

2021 졸업작품

교과목 포트폴리오



컴퓨터정보공학과 PA
20213953 김나연

● 목차

RES

프론트 엔드 구현

SOURCETREE

RESOURCE

정의 / 하부 폴더 / 사용방법

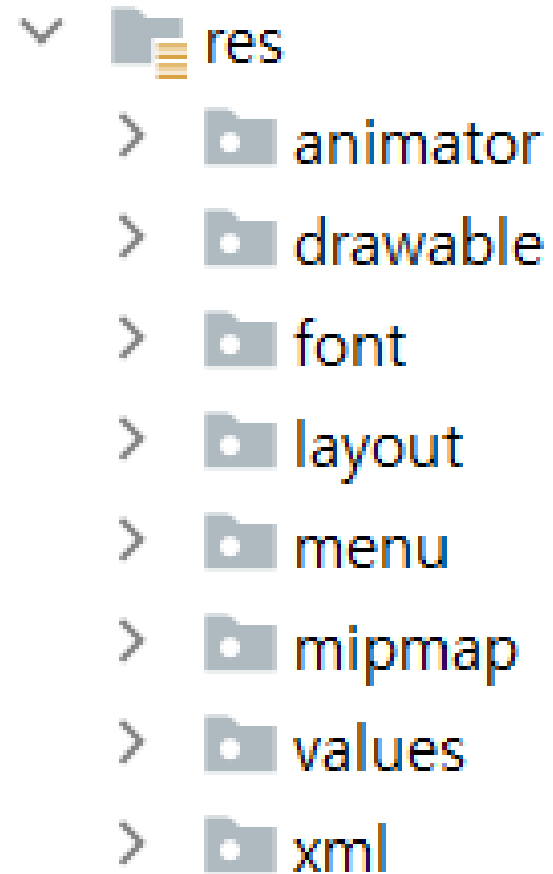
RES?

'resources'폴더, 화면에 표시/재생 되는 파일로 그림 파일,
사운드 파일, XML 파일, 폰트 파일 등으로 이루어짐

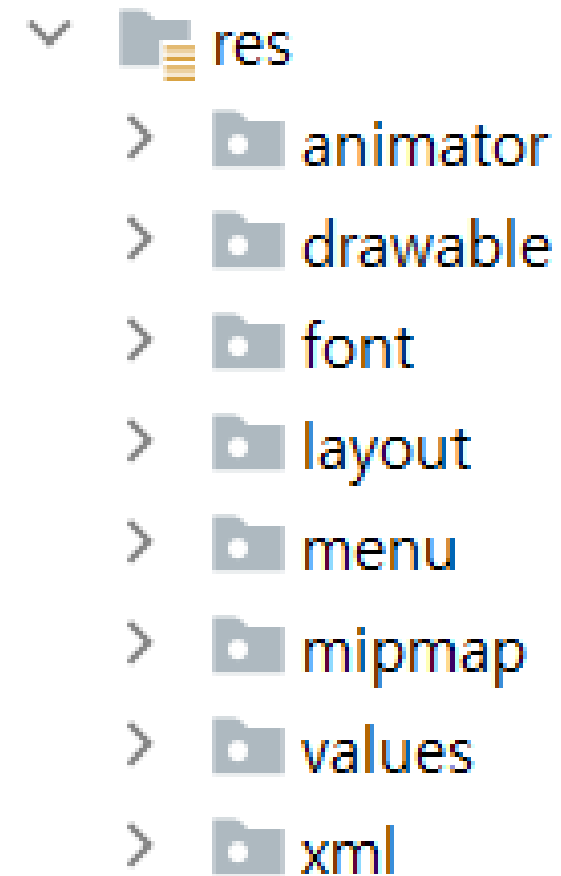
안드로이드 개발자 문서

(<https://developer.android.com/guide/topics/resources/providing-resources.html>)에서는

리소스, 앱 작동 코드와 분리 관리 강조



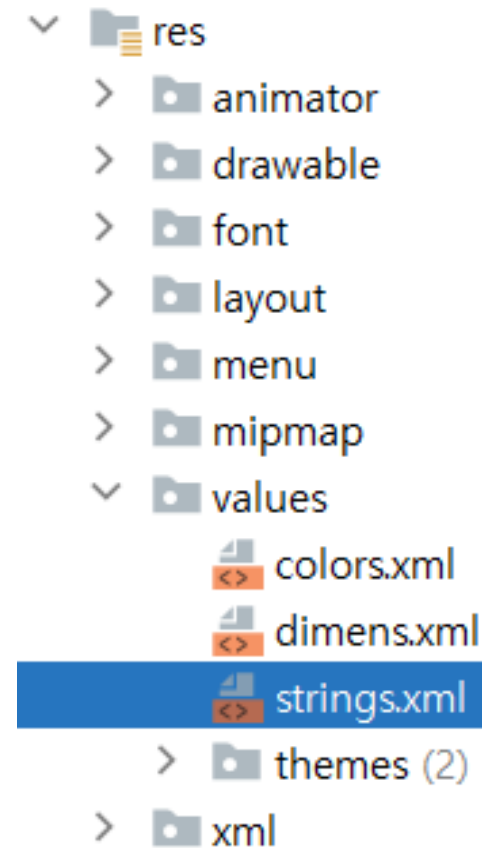
폴더이름	리소스
animator	'property animations' 설정용 XML
drawable	비트맵(png, jpg, gif 등) OR 그림으로 표시되는 XML
font	파일 확장자 ttf, otf, ttc OR <font-family>태크를 포함하는 XML
layout	사용자 인터페이스를 설정하는 XML
menu	우리 앱의 메뉴를 설정하는 XML
mipmap	해상도별로 구분된 앱 아이콘
values	문자열(string), 정수(constants), 색깔(colors) 값을 담고 있는 XML
xml	사용자가 선택한 옵션 등의 설정 내용을 담고 있는 XML



