

OUR
SOPT

SHOUT OUR PASSION TOGETHER

1차 세미나

– 각종 레이아웃의 활용 –

SHOUT OUR PASSION TOGETHER
SOPT

0. 오늘 목표, 프로젝트 생성하기

1. View의 계층 구조

2. ConstraintLayout

3. LinearLayout

4. RelativeLayout

5. 기능을 입혀보자

0. 오늘 목표, 프로젝트 생성하기

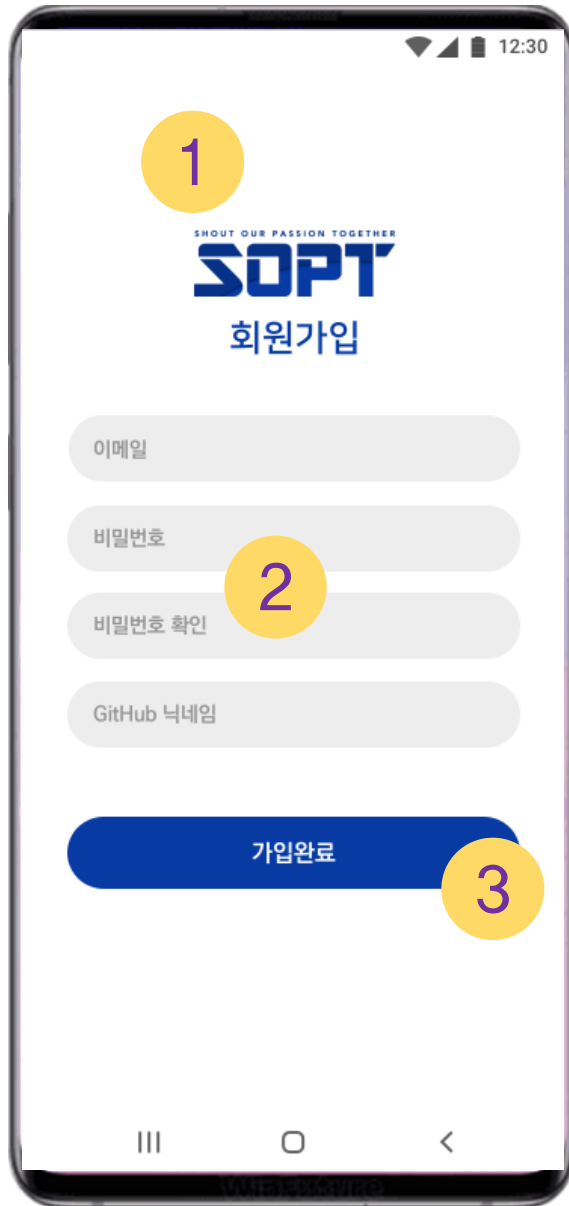
0. 오늘 목표, 프로젝트 생성하기



1. 로그인, 회원가입 화면 완성하기

#	description
1	로그인 화면 이미지
2	아이디 및 비밀번호 입력란
3	로그인 버튼
4	회원가입 버튼

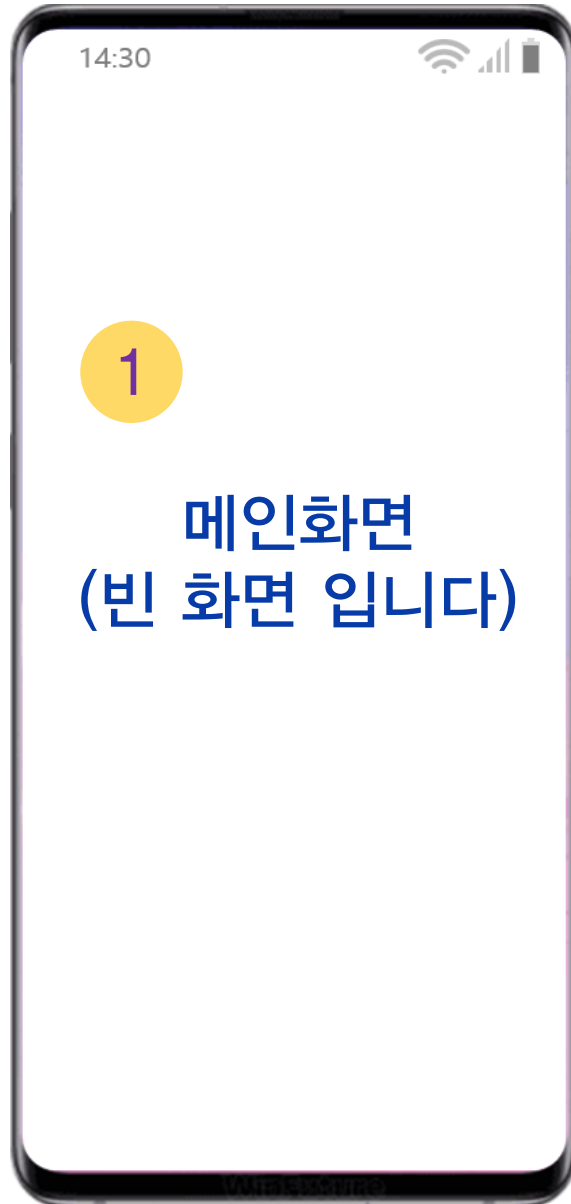
0. 오늘 목표 , 프로젝트 생성하기



1. 로그인, 회원가입 화면 완성하기

#	description
1	회원가입창 이미지
2	아이디, 비밀번호, 비밀번호 확인, 휴대폰 번호 입력란
3	가입완료 버튼

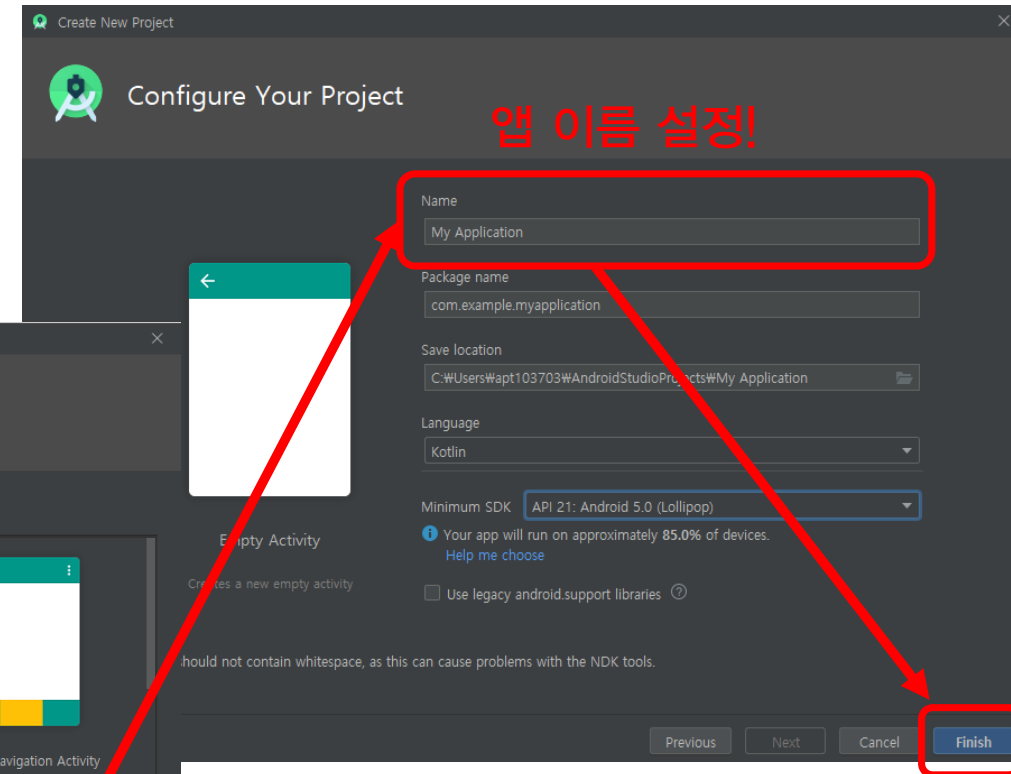
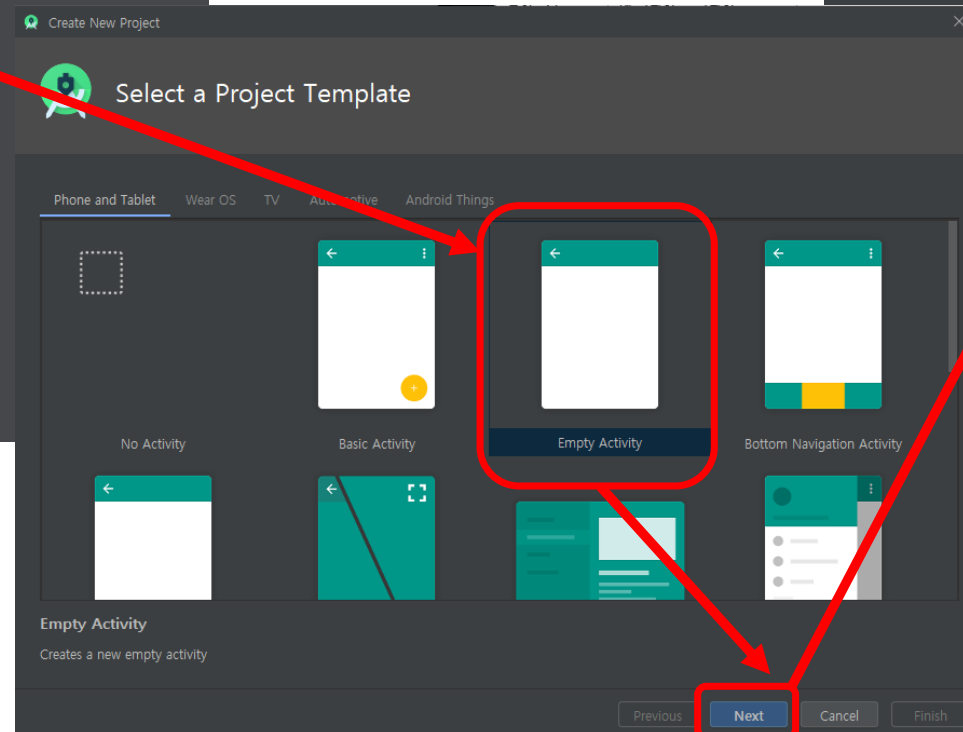
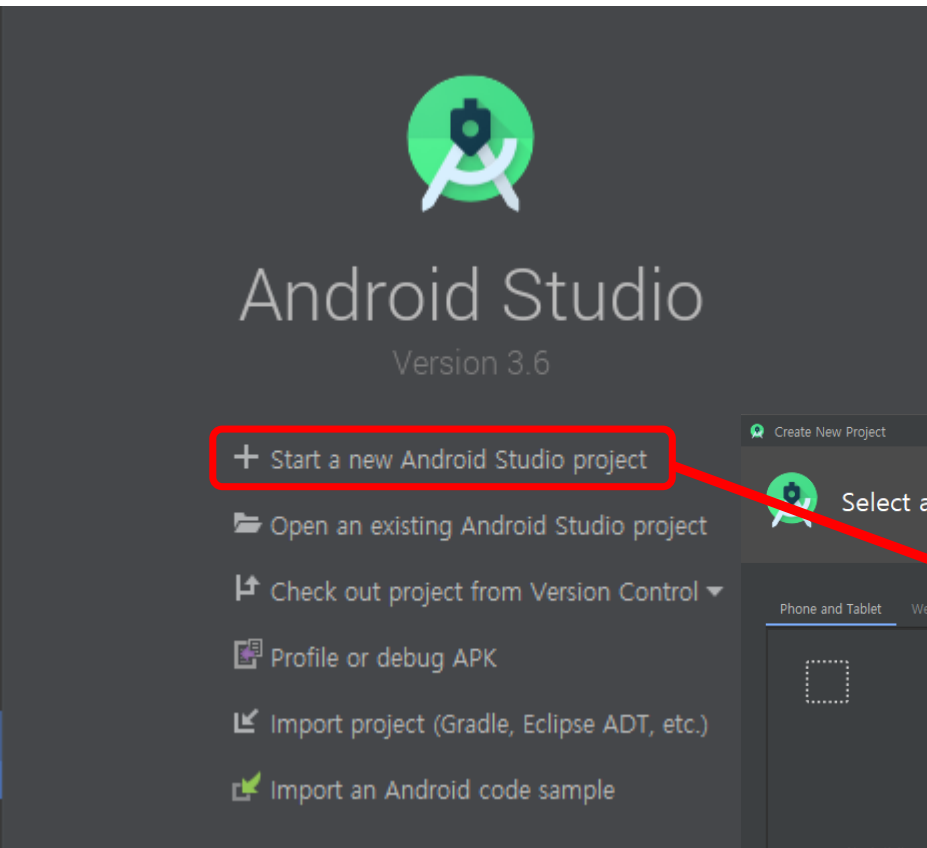
0. 오늘 목표 , 프로젝트 생성하기



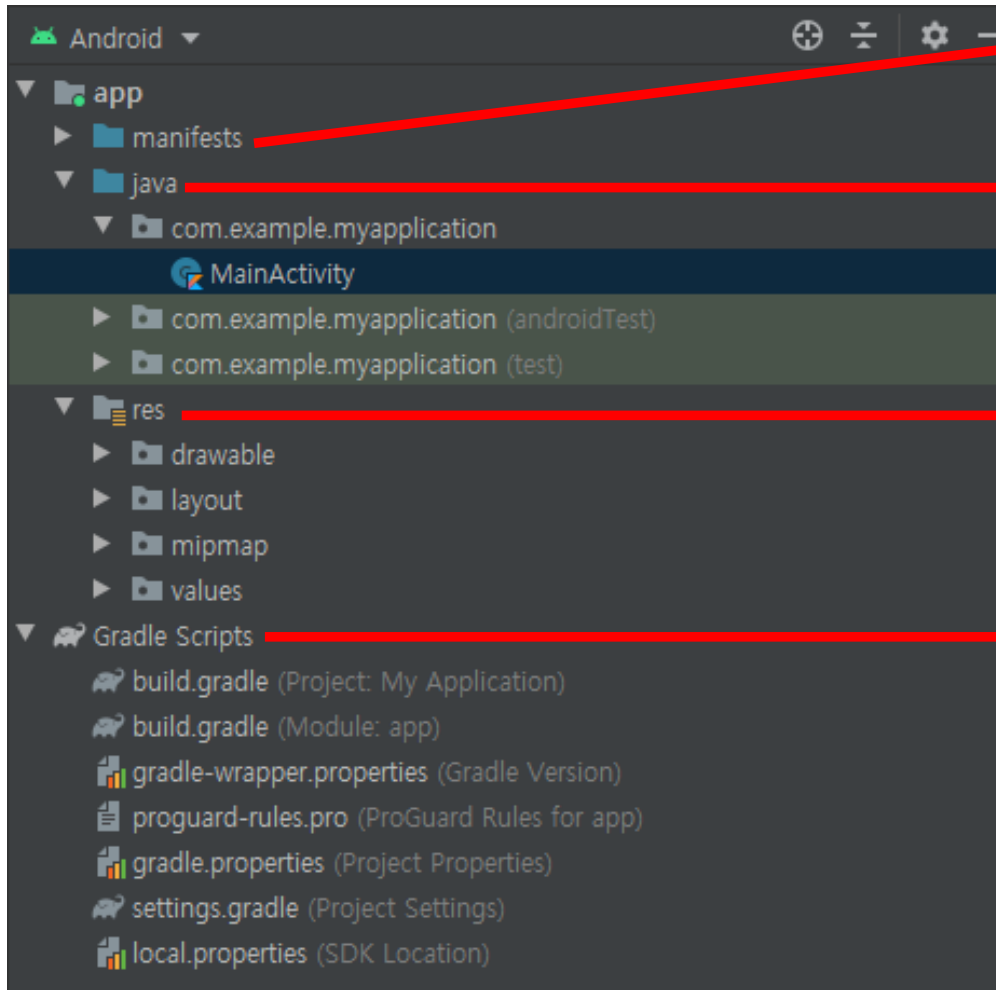
1. 로그인, 회원가입 화면 완성하기

#	Description
1	메인화면 입니다

0. 오늘 목표 , 프로젝트 생성하기



0. 오늘 목표 , 프로젝트 생성하기



앱에 대한 기본 정보 정의

프로젝트 코드

Resource가 들어감(이미지 / 레이아웃 / 문자열 / 스타일 등)

프로젝트 빌드 방법 정의하는 곳

1. View의 계층 구조

부모 – 자식 관계



1. View의 계층 구조

부모 / 자식

ViewGroup

ConstraintLayout, LinearLayout,
RelativeLayout... ???Layout

자식

View

TextView, EditText, Button,
RadioButton,
ImageView...

