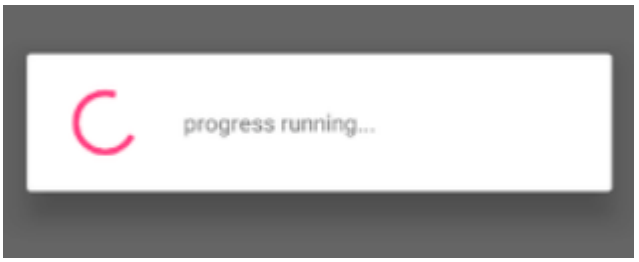
잠금앱

2월 19일

◇ProgressDialog



https://yongyi1587.tistory.com/41

◇ProgressBar

https://recipes4dev.tistory.com/135?category=708155

★CONTEXT

context는 context가 이용되는 java코드에서 해당 부분이 어떤 상황,화면에서 쓰이고 있는지 화면, 액티비티가 어떻게 굴러가고 있는지 전체적인 맥락을 부여한다.

```
dialog = new ProgressDialog(MainActivity.this);
```

ERROR:
'ProgressDialog(android.content.Context)' in 'android.app.ProgressDialog' cannot be applied to '(anonymous android.view.View.OnClickListener)'

ProgressDialog객체를 생성할 때는 Context객체가 파라미터로 전달되어야 하는데 액티비티인 MainActivity객체를 전달하기 위해 파라미터를 MainActivity.this로 지정해야 한다.

```
dialog = new ProgressDialog(MainActivity.this);
```

http://sunphiz.me/wp/archives/483

https://blog.naver.com/PostView.nhn?isHttpsRedirect=true&blogId=huewu&logNo=110085457720

◇android:ems

ems는 text에서 문자의 넓이를 일정하게 유지하게 도와준다.

android:ems나 setEms(n)은 실제 문자열 크기에 상관없이 문자에 폭에 맞게 너비를 설정합니다. ems는 오직 layout_width가 wrap_content일때만 적용됩니다.

https://recipes4dev.tistory.com/84

자, 그럼 EM 단위의 크기로 TextView의 너비(width)를 설정하는 "ems" 속성은 언제 사용하는 걸까요? 바로, TextView에 적용되는 폰트 크기에 따라 TextView의 너비가 항상 동일한 비율의 너비를 가지도록 만들 때 사용될 수 있습니다.

즉, TextView의 폰트 크기가 바뀌어도, 동일한 텍스트에 대해, TextView 내에서 항상 같은 모양으로 표시되도록 만들고자 할 때 유용하게 사용할 수 있습니다. 폰트 크기에 따라 크기 비율을 조절한 것 처럼 말이죠.

◇ .length()

https://hianna.tistory.com/519#:~:text=%EB%AC%B8%EC%9E%90%EC%97%B4%20%EA%B8%B8%EC%9D%B4%20%EA%B5%AC%ED%95%98%EA%B8%B0%20%2D%20length()&text=java.lang..character%20%EA%B0%AF%EC%88%98)%EB%A5%BC%20%EB%A6%AC%ED%84%B4%ED%95%A9%EB%8B%88%EB%8B%A4.

java length() 문자열의 길이를 반환한다.

◇EditText & addTextChangeListener

https://superwony.tistory.com/76

사용자가 Edittext에 입력한 값을 변화할 때 그 변화 과정을 추적한다.

◇ .getText()

https://elfinlas.tistory.com/221
edittext에 입력된 문자열을 가져올 수 있다.

```
text = editText.getText().toString();
textLength = text.length();
```

◇ .setText()

https://itpangpang.tistory.com/354

TextView 안에 문자열을 넣고 싶을 때 사용한다.

※ .setText() 안에는 int형이 들어갈 수 없다.
setText(CharSequence text) 형식으로 값이 들어가야 하고
CharSequence는 String과는 다르지만 같다고 알아도 무관하다.

int 값을 string으로 바꾸기 위해서는 다음 메소드가 필요하다.

```
Integer.toString(value)
```

◇ edittext 자동개행

```
android:inputType="textMultiLine"
android:scrollHorizontally="false"
```

2월 20일

◇ 키보드 숨기기

oncreate 안에 imm을 만들어주고

```
InputMethodManager imm = (InputMethodManager)
getSystemService(Context.INPUT_METHOD_SERVICE);
```

oncreate 밖에 hideKeyboard()를 만들어주고 이 메소드를 사용하면 된다.

```
private void hideKeyboard() {
imm.hideSoftInputFromWindow(editText.getWindowToke
n(),0);
```

◇Toast

https://developer.android.com/guide/topics/ui/notifiers/toasts

Toast.makeText(context, text, duration).show();

두번째 argument는 text type이다.

```
Context context = getApplicationContext();
CharSequence text = "Hello toast!";
int duration = Toast.LENGTH_SHORT;

Toast toast = Toast.makeText(context, text, duration);
toast.show();
```

◇ Button 이벤트 처리

https://recipes4dev.tistory.com/55

```
public void onButton1Clicked(View v){
    Toast toastResult =
Toast.makeText(MainActivity.this,text,Toast.LENGTH
_LONG);
    toastResult.show();
```

◇SeekBar.setOnSeekBarChangeListener

https://jaejong.tistory.com/2

볼륨조절을 같은 동작이 필요할 때 사용 되며 ProgressBar 확장 View이다.
.setOnSeekBarChangeListener을 이용해 값의 변경을 추적할 수 있다.

```
seekBar.setOnSeekBarChangeListener(new
SeekBar.OnSeekBarChangeListener() {
    @Override
    public void onProgressChanged(SeekBar seekBar,
int i, boolean b) {
    }

    @Override
    public void onStartTrackingTouch(SeekBar
seekBar) {
    }

    @Override
    public void onStopTrackingTouch(SeekBar
seekBar) {
    }
});
```

◇ Log.d

https://issell.tistory.com/20

```
Log.d(태그:"cycle", 메시지:"onResume");
```

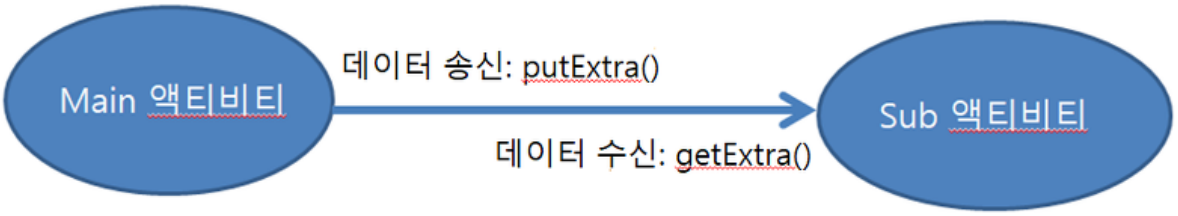
태그는 Logcat에서 검색할 때 찾는 키워드이고 메시지 부분이 logcat에 표시된다.

◇ INTENT

https://coding-factory.tistory.com/203

intent는 액티비티간 데이터를 전달하고 받을 때 쓴다.
데이터는 intent에 담겨 액티비티 사이를 이동한다.

※ 인텐트란
인텐트는 앱 컴포넌트가 무엇을 할 것인지를 담는 메시지 객체입니다. 메시지는 의사소통을 하기 위해 보내고 받는 것이지요. 메시지를 사용하는 가장 큰 목적은 다른 액티비티, 서비스, 브로드캐스트 리시버, 컨텐트 프로바이더 등을 실행하는 것입니다. 인텐트는 그들 사이에 데이터를 주고 받기 위한 용도로도 쓰입니다.



<보내기>
```
Intent finish_intent = new Intent(getApplicationContext(), MainActivity.class);
finish_intent.putExtra(getString(R.string.login_key), str);
setResult(9001, finish_intent);
finish();
```

<받기>
```
if(result.getResultCode() == 9001) {
1.  Intent intent = result.getData();
2.  Intent intent = getIntent();
   CharSequence result_data = intent.getStringExtra(getString(R.string.login_key));
   if(intent != null && result_data != null){}
```

◇ .setResult

새로 띄운 맥티비티에서 이전 액티비티로 인텐트를 전달하고 싶을 때 사용한다.
setResult(응답코드, 인텐트)

◇ .startActivityForResult()

https://jhshjs.tistory.com/49
activity를 여러 개 만들 때 MainActivity에서 subActivity를 띄울 때 쓴다. 뒤에 code로 activity 사이 정보를 교환한다.

https://kimyunseok.tistory.com/40
하지만 .startActivutyForResult는 deprecated되었다.

◇ .registerForActivityResult()

https://kimyunseok.tistory.com/39?category=1035300

2월 25일

◇ Parcelable

Intent는 액티비티 사이 정보를 주고 받는데 쓰이는 편지로
다른 액티비티를 시작하는데 도움을 준다.

이 때 Intent 안에 정수 문자열 등을 넣어서 보낼 수 있다.
이런 부가 정보들은 Intent 안의 Bundle 안에 저장되고 관리된다

정수와 문자열 외 객체를 넣어서 보낼 수 있는데

이때 그 객체는 인터페이스인 parcelable을 참조하는 객체여야 한다.

◇ 액티비티 수명주기

https://bbaktaeho-95.tistory.com/62

액티비티의 상태(켜짐, 꺼짐, 일시정지,,,)에 따라 상태 정보가 만들어지고 시스템은 이 상태정보를 관리한다.
그리고 각각의 상태에 해당하는 메서드를 자동으로 호출한다.

Aa 메서드 이름	액티비티 상태	설명
onCreate	만들어짐	액티비티 생성할 때
onStart	화면에 나타남	화면에 보여지기 시작할 때
onResume	현재 실행 중 화면에 나타남	화면에 나타나 있고 실행중일 때
onPause	화면이 가려짐	액티비티 화면의 일부가 다른 액티비티에 가려짐
onStop	화면이 없어짐	다른 액티비티의 실행으로 완전히 가려짐
onDestroy	종료됨	액티비티 종료됨

2월 26일

◇ SharedPreferences

https://developer.android.com/reference/android/content/SharedPreferences
https://re-build.tistory.com/37

앱의 데이터를 저장하여 관리할 때 쓰인다. 데이터 양이 적고 간단한 설정값일 때 자주 쓰인다.

Context.getSharedPreferences로부터 return 받은 data에 접근하거나 조작하는데 이용되는 인터페이스이다.

data를 접근할 때는 Editor을 사용해야 하고 마지막에 commit을 해야 저장이 된다.

키 값을 입력하고 .put으로 저장할 수 있고 .get으로 값을 뺄 수 있다.

저장
```
SharedPreferences pref = getSharedPreferences("pref",
Activity.MODE_PRIVATE);
SharedPreferences.Editor editor = pref.edit();
editor.putString("name", nameInput.getText().toString());
editor.commit();
```

복귀
```
SharedPreferences pref = getSharedPreferences("pref",
Activity.MODE_PRIVATE);
if((pref != null) && (pref.contains("name"))) {
   String name = pref.getString("name", "");
   nameInput.setText(name);
```

삭제
```
SharedPreferences pref = getSharedPreferences("pref",
Activity.MODE_PRIVATE);
SharedPreferences.Editor editor = pref.edit();
editor.clear();
editor.commit();
```

◇ 앱이 실행이 안되고 강제 종료 될때

Intent 확인

◇ Caused by: java.lang.IllegalStateException: You must either set a text or a view

https://stackoverflow.com/questions/65914250/java-lang-illegalstateexception-you-must-either-set-a-text-or-a-view

이 오류는 Toast의 두번째 argument로 입력된 값이 null이기 때문에 발생한다.

Toast의 show()를 할 때 전달되는 text가 null이면 toast는 보여줄 정보가 없어서 앱을 강제 종료 한다.
그래서 반드시 exception이나 if else문으로 예외 처리를 해줘야 한다.

2월 27일

◇ onNewIntent

https://dev.eyegood.co.kr/entry/Android%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9C-onNewIntent-%EC%95%8C%EC%95%84%EB%B3%B4%EC%9E%90

액티비티에서 다른 액티비티로 갔다가 다시 돌아올 때(x)

intent으로 한 액티비티가 한번 호출된 다음에
intent의 값이 추가된 상태호 다시 호출되면 onNewIntent으로 호출된다.

◇ LanchMode

https://g-y-e-o-m.tistory.com/94

사용 사례	시작 모드	다중 인스턴스?	설명
대부분의 액티비티에 대한 일반적인 시작	`"standard"`	예	기본값입니다. 시스템이 항상 대상 작업에 새 액티비티 인스턴스를 생성하고 인텐트를 해당 인스턴스로 라우팅합니다.
	`"singleTop"`	조건부	액티비티의 인스턴스가 이미 대상 작업의 맨 위에 존재하는 경우 시스템은 새 액티비티 인스턴스를 생성하는 대신 `onNewIntent()` 메서드를 호출하여 인텐트를 해당 인스턴스로 라우팅합니다.
특수한 시작 *(일반 용도로는 권장되지 않음)*	`"singleTask"`	아니요	시스템이 새 작업의 루트에 액티비티를 생성하고 인텐트를 해당 액티비티로 라우팅합니다. 그러나 액티비티 인스턴스가 이미 존재하는 경우 시스템은 새 인스턴스를 생성하는 대신 `onNewIntent()` 메서드를 호출하여 인텐트를 기존 인스턴스로 라우팅합니다.
	`"singleInstance"`	아니요	시스템이 인스턴스를 보유하는 작업에서 다른 액티비티를 시작하지 않는 점을 제외하면 `"singleTask"`와 동일합니다. 액티비티는 항상 해당 작업의 단일 멤버입니다.

singleTop일 때 새로운 액티비티 인스턴트를 생성하지 않고 onNewIntent로 인텐트를 받는다.

◇ Logcat not showing

https://protocoderspoint.com/android-studio-logcat-not-showing-anything-100-working-solution-found/#:~:text=you%20can%20try-,Solution%201%3A%20Restarting%20your%20Android%20Studio,will%20start%20work%20as%20before.

File > Invalidate Caches and Restart > Invalidate and Restart

3월 1일

◇ startActivityForResult

https://junyoung-developer.tistory.com/151

startActivity는 단방향으로 activity를 시작한다.
startActivityForResult는 새로운 activity 시작과 그 activity로부터 결과값을 받을 수 있다.

Fragment를 이용한 activity간 정보 교환을 권장하고
정 원하면 registerForActivityResult를 사용하면 된다.

◇ RES에 string 저장하기

res > values > string.xml

```
<string name="login_key">finishData</string>
```

사용할 때는

```
R.string.login_key
```

◇ registerForActivityResult

https://kimyunseok.tistory.com/39?category=1035300
https://www.youtube.com/watch?v=qO3FFuBrT2E

```
ActivityResultLauncher<Intent> activityResultLauncher;
```

<return값 받기>

```
activityResultLauncher = registerForActivityResult(new
ActivityResultContracts.StartActivityForResult(), new
ActivityResultCallback<ActivityResult>() {
    @Override
    public void onActivityResult(ActivityResult result) {
        if(result.getResultCode() == 9001) {
            Intent intent = result.getData();
            CharSequence result_data =
intent.getStringExtra(getString(R.string.login_key));
            if(intent != null && result_data != null){
            Toast.makeText(getApplicationContext(), result_data,
Toast.LENGTH_LONG).show();
            }
            else if(intent == null){
                Log.d("myError", "no intent value");
            }
            else{
```

```
                Log.d("myError", "no result_data value");
            }
        }
    }
});
```

<새로운 액티비티 열기>

```
Intent intent = new Intent(getApplicationContext(), MenuActivity.class);
activityResultLauncher.launch(intent);
```

<새로운 액티비티에서 intent값 보내기>

```
Intent finish_intent = new Intent(getApplicationContext(),
MainActivity.class);
finish_intent.putExtra(getString(R.string.login_key), str);
setResult(9001, finish_intent);
finish();
```

◇ 상수 설정하기

```
private static final String TAG = "MyError";
```

3월 2일

◇ check plainText empty

https://stackoverflow.com/questions/6290531/how-do-i-check-if-my-edittext-fields-are-empty

방법 1
textView.getText().toString().matches(“”)

방법2
```
private boolean isEmpty(EditTex editText) {
        if ( editText.getText().toString().trim().length() > 0)
                return false;
        else
                return true;
}
```

◇ setResult()

3개의 화면을 만들어 setResult로 3번 화면에서 1번화면으로 가려고 했지만
1번 화면으로 intent.putExtra값은 갔지만 화면에는 2번 화면이 보인다.

이는 setResult가 activity를 소환하는 기능을 가지고 있지 않기 때문이다.
setResult는 intent에 들어 있는 위치 정보로 어디로 정보를 보낼지 알고
Intent intent = new Intent(getApplicationContext(), MainActivity.class)
intent안의 넣은 정보를 그 위치로 보낸다.
Intent.putExtra(“key”, “abcde”);
하지만 activity를 start하지는 못한다. 그래서 setResult는 반드시 finish가 필요하다.

startActivity도 쓸 수 없다.
왜냐하면
startActivity를 쓰면 onResume으로 들어가기 때문에 intent값을 받지 못한다.

3월 3일

◇ Fragment

여러개의 fragment를 결합해서 하나의 액티비티에 보이게 할 수도 있으며 하나의 fragment를 여러 액티비티에서 재사용할 수 있다.

fragment란, 어플리케이션에서 화면에 직접 보이는 공간인 Activity내에서 분할시키고 다른 화면으로 전환할 수 있는 화면 공간의 단위입니다.

Fragment를 이용하려면, 그 상위에 있는 Activity에서 출력할 layout을 제어해줘야 합니다.

▹**fragment.java**와 **fargment.xml**연결

```
import androidx.fragment.app.Fragment
Fragment1 extends Fragment
onCreateView
return inflater.inflate(R.layout.fragment, container, false)
```

▹**fragment.xml**의 **widget**과 **fragment.java**과 연결

```
ViewGroup rootView = (ViewGroup) inflater.inflate(R.layout.fragment_main, container, false);
Button button = rootView.findViewById(R.id.button);
```

▹**fragment.java**에서 **fragment.xml**안의 버튼과 연결

```
ViewGroup rootView = (ViewGroup) inflater.inflate(R.layout.fragment_main, container, false);

Button button = rootView.findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
  @Override
  public void onClick(View view) {
     MainActivity activity = (MainActivity) getActivity();
     activity.onFragmentChanged(0);
  }
});
return rootView;
```

▹**fragment** 전환

fragment안에 있는 버튼으로 다른 fragment로 전환하고 싶을 때

fragment의 전환은 activity 그리고 그 안에 있는 fragmentmanager가 관장한다. 따라서 activity에 전환 메소드를 만들고 fragment의 버튼에서는 activity.메소드로 그 메소드를 불러와야 한다.

1. activity에 화면 전환 메서드 만들기
   public void onFragmentChanged(int index) {
    if (index == 0) {
   getSupportFragmentManager().beginTransaction().replace(R.id.container, menuFragment).commit(); }

   else if (index == 1) {
   getSupportFragmentManager().beginTransaction().replace(R.id.container, mainFragment).commit(); }

2. fragment의 버튼의 onclicklistener 내부에서 activity 객체를 만들고 activity안에 있는 화면 전화 메서드를 불러와서 전환한다.\
   MainActivity activity = (MainActivity) getActivity();
   activity.onFragmentChanged(0);

fhrm

◇ onCreateView

onCreate은 Fragment가 생성될때 호출되고
onCreateView는 onCreate 후에 화면을 구성 할 때 호출되는 부분이다.

fragment.java와 xml을 연결할 때 onCreateView에서 inflate해야 둘이 연결된다.

◇ Inflation

inflation은 xml코드가 메모리를 거쳐 화면에 보이게 되는 과정이다.

3월 4일

◇ fragment의 수명주기

https://ddangeun.tistory.com/50

fargment는 activity처럼 독립적인 수명 주기를 가져서 상태에 따라 자동으로 콜백 당한다. activity에 종속되어 있기 때문에 activity의 수명주기를 따르지만 fagment만이 가지고 있는 상태 정보도 있다.

이렇기 때문에 activity와는 독립적으로 분할화면의 동작을 설정할 수 있다.

◇ 상속과 생성자 그리고 Interface

1. 상속은 부모 객체가 가지고 있는 특성을 모두 가지는 새로운 객체를 만드는 것이다.
   - 부모 객체가 가지고 있던 변수, 메서드를 자식은 바로 사용할 수 있다.
     String name;
     void sleep( ) { System.out.println(this.name + " zzz"};
   - overriding을 이용해서 부모 객체 안의 메서드를 고쳐서 사용할 수 있다.
     void sleep( ) { System.out.println(this.name + " abc"};
   - overloading을 이용해서 부모 객체 안의 메서드를 추가해서 사용할 수 있다.
     overloading은 같은 이름의 메서드가 여러 형태의 자료형을 받고 그에 따라 다른 행동을 함을 의미한다.
     void sleep(int num) { System.out.println(this.name + num +" zzz"};
2. 생성자는 상속을 할 때 ~~반드시 추가해야 하는 생성에 필요한 argument~~를 의미한다.(x)
   생성자는 상속을 할 때 반드시 실시되는 명령을 의미한다. 이때 argument는 필요할 수도 있고 필요하지 않을 수도 있다. 생성자는 argument를 나타내는 말이 아니고 그 안에 있는 명령을 의미한다.
   ```
   class HouseDog extends Dog {
       HouseDog(String name) { this.setName(name);}}
   ```
   여기에서 this.setName(name)을 의미한다.

3. 인터페이스는 새로운 분류기준을 만들어서 객체에 추가함으로써 부모 클래스와는 다른 기준으로 객체를 묶고 새로운 변수, 베서드를 추가함을 의미한다.
   ```
   interface Predator { String getFood}

   class Tigar extends Animal implements Predator {
       public String getFood() { return "apple"}}
   ```

◇ 하나의 activity안의 서로 다른 두 fragment 사이 정보 교환

굳이 interface가 fragment안에 있지 않아도 된다. fragment에 소속된 기능이기 때문에 fragment안에 interface를 만드는 것이다.

▹정보를 보내는 fragment A에서 interface를 만든다

```
public static interface ImageSelectionCallBack {
        public void onImageSelection(int position); }
```

▹정보를 받는 fragment B에 정보를 받아 시작할 메서드를 만든다.

```
public void setImage(int resID) {imageView.setImageResource(resID);}
```

▹activity가 fragment A 안의 interface를 implement하게 하고 fragment B와 연결할 메서드를 interface 내의 메서드에 구현한다.

```
Mainactivity extends AppCompatActivity implements
ListFragment.ImageSelectionCallBack

@Override
public void onImageSelection(int position) {
        viewerFragment.setImage(images[position]);
```

▹activity와 fragment A가 연결되는 순간 onAttach에서 fragment B와 연결한 메서드를 interface 객체를 가져옴으로서 가져 온다.

```
@Override
public void onAttach(Context context) {
        super.onAttach(context);

        if(context instanceof ImageSelectionCallBack) {
                callBack = (ImageSelectionCallBack) context; }}
```

※try catch문을 이용하면 좋다. 이러면 implement되지 않았으면 context안에 있는 callback을 가져오지 못하고 넘어간다

```
@Override
  public void onAttach(@NonNull Context context) {
     super.onAttach(context);
    // ctrl+alt+t
     try {
        listener = (DialogListener) context;
     } catch (ClassCastException e) {
        throw new ClassCastException(context.toString() +
              "must implement DialogListener")
     }
  }
```

▹onAttach를 통해 가져온 객체로 fragment B와 연결된 메서드를 상황에 맞게 사용한다.

```
@Override
public void onClick(View v) {
        if(callBack != null) {
                callBack.onImageSelection(0); }}
```

◇ Option Menu

앱의 화면의 윗부분에 자리를 차지하는 화면으로 앱의 이름과 앱 내의 여러 설정이 있다.

▹ option menu xml 만들기

option menu를 사용하고 싶으면 res>menu 내에 xml을 만들어야 android가 이 directory를 참조한다.