리소스 사용 방법

리소스: res/values/strings.xml
: `<string name="app_name">flowerParty</string>`

JAVA	XML
<code>getString(R.string.app_name)</code>	<code><TextView text="@string/app_name"/></code>



프론트엔드 구현

세부 페이지 ① ② ③ / 식물 일지 ① ②

커스텀 디자인

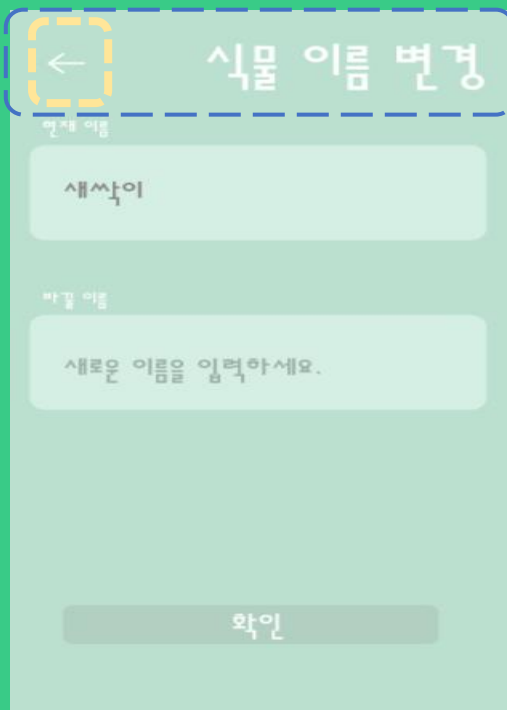


세부페이지

ClickListener, finish(), 커스텀 버튼,
현재시간, texcolor(hint), visibility

세부 페이지

(1)



상단

JAVA //뒤로가기

```
import android.view.View;
import android.widget.ImageView;

public class PlantsNicknameActivity extends AppCompatActivity {
    ImageView imgLArrow;

    imgLArrow = findViewById(R.id.leftArrow_nickname);
    imgLArrow.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            setResult(RESULT_FIRST_USER);
            finish();
        }
    });
}
```

//액티비티 종료

XML

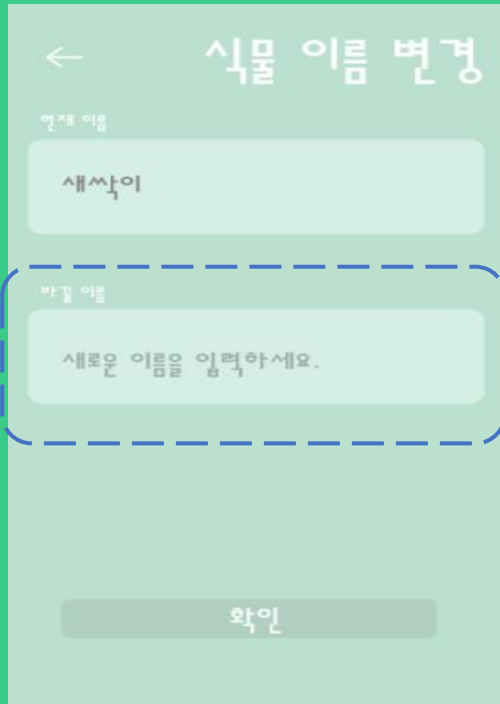
```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="70dp"
    android:orientation="horizontal">

    <ImageView
        android:id="@+id/leftArrow_nickname"
        android:layout_width="30dp"
        android:layout_height="30dp"
        android:layout_marginLeft="20dp"
        android:layout_gravity="center"
        app:srcCompat="@drawable/leftarrow" />

    <TextView
        android:id="@+id/textView16"
        android:layout_width="280dp"
        android:layout_height="wrap_content"
        android:layout_marginLeft="10dp"
        android:layout_marginTop="20dp"
        android:fontFamily="@font/ahn2006_m"
        android:gravity="right"
        android:text="식물 이름 변경"
        android:textColor="@color/white"
        android:textSize="50sp"
        android:textStyle="bold" />
</LinearLayout>
```

세부 페이지

(1)



TEXT 입력

XML

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginHorizontal="30dp"
    android:background="@drawable/shape"
    android:orientation="vertical"
    android:paddingVertical="40dp"
    android:paddingHorizontal="25dp"