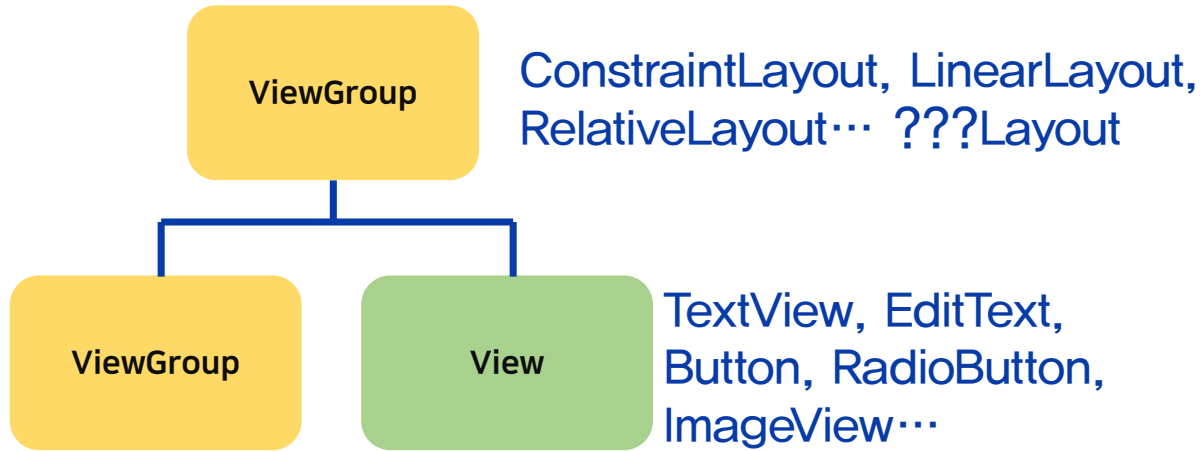
ViewGroup

ViewGroup

View

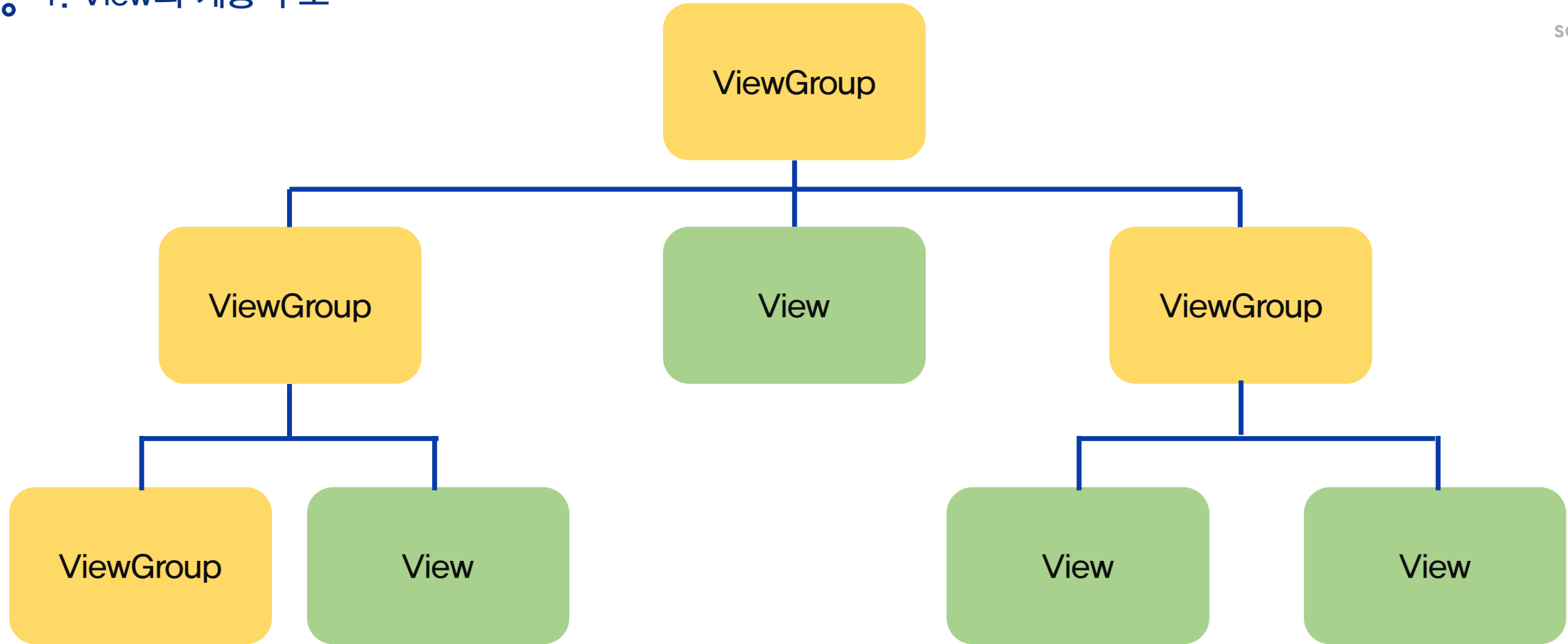


1. View의 계층 구조



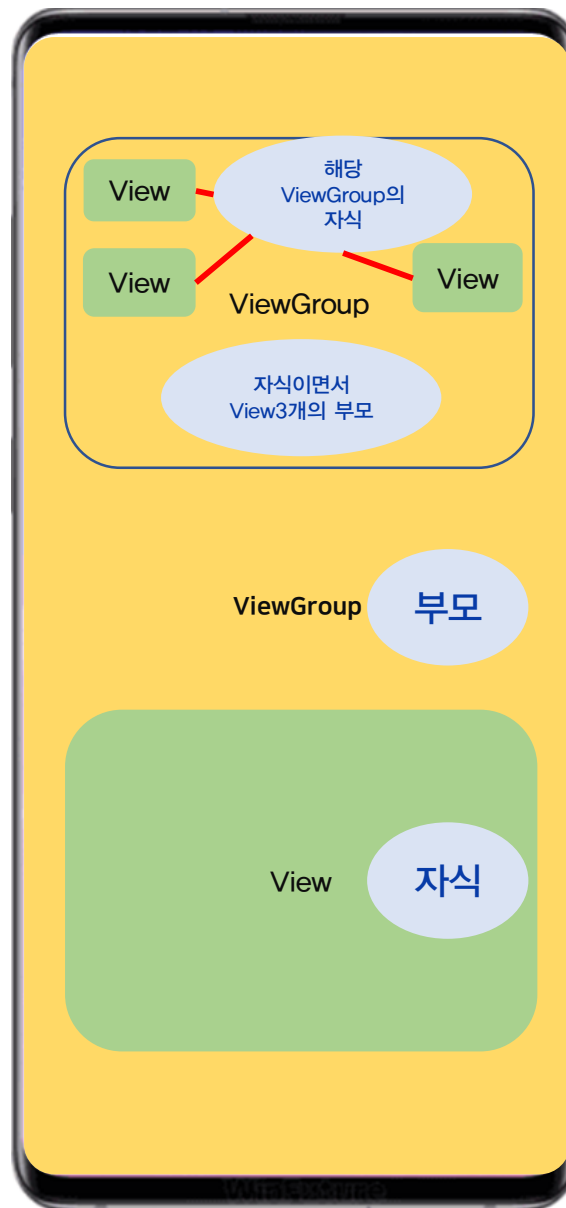
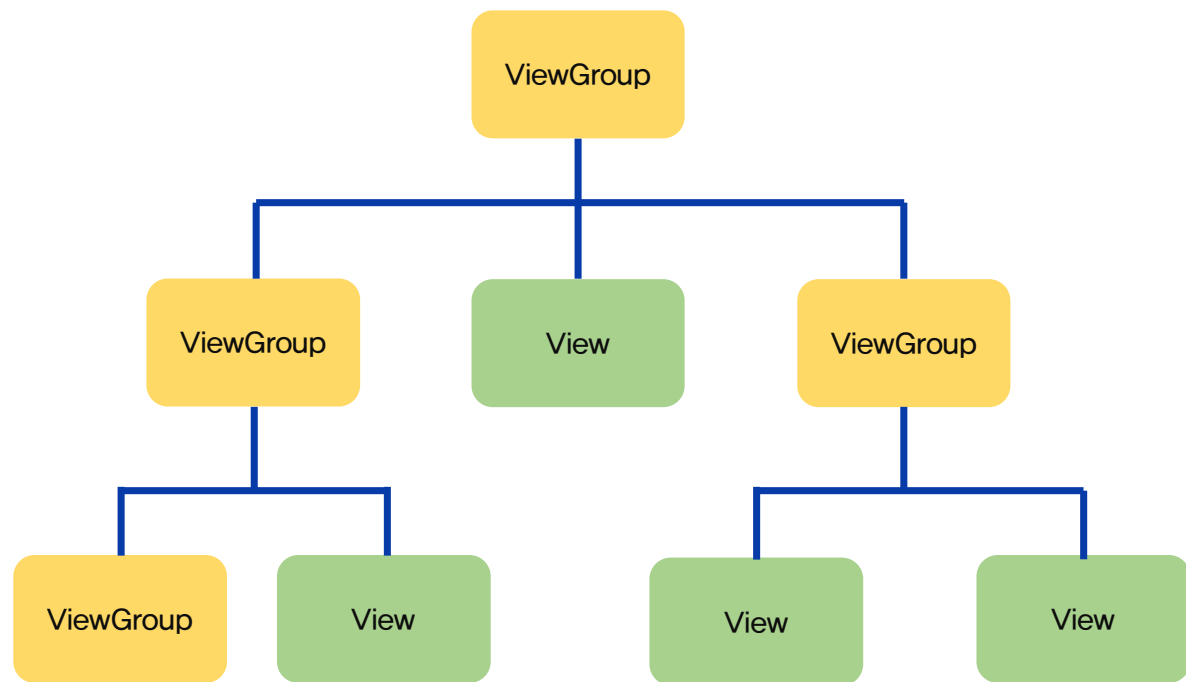


1. View의 계층 구조





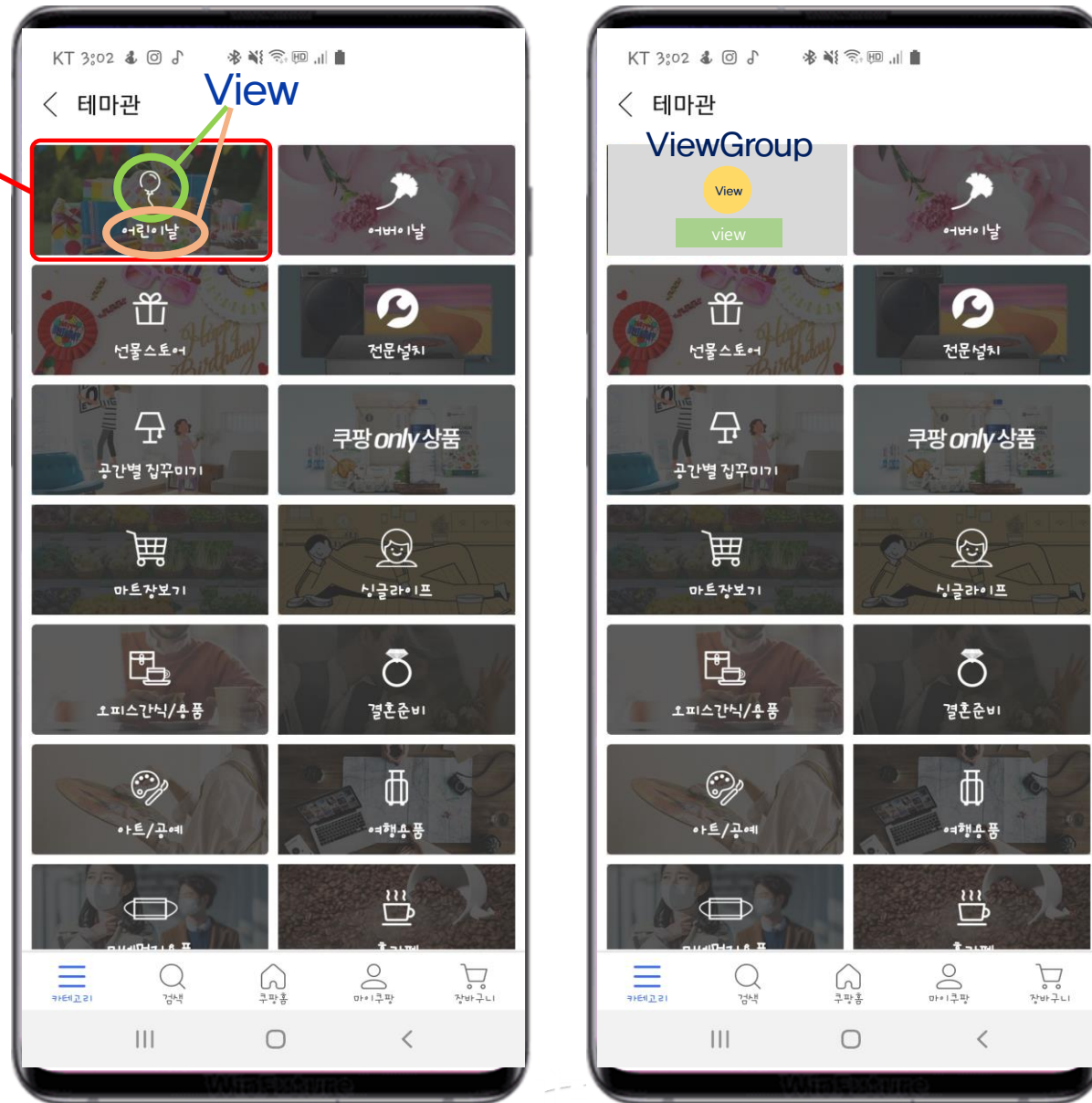
1. View의 계층 구조



1. View의 계층 구조

ViewGroup

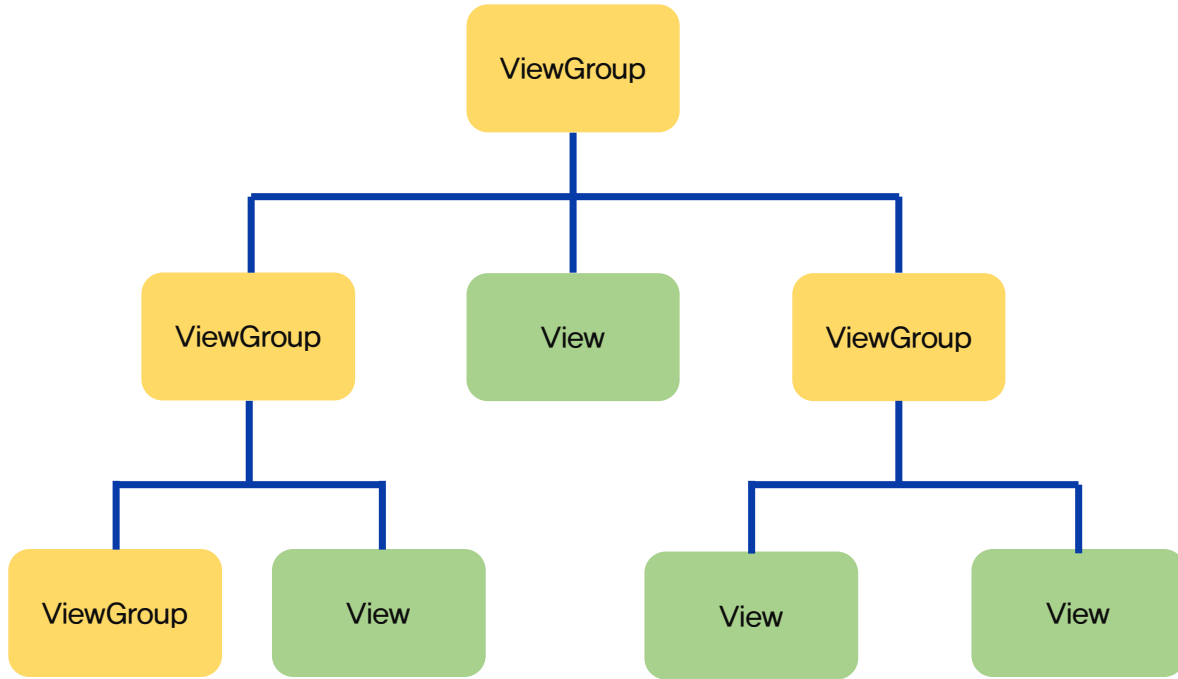
View





1. View의 계층 구조

designer
developer
Planner
sopT



ViewGroup

View

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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



1. View의 계층 구조

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    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"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

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

Width = 가로, Height = 세로. 정말 쉽다..!

위 두개는 View든, viewGroup이든 필수로 정의해 주어야 한다.

Match_parent : 부모에 맞게 꽉 채운다!

Wrap_content : 내용에 맞게 크기를 조정.

Width에 Match_parent를 줬을 때

화면

Width에 wrap_content를 줬을 때

1. View의 계층 구조

```
<TextView  
    android:id="@+id/textView"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_margin="16dp"  
    android:padding="16dp"  
    android:text="TextView"  
    tools:layout_editor_absoluteX="142dp"  
    tools:layout_editor_absoluteY="488dp" />
```

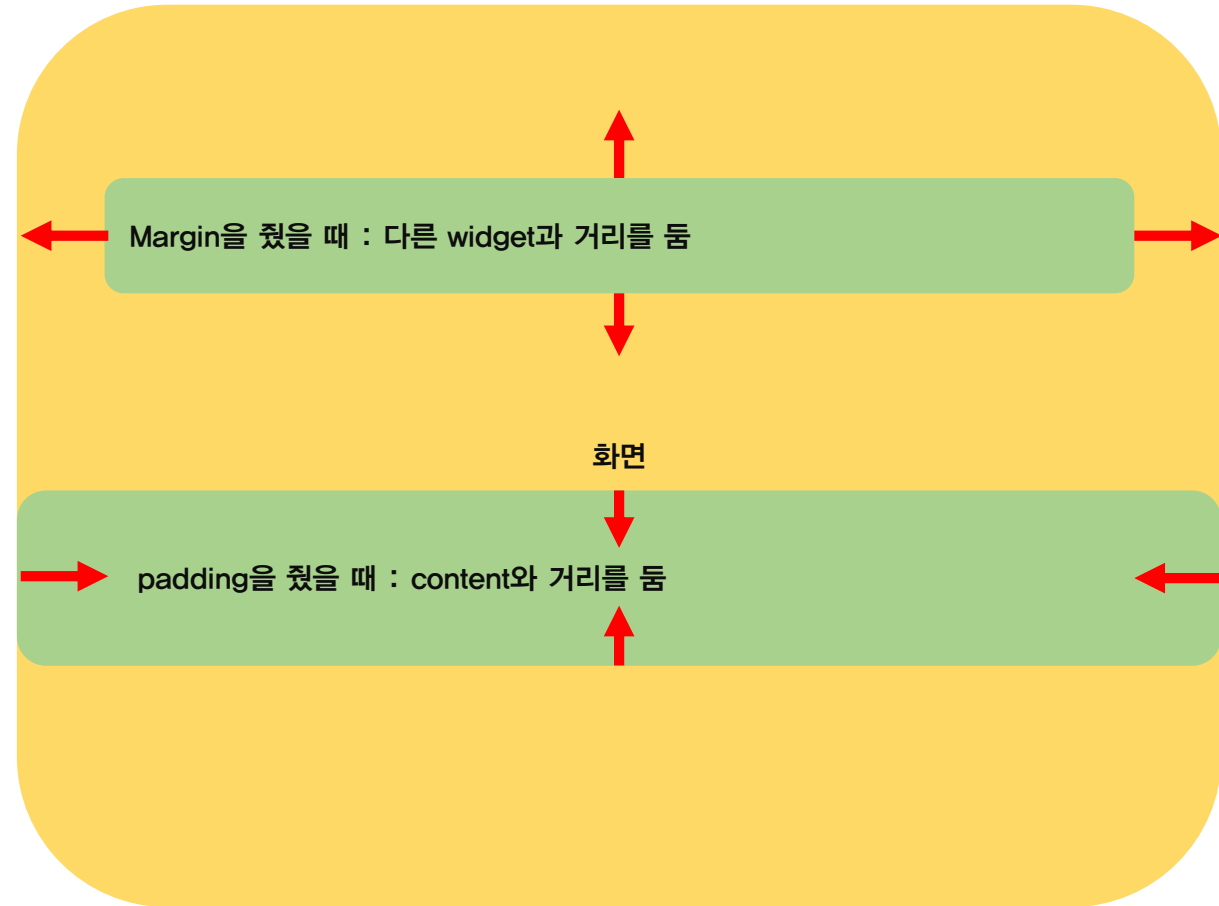
가로

- marginHorizontal, paddingHorizontal
- marginLeft/Right, paddingLeft/Right

세로

- marginVertical, paddingVertical
- marginTop/Bottom, paddingLeft/Right

Width = match_parent인데.. 뭔가 공간이 필요해..!



