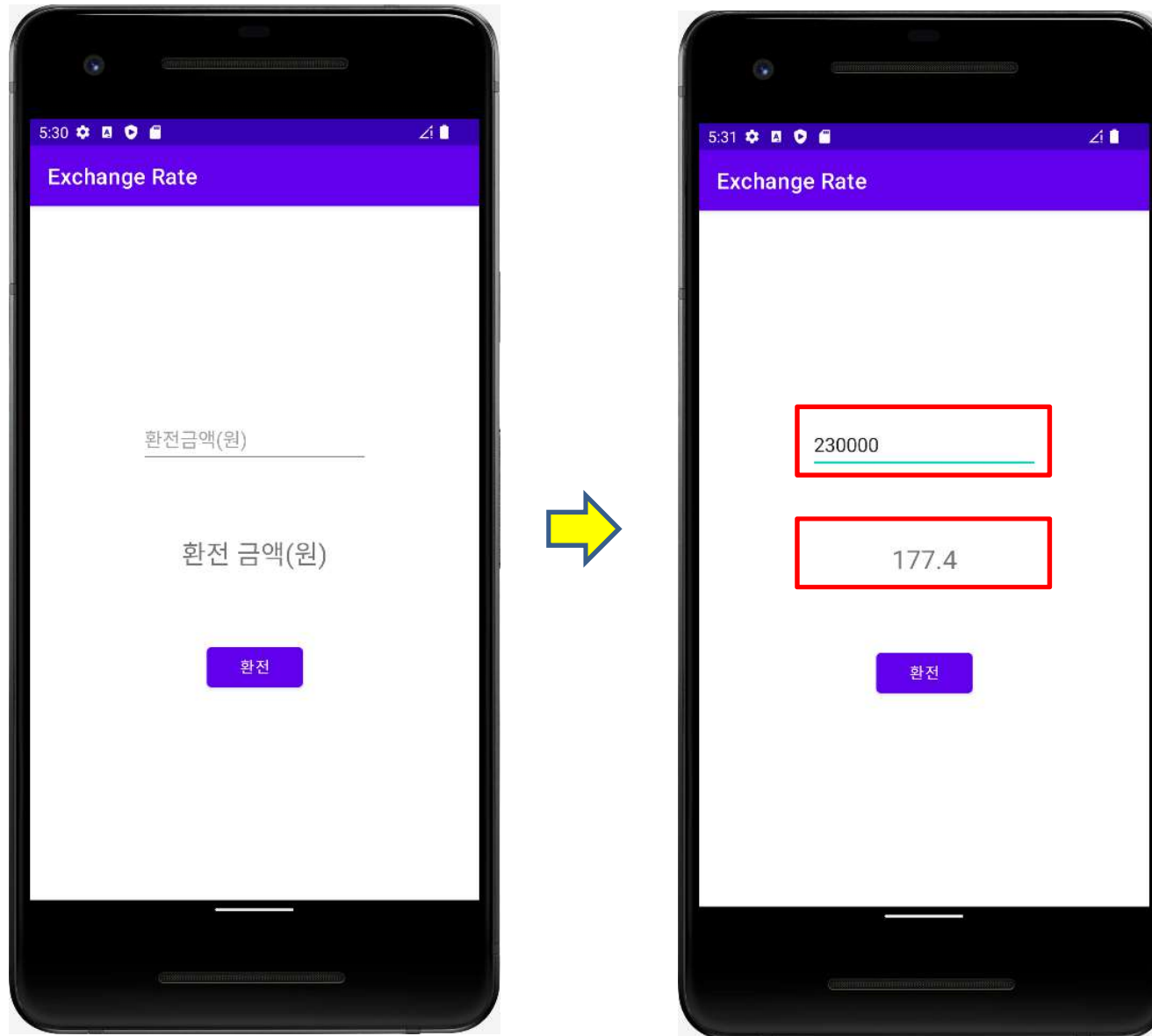


# 환전 앱

Mobile Software  
2022 Fall

All rights reserved, 2022, Copyright by Youn-Sik Hong (편집, 배포 불허)