    <EditText
        android:id="@+id/et_plantNickName"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="@null"
        android:ems="10"
        android:inputType="textPersonName"
        android:fontFamily="@font/ahn2006_m"
        android:textSize="30sp"
        android:textStyle="bold"
        android:textColor="@color/gray"
        android:textColorHint="@color/gray_green"
        android:hint="새로운 이름을 입력하세요." />
```

</LinearLayout>

• match_parent 와 wrap_content 차이

- **match_parent** : 부모(parent), 담고 있는 레이아웃 크기 맞춤
- **wrap_content** : 내용물(content), 내용물의 크기인 글씨 크기 맞춤

• inputType

textCapCharacters : 모든 문자를 대문자로 변환. 실제로는 Shift 키가 자동 설정된 형태로 동작
textAutoCorrect : 입력 텍스트 자동 교정
textAutoComplete : 입력 텍스트 자동 완성
textEmailAddress : e-mail 주소 형식의 텍스트 입력
textEmailSubject : e-mail 제목 형식의 텍스트 입력
textShortMessage : 단문 메시지(short message) 형식의 텍스트 입력
textLongMessage : 장문 메시지(long message) 형식의 텍스트 입력

textPersonName : 사람 이름 입력

textPostalAddress : 우편 번호 형식의 텍스트
textPassword : 패스워드 입력
textVisiblePassword : 보여져야 하는 패스워드
number : 숫자 형식
numberSigned : 부호있는 숫자 입력
numberDecima : 소수점 숫자 입력
numberPassword : 숫자로 구성된 패스워드 입력
phone : 전화번호 형식
datetime : 날짜 및 시간 형식
date : 날짜 형식
time : 시간 형식

• textColor와 textColorHint 차이

- **TextColor** : 입력 글자 색상
- **TextColorHint** : 예시 글자 색상
 - Hint : 예시 글자

상단 - JAVA

✓ OnClickListener 4가지 방법

1 익명클래스

click 이벤트가 어디서 처리되는지 쉽게 확인 가능, 코드가 간결하여 자주 사용

- 이벤트가 일어나는 view(Button 등)의 개수가 적거나,
- 이벤트가 일어나는 view 간의 연관성이 적은 경우
- 이벤트 핸들러 함수 내에서 익명 클래스 외부의 변수를 참조하지 않는 경우
- 간단한 view 클릭 이벤트 테스트 코드를 작성하는 경우

```
Button btn = (Button) findViewById(R.id.btn);

btn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // 동작
    }
});
```

2 익명클래스 객체

익명 클래스 객체 매번 생성 X

- Button이 여러 개이고 Button 간의 연관성이 많은 경우
- 이벤트 핸들러 함수 내에서 익명 클래스 외부의 변수를 참조하지 않는 경우 (1번과 동일)
- 나중에 다른 Button을 추가하여 사용할 가능성이 높은 경우

```
Button btn1 = (Button) findViewById(R.id.btn1);
btn1.setOnClickListener(click);
Button btn2 = (Button) findViewById(R.id.btn2);
btn2.setOnClickListener(click);

View.OnClickListener click = new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        switch (v.getId()) {
            case R.id.btn1:
                // btn1 동작
                break;
            case R.id.btn2:
                // btn2 동작
                break;
        }
    }
};
```

상단 - JAVA

3 Listener 인터페이스를 implements 하는 Listener 클래스

이름이 부여된 Listener 클래스가 만들어져 이벤트가 어디서 처리되는지 명확히 구분

- Listener 인터페이스를 상속해 가독성을 높이고 싶은 경우
- 나중에 다른 view를 추가하여 사용할 가능성이 높은 경우

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        BtnOnClick btnOnClick = new BtnOnClick();  
  
        Button btn1 = (Button) findViewById(R.id.btn1);  
        btn1.setOnClickListener(btnOnClick);  
        Button btn2 = (Button) findViewById(R.id.btn2);  
        btn2.setOnClickListener(btnOnClick);  
    }  
}
```

```
// 클래스를 따로 생성하여 이벤트 리스너 상속  
class BtnOnClick implements View.OnClickListener {  
    @Override  
    public void onClick(View v) {  
        switch (v.getId()) {  
            case R.id.btn1:  
                // btn1 동작  
                break;  
            case R.id.btn2:  
                // btn2 동작  
                break;  
        }  
    }  
}
```

4 Listener 인터페이스를 implements 하는 activity

- 이벤트 핸들러 함수에서 많은 수의 Activity를 액세스해야 하는 경우
- Activity 내부의 view에 대한 click 이벤트를 한 곳에서 처리하고 싶은 경우
- 익명 클래스 또는 별도의 이벤트 Listener를 만들고 싶지 않은 경우

```
// activity에 이벤트 리스너 상속  
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
  
    // Button의 이벤트 리스너로 this(MainActivity) 지정  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        Button btn1 = (Button) findViewById(R.id.btn1);  
        btn1.setOnClickListener(this);  
        Button btn2 = (Button) findViewById(R.id.btn2);  
        btn2.setOnClickListener(this);  
    }  
  
    // View.OnClickListener를 implements하므로 onClick() 함수를 오버라이딩  
    @Override  
    public void onClick(View v) {  
        switch (v.getId()) {  
            case R.id.btn1:  
                // btn1 동작  
                break;  
            case R.id.btn2:  
                // btn2 동작  
                break;  
        }  
    }  
}
```

