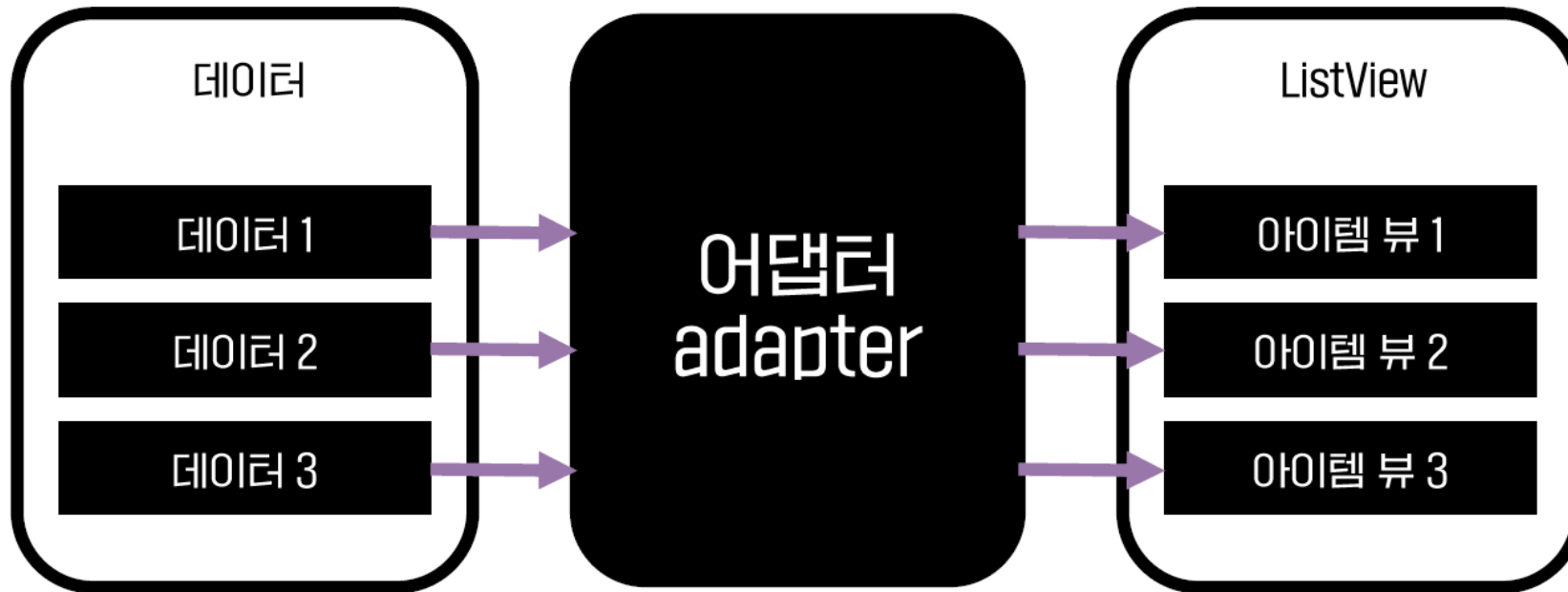


List View / Recycler View

EDCAN 수업 자료 - Android 기초

ListView

- 화면에 데이터들을 리스트의 형태로 보여주는 View
- 데이터들을 리스트로 만들기 위해서는 어댑터를 만들어 줘야 한다.



```

<ListView
    android:id="@+id/list_main_club"
    android:layout_width="0dp"
    android:layout_height="0dp"

    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toTopOf="@+id/btn_main_addItem"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
/>

<Button
    android:id="@+id/btn_main_addItem"
    android:layout_width="0dp"
    android:layout_height="wrap_content"

    android:text="추가"

    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
/>

```

XML

```

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val dataList = mutableListOf<String>("Item 1", "Item 2", "Item 3", "Item 4", "Item 5")
        val adapter = ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, dataList)
        findViewById<ListView>(R.id.list_main_club).adapter = adapter

        findViewById<Button>(R.id.btn_main_addItem).setOnClickListener {
            adapter.add("Item ${dataList.size + 1}")
            adapter.notifyDataSetChanged()
        }
    }
}

```

Kotlin

EDCAN Andorid

Item 1

Item 2

Item 3

Item 4

Item 5

Item 6

Item 7

Item 8

Item 9

Item 10

Item 11

Item 12

Item 13

추가

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)
```

```
        val dataList = mutableListOf<String>("Item 1", "Item 2", "Item 3", "Item 4", "Item 5")
```

화면에 리스트로 보여줄 데이터

```
        val adapter = ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, dataList)
```

어댑터를 만들어준다

```
        findViewById<ListView>(R.id.list_main_club).adapter = adapter
```

만들어준 어댑터를 연결시켜준다

```
        findViewById<Button>(R.id.btn_main_addItem).setOnClickListener {
```

```
            adapter.add("Item ${dataList.size + 1}")
```

리스트에 아이템을 추가한다.

```
            adapter.notifyDataSetChanged()
```

리스트를 새로 그려준다.