# Converting Korean Won to US Dollar



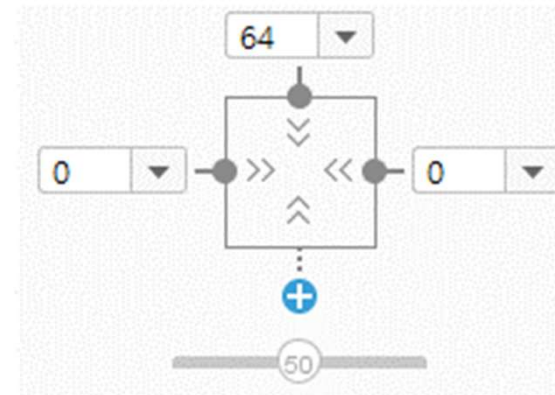
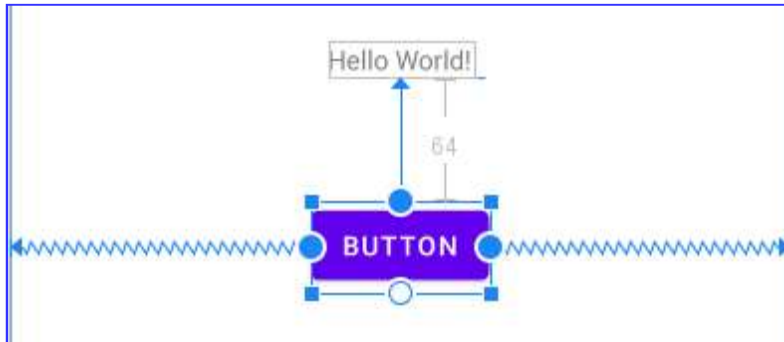
# What to do next?

- **ExchangeRate 프로젝트**
  - EditText, Button 추가 및 속성 변경
- 원화를 달러로 변환
  - 변환 과정에서 에러 처리
  - EditText 에서 직접 이벤트 처리
- 문자열 추가 및 XML 코드에서 참조
- 단위 리소스 및 색상 리소스
  - 정적 레이아웃
  - 동적 레이아웃
- 강의 노트에 포함된 코드: 5-3.소스코드.hwp

# Create a New Project

- 새 프로젝트 생성
  - Project name : **Exchange Rate**
  - Package name : **com.example.exchangerate**
  - Activity : **Empty Activity**
  - Activity name : **MainActivity.kt**
  - Layout name : **activity\_main.xml**
- 자동 생성된 XML 파일의 root view는 **ConstraintLayout**

# Adding the **Button**



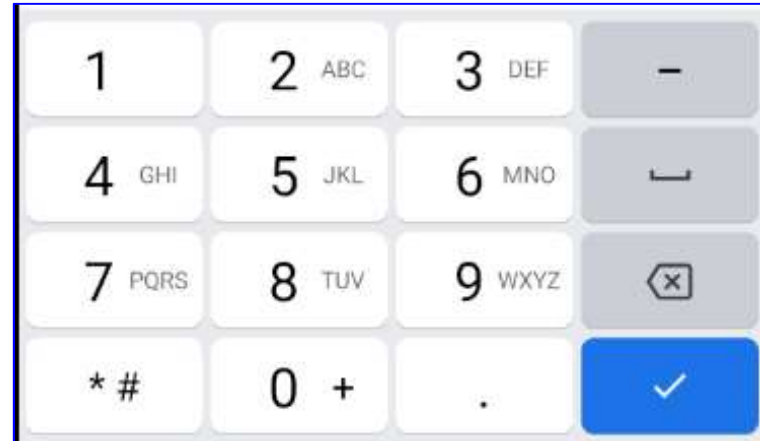
# EditText : **inputType** 속성 (1/2)

- InputType 속성
  - **text**
  - **textEmailAddress**
  - **textUri** : URL 주소 입력
  - **number, phone, ...**
- 키보드 제어
  - **textCapSentences** : 문장의 첫 번째 글자를 대문자로
  - **textCapWords** : 각 단어의 첫 번째 글자를 대문자로
  - **textAutoCorrect** : 자동 완성
  - **textPassword**
  - **textMultiLine**