세부 페이지

(2)



커스텀 시계

XML //시간 설정

```
<Button
    android:id="@+id/timebtn1"
    android:layout_width="80dp"
    android:layout_height="35dp"
    android:layout_gravity="center"
    android:layout_marginHorizontal="10dp"
    android:layout_weight="1"
    android:background="@drawable/shape"
    android:backgroundTint="@color/gray_green"
    android:gravity="center"
    android:onClick="popTimePicker"
    android:text="Select time"
    tools:ignore="OnClick"

/>
```

//클릭시 시간 설정 팝업

세부 페이지

(2)

물주기 일정 추가

시작 시간

01:25

오전

오후

종료시간

01:25

오전

오후

커스텀 시계

JAVA //시간 설정

```
package com.example.flowerparty.activity;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Locale;
import android.app.AlertDialog;
import android.app.TimePickerDialog;
import android.widget.TimePicker;
import com.example.flowerparty.R;

public class PlantsControlDetailActivity extends AppCompatActivity {

    //시간선택
    Button timebtn1;
    Button button3;
    int hour, minute;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_plants_control_detail);
```

```
// 시작시간선택1
timebtn1 = findViewById(R.id.timebtn1);
long systemTime = System.currentTimeMillis();
Date d = new Date(systemTime);
SimpleDateFormat format1 = new SimpleDateFormat("hh:mm");
String formatDate = format1.format(d);
timebtn1.setText(formatDate);
```

```
// 취소버튼
button3 = findViewById(R.id.button3);
button3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) { finish(); }
});
```

public void popTimePicker (View View){ //시계 팝업

```
TimePickerDialog.OnTimeSetListener onTimeSetListener
    = new TimePickerDialog.OnTimeSetListener() {
    @Override
    public void onTimeSet(TimePicker timePicker, int selectedHour, int selectedMinute) {
        hour = selectedHour;
        minute = selectedMinute;
        timebtn1.setText(String.format(Locale.getDefault(), "%02d:%02d", hour, minute));
    }
};

int style = AlertDialog.THEME_HOLO_DARK;

TimePickerDialog timePickerDialog =
    new TimePickerDialog(this, style, onTimeSetListener, hour, minute, true);
timePickerDialog.setTitle("Select Time");
timePickerDialog.show();}
```

세부 페이지

(2)

물주기 일정 추가

시작 시간

01:25

오전

오후

종료시간

01:25

오전

오후

커스텀 버튼

XML

```
<RadioGroup
    android:id="@+id/toggle"
    android:layout_width="80dp"
    android:layout_height="35dp"
    android:layout_gravity="center"
    android:background="@drawable/shape"
    android:backgroundTint="@color/gray_green"
    android:checkedButton="@+id/offer"

    android:orientation="horizontal">

    <RadioButton
        android:id="@+id/am1"
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_marginLeft="1dp"
        android:layout_marginTop="1dp"
        android:layout_marginBottom="1dp"
        android:layout_weight="1"
        android:background="@drawable/ampm_back"
        android:button="@null"
        android:gravity="center"
        android:text="오전"
        android:textColor="@color/white" />
```

```
<RadioButton
    android:id="@+id/pm1"
    android:layout_width="0dp"
    android:layout_height="match_parent"
    android:layout_marginTop="1dp"
    android:layout_marginRight="1dp"
    android:layout_marginBottom="1dp"
    android:layout_weight="1"
    android:background="@drawable/ampm_back"
    android:button="@null"
    android:gravity="center"
    android:text="오후"

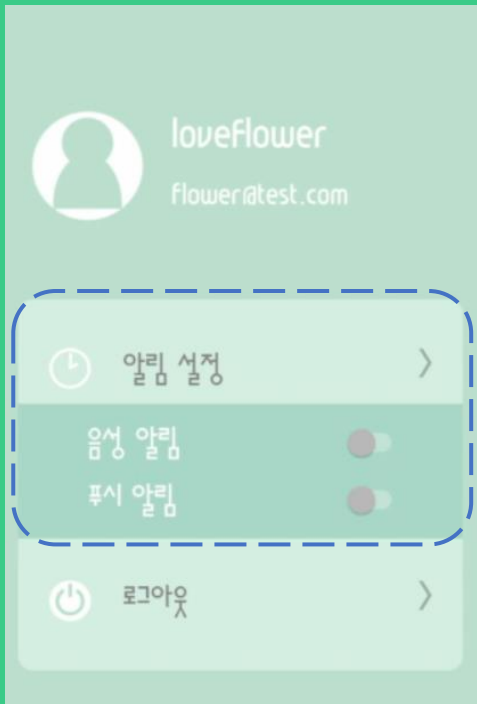
    android:textColor="@color/white" />
</RadioGroup>
```

drawable

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<selector xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:drawable="@drawable/shape"
        android:state_checked="true" />
    <item android:drawable="@drawable/grayshapebtn" />
</selector>
```

세부 페이지

(3)



Visibility

XML // 클릭시 하위 메뉴 노출

