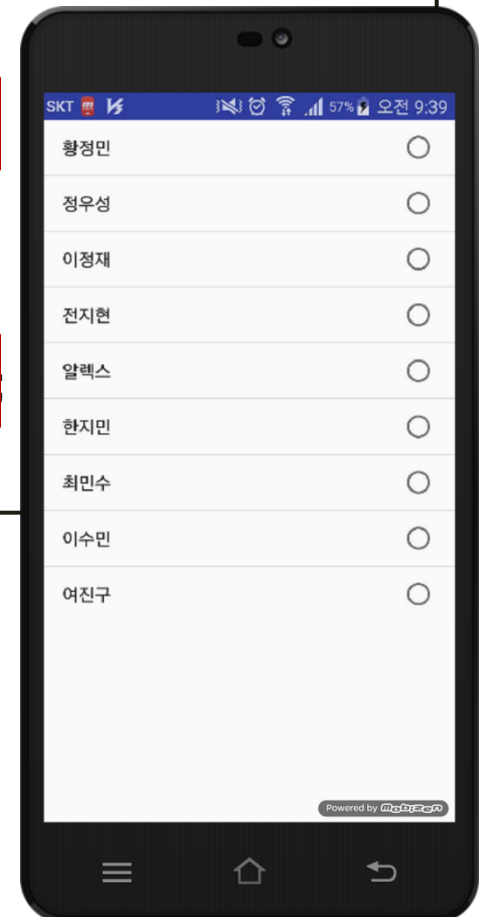
```
final ListView list = (ListView) findViewById(R.id.list);  
ArrayAdapter<String> adapter = new ArrayAdapter<String>(  
    this,  
    android.R.layout.simple_list_item_single_choice,  
    actors  
);  
list.setChoiceMode(ListView.CHOICE_MODE_SINGLE);  
list.setAdapter(adapter);
```

android.R.layout.*simple\_list\_item\_single\_choice*

리스트화면에 라디오 버튼 형태로 보이도록 설정한다.

*list.setChoiceMode(ListView.CHOICE\_MODE\_SINGLE)*;

실제 사용자가 하나를 선택 가능하게 한다.



# 기본 ListView 구현 – 체크박스(멀티초이스)

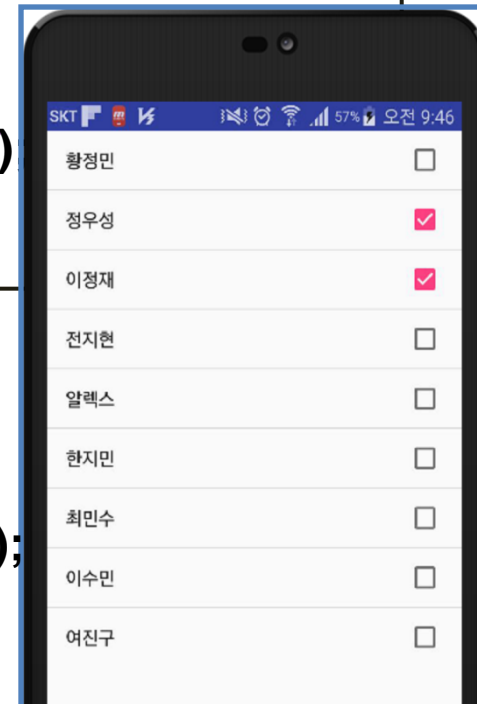
```
final ListView list = (ListView) findViewById(R.id.list);  
ArrayAdapter<String> adapter = new ArrayAdapter<String>(  
    this,  
    android.R.layout.simple_list_item_multiple_choice,  
    actors  
);  
list.setChoiceMode(ListView.CHOICE_MODE_MULTIPLE);  
list.setAdapter(adapter);
```

android.R.layout.*simple\_list\_item\_multiple\_choice*

리스트화면에 체크 버튼 형태로 보이도록 설정한다.

list.setChoiceMode(ListView.*CHOICE\_MODE\_MULTIPLE*);

실제 사용자가 여러개를 선택 가능하게 한다.



# Listview 데이터 추가 – xml

---

```
<LinearLayout>
    <EditText
        android:id="@+id/data"
        android:layout_weight="1"/>
    <Button
        android:id="@+id/addBtn"
        android:text="추가" />
</LinearLayout>
<ListView
    android:id="@+id/list"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
</ListView>
```

# Listview 데이터 추가 – java

---

```
final EditText data = (EditText)findViewById(R.id.data);
Button addBtn = (Button)findViewById(R.id.addBtn);
addBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String addData = data.getText().toString();
        adapter.add(data.getText().toString());
        data.setText("");
        list.scrollToPosition(adapter.getCount()-1);
    }
});
```

# ListView 데이터 삭제 – xml

---

**<Button**

**android:id="@+id/delBtn"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content"**

**android:text="삭제" />**

**<ListView**

**android:id="@+id/list"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content" >**

**</ListView>**

# Listview 데이터 삭제 – java

---

```
Button delBtn = (Button)findViewById(R.id.delBtn);
delBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (list.getChoiceMode() == ListView.CHOICE_MODE_SINGLE) {
            int position = list.getCheckedItemPosition();
            String item = (String) list.getItemAtPosition(position);
            adapter.remove(item);
        }
    }
});
```

# ListView 멀티 초이스 데이터 확인 – xml

---

**<Button**

**android:id="@+id/showBtn"**

**android:text="선택확인" />**

**<TextView**

**android:id="@+id/showText"**

**android:text="선택된 아이템"**

**android:padding="10dp"/>**

**<ListView**

**android:id="@+id/list"**

**android:layout\_width="match\_parent"**

**android:layout\_height="wrap\_content">**

**</ListView>**



# Listview 멀티 초이스 데이터 확인- java

---

```
Button showBtn = (Button)findViewById(R.id.showBtn);
final TextView showText = (TextView)findViewById(R.id.showText);
showBtn.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        SparseBooleanArray array = list.getCheckedItemPositions();
        ArrayList<String> select = new ArrayList<String>();
        for (int index = 0; index < array.size(); index++) {
            int position = array.keyAt(index);
            if (array.get(position)) {
                select.add((String)list.getItemAtPosition(position));
            }
        }
        showText.setText(select.toString());
    }
});
```