

## [사인페 기말부분 필기 정리노트] 10.20~

참고) 8장은 프래그먼트인데, 프래그먼트는 액티비티를 흥내낸 것이므로 먼저 강의하면 효율이 떨어짐!

### #액티비티 생명주기 (7장 ppt.88)

-이 함수들은 다 외워야 한다. 애네는 다 액티비티에 정의된 메소드 들이다. 우리가 이것 사용한다고 하면 다 오버라이드 해서 사용하는 것이다.

-인텐트 : 어떤 컴포넌트에서 다른 컴포넌트를 실행시키고 싶을 때.. 매개체

ppt.91

-onCreate() : 앱이 처음 실행될 때 호출됨. 내부적으로 생성하고 우리에게 초기화 하라고 하는 거기 때문에 이때 우리 액티비티는 화면에 보이지 않는다.

-onStart()상태 끝나면 액티비티가 화면에 보인다.

-onResume()호출 끝나면 이벤트 처리까지 가능한 상태. (화면도 보이고)

-백버튼 누르면 onPause() – onStop() – onDestroy() : 이 액티비티는 액티비티 매니저의 관리 대상이 아니게 됨.

근데 메모리에서 액티비티 객체가 사라진건 아님.

-Landscape방향으로 바뀌보면, onPause() – onStop() – onDestroy() 뜨면서 화면 전환 순간 아까의 액티비티 사라짐. 그리고 화면 전환 후의 새로운 onCreate() – onStart() – onResume() 으로 액티비티가 새로 생김. 이 전꺼 저장해서 불러옴. // 아이패드처럼 회전하면 완전 레이아웃 바뀌는건 켈러파이어, 화면 방향 바껴도 재시작 안하게끔 요청도 가능함. 후에 배울 예정

### #액티비티의 life cycle을 확인해보자.(1)

1. DialogActivity를 하나 더 만든다.(MainActivity에서 버튼을 눌러서 다이얼로그 띄워볼것임)
2. (DialogActivity)클릭 리스너 하나 단다

#### 3. (MainActivity에서)다른 액티비티 띄울 때는 startActivity()라는 메소드 사용.

이 메소드의 매개변수는 실행하고자 하는 컴포넌트의 정보를 담은 Intent 객체

인텐트를 건내야 하는데, 지금은 그냥 ()안에서 바로 생성함.

```
binding.toDialogActivityButton.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        startActivity(new Intent( packageContext: MainActivity.this, DialogActivity.class));  
    }  
});
```

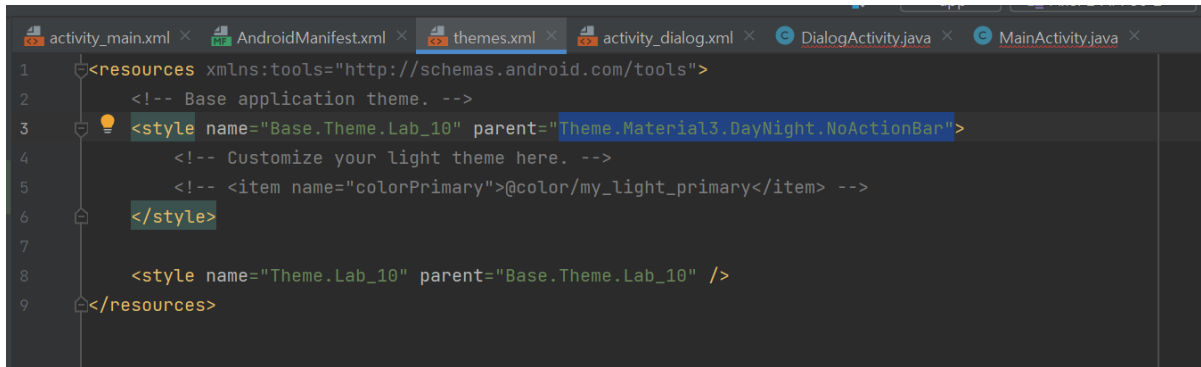
```
// Intent intent = new Intent(MainActivity.this, SecondActivity.class);
```

```
startActivity(intent);
```

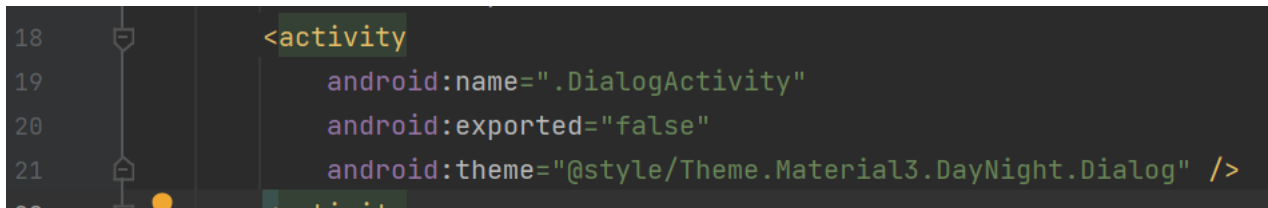
4. (DialogActivity)버튼 단다. 버튼 누르면 DialogActivity가 종료되도록. 그러면 그 전에 실행된 MainActivity의 화면이 보이겠죠? dialog형태 Activity 띄우면, 기존의 activity는 onPause 상태가 됨.
5. 코드에서 백 버튼을 누른것과 같은 효과를 내는 액티비티의 메소드 : finish()

### #themes.xml 스타일도 상속이 가능

NoActionBar : 상단에 툴바(액션바)가 나오지 않는 style



### Manifest.xml



특정 액티비티에 대해서는 내가 다른 테마(다이얼로그 스타일)를 적용하겠다.

이 코드의 예제론, 이 액티비티가 다이얼로그 형식으로 보이게끔 해줌!

### #액티비티가 일부 보일 때도 있다.

DialogActivity가 실행되면, 화면에는 MainActivity보이는데 이벤트 처리 위한 포커스는 DialogActivity 에 가있음.

메인은 pause 상태가 된 것. resume이 실행되면 그때부터 이벤트 처리가 가능하다.

즉, 다른 액티비티를 띄우면 원래는 내가 안보이는데, 이렇게 일부 보일 수도 있다. 이 때 MainActivity는 Stop이 아니라 pause까지 가서 대기할 수 있다.

DialogActivity의 Finish 버튼 누르면 다시 MainActivity의 resume이 호출되어 resumed 상태로 간다.