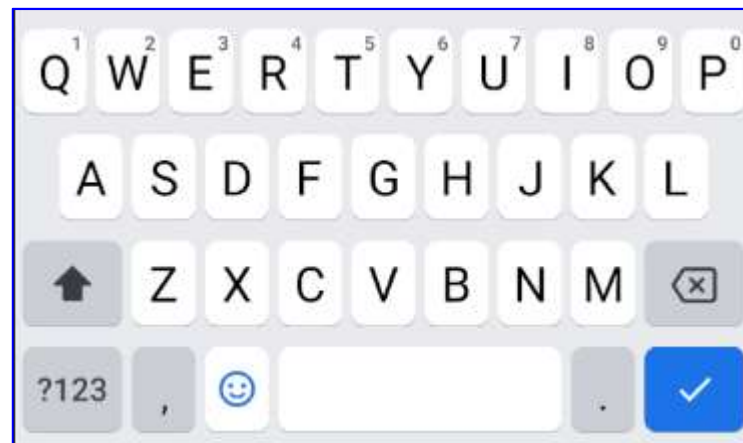
# EditText : **inputType** 속성 (2/2)



`android:inputType="textEmailAddress"`

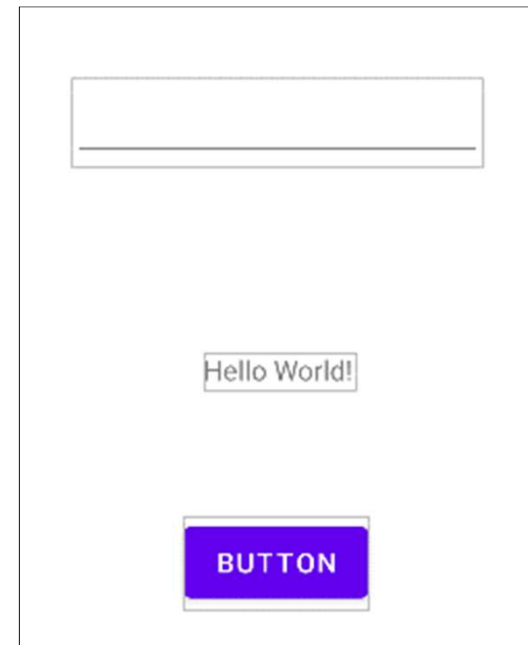
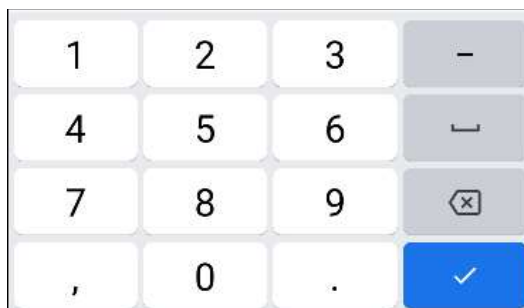
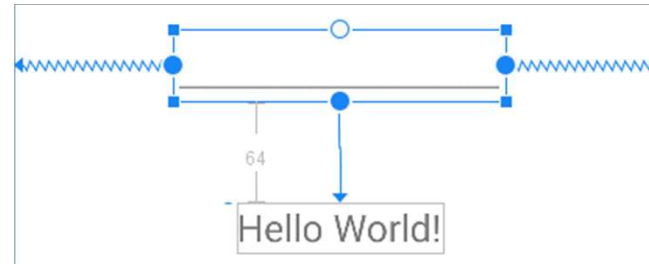
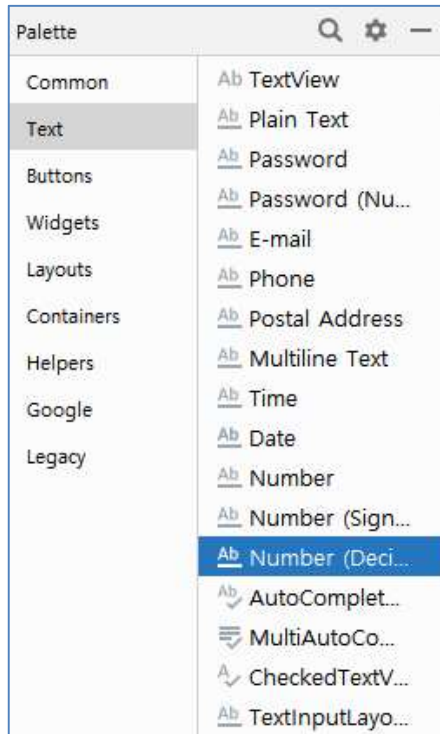