```
<LinearLayout
    android:id="@+id/settingBox"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/nav"
    android:orientation="vertical"
    android:visibility="gone"
    android:layout_marginVertical="5dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginHorizontal="10dp"
        android:layout_marginVertical="15dp"
        android:orientation="vertical">

        <LinearLayout
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="15dp"
            android:gravity="center"
            android:orientation="horizontal">
```

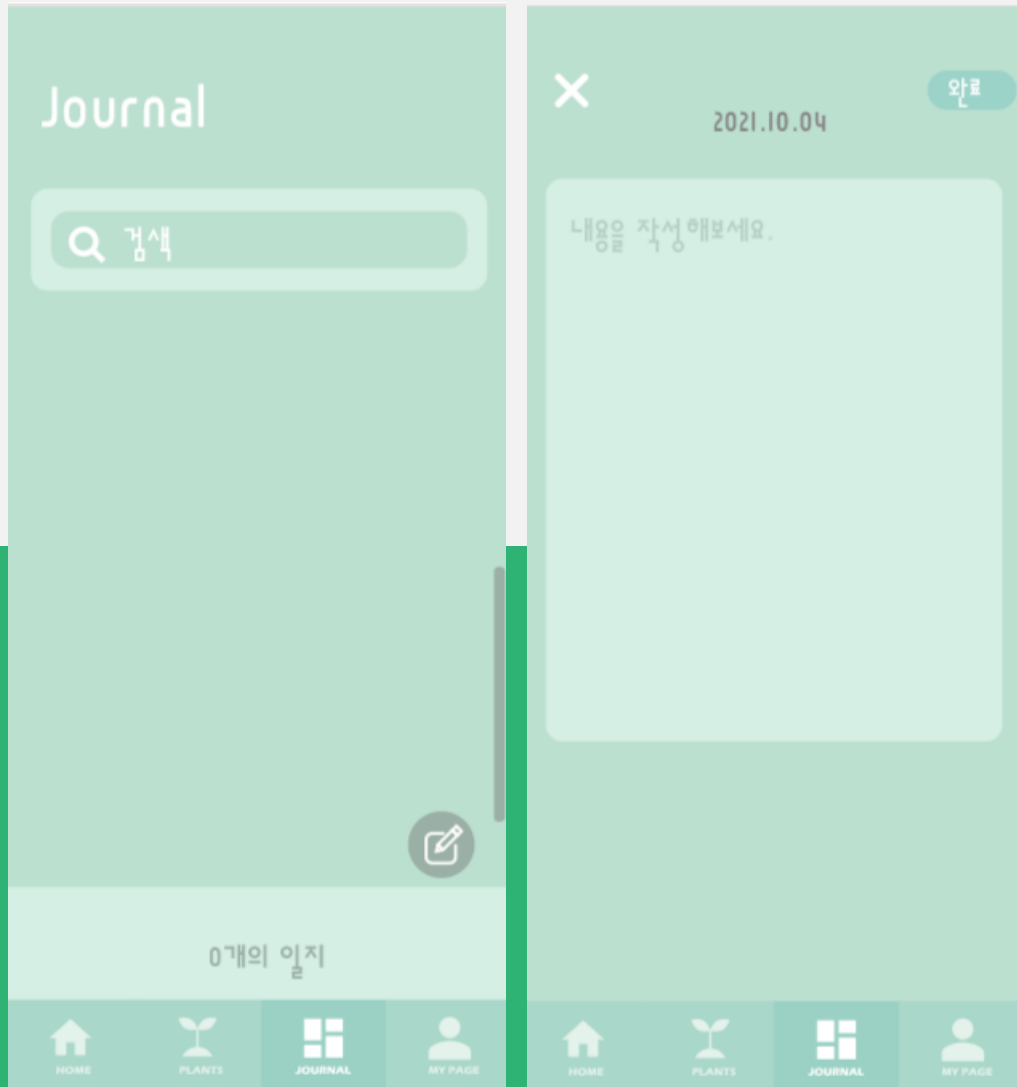
```
<TextView
    android:id="@+id/textView11"

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:fontFamily="@font/ahn2006_m"
    android:gravity="center"
    android:text="음성 알림"
    android:textColor="@color/white"
    android:textSize="25sp"
    android:textStyle="bold" />
```

```
<Switch
    android:id="@+id/switch2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginRight="30dp"
    android:layout_weight="1"
    android:gravity="center" />
```