### #액티비티의 life cycle을 확인해보자.(2)

1. SecondActivity를 하나 더 만든다. (1)처럼 리스너 달아주고,...
2. MainActivity에 버튼 하나 더 만든다. (to\_second\_activity\_button)

3. 클릭 리스너 복사해서 달아준다. 실행은 세컨액티비티 하게끔 바꿔주고.
4. secondActivity 누르면 Main이 stop(완전히 가려서 안보임)까지 간다. 근데 finish누르면 Main이 다시 resumed 상태로 옴.

### #액티비티가 갑자기 강제종료되는 상황

이유? : 스마트폰 하드웨어의 사양이 떨어지므로 메모리(리소스)가 부족해지면 기존에 죽지 않은 앱들을 강제로 죽임.  
resumed상태의 액티비티는 강제종료의 우선순위에서 가장 떨어져 있음!

스마트폰의 Overview에 있는 애들은 다 stopped 상태에 있는 애들이다.

꼭 destroy까지 안가고 Stopped 상태에서 바로 죽어버릴 수도 있구나.. pause에서도 죽을 순 있지만 경험하기가 상당히 어려울 것. → 우리는 이 상황들에서 내 액티비티의 상황을 백업하는 코드를 넣어줘야 함

Ex 사용가능한상황) 지도앱 : 리즘드 상태에는 지피에스 정보를 계속 끌어다 씬. 화면이 안보이면 잠시 pause나 stop에서 지피에스 연결을 잠시 끊을 수 있다.

### #(인텐트에 의해 컴포넌트 실행 시) Extra data 보내기

Intent에 **putExtra()**라는 메소드가 있음.

putExtra()이용하면 data직렬화 하지 않아도 key값을 이용해 쉽게 value 넣을 수 있음.

```
.putExtra("extra", "from main activity")
```

Intent를 받은 Activity에서는 .. (세컨 액티비티겠지) key값 이용해 찾아서 사용함.

getXXXExtra( ) 함수

```
binding.textViewFormMainActivity.setText(getIntent().getStringExtra( name: "extra"));  
//getIntent()는 액티비티의 메소드. 인텐트를 받아온다. 그리고 그 인텐트에 있는 엑스트라(키값)를 가져온다.
```

### # startActivityResult() – 결과 되돌리기 / onActivityResult()

starActivity와는 다르게 **startActivityResult()** 는 결과를 바라면서 액티비티를 띄우는 함수이다!

```
//이걸 누르면 똑같이 세컨 액티비티를 띄울건데, 그냥 단순히 띄우는게 아니라, 띄우면서 결과를 기대함
binding.toSecondActivityForResultButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        startActivityForResult(new Intent( packageContext: MainActivity.this, SecondActivity.class).
            putExtra( name: "extra", value: "from main activity"), requestCode: 1000);
        //내가 여러개의 요청을 보낼 수 있기 때문에 어떤 요청에 대해 식별자(번호표)를 하나 붙인다.
```

-requestCode : 어떤 대답에 대한 요청이 왔는지 요청을 식별하기 위한 번호

대답을 받으려면 **onActivityResult()** 메소드를 오버라이드 해놓아야! 콜백 메소드다.

```
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    //무슨 요청에 대한 대답인지, 상수, 데이터
}
```

대답이 도착하면 이 데이터로 무언갈 하고싶은데 이 메소드는 원래 부모에 있는거라, 부모꺼는 코드가 정해져 있으므로 오버라이드 해주어야 한다. 결과값도 Intent로 만들어서 돌려주면 된다. 밑에 코드가 오버라이드 한 것

```
@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if(requestCode==1000&&resultCode==RESULT_OK){
        binding.resultFormSecondActivityTextView.setText(data.getStringExtra( name: "result"));
        data.getStringExtra( name: "result");
    }
    //무슨 요청에 대한 대답인지, 상수, 데이터
}
```

근데 여기서, 여러 요청에 대한 정보들이 하나의 함수로 들어오기 때문에 If, else if, else if.. 이렇게 하나의 함수 안에 다 짜야한다. 우리는 AppCompatActivity를 상속받고 있는데 AppCompatActivity에선 이게 마음에 들지 않아서 새로운 방법을 사용한다. → 클릭 리스너가 각각의 요청에 대한 콜백을 준비해 놓는 것 처럼, **외부 요청에 대해 각각의 콜백 함수를 준비해서 처리한다.**

## [AppCompatActivity]

- 확장 라이브러리
- 기존의 onActivityResult에서 모든 로직을 짜야 하는 게 맘에 안든다.(요청과 응답 형식)
- 외부 요청을 고유한 callback 형식으로 변경

**#ActivityResultLauncher<I> (Oncreate 함수 안에)**

```

ActivityResultLauncher<Intent> launcher = registerForActivityResult(new ActivityResultContracts.StartActivityForResult(),
    new ActivityResultCallback<ActivityResult>() {
        @Override
        public void onActivityResult(ActivityResult o) {
            Log.d(tag: "CVJ", msg: "onActivityResult()-callback");
            if(o.getResultCode()==RESULT_OK){
                binding.resultFormSecondActivityTextView.setText(o.getData().getStringExtra(name: "result"));
            }
        }
    });

```

-첫번째 매개변수(Contract객체): 액티비티가 외부에 요청하는 일의 종류 다양함. 어떤 목적으로 액티비티를 띄우냐? 내가 등록하려는 리스너가 어떤 종류의 일인지. ActivityResultContracts. 뒤가 일의 종류임! (여기선 다른 액티비티 띄우고 결과 돌려받음)

-두번째 매개변수(Callback객체): 그 일의 응답이 왔을 때 (결과가 오면) 뭐하고싶니? Callback메소드넣어주면됨, 리스너처럼 익명클래스

-ActivityResultLauncher객체는 **registerForActivityResult()** 함수로 만들어짐

**#launcher** StartActivityForResult()호출이랑 같아짐 == 다른 액티비티를 뜨게(런치) 해준다

런처를 통해 요청을 언제 할지는 우리가 결정하는 것.

```

//이걸 누르면 똑같이 세컨 액티비티를 띄울건데, 그냥 단순히 띄우는게 아니라, 띄우면서 결과를 기대함
binding.toSecondActivityForResultButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        launcher.launch(new Intent(packageContext: MainActivity.this, SecondActivity.class)
            .putExtra(name: "extra", value: "from main activity"))
    }
});

```

//런처라는 애는 <I>를 기대하는데 우리는 I를 intent로 설정했음

//여기선 requestCode 내가 만들 필요 없음

11.03

**#인텐트에 'flag'라는 요소도 넣을 수 있음.**

-언제 flag사용? : Activity 호출하다보면 발생하는 Activity의 중복문제나 흐름을 제어하고 싶을 때

ex) 회원가입 처럼 a가 b를 띄우고, b가 c를, c가 d를 띄우고 결과를 d가 a에게 돌려주는..