```
        }
```

```
    }
```

```
}
```

ListView 어댑터 기능

`add()` : 데이터 추가

`clear()` : 데이터 전체 삭제

`remove(item)` : 데이터 삭제

`getPosition(item)` : 해당 아이템의 위치

`notifyDataSetChanged()` : 리스트 새로 그리기

ListView 단점

- 리스트 아이템 커스텀을 할 수 없다.
- 리스트 아이템이 많아지면 화면에 보여주기 위한 시간이 오래 걸린다.
- 구글이 ListView 보다는 RecyclerView를 쓰라고 한다.

여러가지 이유 때문에 RecyclerView를 사용한다.

Recycler View

- 리스트 아이템 커스텀을 할 수 있다.
- 어댑터를 직접 만들어서 사용한다.
- ViewHolder를 만들어 줘야 한다.
- 리스트 아이템 레이아웃을 직접 만들어 줘야 한다.

리스트
아이템 디자인
(XML)

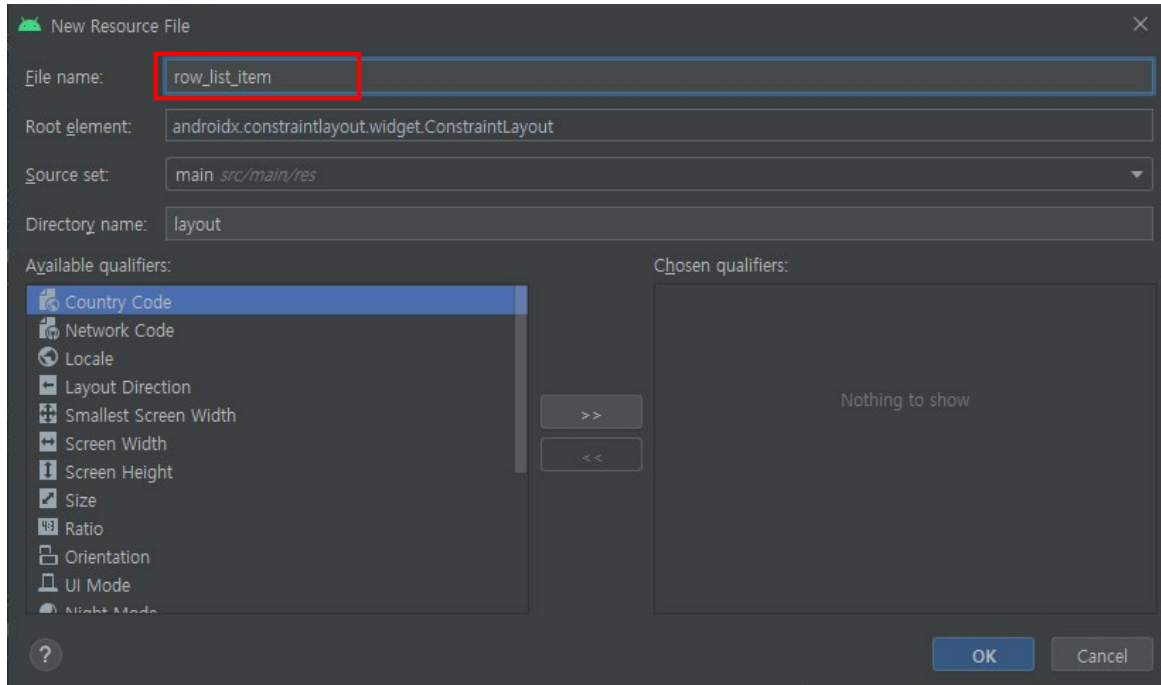
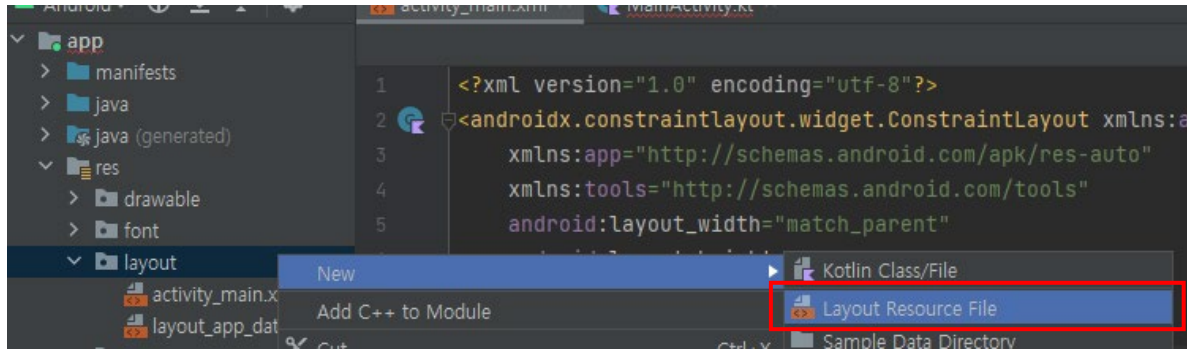
리스트
ViewHolder
(class)

리스트
Adapter
(class)

만들어 줘야 하는것들

Recycler View 사용법

1. 리스트 아이템 레이아웃 만들기



Recycler View 사용법

1. 리스트 아이템 레이아웃 만들기

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:paddingVertical="16dp">
    <TextView
        android:id="@+id/txt_listItem_number"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:text="기본 값"
        android:textSize="20dp"

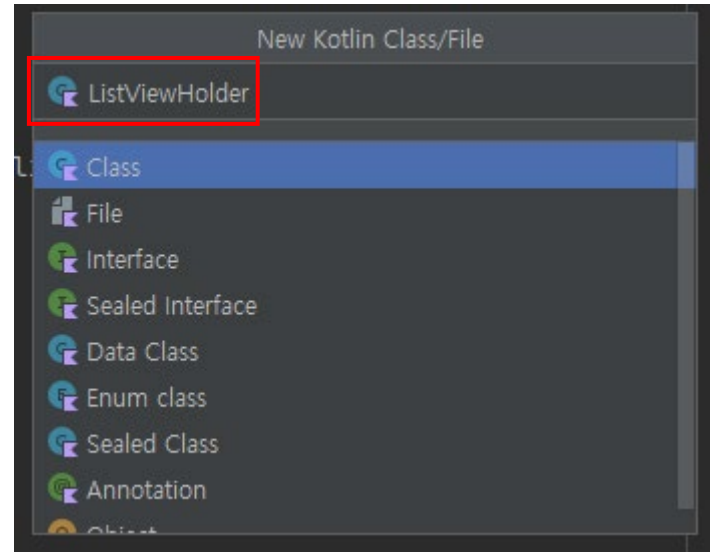
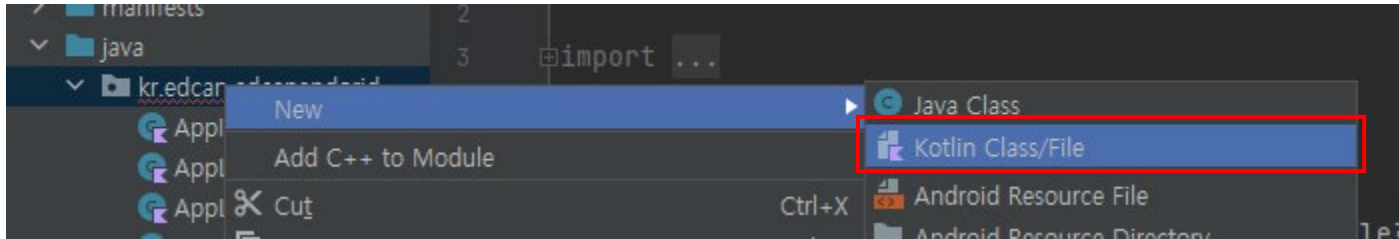
        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
    />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Recycler View - ViewHolder

- 리스트에서 아이템 하나를 클래스로 표현한 것
- 하나의 리스트 아이템에 있는 뷰들이 모여있다.
- 모든 ViewHolder은 **RecyclerView.ViewHolder** 추상 클래스를 상속 받는다.
- 객체 생성시 View를 하나 인자로 받는다.

Recycler View 사용법

2. ViewHolder 만들기



Recycler View 사용법

2. ViewHolder 만들기

```
class ListViewHolder(val row : View) : RecyclerView.ViewHolder(row) {  
    val number = row.findViewById<TextView>(R.id.txt_listItem_number)  
}
```

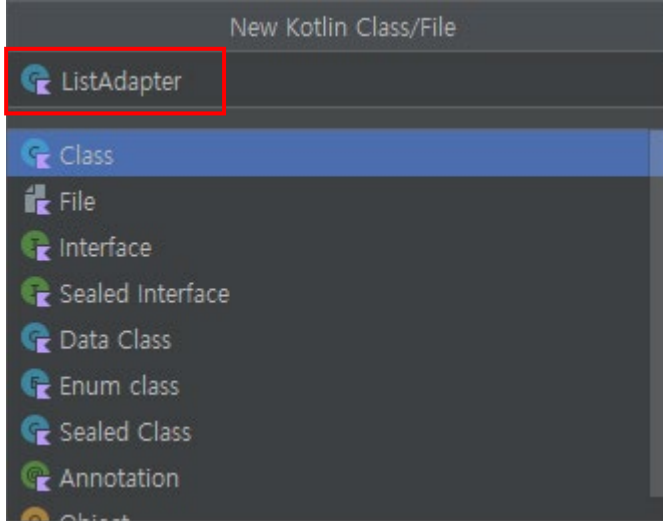
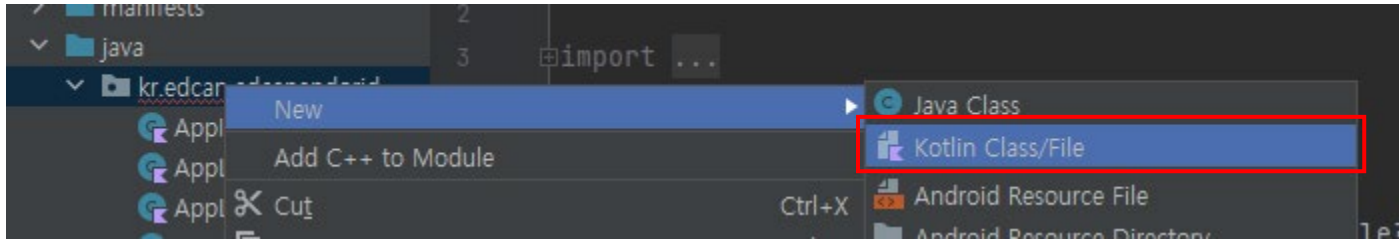
리스트 아이템의 뷰들을 가져와준다.

Recycler View - Adapter

- 데이터들을 각각의 리스트 아이템으로 만들어 준다.
- RecyclerView.Adapter 추상 클래스를 상속 받아준다.
이때 만들어준 ViewHolder클래스를 제네릭으로 넣어준다.

Recycler View 사용법

2. Adapter 만들기



Recycler View 사용법

2. Adapter 만들기