```
</LinearLayout>
```

- **Visibility = "gone"**
 - 기준이 된 뷰가 사라져 해당 뷰 위치가 변경됨
- **Visibility = "invisible"**
 - 뷰가 공간을 그대로 차지하고 있어, 빈공간으로 보임

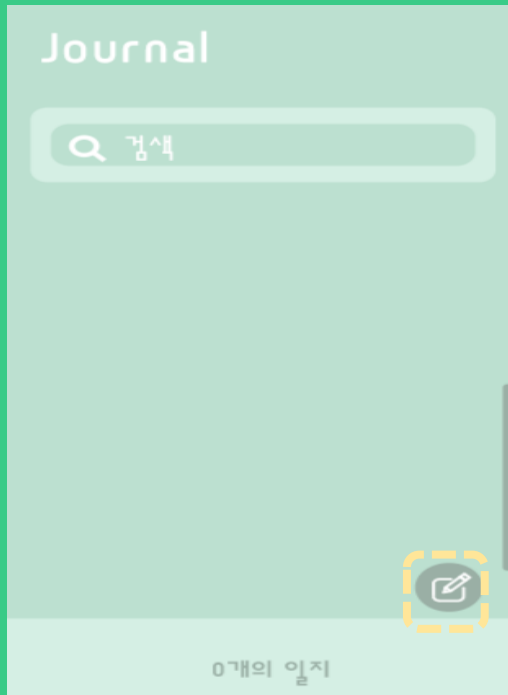


식물일지

커스텀 버튼2, 현재 날짜, 글쓰기,
화면전환(intent)

식물일지

(1)



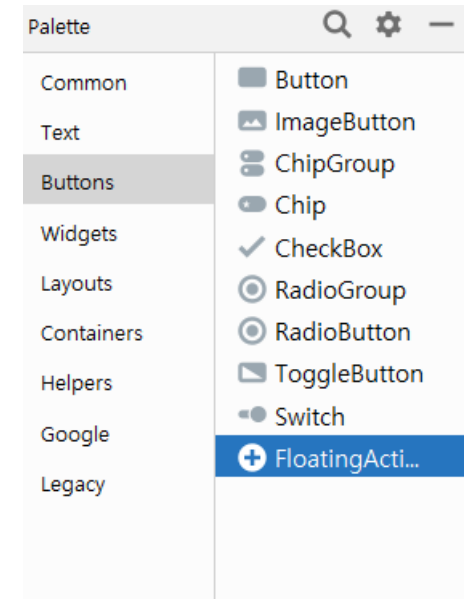
커스텀 버튼2

XML //원형버튼

```
<com.google.android.material.floatingactionbutton.FloatingActionButton  
    android:id="@+id/wbtn"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="end|bottom"  
    android:layout_marginRight="16dp"  
    android:layout_marginBottom="160dp"  
    android:src="@drawable/newpost"  
    app:backgroundTint="@color/gray"  
    app:elevation="6dp"  
    app:fabSize="normal">
```

```
</com.google.android.material.floatingactionbutton.FloatingActionButton>
```

//Design 탭



//floatingActionButton
//android:src="@drawable/이미지소스명"

식물일지

(2)



화면전환 INTENT

JAVA //화면 전환

```
import android.content.Intent;

wbtn = (FloatingActionButton) rootView.findViewById(R.id.wbtn);
wbtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // 새로운 메모 작성
        Intent intent = new Intent(ct, JournalDiaryActivity.class);
        startActivityForResult(intent, 0);
    }
});

return rootView;
```

AndroidManifest.xml

```
<activity android:name=".activity.JournalDiaryActivity" />
```

상황별 Intent

1 액티비티 → 액티비티

```
Intent intent = new Intent(this, subActivity.class);
//intent를 정의하면서 어느 액티비티로 갈지
startActivity(intent);
```

2 프래그먼트 → 액티비티

```
Intent intent = new Intent(getActivity(),
subActivity.class); //this 대신 getActivity()
startActivity(intent);
```

식물일지

(3)



커스텀 날짜

JAVA //현재 날짜

```
public class JournalDiaryActivity extends AppCompatActivity {

    TextView datetxt;
    ImageView xmark1;
    Button Button2;
    private RbPreference pref;
    EditText et_journal_title, et_journal_contents;
    public static final String TAG = "JournalDiaryActivity";
    int idx;
    String title, content;

    // 현재시간출력
    long systemTime = System.currentTimeMillis();
    Date d = new Date(systemTime);
    SimpleDateFormat format1 = new SimpleDateFormat("YYYY.MM.dd");
    String formatDate = format1.format(d);

    datetxt =(TextView) findViewById(R.id.datetxt);
    datetxt.setText(formatDate);
```

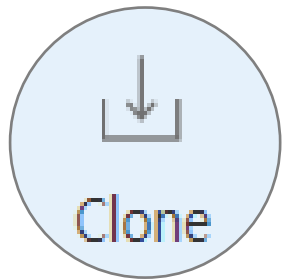
XML //현재 날짜