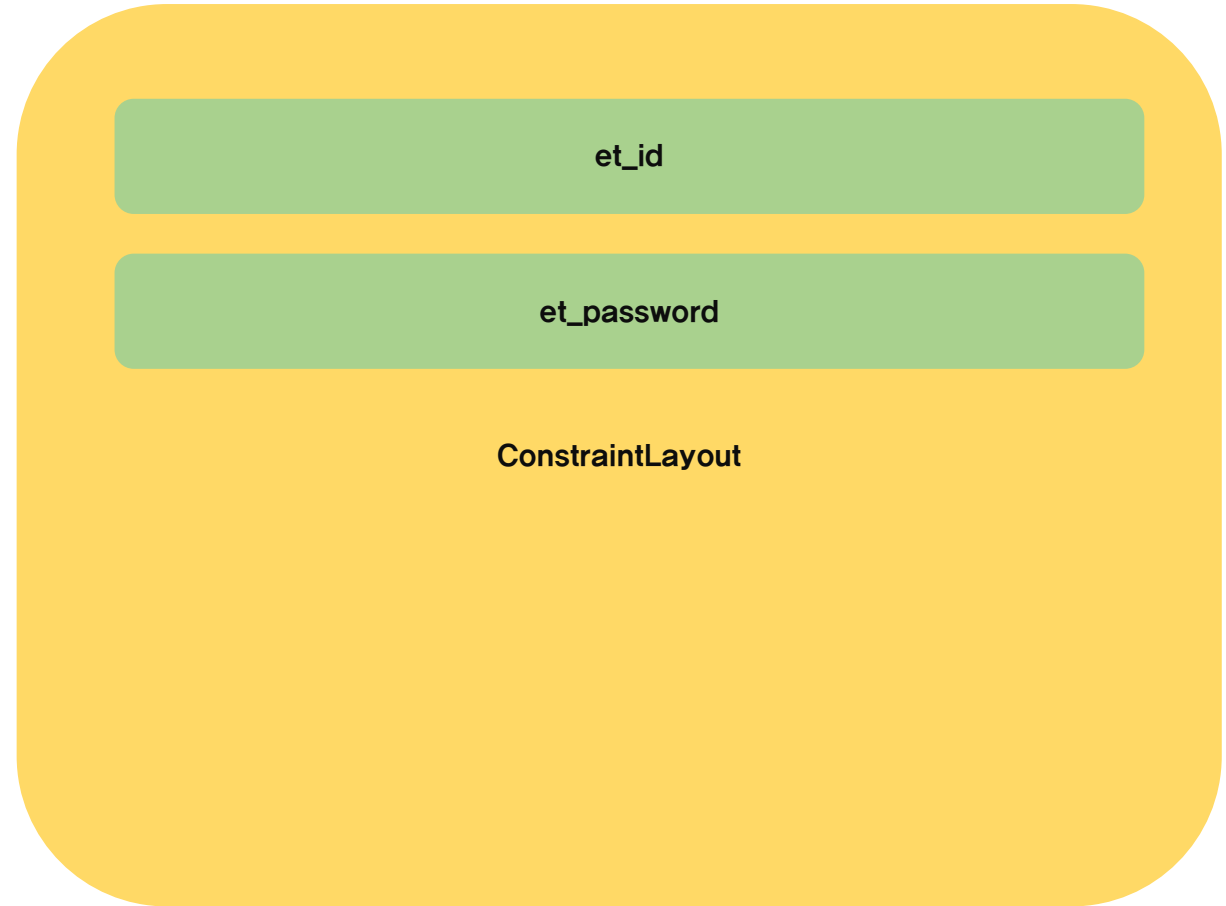


1. View의 계층 구조

```
<EditText
    android:id="@+id/et_id"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="24dp"
    android:layout_marginHorizontal="24dp"
    android:inputType="text"
    android:hint="아이디"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/img_main" />

<EditText
    android:id="@+id/et_password"
    android:layout_width="match_parent"
    android:layout_marginHorizontal="24dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="16dp"
    android:inputType="textPassword"
    android:hint="비밀번호"
    app:layout_constraintEnd_toEndOf="@+id/et_id"
    app:layout_constraintStart_toStartOf="@+id/et_id"
    app:layout_constraintTop_toBottomOf="@+id/et_id" />
```

Id – View/ViewGroup의 유일한 이름(식별자)



2. Constraint Layout

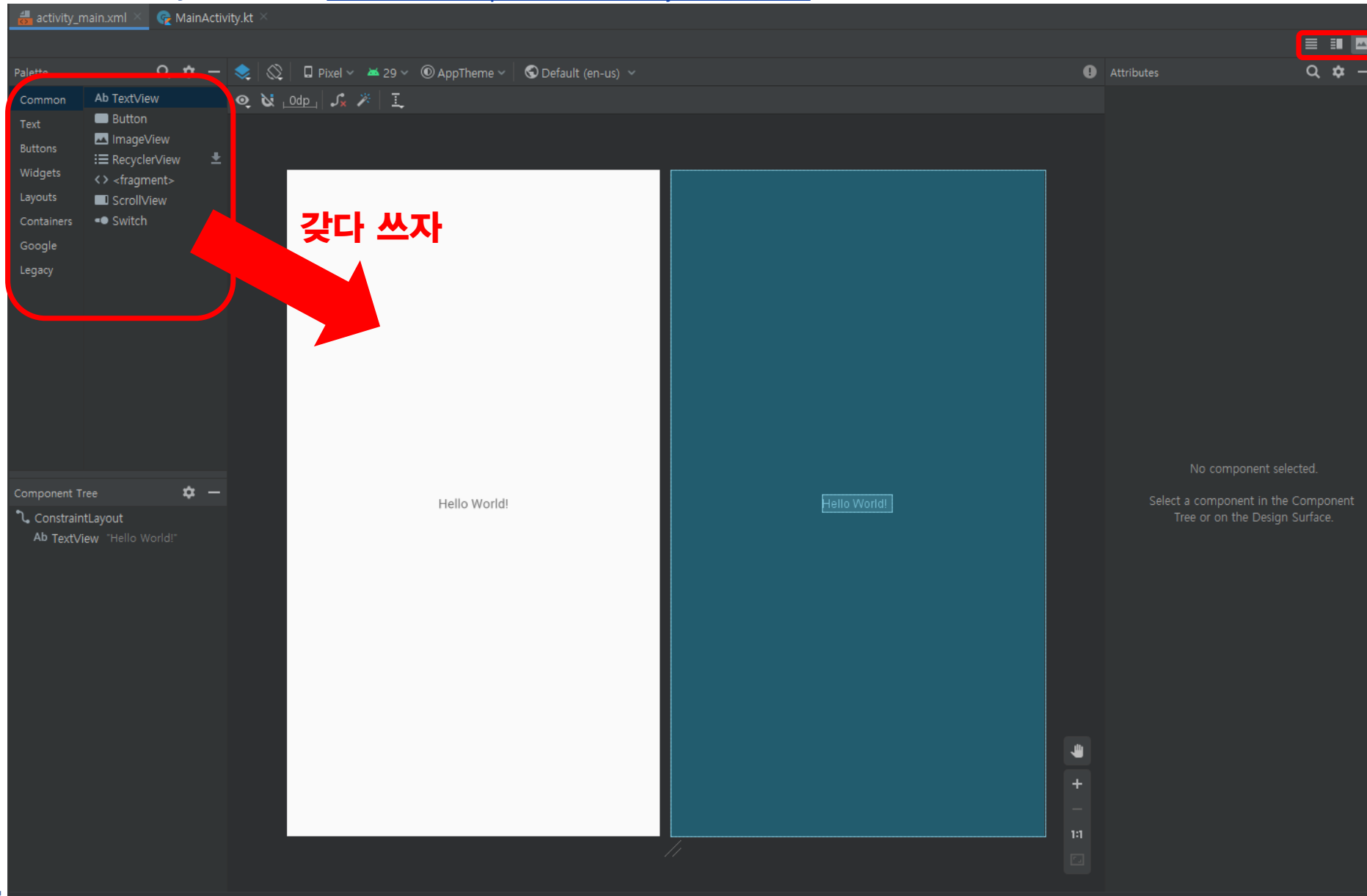


2. ConstraintLayout

[ConstraintLayout 문서](#)

[Android Developer ConstraintLayout 공식 문서](#)

deSigner
develop
Planner
sopT





상대적 제약조건 걸기.

가로와 세로 모두 제약조건이 걸려 있어야 한다!





2. ConstraintLayout

[ConstraintLayout 문서](#)

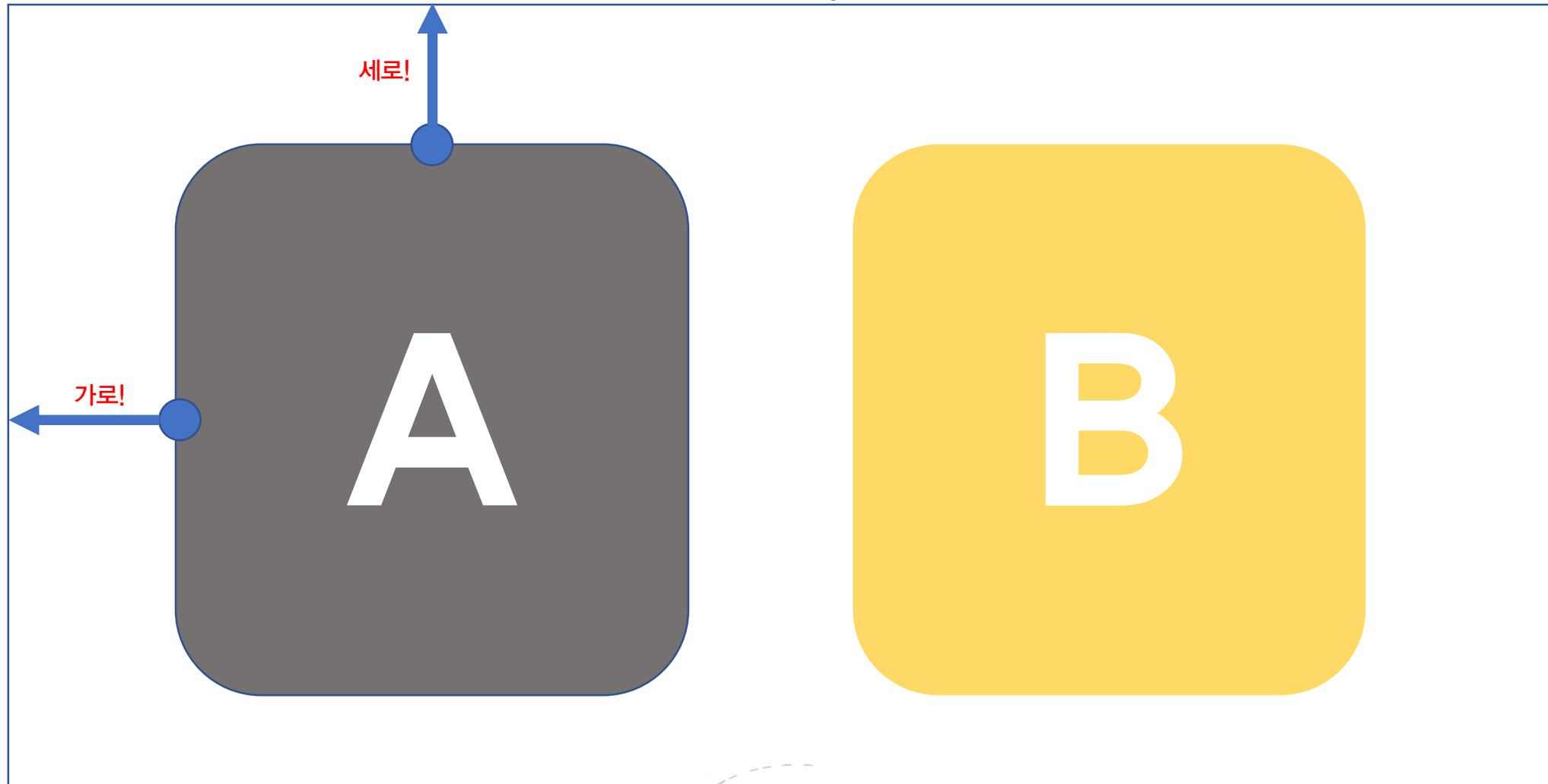
[Android Developer ConstraintLayout 공식 문서](#)

deSigner
develop
Oper
Planner
sopT

상대적 제약조건 걸기.

가로와 세로 모두 제약조건이 걸려 있어야 한다!

CosntraintLayout





2. ConstraintLayout

[ConstraintLayout 문서](#)

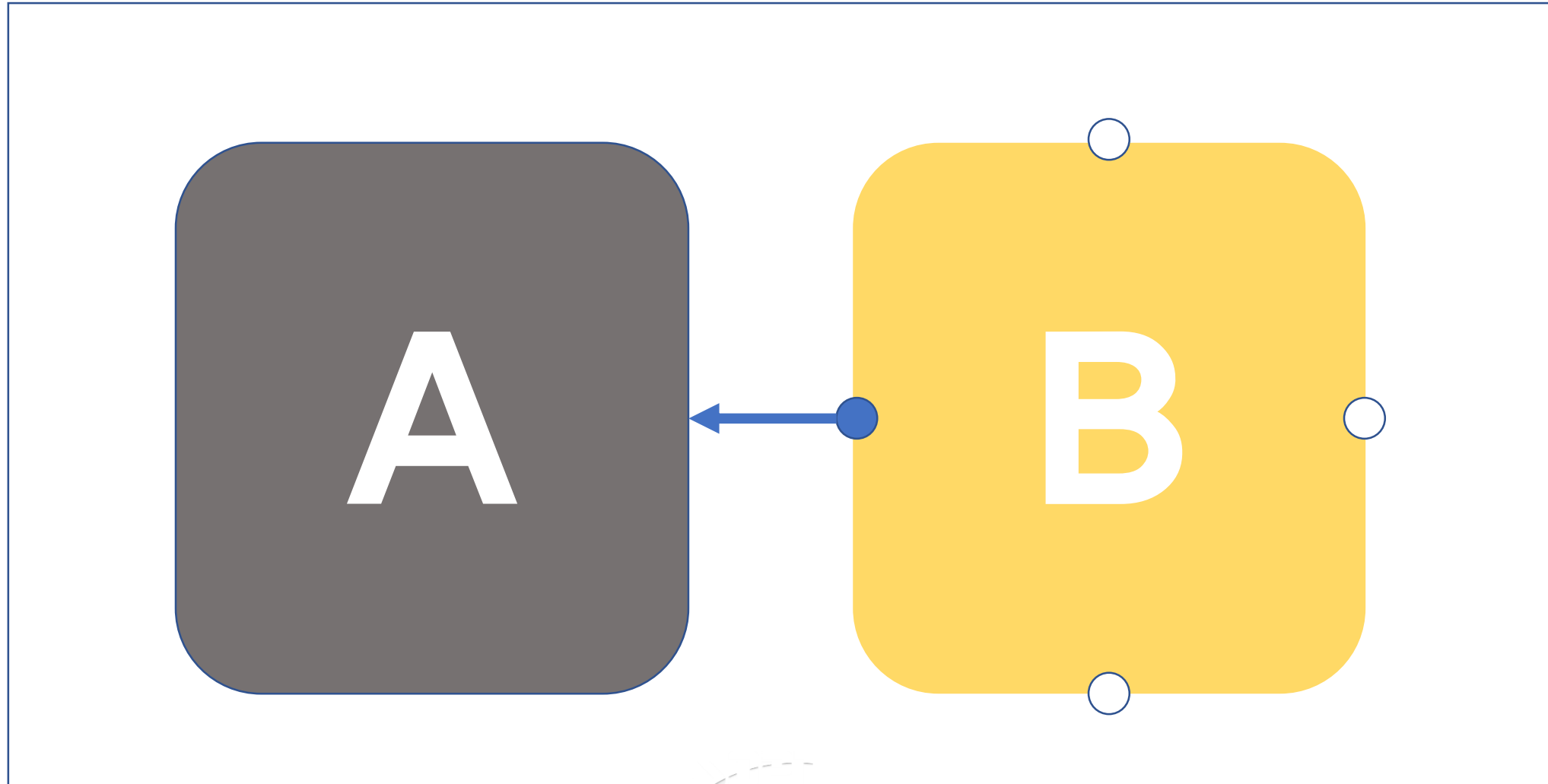
[Android Developer ConstraintLayout 공식 문서](#)

deSigner
develop
Oper
Planner
sopT

상대적 제약조건 걸기.

가로와 세로 모두 제약조건이 걸려 있어야 한다!

CosntraintLayout





2. ConstraintLayout

[ConstraintLayout 문서](#)

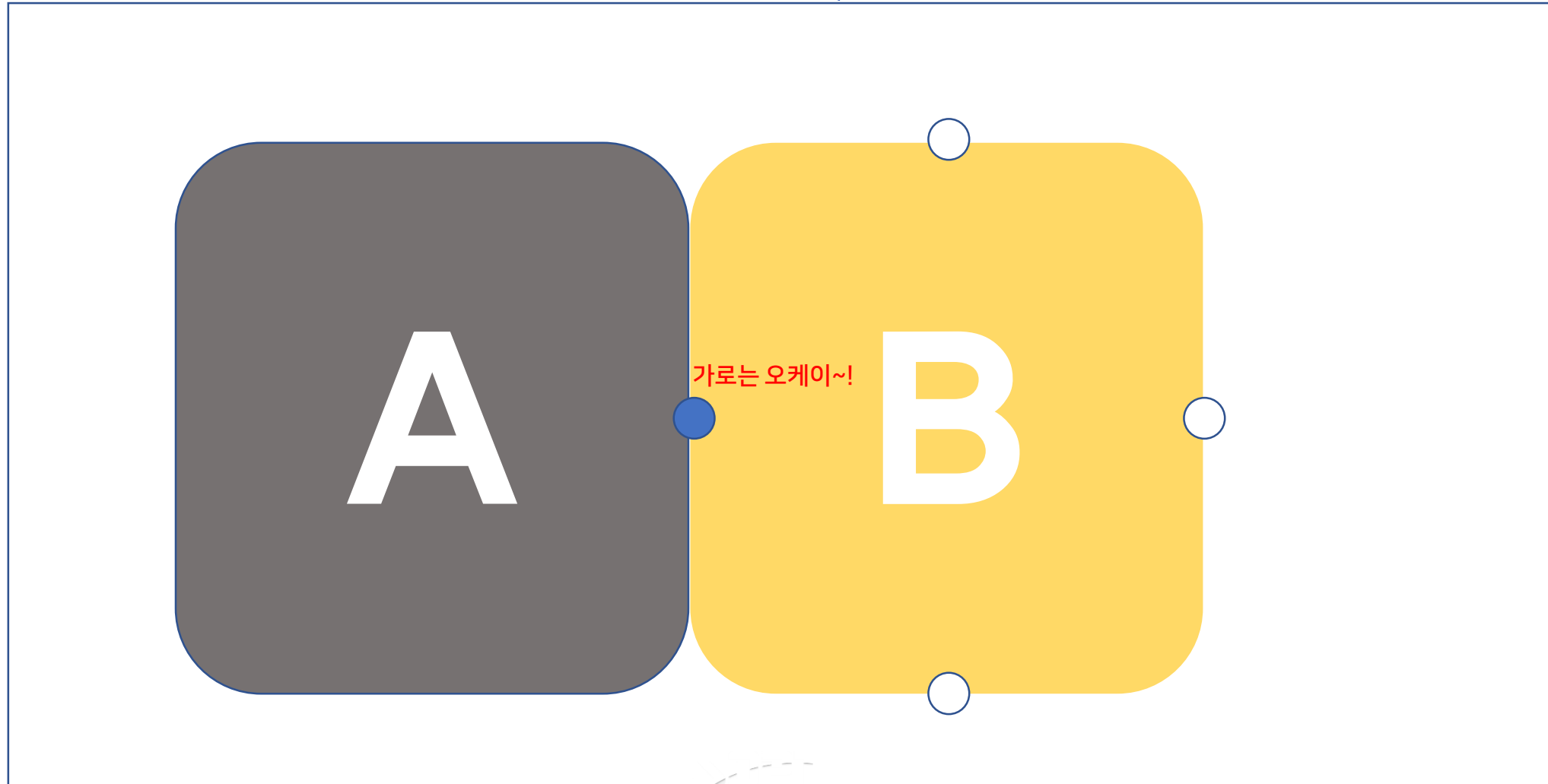
[Android Developer ConstraintLayout 공식 문서](#)

deSigner
develop
Oper
Planner
sopT

상대적 제약조건 걸기.

가로와 세로 모두 제약조건이 걸려 있어야 한다!