→ intent에 대한 결과 응답의 책무를 다음 activity에 넘길 수 있음

1. (MainActivity) 에서 SecondActivity로 intent 보냄

```
//이걸 누르면 똑같이 세컨 액티비티를 띄울건데, 그냥 단순히 띄우는게 아니라, 띄우면서 결과를 기대함
binding.toSecondActivityForResultButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        launcher.launch(new Intent( packageContext: MainActivity.this, SecondActivity.class)
            .putExtra( name: "extra", value: "from main activity")
            .addFlags(Intent.FLAG_ACTIVITY_NO_HISTORY));
    }
});
```

//result data를 받을 때 생성된 모든 activity를 destroy() 호출함. 자동으로 세컨 액티비티의 생명주기 destroy

2. (SecondActivity) 에서 ThirdActivity로 intent 보냄

```
binding.finishWithForwardResultButton.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View view) {
        startActivity(new Intent( packageContext: SecondActivity.this, ThirdActivity.class)
            .putExtra( name: "extra", value: "from main activity")
            .addFlags(Intent.FLAG_ACTIVITY_FORWARD_RESULT));
    }
});
```

// C에서 setResult를 했고 A까지 가야 한다면 B에서 setResult를 하면 안됨.

// B에서 굳이 finish()하지 않은 이유는, A에서 FLAG\_ACTIVITY\_NO\_HISTORY 를 썼기 때문에~

3. (ThirdActivity) 에서 result data 생성 후 finish() 호출. (=finish()전에 setResult 놓아야 한다.)

```
binding.finishWithResultButton.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View view) {
        setResult(RESULT_OK,new Intent().putExtra( name: "result", value: "from third activity"));
        finish();
    }
});
```

<주의사항 정리>

C(setResult and finish) -> B(finish) -> A( C의 intent를 받게됨)

C(setResult and finish) -> B(setResult and finish) -> A( B의 intent를 받게됨)

#Activity정보를 임시로 저장했다가 호출하기

가로로 화면 전환했을 때 원래 액티비티 종료되는데, 메모리에 액티비티 상태를 잠시 저장해 뒀다가 복원

(영구 저장x 앱 종료가 아니라 액티비티 종료시 임시저장)

- bundle(map): 안드로이드에서 만든 자료구조. 컴포넌트의 데이터를 저장하기 위한 Map객체

(ex) 화면을 가로로 전환했을 때도 입력한 정보가 남아있게 하고 싶다..

## #onSaveInstanceState 상태 저장

```
@Override
protected void onSaveInstanceState(@NonNull Bundle outState) {
    //상태 저장 가능한 시점 제공
    Log.d( tag: "cyj", msg: "onSaveInstanceState()");
    super.onSaveInstanceState(outState);

    outState.putString("save", binding.sampleEditText.getText().toString());
    //키 값, value
}
```

(MainActivity는 하나의 인스턴스이다)

## #onRestoreInstanceState 상태 복원

```
@Override
protected void onRestoreInstanceState(@NonNull Bundle savedInstanceState) { //위의 번들이 다시 들어온다
    //상태 복원 가능한 시점 제공
    Log.d( tag: "cyj", msg: "onRestoreInstanceState()");
    super.onRestoreInstanceState(savedInstanceState);

    binding.sampleEditText.setText(savedInstanceState.getString( key: "save"));
    //savedInstanceState.getString("save"); 이걸 위의 setText안에 넣어줬음
}
```

로그로 확인가능 : Stop()뒤에 저장, Start()뒤에 복원

-액티비티가 처음 실행될 때는 Bundle이 Null이다.

```
20 public class MainActivity extends AppCompatActivity {
    8 usages
21     private ActivityMainBinding binding;
22
23     @Override
24     protected void onCreate(Bundle savedInstanceState) {
25         Log.d( tag: "cyj", msg: "onCreate()");
26         super.onCreate(savedInstanceState);
27         binding = ActivityMainBinding.inflate(getLayoutInflater());
28         setContentView(binding.getRoot());
    }
```

## #화면 전환해도 새로 시작하지 말아주세요 속성

1. manifest에 원하는 activity에 android:configChanges="screenSize|orientation"를 추가한다.

// 화면 '크기' 및 화면 '방향' 변경이 발생할 때 액티비티 유지

```
25 <activity
26     android:name=".MainActivity"
27     android:configChanges="screenSize|orientation"
28     android:exported="true">
29     <intent-filter>
```

2. 그 대신에 그에 따른 side effect는 니가 처리해!

액티비티에 예약된 함수 호출해주고 우리는 그것을 오버라이드 해야한다.

```
@Override
public void onConfigurationChanged(@NonNull Configuration newConfig) { //너의 환경에 대한 구성이 바꼈다 콜백
    Log.d( tag: "cyj|", msg: "onconfigurationChanged()");
    super.onConfigurationChanged(newConfig);
}
```

1번 속성 추가 후, 화면 전환하면 onConfigurationChanged() 함수가 호출된다. 만약 화면의 방향이 전환되었을 때 수행해야 하는 작업이 있는 경우 이 함수에 추가하면 된다.

화면 전환하면 stop()-pause()-destroy() 이런거 안뜨고 애 로그만 뜬.

# **TextView를 상속받은 아이들(EditText)**은 아이디 달아두기만 하면 자동으로 번들에 save 후 restore 해줌.(super)

➔ 따라서 텍스트뷰의 자식이 아닌데 이 상태를 복원해야 할 때 위와 같은 코드들을 사용한다.

## [파이어베이스]

프로젝트를 하나 만들면 하나의 서버가 생기는 것이다.

-데이터베이스 : 실시간 데이터베이스, cloud firestore

-storage : 저장소. 이미지 파일 같은거 저장. 이미지 자체가 아니라 url이 들어있음.