▹ menu xml을 activity에 연결하기

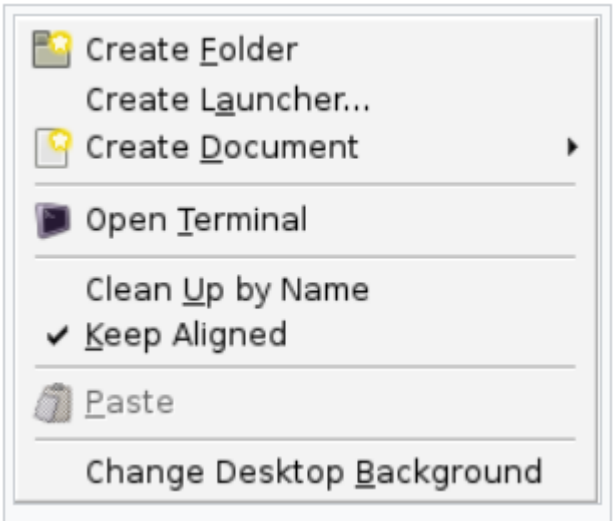
onCreateOptionsMenu(Menu menu) 메서드를 activity에 overriding해서 사용할 수 있다. 이 메서드는 activity가 만들어질 때 자동으로 호출되고 이때 menu.xml을 inflate하면 연결된다.

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true; }
```

☆메뉴를 선택했을 때 처리하는 방법
사용자가 하나의 menu item을 선택했을 때 자동 호출되는 onOptionsItemSelected을 override해서 item의 id값을 입력하고 원하는 작업을 추가하면 된다.

```
@Override
public boolean onOptionsItemSelected(@NonNull MenuItem item) {

   int curId = item.getItemId();
   switch (curId) {
      case R.id.menu_refresh:
         Toast.makeText(this, "새로고침 메뉴가 선택되었습니다.",
Toast.LENGTH_LONG).show();
         break;
,,,
      default:
         break;   }
 return super.onOptionsItemSelected(item); }
```

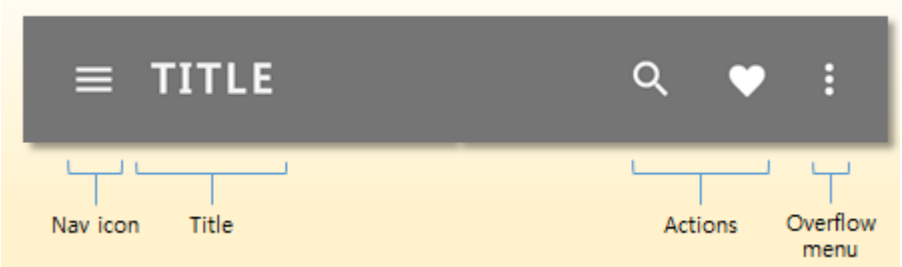
◇ ContextMenu



화면을 길게 누르면 나타나는 메뉴이다.
option menu처럼 뷰에 등록을 하기 위해 registerForContextMenu 메서드를 overriding하고 각각의 메뉴 아이템(복사, 붙여넣기 등등) 선택했을 때 onContextItemSelected 메서드가 호출된다.

◇ ActionBar

https://recipes4dev.tistory.com/141



▹ ActionBar 불러오기
onCreate 안에 있는 getSupprtActionBar()메서드로 불러올 수 있다.
ActionBar actionbar = getSupportActionBar();

▹ .setDisplayOptions()
액션바가 보이는 모양을 바꾸도록 할 수 있는 메서드이다.

▹ Action바 안에 원하는 layout을 넣기
menu.xml에 app:actionLayout으로 @layout/ 으로 넣을 수 있다.
app:actionLayout="@layout/search_layout"

그 다음에 onCreateOptionMenu안에서 menu.xml과 java 파일을 연결해줄 수 있다.

3월 5일

◇ Tabs and BottomNavigation

https://recipes4dev.tistory.com/115

https://brunch.co.kr/@gdw0809/2

상단탭: 화면의 위를 차지하는 상단탭은 스와이프를 지원하며 동일한 주제 내에서 작은 분류를 하고

하단탭: 화면 아래의 bottomnavigation은 다른 주제의 큰 분류를 3~4개 정도 놓는다.

◇ CoordinatorLayout

보통 액션바와 함께 쓰여서 액션바와 나머지 뷰 간 공간을 할당해준다.

스크롤 이벤트에 따라 상단 앱바의 변화를 줄 때 자주 사용한다.

◇ 앱은 기본으로 액션바가 표시된다

manifest.xml

<style name="Theme.SampleTap"
parent="Theme.MaterialComponents.DayNight.NoActionBar">

NoActionBar로 바꿔야 기본 액션바가 없어진다.

3월 7일

☆ 액션바 툴바 앱바

액션바: https://recipes4dev.tistory.com/141
툴바: https://recipes4dev.tistory.com/149?category=726986

앱바가 화면의 상단에 있는 화면의 구성요소이고
액션바와 툴바는 앱바를 만들기 위함 위젯이다.

☆탭 내비게이션

https://recipes4dev.tistory.com/115

탭은 여러 페이지로 구성된 UI레이아웃을 화면의 동일한 영역에 번갈아 표시할 때 사용한다.
탭은 반드시 tabwidget과 framelayout으로 구성되어야 한다.

방법 1

```
 // "Tab Spec" 태그(Tag)를 가진 TabSpec 객체 생성.
   TabHost.TabSpec ts = tabHost.newTabSpec("Tab Spec") ;

   // 탭이 눌려졌을 때 FrameLayout에 표시될 Content 뷰에 대한 리소스 id 지정.
   ts.setContent(R.id.content) ;

   // 탭에 표시될 문자열 지정.
   ts.setIndicator("TAB") ;

   // TabHost에 탭 추가.
   tabHost.addTab(ts)  ;
```

방법 2

```
TabLayout tabs = findViewById(R.id.tabs);
tabs.addTab(tabs.newTab().setText(“통화기록”);

tabs.addOnSelectedListener(new TabLayout.OnTabSelectedListener( {
} )
```

3월 8일 Viewpager

◇ ViewPager

https://recipes4dev.tistory.com/148

뷰페이저는 데이터를 페이지 단위로 만들고 이 페이지를 전환할 수 있도록 만들어 주는 컨테이너이다.

여러 데이터의 묶음을 표시하기 위해 안드로이드는 반드시 데이터를 adapter을 이용해 가공하는데 viewpager의 경우 pageradapter가 필요하다.

▹ pagerAdapter 쓰는 법

pagerAdapter는 abstract class이다.
하위 클래스를 만들어야 하고 추상 메서드 instantiateItem, destroyItem, getCount, isViewFromObject를 오버라이드 해야 한다.

▹fragment를 viewpager을 이용해서 전환하는 법
fragmentstatepageradapter를 상속받은 클래스를 만들어야 한다.

◇ Adapter

사용자가 정의한 데이터 리스트를 입력으로 받아 화면에 표시할 뷰를 생성한다. 하나의 데이터는 하나의 아이템뷰로 일치되면 이를 묶어서 표시하게 된다.

◇ abstract class

https://opentutorials.org/course/1223/6062

class 안에 추상 메서드가 하나라도 들어있으면 그 클래스는 추상클래스가 되고
객체를 생성할 수 없고 반드시 하위 클래스를 만들고 그 클래스에서 추상 메서드를 오버라이딩해야 사용할 수 있다.

3월 12일

◇ fragment 전환

1. FragmentCallback interface를 만들어 onFragmentSelected(int, Bundle) 메서드를 추가한다.
2.
```
@Override
public void onFragmentSelected(int position, Bundle bundle) {
  Fragment curFragment = null;

  if(position == 0) {
     curFragment = fragment1;
     toolbar.setTitle("첫번째 화면");
  } else if(position == 1) {
     curFragment = fragment2;
     toolbar.setTitle("두번째 화면");
  } else if(position == 2) {
     curFragment = fragment3;
     toolbar.setTitle("세번째 화면");
  }

  getSupportFragmentManager().beginTransaction().replace(R.id.container,
curFragment).commit();
}
```
3. 원하는 위치에 onFragmentSelected메서드를 넣어서 사용한다.

◇ EditText에서 inputtype

https://developer.android.com/reference/android/widget/EditText

반드시 어떤 형식의 정보가 입력되는지 inputtype를 결정해줘야 한다.
입력 형식에 맞게 keyboard의 종류가 결정된다.

android:inputType
여기에 사용할 수 있는 inputtype의 종류가 나온다.

3월 13일

◇ 오늘 날짜 불러오기

▹ date을 string으로 바꾸기
https://stackoverflow.com/questions/5683728/convert-java-util-date-to-string

```
String pattern = "MM/dd/yyyy HH:mm:ss";
DateFormat df = new SimpleDateFormat(pattern);
Date today = Calendar.getInstance().getTime();
String todayAsString = df.format(today);
```

▹ dateFormat & simpledateformat
https://developer.android.com/reference/java/text/DateFormat?authuser=1
DateFormat is an abstract class for date/time formatting subclasses which formats and parses dates or time in a language-independent manner.
data/time format subclass를 위한 추상 클래스이다. date <-> text는 subclass인 simpledateformat을 이용하면 된다.

The date/time formatting subclass, such as SimpleDateFormat, allows for formatting (i.e., date → text), parsing (text → date), and normalization.

SimpleDateFormat allows you to start by choosing any user-defined patterns for date-time formatting. However, you are encouraged to create a date-time formatter with either getTimeInstance, getDateInstance, or getDateTimeInstance in DateFormat.

▹ calender
https://developer.android.com/reference/java/util/Calendar?authuser=1#fields
실제 날짜와 시간을 우리가 가공할 수 있는 data형태인 calendarfield로 변환해준다.

◇ mainActivity.xml에 fragment.xml 넣을 공간 마련하기

You can add your fragment to the activity's view hierarchy either by defining the fragment in your activity's layout file or by defining a fragment container in your activity's layout file and then programmatically adding the fragment from within your activity.

In either case, you need to add a FragmentContainerView that defines the location where the fragment should be placed within the activity's view hierarchy. It is strongly recommended to always use a FragmentContainerView as the container for fragments, as FragmentContainerView includes fixes specific to fragments that other view groups such as FrameLayout do not provide.

▹ fragmentcontainerview 초기화

1. fragmentcontainerview에는 android:name="com.example.ExampleFragmen"를 넣어서 초기화 할 수 있다. The android:name attribute specifies the class name of the Fragment to instantiate. When the activity's layout is inflated, the specified fragment is instantiated,
2. FragmentTransaction을 이용 할 수도 있다.

◇ activity에서 fragment 안에 있는 view, widget 제어하기

https://kitesoft.tistory.com/82
중요한 것은 fragment 내부의 작용은 fragment내에서 처리하고 activity와 fragment는 서로 data를 주고 받으면서 activity에서 처리할 부분은 activity에서 처리해야 한다.

따라서 fragment안의 버튼은 fragment안에 있는 setonclickedlistener로 조절하고 메서드를 하나 더 만들어서 눌렸다는 정보를 activity에 전달해줘야 한다.

▹main activity에서 fragment 객체를 만들고 fragmentmanger의 findFragmentByid로 그 fragment를 가져온 다음 직접 fragment의 view를 바꾼다

1. findFragmentById()를 이용해서 fragment찾기
   사용할 fragment을 찾으려면 findFragmentById()를 사용해야 하는데 이 메서드는 Activity클래스에는 존재하지 않고 FragmentManager클래스에 존재한다.
2. View fragView = fragment.getView().findViewById()로 view가져오기
3. Button button1 = (Button) fragView.findViewById(R.id.button_birthdate);

```
mainFragment = (MainFragment)fragmentManager.findFragmentById(R.id.container_frame);
TextView textView = (TextView)
mainFragment.getView().findViewById(R.id.textView_positive);
textView.setText(str);
```

☆fragment에 view를 바꾸는 메서드를 만들고 activity에서 fragment.메서드를 이용해 연결한다
이때 fragmentmanager를 이용해서 꼭 fragment 객체에 findfragmentbyid로 fragment값을 가져와야 한다.

```
//main activity
@Override
public void textChange(String str) {
  mainFragment =
(MainFragment)fragmentManager.findFragmentById(R.id.container_frame);
  mainFragment.setTextViewValue(str);
}

//main fragment
public void setTextViewValue(String str){
  positive_text.setText(str);
}
```

◇ Fragment에서 Toast 사용하기

https://horae.tistory.com/entry/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9C-Fragment-%EC%97%90%EC%84%9C-Toast-%EC%82%AC%EC%9A%A9%ED%95%98%EA%B8%B0

toast를 사용하려면 toast가 표시되는 context가 필요하다.

1. Context 객체 만들기
2. fragment를 감싸고 있는 context 가져오기
3. makeText(context, “string”, length)

3월 14일

◇DatePickerDialog

https://developer.android.com/guide/topics/ui/controls/pickers

▹ Createing a Picker

1. DialogFragment를 상속해야 하고,
2. DatePickerDialog.OndateSetListener를 implement해야 한다.(맞아? 맞네 꼭 해야함)
3. 추상 메서드 onDateSet을 추가해서 고른 날짜로 무엇을 할 지 적어준다
4. onCreateDialog를 통해 .xml파일 없이 제공되는 달력 dialog를 쓸 수 있다. 이때 DatePickerDialog를 return해줘야 한다

▹ Showing a Picker

dialogFragment를 켜주고 싶은 버튼 onclicklistener 내에 FragmentDateDialog 객체를 만들어주고 .show()메서드를 사용한다.

◇ FragmentManager

https://developer.android.com/guide/fragments/fragmentmanager

▹ FragmentManager is the class responsible for performing actions on your app's fragments, such as adding, removing, or replacing them, and adding them to the back stack.

▹ Every FragmentActivity and subclasses thereof, such as AppCompatActivity, have access to the FragmentManager through the getSupportFragmentManager() method.

▹Fragment는 다른 Fragment를 안에 품을 수 있으며 getChildFragmentManager(), getParentFragmentManager() 메서드를 통해 각각의 fragmentmanager을 참조할 수 있다.

◇ Caused by: java.lang.NullPointerException: Attempt to invoke virtual method 'android.view.View android.view.View.findViewById(int)' on a null object reference

https://fluorite94.tistory.com/15

에러가 발생하는 이유는 fragment의 view가 inflate하기 전에 cmponent를 호출했기 때문이다.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  setContentView(R.layout.activity_main);
{
```

```
View view_frag = fragmentInfo.getView();
button_date_activity = (Button)view_frag.findViewById(R.id.button_birthdate); }
```

onCreate state에서는 아직 fragment가 연결되지 않는다.

3월 15일

◇ Fragment Transactions

https://developer.android.com/guide/fragments/transactions

fragmentmanager는 fragment의 추가, 제서, 교체를 관장하는데 fragmenttransaction class를 통해 fragment 변경이 가능하다. 이때 manager가 관리하는 back stack에 transaction 과정을 저장해서 뒤로 돌아갈 수 있다.

FragmentManager fragmentManager = getFragmentManager();
FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction();

▹ .beginTransaction()
▹.setReorderingAllowed()
▹.add()
▹.remove()
▹.replace()
▹.commit()

▹ .show() .hide()
▹.detach() .attach()

◇ Fragment LifeCycle

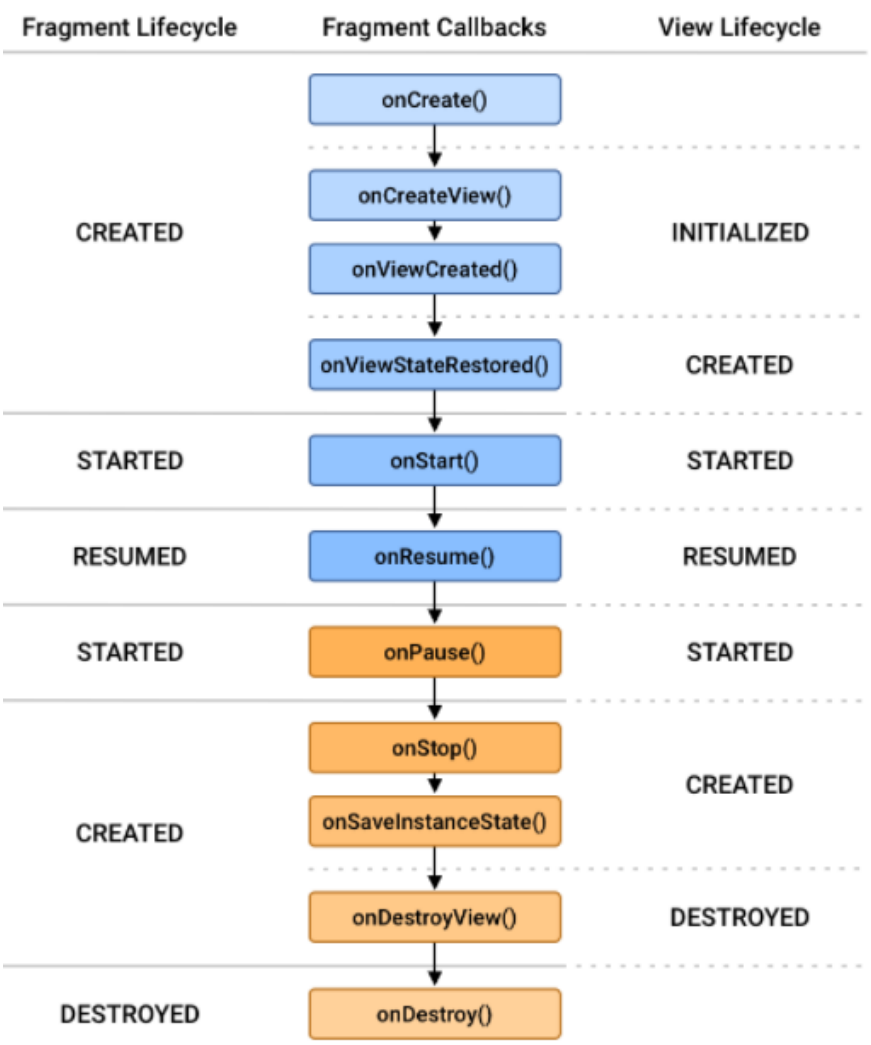
▹By building Fragment on top of Lifecycle, you can use the techniques and classes available for Handling Lifecycles with Lifecycle-Aware Components.

This component could automatically start listening when the fragment becomes active and stop when the fragment moves to an inactive state.

▹ fragment의 lifecycle 확인 방법

1. LifeCycleOwner을 implement해서 getlifecycle() method를 사용한다. getlifecycle() method는 Lifecycle객체를 return하는데 이를 통해 현재 어떤 state에 있는지 확인 가능하다
2. Fragment class에는 callback method가 있는데 callback method는 fragment의 lifecycle와 연동되어 해당 state에 도달하면 자동으로 시작된다.

▹ fragment View의 lifecycle
fragment의 view는 fragment와는 독립적인 lifecycle을 가진다.



▹ INITIALIZED
When a fragment is instantiated, it begins in the INITIALIZED state. For a fragment to transition through the rest of its lifecycle, it must be added to a FragmentManager. The FragmentManager is responsible for determining what state its fragment should be in and then moving them into that state.

- onAttach()
  The onAttach() callback is invoked when the fragment has been added to a FragmentManager and is attached to its host activity. At this point, the fragment is active, and the FragmentManager is managing its lifecycle state. At this point, FragmentManager methods such as findFragmentById() return this fragment.

▹CREATED
When your fragment reaches the CREATED state, it has been added to a FragmentManager and the onAttach() method has already been called.

- onCreate()
  created transition invokes the onCreate() callback. The callback also receives a savedInstanceState Bundle argument containing any state previously saved by onSaveInstanceState()

- onCreateView()
  The fragment's view Lifecycle is created only when your Fragment provides a valid View instance. In most cases, you can use the fragment constructors that take a @LayoutId, which automatically inflates the view at the appropriate time. You can also override onCreateView() to programmatically inflate or create your fragment's view.

- onViewCreate()
  If and only if your fragment's view is instantiated with a non-null View, that View is set on the fragment and can be retrieved using getView(). The getViewLifecycleOwnerLiveData() is then updated with the newly INITIALIZED LifecycleOwner corresponding with the fragment's view. The onViewCreated() lifecycle callback is also called at this time.

▹STARTED
▹RESUMED
the fragment is ready for user interaction and onResumed() callback is invoked. Fragments that are not RESUMED should not manually set focus on their views or attempt to handle input method visibility.

▹DOWNWARD STARTED
- onPause()

◇ Displaying Dialogs With DialogFragment

https://developer.android.com/guide/fragments/dialogs

▹ View만들기
DialogFragment를 상속하면 onCreateDialog를 override해서 이미 제공해주는 형식에 따라 view를 생성 할 수 있다. custom view를 inflate하고 싶으면 fragment와 동일하게 onCreateView()를 override해서 inflate하면 된다.

※ Showing the DialogFragment
It is not necessary to manually create a FragmentTransaction to display your DialogFragment. Instead, use the show() method to display your dialog. You can pass a reference to a FragmentManager and a String to use as a FragmentTransaction tag. When creating a DialogFragment from within a Fragment, you must use the Fragment's child FragmentManager to ensure that the state is properly restored after configuration changes. A non-null tag allows you to use findFragmentByTag() to retrieve the DialogFragment at a later time.

```
new PurchaseConfirmationDialogFragment().show(
    getChildFragmentManager(), PurchaseConfirmationDialog.TAG);
```

For more control over the FragmentTransaction, you can use the show() overload that accepts an existing FragmentTransaction.

Note: Because the DialogFragment is automatically restored after configuration changes, consider only calling show() based on user actions or when findFragmentByTag() returns null, indicating that the dialog is not already present.

3월 16일

fragment.show()
fragmentManager child parent

◇ CallBack Function

https://stackoverflow.com/questions/9596276/how-to-explain-callbacks-in-plain-english-how-are-they-different-from-calling-o/9652434#9652434

callback fuction은 다른 function에 trigger되어서 시작되어지는 fuction이다. 보통은 사용자에 의해서 event가 발생했을 때 callback fuction이 event에 의해서 trigger된다.

callback fuction은 trigger의 주체가 되는 다른 fuction을 argument로 갖는다.

◇ Fragment에서 Toast쓰기

1. onCreateView에서 inflater로 rootView를 가져오고 getContext()로 context를 가져온다.
   context = container.getContext();

2. rootView.findViewById로 widget을 가져온다.

◇ 상수 설정

각 activity fragment마다 Log.d를 위한 Tag 작성
public static final String TAG = "FragmentInfo";
필요한 상수 지성
public static final String PATTERN = "MM/dd/yyyy";

◇ Principles of Navigation

▹Navigation state is represented as a stack of destinations
앱이 시작했을 때부터 여러 activity와 fragment들을 backstack에 쌓이고 빠지는 과정을 거친다. navigation component의 필요성은 이런 모든 activity와 fragment가 쌓여 있는 backstack을 미리 잘 정리해서 사용자에게 보여줌으로써 편리성을 증가시키는 것이다.

그냥 버튼을 사용해서 fragment들을 교체해주는 것과 navigation component들을 사용하는 것의 차이는 navigation은 복잡한 화면 교체를 하나로 묶어 stack에 넣어서 관리함으로써 back버튼이나 다른 navigation item을 눌렀을 때 많은 변화가 한번에 일어나는 것이다.

▹task
https://developer.android.com/guide/components/activities/tasks-and-back-stack

task는 사용자가 앱을 사용하기 위해 사용되는 activity들의 묶음이다. 이 activity들은 backstack에 쌓여서 관리된다.

▹ Deep Link
deep link는 앱에 있는 특정 부분으로 바로 가는 link를 제공한다. 이때 그 fragment이나 activity로만 가는 것이 아니고 그 목적지에 도달하기 위해 거쳐야 하는 activity와 fragment들을 backstack에 순서에 맞게 저장해서 그 backstack 자체를 가져오는 것이다.

3월 19일

◇ Material Design for Android

https://developer.android.com/guide/topics/ui/look-and-feel

Material design is a comprehensive guide for visual, motion, and interaction design across platforms and devices.

meterial design은 앱이나 platform 안에 있는 구성 요소들의 움직임과 형태 색 위치정보에 대한 디자인이다.

◇ Add the AppBar

https://developer.android.com/training/appbar

앱바의 목적은 3가지이다.