CosntraintLayout





2. ConstraintLayout

[ConstraintLayout 문서](#)

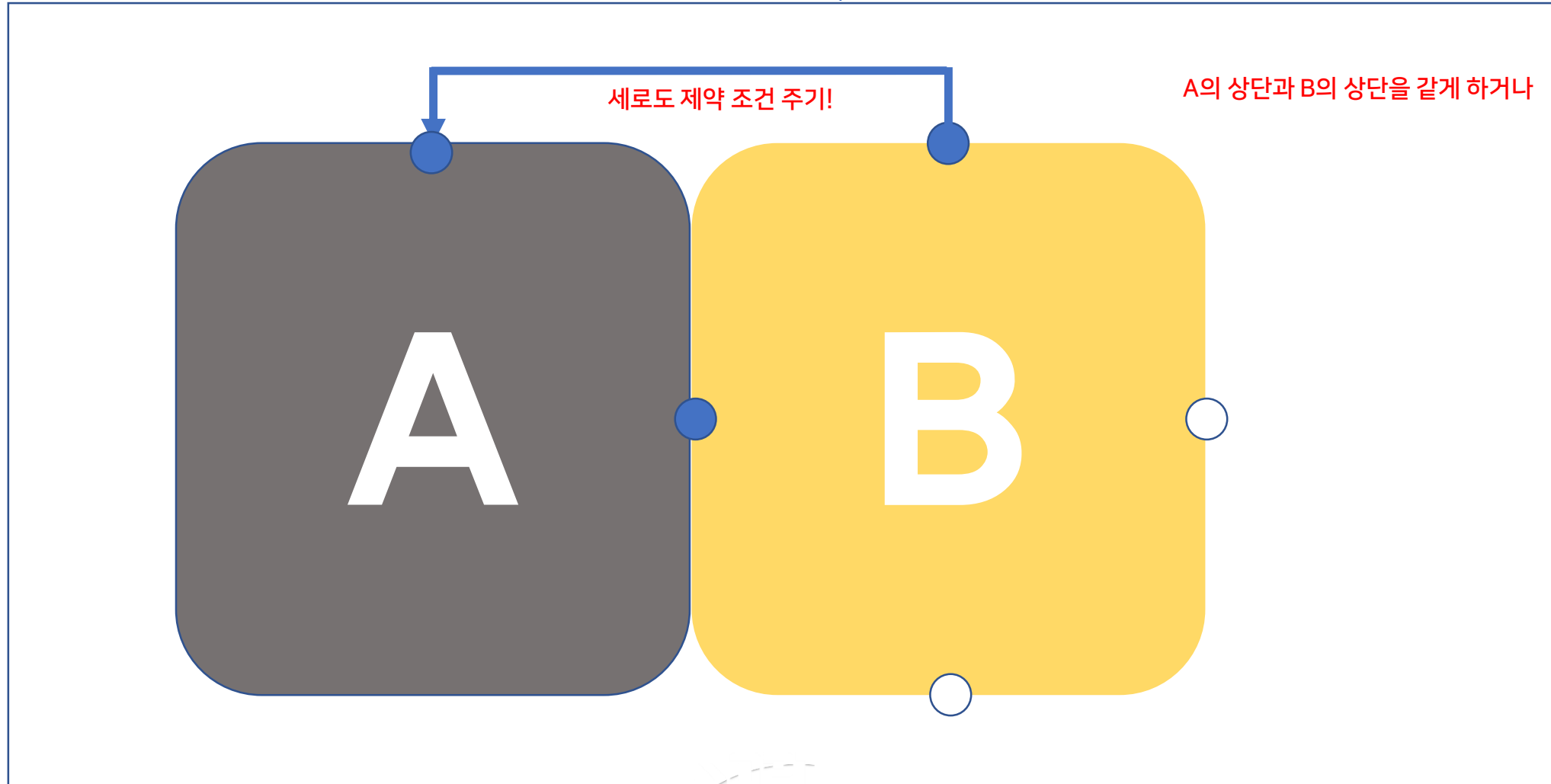
[Android Developer ConstraintLayout 공식 문서](#)

deSigner
develop
Oper
Planner
sopT

상대적 제약조건 걸기.

가로와 세로 모두 제약조건이 걸려 있어야 한다!

CosntraintLayout





2. ConstraintLayout

[ConstraintLayout 문서](#)

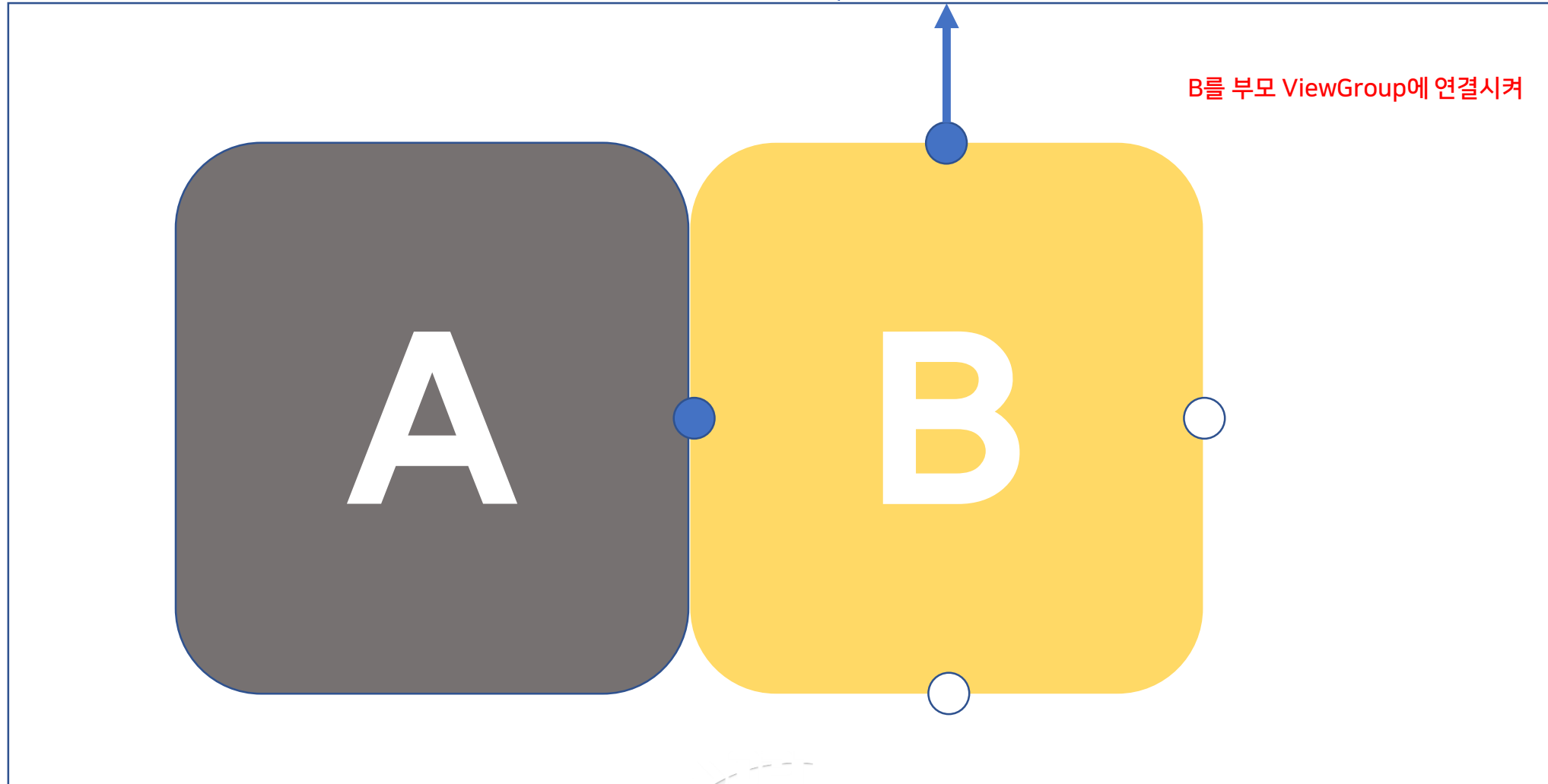
[Android Developer ConstraintLayout 공식 문서](#)

deSigner
develop
Oper
Planner
sopT

상대적 제약조건 걸기.

가로와 세로 모두 제약조건이 걸려 있어야 한다!

CosntraintLayout





2. ConstraintLayout

[ConstraintLayout 문서](#)

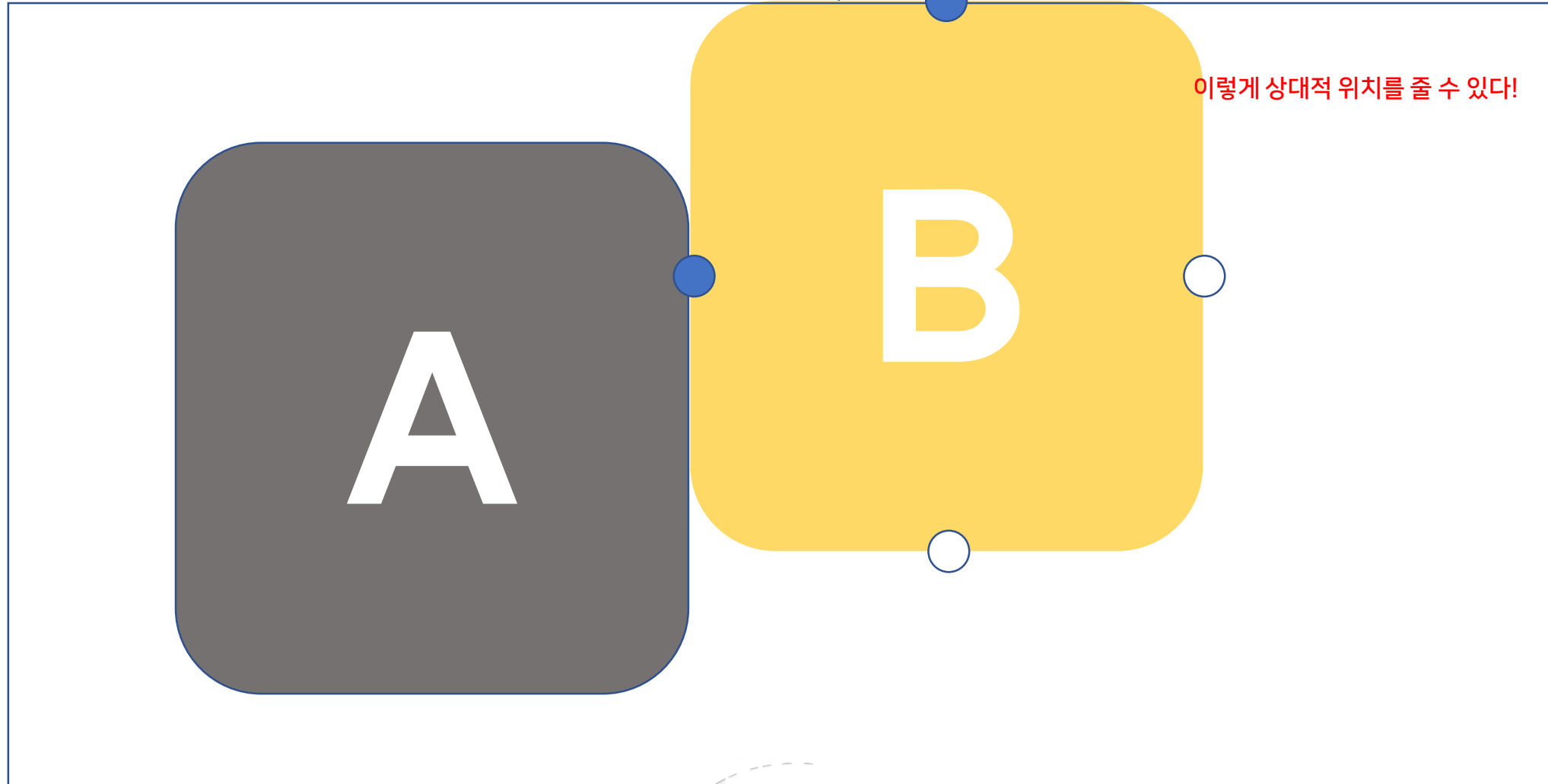
[Android Developer ConstraintLayout 공식 문서](#)

deSigner
develop
Oper
Planner
sopT

상대적 제약조건 걸기.

가로와 세로 모두 제약조건이 걸려 있어야 한다!

CosntraintLayout





2. ConstraintLayout

[ConstraintLayout 문서](#)

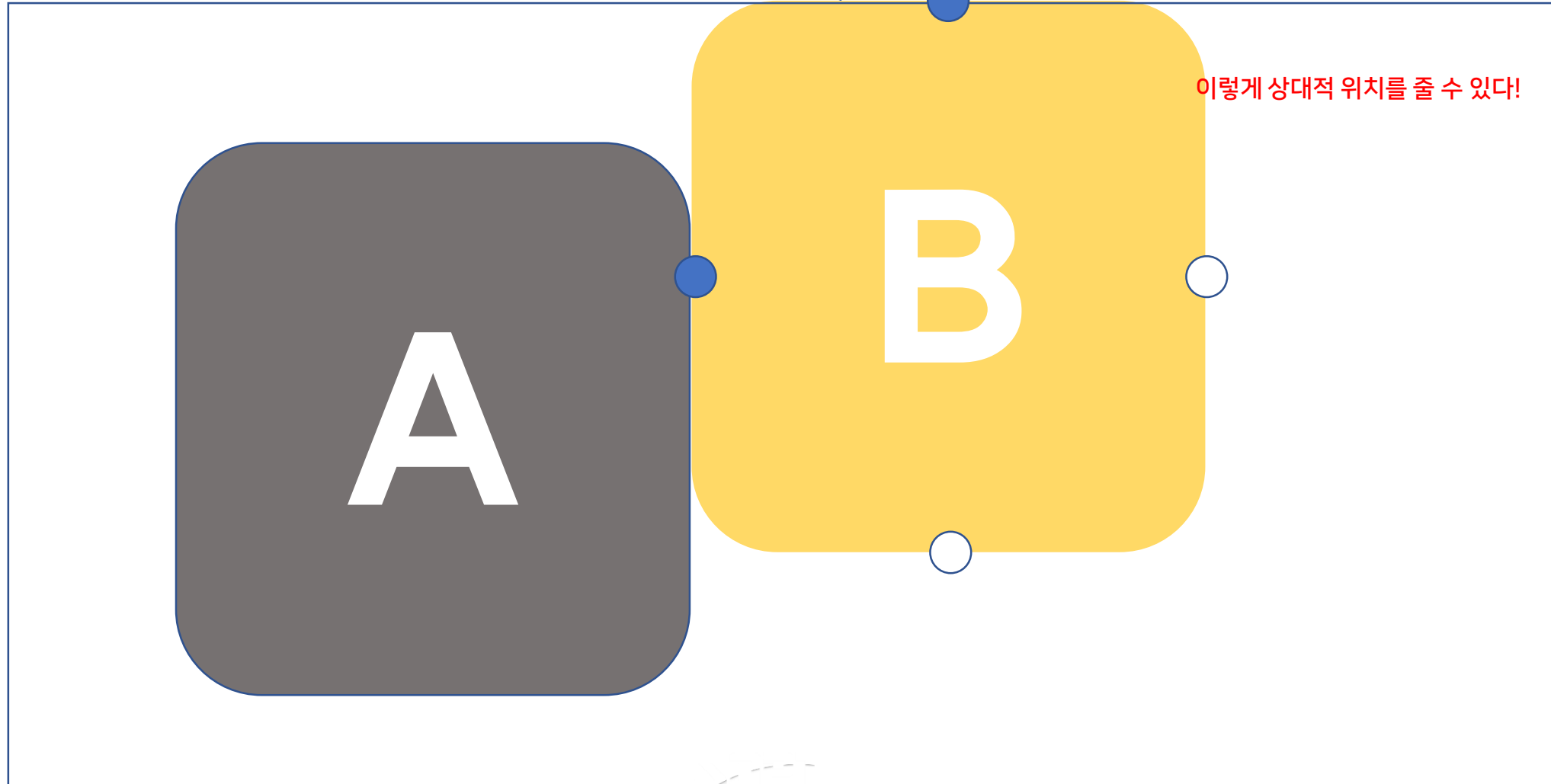
[Android Developer ConstraintLayout 공식 문서](#)

deSigner
develop
Oper
Planner
sopT

상대적 제약조건 걸기.

가로와 세로 모두 제약조건이 걸려 있어야 한다!

CosntraintLayout





2. ConstraintLayout

[ConstraintLayout 문서](#)

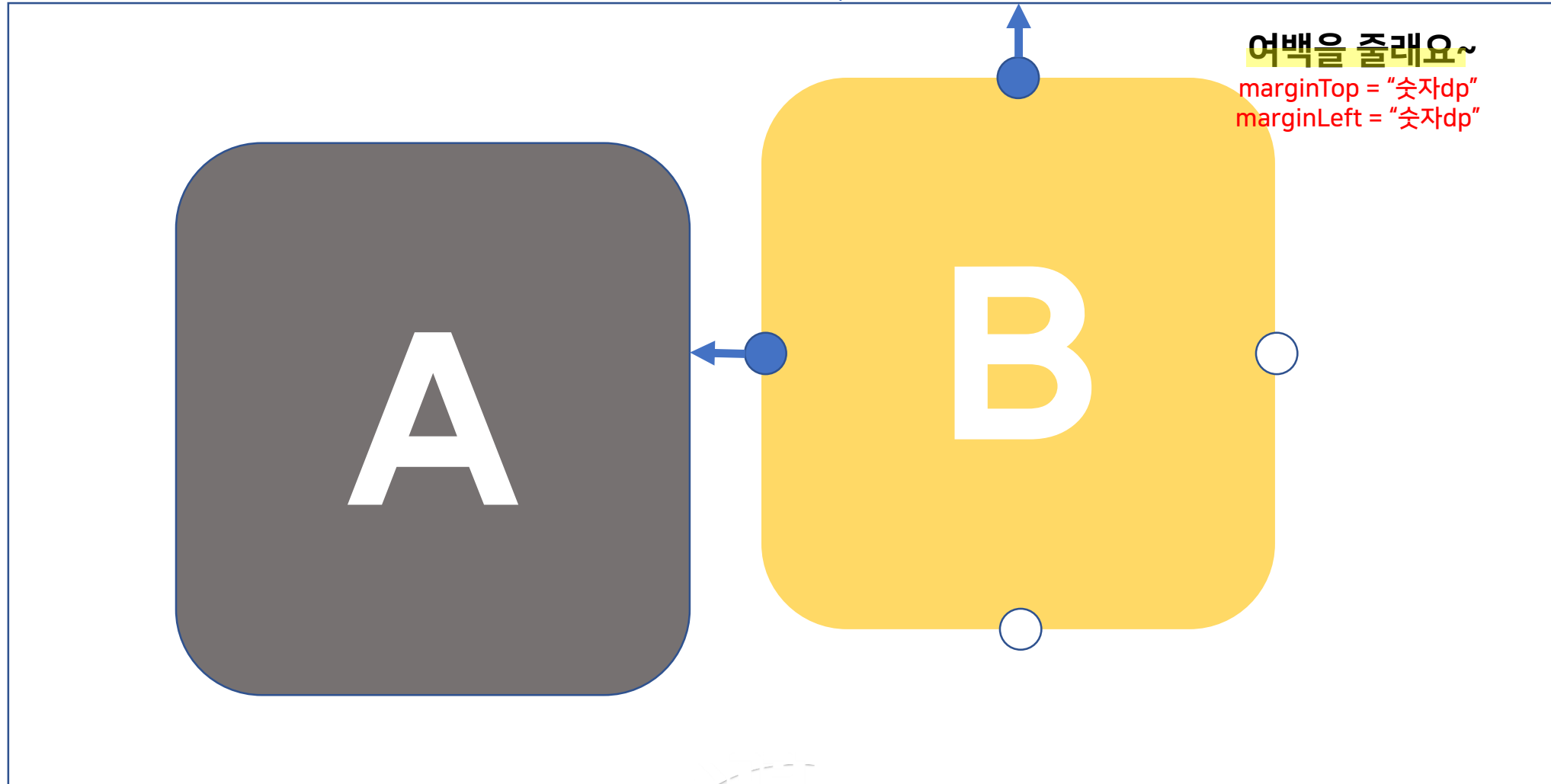
[Android Developer ConstraintLayout 공식 문서](#)

deSigner
develop
Planner
sopT

상대적 제약조건 걸기.