-인증 : 회원관리

## #서명 알아내는 법

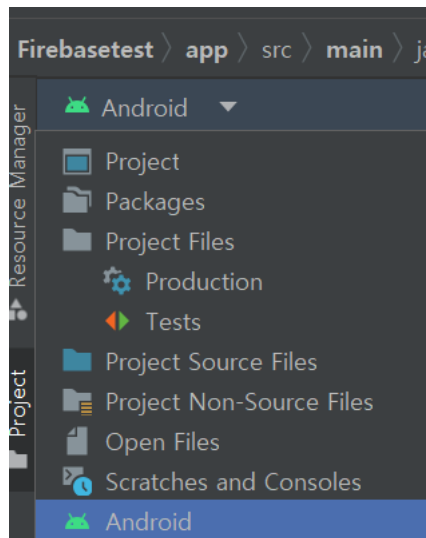
1.안드로이드 터미널에



gradle signingReport + ctrl enter 해서 SHA1를 복사해서 붙여넣기

## 2. 프로젝트로 바꾸기

Json 파일 복사해서 app에 붙여넣기



## 3.파이어베이스 sdk버전:kotlin선택

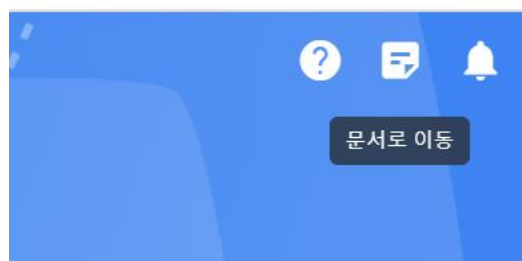
## 4.다른애들이 서버에 붙으려면

json파일 복사해서 집어넣고, 디지털 지문 추가해서 각각 자기꺼 등록해줌

+bom에서 알아서 라이브러리 버전 맞춰줌

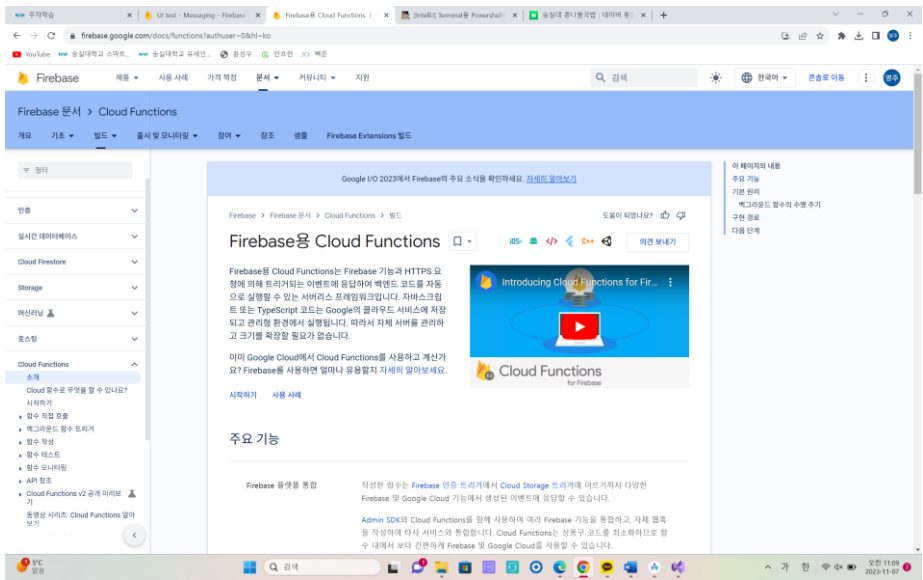
```
51
52 dependencies { this: DependencyHandlerScope
53     // Import the Firebase BoM
54     implementation(platform("com.google.firebase:firebase-bom:32.5.0"))
```

## #개발자 문서 같은 느낌

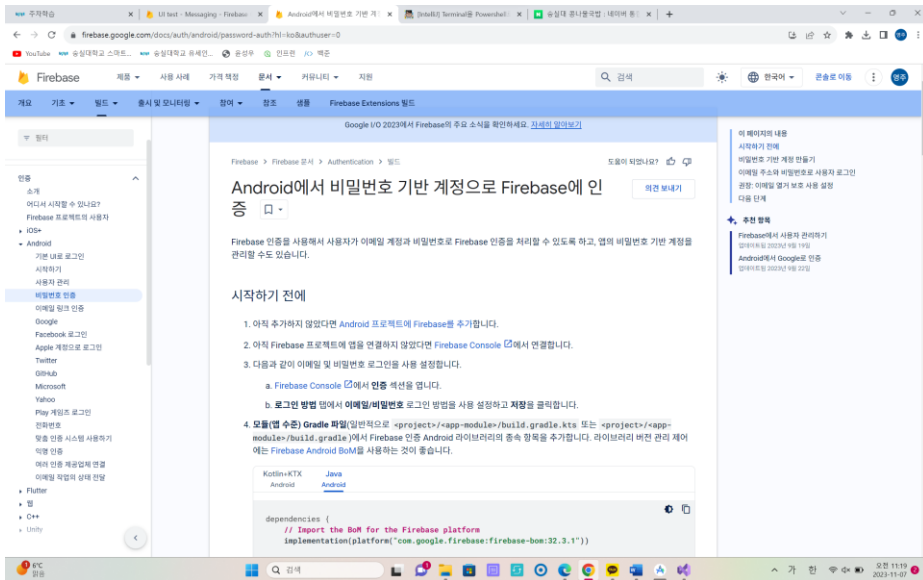


## #메세징 서비스 함수...? 가 언제 호출되느냐

질문을 너무 많이 받아서 미리 얘기함



## #교수님이 PW 프로젝트에서 참고하신 문서



11.10 수업 필기

10.14 수업 필기

디바이스 익스플로러에서 Data-data-내 프로젝트 이름 누르면 -files라는 폴더에 res폴더의 이미지 업로드 한다.

로그인을 안한 사람이 업로드하면 public이라는 폴더가생김

사용자의

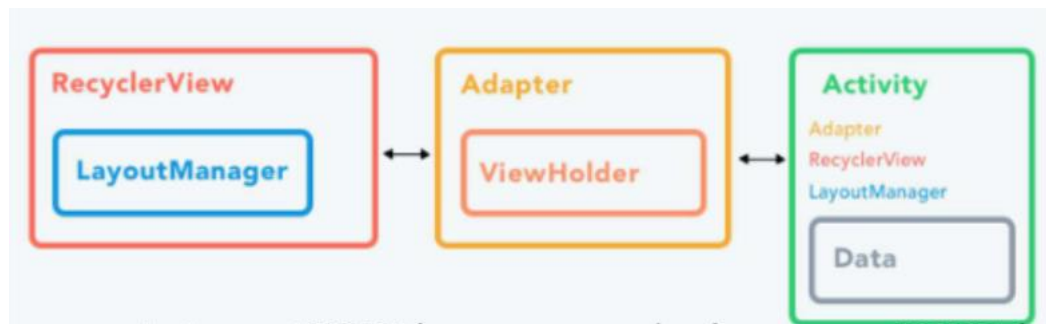
## [11.17 RecyclerView 목록을 보여주는 뷰 (뷰의 자식이다)]

리사이클러뷰==어댑터뷰.