`android:inputType="phone"`



`android:inputType="text |  
textCapWords"`

# Adding the EditText





# Setting the attribute values of the **EditText**

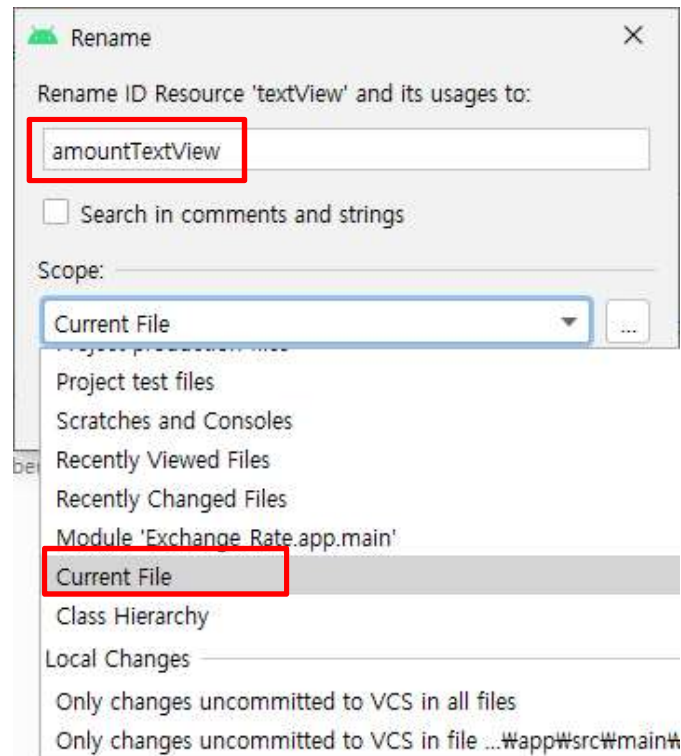
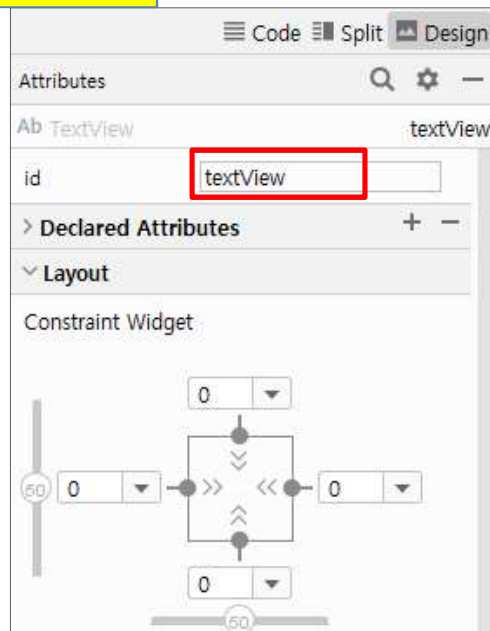
```
<EditText  
    android:id="@+id/wonEditText"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginBottom="64dp"  
    android:ems="10"  
    android:hint="환전금액(원)"  
    android:inputType="numberDecimal"  
    app:layout_constraintBottom_toTopOf="@+id/textView"  
    app:layout_constraintEnd_toEndOf="parent"  
    app:layout_constraintHorizontal_bias="0.5"  
    app:layout_constraintStart_toStartOf="parent" />
```



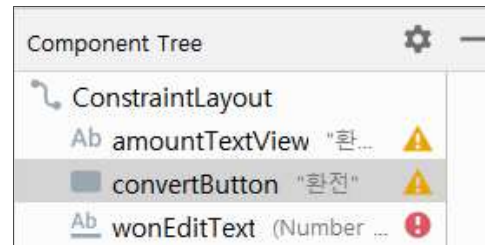
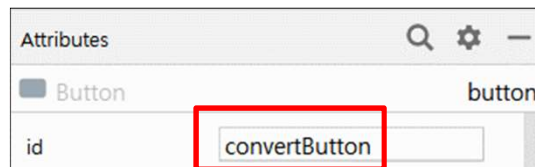
Q- ems	×	⚙	—
Ab EditText	dollarEditText		
ems	10		
> inputType	numberDecimal		
maxEms			
minEms			