```
class ListAdapter(val items : List<Int>) : RecyclerView.Adapter<ViewHolder>() {  
    }  
}
```

리스트로 보여줄 데이터들

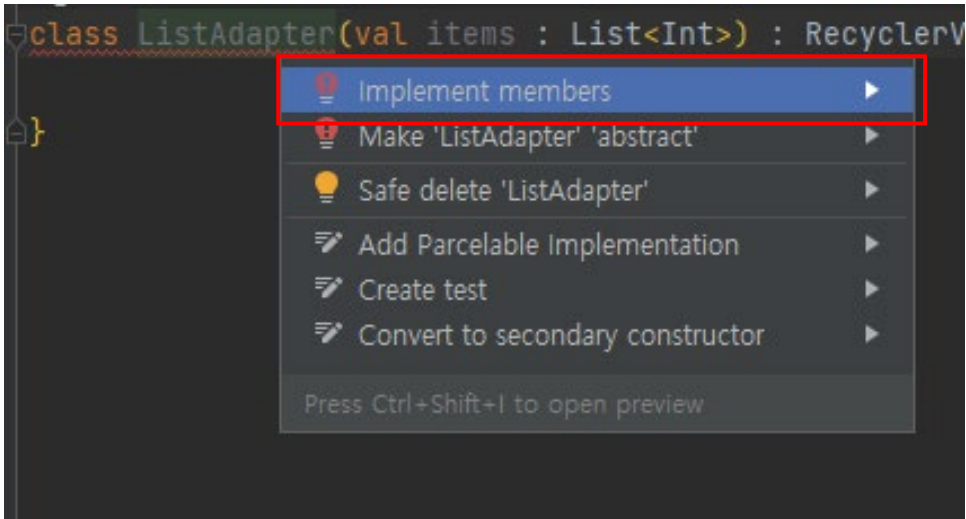
RecyclerView.Adapter를 상속

만들어준 ViewHolder

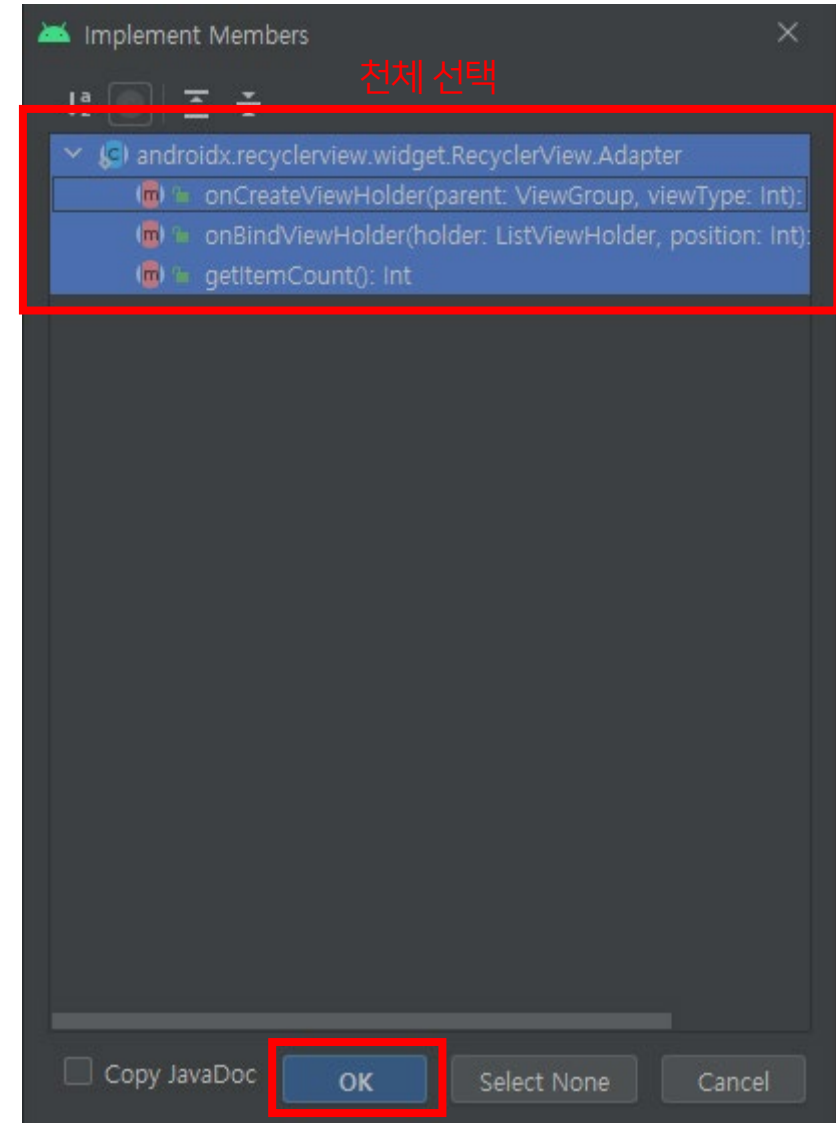
이 부분에서 에러가 난다.

Recycler View 사용법

2. 상속 받은 추상클래스 Implement 해주기



에러 난 부분에서 Alt + Enter



Recycler View 사용법

2. 상속 받은 추상클래스 Implement 해주기

```
class ListAdapter(val items : List<Int>) : RecyclerView.Adapter<ListViewHolder>() {  
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ListViewHolder {  
        TODO("Not yet implemented")  
    } 리스트 아이템별 ViewHolder 만드는 메서드  
  
    override fun onBindViewHolder(holder: ListViewHolder, position: Int) {  
        TODO("Not yet implemented")  
    } 리스트 아이템에 데이터를 채워넣어주는 메서드  
  
    override fun getItemCount(): Int {  
        TODO("Not yet implemented")  
    } 아이템의 전체 개수를 반환하는 메서드  
}
```

RecyclerView.Adapter 추상 클래스는 3개의 메서드를 구현 해줘야 한다.

Recycler View 사용법

2. 상속 받은 추상클래스 Implement 해주기

```
override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): ViewHolder {  
    val view = LayoutInflater.from(parent.context).inflate(R.layout.row_list_item, parent, false)  
    return ViewHolder(view)  
}
```

만들어준 레이아웃 파일

LayoutInflater는 뷰를 동적으로 만들어 주는 역할을 한다.

만들어준 뷰를 ViewHolder로 만들어서 반환한다.

Recycler View 사용법

2. 상속 받은 추상클래스 Implement 해주기

```
override fun onBindViewHolder(holder: ListViewHolder, position: Int) {  
    holder.number.text = items[position].toString()  
}
```

ViewHolder에 데이터를 채워 넣어 준다.

Recycler View 사용법

2. 상속 받은 추상클래스 Implement 해주기