1. 사용자가 앱을 사용하면서 지금 어떤 상황인지 알려줘야 한다. 어느 activity이며 어떤 fragment를 보고 있는지 지금 어떤 action을 취해서 이 화면으로 들어왔는지 알려줘야 한다.
2. 중요한 action들을 빠르게 접근할 수 있게 해야 한다. search기능 등이다.
3. navigation이나 menu를 접근할 수 있게 한다.

◇ Set up the AppBar

https://developer.android.com/training/appbar/setting-up

앱바를 만드는 방법은 actionbar와 toolbar가 있는데 activity를 만들면 기본으로 actionbar가 붙는다.

하지만 appbar를 만들 때 꼭 toolbar을 이용해라. 여러 device에서 문제 없이 구현되고 더 많은 기능을 포함함다.

▹ Toolbar로 appbar을 구현하기 위한 준비과정

1. v7 appcompat support library를 project에 추가한다.
2. activity가 appcompatactivity를 상속해야 한다.
3. app manifest에서 <application> 부분을 NoActionbar theme으로 바꿔야 기본 actionbar를 사용한 appbar가 없어진다.

▹ Toolbar 위젯을 activity에 추가하고 android:elevation 값을 주어 화면 위에 떠 보이게 한다.
▹ Toolbar를 앱의 appbar로 사용되게 만들기: setSupportActionBar()메소드의 인자로 toolbar 객체를 넣는다.

☆ v7 appcompat에서 제공하는 여러 appbar utility method를 사용하려면 getSupportActionBar()로 객체를 가져와서 사용할 수 있다.

◇ Add and handle action

https://developer.android.com/training/appbar/actions

The app bar allows you to add buttons for user actions.

▹ add action button: appbar에 넣고 싶은 모든 버튼들은 res/menu/안에 들어있어야 한다.
▹ app: showAsAction으로 버튼을 어떠한 상황에서 보이게 할지 정할 수 있다.
▹ appbar전용 icon은 여기서 보자 https://fonts.google.com/icons?selected=Material+Icons

◇ Appbar의 디자인

https://material.io/components/app-bars-top#usage

3월 20일

☆ Support Library

https://developer.android.com/topic/libraries/support-library

The Support Libraries provide a wide range of classes for building apps, from fundamental app components, to user interface widgets, to media handling, to TV app components

For information about how to add support library code to your app development project, see Support Library Setup. For information on how to include specific support library packages in your project, see Support Library Packages.

▹ backward compatibility
os나 제품, 기술이 이전 버전과 호환이 되어 작동하는 속성을 의미한다.

Backward compatibility (sometimes known as backwards compatibility) is a property of an os, product, or tech that allows for interoperability with an older legacy system, or with input designed for such a system, especially in telecommunications and computing.

1. App Components: These Support Library classes provide backward-compatible implementations of important, core platform features.
2. User Interface: These support library classes provide implementations of key user interface widgets and behaviors, and help you create more modern app interfaces on earlier devices.
3. Maerial Design: The support libraries provide a number of classes for implementing Material Design user interface recommendations.
4. Graphics: The android.support.graphics.drawable package provides support for vector drawables. By using vector drawables, you can replace multiple PNG assets with a single vector graphic, defined in XML.
5. Accessibility: The android.support.v4.view.accessibility package provides compatibility classes for implementing accessibility features introduced in API level 14 and later, which allow accessibility services to observe and identify user interaction with items displayed on screen.
6. Media Playback
7. TV Apps
8. Wear Apps
9. Utilities

◇ DrawerLayout

https://developer.android.com/reference/androidx/drawerlayout/widget/DrawerLayout

DrawerLayout은 top-level-container로 화면 전체를 감싸는 layout으로 설정되어야 하고

drawer view들과 상호작용해서 화면의 오른쪽 또는 왼쪽 모서리에서 drawer view를 꺼낼 수 있게 도와준다.

▹ xml

1. 화면 전체를 감싸는 drawerlayout 첫 화면에 들어가야 할 다른 요소들을 matchparent로 drawerlayout을 꽉 차게 한다
2. drawer을 추가한다. layoutgravity로 방향을 정해준다.
3. 보통 원하는 width를 설정해주고 height는 matchparent로 쓴다.

▹Design tip

1. Left/Start: 앱 전체에 대한 navigation을 보여주게 한다.
2. Right/End: 현재 화면에 대한 여러 action을 보여준다.

▹ java
drawerlistener
simpledrawerlistener

☆navigation drawer을 사용하던 그냥 drawer을 사용하던 모두 전체 layout을 drawerlayout으로 감싸줘야 한다.
navigation drawer을 사용하려면 전용 navigationUI을 사용하자
그냥 drawer을 쓸 때도 navigationUI를 사용하고 menu를 추가하지 않으면 된다.
https://www.youtube.com/watch?v=bjYstsO1PgI

◇ Designing Responsive Content

https://developer.android.com/guide/navigation/navigation-form-factors

▹responsive web design
Responsive web design (RWD) or responsive design is an approach to web design that aims to make web pages render well on a variety of devices and window or screen sizes from minimum to maximum display size to ensure usability and satisfaction.

▹You can find more examples of responsive design patterns and ideas for adaptive layouts on material.io.

3월 21일

◇ Navigation Overview

https://developer.android.com/guide/navigation

▹ Navigation 이란
navigation이란 앱 내의 다양한 content 사이를 돌아다니고, 들어갔다가 나갔다 하는 모든 상호과정을 의미한다.

▹ Android JetPack 안의 Navigation component는 버튼, 스와이프, appbar, navigation drawer 이용해 navigation을 구현하게 도와준다.

▹Navigation component의 구성
1. Navigation graph: navigation과 관련있는 모든 정보를 보여주는 xml 파일이다.
   ☆destination: 앱 안의 독립적으로 존재하는 공간. 하나의 화면이라 생각하자.
2. NavHost: navigation graph가 보여주고 있는 fragment 간의 동작을 모두 담고 있는 empty container이다. NavHostFragment를 사용한다.
3. NavController: navhost안의 app navigation을 관리하는 객체이다.

◇ Navigation Getting started

https://developer.android.com/guide/navigation/navigation-getting-started#groovy

▹ navigation graph 생성하기
navigation graph는 res에서 new resource file을 통해 만들 수 있다. 이때 resource type이 navigation이어야 한다.

navigation graph의 구성요소는 destination과 action이다.
1. Destinations are the different content areas in your app.
2. Actions are logical connections between your destinations that represent paths that users can take.

▹adding a NavHost to an activity
navigation graph속의 fragment들이 들어갔으면 하는 부분에
androidx.fragment.app.Fragment ContainerView를 사용해서 container을 만들어야 한다.

1. android:name: NavHost 정보를 가지고 있는 class인 androidx.navigation.fragment.NavHostFragment를 적어야 한다.
2. app:navGraph:@navigation/파일이름을 넣어서 navgraph와 연결해야 한다.
3. app:defaultNavHost: NavHost의 default 성질을 넣는 것이다.

☆ navigation component은 하나의 activity에서 여러개의 fragment를 관리하기 위해 디자인되었다. 따라서 여러 개의 activity로 구동되는 앱은 각 activity마다 그 안을 관리하는 navigation graph가 필요하다.

▷ Create Destination
https://developer.android.com/guide/navigation/navigation-create-destinations#placeholders

이미 있는 fragment나 activity를 destination으로 설정할 수 있다. Navigation Editor을 이용해서 new destination을 만들 수 있다. placeholder을 미리 만들고 나중에 원하는 fragment나 activity로 교체할 수 있다.

◇ Upate UI components with NavigationUI

https://developer.android.com/guide/navigation/navigation-ui#java
navgraph를 사용하면서 appbar, navigationdrawer 사용하고 싶으면 여기 보면된다
navigationUI는 navigation component 안에 appbar, navigationdrawer, bottom navigation을 추가하는 것을 도와준다.

▷ appbar
navgraph 안에 있는 android:label은 appbar에 내 현재 위치를 최신화하는데 정보를 제공한다.

▷AppBar Configuration
가장 처음 화면인 root destination을 인식해서 그 화면에서만 navigatoin icon을 좌상단에 노출하고 다른 화면으로 가면 그 위치에 뒤로가기 버튼을 표시하게 하는 객체이다.

사용하기 위해서는 appbarconfiguration 객체를 만들고 이를 navigationgraph에 넘겨주면 된다.

▷ navigaton drawer
https://www.youtube.com/watch?v=bjYstsO1PgI

3월 23일

◇ Navigation to a destination

https://developer.android.com/guide/navigation/navigation-navigate

destination을 만들고 destination간 action을 설정하고 난 뒤 각 NavHost 당 하나의 NavController을 만들어야 한다.
NavController은 destination 사이의 navigation을 도와주는 객체이다.

fragment, activity, view에서 NavController을 가져오기 위해서는 각각 method를 이용해야 한다.

1. fragment: NavHostFragment.findNavController(Fragment)
2. activity: Navigation.findNavController(Activity, @IdRes int viewId)
3. view: Navigation.findNavController(View)

NavController을 가져 온 다음에 navigate()에 있는 overload를 call해서 destination 사이를 돌아다닐 수 있다.

각각의 overload는 다양한 navigation 경우를 가능하게 도와준다.

▹ Using Safe Args to navigate with type-safety

destination 사이 navigation과 destination간 data 전달을 도와주는 방법으로 가장 안전한 방법이다.

1. gradle에 safe args를 추가해서 safe args를 enable시킨다.
    a. biuld.gradle.앱이름

```
buildscript {
    repositories {
        google()
    }
    dependencies {
        def nav_version = "2.4.1"
        classpath
"androidx.navigation:navigation-safe-args-gradle-plugin:$nav_version"   }
 }
```

    b. build.gradle:app의 plugins{ } 안에 추가해야 한다.

```
        plugins {
 id 'androidx.navigation.safeargs'}
```

2. safe args를 이용해서 출발 destination을 가르키는 class를 만들고 그 destination에서 출발하는 여러 action을 가르키는 class를 만든다. 그리고 각 class당 method도 생성한다. 이 method들은 각 action과 연결되어 action parameter를 인자로 받고 NavDirection 객체를 return한다. 이 객체는 navigate()에 전달되고 navigation이 일어난다.
3. onClick 안에 safe args를 이용해서 destination 사이를 navigate한다.

▷ Navigate using ID

1. navigation.xml 안에 있는 destination이나 action의 ID를 가져온다
2. button의 setonclickIlstener 내부에서 navigation(id)를 이용해서 navigate 한다.

▷Navigation and the back stack

navigation은 backstack를 관리하면서 전 destination으로 돌아갈 수 있게 도와준다.

backstack을 수동으로 편집해야 하는 경우가 있다.

1. floating window: dialog 같은 floating window를 implement한 destination은 그전 destination을 나머지 화면에 보여준다. 이 경우는 원래 하던대로 backstack위에 destination을 쌓으면 된다. 하지만 dialog 위에 새로운 destination이 올라가게 되면 dialog 화면을 pop해줘야 한다.
2. 로그인 화면: 로그인을 한 뒤는 backstack을 모두 초기화해서 로그인 전 화면으로 돌아갈 수 없도록 해야 한다.

3월 24일

◇ Drawer 조작하기

▹ navigationUI 사용하기

1. navHost인 fragment container view를 이용해서 nav controller을 가져와야 한다.

```
navHostFragment = (NavHostFragment)
getSupportFragmentManager().findFragmentById(R.id.nav_host_fragment);
navController = navHostFragment.getNavController();
```

2. drawerlayout과 navview를 findviewbyid를 이용해 가져온다.

```
drawerLayout = findViewById(R.id.nav_drawer_layout);
navView = findViewById(R.id.nav_view);
```

3. appbarconfiguration을 이용해 navigation drawer을 만들고

```
appBarConfiguration = new AppBarConfiguration.Builder(navController.getGraph())
        .setOpenableLayout(drawerLayout).build();
```

4. setupActionBarWithNavController을 이용해 actionbar로 navdrawer열 수 있게 한다.

```
NavigationUI.setupActionBarWithNavController(this, navController, appBarConfiguration);
```

5. up 버튼이 눌리면 뒤로 돌아갈 수 있게 onSupportNavigationUp{}을 override한다.

```
@Override
public boolean onSupportNavigateUp() {
    return NavigationUI.navigateUp(navController, appBarConfiguration)
            || super.onSupportNavigateUp();
}
```

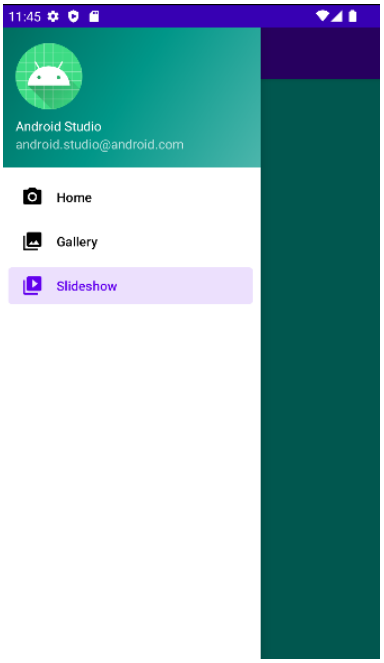
◇ 안드로이드 기본 icon 사용하기

res / new / vector asset / icon click

3월 25일

◇ NavigationUI 없이 NavigationDrawer 만들기

https://www.youtube.com/watch?v=zYVEMCiDcmY



▹ xml만들기

1. menu에 xml을 만들어 navigationDrawer에 들어갈 목록을 만든다.

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    tools:showIn="navigation_view">

    <group android:checkableBehavior="single">
        <item
            android:id="@+id/nav_home"
            android:icon="@drawable/ic_menu_camera"
            android:title="@string/menu_home" />
        <item
            android:id="@+id/nav_gallery"
            android:icon="@drawable/ic_menu_gallery"
            android:title="@string/menu_gallery" />
        <item
            android:id="@+id/nav_slideshow"
            android:icon="@drawable/ic_menu_slideshow"
            android:title="@string/menu_slideshow" />
    </group>
</menu>
```

2.

```
<com.google.android.material.navigation.NavigationView
  android:id="@+id/nav_view"
  android:layout_width="wrap_content"
  android:layout_height="match_parent"
  android:layout_gravity="start"
  android:fitsSystemWindows="true"
  app:headerLayout="@layout/nav_header_main"
  app:menu="@menu/activity_main_drawer" />
```

3. main.xml의 구성

```
<androidx.drawerlayout.widget.DrawerLayout >

  <androidx.coordinatorlayout.widget.CoordinatorLayout>

    <com.google.android.material.appbar.AppBarLayout>

      <androidx.appcompat.widget.Toolbar />

    </com.google.android.material.appbar.AppBarLayout>

    <FrameLayout />

  </androidx.coordinatorlayout.widget.CoordinatorLayout>

  <com.google.android.material.navigation.NavigationView />
</androidx.drawerlayout.widget.DrawerLayout>
```

▹ activity.java

1. NavigationView.OnNavigationItemSelectedListener 인터페이스를 implement하고 onNavigationItemSelected 메서드를 override하고 목록을 선택할 때 어떤 작업을 할지 정한다
   onBackPress메서드를 override하고 뒤로가기 누르면 navigationDrawer들어가게 만든다

```
public class MainActivity extends AppCompatActivity implements
NavigationView.OnNavigationItemSelectedListener

@Override
public boolean onNavigationItemSelected(@NonNull MenuItem item) {
  int id = item.getItemId();
  if (id == R.id.nav_home) {
     onFragmentSelected(0, null);
  } else if (id == R.id.nav_gallery) {
     onFragmentSelected(1, null);
  } else if (id == R.id.nav_slideshow) {
     onFragmentSelected(2, null);
  }
```

```
    drawer.closeDrawer(GravityCompat.START);
    return true;
  }

  @Override
  public void onBackPressed() {
    if(drawer.isDrawerOpen(GravityCompat.START)) {
        drawer.closeDrawer(GravityCompat.START);
    } else {
        super.onBackPressed();
    }
  }
```

2. drawer_layout을 찾고 기능 추가

```
drawer = findViewById(R.id.drawer_layout);
ActionBarDrawerToggle toggle = new ActionBarDrawerToggle(
      this, drawer, toolbar, R.string.navigation_drawer_open,
R.string.navigation_drawer_close);

drawer.addDrawerListener(toggle);
toggle.syncState();
```

3. ActionBarDrawerToggle을 사용하면 actionBar에 햄버거 아이콘이 생겨서 navigation drawer을 열었다가 닫았다 할 수 있다.
   ActionBarDrawerToggle 객체를 만들 때 시각장애인을 위한 string 값을 두개 인수로 전달해야 한다.

   res>string.xml에 string 두개를 추가한다.
```
<string name="navigation_drawer_open">Open navigation drawer</string>
<string name="navigation_drawer_close">Close navigation drawer</string>
```

4. navigationview안에 있는 item이 클릭되었을 때 작동할 listener을 연결하고 override한다.
```
NavigationView navigationView = findViewById(R.id.nav_view);
navigationView.setNavigationItemSelectedListener(this);
```

5. onBackPressed()를 override해서 뒤로가기를 눌렀을 때 drawer가 닫히게 하자
```
@Override
public void onBackPressed() {
  if(drawerLayout.isDrawerOpen(GravityCompat.START)) {
      drawerLayout.closeDrawer(GravityCompat.START);
  } else {
      super.onBackPressed();
  }
}
```

6. navigationView.setNavigationItemSelectedListener(this) 은 callback 어쩌구 때문에 this 대신 new OnNavigationItemSelectedListener을 써도 되지만
코드 좋아보이게 this 쓰고 activiity에 implements NavigationView.OnNavigationItemSelectedListener하고 onNavigationitemSelected을 override한다.

7. onNavigationItemSelected, setCheckedItem

```
if(savedInstanceState == null) {
  //initialize mainFragment to container
  getSupportFragmentManager().beginTransaction()
        .replace(R.id.container_frame, mainFragment).commit();
  //initialize home item in menu
  navigationView.setCheckedItem(R.id.menu_main);
}
```

◇ actionbar와 statusbar 색 지정

theme <item > 설정을 바꿔주면 noactionbar로 설정하고 statusbar를 투명하게 할 수 있다.

```
<!-- Status bar color transparent -->
<item name="android:statusBarColor">@android:color/transparent</item>
<!-- disable ActionBar and not presenting title -->
<item name="windowActionBar">false</item>
<item name="windowNoTitle">true</item>
```

◇ Gravity와 Layout_Gravity

gravity는 view 안에 있는 알맹이의 정렬을 설정해주고
layout_gravity는 layout안에 있는 view의 정렬을 설정해준다

3월 26일

◇ current wallpaper 가져오기 또는 변경

https://stackoverflow.com/questions/9939329/get-current-wallpaper

1. wallpaperManager 객체를 만든다. 지금 현재 wallpaper을 가져올거기 때문에 wallpapermanager.getinstance(this)을 사용한다.
   WallpaperManager wallpaperManager = WallpaperManager.getInstance(this)

2. drawable 파일로 wallpaper을 가져온다.
   Drawable wallpaperDrawable = wallpaperManager.getDrawable();

   final WallpaperManager wallpaperManager = WallpaperManager.getInstance(this);
   final Drawable wallpaperDrawable = wallpaperManager.getDrawable();

3. SecurityException tell you to grant READ_EXTERNAL_STORAGE permission

◇ How to request a run time permission

https://www.youtube.com/watch?v=SMrB97JuIoM
https://gist.github.com/codinginflow/5a149b51e7f38c3b1e5612849e055dc4

1. manifest.xml파일에 들어가서
   <manifest
   <uses-permission
   android:name=”android.permission.READ_EXTERNAL_STORAGE”/>

2. mainActivity에 들어가서
   private int STORAGE_PERMISSION_CODE=1; 선언한다

3. if (ContextCompat.checkSelfPermission() ==
   Packagemanager.PERMISSION_GRANTED)
   { … }
   else
   { requestStoragePermission(); }

4. requestStoragePermission() 밑에 선언한다.

   ActivityCompat.requestPermissions(Activity.this, new String[]
   {Manifest.permission.READ_EXTERNAL_STORAGE},
   STORAGE_PERMISSION_CODE); 이부분이 permission을 허용하는 부분이다.

5. onRequestPermissionResult를 Override해서 permission이 허용되었는지 확인해야 허용되었을 때 app을 계속 진행해야 한다.

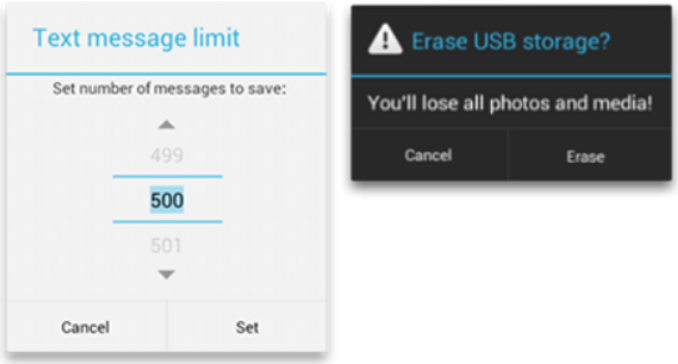
◇ Manifest.xml

a summary of a app

1. user permission: 인터넷 연결이 필요하거나 카메라 연결이 필요하거나 스마트폰에 memory에 접속해서 data를 읽고 써야 하거나 이런 경우 permission을 menifest에 등록해야 한다.
2. declare activities, services, content provider
3. declare api level
4. declare hardware and software used by an app: 카메라나 블루투스, 멀티 터치스크린과 연결해야 할 때 그들과 소통한다고 명시해야 한다.

3월 28일

◇Dialog



https://developer.android.com/guide/topics/ui/dialogs

▹ Dialog class 사용법
dialog의 기본적인 성질은 dialog class에 다 정의되어 있지만 subclass인 Alertdialog와 PickerDialog를 사용해야 한다.
progressdialog는 deprecate되었다.

▹Dialog와 DialogFragment
dialog class는 dialog의 모양과 구조를 정의하지만 dialogFragment를 내가 사용하고 싶어하는 dialog의 container로 삼아야 한다.

dialogfragment를 사용하여 dialog를 관리하면 dialog class와 그 안에 있는 여러 method를 사용할 필요 없이 dialog를 만들고 관리할 수 있다.

▹dialog의 활용
https://material.io/components/dialogs#usage

☆ dialog 만들기
일회적으로 dialog를 띄울 꺼면 그냥 activity에서 바로 buiuld dialog를 해도 된다.
하지만 dialog를 여러 번 띄우고 data교환을 해야 한다면 fragment로 만드는 것이 옳다

1. activity에서 간단하게 만들기

```
//start dialog
myDialog = new AlertDialog.Builder(MainActivity.this);
myDialog.setTitle(R.string.dialog_title);
//adding edittext
EditText editTextConfidence = new EditText(MainActivity.this);
editTextConfidence.setInputType(InputType.TYPE_CLASS_TEXT);
myDialog.setView(editTextConfidence);
//actions
myDialog.setPositiveButton("저장", new DialogInterface.OnClickListener() {
  @Override
```

```
    public void onClick(DialogInterface dialogInterface, int i) {
        positiveText = editTextConfidence.getText().toString();
    }
});
myDialog.setNegativeButton("취소", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {
        dialogInterface.cancel();
    }
});

myDialog.show();
```

2. fragment이용하기

▹dialog fragment 만들기

custom하고 싶은 xml을 만들고 class를 만든 다음에 dialogfragment를 extend한다.
oncratedialog()를 override해서 원하는 dialog를 build하고 xml를 inflate한다음에
builder.create()를 return해야 한다.

```
AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());

LayoutInflater inflater = getActivity().getLayoutInflater();
View view = inflater.inflate(R.layout.custom_dialog, null);

builder.setView(view)
        .setTitle(R.string.dialog_title);
//actions
builder.setPositiveButton("저장", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {

    }
});
builder.setNegativeButton("취소", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialogInterface, int i) {

    }
});

positiveText = view.findViewById(R.id.positive_text);

return builder.create();
```

▹dialog fragment를 activity에서 열기

```
CustomDialog customDialog = new CustomDialog();
customDialog.show(getSupportFragmentManager(), "positive saying dialog");
```

▹ custom layout 사용하기
If you want a custom layout in a dialog, create a layout and add it to an AlertDialog by calling setView() on your AlertDialog.Builder object.