### #LayoutManager

우리가 직접 만들진 않고, 라이브러리에서 이미 많이 제공하므로 하나 골라 쓰면 됨.

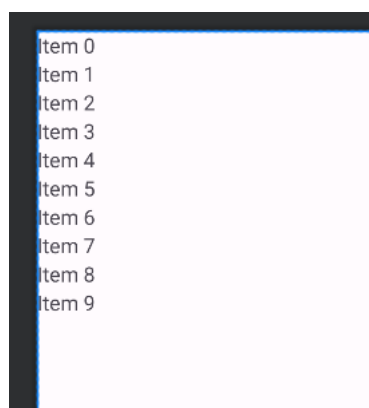
우리가 만들 거는 Adapter. 아래 이미지 강 외워버려



<실습08>

#xml 만들기 - 리사이클러뷰 띄울 액티비티에 리사이클러뷰 세팅

```
activity_main.xml x MainActivity.java x build.gradle.kts (:app) x item.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.recyclerview.widget.RecyclerView xmlns:android="http://schemas.android.com/apk/res/android"
3     android:id="@+id/recyclerView"
4     android:layout_width="match_parent"
5     android:layout_height="match_parent"/>
```



우리가 몇 개를 만들건지 모름.. 강 디폴트로 이렇게 9개 만들어 지는듯?

한 행 한 행이 모두 ViewGroup이다.

#간단하게 데이터를 만들어봄

```

33      List<String> list = new ArrayList<>();
34      for (int i = 0; i < 20; i++) {
35          list.add("Item=" + i);
36      }

```

#바인딩 3개 중에서 첫번째줄: 리니어레이아웃메니저(디폴트는 vertical)을 세팅해줌

```
binding.recyclerView.setLayoutManager(new LinearLayoutManager(context, this));
```

#나머지 세줄을 보면 recyclerView는 준비가 됐고, 그럼 adapter를 세팅해줘야 하는데, 그럼 viewHolder가 필요하네

```

38      binding.recyclerView.setLayoutManager(new LinearLayoutManager(context, this));
39      binding.recyclerView.setAdapter(new MyAdapter(list));
40      binding.recyclerView.addItemDecoration(new MyItemDecoration());
41  }

```

마지막 한줄은 필수가 아니니까 강의 안할것임. 뭐 목록 3개마다 간격을 띄운다 이런거..

#ViewHolder는 RecyclerView안의 ViewHolder를 상속해야 한다.

```
private class MyViewHolder extends RecyclerView.ViewHolder
```

#item.xml만드니까 textView하나밖에 안나타남.

한 행의 레이아웃을 내가 만들어야 한다! -->>> Item.xml 만들어서 레이아웃 만들어주세요

xml의 이름이 item이니까 애에 대한 binding은 ItemBinding일 것. → 다시 main 액티비티로 넘어와서..

#Adapter만들기

```

private class MyAdapter extends RecyclerView.Adapter<MyViewHolder>{

}

```

이거하면 3개 메소드 추가하라고 뜬다.

MainActivity의 나머지 자바 코드는 강 08파일에 필기한거 보삼~

<https://velog.io/@hyeryeong/RecyclerView-%EC%9E%98-%EC%82%AC%EC%9A%A9%ED%95%98%EA%B8%B01>

참고블로그

## [Fragment 기본] 11.20~

Fragment도 하나의 뷰 덩어리들을 관리하는 관리자이다. (Activity도 마찬가지다)

프레그먼트 교체는 프레그먼트 매니저 통해서 해야함

### #Fragment 생명주기

액티비티의 생명주기를 그대로 따르는 View이다.



add한 fragment위에 또 add가능 (한 영역에 여러 개의 프레그먼트가 붙을 수 있다)

뒤에 있는 애는 생명주기에 변화 없이 계속 resumed상태이다.

### #생명주기에서..

onCreateView() :인플레이션

onViewCreated() : create되고 난 다음 listener 달기..

### #대략적인 작성법

1.MainActivity에서는 fragment를 관리할 매니저를 얻는다. getSupportFragmentManager() 함수 액티비티 꺼인거 확인가능!

```
getSupportFragmentManager() //프레그먼트 매니저를 얻는다
    .beginTransaction() //매니저에게 부탁할 일을 시작
    .add(R.id.main_content, new OneFragment()) //add, replace, remove가능, 어디에 프레그먼트를 붙일 것인지
    .commit(); //트랜잭션 안의 일이 시작.
```

2. (Fragment에서는) onCreateView()에서 fragment를 인플레이션 한다

```

15 public class OneFragment extends Fragment {
16
17     @Nullable
18     @Override
19     public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
20         FragmentOneBinding binding = FragmentOneBinding.inflate(inflater, container, attachToParent: false);
21         return binding.getRoot(); //인플레이션해서 리턴. 리턴 타입이 view.
22     }

```

3. (Fragment에서는) onViewCreated()에서 listener를 단다.

```

24 @Override
25 public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
26     super.onViewCreated(view, savedInstanceState);
27
28     Button button = view.findViewById(R.id.button);
29
30     button.setOnClickListener(v -> {
31         //Fragment 내에서 Activity 객체 획득은 getActivity() 로...
32         getActivity() FragmentActivity
33             .getSupportFragmentManager() FragmentManager
34             .beginTransaction() FragmentTransaction
35             .replace(R.id.main_content, new TwoFragment())
36             .addToBackStack(name: null)
37             .commit();
38     });
39 }
40 }

```

여기서는 그 다음 TwoFragment를 관리하기 위한 프래그먼트 매니저 호출하고 있네.

#addToBackStack을 쓰면 또 add, remove, replace의 생명주기가 달라진다

이걸 안쓰면 TwoFragment 종료 시, 아예 앱이 종료되는데, 이걸 쓰면 이전 상태(OneFragment) 복원

```

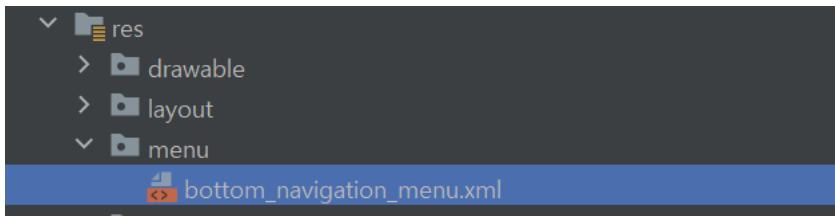
31 //Fragment 내에서 Activity 객체 획득은 getActivity() 로...
32 getActivity() FragmentActivity
33     .getSupportFragmentManager() FragmentManager
34     .beginTransaction() FragmentTransaction
35     .replace(R.id.main_content, new TwoFragment())
36     .addToBackStack(name: null)
37     .commit();

```

replace나, remove는 완전히 사라지는 걸 포함하는 함수인데 addToBackStack에 걸리면 완전히 죽이는게 아니라 onDestroyView()까지 갔다가 → 백버튼 누르면 onCreateView()로 간다.