가로와 세로 모두 제약조건이 걸려 있어야 한다!

CosntraintLayout





2. ConstraintLayout

[ConstraintLayout 문서](#)

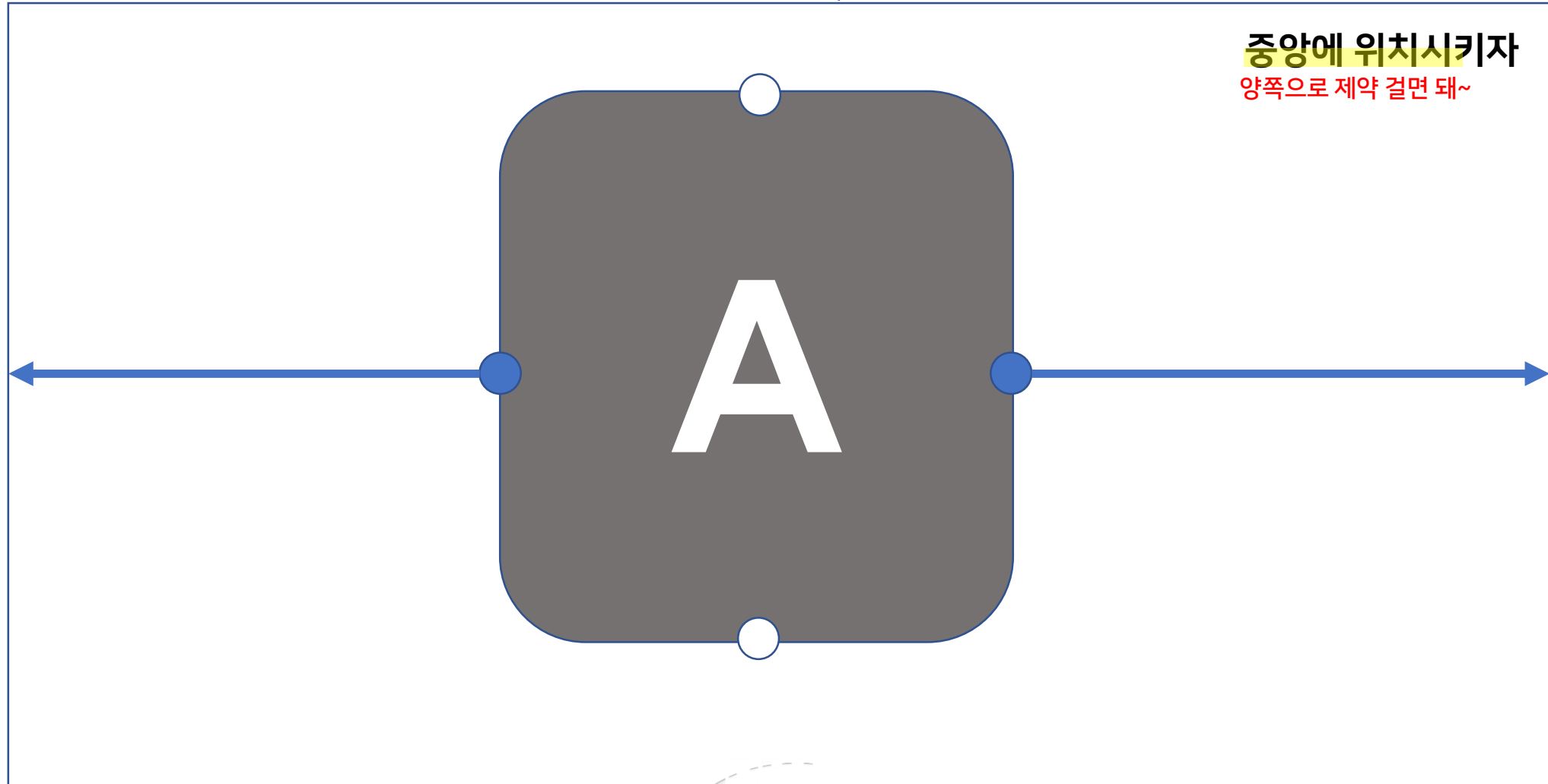
[Android Developer ConstraintLayout 공식 문서](#)

deSigner
develop
Oper
Planner
sopT

상대적 제약조건 걸기.

가로와 세로 모두 제약조건이 걸려 있어야 한다!

CosntraintLayout





2. ConstraintLayout

[ConstraintLayout 문서](#)

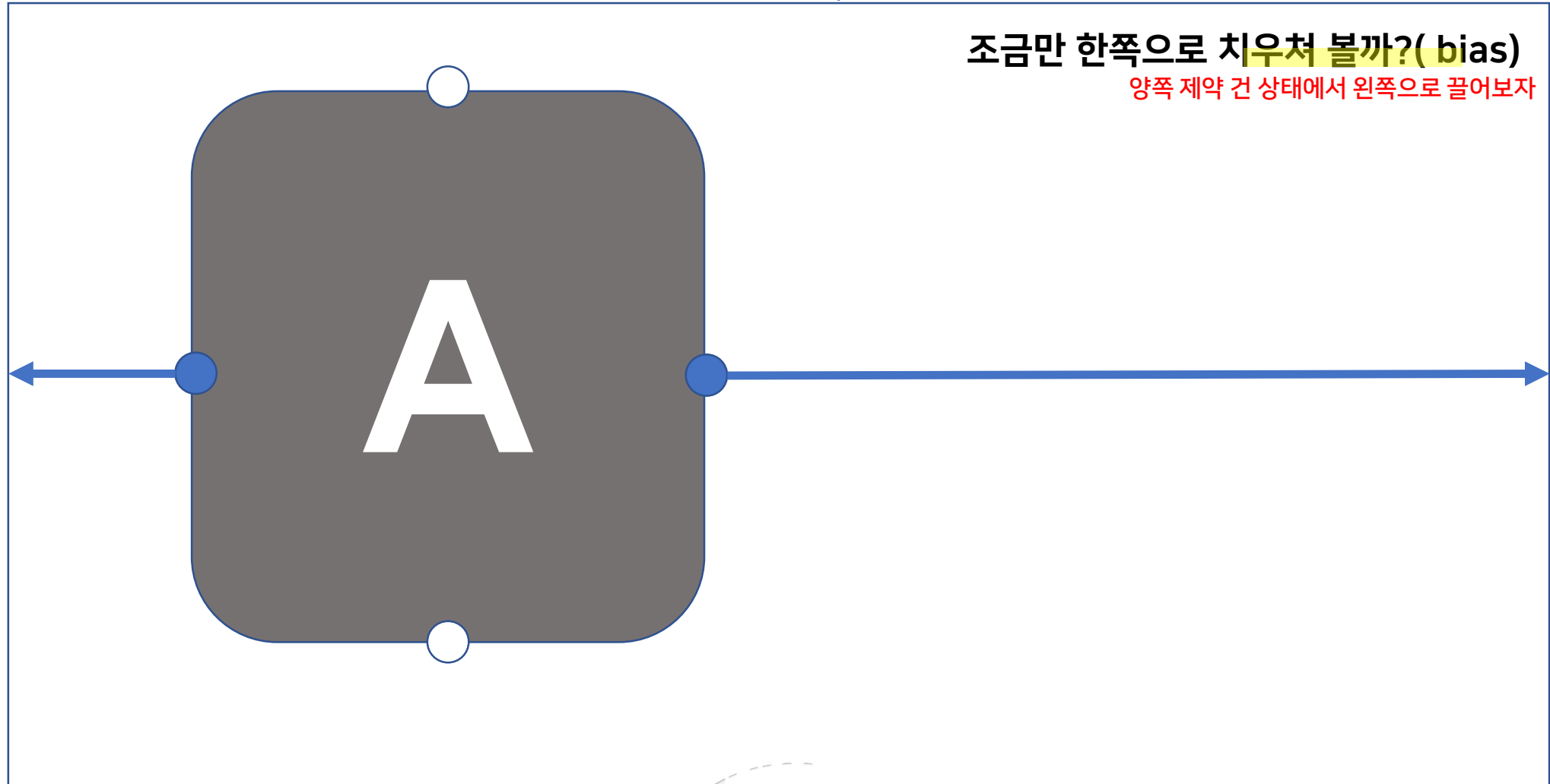
[Android Developer ConstraintLayout 공식 문서](#)

deSigner
develop
Oper
Planner
sopT

상대적 제약조건 걸기.

가로와 세로 모두 제약조건이 걸려 있어야 한다!

CosntraintLayout





2. ConstraintLayout

[ConstraintLayout 문서](#)

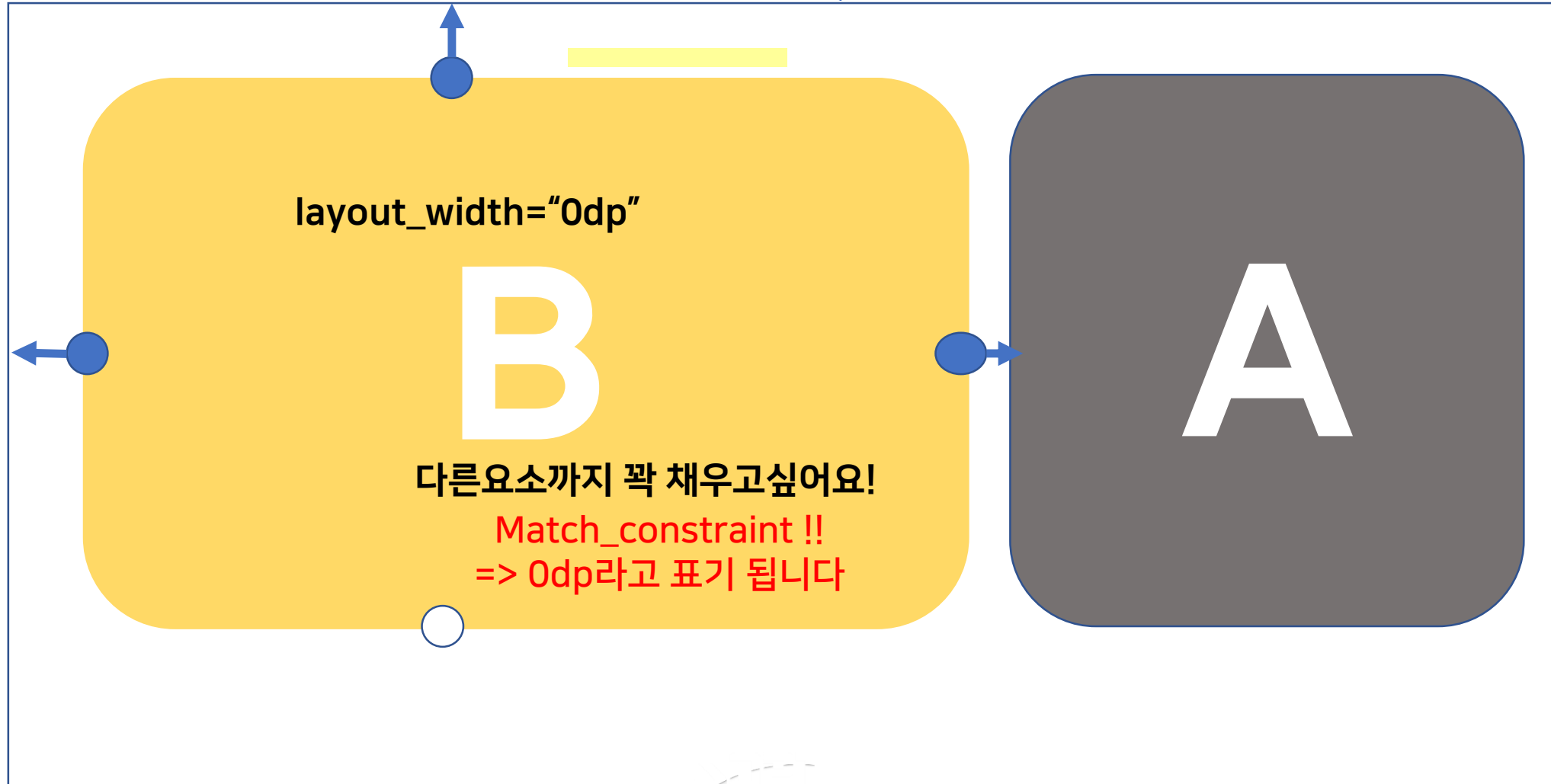
[Android Developer ConstraintLayout 공식 문서](#)

deSigner
develop
Oper
Planner
sopT

상대적 제약조건 걸기.

가로와 세로 모두 제약조건이 걸려 있어야 한다!

CosntraintLayout





2. ConstraintLayout

[ConstraintLayout 문서](#)

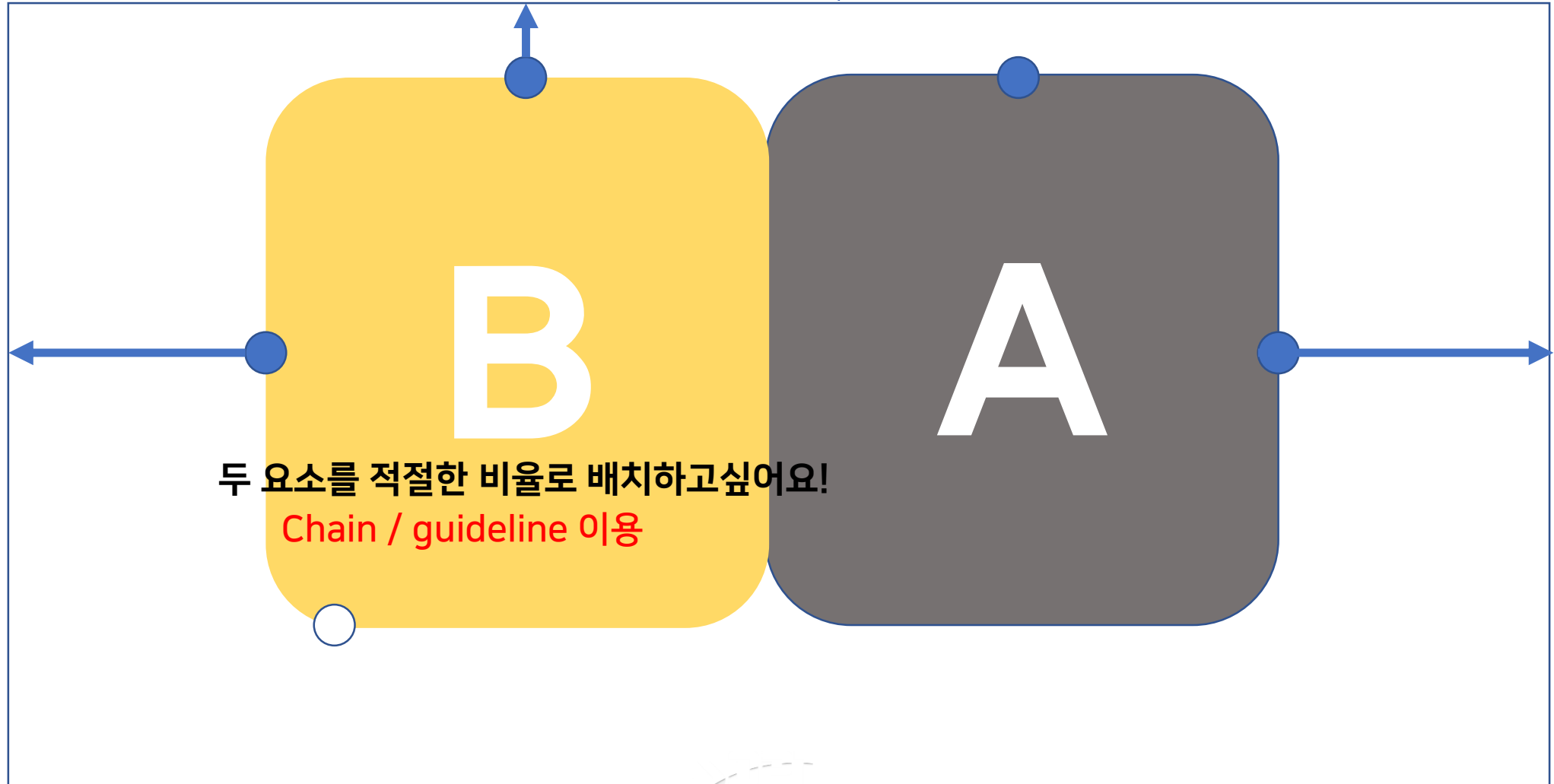
[Android Developer ConstraintLayout 공식 문서](#)

deSigner
develop
Planner
sopT

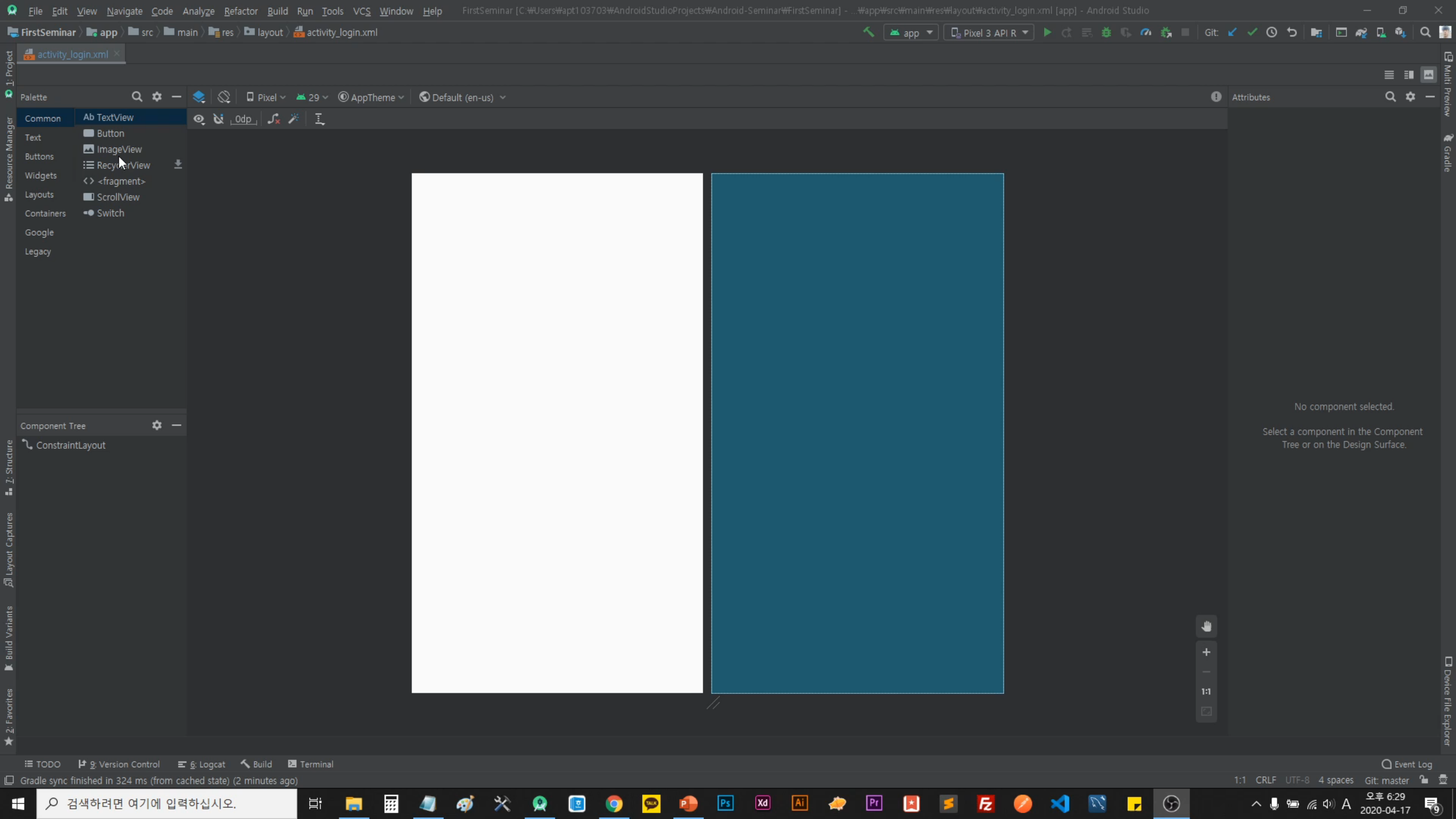
상대적 제약조건 걸기.