```
AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
    // Get the layout inflater
    LayoutInflater inflater = requireActivity().getLayoutInflater();

    // Inflate and set the layout for the dialog
    // Pass null as the parent view because its going in the dialog layout
    builder.setView(inflater.inflate(R.layout.dialog_signin, null))
```

▹ fragment dialog와 activity연결하기
interface를 통해서 activity와 연결해야 한다.

◇IllegalStateException : The specified child already has a parent.

https://blog.naver.com/zion830/221274053542

view를 xml을 통해 만들지 않고
view에 .setview를 이용해 만들 때

view에 접근하기 위해 두번 클릭을 하면 setview가 두번 적용되어 오류가 뜬다.

결국 view를 만들고 디자인 하는 부분은 xml에서만 건들이는 것이 좋다.

◇icon 들어간 button

https://stackoverflow.com/questions/3634191/how-to-center-icon-and-text-in-a-android-button-with-width-set-to-fill-parent

material button을 사용하면 많은 기능이 있다.
app:icon과 app:iconGravity = textstart를 사용하면 된다.

◇ fragment안에서 dialog열기

https://www.youtube.com/watch?v=LUV_djRHSEY

▹쉬운방법(안되는 방법)

activity안에 B fragment가 있고 B fragment에서 dialog를 열고 싶을 때

1. B fragment에서 dialog를 열때
   똑같이 dialogfragment 객체를 만들어서 .show()메서드를 이용하면 된다.
   CustomDialog customDialog = new CustomDialog();
   customDialog.show(getfragmentManager(), tag);
2. dialog에서 B fragment view를 바꿀 때
   dialog에서 B fragment로 정보를 넘겨 주기 위해서
   간단하게 B fragment 객체를 만들어주고
   MainFragment mainfragment = (MainFragment)
   getActivity.getFragmentManager().findFragmentByTag("MainFragment");
   mainfragment.textview.setText();

   getActivity로 activity를 접속해 activity의 fragmentmanager을 가져와 바로 fragment에 접속해 fragment 안의 view를 바꾸는 것인데 좋은 방법은
   아니다

▹ 쉬운 방법(되긴하는데 제한적이고 올지 않음 인줄 알았으나 사람들이 많이 쓴다 옳지 않지는 않은데 약간 더러운 방법)

https://www.youtube.com/watch?v=QMwaNN_aM3U
https://www.youtube.com/watch?v=c6c1giRekB4

Fragment B 에 Timepicker.OnTimeListener를 implement한다.
onTimeSet 메서드를 override하고 그 안에서 골라진 시간으로 fragment view를 조작한다.
문제는 onCreateDialog에서 return 값이
TimePickerDialog(getActivity(), this, hour, minute
, DateFormat.is24HourFormat(getActivity()));
인데 this가 들어가면 안된다.

TimePickerDialog(Context context, TimePickerDialog.OnTimeSetListener listener, int hourOfDay, int minute, boolean is24HourView)
식으로 들어가야 한다. 결국 fragment의 listener와 연결해 줘야 한다.

▹ 옳은방법(안되는 방법)

1. dialog fragment 안에 listener interface를 만들고 메서드를 만든다.
2. 차이는 여기에서 난다.
   onAttach안에서 context와 dialog의 listener 객체를 연결할 때
   activity에서는 listener = (interface) context;
   아니면 listener = (interface) getActivity;
   를 사용하지만
   fragment안에서 dialog를 꺼낼 때는
   listener = (interface) getTargetFragment; 를 쓴다

getTargetFragment는 deprecate되었다.

3. fragment에서 dialog를 열 때도 다르다
   CustomDialog customDialog = new CustomDialog();
   customDialog.setTargetFragment(MainFragment.this, requestcode);

3월 29일

◇ fragment안에서 datepickerdialog fragment열기 using Bundle and setFragmentResult

https://www.youtube.com/watch?v=OEhSuv-_a9w

▹열기(???)
fragmentmanager를 activity로부터 가져와야 한다.
customPicker = new CustomPicker();
fragmentManager = requireActivity().getSupportFragmentManager();
customPicker.show(fragmentManager, "TimePickerFragment");

▹정보 가져오기
결국 성공하지 못했다. setFragmentResult나 viewmodel 또는 interface를 통해 소통해야 하는데 완벽하게 구현하지 못했다.

▹ java에서 picker구현하기

```
        TimePickerDialog.OnTimeSetListener onTimeSetListener = new
        TimePickerDialog.OnTimeSetListener() {
          @Override
          public void onTimeSet(TimePicker timePicker, int i, int i1) {
                start_hour = i;
                start_minute = i1;
                startTime.setText(String.format(Locale.getDefault(), "%02d:%02d", start_hour,
                 start_minute));  }
        };
TimePickerDialog timePickerDialog = new TimePickerDialog(getActivity(),
AlertDialog.THEME_HOLO_DARK, onTimeSetListener, start_hour,start_minute,true);
timePickerDialog.setTitle("시작 시간");
timePickerDialog.show();
```

◇ communicating with fragments

https://developer.android.com/guide/fragments/communicate

▹To properly react to user events, or to share state information, you often need to have channels of communication between an activity and its fragments or between two or more fragments. To keep fragments self-contained, you should not have fragments communicate directly with other fragments or with its host activity.
쉽게 설명을 하면 두 fragment는 직접적으로 통신을 하면 안된다. 그래서 보통은 communicate 전용 interface를 만든 다음 이것을 activity에 implement하고 그 interface를 통해 소통한다.

▹fragment 사이나 fragment와 activity 사이 user event, state information 공유 방법은 두가지가 있다.

1. ViewModel: 더 광범위한 상황에서 지속해야하는 data를 주고 받을 때
2. Fragment Result API: 일회성으로 넘겨주고 쓸나는 data이고 Bundle에 들어갈 크기면

▸ViewModel

▹viewmodel을 사용해야 할 때

1. activity 안에 두개의 fragment가 서로 소통을 해야 할 때
2. fragment 안에 또 다른 fragment가 있고 서로 소통을 할 때
   ※ fragment에서 dialog나 picker을 띄워도 이들은 activity 위에서 생성되는 것이다.
3. navigation view와 fragment와 소통을 할 때

☆ViewModel 이용하기

1. viewmodel에 필요한 dependencies을 추가하고
2. viewmodel java class를 만들고 viewmodel을 상속한다
3. private MutableLiveData<원하는 type> text = new MutableLiveData<>(); mutable을 사용해야만 setText를 사용해서 추가할 수 있다.
4. data를 더라하고 빼는 메서드를 추가한다.
5. fragmentA에 들어가서 onActivityCreated override해서 activity와 observer설정한다

◇ ViewModel

https://developer.android.com/topic/libraries/architecture/viewmodel

view의 구성이 바뀌고 fragment나 activity가 바뀔 때 그 안에 있던 data들은 저장되어야 한다.
그래야 다시 돌아왔을 때 data를 가지고 보여준다.
적은 data는 Bundle에서 onSaveInstanceState()로 저장되지만 더 큰 data는 힘들다.

▹ lifecycle of a ViewModel
viewmodel은 lifecycle이 완전히 끝날 때까지 memory에 남아 있는다. activity는 finished fragment는 detached

viewmodel은 oncreate에서 사용하며 oncreate이 여러번 불려도 finished되기 전까지는 살아있다.

▹share data between fragment
한 activity에 두 fragment가 들어가서 한 쪽의 list에서 item하나를 선택하면 그것을 다른 화면에 보여주려고 해보자. 이때 activity는 두 fragment로부터 정보를 주고 받아 view를 계속 바꿔줘야 하는데 이때 view model을 쓰면 된다.

Notice that both fragments retrieve the activity that contains them. That way, when the fragments each get the ViewModelProvider, they receive the same SharedViewModel instance, which is scoped to this activity.
This approach offers the following benefits:

- The activity does not need to do anything, or know anything about this communication.

- Fragments don't need to know about each other besides the SharedViewModel contract. If one of the fragments disappears, the other one keeps working as usual.
- Each fragment has its own lifecycle, and is not affected by the lifecycle of the other one. If one fragment replaces the other one, the UI continues to work without any problems.

3월 30일

◇ TimePickerDialog 생성하기

1. TimerPIckerDialog.onTimesetLIstener 인터페이스 객체를 activity에서 만든다. onTimeSet 메서드를 override하고 그 안에 선택한 값으로 바꿀 것을 적는다

```
TimePickerDialog.OnTimeSetListener onTimeSetListener = new
TimePickerDialog.OnTimeSetListener() {
  @Override
  public void onTimeSet(TimePicker timePicker, int i, int i1) {
        start_hour = i;
        start_minute = i1;
        startTime.setText(String.format(Locale.getDefault(), "%02d:%02d", start_hour,
         start_minute));  }
};
```

2. activity에서 TimePickerDialog 객체를 new로 생성하고 .show()를 통해서 activity에서 띄울 수 있다. new로 객체를 생성할 때 생성자로 activity의 context, theme, listener, hour, minute, 24시간 설정을 넣어줘야 한다.

```
TimePickerDialog timePickerDialog = new TimePickerDialog(getActivity(),
AlertDialog.THEME_HOLO_DARK, onTimeSetListener, start_hour,start_minute,true);
timePickerDialog.setTitle("시작 시간");
timePickerDialog.show();
```

◇ Launcher

https://namu.wiki/w/%EB%9F%B0%EC%B2%98

스마트폰의 배경화면 시작화면과 아이콘과 디자인을 바꿀 수 있는 앱 형태의 프로그램을 의미

◇ Lock task mode

https://developer.android.com/work/dpc/dedicated-devices/lock-task-mode

lock task mode에서는 사용자가 원해도 홈으로 돌아갈 수도 없고 알림을 받을 수도 없으며 허용된 앱 이외에 다른 앱에 접속할 수도 없다.

☆ screen pin도 lock task mode와 비슷하게 다른 앱의 접근을 받지만 screen pin은 유저가 원하면 screen pin 상태를 벗어날 수 있지만 lock task mode는 그렇지 않다.

DPC- device policy controller 에 허용되어 있는 앱을 제외한 앱들은 system이 lock task mode일 때는 run할 수 없다.

◇ EMM

enterprise mobility management는 회사에서 많은 양의 휴대폰과 테블릿을 회사 용도로 사용할 때 그 기계들을 회사에서 통제하기 위해 만들어진 기술들의 집합이다

mobile device management system은 회사에서 device들을 통제하고 모니터링 하기 위해 만들어진 application platform이다. mdm system은 device policy controller을 통해서 mdm 시스템과 device를 연결한다. 연결을 통해 system에서 전달하는 명령에 따르기도 하고 device와 정보를 주고 받는다

▹ android enterprise mobility Management API
android management api: enterprise token, application configuration, policy, device를 관리하는 method를 가지고 있다.

device administration api: application이 시작할 때 발동되어 device에 policy를 강제하고 device를 관리한다.
device administration api에 있는 devicepolicymanager 클래스와 deviceadminreciever 클래스를 이용하여 device policy controller는 device를 관리한다. device policy controller말고 app에서 이 두 class를 가져와 관리할 수도 있다.

▹method
mdm system이 device에 설치되어 있는 device policy controller(DPC)를 통해 device를 관리하는데 쓰이는 method가 두개 있다.

1. policy: 관리되고 있는 device가 어떻게 동작해야 하는지 정의해 놓은 여러 setting의 집합
   하나의 device는 하나의 policy의 명령만 받아 제어가 된다. policy가 조정할 수 있는 기능은 다음과 같다.
   - disable hardware components
   - Restrict access to the system
   - define installed application
   - configure network settings
2. command: mdm system에 device에 내리는 명령이고 device policy controller에 의해 이 명령들이 수행되고 관리된다. 이 명령은 3가지가 있다.
   - lock device
   - reboot device
   - reset device

▹ device administration(관리)
device의 주인이 누구냐에 따라 권한의 범위가 정해진다. 개인이 그 device의 주인이면 DPC의 권한이 적고 work profile 내에서 admin(관리자)정도의 역할을 하지만 회사가 그 device의 주인이면 DPC는 전체 시스템에 대한 권한을 가지고 다른 공간이 할당되지 않는다.

device owner mode로 바꾸려면 device를 처음 initial settup할 때 설정해야 한다.

device의 주인이 누구이냐에 따라 DPC는 profile owner 또는 device owner의 권한을 가지게 되고 각각일 때 할 수 있는 권한의 정도가 다르다.

- profile owner: device의 상태(cpu, 와이파이 등등)을 모니터링 할 수 있고, hardware 접근을 막을 수 있고 password policy를 설정할 수 있다.
- device owner: profile owner일 때 할 수 있는 모든 것 + 사용자가 setting에 들어가는 것을 막고 + wifi bluetooth 등등 radio 수신을 막을 수 있고 공장 초기화를 시킬 수 있다.
- google play emm api ?

▹화면 고정
화면을 고정하는 방법은 두 가지가 있다.

1. screen pin: 이미 구글에서 지원하는 기능이다.
2. lock task mode: lock task mode는 dpc가 device owner일 때만 가능하다.

◇ library? framework? platform? api?

https://blog.gaerae.com/2016/11/what-is-library-and-framework-and-architecture-and-platform.html

library: 프로그램 제작시 필요한 기능들을 class나 function으로 만들어서 쉽게 사용할 수 있게 만든 data의 집합

framework:

platform: 프로그램 실행 환경으로 이 환경 위에서 프로그램이 돌아간다.

api:

4월 1일

◇ Super와 Super()

http://www.tcpschool.com/java/java_inheritance_super

1. super 키워드
   parent.class를 child.class가 상속받았을 때 child안에서 super.을 이용해 parent안에 있는 메소드나 변수를 참조해서 조작할 수 있다.
2. super() 메소드
   parent.class를 child.class가 상속받았을 때 child안에서 super()을 이용해 parent가 필요로 하는 생성자와 함께 parent.class를 불러와 부모의 생성자를 호출한다.

```java
// ex.3)
class A{
    public void doA() {
        System.out.println("A 클래스의 doA() 입니다.");
    }
}
class B extends A{
    B(){
        // A 클래스의 doA() 함수를 호출합니다.
        super.doA();
    }
}

public class Ex01 {
    public static void main(String[] args) {
        new B();
        // 결과 : A 클래스의 doA() 입니다.
    }
}
```

```java
// ex.4)
class AAA{
    AAA(){
        System.out.println("AAA 생성자");
    }
    AAA(int a){
        System.out.println(a);
    }
}
class BBB extends AAA{
    BBB(){
        super();
        System.out.println("BBB 생성자");
    }
    BBB(int a){
        super(a);
        System.out.println("a = " + a);
    }
}
public class Ex01 {
    public static void main(String[] args) {
        // BBB 클래스의 BBB(int a) 를 호출하면
        // 부모 클래스 AAA 클래스의 AAA(int a) 를 호출합니다.

        /* 결과:
         * 20
         * a = 20
         *
         */
        new BBB(20);
    }
}
```

◇ java의 List 자료형

list는 array와 비슷하나 크기가 정해져 있지 않고 저장되는 정보의 양에 따라 동적으로 변한다는 특징이 있다.

☆ Generics

<>안에는 어떤 객체를 list안에 저장할 것인지 명시하는 것이 generics이다. 이때 기본 자료형 뿐만 아니라 내가 만든 class 예를 들면 <Dog> <Animal>을 넣어서 그 객체를 저장할 수 있다.

ArrayList<String> pitches = new ArrayList<String>();
보통 뒷 부분의 자료형은 생략 가능하다.
ArrayList<String> pitches = new ArrayList<>();

▹ list 자료형은 interface여서 이 interface를 구현한 하위 자료형이 있고 이들을 쓴다.

1. arrayList 자료형
2. Vector 자료형
3. LinkedList 자료형

◇ AutoCompleteTextView

editText처럼 사용자에게 입력을 받아서 list나 array에 저장되어 있는 data 검색을 도와준다.

1. findViewById로 view를 가져오고
   AutoCompleteTextView searchCurrentApp;
   searchCurrentApp = rootView.findViewById(R.id.search_current_app);

2. 검색하고 싶은 data의 묶음을 만든다
   String[] testData = {"one", "two", "three", "four", "fourr", "five"};

3. adapter객체를 만들고 argument로 context, 보여주는 방식 xml, data를 넣어야 한다.

   ```
   ArrayAdapter<String> adapter = new ArrayAdapter<String>(getActivity(),
           android.R.layout.simple_dropdown_item_1line, testData);
   searchCurrentApp.setAdapter(adapter);
   ```

◇ For Each문

array와 list에서만 사용할 수 있는 for문으로 자료형 안에 있는 모든 객체에 대해 for문 안에 있는 반복문을 반복한다.

```
String[] numbers = {"one", "two", "three"};
for(int i=0; i<numbers.length; i++) {
    System.out.println(numbers[i]);
}
```

```
String[] numbers = {"one", "two", "three"};
for(String number: numbers) {
    System.out.println(number);
}
```

◇ Recycler view, Card view

1. layout xml에 recycler view를 추가한다
2. root layout이 cardview인 새로운 xml를 만든다. 여기에는 한 item을 표시하는 모양을 만든다.
3. data를 임시로 저장할 java class를 만든다. 이 class의 목적은 원하는 자료형을 하나로 묶을 class를 만드는데 있다.

```
public class AppInfo {
  //저장을 원하는 자료형을 모아주고
  String packageName;
  String appName;
  Drawable appLogo;
  boolean appStatus;

  //equals hash emthod추가
  @Override
  public boolean equals(Object o) {
     if (this == o) return true;
     if (!(o instanceof AppInfo)) return false;
     AppInfo appInfo = (AppInfo) o;
     return Objects.equals(getPackageName(), appInfo.getPackageName());
  }

  @Override
  public int hashCode() {
     return 0;
  }

 // getting setting method 추가
  public String getPackageName() {
     return packageName;
  }

  public void setPackageName(String packageName) {
```

```
        this.packageName = packageName;
    }

    public String getAppName() {
        return appName;
    }

    public void setAppName(String appName) {
        this.appName = appName;
    }

    public Drawable getAppLogo() {
        return appLogo;
    }

    public void setAppLogo(Drawable appLogo) {
        this.appLogo = appLogo;
    }

    public boolean isAppStatus() {
        return appStatus;
    }

    public void setAppStatus(boolean appStatus) {
        this.appStatus = appStatus;
    }
}
```

4. CustomAdapter class를 만들어서
   RecyclerView.Adapter<AppListAdapter.ViewHolder>를 상속한다.
5. CustomAdapter class 안에 ViewHolder class만든다.

```
public class ViewHolder extends RecyclerView.ViewHolder {
    //object data for the card view
    ImageView appLogo;
    TextView appName;
    CheckBox appStatus;
    public ViewHolder(@NonNull View itemView) {
        //using super connect objects with input View
        super(itemView);
        appLogo = itemView.findViewById(R.id.app_logo);
        appName = itemView.findViewById(R.id.app_name);
        appStatus = itemView.findViewById(R.id.app_status);
    }
}
```

6. 그 외 method들도 override해준다.

```
List<AppInfo> appList;

Context context;
```

```
//constructor for AppListAdapter
//adding appInfos input into appList
AppListAdapter(List<AppInfo> appInfos) {
  this.appList=appInfos;
}

@NonNull
@Override
public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
  if(context == null)
     context = parent.getContext();
  View view = LayoutInflater.from(context).inflate(R.layout.app_item, parent, false);
  ViewHolder viewHolder = new ViewHolder(view);

  return viewHolder;
}

@Override
public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
  holder.appLogo.setImageDrawable(appList.get(position).appLogo);
  holder.appName.setText(appList.get(position).appName);
  if(appList.get(position).appStatus)
     holder.appStatus.setChecked(true);
  else
     holder.appStatus.setChecked(false);
}

@Override
public int getItemCount() {
  //return size to notify how many card needed to be made
  return appList.size();
}
```

7. mainActivity에서 recycler 객체와 layoutmanager, adapter연결하기

```
//recycler 객체와 view 연결하기
RecyclerView appList;
appList = rootView.findViewById(R.id.app_list);

//layoutmanager 설정하고 .setlayoutmanager로 recycler객체에 연결하기
GridLayoutManager gridManager = new GridLayoutManager(getActivity(), 4);
appList.setLayoutManager(gridManager);

// setAdapter를 이용해서 adapter을 연결한다.
AppListAdapter adapter = new AppListAdapter(apps);
appList.setAdapter(adapter);
```

◇ 저장되어 있는 app 표시하기

https://www.youtube.com/watch?v=2ztOyBQemsU&t=1337s]

```
public void loadAppList() {
   PackageManager packageManager = getActivity().getPackageManager();
   Intent intent = new Intent();
   intent.setAction(Intent.ACTION_MAIN);
   intent.addCategory(Intent.CATEGORY_LAUNCHER);
   //getting current app
   List<ResolveInfo> homeApps = packageManager.queryIntentActivities(intent, 0);
   //making a new list
   List<AppInfo> apps = new ArrayList<>();
   for (ResolveInfo info: homeApps) {
      //passing data from homeApp List to apps List
      AppInfo appInfo = new AppInfo();
      appInfo.setAppLogo(info.activityInfo.loadIcon(packageManager));
      appInfo.setPackageName(info.activityInfo.packageName);
      appInfo.setAppName((String) info.activityInfo.loadLabel(packageManager));
      apps.add(appInfo);
   }
   AppListAdapter adapter = new AppListAdapter(apps);
   appList.setAdapter(adapter);
}
```

4월 1일

◇ Realm dependency 설정하기

https://www.mongodb.com/docs/realm/sdk/java/install/

project level gradle 최상단에 이거 추가하고 sync 누르기

```
buildscript {
    repositories {
        google()
        jcenter()
    }
    dependencies {
        classpath "com.android.tools.build:gradle:3.5.1"
        classpath "io.realm:realm-gradle-plugin:10.10.1"
    }
}
```

sync 된 후 app level gradle 상단 근처에 이거 추가하고 sync 누르기

```
apply plugin:
'realm-android'
```

단. apply plugin은 plugin { ... } 밑에 들어가야 한다.

◇ Realm Quick Start

https://www.mongodb.com/docs/realm/sdk/java/quick-start-local/

realm 잘 안되서 그냥 SQLite으로 넘어감

▹initializing realm
application이 처음 run했을 때 한번 realm을 초기화해줘야 한다.
Realm.init(context);
를 이용해 초기화할 수 있는데 context에는 activity fragment application의 context가 들어가도 상관없다. 하지만 한 번만 처음에 초기화 하기 위해 onCreate()에서 .init을 해주는 것이 좋다.

▹data model만들기
어떤 형태 조합으로 data를 묶어 realm DB에 넣을 것인지는 결정해 줘야 한다.
RealmObject를 상속하는 class를 만들어서 그 안에 저장하고 싶은 자료형의 변수 그리고 그 변수에 접속할 getter setter 메서드 그리고 상속자를 만들어 준다.
각 묶음을 대표할 key를 설정하기 위해 @PrimaryKey 태그 밑에 변수를 하나 설정해 준다.

▹Open a realm

RealmConfiguration을 이용해서 DB에 있는 data중에 내가 열고 싶은 data를 특정한다. name과 location를 특정하고 synchronous read write을 할 지 결정한다.