```
override fun getItemCount(): Int {  
    return items.size  
}
```

데이터들의 전체 개수를 반환하다.

Recycler View 사용법

3. 리사이클러뷰와 어댑터를 연결해주기

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/list_main_club"
    android:layout_width="0dp"
    android:layout_height="0dp"

    tools:listitem="@layout/row_list_item"

    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
/>
```

activity_main.xml

```
val recyclerView : RecyclerView = findViewById(R.id.list_main_club)

val datas = listOf(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
val adapter = ListAdapter(datas)

recyclerView.adapter = adapter
recyclerView.layoutManager = LinearLayoutManager(this)
```

리스트가 보여지는 형식

EDCAN Andorid

1

2

3

4

5

6

7

8

9

10

Recycler View 사용법

리사이클러뷰 아이템 클릭 이벤트

```
class ListViewHolder(val row : View) : RecyclerView.ViewHolder(row) {  
    val number = row.findViewById<TextView>(R.id.txt_listItem_number)  
  
    init {  
        row.setOnClickListener {  
            Toast.makeText(row.context, "${number.text} 클릭", Toast.LENGTH_LONG).show()  
        }  
    }  
}
```

ViewHolder에서 들어온 view에 이벤트를 정의 해준다.

Recycler View 사용법

리사이클러뷰 데이터 추가하기

```
adapter.items.add(11)
```

 이때 items가 mutable list일 때 만 추가 가능하다.

```
adapter.notifyDataSetChanged()
```

 리스트를 새로 그린다.

RecyclerView 예제 만들기

완성 화면

YoutubeClone



노동요

sake L 조회수 100000000회

토스뱅크
데이터 설계사상

SLASH 21

조한기



토스 | SLASH 21 - 토스 서비스를 구성하는 서버 기술

토스 조회수 12123회



드르륵 안돼서 분필 부수고 던지는 한석원쌤

초록색연필 조회수 1523회



RecyclerView 예제 만들기

```
data class Youtube(  
    val title : String,  
    val thumbnail : String,  
    val profileImg : String,  
    val channelName : String,  
    val viewCount : Int,  
)
```

Youtube.kr

```
val youtubeData = mutableListOf<Youtube>(  
    Youtube(  
        title = "토스 | SLASH 21 - 프론트엔드 웹 서비스에서 우아하게 비동기 처리하기",  
        thumbnail = "https://i.ytimg.com/vi/FvRtoViuJ6g/hq720.jpg?sqp=-oaymwEcC0gCEMoBSFXyq4qpAw4IARUAAIhCG",  
        profileImg = "https://yt3.ggpht.com/5R_CRJL68b5Kt0TC07YYW70y99b_nMf1MhrDRSETg83kAsGfWqS7EFahyfXFV2n",  
        channelName = "토스",  
        viewCount = 100  
    ),  
    Youtube(  
        title = "토스 | SLASH 21 - 실무에서 바로 쓰는 Frontend Clean Code",  
        thumbnail = "https://i.ytimg.com/an_webp/edWbHp_k_9Y/mqdefault_6s.webp?du=3000&sqp=CLqSypIG&rs=A0n4",  
        profileImg = "https://yt3.ggpht.com/5R_CRJL68b5Kt0TC07YYW70y99b_nMf1MhrDRSETg83kAsGfWqS7EFahyfXFV2n",  
        channelName = "토스",  
        viewCount = 30000  
    ),  
    Youtube(  
        title = "토스 | SLASH 21 - 토스 서비스를 구성하는 서버 기술",  
        thumbnail = "https://i.ytimg.com/vi/KoL0bZ9A3Kc/hq720.jpg?sqp=-oaymwEcC0gCEMoBSFXyq4qpAw4IARUAAIhCG",  
        profileImg = "https://yt3.ggpht.com/5R_CRJL68b5Kt0TC07YYW70y99b_nMf1MhrDRSETg83kAsGfWqS7EFahyfXFV2n",  
        channelName = "토스",  
        viewCount = 12123  
    ),  
    Youtube(  
        title = "토스 | SLASH 21 - Micro-frontend React, 점진적으로 도입하기",  
        thumbnail = "https://i.ytimg.com/vi/DHPeeEvDbdo/hq720.jpg?sqp=-oaymwEcC0gCEMoBSFXyq4qpAw4IARUAAIhCG",  
        profileImg = "https://yt3.ggpht.com/5R_CRJL68b5Kt0TC07YYW70y99b_nMf1MhrDRSETg83kAsGfWqS7EFahyfXFV2n",  
        channelName = "토스",  
        viewCount = 4324  
    )  
)
```

RecyclerView 예제 만들기

```
<ImageView
    android:id="@+id/img_youtubeItem_thumbnail"
    android:layout_width="0dp"
    android:layout_height="wrap_content"

    android:adjustViewBounds="true"
    android:src="@drawable/youtube_default"

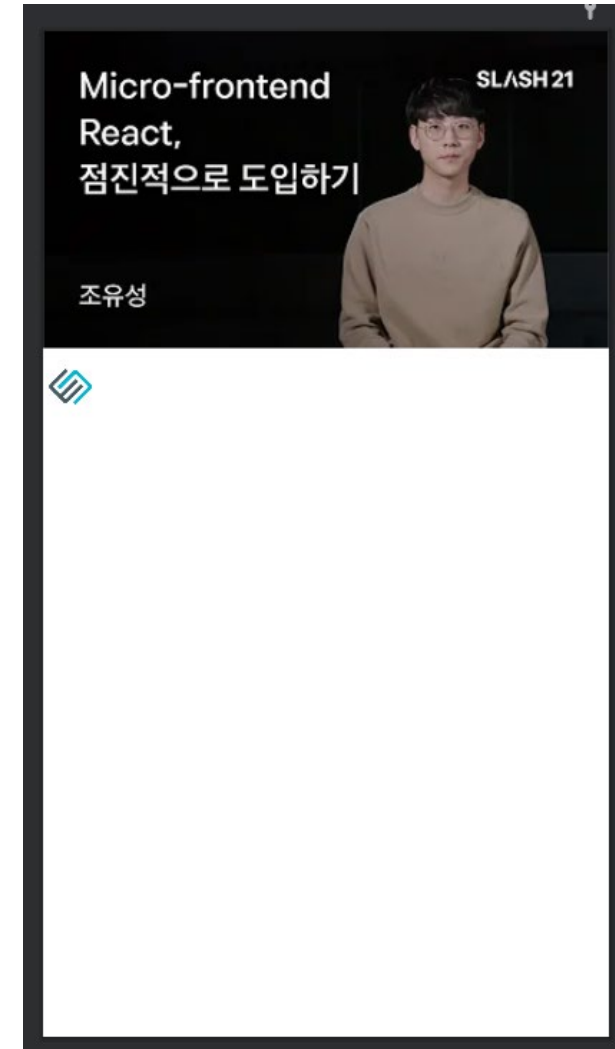
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
/>

<ImageView
    android:id="@+id/img_youtubeItem_channel"
    android:layout_width="40dp"
    android:layout_height="40dp"

    android:src="@drawable/edcan"

    android:layout_marginTop="8dp"

    app:layout_constraintTop_toBottomOf="@id/img_youtubeItem_thumbnail"
    app:layout_constraintStart_toStartOf="parent"
/>
```