가로와 세로 모두 제약조건이 걸려 있어야 한다!

CosntraintLayout



두 요소를 적절한 비율로 배치하고싶어요!
Chain / guideline 이용





우선 기본적인 개념만 갖고 직접 해봅시다!

가로와 세로 모두 제약조건이 걸려 있어야 한다!



2. ConstraintLayout

[ConstraintLayout 문서](#)

[Android Developer ConstraintLayout 공식 문서](#)

가로와 세로 모두 제약조건이 걸려 있어야 한다!

deSigner
develOper
Planner
sopT



뭐 만들지 한번 볼까?

1. 로그인, 회원가입 화면 완성하기

#	description
1	로그인 화면 이미지
2	아이디 및 비밀번호 입력란
3	로그인 버튼
4	회원가입 버튼

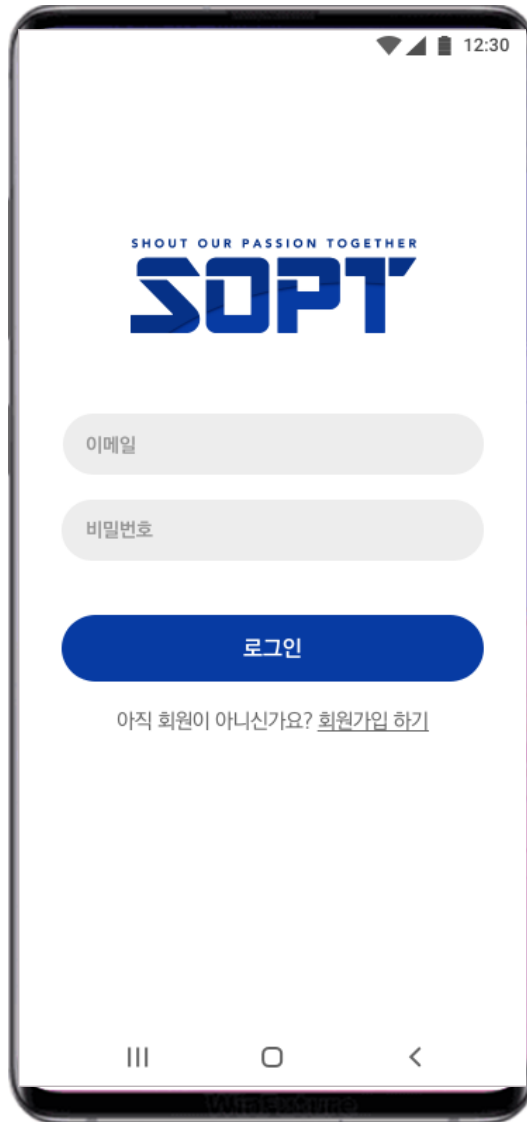


2. ConstraintLayout

[ConstraintLayout 문서](#)

[Android Developer ConstraintLayout 공식 문서](#)

deSigner
develOper
Planner
sopT



요소는 이렇게 배치해두면 되겠다..!



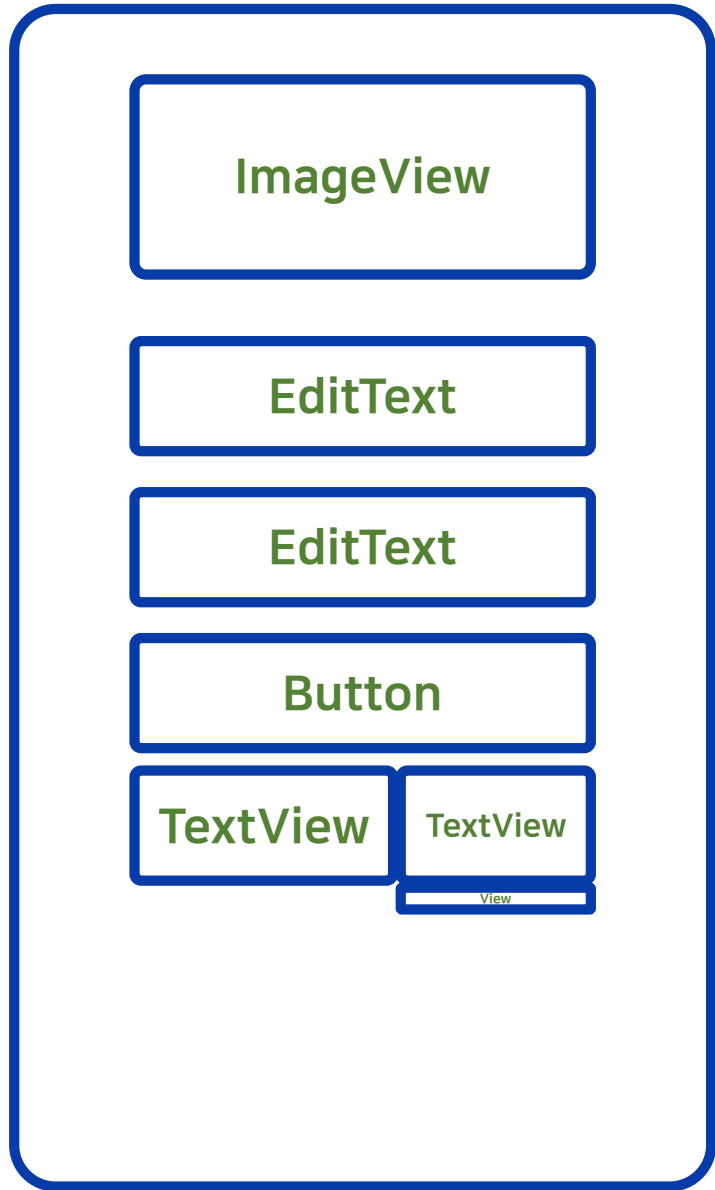


2. ConstraintLayout

[ConstraintLayout 문서](#)

[Android Developer ConstraintLayout 공식 문서](#)

deSigner
develOper
Planner
sopT



직접 뷰를 만들어보세요!

모르는게 생기면 팀원에게 물어보거나
공식문서를 참조하시면 됩니다!

3. LinearLayout



LinearLayout?

이름 그대로 선형으로 배치하는 Layout

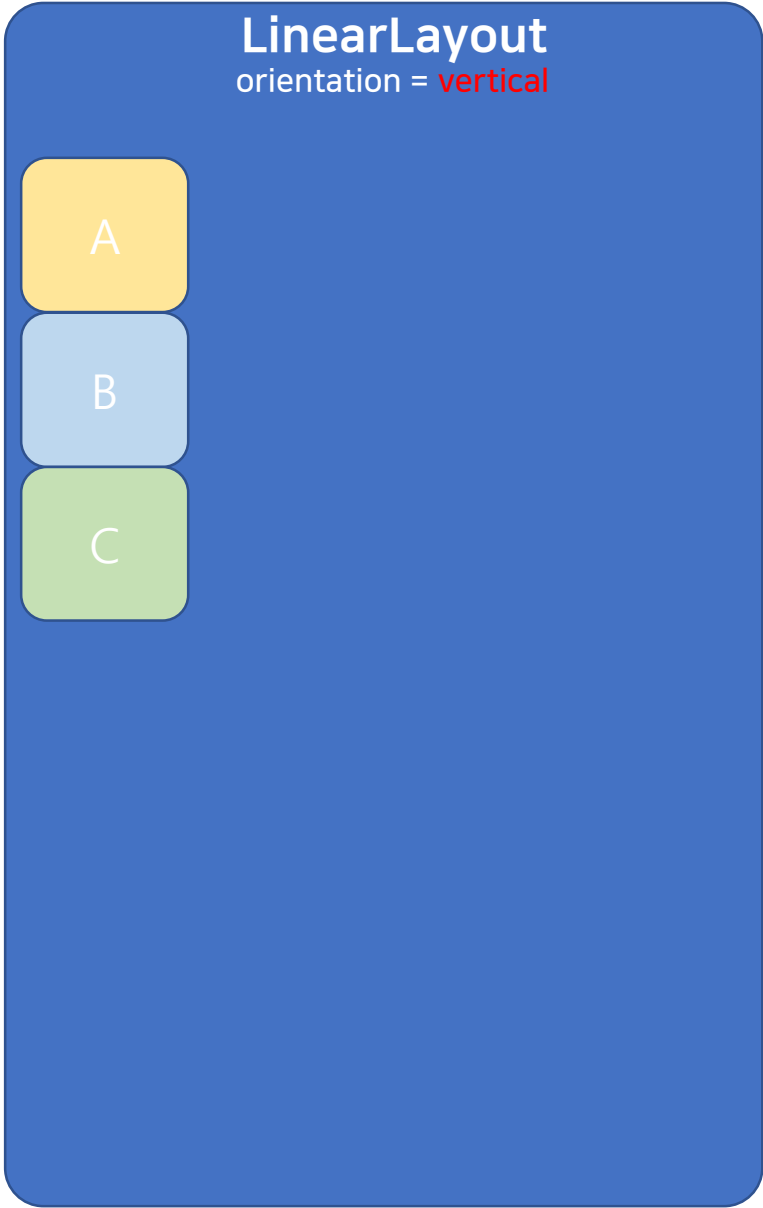
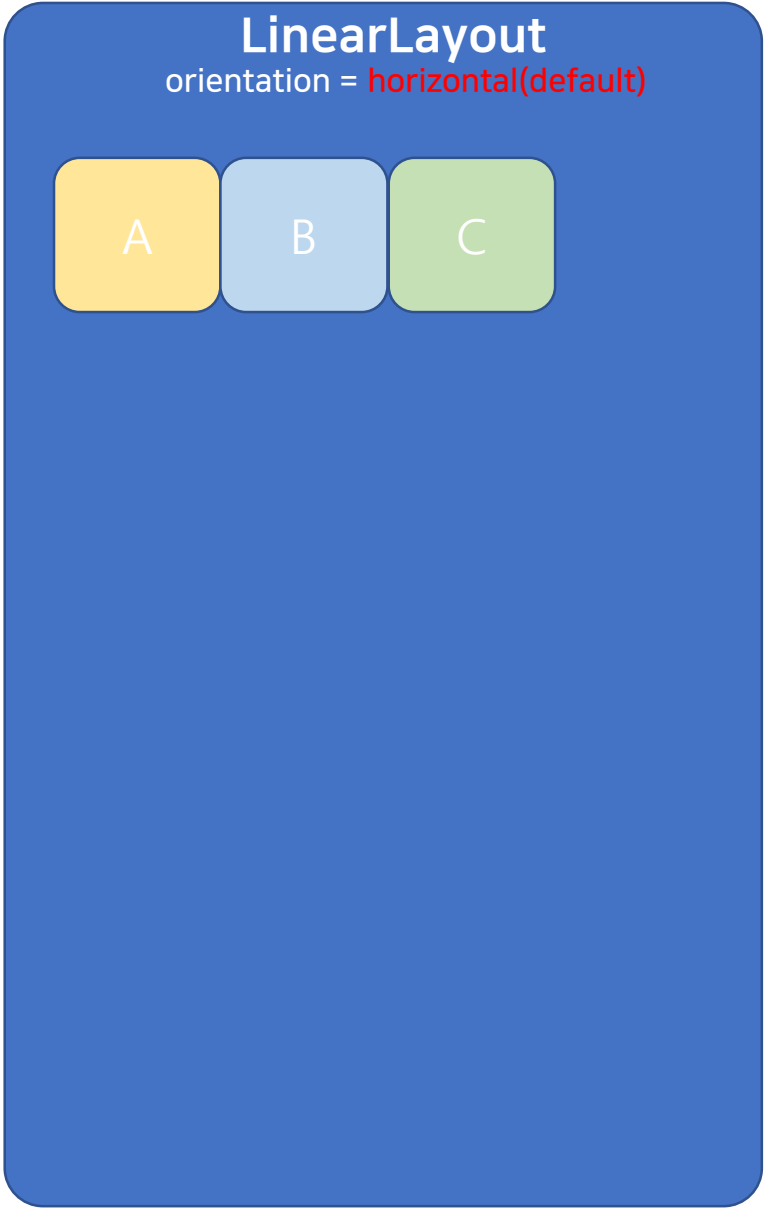




3. LinearLayout

공식 문서

deS
develop
Oper
Planner
sopT





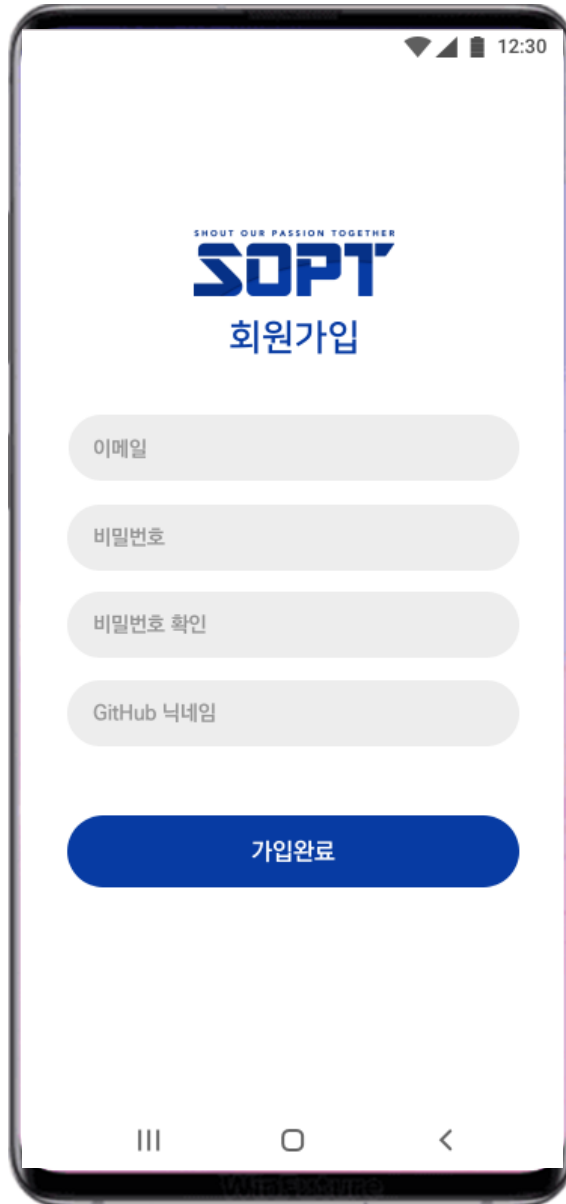
LinearLayout으로 회원가입 뷰를 짜보자

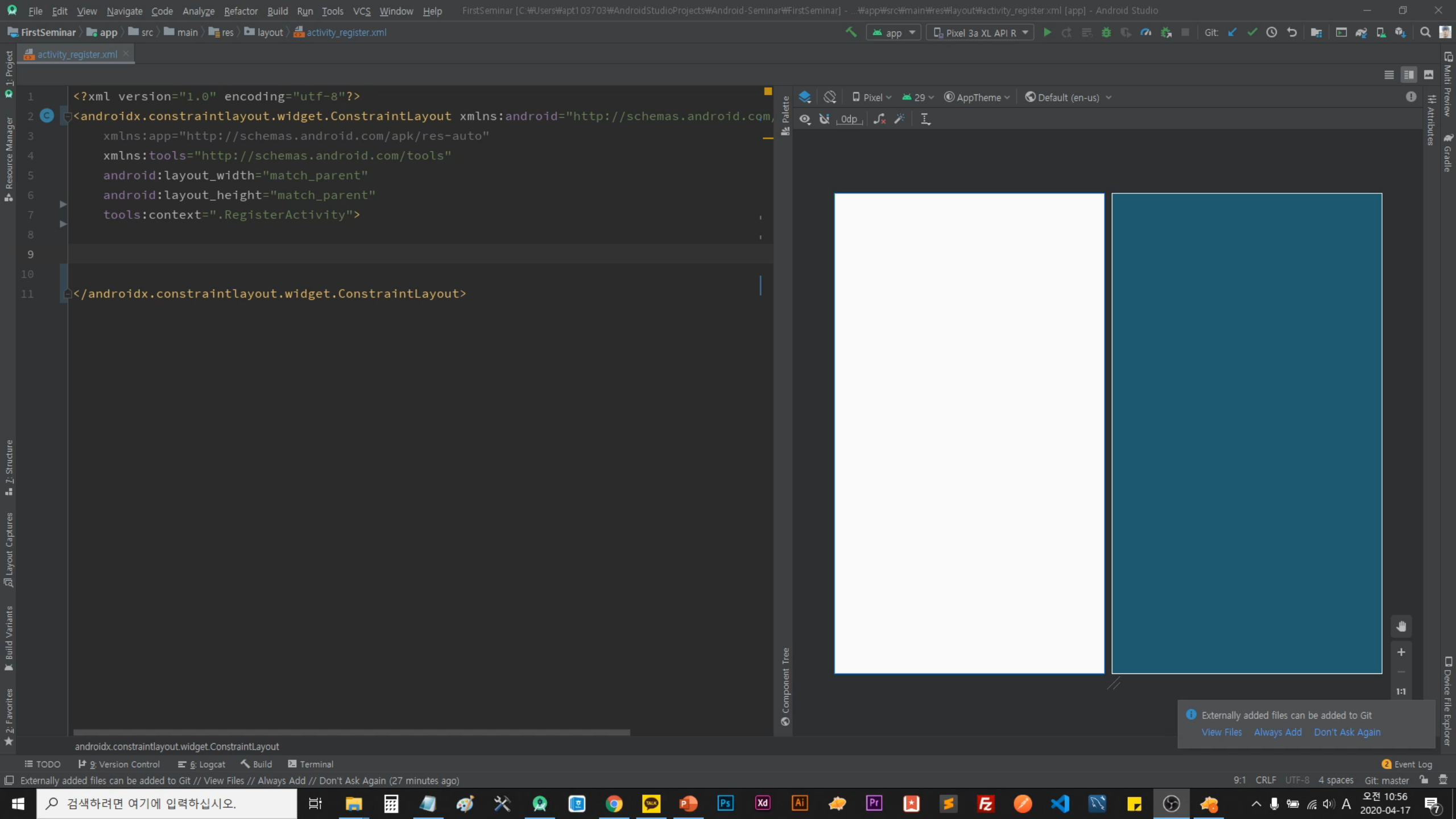


3. LinearLayout

공식 문서

deSigner
develop
Oper
Planner
sopT







3. LinearLayout

공식 문서

deSigner
develOper
Planner
sopT



직접 뷰를 만들어보세요!