```
String realmName = "My Project";
RealmConfiguration config = new
RealmConfiguration.Builder().name(realmName).build();
```

▹What for changes
OrderedRealmCollectionChangeListener 객체와 addChangeListener() 메서드를 이용해 변화를 관찰할 수 있다.

4월 6일

◇ SQLite 문법

https://intellipaat.com/blog/tutorial/sql-tutorial/

▹Table
Sql은 표의 형식으로 data를 저장한다.
그리고 표 안에 몇개의 column을 넣을 것인지 명시해야 한다.
표 안의 row 개수는 결국 우리가 data를 저장한 만큼 생성된다.

여러 개의 column이 모여서 table이 되고 여러 table이 모여서 database를 이룬다

▹Create Table in SQL
1. table의 이름을 설정하고
2. 그 table에 들어갈 column의 개수와 각각 이름을 정해준다
3. 각 column에 들어갈 data type를 정한다

▹Drop Table in SQL
Sql에서 drop a table은 그 table과 그 안에 있는 data를 삭제한다는 의미이다.

▹SQL insert into Query
insert 구문은 하나의 record(row)를 table에 추가할 때 사용한다. 이때 추가하는 data는 반드시 table과 같은 개수의 column을 가지고 각 column안에 저장된 자료형도 같아야 한다.

INSERT INTO tablename VALUES (value1, value2,….valueN);

▹Select statement in SQL
select구문은 table에서 정해진 규칙에 따라 data를 가져오거나 table 전체를 가져오는 것을 도와준다
select구문은 column을 선택할 수는 있지만 column에서 어떤 row를 고를지는 선택할 수 없다.

SELECT column1, column2, columnN
FROM tablename;

▹Where statement in SQL
where문을 이용하면 select구문에서 조건을 추가할 수 있다.

SELECT column1, column2, columnN
FROM tablename
WHERE [condition];

▹ SQL에서 연산자
where문에서 condition을 추가할 때는 and, or, not 조건문을 사용할 수 있다.

```
SELECT column1, column2, …, columnN
FROM tablename
WHERE [condition1], … AND [conditionN];

SELECT column1, column2, … columnN
FROM tablename
WHERE [condition1], …, OR [conditionN];

SELECT column1, column2, … columnN
FROM tablename
WHERE NOT [condition1];
```

▹IN 연산자
select문을 이용해서 column을 고른 다음 where과 in을 이용해서 row를 고를 수 있다.

```
SELECT column1, column2….
FROM table_name
WHERE column1 IN (value1, value2….);
```

▹BETWEEN 연산자
select문을 이용해서 column을 고른 다음 where과 between을 이용해서 특정 구간의 row를 가져올 수 있다.

```
SELECT column1, column2….
FROM table_name
WHERE column1 BETWEEN Value1 AND Value2;
```

▹DELETE & TRUNCATE in SQL
DELETE는 특정 row나 row전부를 지우는 명령어이다. 조건문을 만족하는 row는 모두 삭제된다.

```
DELETE FROM table_name
[WHERE condition];
```

TRUNCATE는 table의 구조 빼고 안에 저장되어 있는 모든 row를 삭제한다. 구조란 처음에 설정한 column의 개수와 저장할 자료형이다.

```
TRUNCATE TABLE table_name;
```

▹UPDATE in SQL
update란 data가 일단 한번 저장된 다음 이미 저장되어 있는 data를 고칠 때를 의미한다.
추가를 할 때는 insert into 문을 쓰면 된다.
한 row안에 있는 여러 columnn을 고칠 수도 있고 한 column안에 여러 row를 한번에 고칠 수 있다.

```
UPDATE table_name
SET col1=val1, col2=val2…
```

[Where condition];

4월 7일

◇Save data using SQLite

▹ SQLite의 장단점
자동으로 정보를 업데이트 시켜주지 않고 모든 명령을 작성해야 해서 Room Persistence Library를 이용하는 것을 추천한다.

▹ Define a Schema and contract
schema는 data를 어떤 방식으로 저장 할 지 선언해 놓는 것이다. 그러면 그 양식대로 data를 저장해 나갈 수 있다.
schema 안에 contract class를 또 정의할 수 있는데 contract는 table, column 이름을 상수로 선언한다. 그래서 package안에 있는 다른 class에서 contract에 접속해 column 이름을 바꾸면 column을 사용하는 모든 곳에서 이름이 바뀐다.

```
public final class FeedReaderContract {
    // To prevent someone from accidentally instantiating the contract class,
    // make the constructor private.
    private FeedReaderContract() {}

    /* Inner class that defines the table contents */
    public static class FeedEntry implements BaseColumns {
        public static final String TABLE_NAME = "entry";
        public static final String COLUMN_NAME_TITLE = "title";
        public static final String COLUMN_NAME_SUBTITLE = "subtitle";
    }
}
```

▹ create database using SQL helper
SQLiteOpenHelper를 상속하는 class를 만들어 그 안에 override된 onCreate메서드에서 database를 만들어야 한다.

```
private static final String SQL_CREATE_ENTRIES =
    "CREATE TABLE " + FeedEntry.TABLE_NAME + " (" +
    FeedEntry._ID + " INTEGER PRIMARY KEY," +
    FeedEntry.COLUMN_NAME_TITLE + " TEXT," +
    FeedEntry.COLUMN_NAME_SUBTITLE + " TEXT)";

private static final String SQL_DELETE_ENTRIES =
    "DROP TABLE IF EXISTS " + FeedEntry.TABLE_NAME;


public class FeedReaderDbHelper extends SQLiteOpenHelper {
    // If you change the database schema, you must increment the database version.
```

```
    public static final int DATABASE_VERSION = 1;
    public static final String DATABASE_NAME = "FeedReader.db";

    public FeedReaderDbHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(SQL_CREATE_ENTRIES);
    }
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        // This database is only a cache for online data, so its upgrade policy is
        // to simply to discard the data and start over
        db.execSQL(SQL_DELETE_ENTRIES);
        onCreate(db);
    }
}
```

만든 database에 접속하려면 helper의 객체를 만들고 new를 이용해 초기화 해야한다. 이때 인자에는 context가 들어간다.

```
FeedReaderDbHelper dbHelper = new FeedReaderDbHelper(getContext());
```

▷Put information into a database

```
// Gets the data repository in write mode
SQLiteDatabase db = dbHelper.getWritableDatabase();

// Create a new map of values, where column names are the keys
ContentValues values = new ContentValues();
values.put(FeedEntry.COLUMN_NAME_TITLE, title);
values.put(FeedEntry.COLUMN_NAME_SUBTITLE, subtitle);

// Insert the new row, returning the primary key value of the new row
long newRowId = db.insert(FeedEntry.TABLE_NAME, null, values);
```

▷ Read information from a database

일단 getReadableDatabase() 메서드를 이용해 read모드로 들어간다.

```
SQLiteDatabase db = dbHelper.getReadableDatabase();
```

db.query()메서드나 db.rawquery()메서드를 이용해서 읽고 싶은 datagroup을 만들고 그 group을 가리키는 cursor 객체를 return 받는다.

```
String query = "select positive_saying from tb_positive order by _id desc limit 1";
Cursor cursor = DB.rawQuery(query, null);
```

반드시 while cursor.movetonext()를 이용해 처음에 -1 위치에 있던 cursor를 오른쪽으로 움직이면서 data를 읽어야 한다.

```
List itemIds = new ArrayList<>();
while(cursor.moveToNext()) {
  long itemId = cursor.getLong(
```

```
            cursor.getColumnIndexOrThrow(FeedEntry._ID));
        itemIds.add(itemId);
    }
    cursor.close();
```

▷ Delete information from a database

```
// Define 'where' part of query.
String selection = FeedEntry.COLUMN_NAME_TITLE + " LIKE ?";
// Specify arguments in placeholder order.
String[] selectionArgs = { "MyTitle" };
// Issue SQL statement.
int deletedRows = db.delete(FeedEntry.TABLE_NAME, selection, selectionArgs);
```

▷Update a database

▷Persisting database connection

getWritableDatabase()와 getReadableDatabase()는 database가 close()되어 있을 때는 접속하기 힘드니 database를 최대한 오랫동안 열어 놓아야 한다. 따아서 보통 Activity의 onDestroy()메서드에서 db를 close()한다.

```
@Override
protected void onDestroy() {
    dbHelper.close();
    super.onDestroy();
}
```

◇ DATA Types in SQLite

https://www.sqlite.org/datatype3.html

▷Storage Class and Data type

Storage Class는 SQLite database에 저장될 수 있는 자료형의 묶음이다. Storage Class의 종류는 null, integer, real, text, blob가 있다. 각 class는 data의 길이에 따라 더 세분화 되지만 세분화되어 저장될 뿐 입력할 때나 출력할 때는 다시 5가지 storage class의 type으로 변환된다.

▷Boolean type

Boolean type는 따로 존재하지 않는다. 대신 integer자료형으로 저장하고 sql은 true와 false를 1과 0으로 변역해서 integer에 저장한다.

▷Date and Time type

Date and Time type는 따로 존재하지 않는다. 대신 Date Time Funtion을 이용해 text, real, integer로 변환해서 저장하면 된다.

4월 8일

◇Service

https://developer.android.com/guide/components/services

service는background에서 user interface없이 계속 작동하는 application component이다.

▹종류

- background service: 앱이 run하고 있을 때만 동작하고 앱이 terminate되면 동작을 멈춤
- foreground service: 앱이 run하고 있지 않아도 계속 살아서 동작하는 service이다. foreground는 반드시 notification을 띄워야 한다.
- bound service: service와 연결된 component가 살아있을 때만 동작하는 service이다. 이때 service는 다른 component와 bindservice()로 연결되어야 한다.
-

▹basic of service

service를 상속하는 class를 만들어줘야 한다. 그리고 그 안에 service lifecycle을 관리하는 method들을 override해야 한다. 필요하다면 다른 component를 service에 bind시켜줄 method도 추가해야 한다.

▹important callback method to override

- onstartcommand(): 다른 component에서 startservice()를 이용해 service를 시작했을 때 이 method가 call된다. 이 때 service를 멈추기 위해 stopself()나 stopservice()를 이용해 service를 멈추는 코드를 적어줘야 한다.

- onbind(): 다른 component가 bindservice()를 이용해 service와 연결되고 싶어할 때 call된다.

- oncreate(): service가 최초로 생성되었을 때 call된다. 이미 service가 run하고 있으면 oncreate는 call되지 않는다.

- onDestroy():

◇AlarmManager을 이용해서 Activity 열기

▹alarmmanager을 이용해서 지정된 시간에 intent가 발생하도록 한다.

1. alarmmanager 객체를 만들고 getSystemService를 이용해 초기화해준다
   //alarm manager instance from the system service
   AlarmManager alarmManager = (AlarmManager)
   this.getSystemService(Context.ALARM_SERVICE);

2. broadcast를 만들고 pending intent를 이용해 broadcast와 연결한다
   //create an alarm receiver, which waits and listens for an alarm event

```
Intent intentAlarmReceiver = new Intent(this, AlarmReceiver.class);

//send alarm receiver to broadcast
PendingIntent pendingIntent = PendingIntent.getBroadcast(this, 0,
intentAlarmReceiver, 0);
```

3. pending intent와 알람 시간을 인자로 전달해서 alarmmanager을 set한다
```
//set time for alarm and attach to broadcast
alarmManager.set(AlarmManager.RTC_WAKEUP, System.currentTimeMillis()+5000
,pendingIntent);
```

▹pendingintent.getbroadcast를 이용해 broadcast에게 앱을 시작할 권한을 준다.
https://velog.io/@haero_kim/Android-PendingIntent-%EA%B0%9C%EB%85%90-%EC%9D%B5%ED%9E%88%EA%B8%B0

pending intent는 기다리고 있는 intent이다. 내가 나중에 intent를 이용해 앱을 구동해야 하는데 지금은 다른 작업을 하고 싶으면 다른 앱에 intent를 쓸수 있는 권한을 주어서 다른 앱을 사용하고 있다가 특정 시점이 되면 내가 원하는 앱을 intent를 통해 run할 수 있다.

▹pending intent와 flag
https://developer88.tistory.com/187

루틴앱

4월 16일

◇ 익명 하위 클래스

https://limkydev.tistory.com/226

▹익명 하위 클래스를 사용하는 이유
익명 하위 클래스는 이름이 없다는 것은 별로 기억되지 않아도 된다는 거겠죠..

나중에 다시 불러질 이유가 없다는 뜻입니다. 이 말을 좀 더 있어보이게 말하면...
프로그램에서 일시적으로 한번만 사용되고 버려지는 객체입니다. 좀 더 풀어서 생각해보면 일시적으로 사용된다는 것은 나중에 재사용이 되지 않는다는 것이며, 재사용이 될 필요가 없다는 뜻은 확장성이 그렇게 좋지 못하다는 뜻입니다.

▹익명클래스 구현 방법
익명클래스 구현 방법은 두 가지가 있는데 상속관계에 있어서 익명객체를 만드는 방법과 인터페이스를 기반으로 익명객체를 만드는 것이다

▹인터페이스 기반 익명객체
?

4월 17일

◇toolbar title center align하기

toolbar view 안에 textview 넣어서 layout_gravity를 center하면된다

```
<androidx.appcompat.widget.Toolbar
        android:id="@+id/toolbar_main"
        android:layout_width="match_parent"
        android:layout_height="?attr/actionBarSize"
        android:background="@color/black"
        app:title=""
        android:elevation="4dp"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar"
        app:popupTheme="@style/ThemeOverlay.AppCompat.Light">

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

```
        android:layout_height="wrap_content"
        android:text="TODAY"
        android:layout_gravity="center"
        android:textSize="30sp"/>

    </androidx.appcompat.widget.Toolbar>
```

4월 18일

◇ Collapsible Toolbar

```
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/purple_500"
    tools:context=".SubActivity">

    <com.google.android.material.appbar.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="280dp"
        android:fitsSystemWindows="true"
        android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">

        <com.google.android.material.appbar.CollapsingToolbarLayout
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:fitsSystemWindows="true"
            app:contentScrim="?attr/colorPrimary"
            app:layout_scrollFlags="scroll|snap|exitUntilCollapsed"
            app:title=" "
            android:id="@+id/sub_toolbar">

            <androidx.appcompat.widget.Toolbar
                android:layout_width="match_parent"
                android:layout_height="?attr/actionBarSize"
                app:title=" "
                app:layout_collapseMode="pin">

                <TextView
                    android:id="@+id/title_sub"
```

```
                android:layout_width="378dp"
                android:layout_height="wrap_content"
                android:text="WORKOUT"
                android:textColor="@color/white"
                android:gravity="center"
                android:textSize="30sp"/>

        </androidx.appcompat.widget.Toolbar>

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="GOOL: 99DAYS"
            android:textSize="20sp"
            android:textColor="@color/white"
            android:layout_gravity="center_horizontal"
            android:layout_marginTop="100dp"/>
    </com.google.android.material.appbar.CollapsingToolbarLayout>

</com.google.android.material.appbar.AppBarLayout>

<androidx.core.widget.NestedScrollView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/sub_activity_bg"

app:layout_behavior="com.google.android.material.appbar.AppBarLayout$ScrollingViewBeh
avior">



</androidx.core.widget.NestedScrollView>

</androidx.coordinatorlayout.widget.CoordinatorLayout>
```

▹root layout 을 coordinatorlayout으로 하고 그 안에 appbarlayout을 넣는다. appbarlayout의 크기는 height를 확대되었을 때 기준으로 한다

```
<androidx.coordinatorlayout.widget.CoordinatorLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/purple_500"
    tools:context=".SubActivity">
```

```
<com.google.android.material.appbar.AppBarLayout
    android:layout_width="match_parent"
    android:layout_height="280dp"
    android:fitsSystemWindows="true"
    android:theme="@style/ThemeOverlay.AppCompat.Dark.ActionBar">
```

▹appbarlayout안에 collapsible toolbar layout을 넣고 그 안에 또 toolbar를 넣는다. collapsible toolbar의 title은 확장되었을 때 왼쪽 아래에 잡히게 되고 수축했을 때 위로 올라간다. toolbar의 제목은 수축과 상관없이 위쪽에 자리잡는다.

```
    <com.google.android.material.appbar.CollapsingToolbarLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fitsSystemWindows="true"
        app:contentScrim="?attr/colorPrimary"
        app:layout_scrollFlags="scroll|snap|exitUntilCollapsed"
        app:title=" "
        android:id="@+id/sub_toolbar">

        <androidx.appcompat.widget.Toolbar
            android:layout_width="match_parent"
            android:layout_height="?attr/actionBarSize"
            app:title=" "
            app:layout_collapseMode="pin">
```

▹ collapsible toolbar layout 안에 내용을 넣기 위해서는 collapsible toolbar layout 안에 원하는 구성성분을 넣어주고 이들은 수축 했을 때 안 보이게 된다. 그리고 layout margin을 이용해서 위치를 조정할 수 있다.

```
 <com.google.android.material.appbar.CollapsingToolbarLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:fitsSystemWindows="true"
        app:contentScrim="?attr/colorPrimary"
        app:layout_scrollFlags="scroll|snap|exitUntilCollapsed"
        app:title=" "
        android:id="@+id/sub_toolbar">

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="GOOL: 99DAYS"
            android:textSize="20sp"
            android:textColor="@color/white"
            android:layout_gravity="center_horizontal"
            android:layout_marginTop="100dp"/>
    </com.google.android.material.appbar.CollapsingToolbarLayout>
```

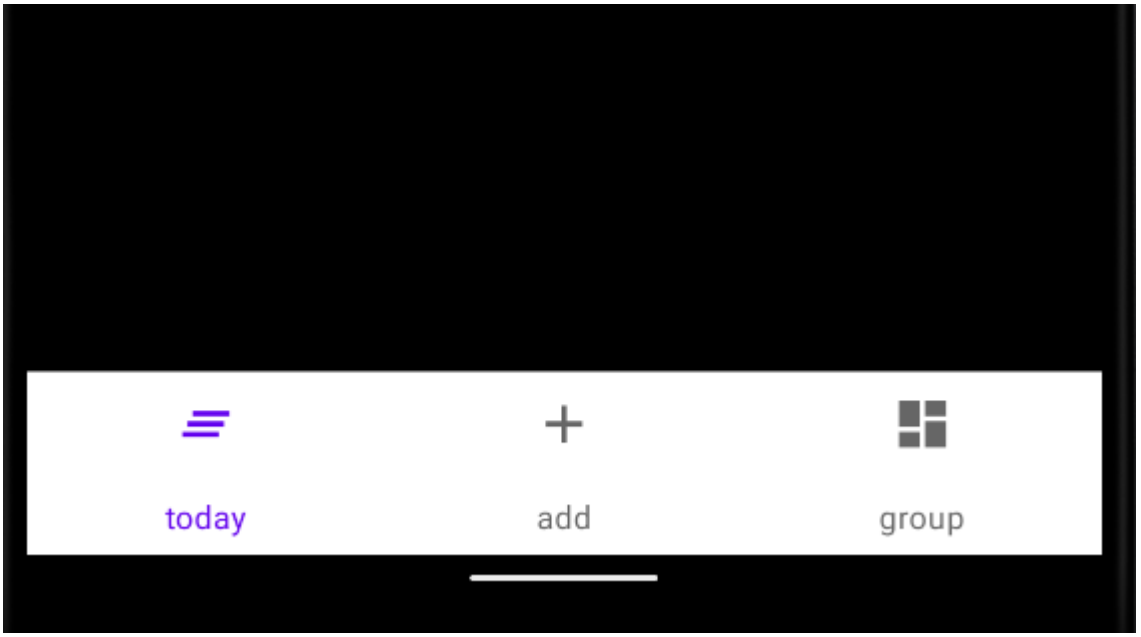
▹ toolbar가 수축했을 때 아래에 숨겨진 내요을 보여주기 위해서는 nestedscroolview 안에 내용을 담으면 된다.

```
<androidx.core.widget.NestedScrollView
      android:layout_width="match_parent"
      android:layout_height="match_parent"
      android:background="@drawable/sub_activity_bg"

app:layout_behavior="com.google.android.material.appbar.AppBarLayout$ScrollingViewBehavior">



</androidx.core.widget.NestedScrollView>
```

◇Custom CollapsibleToolbar

둥근 모양을 만들기 위해서는 collapsible toolbar의 background와 coordinatorlayout의 background가 동일해야 하며 drawable을 이용해 왼쪽 귀퉁이만 둥근 drawable 파일을 만들어주고 NestedScrollView
의 background를 둥글게 설정해주면 된다.

◇ Bottom Navigation



▹bottomnavigationView를 추가하고 바닥에 붙인다 자동으로 바닥으로 가지 않는다

```
<com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/bottom_navigation"
        android:layout_width="match_parent"
        android:layout_height="70dp"
        android:background="@color/white"
        app:menu="@menu/main_bottom_navigation"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent" />
```

▹menu 파일을 만들어 bottom navigation을 위한 menu를 만든다
menu에 showAsAction설정은 안해도된다. view.xml파일에서 조작할 수 잇다

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto">

   <item
      android:id="@+id/main_nav_today"
      android:icon="@drawable/list"
      android:title="@string/main_nav_today"
      app:showAsAction="always"/>
   <item
      android:id="@+id/main_nav_add"
      android:icon="@drawable/add"
      android:title="@string/main_nav_add"
      app:showAsAction="always"/>
   <item
      android:id="@+id/main_nav_group"
      android:icon="@drawable/grid"
      android:title="@string/main_nav_group"
```

```
        app:showAsAction="always"/>

</menu>
```

▹bottom navigation View안에 app:menu 로 menu파일을 연결한다

```
        app:menu="@menu/main_bottom_navigation"
```

▹item을 눌렀을 때 동작을 인식하기 위해서는 일단 onItemSelectedListener를 implement하고 onNavigationItemSelected메서드를 오버라이드한다.

```
        public class MainActivity extends AppCompatActivity implements
        NavigationBarView.OnItemSelectedListener{
         @Override
           public boolean onNavigationItemSelected(@NonNull MenuItem item) {
              Fragment selectedFragment = null;

              switch (item.getItemId()) {
                 case R.id.main_nav_today:
                    Log.d(TAG, "onNavigationItemSelected: today selected");
                    selectedFragment = new TodayFragment();
                    break;
                 case R.id.main_nav_add:

                    Log.d(TAG, "onNavigationItemSelected: add selected");
                    Intent intent = new Intent(getApplicationContext(), SubActivity.class);
                    startActivity(intent);
                    break;
                 case R.id.main_nav_group:
                    Log.d(TAG, "onNavigationItemSelected: group selected");
        //            selectedFragment = new GroupFragment();
                    break;
              }

              getSupportFragmentManager().beginTransaction().replace(R.id.container_main
                    , selectedFragment).commit();

              return true;
           }
```

◇bottom navigation view객체를 만들어 xml파일이랑 연결하고 그 객체에 setOnItemListener(context)를 연결한다.