RecyclerView 예제 만들기

<TextView

```
android:id="@+id/txt_youtubeItem_title"
android:layout_width="wrap_content"
android:layout_height="wrap_content"

android:text="대충 제목"

android:layout_marginHorizontal="12dp"

app:layout_constraintTop_toTopOf="@id/img_youtubeItem_channel"
app:layout_constraintStart_toEndOf="@id/img_youtubeItem_channel"
app:layout_constraintEnd_toEndOf="parent"

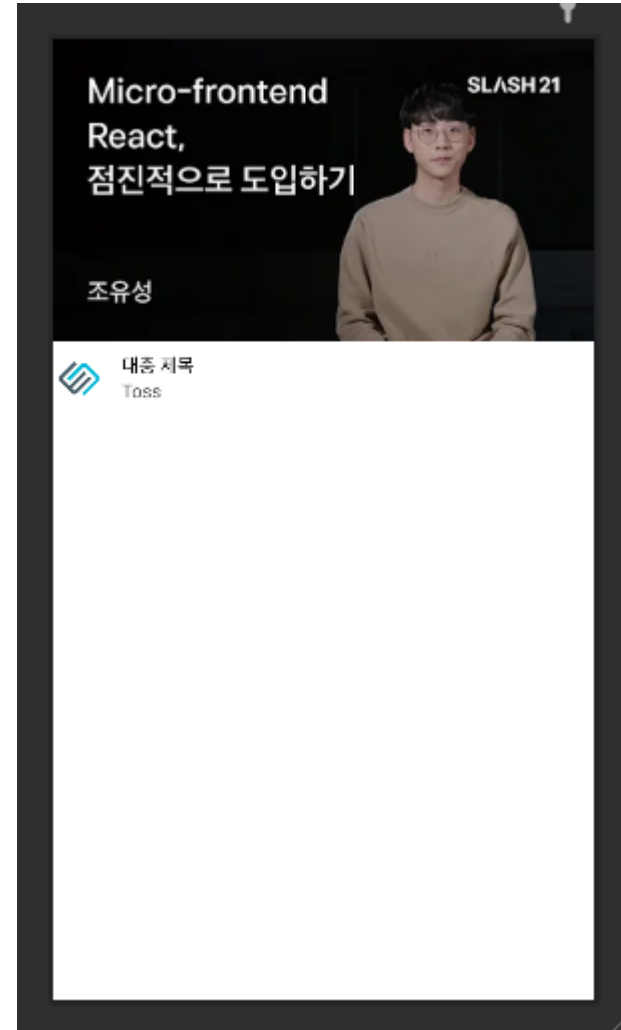
app:layout_constraintHorizontal_bias="0"
/>
```

<TextView

```
android:id="@+id/txt_youtubeItem_channelName"
android:layout_width="wrap_content"
android:layout_height="wrap_content"

android:text="Toss"
android:textColor="#575757"

app:layout_constraintTop_toBottomOf="@id/txt_youtubeItem_title"
app:layout_constraintStart_toStartOf="@+id/txt_youtubeItem_title"
/>
```



RecyclerView 예제 만들기

<TextView

android:id="@+id/txt_youtubeItem_viewCount"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="조회수 100회"

android:textColor="#575757"

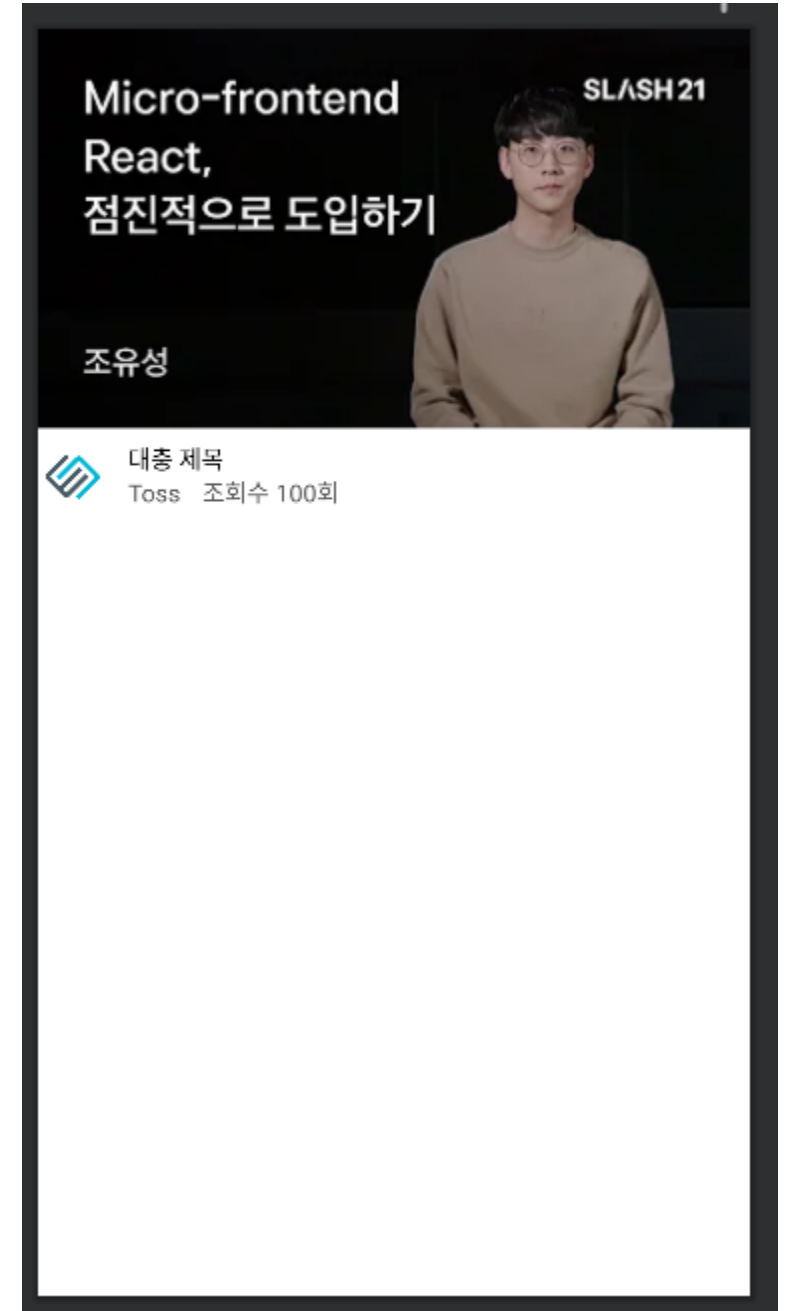
android:layout_marginHorizontal="12dp"

app:layout_constraintTop_toTopOf="@+id/txt_youtubeItem_channelName"

app:layout_constraintStart_toEndOf="@id/txt_youtubeItem_channelName"

/>

row_youtube_list_item.xml (3/3)



RecyclerView 예제 만들기

```
class YoutubeViewHolder(private val row : View) : RecyclerView.ViewHolder(row) {
    val thumbnailImg : ImageView = row.findViewById(R.id.img_youtubeItem_thumbnail)
    val channelImg : ImageView = row.findViewById(R.id.img_youtubeItem_channel)
    val title : TextView = row.findViewById(R.id.txt_youtubeItem_title)
    val channelName : TextView = row.findViewById(R.id.txt_youtubeItem_channelName)
    val viewCount : TextView = row.findViewById(R.id.txt_youtubeItem_viewCount)

    fun onBind(youtubeData : Youtube){
        title.text = youtubeData.title
        channelName.text = youtubeData.channelName
        viewCount.text = "조회수 ${youtubeData.viewCount}회"

        Glide.with(row.context).load(youtubeData.thumbnail).into(thumbnailImg)
        Glide.with(row.context).load(youtubeData.profileImg).into(channelImg)

        row.setOnClickListener {
            val intent = Intent(Intent.ACTION_VIEW, Uri.parse(youtubeData.url))
            row.context.startActivity(intent)
        }
    }
}
```

ViewHolder에서 데이터를 채워 넣는 함수를 만들어주는 게 더 좋다.

onBind함수는 어댑터에서 실행 시킨다.