모르는게 생기면 팀원에게 물어보거나
공식문서를 참조하시면 됩니다!



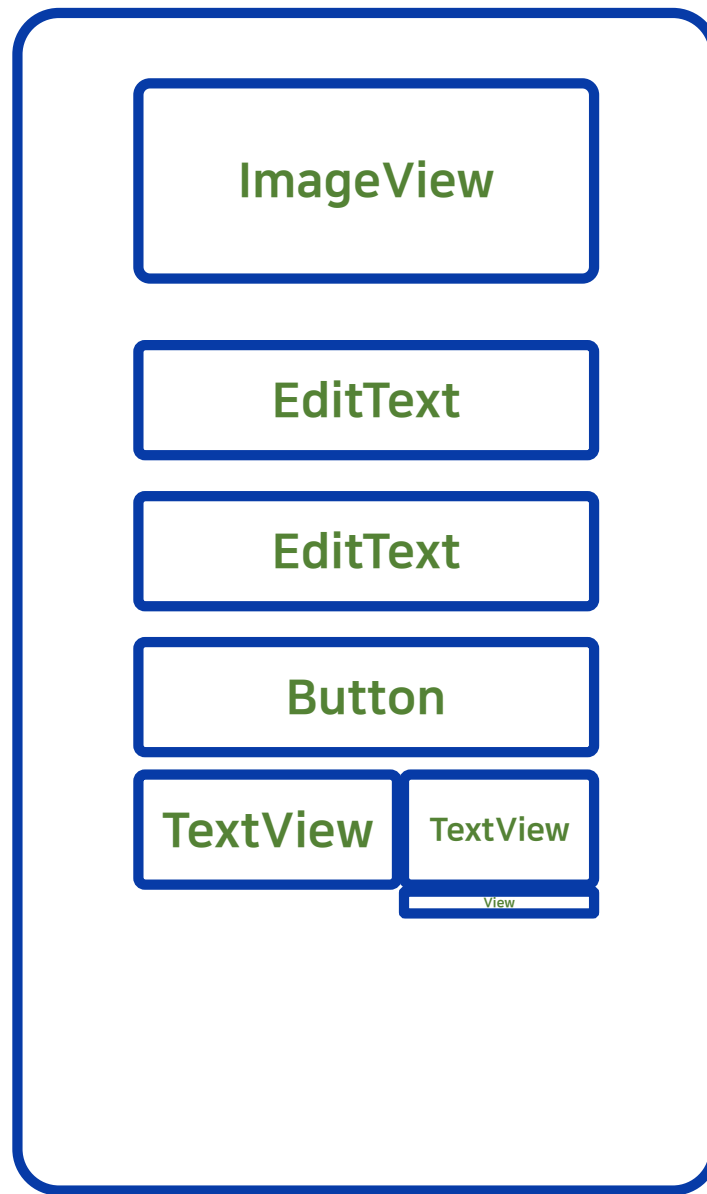
3. LinearLayout

공식 문서

<참고>LinearLayout으로 로그인 뷰 어떻게?

vertical

designer
developer
Planner
sopT



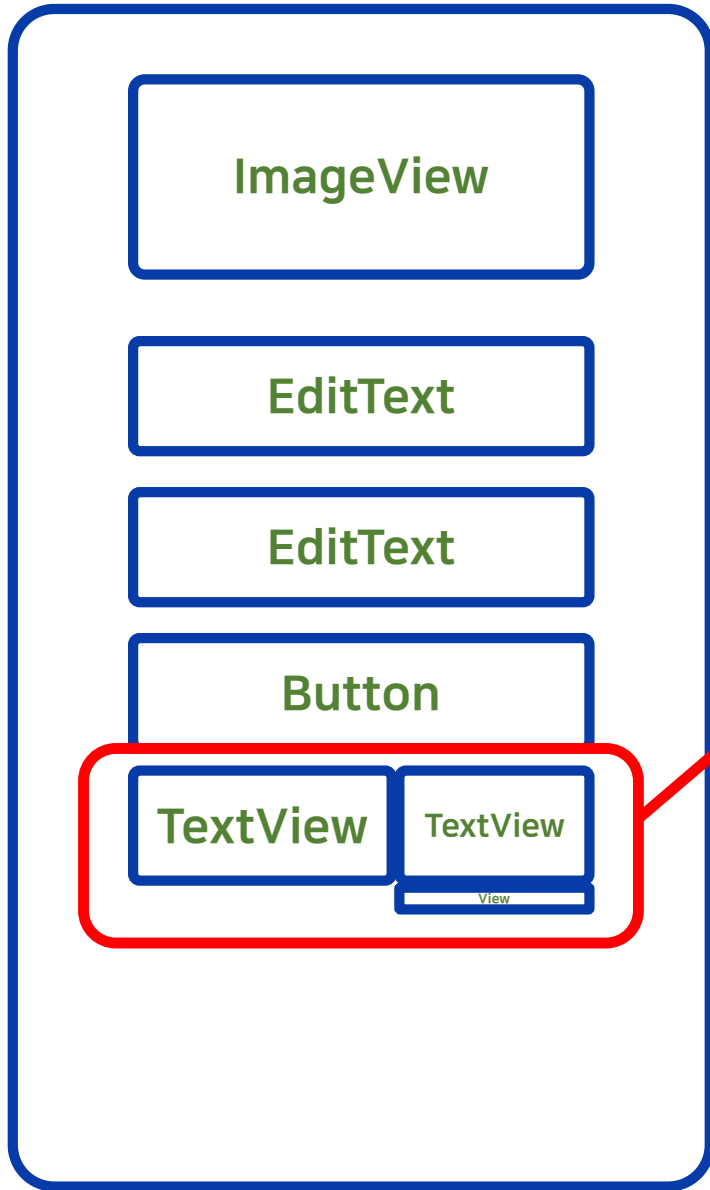


3. LinearLayout

공식 문서

<참고>LinearLayout으로 로그인 뷰 어떻게?

designer
developer
Planner
sopT



전체적으로 Vertical이네!

여긴 가로로 Linear.. 또 세로로 Linear가 있네?

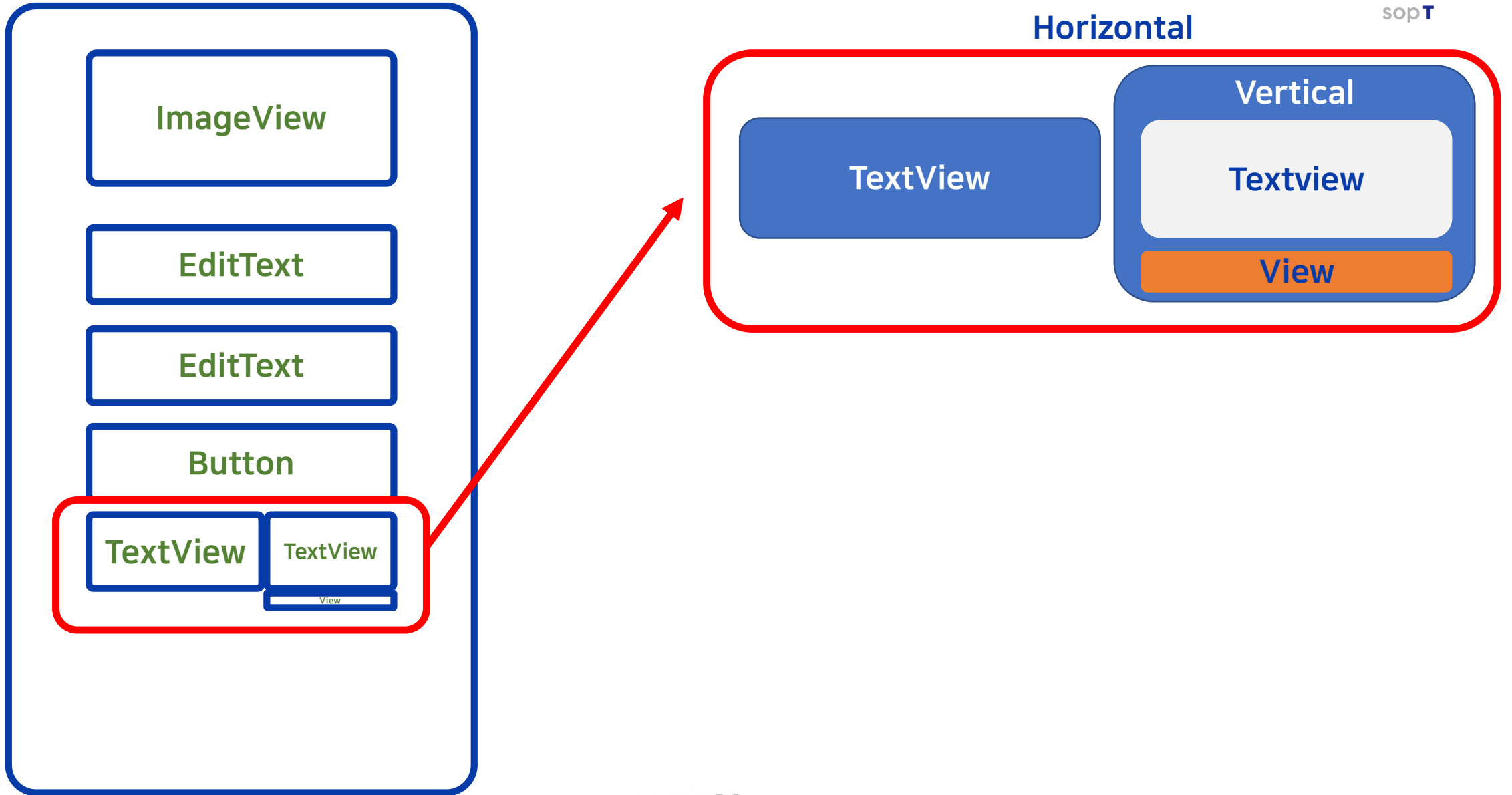


3. LinearLayout

공식 문서

<참고>LinearLayout으로 로그인 뷰 어떻게?

designer
developer
Planner
sopT



4. RelativeLayout



RelativeLayout?

부모 - 자식

대상 - 자식

상대적 위치 적용.



4. RelativeLayout

공식문서

deS
develO
Planner
sopT



Android:layout_alignParentTop



Android:layout_alignParentBottom



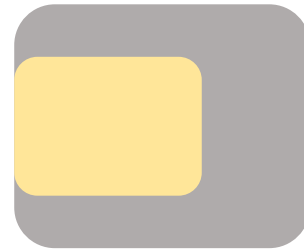
Android:layout_alignParentLeft



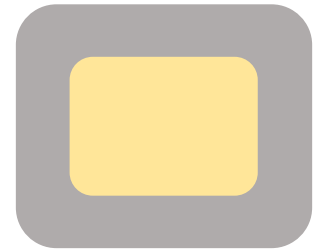
Android:layout_alignParentRight



Android:layout_centerHorizontal



Android:layout_centerVertical



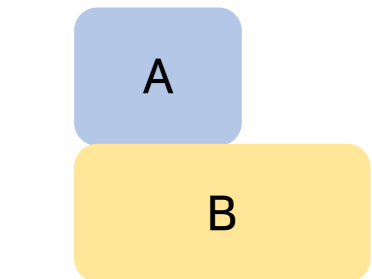
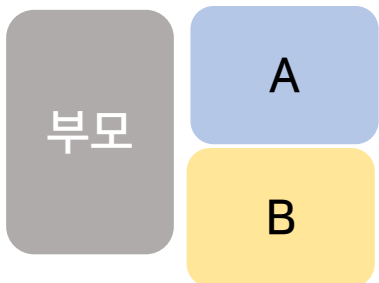
Android:layout_centerInParent



4. RelativeLayout

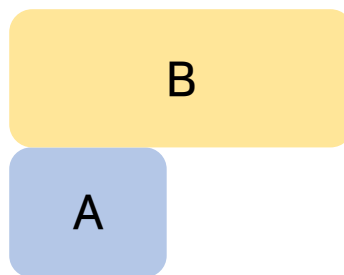
공식문서

designer
developer
Planner
sopT

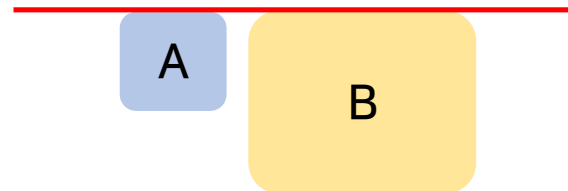


android:layout_above="B id"

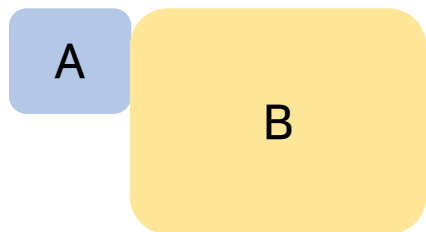
A 기준.



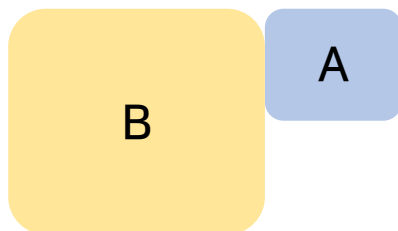
android:layout_below="B id"



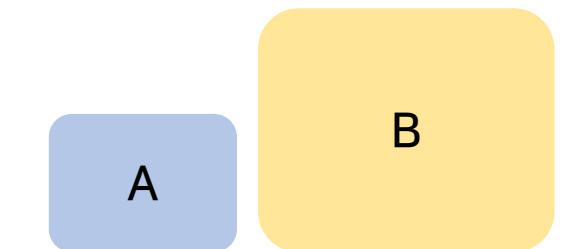
android:layout_alignTop="B id "



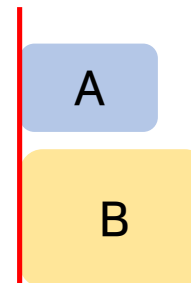
android:layout_toLeftOf="B id "



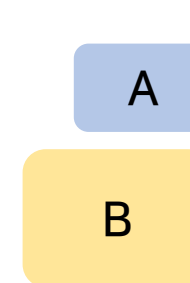
android:layout_toRightOf="B id"



android:layout_alignBottom="B id"



android:layout_alignLeft="B id"



android:layout_alignRight="B id"