```
        bottomNavigationView = findViewById(R.id.bottom_navigation);
        //set listener for bottom navigation
        bottomNavigationView.setOnItemSelectedListener(this);
```

◇ Custom Bottom Navigation

https://material.io/components/bottom-navigation/android#using-bottom-navigation

icon, icontext 색변경

android:background="@color/black"
app:itemIconTint="#6C6C6C"
app:itemTextColor="@color/black"

icon text visibility 결정

labelVisibiliityMode

◇StatusBar 색상 변경 및 조작

https://developer.android.com/reference/android/view/Window.html#setStatusBarColor(int)

```
public void transparentStatusBar() {
    Window window = this.getWindow();
    window.setStatusBarColor(ContextCompat.getColor(this, R.color.black));
}
```

◇ Recycler View

https://developer.android.com/guide/topics/ui/layout/recyclerview?gclid=Cj0KCQjwmPSSBhCNARIsAH3cYgaJdvLS0z2CGNL1LvVfHqjMNzmlTgL_GMSuRgv19E0OkrxjgrHBXboaAlLoEALw_wcB&gclsrc=aw.ds

▹recycler view란

recycler view는 대량의 data를 원하는 묶음으로 편집해서 보여주기 용의하다. 전체적인 view는 스크롤이 가능한데 각각의 item을 보여주는 view는 없어지지 않고 재사용 된다. 스크롤을 하면서 view안에 들어 있는 data만 교체되는 것이다.

▹recycler view를 구성하는 class

- RecyclerView 클래스: RecyclerView 클래스는 viewGroup클래스를 상속한다. cardView를 묶는 viewGroup이다.
- ViewHolder 클래스: list안에 있는 개별 element는 viewHolder객체로 관리된다. viewHolder안에는 data가 없다. viewHolder객체가 만들어지면 RecycleView 클래스가 viewHolder와 data를 결합한다.
- Adapter 클래스: data와 viewholder를 결합하는 것이 Adapter 클래스 안에 있는 메서드이다.
- LayoutManager클래스: LayoutManager클래스가 각각의 element를 list안에 정렬시킨다.

▹recycler view 디자인 순서

1. layout manager를 정해서 각각의 element들을 어떻게 정렬할 지 정한다. linearlayoutmanager, gridlayoutmanager, staggeredgridlayoutmanager 들 중 하나를 정하거나 custom layoutmanager을 만든다.
2. individual element를 담당하는 view를 xml파일을 통해 디자인한다.

3. Adapter, ViewHolder 클래스를 만든다
   이 클래스들은 data가 어떻게 display될 지 결정한다. viewholder는 각 element를 담당하는 view를 관리하고 adapter는 viewholder 객체를 만들어 필요한 data를 할당한다. 이 과정을 binding이라고 한다.

4. Adapter 클래스를 상속할 때 반드시 3가지 메서드를 오버라이드해야 한다.
   - onCreateViewHolder(): 여기서 ViewHolder객체가 생성된다. 그리고 한 element를 담당하는 xml파일에 할당된다. 하지만 여기 안에 data는 아직 들어있지는 않다.
   - onBindViewHolder(): 여기서 binding이 진행된다. 각 element를 담당하는 xml파일의 여러 widget에 값을 넣는다.
   - getItemCount(): 이 메서드를 통해 recycler view는 data의 전체 크기를 안다

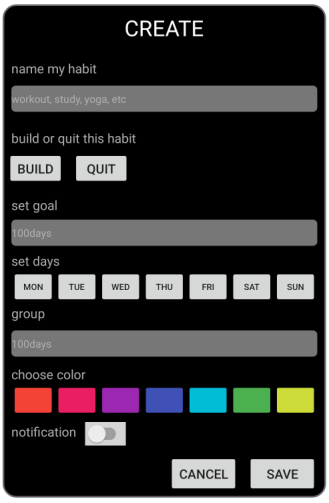
```
public class CustomAdapter extends RecyclerView.Adapter<CustomAdapter.ViewHolder> {

    private String[] localDataSet;

    /**
     * Provide a reference to the type of views that you are using
     * (custom ViewHolder).
     */
    public static class ViewHolder extends RecyclerView.ViewHolder {
        private final TextView textView;

        public ViewHolder(View view) {
            super(view);
            // Define click listener for the ViewHolder's View

            textView = (TextView) view.findViewById(R.id.textView);
        }

        public TextView getTextView() {
            return textView;
        }
    }

    /**
     * Initialize the dataset of the Adapter.
     *
     * @param dataSet String[] containing the data to populate views to be used
     * by RecyclerView.
     */
    public CustomAdapter(String[] dataSet) {
        localDataSet = dataSet;
    }

    // Create new views (invoked by the layout manager)
```

```
    @Override
    public ViewHolder onCreateViewHolder(ViewGroup viewGroup, int viewType) {
        // Create a new view, which defines the UI of the list item
        View view = LayoutInflater.from(viewGroup.getContext())
                .inflate(R.layout.text_row_item, viewGroup, false);

        return new ViewHolder(view);
    }

    // Replace the contents of a view (invoked by the layout manager)
    @Override
    public void onBindViewHolder(ViewHolder viewHolder, final int position) {

        // Get element from your dataset at this position and replace the
        // contents of the view with that element
        viewHolder.getTextView().setText(localDataSet[position]);
    }

    // Return the size of your dataset (invoked by the layout manager)
    @Override
    public int getItemCount() {
        return localDataSet.length;
    }
}
```

recyclerview with sqlite database in android
https://www.youtube.com/watch?v=ow0Fj0xhjR0&list=PLGY_UftsHsIZDnXQk4BLUb-voVO4hQMmm

◇Positive Negative Button까지 Custom인 dialog 만들기

▹버튼까지 있는 xml파일을 만든다




```
//    public void openAddHabit() {
//        createDialog.setContentView(R.layout.dialog_add_habit);
//        //changing background transparent
//        createDialog.getWindow().setBackgroundDrawable(new ColorDrawable((Color.TRANSPARENT)));
//
//        Button saveButton = createDialog.findViewById(R.id.create_save_btn);
//        Button cancelButton = createDialog.findViewById(R.id.create_cancel_btn);
//        saveButton.setOnClickListener(new View.OnClickListener() {
//            @Override
//            public void onClick(View view) {
//                createDialog.dismiss();
//            }
//        });
//        cancelButton.setOnClickListener(new View.OnClickListener() {
//            @Override
//            public void onClick(View view) { createDialog.dismiss(); }
//        });
//        createDialog.show();
//    }
```

▹activity에서 모든 코드를 다 쓸 경우 Dialog 객체를 만들고 new로 초기화를 해준 다음 setContentView()를 이용해 xml파일을 inflate해주면된다.

Dialog settingDialog, createDialog;

```
@Override
  protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);recycler
      setContentView(R.layout.activity_main);
      //find view by id
      settingDialog = new Dialog(this);
      createDialog = new Dialog(this);
```

```
public void openSetting() {
      settingDialog.setContentView(R.layout.dialog_setting);    }
```

그리고 보여주고 싶은 dialog를 제외한 나머지 화면을 불투명하게 보이게 한다.

```
 //changing background transparent
  settingDialog.getWindow().setBackgroundDrawable(new
ColorDrawable((Color.TRANSPARENT)));
```

버튼 객체를 만들고 view와 binding해줘서 click listener를 연결한다.

```
      Button saveButton = settingDialog.findViewById(R.id.setting_save_btn);
      saveButton.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
            settingDialog.dismiss();
         }
      });
```

마지막으로 .show()메서드를 이용해 화면으로 띄우면 된다.

```
      settingDialog.show();
```

▹ 하지만 완전 custom dialog를 사용하면 view와 binding할 버튼의 수가 많아진다. 그럴 때는 새로운 class를 만들어 dialog를 담당하게 하면 좋다.

우선 dialogfragment를 상속하는 클래스를 만들고 onCreateDialog()메서드를 오버라이드한다.

```
public class HabitDialog extends DialogFragment {
   @NonNull
   @Override
   public Dialog onCreateDialog(@Nullable Bundle savedInstanceState) {
```

다음으로 dialog 객체를 만들어주고 new로 초기화하되 getActivity()를 인자로 전달해서 보여지는 activity의 context를 가져오게 한다. 그리고 아까와 동일하게 화면의 나머지 부분을 불투명하게 하고 xml파일과 binding한다.

```
      Dialog createDialog = new Dialog(getActivity());
       createDialog.getWindow().setBackgroundDrawable(new
ColorDrawable((Color.TRANSPARENT)));
       createDialog.setContentView(R.layout.dialog_add_habit);
```

button 객체를 만들고 view와 binding 후 listener를 연결한다.

```
       Button saveButton = createDialog.findViewById(R.id.create_save_btn);
       Button cancelButton = createDialog.findViewById(R.id.create_cancel_btn);
       saveButton.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View view) {
             createDialog.dismiss();
          }
       });
       cancelButton.setOnClickListener(new View.OnClickListener() {
```

```
        @Override
        public void onClick(View view) { createDialog.dismiss(); }
    });
```

마지막으로 아까 만든 dialog 객체를 return한다.

```
    return createDialog;
```

dialog를 activity에서 보여주고 싶을 때는 만들어 놓은 class의 객체를 불러오고 .show()를 이용해 화면에
띄운다.

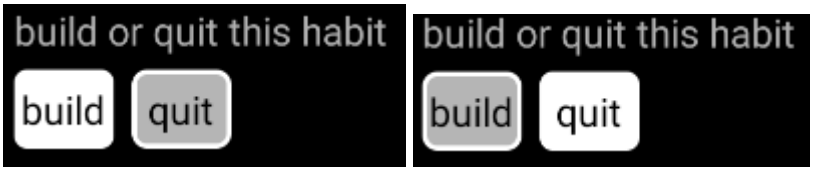
```
  public void openAddHabit() {
    HabitDialog habitDialog = new HabitDialog();
    habitDialog.show(getSupportFragmentManager(), "create new habit");
  }
```

◇ 다중 선택, 단일 선택 버튼 구현

▹버튼의 종류

- Button
- ImageButton
- Toggle: 눌렀을 때 안에 글자가 변한다
- Switch: 스위치처럼 생긴 버튼으로 동그란 원형이 오른쪽 왼쪽으로 움직인다
- Floating:
- Checkbox: 둘 이상의 선택이 가능한 버튼
- Radio button: 오직 하나만 선택이 가능한 버튼

▹ 단일 선택 버튼 구현 Custom Radio Button



xml파일 안에 radio group을 설정하고 그 안에 radio button을 넣는다

```
<RadioGroup
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="5dp"
        android:layout_marginLeft="10dp">

        <RadioButton
            android:layout_width="38dp"
            android:layout_height="30dp"
            android:gravity="center"
            android:text="build"
```

```
            android:textSize="15sp"
            android:textColor="@drawable/text_selector"
            android:background="@drawable/radio_selector"
            android:button="@android:color/transparent"/>

        <RadioButton
            android:layout_width="38dp"
            android:layout_height="30dp"
            android:gravity="center"
            android:layout_marginLeft="7dp"
            android:text="quit"
            android:textSize="15sp"
            android:textColor="@drawable/text_selector"
            android:background="@drawable/radio_selector"
            android:button="@android:color/transparent"/>

    </RadioGroup>
```

drawable <shape>파일을 만들어 버튼이 선택되었을 때 그리고 선택되지 않았을 때 버튼 모양을 만들어준다. 그리고 drawable <selector>파일을 두개 만들어 radio_selecter와 text_selecter를 만들어준다

```
// radio_selecter
<selector xmlns:android="http://schemas.android.com/apk/res/android">

   <item
      android:state_checked="true"
      android:drawable="@drawable/radio_selected"/>
   <item
      android:state_checked="false"
      android:drawable="@drawable/radio_unselected"/>

</selector>

//text_selecter
<selector xmlns:android="http://schemas.android.com/apk/res/android">

   <item
      android:state_checked="true"
      android:color= "#B5B5B5"/>

   <item
      android:state_checked="false"
      android:color= "@color/black"/>

</selector>
```

radiobutton안에 다음 속성을 부여한다

```
        android:textColor="@drawable/text_selector"
        android:background="@drawable/radio_selector"
        android:button="@android:color/transparent"
```

▹다중 선택 버튼 구현 Custom CheckBox

다중 선택은 check box를 사용하는 것이 좋다 방법은 radio button과 동일

```
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginStart="10dp"
    android:layout_marginRight="10dp"
    android:orientation="horizontal">
    <CheckBox
        android:layout_width="38dp"
        android:layout_height="30dp"
        android:text="mon"
        android:gravity="center"
        android:textColor="@color/black"
        android:background="@drawable/radio_selector"
        android:button="@android:color/transparent"/>
    <CheckBox
        android:layout_width="38dp"
        android:layout_height="30dp"
        android:layout_marginLeft="7dp"
        android:text="tue"
        android:gravity="center"
        android:textColor="@color/black"
        android:background="@drawable/radio_selector"
        android:button="@android:color/transparent"/>
    <CheckBox
        android:layout_width="38dp"
        android:layout_height="30dp"
        android:layout_marginLeft="7dp"
        android:text="wed"
        android:gravity="center"
        android:textColor="@color/black"
        android:background="@drawable/radio_selector"
        android:button="@android:color/transparent"/>
    <CheckBox
        android:layout_width="38dp"
        android:layout_height="30dp"
        android:layout_marginLeft="7dp"
        android:text="thu"
        android:gravity="center"
        android:textColor="@color/black"
        android:background="@drawable/radio_selector"
        android:button="@android:color/transparent"/>
    <CheckBox
```

```
        android:layout_width="38dp"
        android:layout_height="30dp"
        android:layout_marginLeft="7dp"
        android:text="fri"
        android:gravity="center"
        android:textColor="@color/black"
        android:background="@drawable/radio_selector"
        android:button="@android:color/transparent"/>
    <CheckBox
        android:layout_width="38dp"
        android:layout_height="30dp"
        android:layout_marginLeft="7dp"
        android:text="sat"
        android:gravity="center"
        android:textColor="@color/black"
        android:background="@drawable/radio_selector"
        android:button="@android:color/transparent"/>
    <CheckBox
        android:layout_width="38dp"
        android:layout_height="30dp"
        android:layout_marginLeft="7dp"
        android:text="sun"
        android:gravity="center"
        android:textColor="@color/black"
        android:background="@drawable/radio_selector"
        android:button="@android:color/transparent"/>
```

▹custom color picker button
radio button을 이용해서 색을 고르는 버튼을 만드려 한다 색은 중복해서 선택하지 않으니 radio button을 이용하는 것이 좋다
radiobutton의 속성에서 색을 바꾸고 text를 없앤 다음에 여백을 없애면 된다.

```
    <RadioButton
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:buttonTint="#FFC107"
    android:minWidth="0dp"
    android:minHeight="0dp" />
```

◇ Custom switch

switch는 크게 두 부분으로 나뉜다. 동그란 원형 부분은 thumb이고 on/off에 따라 움직이는 경로는 track이라고 불린다. thumb와 track를 담당하는 drawable를 xml파일로 만들어 줘야 한다.
▹ on off일 때 변화를 주려면 <selector>를 선택하고 각<item>에 state_checked 속성을 넣어줘야 다른 모양을 보여줄 수 있다.

```
//track
<selector xmlns:android="http://schemas.android.com/apk/res/android"
    android:shape="rectangle">
   <item android:state_checked="true">
      <shape android:shape="rectangle">
         <solid android:color="@color/white"/>
         <corners android:radius="25dp"/>
      </shape>
   </item>
   <item android:state_checked="false">
      <shape android:shape="rectangle">
         <solid android:color="#6A6A6A"/>

         <corners android:radius="25dp"/>
      </shape>
   </item>
</selector>
```

▹on/off에 따라 변화가 필요 없으면 <shape>으로 만들어도 된다.

```
//thumb
<shape xmlns:android="http://schemas.android.com/apk/res/android"
   android:shape="oval">
   <solid android:color="#B5B5B5"/>
   <stroke android:color="@color/white" android:width="2dp"/>
   <size android:height="25dp" android:width="25dp"/>
</shape>
```

▹마지막으로 appcompat switch를 main layout안에 넣고 다음 속성을 추가한다.

```
<androidx.appcompat.widget.SwitchCompat
android:id="@+id/swt_ntf"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginLeft="10dp"
android:thumb="@drawable/thumb"
app:track="@drawable/track"
android:checked="true"
app:showText="true"
android:textOn="on"
android:textOff="off"/>
```

◇ 다수의 button의 onclick메서드를 연결해야 하는 경우

https://www.youtube.com/watch?v=GtxVILjLcw8

깔끔해 보일 수 있지만 사실 의미가 없고 case문에 의해 오히려 더 복잡해질 수도 있다.

▹각각 모든 버튼에 setOnCickedListener를 연결할 필요 없이 버튼이 들어 있는 class에 View.OnClickedListener를 implement하고 onclicked method를 오버라이드하면 된다.

public class HabitDialog extends DialogFragment implements View.OnClickListener {

▹버튼 객체와 setOnClickLIstener를 연결한다.

```
//set on click listener
btn_save.setOnClickListener(this);
btn_cancel.setOnClickListener(this);
```

▹onClick() 메서드를 오버라이드해서 switch문으로 어떤 버튼이 눌린 건지 확인한다.

◇JAVA FIELD

java의 클래스 내부는 field와 method으로 구성된다.
feild는 클래스 내에 정의 되어 있는 변수를 의미한다. 멤버변수, 전역변수라고도 한다.
객체가 만들어졌을 때 field는 객체의 정보를 저장하는 용도로 사용될 수 있고
method는 객체의 동작을 선언하는 용도로 사용될 수 있다.

◇Radio Button에서 선택된 button찾는 법

radiogroup 객체를 만들어 view와 바인딩하고 .getCheckedRadioButtonId() 메서드를 이용해서 선택된 radiobutton의 xml id를 가져올 수 있다. 이때 아무 것도 선택되지 않으면 -1을 return한다.

```
//find with button is checked
int StatusId = rg_status.getCheckedRadioButtonId();
int ColorID = rg_color.getCheckedRadioButtonId();
```

◇ CheckBox 선택되었는지 확인

checkbox 객체를 만들고 view와 binding한 다음에 checkbox.isChecked()메서드를 이용해 check되어 있으면 true를 없으면 false를 return한다.

```
private ArrayList<Integer> getCheckedDays() {
    ArrayList<Integer> selectedDays = new ArrayList<>();
    CheckBox[] days = {cb_mon, cb_tue, cb_wed, cb_thu, cb_fri, cb_sat, cb_sun};
    for(int i = 0; i < days.length; i++) {
        if(days[i].isChecked()) {
            selectedDays.add(i);
        }
    }
    return selectedDays;
}
```

◇ List에 값이 있는지 확인하기

https://retrieverj.tistory.com/80
DB의 selectList를 통해서 얻어오는 값이 NULL인지 체크하고 싶을 때,
list == null로 비교해보면, 안타깝게도 false가 나온다.
이유는 return값이 []이기 때문이다.

list.isEmpty()메소드를 쓰자

◇Checkbox, RadioButton, EditText값을 입력하지 않으면 확인 button을 disable하기

https://www.youtube.com/watch?v=Vy_4sZ6JVHM
▹button의 속성 중에 android:enable 속성을 추가해서 button의 초기 상태 false로 해서 disable시킨다.

```
<androidx.appcompat.widget.AppCompatButton
            android:id="@+id/create_save_btn"
            android:layout_width="80dp"
            android:layout_height="35dp"
            android:layout_marginRight="10dp"
            android:layout_marginBottom="10dp"
            android:text="save"
            android:background="@drawable/radio_unselected"
            android:textSize="15sp"
            android:enabled="false"/>
```

▹editText와 button이 입력되었는지 확인하는 메서드를 만든다.

```
private void enableSaveButton() {
    //charSequence is text input by three editText view
    //so need to get input directly from the view
    String habitNameInput = et_name.getText().toString().trim();
    String habitGoalInput = et_goal.getText().toString().trim();
    String habitGroupInput = et_group.getText().toString().trim();
    //find with button is checked
    int StatusId = rg_status.getCheckedRadioButtonId();
    int ColorId = rg_color.getCheckedRadioButtonId();
    ArrayList<Integer> DaysId = getCheckedDays();

    Boolean editTextFilled = !habitNameInput.isEmpty() && !habitGoalInput.isEmpty()
            && !habitGroupInput.isEmpty();
    Boolean buttonChecked = (StatusId != -1) && (ColorId != -1) && !DaysId.isEmpty();

    btn_save.setEnabled( buttonChecked && editTextFilled);
}
```

editText의 경우는 .getText로 입력 값을 String에 저장하고 String.isEmpty로 비어있는지 확인한다
radiobutton의 경우는 radioGroup .getCheckedRadioButtonId() 메서드를 이용해 radioGroup 안에서 선택된 버튼의 id를 가져온다. 그리고 선택이 없을 경우 -1를 반환하므로 -1과 비교하면 된다.

```
String habitNameInput = et_name.getText().toString().trim();
        String habitGoalInput = et_goal.getText().toString().trim();
        String habitGroupInput = et_group.getText().toString().trim();
        //find with button is checked
        int StatusId = rg_status.getCheckedRadioButtonId();
        int ColorId = rg_color.getCheckedRadioButtonId();
```

여러개의 checkbox들 중 하나라도 선택되었는지를 확인하기 위해서는 view와 binding되어있는 checkbox 객체에 .isChecked() 메서드를 연결해 check 유무를 알 수 있다. check되어 있는 checkbox의 id를 ArrayList<Integer>안에 저장한다.

```
//cb_mon, cb_tue, cb_wed, cb_thu, cb_fri, cb_sat, cb_sun는 모두 findViewByid로 view와 연결하였다
private ArrayList<Integer> getCheckedDays() {
        ArrayList<Integer> selectedDays = new ArrayList<>();
        CheckBox[] days = {cb_mon, cb_tue, cb_wed, cb_thu, cb_fri, cb_sat, cb_sun};
        for(int i = 0; i < days.length; i++) {
            if(days[i].isChecked()) {
                selectedDays.add(i);
            }
        }
        return selectedDays;
    }
```

check되어 있는 checkbox의 id를 ArrayList<Integer>안이 비어있는지 확인하기 위해 .isEmpty()메서드를 이용한다.

```
ArrayList<Integer> DaysId = getCheckedDays();
!DaysId.isEmpty()
```

▹EditText 편집 확인
TextWatcher field를 만들고 클릭이 이루어질 때마다 buttonEnable메서드를 불러와 확인작업을 하게 한다.

```
 private TextWatcher editTextWatcher = new TextWatcher() {
        @Override
        public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {       }
        @Override
        public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
            enableSaveButton();
        }
        @Override
        public void afterTextChanged(Editable editable) {       }
    };
```

그리고 editText.addTextChangeListenerr에 TextWatcher field를 전달한다.

```
et_name.addTextChangedListener(editTextWatcher);
et_goal.addTextChangedListener(editTextWatcher);
et_group.addTextChangedListener(editTextWatcher);
```

▹RadioButton 편집 확인

RadioGroup.OnClickedChangeListener field를 만들고 클릭이 이루어질 때마다 buttonEnable메서드를 불러와 확인작업을 하게 한다.

```
private RadioGroup.OnCheckedChangeListener radioButtonWatcher = new
RadioGroup.OnCheckedChangeListener() {
    @Override
    public void onCheckedChanged(RadioGroup radioGroup, int i) {
        enableSaveButton();
    }
};
```

그리고 radioButton.setOnCheckedChangeListener에 OnCheckedChangeListener field를 전달한다.

```
rg_color.setOnCheckedChangeListener(radioButtonWatcher);
rg_status.setOnCheckedChangeListener(radioButtonWatcher);
```

▹Checkbox 편집 확인

View.OnClickListener checkboxWatcher field를 만들고 클릭이 이루어질 때마다 buttonEnable메서드를 불러와 확인작업을 하게 한다.

```
private View.OnClickListener checkboxWatcher = new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        enableSaveButton();
    }
};
```

그리고 checkbox.setOnClickListener에 View.OnClickListener checkboxWatcher field를 전달한다.

```
private void checkboxListener() {
    cb_mon.setOnClickListener(checkboxWatcher);
    cb_tue.setOnClickListener(checkboxWatcher);
    cb_wed.setOnClickListener(checkboxWatcher);
    cb_thu.setOnClickListener(checkboxWatcher);
    cb_fri.setOnClickListener(checkboxWatcher);
    cb_sat.setOnClickListener(checkboxWatcher);
    cb_sun.setOnClickListener(checkboxWatcher);
}
```

◇ <루틴앱> SQL database 만들기

https://www.youtube.com/watch?v=hJPk50p7xwA&list=PLSrm9z4zp4mGK0g_0_jxYGgg3os9tqRUQ&index=1

```
public class HabitDatabaseHelper extends SQLiteOpenHelper {
    //TAG
    private static final String TAG = "HabitDatabaseHelper";
    //database create var
    private static final String DATABASE_NAME = "habit.db";
    private static final int DATABASE_VERSION = 1;
    //table & column name
    private static final String TABLE_NAME = "tb_habit";
    private static final String COLUMN_NAME = "name";
    private static final String COLUMN_STATUS = "build_or_quit";
    private static final String COLUMN_GOAL = "goal";
    private static final String COLUMN_MON = "mon";
    private static final String COLUMN_TUE = "tue";
    private static final String COLUMN_WED = "wed";
    private static final String COLUMN_THU = "thu";
    private static final String COLUMN_FRI = "fri";
    private static final String COLUMN_SAT = "sat";
    private static final String COLUMN_SUN = "sun";
    private static final String COLUMN_GROUP = "h_group";
    private static final String COLUMN_COLOR = "color";
    private static final String COLUMN_NOTIFY = "notification";

    //query
    private static final String CREATE_QUERY = "create table tb_habit (_id integer primary
key autoincrement, name text, build_or_quit boolean" +
            ", goal integer, mon boolean, tue boolean, wed boolean, thu boolean, fri boolean, sat
boolean, sun boolean, h_group text" +
            ", color integer, notification boolean);";
    private static final String UPDATE_QUERY = "drop table tb_habit";


    public HabitDatabaseHelper(@Nullable Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase sqLiteDatabase) {
        sqLiteDatabase.execSQL(CREATE_QUERY);
    }

    @Override
    public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
        sqLiteDatabase.execSQL(UPDATE_QUERY);
```

```
        onCreate(sqLiteDatabase);
    }
```

▹sqliteopenhelper클래스를 상속하는 클래스를 만들고 constructor를 만들어 줘야 한다. constructor는 context인자만 놔두고 나머지는 삭제한다.