YoutubeViewHolder.kt

RecyclerView 예제 만들기

```
<androidx.recyclerview.widget.RecyclerView
    android:id="@+id/list_main"
    android:layout_width="0dp"
    android:layout_height="0dp"

    tools:listitem="@layout/row_youtube_list_item"

    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
/>
```

activirt_main.xml

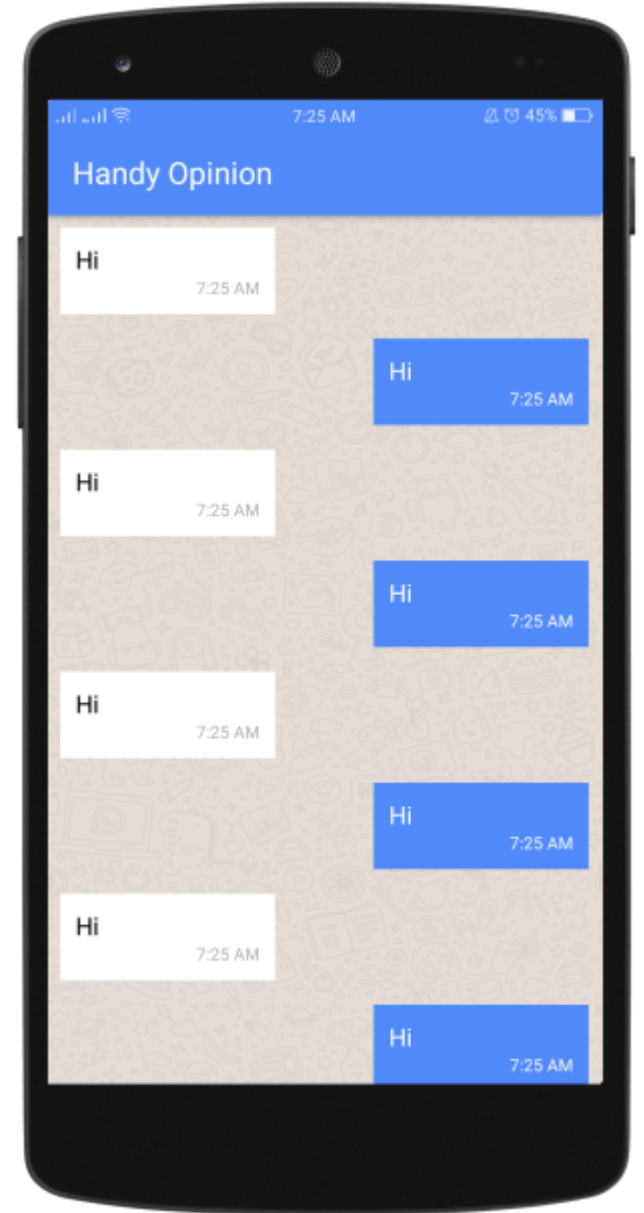
```
findViewById<RecyclerView>(R.id.list_main).apply{
    adapter = YoutubeAdapter(youtubeData)
    layoutManager = LinearLayoutManager(this@MainActivity)
}
```

MainActivity.kt

RecyclerView 멀티 리스트 레이아웃

RecyclerView 멀티 리스트 레이아웃

- 하나의 리스트지만 리스트 아이템을 다양하게 만들어줄 수 있다.
- 새로운 layout과 ViewHolder를 만들어 줘야 한다
- onCreateViewHolder 메소드에서 데이터에 따라서 ViewHolder를 다르게 만들어준다.



예제 앱 보기



예제 앱 만들기

```
dependencies {  
    //Circle ImageView  
    implementation 'de.hdodenhof:circleimageview:3.1.0'  
  
    //Glide  
    implementation "com.github.bumptech.glide:glide:4.12.0"  
}
```

원형 이미지 만들기

url로 이미지 가져오기

app/build.gradle

예제 앱 만들기

```
data class TalkData(  
    val writer : String,  
    val content : String,  
    val profileImg : String  
)
```

TalkData.kt

채팅 데이터를 저장하는 클래스

```
val talkDataList = mutableListOf<TalkData>(  
    TalkData("박희찬", "Hello World", "https://i.imgur.com/bMel0Ho.jpg"),  
    TalkData("박희찬", "안녕 인수야", "https://i.imgur.com/bMel0Ho.jpg"),  
    TalkData("장인수", "저는 장인수 입니다 1", "https://i.imgur.com/GbzdfiJ.png"),  
    TalkData("박희찬", "그래 인수야", "https://i.imgur.com/bMel0Ho.jpg"),  
    TalkData("장인수", "저는 장인수 입니다 2", "https://i.imgur.com/GbzdfiJ.png"),  
    TalkData("장인수", "저는 장인수 입니다 3", "https://i.imgur.com/GbzdfiJ.png"),  
    TalkData("장인수", "저는 장인수 입니다 4", "https://i.imgur.com/GbzdfiJ.png"),  
    TalkData("박희찬", "알겠어 인수야", "https://i.imgur.com/bMel0Ho.jpg"),  
    TalkData("장인수", "저는 장인수 입니다 5", "https://i.imgur.com/GbzdfiJ.png"),  
    TalkData("박희찬", "ㅇㅋ 인수야", "https://i.imgur.com/bMel0Ho.jpg"),  
    TalkData("박희찬", "응 인수야", "https://i.imgur.com/bMel0Ho.jpg"),  
    TalkData("장인수", "저는 장인수 입니다 6", "https://i.imgur.com/GbzdfiJ.png"),  
    TalkData("박희찬", "그래 인수야", "https://i.imgur.com/bMel0Ho.jpg"),  
)
```

채팅 데이터

예제 앱 만들기

1. xml 만들기

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
  <solid android:color="#00b8d4"/>
  <corners
    android:bottomLeftRadius="12dp"
    android:topLeftRadius="12dp"
    android:topRightRadius="12dp"
  />
</shape>
```

drawable/sp_talk_bg_me.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"

    android:paddingVertical="12dp"
    android:paddingHorizontal="12dp"
    >

    <TextView
        android:id="@+id/txt_talkMe_content"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

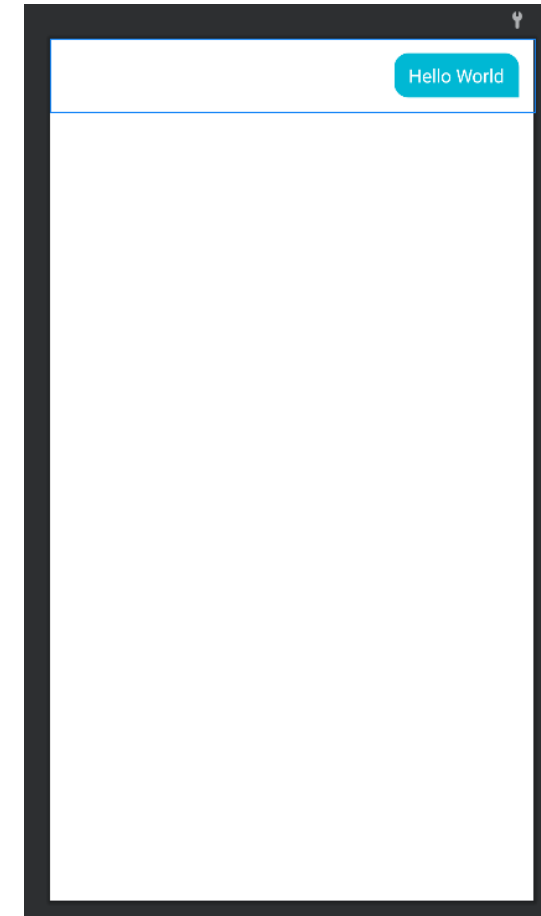
        android:text="Hello World"
        android:textColor="@color/white"
        android:textSize="16sp"

        android:background="@drawable/sp_talk_bg_me"

        android:paddingHorizontal="12dp"
        android:paddingVertical="8dp"

        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
    />

</androidx.constraintlayout.widget.ConstraintLayout>
```