# Change the values of the attribute **id**

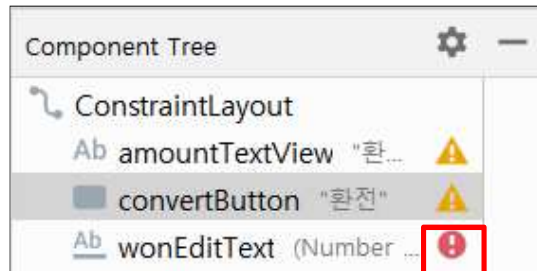
## TextView



## Button



# Fix the layout error of the **EditText**



```
android:minHeight="48dp"
```

# What to do next?

- ExchangeRate 프로젝트
  - EditText, Button 추가 및 속성 변경
- 원화를 달러로 변환
  - 변환 과정에서 에러 처리
  - EditText 에서 직접 이벤트 처리
- 문자열 추가 및 XML 코드에서 참조
- 단위 리소스 및 색상 리소스
  - 정적 레이아웃
  - 동적 레이아웃
- 강의 노트에 포함된 코드: 5-3.소스코드.hwp

# Adding the string resource : **strings.xml**

res/values/strings.xml

새 문자열(string) 리소스를 추가하려면  
리소스 이름(name)과 리소스 값(value)을 정의  
<string **name**="리소스 이름"> 리소스 값 </string>

예:

<string **name**="exchange\_rate">1296.3</string>



```
<resources>  
  <string name="app_name">Exchange Rate</string>  
  <string name="exchange_rate">1296.3</string>  
</resources>
```

# Getting the string resource

res/values/strings.xml

```
<resources>
    <string name="app_name">Exchange Rate</string>
    <string name="exchange_rate">1296.3</string>
</resources>
```

MainActivity.kt

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

        var amountTextView: TextView = findViewById(R.id.amountTextView)
        val convertButton: Button = findViewById(R.id.convertButton)
        convertButton.setOnClickListener { it: View!
            val rate:String = "1296.3"
            amountTextView.text = "환율=$rate, 문자 개수=${rate.length}"
        }
    }
}
```

환전금액(원)

환율=1296.3, 문자 개수=6

환전

'환전' 버튼만 클릭

# Converting Korean Won to US Dollar

## MainActivity.kt

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        var amountTextView: TextView = findViewById(R.id.amountTextView)  
        var wonEditText: EditText = findViewById(R.id.wonEditText)  
  
        val convertButton: Button = findViewById(R.id.convertButton)  
        convertButton.setOnClickListener {  
            var wonValue: Double = wonEditText.text.toString().toDouble()  
            val rate: Double = getString(R.string.exchange_rate).toDouble()  
            val dollarValue: Double = wonValue / rate  
            amountTextView.text = dollarValue.toString()  
        }  
    }  
}
```

230000

177.42806449124433

환전

## Error handling during conversion process

- 소수점 이하 자릿수 제한
- 숫자가 아닌 기호를 포함
- 값을 입력하지 않았을 때



# Error handling: 소수점 이하 자릿수 제한

```
val convertButton: Button = findViewById(R.id.convertButton)
convertButton.setOnClickListener { it: View!
    var wonValue: Double = wonEditText.text.toString().toDouble()
    val rate: Double = getString(R.string.exchange_rate).toDouble()
    amountTextView.text = String.format("%.1f", wonValue / rate)
}
```

230000

177.4

환전

```
val dollarValue: Double = round(wonValue / rate)
```

여러 개 클래스 중  
선택이 필요!

Imports	
round(float) (of java.lang.Math)	< Android API 32 Platform > (android.jar)
round(float) (of java.lang.StrictMath)	< Android API 32 Platform > (android.jar)
round(short) (of android.util.Half)	< Android API 32 Platform > (android.jar)
round(kotlin.Double) (kotlin.math)	Gradle: org.jetbrains.kotlin:kotlin-stdlib:1.7.0 (kotlin-stdlib-1.7.0.jar)

```
val convertButton: Button = findViewById(R.id.convertButton)
convertButton.setOnClickListener { it: View!
    var wonValue: Double = wonEditText.text.toString().toDouble()
    val rate: Double = getString(R.string.exchange_rate).toDouble()
    val dollarValue: Double = round(x: wonValue*10 / rate) / 10 // 177.4 출력
    // val dollarValue: Double = round(wonValue / rate) // 177.0 출력
    amountTextView.text = dollarValue.toString()
}
```

자릿수만큼 곱하고  
자릿수만큼 나눠야 함.