[Fragment사용 예 - BottomNavigation] 개발자 문서에 검색하면 만드는 순서 나옴!

#BottomNaigation만들려면 따로 메뉴 작성해 줘야함. 메뉴 만든 후에 Bottom~에 app:menu로 건네주면 된다.



```
<item
    android:id="@+id/page_4"
    android:enabled="true"
    android:icon="@drawable/ic_news"
    android:title="News" />
```

enabled = false하면 사용자가 눌러도 반응없는 비활성화 상태 된다.

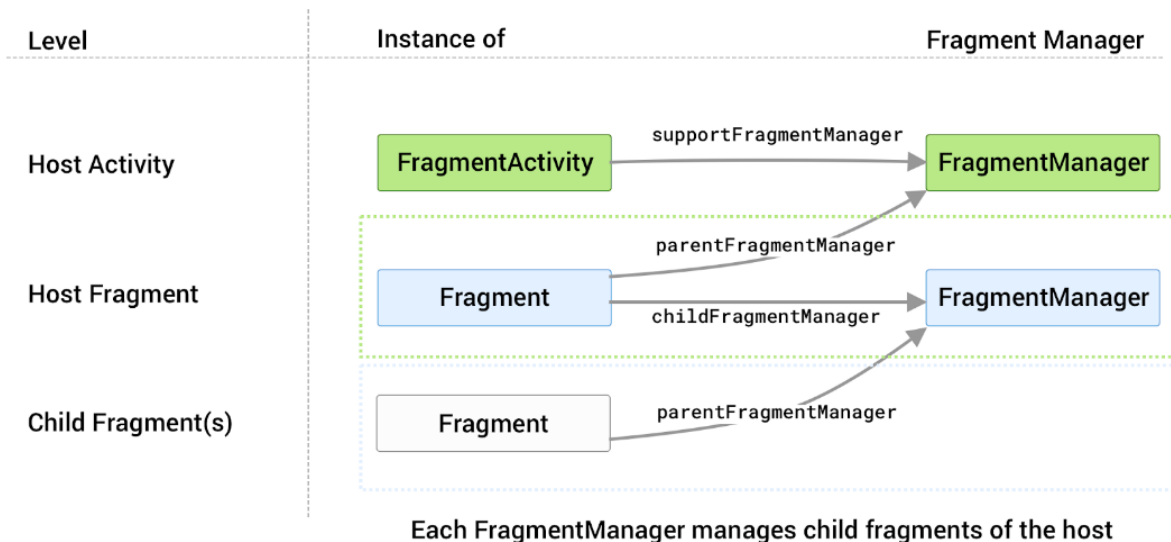
### #fragment의 골격 (빈 프래그먼트 만든거에서 본다)

- **onCreate()** : 액티비티와 같습니다. 리소스들을 초기화 해주는 단계로 프래그먼트를 생성하면서 넘겨준 값들이 있다면, 변수에 넣어줍니다. //주고받을 데이터 있다면 setting 해야함!!
- **onCreateView()** : 레이아웃을 inflate 하는 메소드입니다. view 객체를 얻을 수 있어 view 와 관련된 객체들을 초기화할 수 있습니다.
- **onActivityCreated()** : 프래그먼트의 onCreateView() 와 액티비티의 onCreate()가 호출되고 나서 호출됩니다. 액티비티와 프래그먼트의 뷰가 모두 생성된 상태로 뷰를 변경하는 작업이 가능합니다.

#

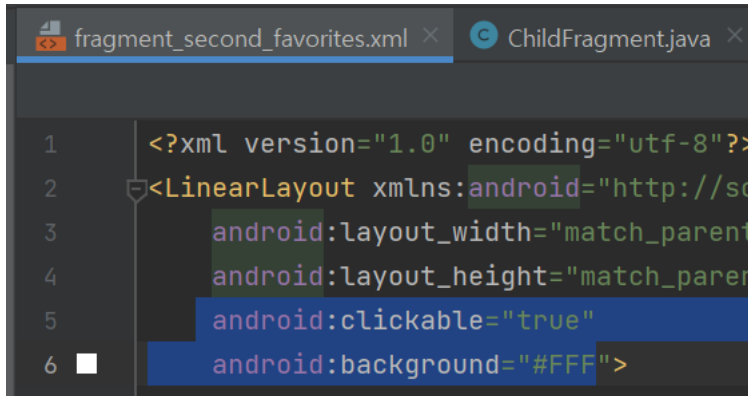
FragmentActivity는 액티비티를 뜻한다. supportFragmentManager로 매니저 얻고 매니저를 통해 프래그먼트 붙임.

Fragment입장에서 나를 붙여준/관리해주는 애를 찾는 법 : getParentFragmentManager



#첫번째줄 : 뒤로 클릭이 가지 않고, 내가 클릭을 받겠다.

#두번째줄 : add는 원래 디폴트로 투명한데, 배경색 설정해서 밑에놈 안보이게 할 수 있음



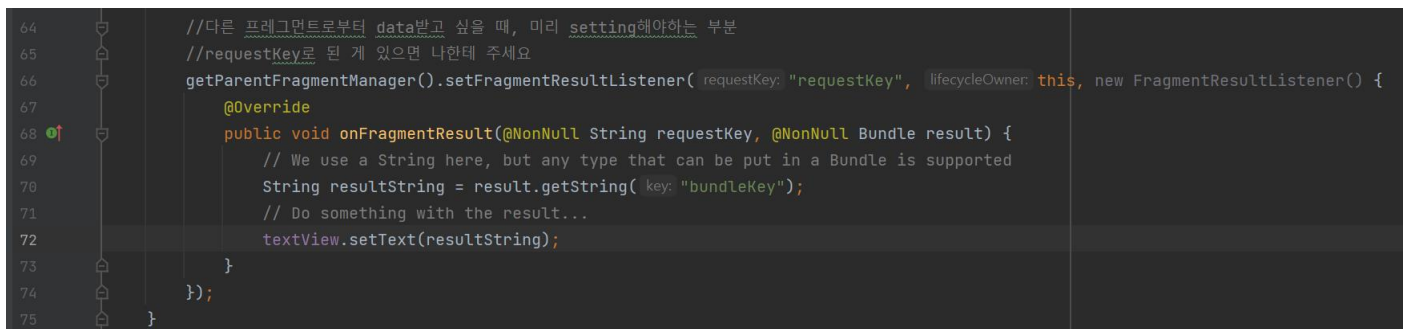
```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="match_parent"
4     android:layout_height="match_parent"
5     android:clickable="true"
6     android:background="#FFF">
```