layout/row_talk_me.xml

예제 앱 만들기

1. xml 만들기

```
<?xml version="1.0" encoding="utf-8"?>
<shape xmlns:android="http://schemas.android.com/apk/res/android">
  <solid android:color="#ffffff"/>
  <corners
    android:bottomLeftRadius="12dp"
    android:bottomRightRadius="12dp"
    android:topRightRadius="12dp"
  />
</shape>
```

drawable/sp_talk_bg_you.xml

예제 앱 만들기

1. xml 만들기

```
<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    xmlns:app="http://schemas.android.com/apk/res-auto"

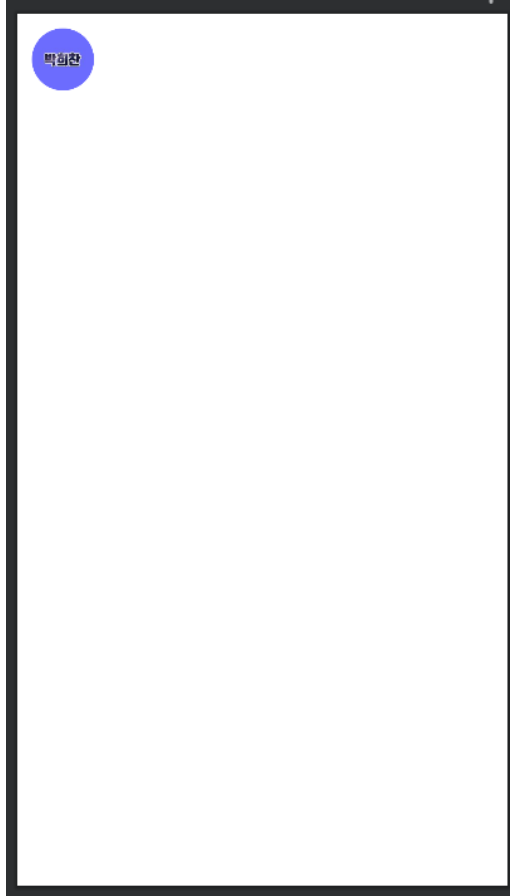
    android:paddingVertical="12dp"
    android:paddingHorizontal="12dp"
    >

    <de.hdodenhof.circleimageview.CircleImageView
        android:id="@+id/img_talkYou_profile"
        android:layout_width="52dp"
        android:layout_height="52dp"

        android:src="@drawable/profile_img"

        app:layout_constraintTop_toTopOf="parent"
        app:layout_constraintBaseline_toBottomOf="parent"
        app:layout_constraintStart_toStartOf="parent"
    />

</androidx.constraintlayout.widget.ConstraintLayout>
```



예제 앱 만들기

1. xml 만들기

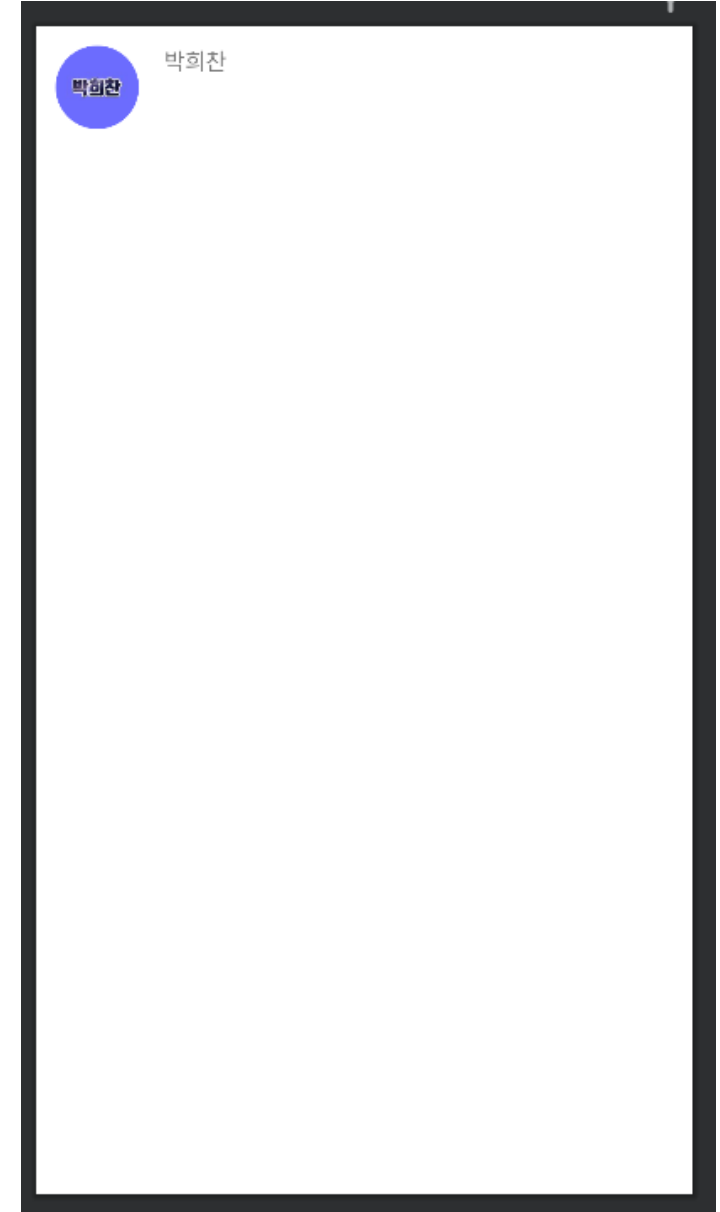
```
<TextView
    android:id="@+id/txt_talkYou_writer"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:text="박희찬"

    android:layout_marginStart="16dp"

    app:layout_constraintTop_toTopOf="@id/img_talkYou_profile"
    app:layout_constraintStart_toEndOf="@id/img_talkYou_profile"
/>
```

layout/row_talk_you.xml (2/3)