```
public HabitDatabaseHelper(@Nullable Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
```

▹그리고 상수 설정을 이용해 database이름, database 버전, table이름, 사용할 column을 설정해 줘야한다.

```
    //TAG
    private static final String TAG = "HabitDatabaseHelper";
    //database create var
    private static final String DATABASE_NAME = "habit.db";
    private static final int DATABASE_VERSION = 1;
    //table & column name
    private static final String TABLE_NAME = "tb_habit";
    private static final String COLUMN_NAME = "name";
    private static final String COLUMN_STATUS = "build_or_quit";
    private static final String COLUMN_GOAL = "goal";
    private static final String COLUMN_MON = "mon";
    private static final String COLUMN_TUE = "tue";
    private static final String COLUMN_WED = "wed";
    private static final String COLUMN_THU = "thu";
    private static final String COLUMN_FRI = "fri";
    private static final String COLUMN_SAT = "sat";
    private static final String COLUMN_SUN = "sun";
    private static final String COLUMN_GROUP = "h_group";
    private static final String COLUMN_COLOR = "color";
    private static final String COLUMN_NOTIFY = "notification";
```

▹그리고 사용할 query, 즉 sql명령어 문장들을 선언해 준다. 필수적이지는 않으나 유지보수에 용의해진다.

```
    //query
    private static final String CREATE_QUERY = "create table tb_habit (_id integer primary
key autoincrement, name text, build_or_quit boolean" +
            ", goal integer, mon boolean, tue boolean, wed boolean, thu boolean, fri boolean, sat
boolean, sun boolean, h_group text" +
            ", color integer, notification boolean);";
    private static final String UPDATE_QUERY = "drop table tb_habit";
```

▹ create database using SQL helper

SQLiteOpenHelper를 상속하는 class를 만들어 그 안에 override된 onCreate메서드에서 database를 만들어야 한다.

```
@Override
  public void onCreate(SQLiteDatabase sqLiteDatabase) {
     sqLiteDatabase.execSQL(CREATE_QUERY);
  }
```

▹oncreate 메서드를 오버라이드해서 db.execSQL 메서드와 database create query를 이용해 database을 만들어 준다. 인자는 id, 그리고 사용할 column을 전달하면 된다

```
@Override
  public void onUpgrade(SQLiteDatabase sqLiteDatabase, int i, int i1) {
     sqLiteDatabase.execSQL(UPDATE_QUERY);
     onCreate(sqLiteDatabase);
  }
```

◇String을 Int로 바꿀 때 주의점 valueof, parseInt

editText의 input type을 설정하면 입력할 때 키보드 모양을 설정할 수 있다.
보통은 두가지 type를 많이 쓴다. numberdecimal과 text이다.

text의 경우 그 안의 값을 가져올 때 getText()메서드를 이용한다.
et_name.getText().toString().trim();

numberDecimal일때 Integer.valueOf()메서드를 쓰면 안되고 Integer.parseInt()메서드를 이용해야 한다.
goal = Integer.parseInt(et_name.getText().toString().trim());

▹valueOf와 parseInt의 차이
https://www.geeksforgeeks.org/integer-valueof-vs-integer-parseint-with-examples/#:~:text=valueOf()%20returns%20an%20Integer,()%20returns%20a%20primitive%20int.&text=Both%20String%20and%20integer%20can,passed%20as%20parameter%20to%20Integer.

그냥 보통은 parseInt()메서드를 쓴다.

1. Integer.valueOf() returns an Integer object while Integer.parseInt() returns a primitive int.
2. Both String and integer can be passed a parameter to Integer.valueOf() whereas only a String can be passed as parameter to Integer.parseInt().
3. Integer.valueOf() can take a character as parameter and will return its corresponding unicode value whereas Integer.parseInt() will produce an error on passing a character as parameter.

◇ <루틴앱> SQL database에 정보 저장하기

https://www.youtube.com/watch?v=RGzblJuat1M&t=371s

```
//SQL database에 있는 저장 메소드
void addHabit(String name, Boolean status, Integer goal, Boolean mon, Boolean tue,
Boolean wed, Boolean thu, Boolean fri
        , Boolean sat, Boolean sun, String group, int color, Boolean notification) {
    SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();

    //use content value objet to pass data groups that we want to save
    ContentValues contentValues = new ContentValues();
    contentValues.put(COLUMN_NAME, name);
    contentValues.put(COLUMN_STATUS, status);
    contentValues.put(COLUMN_GOAL, goal);
    contentValues.put(COLUMN_MON, mon);
    contentValues.put(COLUMN_TUE, tue);
    contentValues.put(COLUMN_WED, wed);
    contentValues.put(COLUMN_THU, thu);
    contentValues.put(COLUMN_FRI, fri);
    contentValues.put(COLUMN_SAT, sat);
    contentValues.put(COLUMN_SUN, sun);
    contentValues.put(COLUMN_GROUP, group);
    contentValues.put(COLUMN_COLOR, color);
    contentValues.put(COLUMN_NOTIFY, notification);

    //pass content value to sql database object
    long result = sqLiteDatabase.insert(TABLE_NAME, null, contentValues);

    if(result == -1){
        Log.d(TAG, "addHabit: write data failed");
    }
}

//dialog fragment 클래스에 있는 view에 입력되는 값을 DB로 보내는 메서드
private void saveNewHabitData() {
    HabitDatabaseHelper habitDatabaseHelper = new HabitDatabaseHelper(getActivity());
    //get data from the view
    String name, group ;
    int goal, color;
    Boolean mon, tue, wed, thu, fri, sat, sun, status, notification;
    //get edit text data
    name = et_name.getText().toString().trim();
    goal = Integer.parseInt(et_name.getText().toString().trim());
    group = et_group.getText().toString().trim();
    //get build quit data (build == true, quit == false)
    if(rg_status.getCheckedRadioButtonId() == R.id.radio_build) {
        status = true;
```

```
}else {
    status = false;
}
//get color data
switch(rg_color.getCheckedRadioButtonId()) {
    case R.id.color_0: color = 0;
        break;
    case R.id.color_1: color = 1;
        break;
    case R.id.color_2: color = 2;
        break;
    case R.id.color_3: color = 3;
        break;
    case R.id.color_4: color = 4;
        break;
    case R.id.color_5: color = 5;
        break;
    case R.id.color_6: color = 6;
        break;
    case R.id.color_7: color = 7;
        break;
    default: color = -1;
    break;
}
//get days data
mon = tue = wed = thu = fri = sat = sun = false;
ArrayList<Integer> DaysId = getCheckedDays();
for(Integer chosenDay: DaysId) {
    switch (chosenDay) {
        case 0: mon = true;
                break;
        case 1: tue = true;
            break;
        case 2: wed = true;
            break;
        case 3: thu = true;
            break;
        case 4: fri = true;
            break;
        case 5: sat = true;
            break;
        case 6: sun = true;
            break;
    }
}
//get notification data
notification = swt_ntf.isChecked();
```

```
        //pass data to data base helper
        habitDatabaseHelper.addHabit(name, status, goal, mon, tue, wed, thu, fri, sat, sun
                , group, color, notification);
    }
```

▹일단 database helper 클래스에 table에 data를 추가하는 메서드를 만든다.
SQLiteDATAbase 객체를 만들고 this.getWritableDatabase값으로 초기화해서 이 객체가 값을 쓰는데 이용되는 것을 알려준다.

```
 void addHabit(String name, Boolean status, Integer goal, Boolean mon, Boolean tue,
Boolean wed, Boolean thu, Boolean fri
            , Boolean sat, Boolean sun, String group, int color, Boolean notification) {
        SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();
```

▹값을 넣어주기 전에 ContentValues 객체를 만들어 그 안에 차곡차곡 값들을 넣어준다.

```
//use content value objet to pass data groups that we want to save
        ContentValues contentValues = new ContentValues();
        contentValues.put(COLUMN_NAME, name);
        contentValues.put(COLUMN_STATUS, status);
        contentValues.put(COLUMN_GOAL, goal);
        contentValues.put(COLUMN_MON, mon);
        contentValues.put(COLUMN_TUE, tue);
        contentValues.put(COLUMN_WED, wed);
        contentValues.put(COLUMN_THU, thu);
        contentValues.put(COLUMN_FRI, fri);
        contentValues.put(COLUMN_SAT, sat);
        contentValues.put(COLUMN_SUN, sun);
        contentValues.put(COLUMN_GROUP, group);
        contentValues.put(COLUMN_COLOR, color);
        contentValues.put(COLUMN_NOTIFY, notification);
```

▹마지막으로 SQLiteDATAbase 객체에 contentvaluse 객체를 전달에 한번에 값을 입력한다. 그리고 입력이 안되었을 때 -1을 return하므로 한번 확인해주는 코드를 넣는다.

```
//pass content value to sql database object
        long result = sqLiteDatabase.insert(TABLE_NAME, null, contentValues);

        if(result == -1){
            Log.d(TAG, "addHabit: write data failed");
        }
```

▹dialog fragment에 db에 정보를 전달하는 메서드를 만든다. 그리고 그 안에서 databasehelper클래스의 객체를 만들어 준다.

```
 private void saveNewHabitData() {
        HabitDatabaseHelper habitDatabaseHelper = new HabitDatabaseHelper(getActivity());
```

▹값을 임시로 저장하기 위한 필드들을 만들어 준다

```
        String name, group ;
        int goal, color;
        Boolean mon, tue, wed, thu, fri, sat, sun, status, notification;
```

▹edit text에 입력 된 값을 가져온다

```
//get edit text data
name = et_name.getText().toString().trim();
goal = Integer.parseInt(et_name.getText().toString().trim());
group = et_group.getText().toString().trim();
```

▹두 개의 radiobutton중에서 어떤 버튼이 눌려있는지 확인을 위해 .getCheckedRadioButton()메서드를 사용하고 if문을 이용해 boolean 변수에 true, false값을 초기화 한다.

```
//get build quit data (build == true, quit == false)
if(rg_status.getCheckedRadioButtonId() == R.id.radio_build) {
   status = true;
}else {
   status = false;
}
```

▹여러 개의 색을 결정하는 radiobutton중에서 어떤 버튼이 눌려있는지 확인을 위해 .getCheckedRadioButton()메서드를 사용하고switch문을 이용해 int 변수에 0부터 값을 입력한다.

```
//get color data
switch(rg_color.getCheckedRadioButtonId()) {
   case R.id.color_0: color = 0;
      break;
   case R.id.color_1: color = 1;
      break;
   case R.id.color_2: color = 2;
      break;
   case R.id.color_3: color = 3;
      break;
   case R.id.color_4: color = 4;
      break;
   case R.id.color_5: color = 5;
      break;
   case R.id.color_6: color = 6;
      break;
   case R.id.color_7: color = 7;
      break;
   default: color = -1;
   break;
}
```

▹checkbox에 있는 여러 버튼에 중복적으로 눌리는 버튼중 어떤 버튼이 눌렸는지 확인하기 위해 눌린 버튼을 ArrayList에 저장하고 for each문을 이용해 list안에 있는 모든 값을 확인하고 switch문을 중첩해 그 값에 맞는 요일에 true값을 저장한다

```
private ArrayList<Integer> getCheckedDays() {
   ArrayList<Integer> selectedDays = new ArrayList<>();
```

```
        CheckBox[] days = {cb_mon, cb_tue, cb_wed, cb_thu, cb_fri, cb_sat, cb_sun};
        for(int i = 0; i < days.length; i++) {
            if(days[i].isChecked()) {
                selectedDays.add(i);
            }
        }
        return selectedDays;
    }

//get days data
        mon = tue = wed = thu = fri = sat = sun = false;
        ArrayList<Integer> DaysId = getCheckedDays();
        for(Integer chosenDay: DaysId) {
            switch (chosenDay) {
                case 0: mon = true;
                        break;
                case 1: tue = true;
                    break;
                case 2: wed = true;
                    break;
                case 3: thu = true;
                    break;
                case 4: fri = true;
                    break;
                case 5: sat = true;
                    break;
                case 6: sun = true;
                    break;
            }
        }
```

▹switch button이 눌렸는지 확인을 위해 .ischeck()메서드를 사용해 boolean값을 저장한다.

```
//get notification data
        notification = swt_ntf.isChecked();
```

▹마지막으로 아까 만든 data base helper 객체 안에 있는 addHabit()메서드를 가져와서 그 안에 입력된 data들을 전달한다.

```
//pass data to data base helper
        habitDatabaseHelper.addHabit(name, status, goal, mon, tue, wed, thu, fri, sat, sun
                , group, color, notification);
```

◇ <루틴앱> SQL database에 저장한 정보를 불러오기

https://www.youtube.com/watch?v=VQKq9RHMS_0&list=PLSrm9z4zp4mGK0g_0_jxYGgg3os9tqRUQ&index=3

```
//HabitDataBaseHelper
```

```
    private static final String READ_TODAY_QUERY = "select name, build_or_quit, goal,
h_group, color, count from tb_habit";


Cursor readTodayData() {
        SQLiteDatabase sqLiteDatabase = this.getReadableDatabase();
        Cursor cursor = null;
        String todayQuery = "";
        Calendar calendar = Calendar.getInstance();
        int day = calendar.get(Calendar.DAY_OF_WEEK);
        switch (day) {
            case Calendar.MONDAY:
                todayQuery = " where mon=true;";
                break;
            case Calendar.TUESDAY:
                todayQuery = " where tue=true;";
                break;
            case Calendar.WEDNESDAY:
                todayQuery = "mon";
                break;
            case Calendar.THURSDAY:
                todayQuery = "mon";
                break;
            case Calendar.FRIDAY:
                todayQuery = "mon";
                break;
            case Calendar.SATURDAY:
                todayQuery = "mon";
                break;
            case Calendar.SUNDAY:
                todayQuery = "mon";
                break;
            default:
                Log.d(TAG, "readTodayData: error at reading day of week");
                break;
        }
        if(sqLiteDatabase != null) {
            cursor = sqLiteDatabase.rawQuery(READ_TODAY_QUERY + todayQuery, null);
        }

        return cursor;
    }

//TodayFragement
void storeDataInArray() {
        Cursor cursor = habitDatabaseHelper.readTodayData();
        if(cursor.getCount() == 0) {
            Log.d(TAG, "storeDataInArray: cursor with no data");
```

```
        }else{
            while(cursor.moveToNext()) {
                //add status, goal, h_group to make detail text
                String detail = "";
                if(cursor.getInt(1) == 1) {
                    detail = "<" + cursor.getString(3) + "> lets start " + cursor.getString(0)
                            + " for " + cursor.getInt(2) + "days";
                } else {
                    detail = "<" + cursor.getString(3) + "> lets quit " + cursor.getString(0)
                            + " for " + cursor.getInt(2) + "days";
                }

                habit_name.add(cursor.getString(0));
                habit_detail.add(detail);
                habit_count.add(cursor.getInt(5));
                habit_color.add(cursor.getInt(4));
            }
        }
    }

    void checkDataInArray() {
        for(String name: habit_name) {
            Log.d(TAG, "storeDataInArray: "+name);
        }
        for(String detail: habit_detail) {
            Log.d(TAG, "checkDataInArray: "+detail);
        }
        for(int count: habit_count) {
            Log.d(TAG, "checkDataInArray: "+Integer.toString(count));
        }
        for(int color: habit_color) {
            Log.d(TAG, "checkDataInArray: "+Integer.toString(color));
        }
    }
```

▹일단 databasehelper 클래스에 data를 read해서 cursor를 return하는 메서드를 만든다. SQLiteDatabase 객페를 만들고 this.getReadableDatabase()메서드를 이용해 읽는 동작을 할 것이라고 초기화해준다.

```
Cursor readTodayData() {
        SQLiteDatabase sqLiteDatabase = this.getReadableDatabase();
        Cursor cursor = null;
}
```

◇ 오늘 날짜를 확인해 query를 작성해야 하므로 string 변수를 만들고 switch를 이용해서 mon~sun 중에 어떤 값이 true인지에 따라 다른 string을 작성한다.

```
String todayQuery = "";
        Calendar calendar = Calendar.getInstance();
```

```
        int day = calendar.get(Calendar.DAY_OF_WEEK);
        switch (day) {
            case Calendar.MONDAY:
                todayQuery = " where mon=true;";
                break;
            case Calendar.TUESDAY:
                todayQuery = " where tue=true;";
                break;
            case Calendar.WEDNESDAY:
                todayQuery = "mon";
                break;
            case Calendar.THURSDAY:
                todayQuery = "mon";
                break;
            case Calendar.FRIDAY:
                todayQuery = "mon";
                break;
            case Calendar.SATURDAY:
                todayQuery = "mon";
                break;
            case Calendar.SUNDAY:
                todayQuery = "mon";
                break;
            default:
                Log.d(TAG, "readTodayData: error at reading day of week");
                break;
        }
```

◇ SqliteDatabase 객체에 저장되어 있는 data가 있을 때 .rawQuery()에 read를 위한 query를 전달해서 cursor를 return받는다. 이 때 sql query는 select column1, column2, column3, … from table_name where day = todayday를 사용했다.

```
// private static final String READ_TODAY_QUERY = "select name, build_or_quit, goal,
h_group, color, count from tb_habit";
// todayQuery = " where tue=true;";

if(sqLiteDatabase != null) {
            cursor = sqLiteDatabase.rawQuery(READ_TODAY_QUERY + todayQuery, null);
        }

return cursor;
```

◇data를 받아서 recycler view를 만들어야 하는 fragment에 list변수를 만들어준다.

```
ArrayList<String> habit_name, habit_detail;
    ArrayList<Integer> habit_count, habit_color;
 habit_name = new ArrayList<>();
        habit_detail = new ArrayList<>();
        habit_count = new ArrayList<>();
```

```
habit_color = new ArrayList<>();
```

◇database에 있는 data를 읽어 만들어 준 list 변수에 저장하는 메서드를 만든다. 그리고 그 안에서 databasehelper클래스 안에 있는 readTodayData()메서드를 불러와 Cursor 변수에 저장한다.

```
void storeDataInArray() {
    Cursor cursor = habitDatabaseHelper.readTodayData();
```

◇cursor에 data가 있는지 확인하고 있으면 while(cursor.moveToNext)로 커서를 계속 다음으로 움직이면서 data를 가져온다. data를 가져올 때 cursor.getString(i), cursor.getInt(i)메서드를 사용하면 한 row안의 i번째 column의 값을 가져온다. 그리고 그 값을 ArrayList.add()메서드를 이용해 List안에 넣는다.

```
 Cursor cursor = habitDatabaseHelper.readTodayData();
    if(cursor.getCount() == 0) {
       Log.d(TAG, "storeDataInArray: cursor with no data");
    }else{
       while(cursor.moveToNext()) {
          //add status, goal, h_group to make detail text
          String detail = "";
          if(cursor.getInt(1) == 1) {
             detail = "<" + cursor.getString(3) + "> lets start " + cursor.getString(0)
                   + " for " + cursor.getInt(2) + "days";
          } else {
             detail = "<" + cursor.getString(3) + "> lets quit " + cursor.getString(0)
                   + " for " + cursor.getInt(2) + "days";
          }

          habit_name.add(cursor.getString(0));
          habit_detail.add(detail);
          habit_count.add(cursor.getInt(5));
          habit_color.add(cursor.getInt(4));
       }
    }
```

◇ Arraylist안에 값이 잘 전달되었는지 확인해준다.

```
void checkDataInArray() {
    for(String name: habit_name) {
       Log.d(TAG, "storeDataInArray: "+name);
    }
    for(String detail: habit_detail) {
       Log.d(TAG, "checkDataInArray: "+detail);
    }
    for(int count: habit_count) {
       Log.d(TAG, "checkDataInArray: "+Integer.toString(count));
    }
    for(int color: habit_color) {
       Log.d(TAG, "checkDataInArray: "+Integer.toString(color));
    }
```

```
    }
```

◇ <루틴앱> SQL database에 저장한 정보로 RecyclerView 만들기

https://www.youtube.com/watch?v=VQKq9RHMS_0&list=PLSrm9z4zp4mGK0g_0_jxYGgg3os9tqRUQ&index=3

```
//Adapter
public class TodayAdapter extends RecyclerView.Adapter<TodayAdapter.TodayViewHolder>
{
    //TAG
    private static final String TAG = "TodayAdapter";

    //var
    private Context context;
    private ArrayList<String> habit_name, habit_detail;
    private ArrayList<Integer> habit_count, habit_color;

    public TodayAdapter(Context context, ArrayList<String> habit_name, ArrayList<String>
habit_detail
            , ArrayList<Integer> habit_count, ArrayList<Integer> habit_color) {
        this.context = context;
        this.habit_name = habit_name;
        this.habit_detail = habit_detail;
        this.habit_count = habit_count;
        this.habit_color = habit_color;
    }

    @NonNull
    @Override
    public TodayAdapter.TodayViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
        LayoutInflater inflater = LayoutInflater.from(context);
        View view = inflater.inflate(R.layout.cardview_habit, parent, false);
        return new TodayViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull TodayAdapter.TodayViewHolder holder, int
position) {
        holder.cardView_bg.setBackgroundColor(habit_color.get(position));
        holder.tv_name.setText(habit_name.get(position));
        holder.tv_detail.setText(habit_detail.get(position));
//        holder.tv_count.setText(habit_count.get(position));
    }
```

```
    @Override
    public int getItemCount() {
        return habit_name.size();
    }

    public class TodayViewHolder extends RecyclerView.ViewHolder{
        ConstraintLayout cardView_bg;
        TextView tv_name, tv_detail, tv_count;

        public TodayViewHolder(@NonNull View itemView) {
            super(itemView);
            cardView_bg = itemView.findViewById(R.id.habit_layout);
            tv_name = itemView.findViewById(R.id.habit_name);
            tv_detail = itemView.findViewById(R.id.habit_detail);
            tv_count = itemView.findViewById(R.id.habit_count);

        }
    }
}

//fragment with recyclerview
    HabitDatabaseHelper habitDatabaseHelper;
    ArrayList<String> habit_name, habit_detail;
    ArrayList<Integer> habit_count, habit_color;
    TodayAdapter todayAdapter;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        // Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_today, container, false);
        recyclerView = view.findViewById(R.id.recycler);

        //getting data from db to bind it to a adapter
        habitDatabaseHelper = new HabitDatabaseHelper(getActivity());
        habit_name = new ArrayList<>();
        habit_detail = new ArrayList<>();
        habit_count = new ArrayList<>();
        habit_color = new ArrayList<>();
        storeDataInArray();
        checkDataInArray();

        //initialize todayAdapter and set Adapter for recycler view
        todayAdapter = new TodayAdapter(getActivity(), habit_name, habit_detail, habit_count,
habit_color);
        recyclerView.setAdapter(todayAdapter);
        recyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));
```

```
        return view;
    }
```

▹Adapter class를 만들어서 RecyclerView.Adapter<TodayAdapter.TodayViewHolder>를 상속하게 한다.
그리고 그 안에 상속자를 만들어주고 상속자에 context와 원하는 data를 입력하게 한다.
그리고 필요한 필수 메서드를 오버라이드한다.