## # 프래그먼트끼리 data 주고받기

일단 애네는 같은 fragmentmanager통해서 관리 받고 있음을 상기



..FavoriteFragment의 onCreate( ) 안



```
64 //다른 프래그먼트로부터 data받고 싶을 때, 미리 setting해야하는 부분
65 //requestKey로 된 게 있으면 나한테 주세요
66 getParentFragmentManager().setFragmentResultListener( requestKey: "requestKey", lifecycleOwner: this, new FragmentResultListener() {
67     @Override
68     public void onFragmentResult(@NonNull String requestKey, @NonNull Bundle result) {
69         // We use a String here, but any type that can be put in a Bundle is supported
70         String resultString = result.getString( key: "bundleKey");
71         // Do something with the result...
72         textView.setText(resultString);
73     }
74 });
75 }
```

..SecondFavoriteFragment



```

@Override
public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
    super.onViewCreated(view, savedInstanceState);

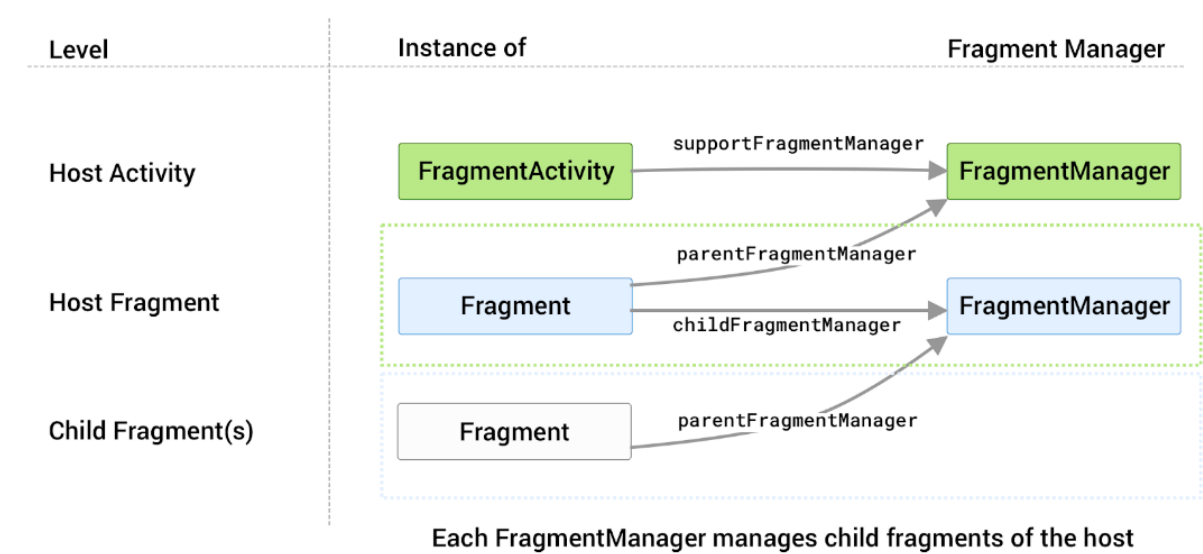
    Button button = view.findViewById(R.id.previous);
    button.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Bundle result = new Bundle();
            result.putString("bundleKey", "from the 2nd fragment");
            getParentFragmentManager().setFragmentResult(requestKey: "requestKey", result);

            getParentFragmentManager().beginTransaction().remove(fragment: SecondFavoritesFragment.this).commit();
        }
    });
}

```

setFragmentResult하면 1에게 다이렉트로 주는 것이 아니라, 매니저에게 맡겨줬다가 1이 보이게 되면 전달

### #PlaceFragment



Place랑 child는 파란색으로 표시된 매니저를 통해서 관리된다

#this : 이 프래그먼트 객체(나)가 보이게 되면 해당 key를 보내주세요. 관찰의 대상이 "받을 놈."

```

62 resultListener(requestKey: "requestKeyForPlaces", lifecycleOwner: this, new Fragment
63
64
65

```

[11.28] DB\_practice 파일에 내가 한 버전이 보기 쉬움. 이 파일 지우지 말구 공부해!

# RDBMS ← 데이터베이스 ← 테이블 ← 데이터

SQLite는 그 앱의 라이브러리 내에 있다. 해당 앱만 사용!

```
SELECT title, content
```

```
FROM tableName
```

```
WHERE
```

```
SELECT id, department (열 고르기)
```

```
FROM student
```

```
WHERE id >= 20230000 (행 고르기. if조건이라고 생각하면 됨)
```

```
ORDER BY id
```

[쉬는시간이후..]

Cursor 각각의 행을 가리키는 포인터. 2차원 배열 순회

데이터를 내부에서 계산만 어느정도 해두고 앞뒤로 데이터를

Select로 받은 데이터는 항상 이미지/2차원배열/행렬 이다.