```
<TextView
    android:id="@+id/datetxt"
    android:layout_width="135dp"
    android:layout_height="30dp"
    android:layout_gravity="bottom|center"
    android:layout_weight="1"
    android:backgroundTint="@color/main"
    android:ems="10"
    android:fontFamily="@font/ahn2006_m"
    android:gravity="center"
    android:text="2021.10.04"
    android:textColor="@color/gray"
    android:textSize="28sp"
    android:textStyle="bold" />
```

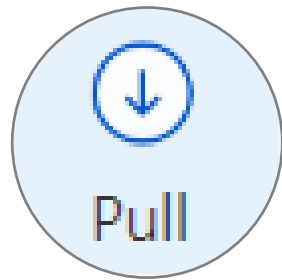
SOURCETREE

CLONE / PULL / PUSH

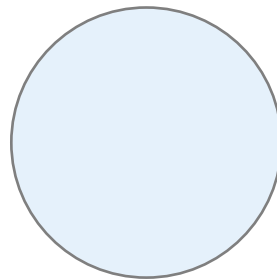
SOURCETREE 사용



CLONE



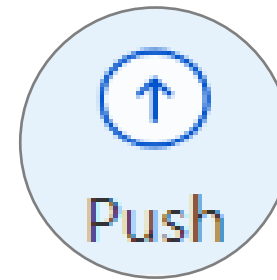
PULL



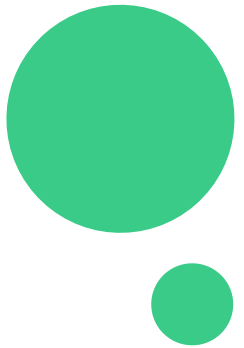