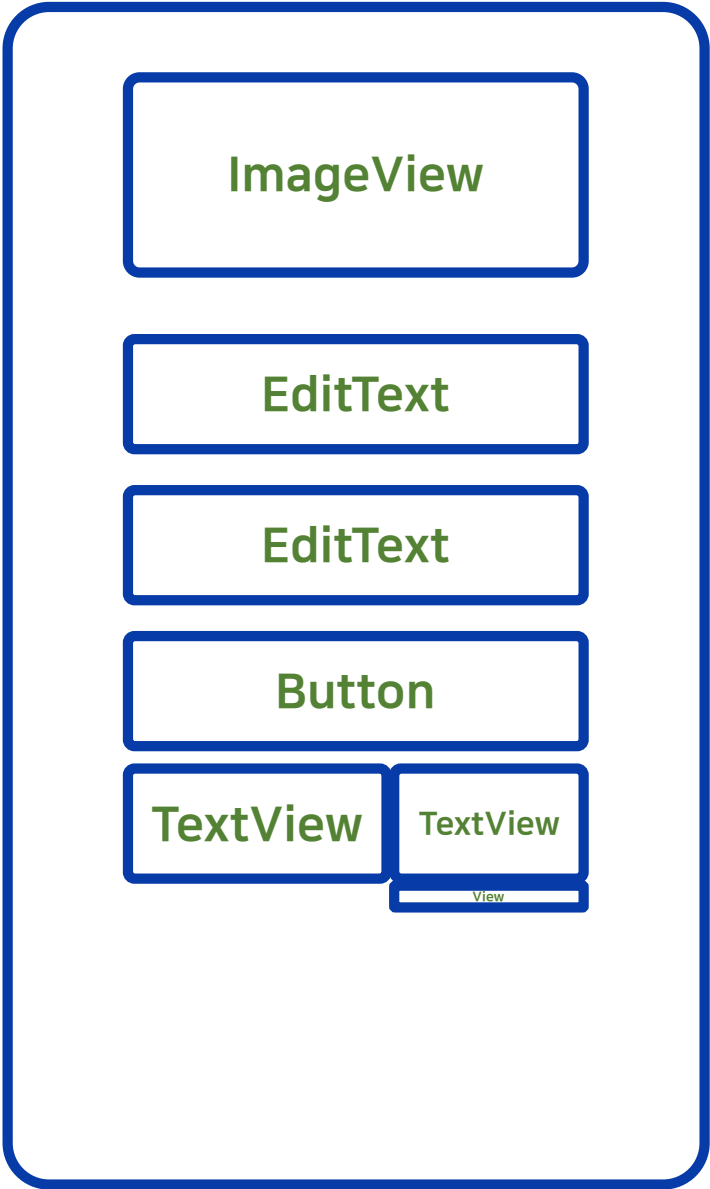


RelativeLayout으로 로그인 뷰를 짜보자

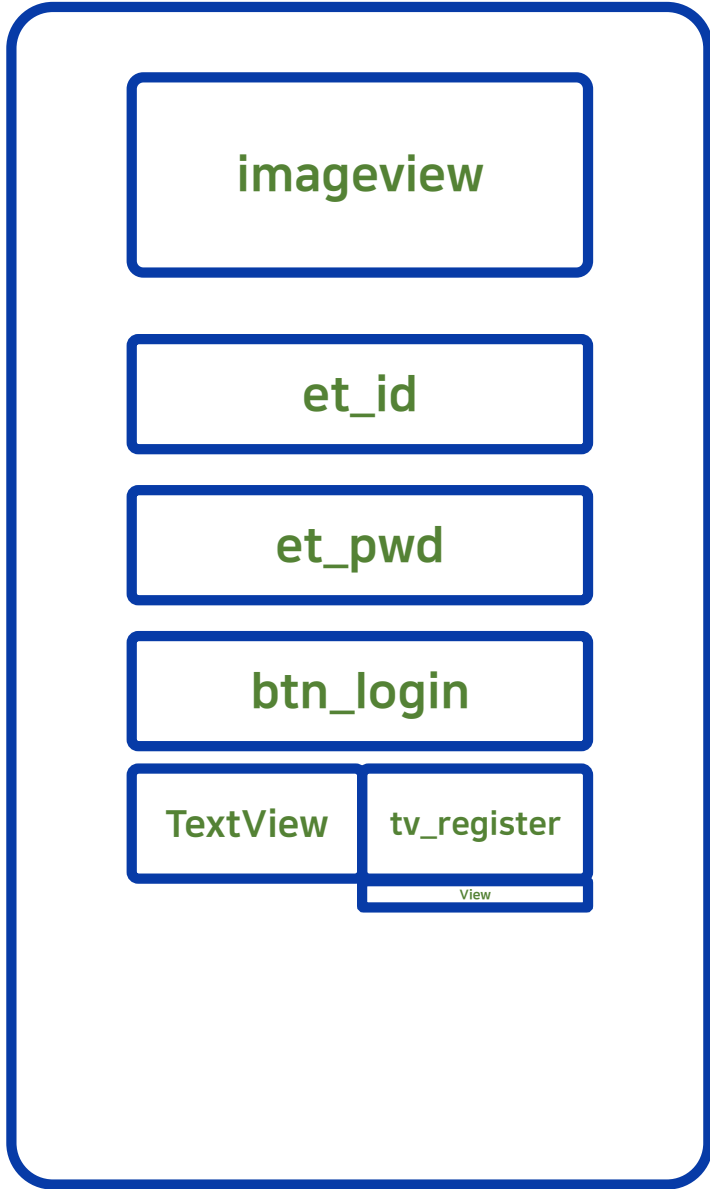
4. RelativeLayout

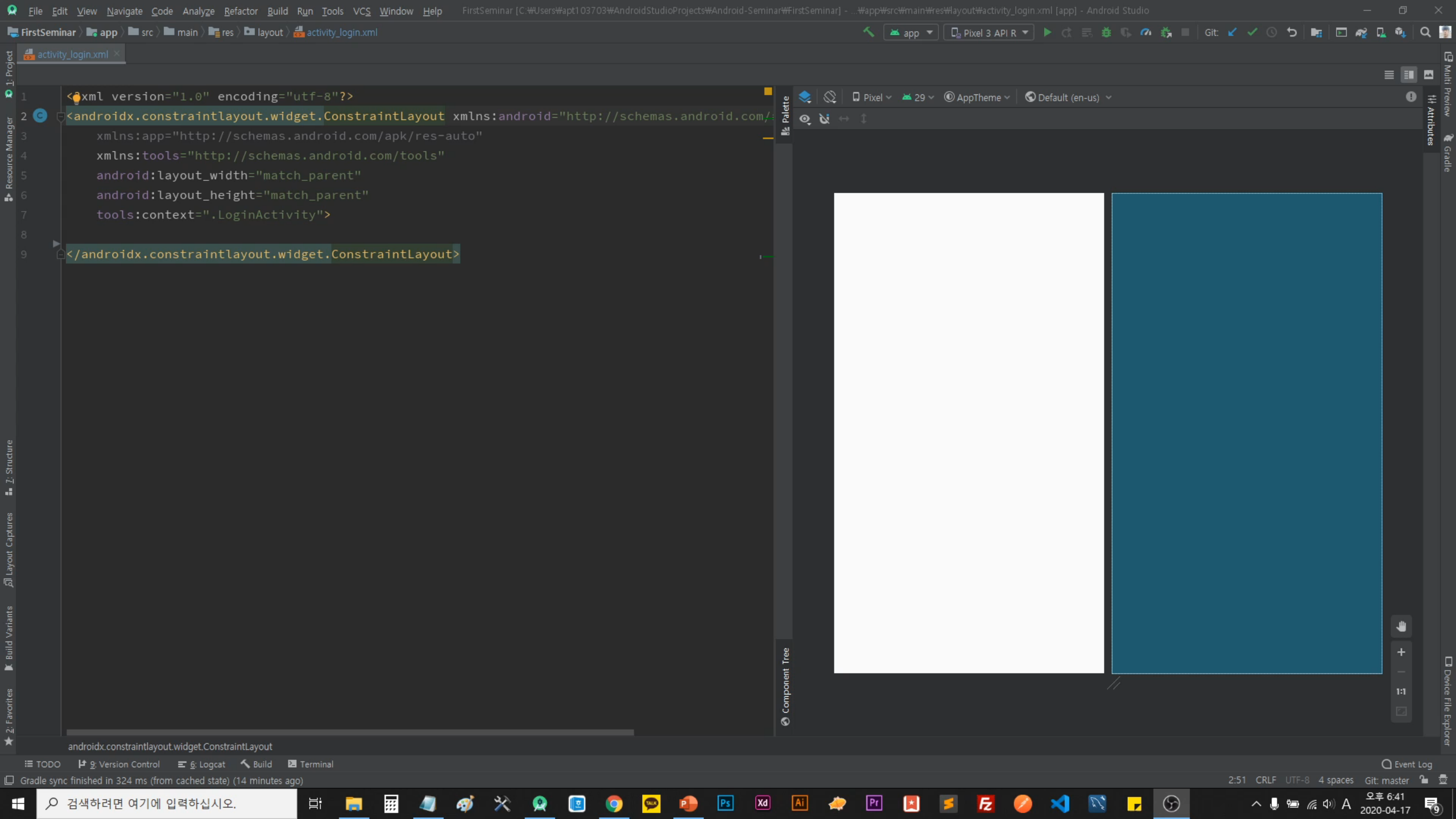
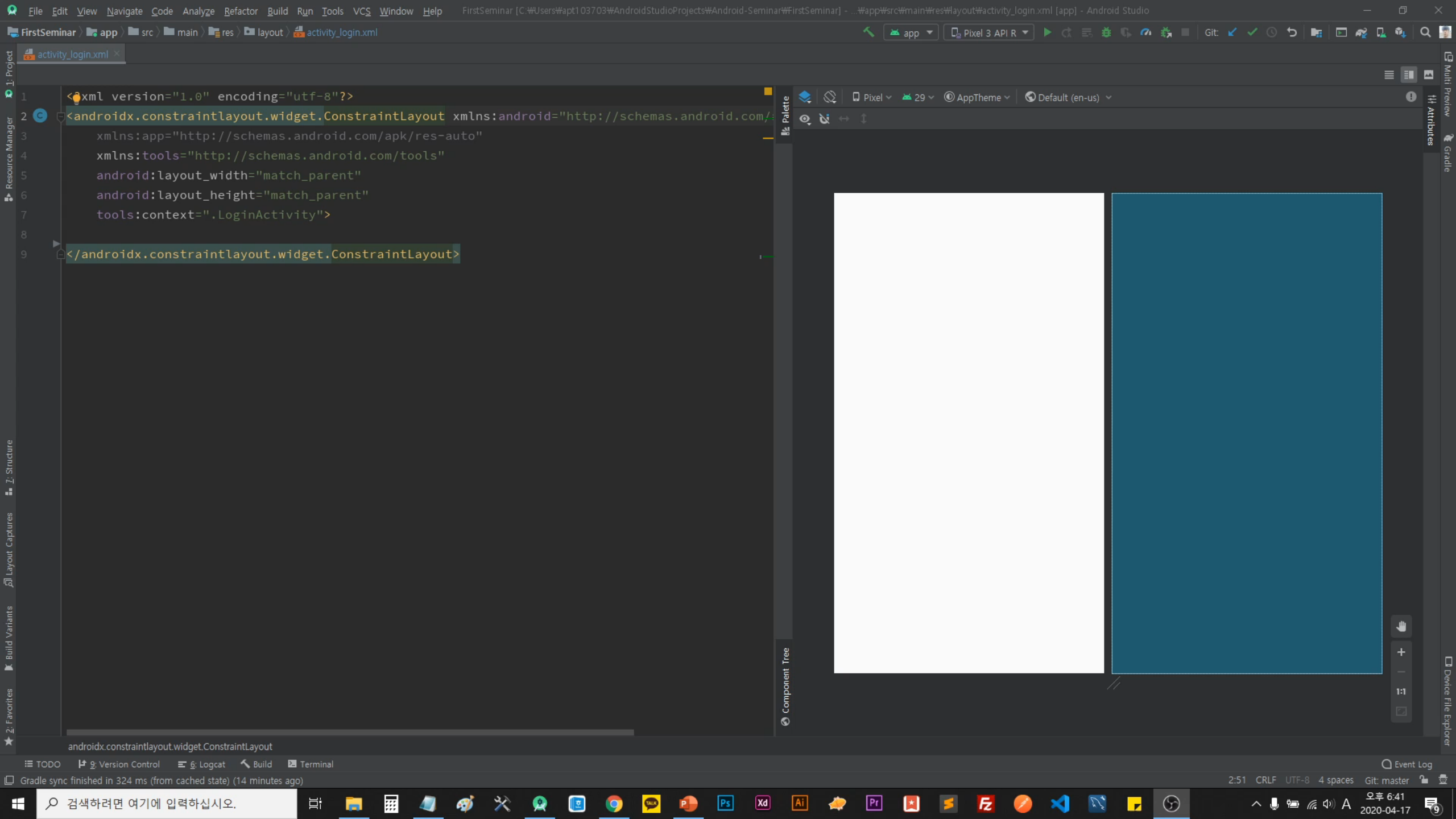
공식문서

designer
developer
Planner
sopT



Id값을 바꿔 주자





5. 기능을 입혀보자



5. 기능을 입혀보자

The screenshot shows the Android Studio interface. On the left, the 'Project' view displays the file structure of the 'app' module. The 'org.sopt.project26th' package is expanded, and the 'LoginActivity' file is selected, highlighted with a red rectangle. On the right, the 'Code' view shows the Java code for 'LoginActivity'. The 'onCreate' method is highlighted with a red rectangle. The code is as follows:

```
1 package org.sopt.project26th
2
3 import ...
4
5
6 class LoginActivity : AppCompatActivity() {
7
8     override fun onCreate(savedInstanceState: Bundle?) {
9         super.onCreate(savedInstanceState)
10        setContentView(R.layout.activity_login)
11    }
12 }
13
```

5. 기능을 입혀보자

잠깐.. onCreate()와 더불어 알아둬야 할 게 있어!

액티비티 생명주기

onCreate - 액티비티 생성시 처음 호출되는 부분.

onStart - 화면이 처음으로 사용자에게 보여질 때

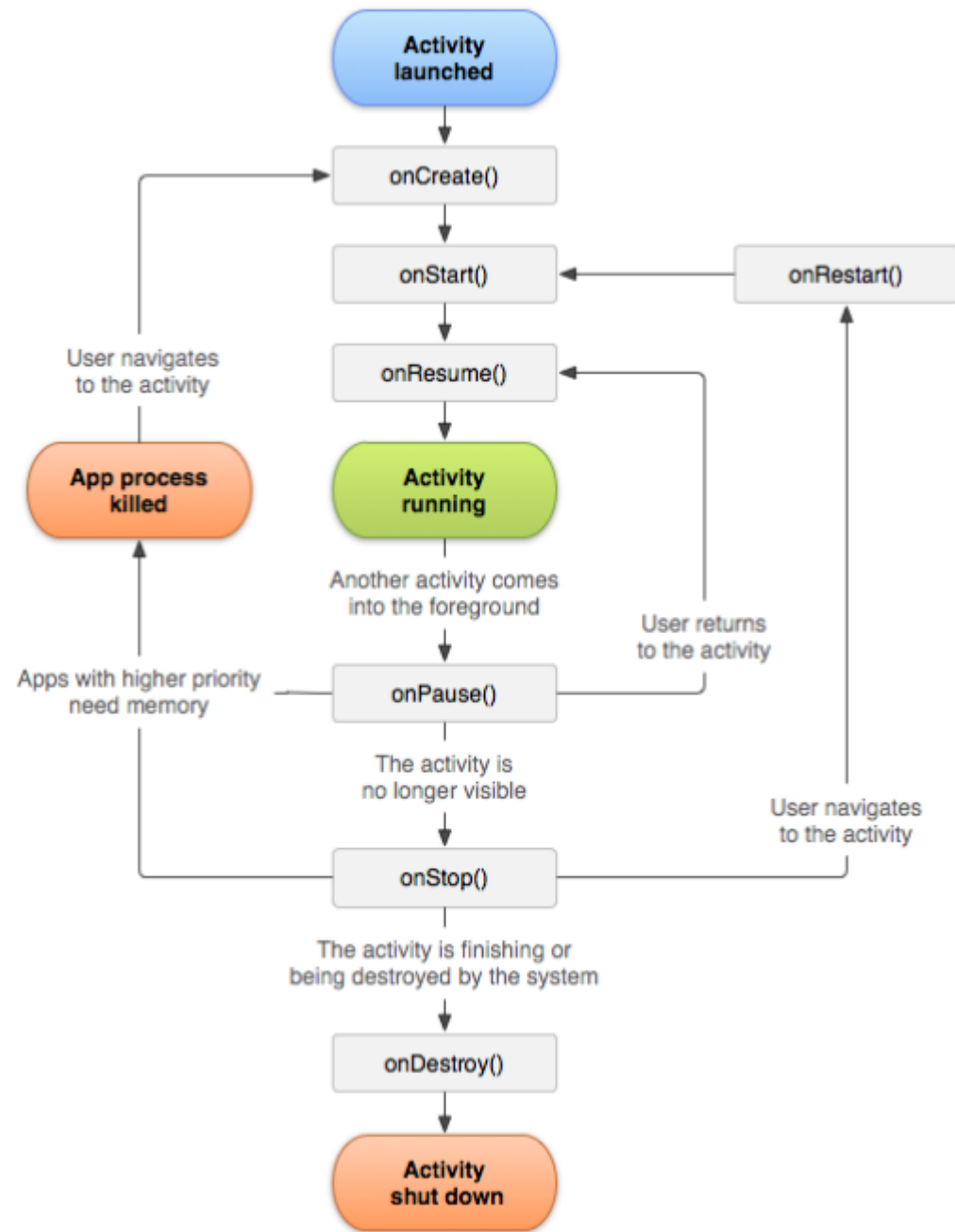
onResume - 화면이 맨 앞에 보여질 때

onPause - 사용자가 화면을 떠날 조짐이 보일 때

onStop - 화면이 더 이상 보이지 않을 때

onDestroy - 화면이 완전히 끝나기 전

공식문서



5. 기능을 입혀보자



1. 로그인, 회원가입 화면 완성하기

#	description
1	아이디 - 한 줄 입력 제한 비밀번호 - ***표시로 입력되도록 하기
2	아이디/비밀번호 하나라도 비어있다면 Toast 띄우기 둘 다 입력된 경우 MainActivity로 화면전환
3	회원가입 창 띄우기

5. 기능을 입혀보자



1. 로그인, 회원가입 화면 완성하기

#	description
1	아이디 - 한 줄 입력 제한 비밀번호 - ***표시로 입력되도록 하기
2	아이디/비밀번호 하나라도 비어있다면 Toast 띄우기 둘 다 입력된 경우 MainActivity로 화면전환
3	회원가입 창 띄우기



5. 기능을 입혀보자

1. 로그인, 회원가입 화면 완성하기

#	description
1	아이디 - 한 줄 입력 제한 비밀번호 - ***표시로 입력되도록 하기
2	아이디/비밀번호 하나라도 비어있다면 Toast 띄우기 둘 다 입력된 경우 MainActivity로 화면전환
3	회원가입 창 띄우기

1. editText inputType 설정

2. `Toast.makeText(this,"메시지",Toast.LENGTH_SHORT).show()`

아이디/ 비밀번호를 확인하세요

3. `startActivity(Intent(this,MainActivity::class.java))`

화면전환

#	description
1	아이디 - 한 줄 입력 제한 비밀번호 - ***표시로 입력되도록 하기
2	아이디/비밀번호 하나라도 비어있다면 Toast 띄우기 둘 다 입력된 경우 MainActivity로 화면전환
3	회원가입 창 띄우기



5. 기능을 입혀보자

Kotlin Extention을 사용하면 findViewById를 하지 않아도 View에 접근이 가능하다!

1. 로그인, 회원가입 화면 완성하기

#	description
1	아이디 - 한 줄 입력 제한 비밀번호 - ***표시로 입력되도록 하기
2	아이디/비밀번호 하나라도 비어있다면 Toast 띄우기 둘 다 입력된 경우 MainActivity로 화면전환
3	회원가입 창 띄우기

```
class LoginActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_login)  
  
        val et_id = findViewById<EditText>(R.id.et_id)  
        val et_pwd = findViewById<EditText>(R.id.et_password)  
        val btn_login = findViewById<Button>(R.id.btn_login)  
        val tv_register = findViewById<TextView>(R.id.tv_register)  
  
        btn_login.setOnClickListener {  
            if(et_id.text.isNullOrEmpty()||et_pwd.text.isNullOrEmpty()){  
                Toast.makeText(this,"아이디와 비밀번호를 확인하세요",Toast.LENGTH_SHORT).show()  
            }else{  
                val intent = Intent(this,MainActivity::class.java)  
                startActivity(intent)  
            }  
        }  
        tv_register.setOnClickListener {  
            val intent = Intent(this,RegisterActivity::class.java)  
            startActivity(intent)  
        }  
    }  
}
```

5. 기능을 입혀보자

버튼 클릭 이벤트 : `setOnClickListener`

```
class LoginActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_login)  
  
        btn_login.setOnClickListener {  
            if(et_id.text.isNullOrEmpty() || et_password.text.isNullOrEmpty()) {  
                Toast.makeText(this, "아이디와 비밀번호를 확인하세요", Toast.LENGTH_SHORT).show()  
            } else {  
                val intent = Intent(this, MainActivity::class.java)  
                startActivity(intent)  
            }  
        }  
  
        tv_register.setOnClickListener {  
            val intent = Intent(this, RegisterActivity::class.java)  
            startActivity(intent)  
        }  
    }  
}
```

1. 로그인, 회원가입 화면 완성하기

#	description
1	아이디 - 한 줄 입력 제한 비밀번호 - ***표시로 입력되도록 하기
2	아이디/비밀번호 하나라도 비어있다면 Toast 띄우기 둘 다 입력된 경우 MainActivity로 화면전환
3	회원가입 창 띄우기

5. 기능을 입혀보자



1. 로그인, 회원가입 화면 완성하기

#	description
1	아이디 한 줄 입력 제한하기
2	비밀번호, 비밀번호 확인 역시 ***로 뜨게 하기
3	가입 완료 버튼 클릭시 LoginActivity로 돌아가기



5. 기능을 입혀보자

필수!

화면 끝까지 글이 가지 않고 개행이 되네?
Guideline 이용해보기

[기본 과제1] ConstraintLayout 심화 학습!

ConstraintDimentionRatio 이용해서
이미지 1:1 비율로!

(구현 방법 및 새로 알게 된 내용 Github
ReadMe에 적어두기)

Hint : [ConstraintLayout 공식 문서](#)
[ConstraintLayout 문서2](#)





5. 기능을 입혀보자

필수!

[기본 과제2] 회원가입 및 로그인 기능 구현하기

- 회원가입 완료시 LoginActivity로 돌아오고, 회원가입 성공한 id와 pw가 입력되어 있도록 구현.
- (구현 방법 및 새로 알게 된 내용 Github ReadMe에 적어두기)

Hint : startActivityForResult, intent.putExtra



5. 기능을 입혀보자

선택!

[성장 과제] 로그인 화면 만들기 – ConstraintLayout, LinearLayout, RelativeLayout 3가지로 모두 만들어 보기



5. 기능을 입혀보자

선택!

[성장 과제2] 자동 로그인 구현하기

회원가입시 LoginActivity로 돌아와 가입한 id,pw로 자동 로그인 하기,
로그인 하여 MainActivity로 간 경우 앱을 종료했다가 다시 켜면 LoginActivity에서 자동 로그인 하기

(구현 방법 및 새로 알게 된 내용 Github ReadMe에 적어두기)

Hint : SharedPreferences



과제 제출 방법

본인 Github repository에 과제를 올리고,
해당 링크를 **팀장이 모아 한번에 제출**

Ex) 홍길동 - '과제 깃허브 주소'
김철수 - '과제 깃허브 주소 '

chjune0205@gmail.com

OUR
SOPT

SHOUT OUR PASSION TOGETHER

감사합니다^^

SHOUT OUR PASSION TOGETHER
SOPT