이제야 나다

<TextView

```
android:id="@+id/txt_talkYou_content"
android:layout_width="wrap_content"
android:layout_height="wrap_content"

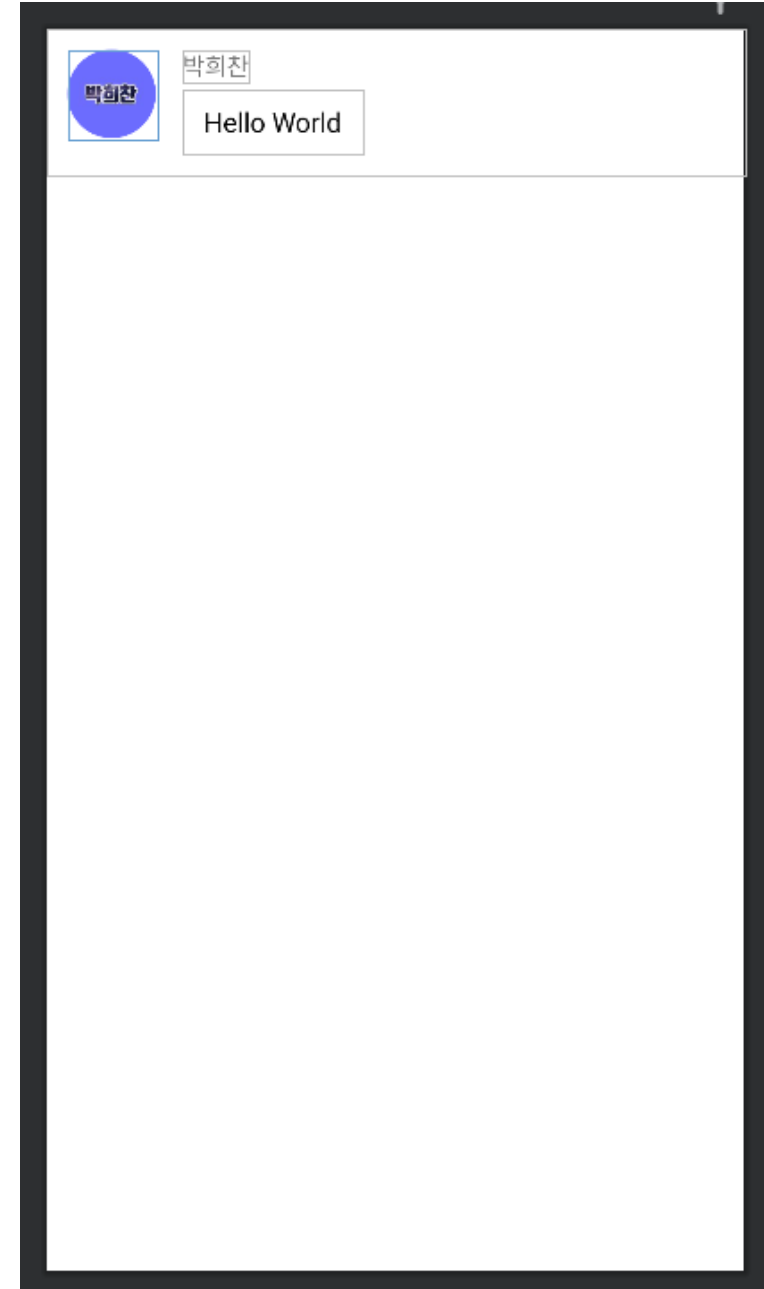
android:text="Hello World"
android:textColor="@color/black"
android:textSize="16sp"

android:background="@drawable/sp_talk_bg_you"

android:paddingHorizontal="12dp"
android:paddingVertical="8dp"

android:layout_marginTop="4dp"

app:layout_constraintTop_toBottomOf="@id/txt_talkYou_writer"
app:layout_constraintStart_toStartOf="@+id/txt_talkYou_writer"
/>
```



예제 앱 만들기

2. viewHolder만들기

```
class TalkMeViewHolder(val view : View) : RecyclerView.ViewHolder(view) {  
    val content : TextView = view.findViewById(R.id.txt_talkMe_content)  
  
    fun onBind(talkData: TalkData) {  
        content.text = talkData.content  
    }  
}
```

TalkMeViewHolder.kt

예제 앱 만들기

2. viewHolder만들기

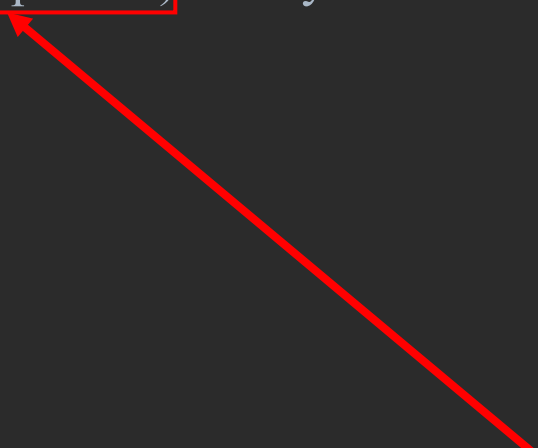
```
class TalkYouViewHolder(val view : View) : RecyclerView.ViewHolder(view) {  
    val profileImg : ImageView = view.findViewById(R.id.img_talkYou_profile)  
    val writer : TextView = view.findViewById(R.id.txt_talkYou_writer)  
    val content : TextView = view.findViewById(R.id.txt_talkYou_content)  
  
    fun onBind(talkData: TalkData) {  
        content.text = talkData.content  
        writer.text = talkData.writer  
  
        Glide.with(profileImg).load(talkData.profileImg).into(profileImg)  
    }  
}
```

TalkYouViewHolder.kt

예제 앱 만들기

3. Adapter 만들기

```
class TalkListViewAdapter(val talkDataList : MutableList<TalkData>)
: RecyclerView.Adapter<RecyclerView.ViewHolder>() {
    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int): RecyclerView.ViewHolder {
        return if(viewType == 1) {
            val view = LayoutInflater.from(parent.context)
                .inflate(R.layout.row_talk_me, parent, false)
            TalkMeViewHolder(view)
        } else {
            val view = LayoutInflater.from(parent.context)
                .inflate(R.layout.row_talk_you, parent, false)
            TalkYouViewHolder(view)
        }
    }
}
```



viewType 값에 따라서
서로 다른 레이아웃과
viewHolder를 만들어준다.

예제 앱 만들기

3. Adapter 만들기

```
class TalkListViewAdapter(val talkDataList : MutableList<TalkData>)
    : RecyclerView.Adapter<RecyclerView.ViewHolder>() {

    ...

    override fun getItemViewType(position: Int): Int {
        return if(talkDataList[position].writer == "박희찬") 1 else 2
    }
}
```

TalkListViewAdapter.kt (2/3)

viewType 값은 getItemViewType 메서드가 반환하는 값이다.
getItemViewType 메서드를 재정의 해준다.

예제 앱 만들기

3. Adapter 만들기

```
class TalkListViewAdapter(val talkDataList : MutableList<TalkData>)
    : RecyclerView.Adapter<RecyclerView.ViewHolder>() {

    override fun onBindViewHolder(holder: RecyclerView.ViewHolder, position: Int) {
        if(holder is TalkMeViewHolder)
            (holder as TalkMeViewHolder).onBind(talkDataList[position])
        else
            (holder as TalkYouViewHolder).onBind(talkDataList[position])
    }

    override fun getItemCount(): Int = talkDataList.size
}
```

기본 holder는 onBind 메서드가 없기 때문에 형변환을 해줘야 한다.

예제 앱 만들기

```
<androidx.recyclerview.widget.RecyclerView
```

```
    android:id="@+id/recycler_main_chat"
```

```
    android:layout_width="0dp"
```

```
    android:layout_height="0dp"
```

```
    tools:listitem="@layout/row_talk_you"
```

```
    app:layoutManager="androidx.recyclerview.widget.LinearLayoutManager"
```

```
    android:background="#f1f3f5"
```

```
    app:layout_constraintTop_toTopOf="parent"
```

```
    app:layout_constraintBottom_toBottomOf="parent"
```

```
    app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintEnd_toEndOf="parent"
```

```
</>
```

예제 앱 만들기

4. 리사이클러뷰와 어댑터 연결 해주기

```
val adapter = TalkListViewAdapter(talkDataList)  
findViewById<RecyclerView>(R.id.recycler_main_chat).adapter = adapter
```