WORK



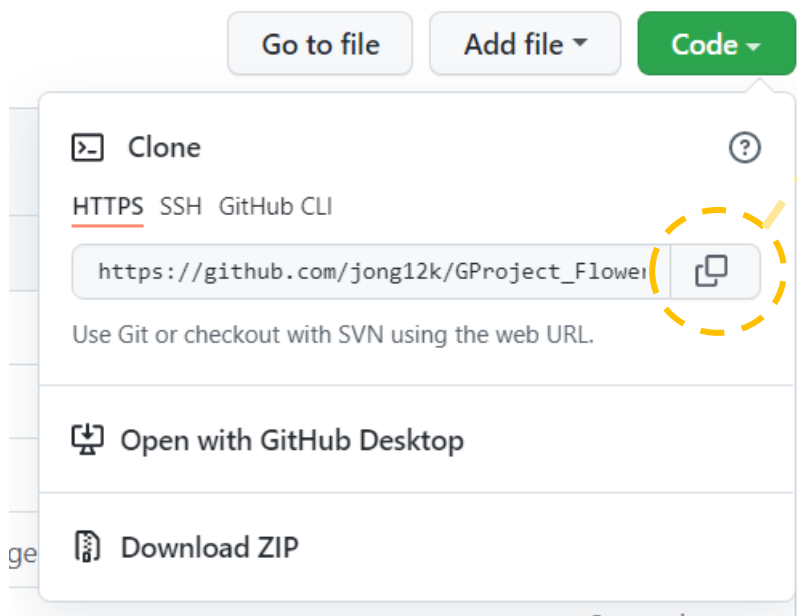
COMMIT



PUSH



Clone



Clone

Cloning is even easier if you set up a [remote account](#)

소스 경로 / URL:

탐색

저장소 종류: ? 경로 / URL 누락

목적지 경로:

탐색

이름:

Local Folder:

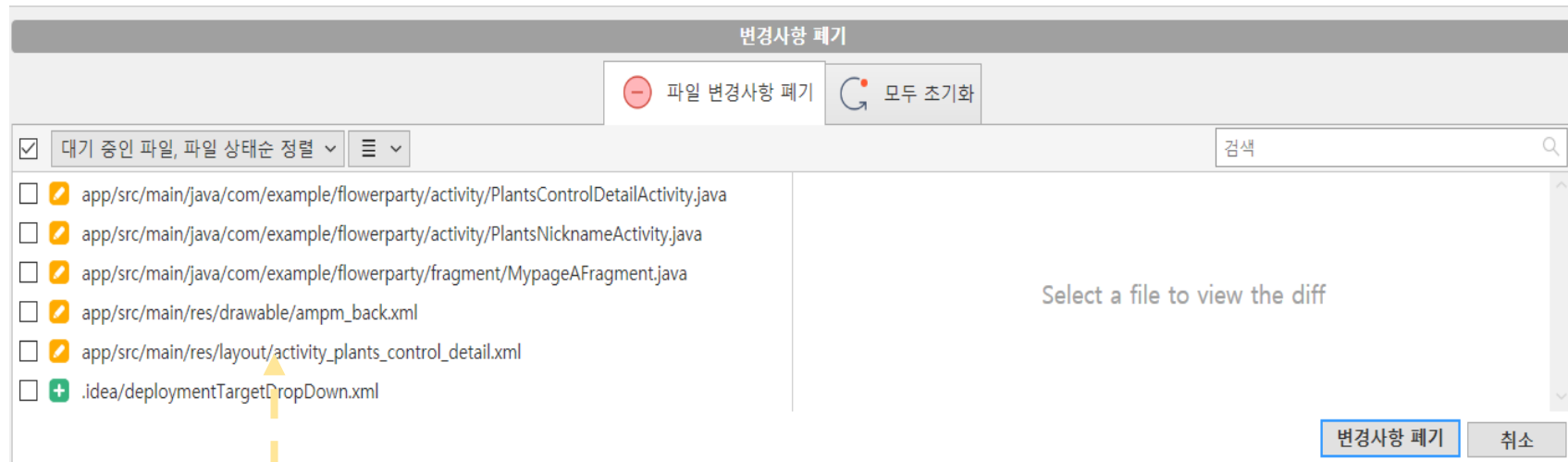
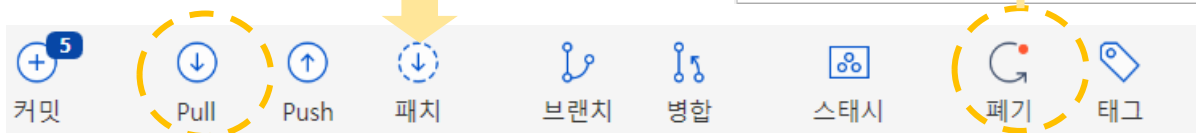
[루트]

> 고급 옵션

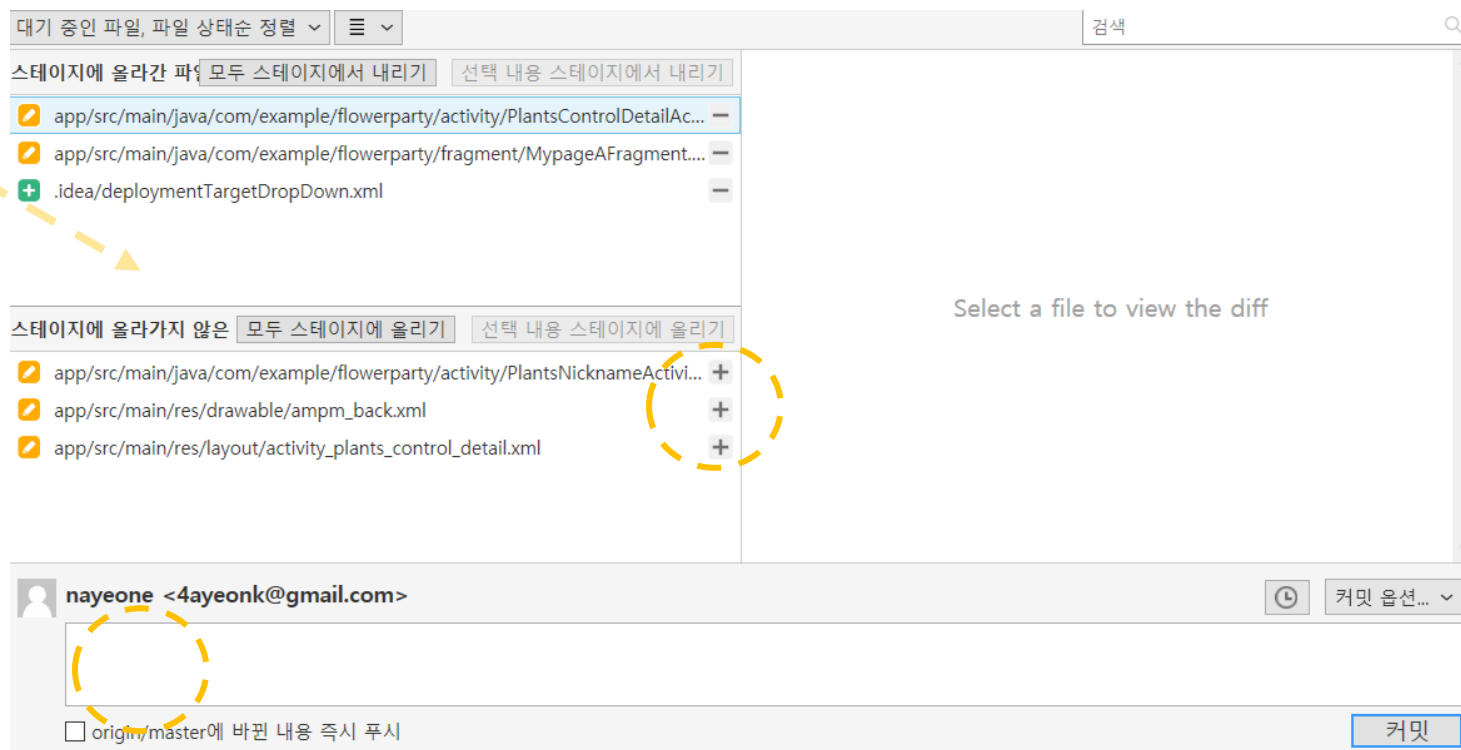
클론

PULL

원격과 소스트리 동기화



COMMIT



PUSH



Push : Flower

다음 저장소에 푸시: origin ▼ https://github.com/jong12k/GProject_FlowerParty.git

푸시할 브랜치

푸시?	로컬 브랜치	리모트 브랜치	추적?
<input checked="" type="checkbox"/>	master	master ▼	<input checked="" type="checkbox"/>

☒ 모두선택

☒ 모든 태그 푸시 ☐ 강제 푸시

Push 취소

“

감사합니다

”