```
public class TodayAdapter extends RecyclerView.Adapter<TodayAdapter.TodayViewHolder>
{

    public TodayAdapter(Context context, ArrayList<String> habit_name, ArrayList<String>
habit_detail
            , ArrayList<Integer> habit_count, ArrayList<Integer> habit_color) {

    }

    @NonNull
    @Override
    public TodayAdapter.TodayViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
        }

    @Override
    public void onBindViewHolder(@NonNull TodayAdapter.TodayViewHolder holder, int
position) {
            }

    @Override
    public int getItemCount() {
            }
```

▹그리고 Adapter 클래스 안에 ViewHolder 클래스를 만들어서 RecyclerView.ViewHolder를 상속하게 한다.

```
public class TodayViewHolder extends RecyclerView.ViewHolder{
        public TodayViewHolder(@NonNull View itemView) {
            super(itemView);
            }
    }
```

▹준비가 끝났으면 첫번째로 생성자에서 입력받은 arraylist들을 필드변수 안에 저장한다

```
  //var
    private Context context;
    private ArrayList<String> habit_name, habit_detail;
    private ArrayList<Integer> habit_count, habit_color;

   public TodayAdapter(Context context, ArrayList<String> habit_name, ArrayList<String>
habit_detail
            , ArrayList<Integer> habit_count, ArrayList<Integer> habit_color) {
```

```
        this.context = context;
        this.habit_name = habit_name;
        this.habit_detail = habit_detail;
        this.habit_count = habit_count;
        this.habit_color = habit_color;
    }
```

▹다음으로 onCreateViewHolder 메서드 안에서는 card view xml파일을 inflate하고 반환값으로 viewHolder객체를 만들고 그 안에 inflate한 view를 전달한다.

```
@NonNull
    @Override
    public TodayAdapter.TodayViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
        LayoutInflater inflater = LayoutInflater.from(context);
        View view = inflater.inflate(R.layout.cardview_habit, parent, false);
        return new TodayViewHolder(view);
    }
```

▹viewHolder 클래스로 가서 그 안에서 cardView안에 있는 widget과 객체들을 연결해준다.

```
 public class TodayViewHolder extends RecyclerView.ViewHolder{
        ConstraintLayout cardView_bg;
        TextView tv_name, tv_detail, tv_count;

        public TodayViewHolder(@NonNull View itemView) {
            super(itemView);
            cardView_bg = itemView.findViewById(R.id.habit_layout);
            tv_name = itemView.findViewById(R.id.habit_name);
            tv_detail = itemView.findViewById(R.id.habit_detail);
            tv_count = itemView.findViewById(R.id.habit_count);

        }
    }
```

▹xml파일과 연결되어 있는 객체들은 onBindViewHolder메서드에서 data들과 binding해준다.

```
 @Override
    public void onBindViewHolder(@NonNull TodayAdapter.TodayViewHolder holder, int
position) {
        holder.cardView_bg.setBackgroundColor(habit_color.get(position));
        holder.tv_name.setText(habit_name.get(position));
        holder.tv_detail.setText(habit_detail.get(position));
//        holder.tv_count.setText(habit_count.get(position));
    }
```

▹마지막으로 getItemCount메서드에 전체 recyclerView의 크기를 전달해준다.

```
@Override
    public int getItemCount() {
        return habit_name.size();
    }
```

▹recyclerview를 보여주고 싶은 클래스로 가서 recyclerView객체를 xml파일과 연결해주고 Adapter객체를 만들어 생성자에 필요한 context와 data를 포함하는 arraylist들을 전달한다.

```
ArrayList<String> habit_name, habit_detail;
    ArrayList<Integer> habit_count, habit_color;
 habit_name = new ArrayList<>();
        habit_detail = new ArrayList<>();
        habit_count = new ArrayList<>();
        habit_color = new ArrayList<>();
        storeDataInArray();

//initialize todayAdapter and set Adapter for recycler view
RecyclerView recyclerView = view.findViewById(R.id.recycler);
TodayAdapter todayAdapter = new TodayAdapter(getActivity(), habit_name, habit_detail,
habit_count, habit_color);
```

▹recyclerView에 .setAdapter()메서드와 인자로 adapter객체를 전달해 recycler에 내가 만든 adapter를 연결한다. 그리고 .setLayoutManager()메서드와 원하는 layout를 인자로 전달한다.

```
recyclerView.setAdapter(todayAdapter);
        recyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));
```

◇ Color값 저장과 background색 바꾸기

▹값을 저장 할때 Color.parseColor(“#xxxxxx”)를 이용해서 색을 int형태로 저장한다.

```
 //get color data
        switch(rg_color.getCheckedRadioButtonId()) {
            case R.id.color_0: color = Color.parseColor("#F44336");
                break;
            case R.id.color_1: color = Color.parseColor("#E91E63");
                break;
            case R.id.color_2: color = Color.parseColor("#9C27B0");
                break;
            case R.id.color_3: color = Color.parseColor("#3F51B5");
                break;
            case R.id.color_4: color = Color.parseColor("#03A9F4");
                break;
            case R.id.color_5: color = Color.parseColor("#009688");
                break;
            case R.id.color_6: color = Color.parseColor("#8BC34A");
                break;
            case R.id.color_7: color = Color.parseColor("#FFC107");
                break;
            default: color = -1;
                Log.d(TAG, "saveNewHabitData: color data saving failed");
            break;
```

}

▹layout이나 widget의 background color를 바꾸기 위해서는 .setBackgroundColor()메서드와 그 안에 ColorInt로 아까 저장한 parseInt값을 전달해서 바꿀 수 있다.

```
cardView_bg.setBackgroundColor(habit_color.get(position));
```

◇ transaction replace와 add차이

https://stackoverflow.com/questions/18634207/difference-between-add-replace-and-addtobackstack

add()는 container에 이미 있는 fragment 위에 새로운 fragment를 추가한다.
replace()는 container에 있는 fragment를 삭제하고 새로운 fragment를 추가한다.

◇4 Ways to Get Current Date in Android

https://www.thecrazyprogrammer.com/2016/10/get-current-date-in-android.html

1. format을 만든 다음에 그 format에 맞게 string값을 만드는 것이 가장 보편적이다.

```
SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");
Calendar c = Calendar.getInstance();
String date = sdf.format(c.getTime());
```

2. 오늘의 년도, 주, 달, 일, 요일을 알고 싶으면 .get()메서드를 이용하고 원하는 형태를 고르면 된다.

```
Calendar c = Calendar.getInstance();
int day = c.get(Calendar.DAY_OF_MONTH);
int month = c.get(Calendar.MONTH);
int year = c.get(Calendar.YEAR);
int day = c.get(Calendar.DAY_OF_WEEK);
String date = day + "/" + (month+1) + "/" + year;
```

◇Calendar.DAY_OF_WEEK

https://developer.android.com/reference/java/util/Calendar#DAY_OF_WEEK

오늘의 요일을 알고 싶으면 Calendar 객체를 만들고 Calendar.getInstance()를 이용해 오늘의 시간, 날짜에 대한 모든 정보를 불러오고 .get(Calendar.DAY_OF_WEEK)메서드를 이용해 요일을 int형을 가져올 수 있다.

```
Calendar c = Calendar.getInstance();
int day = c.get(Calendar.DAY_OF_WEEK)
```

이 때 요일은 일요일부터 토요일까지 1~7의 정수로 반환된다.

◇Clickable recycler view

recycler view의 각 element를 clickable하게 만드려면 Adapter에서 onBinderViewHolder메서드 안에 cardview layout을 가져온 객체에 setOnClickListener를 설정하면 된다.
일단 ViewHolder에서 view와 layout객체를 연결힌다

```
public class TodayViewHolder extends RecyclerView.ViewHolder{
    ConstraintLayout cardView_bg;

    public TodayViewHolder(@NonNull View itemView) {
        super(itemView);
        cardView_bg = itemView.findViewById(R.id.habit_layout);
        }
  }
```

그리고 onBindViewHolder메서드 안에서 setOnClickListener를 설정한다.

```
@Override
  public void onBindViewHolder(@NonNull TodayAdapter.TodayViewHolder holder, int
position) {
    int temp = position;
    ColorInt currentRecyclerBackgroundColor;

    holder.cardView_bg.setBackgroundColor(Color.parseColor("#808080"));
        holder.cardView_bg.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            //change color and check today status to true
            holder.cardView_bg.setBackgroundColor(habit_color.get(temp));
             }
    });
  }
```

◇Update column data in SQL

```
void updateCount(String name, int count) {
    SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();
    ContentValues contentValues = new ContentValues();
    contentValues.put(COLUMN_COUNT, count);

    long result = sqLiteDatabase.update(TABLE_HABIT, contentValues, "name=?", new
String[]{name});
    if(result == -1){
        Log.d(TAG, "addHabit: update data failed");
    }
```

```
    }
```

SQLiteDatabase 객체를 만들고 this.getWritableDatabase()로 쓰기 모드로 설정한다.

```
        SQLiteDatabase sqLiteDatabase = this.getWritableDatabase();
```

정보를 그냥 쓸 때와 동일하게 ContentValues 객체를 만들고 이 안에 원하는 data를 일단 저장한다.

```
ContentValues contentValues = new ContentValues();
        contentValues.put(COLUMN_COUNT, count);
```

SQLiteDatabase 객체에 .update()메서드를 이용하여 정보를 update한다. 이때 인자로는 최신화하고 싶은 table, contentValues, 바꾸고 싶은 column를 표시해 줄 SQL문법, column에 있는 바꾸고 싶은 row의 값 순서로 넣는다.

```
 long result = sqLiteDatabase.update(TABLE_HABIT, contentValues, "name=?", new
String[]{name});
        if(result == -1){
            Log.d(TAG, "addHabit: update data failed");
        }
```

◇List에 찾는 element가 존재하는지 확인하는 법

https://www.baeldung.com/find-list-element-java

◇update table in a database only once a day

```
 void initializeDataDatabase() {
        Calendar calendar = Calendar.getInstance();
        int currentDay = calendar.get(Calendar.DAY_OF_MONTH);
        SharedPreferences setting = getActivity().getSharedPreferences("PREFS", 0);
        int lastDay = setting.getInt("day", 0);

        if(lastDay != currentDay) {
            SharedPreferences.Editor editor = setting.edit();
            editor.putInt("day", currentDay);
            editor.commit();
            //write date data once a day
            HabitDatabaseHelper habitDatabaseHelper = new
HabitDatabaseHelper(getActivity());
            for(String habit: habit_name) {
                habitDatabaseHelper.addTodayHabit(habit);
            }
        }
    }
```

▹ SharedPreferences
https://developer.android.com/reference/android/content/SharedPreferences
Context.getSharedPreferences에서 반환된 data들에 접근하고 조작하는 인터페이스이다. 어떠한 preference든 이 클래스를 조금이라도 포함되어 모든 client들이 이 클래스를 포함하게 된다. preference의 조작은 반드시 Editor를 통해서 이루어져야 하고 마지막에 commit()을 해야 한다.
preference 안에 있는 data들은 get메서드를 이용해서 추출가능하다.

https://developer.android.com/training/data-storage#pref
안드로이드에서 제공하는 app data 저장 방법은 4가지가 있다.

1. App-specific storage: 이 앱에서만 사용하는 data를 저장하는 방법으로 민감한 data는 internal storage 공간에 아니면 external storage 공간에 저장할 수 있다.
2. Shared storage: 이 앱이 다른 앱들과 공유하고 싶은 data들을 저장하는 곳으로 media, documents, file들을 주로 저장한다.
3. Preferences: Store private, primitive data in key-value pairs.
4. Databases: Store structured data in a private database using the Room persistence library.

◇sqlite query with multiple ? where clause

https://stackoverflow.com/questions/9185441/sqlite-query-method-where-clause-is-only-taking-double-quotes-strings

◇recycler view swip action 추가하기

```
 public View onCreateView(LayoutInflater inflater, ViewGroup container,
                          Bundle savedInstanceState) {
// Inflate the layout for this fragment
        View view = inflater.inflate(R.layout.fragment_today, container, false);
        recyclerView = view.findViewById(R.id.recycler);


//set swipe action
        ItemTouchHelper itemTouchHelper = new ItemTouchHelper(cardViewSwipeCallback);
        itemTouchHelper.attachToRecyclerView(recyclerView);
        return view;
}

ItemTouchHelper.SimpleCallback cardViewSwipeCallback = new
ItemTouchHelper.SimpleCallback(0, ItemTouchHelper.LEFT) {
        @Override
        public boolean onMove(@NonNull RecyclerView recyclerView, @NonNull
RecyclerView.ViewHolder viewHolder, @NonNull RecyclerView.ViewHolder target) {
            return false;
```

```
    }

    @Override
    public void onSwiped(@NonNull RecyclerView.ViewHolder viewHolder, int direction) {
        if(direction == ItemTouchHelper.LEFT) {
            //get the row of element in a recycler view with is swiped
            int position = viewHolder.getAdapterPosition();
            String name = habit_name.get(position);
            //go to the sub class with selected element data
            MainActivity mainActivity = (MainActivity) getActivity();
            mainActivity.moveToHabitSpecific(name);
        } else {
            Log.d(TAG, "onSwiped: swipe direction error");
        }
    }
};
```

▹ItemTouchHelper.SimpleCallback 필드를 만들고 인자로 drag방향, swipe방향을 입력한다. 나의 경우는 draw하지 않으므로 0을 전달하고 swipe 방향은 left만 전달했다. 그리고 메서드 두개를 오버라이드 한다.

```
ItemTouchHelper.SimpleCallback cardViewSwipeCallback = new
ItemTouchHelper.SimpleCallback(0, ItemTouchHelper.LEFT) {
    @Override
    public boolean onMove(@NonNull RecyclerView recyclerView, @NonNull
RecyclerView.ViewHolder viewHolder, @NonNull RecyclerView.ViewHolder target) {
        return false;
    }

    @Override
    public void onSwiped(@NonNull RecyclerView.ViewHolder viewHolder, int direction) {
     };
```

▹swipe 했을 때 취할 동작은 onSwiped()메서드 안에 작성하면 된다.

```
  @Override
    public void onSwiped(@NonNull RecyclerView.ViewHolder viewHolder, int direction) {
        if(direction == ItemTouchHelper.LEFT) {
            //get the row of element in a recycler view with is swiped
            int position = viewHolder.getAdapterPosition();
            String name = habit_name.get(position);
            //go to the sub class with selected element data
            MainActivity mainActivity = (MainActivity) getActivity();
            mainActivity.moveToHabitSpecific(name);
        } else {
            Log.d(TAG, "onSwiped: swipe direction error");
        }
    }
```

◇ SharedPreference - Custom dialog에 입력되는 값 저장하고 다시 돌아왔을 때 저장한 설정값 보여주기

https://www.youtube.com/watch?v=jiD2fxn8iKA
shared preference는 아주 간단한 key-value 형태의 값을 저장하고 불러오기 쉽고 앱에 저장이 되어 있어서 앱을 종료한 다음 다시 열어도 정보는 유지된다.

```
      //show current setting data
      SharedPreferences sharedPreferences =
getActivity().getSharedPreferences("SettingPrefs", Context.MODE_PRIVATE);
       //read data from shared preference and initialize widgets
       readSettingData();

       btn_save.setOnClickListener(new View.OnClickListener() {
          @Override
          public void onClick(View view) {
             //save setting change
             saveSettingData();
           }
       });


 public void readSettingData() {
       int sort = sharedPreferences.getInt("sort", -1);
       Boolean mode = sharedPreferences.getBoolean("mode", false);
       Boolean notification = sharedPreferences.getBoolean("notification", false);

       rg_sort.check(sort);
       swt_mode.setChecked(mode);
       swt_ntf.setChecked(notification);
   }

   public void saveSettingData() {
       SharedPreferences.Editor editor = sharedPreferences.edit();
       int sort = rg_sort.getCheckedRadioButtonId();
       Boolean mode = swt_mode.isChecked();
       Boolean notification = swt_ntf.isChecked();
       editor.putInt("sort", sort);
       editor.putBoolean("mode", mode);
       editor.putBoolean("notification", notification);
       editor.commit();

       checkSettingData(sort, mode, notification);
   }

   private void checkSettingData(int sort, Boolean mode, Boolean ntf) {
       Log.d(TAG, "checkSettingData(sort): " + sort);
```

```
        Log.d(TAG, "checkSettingData(mode): " + mode);
        Log.d(TAG, "checkSettingData(ntf): " + ntf);
    }
```

▹SharedPreference 객체를 만들고 .getSharedPreferences를 이용해 초기화해 준다.
getSharedPreferences에 전달할 인자인 name, mode에 따라 그 name과 mode의 파일이 생성되고 그 파일에 원하는 정보가 저장된다.

```
 //show current setting data
       SharedPreferences sharedPreferences =
getActivity().getSharedPreferences("SettingPrefs", Context.MODE_PRIVATE);
```

▹값을 저장하기 위해서는 SharedPreference.Editor 객체를 가져와서 원하는 파일의 정보를 가지고 있는 SharedPreferences 객체 sharedPreference.edt()메서드로 edit모드로 초기화해준다.
그리고 editor 객체 .putString, .putInt, .putBoolean 메서드를 이용해 key 값과 함께 저장해준다.
마지막으로 editor.commit()을 해준다.

```
   SharedPreferences sharedPreferences =
getActivity().getSharedPreferences("SettingPrefs", Context.MODE_PRIVATE);

SharedPreferences.Editor editor = sharedPreferences.edit();
            editor.putInt("sort", sort);
        editor.putBoolean("mode", mode);
        editor.putBoolean("notification", notification);
        editor.commit();
```

▹값을 읽을 때는 원하는 파일의 정보를 가지고 있는 SharedPreferences 객체를 만들어 .getString, .getInt, .getBoolean 메서드를 이용해 key값으로 원하는 data를 가져오면 된다.

```
       SharedPreferences sharedPreferences =
getActivity().getSharedPreferences("SettingPrefs", Context.MODE_PRIVATE);
 Boolean mode = sharedPreferences.getBoolean("mode", false);
        Boolean notification = sharedPreferences.getBoolean("notification", false);
```

◇ Up Button

업 버튼은 한 activity에서 다른 activity로 넘어 갔을 때 parent activity로 돌아가기 위한 버튼이 actionbar에 생기는 버튼이다.

▹처음으로 두번째 activity에서 actionbar 객체를 만들고 getSupportActionBar()메서드를 이용해 초기화 해줘야 한다.

```
 Toolbar toolbar;
    ActionBar actionBar;
 //set toolbar
```

```
toolbar = findViewById(R.id.toolbar_sub);
setSupportActionBar(toolbar);
actionBar = getSupportActionBar();
```

▹actionbar 객체에 .setDisplayHomeAsUpEnable()메서드를 전달해 up버튼을 활성화하고 true를 전달해야 한다.

```
//set up button
actionBar.setDisplayHomeAsUpEnabled(true);
```

▹마지막으로 menifest file에 가서 두번째 activity의 parent activity를 돌아가고 싶은 activity로 설정해줘야 한다.

```
<activity
    android:name=".SubActivity"
    android:parentActivityName=".MainActivity"
    android:exported="true">
</activity>
```

◇ 뒤로가기 버튼 오버라이드

뒤로가기 버튼을 눌렀을 때 특정 동작을 하기 원하면 activity에서 onBackPressed()메서드를 오버라이드해서 super를 지우고 원하는 동작을 적으면 된다.

나는 뒤로 돌아갔을 때 recycler view가 refresh되도록 하기 위해 새 activity로 넘어가게 설정했다.

```
@Override
  public void onBackPressed() {
    Intent intent = new Intent(SubActivity.this, MainActivity.class);
    startActivity(intent);
  }
```

◇onClickListener feild for multiple button

switch문을 사용하고 view.getID()메서드를 이용해서 listener를 부르는 view의 id를 가져온다.

```
@Override
public void onClick(View v) {
  switch (v.getId()) {
    case R.id.oneButton:
      // do your code
      break;
    case R.id.twoButton:
      // do your code
      break;
    case R.id.threeButton:
      // do your code
```

```
            break;
        default:
            break;
    }
}
```

◇XML 주석

<!-- 주석 내용 -->

◇ Custom Calendar View using RecyclerView

https://www.youtube.com/watch?v=Ba0Q-cK1fJo&t=197s

▹recycler view안에 cardview 꽉 채우기
adapter를 만드는 과정은 똑같다.
다만 일반적인 recyclerview와는 다르게 가로 7줄 세로 6줄로 view가 꽉차야 하고 스크롤이 되면 안된다.
layoutmanager를 gridlayaout으로 하고 인자로 7을 전달하면 가로 7개는 성립된다.

```
 CalendarAdapter calendarAdapter = new CalendarAdapter(this, dayInMonth
                , todayYearMonth, color, name);
        RecyclerView.LayoutManager layoutManager = new
GridLayoutManager(getApplicationContext(), 7);
        rv_calendar.setAdapter(calendarAdapter);
        rv_calendar.setLayoutManager(layoutManager);
```

다만 세로에 cardview가 오직 6개 들어가게 하기 위해서 adapter onCreateViewHolder에서 cardview의 세로 크기를 제한해야 한다.

```
 public CalendarAdapter.CalendarViewHolder onCreateViewHolder(@NonNull ViewGroup
parent, int viewType) {
        LayoutInflater inflater = LayoutInflater.from(context);
        View view = inflater.inflate(R.layout.cardview_calendar, parent, false);
        ViewGroup.LayoutParams layoutParams = view.getLayoutParams();
        layoutParams.height = (int) (parent.getHeight() * 0.166666666);

        return new CalendarViewHolder(view);
    }
```

▹adapter에 전달할 daysInMonth ArrayList를 만든다

```
        private ArrayList<String> daysInMonthArray(LocalDate date) {
        ArrayList<String> daysInMonthArray = new ArrayList<>();
        YearMonth yearMonth = YearMonth.from(date);

        int daysInMonth = yearMonth.lengthOfMonth();
```

```
        LocalDate firstOfMonth = selectedDate.withDayOfMonth(1);
        int dayOfWeek = firstOfMonth.getDayOfWeek().getValue();

        for(int i = 1; i <= 42; i++){
            if(i <=dayOfWeek || i> daysInMonth + dayOfWeek) {
                daysInMonthArray.add("");
            } else {
                daysInMonthArray.add(String.valueOf((i - dayOfWeek)));
            }
        }
        return daysInMonthArray;
    }
```

결국 6행 7열에서 5월 1일 전의 빈칸들에는 “”를 저장하고 1일부터 30일까지는 1~30을 저장하고 나머지는 다시 “”을 저장해야 한다.

▸LocalDate값으로 현재 오늘의 year-month-date 정보를 전달 받는다

```
 private ArrayList<String> daysInMonthArray(LocalDate date) {
```

▸YearMonth 자료형 한에 오늘 날의 년도와 달만 저장한다.

```
YearMonth yearMonth = YearMonth.from(date);
```

▸달의 전체 일수를 저장하고

```
int daysInMonth = yearMonth.lengthOfMonth();
```

▸첫번째 날의 year-month-day를 저장한다

```
        LocalDate firstOfMonth = selectedDate.withDayOfMonth(1);
```

▸첫번째 날이 무슨 요일인지 확인해 앞에 빈칸이 몇개인지 저장한다

```
        int dayOfWeek = firstOfMonth.getDayOfWeek().getValue();
```

▸for문으로 1부터 42까지 가면서 dayofweek 이하일 때는 “”를 저장하고
daysInMonth + dayOfWeek 이상일 때도 “”를 저장한다.
그리고 그 외는 1부터 달의 마지막 날까지 더한다. 그리고 그 arraylist를 반환한다.

```
 for(int i = 1; i <= 42; i++){
            if(i <=dayOfWeek || i> daysInMonth + dayOfWeek) {
                daysInMonthArray.add("");
            } else {
                daysInMonthArray.add(String.valueOf((i - dayOfWeek)));
            }
        }
return daysInMonthArray;
```

▹daysInMonth ArrayList에 오늘 날짜 정보가 들어 있는 LocalDate를 전달하고 ArrayList를 만들어서 adapter에 전달한다.

```
selectedDate = LocalDate.now();
```

```
ArrayList<String> dayInMonth = daysInMonthArray(selectedDate);
CalendarAdapter calendarAdapter = new CalendarAdapter(this, dayInMonth
        , todayYearMonth, color, name);
```

알람 앱

◇Thread

https://developer.android.com/guide/components/processes-and-threads