파일 자체를 Map이라고 생각

## [12.05]

notification 띄우고 이런거나 alarm 파일은 시험범위 아님~  
sdk버전에 따라 2버전 짜야하는 그런거는 안냄

### [외부 저장소 - 앱 별 공간]

Android 개발자 > 문서 > 가이드

## 앱별 파일에 액세스

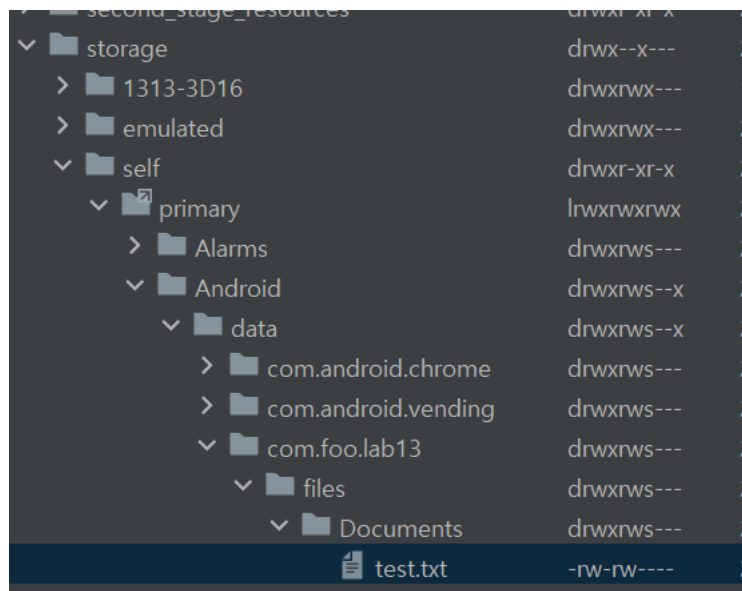
### #파일에 쓰는 코드

```
19 public class Lab13_2Activity extends AppCompatActivity {
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         ActivityLab132Binding binding = ActivityLab132Binding.inflate(getLayoutInflater());
25         setContentView(binding.getRoot());
26         binding.btn.setOnClickListener(new View.OnClickListener() {
27             @Override
28             public void onClick(View v) {
29                 File file = new File(getExternalFilesDir(Environment.DIRECTORY_DOCUMENTS), "child: text.txt");
30
31                 if (!file.exists()) { // 파일이 없다면 새로 만들어 준다.
32                     try {
33                         file.createNewFile();
34                     } catch (IOException e) {
35                         throw new RuntimeException(e);
36                     }
37                 }
38
39                 //이 밑 부분은 그냥 자바임. 파일에서 무언가를 읽어오는 내용. 관심 있으면 학습하셈
40                 try (FileWriter writer = new FileWriter(file, append: false)) {
41                     writer.write(binding.content.getText().toString());
42                     writer.flush();
43                 } catch (IOException e) {
44                     throw new RuntimeException(e);
45                 }
46
47                 Toast.makeText(context: Lab13_2Activity.this, text: "file save ok", Toast.LENGTH_SHORT).show();
48
49                 Intent intent = new Intent(packageContext: Lab13_2Activity.this, ReadFileActivity.class);
50                 startActivity(intent);
51             }
52         });
53     }
```

## #파일을 읽어서 텍스트뷰에 뿌려주는 코드

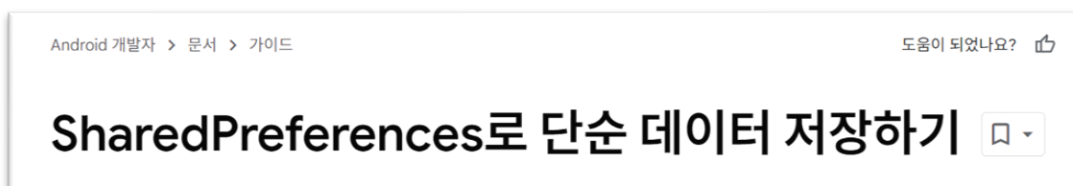
```
18 public class ReadFileActivity extends AppCompatActivity {
19
20     @Override
21     protected void onCreate(Bundle savedInstanceState) {
22         super.onCreate(savedInstanceState);
23         ActivityReadFileBinding binding = ActivityReadFileBinding.inflate(getLayoutInflater());
24         setContentView(binding.getRoot());
25
26         StringJoiner joiner = new StringJoiner( delimiter: "\n");
27
28         // 일반 문서니까 디렉토리 다큐먼트로 세팅
29         File file = new File(getExternalFilesDir(Environment.DIRECTORY_DOCUMENTS), child: "test.txt");
30
31         try (BufferedReader reader = new BufferedReader(new FileReader(file))) {
32             reader.lines().forEach(joiner::add);
33         } catch (IOException e) {
34             throw new RuntimeException(e);
35         }
36
37         binding.fileResult.setText(joiner.toString());
38     }
```

## #여기에 파일 생성된 것 확인 가능하다



getfilesdir() 내부저장소

## [SharedPreferences]



Map을 사용한다고 생각하면된다.

근데 메모리에 저장되는 것이 아니라 xml파일로 저장됨.

앱을 끄고 켜서 다시 접근이 가능하다.

## #SharedPreferences의 호출 방법에 따라 3가지 성격이 나뉜다

```
SharedPreferences localPrefs = getPreferences(MODE_PRIVATE);
SharedPreferences globalCustomPrefs = getSharedPreferences( name: "my_prefs", MODE_PRIVATE);
SharedPreferences globalDefaultPrefs = PreferenceManager.getDefaultSharedPreferences(getApplicationContext());
```

## #데이터 저장 방법

- ① 데이터 저장 위해서 Editor 얻어낸다.
- ② <키,값>으로 데이터 저장
- ③ commit( ) 호출

```
//에디터에 대고 data 넣는다고 생각
SharedPreferences.Editor editor = globalCustomPrefs.edit();
editor.putString("data1", "hello");
editor.putInt("data2", 100);
editor.commit();
```

## #데이터 얻는 방법

```
//데이터 얻을 때는 에디터까지 접근할 필요는 없다.
String data1 = globalCustomPrefs.getString( key: "data1", defValue: "none");
int data2 = globalCustomPrefs.getInt( key: "data2", defValue: 0);

binding.fileResult.setText(data1 + "\n" + data2);
}
```

## [DB vs. sharedPreferences]

db는 같은 성격의 것을 여러 개 저장

sharepreferences는 키,벨류로 해야하는데 몇 개가 될 지모름. 관리를 직접 해야하는데 그게 어려움.

## [BottomSheet]

### BottomSheet 띄울 수 있는 2가지 형태

- ① 화면 고틀머리에 보여서 끌어올릴 수 있는 건 standard
- ② 다이얼로그 형태 modal

#### [④화면 고틀머리에서 올리기]

원래 하던대로 화면디자인 하고 bottomsheets 달고 싶으면 그때 바깥을 CoordinatorLayout으로 감싸주면 됨.  
이 때 BottomSheet 디자인 별도로 빼서 include. 근데 그냥 여기다가 짜도 됨!

```
22
23     <include layout="@layout/bottom_sheet" />
24
```

### #BS xml

```
10     app:behavior_hideable="false"
11     app:behavior_peekHeight="32dp"
12     app:layout_behavior="com.google.android.material.bottomsheet.BottomSheetBehavior">
```

10번째 줄 : standard 경우에 hideable하면.....

11번째 줄 : 화면 고틀머리에 보일 높이 지정

12번째 줄 : 그러면 두개의 리니어레이아웃 중 무엇이 바텀시트로 작동함? 애가 BS로 동작할 애라는걸 알려줌.

#### [②다이얼로그 형태 modal] 다이얼로그, 프래그먼트 2가지 형태로 만들 수 있음

#데이터 주고받음

프레그먼트에서 일어나는 일을 액티비티로 전달