# Error handling: Contains non-numeric symbols

**contains ()**

**replace (old-value, new-value)**

```
val convertButton: Button = findViewById(R.id.convertButton)
convertButton.setOnClickListener { it: View!
    var wonValueStr:String = wonEditText.text.toString()
    if (wonValueStr.contains( other: ".")) {
        wonValueStr = wonValueStr.replace( oldValue: ".", newValue: "")
    }
    var wonValue:Double = wonValueStr.toDouble()
    val rate:Double = getString(R.string.exchange_rate).toDouble()
    amountTextView.text = String.format("%.1f", wonValue / rate )
}
```

# Error handling: When no value is entered

Toast 창 → pop up window → 사용 안내 메시지 출력

```
val convertButton: Button = findViewById(R.id.convertButton)
convertButton.setOnClickListener { it: View!
    var wonValueStr:String = wonEditText.text.toString()
    if (wonValueStr.isNotEmpty()) {
        var wonValueStr:String = wonEditText.text.toString()
        if (wonValueStr.contains("other: ".")) {
            wonValueStr = wonValueStr.replace(oldValue: ".", newValue: "")
        }
        var wonValue:Double = wonValueStr.toDouble()
        val rate: Double = getString(R.string.exchange_rate).toDouble()
        amountTextView.text = String.format("%.1f", wonValue / rate)
    } else {
        Toast.makeText(applicationContext, text: "금액을 입력하세요",
            Toast.LENGTH_LONG).show()
    }
}
```

# 잠깐! imeOptions 이란?

*IME(input method editor)*

EditText 창에 문자열 입력하려고 하면,  
화면 아래쪽에 soft keyboard가 나타남.

➔ 이 keyboard에서 Enter 키를 실행 목적에 맞게 바꿀 수 있는 것을 말함.



# Layout – imeOptions 속성



# IME\_ACTION\_SEND 이벤트 처리

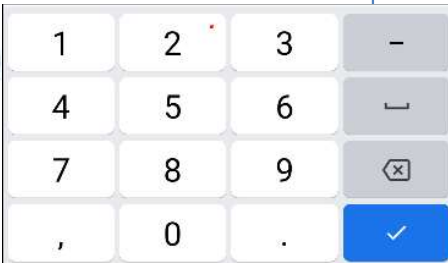
```
var wonEditText: EditText = findViewById(R.id.wonEditText)
wonEditText.setOnEditorActionListener( object: TextView.OnEditorActionListener {
    override fun onEditorAction(v: TextView?, actionId: Int, event: KeyEvent?): Boolean {
        if (actionId == EditorInfo.IME_ACTION_SEND) {
            Toast.makeText(
                applicationContext, wonEditText.text, Toast.LENGTH_LONG
            ).show()
            return true
        }
        return false
    }
})
```



# Domain Knowledge:

## EditText에서 직접 이벤트 처리 (1/2)

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        var wonEditText: EditText = findViewById(R.id.wonEditText)  
        wonEditText.setOnEditorActionListener(  
            object: TextView.OnEditorActionListener {  
                override fun onEditorAction(v: TextView?,  
                    actionId: Int,  
                    event: KeyEvent?): Boolean {  
                    if (actionId == EditorInfo.IME_ACTION_DONE) {  
                        exchangeCurrency(v?.text.toString())  
                        return true  
                    }  
                    return false  
                }  
            })  
    }  
}  
  
private fun exchangeCurrency(value:String) {...}
```



true → soft keypad 없앴

false → soft keypad 남아 있음

## Domain Knowledge: EditText에서 직접 이벤트 처리 (2/2)

```
private fun exchangeCurrency(value:String) {  
    var s:String = value  
    if (s.isNotEmpty()) {  
        if (s.contains( other: ".")) {  
            s = s.replace( oldValue: ".", newValue: "")  
        }  
        var wonValue:Double = s.toDouble()  
        val rate: Double = getString(R.string.exchange_rate).toDouble()  
        amountTextView.text = String.format("%.1f", wonValue / rate)  
    } else {  
        Toast.makeText(applicationContext, text: "금액을 입력하세요",  
            Toast.LENGTH_LONG).show()  
    }  
}
```



# 람다 식 변환

```
dollarEditText.setOnEditorActionListener(  
    object: TextView.OnEditorActionListener {  
        // Add import for 'android.widget.TextView.OnEditorActionListener'  
        // Convert to with  
        // Convert to lambda  
        // Introduce import alias  
        // Convert to run  
        // Put arguments on one line  
        // Convert to apply  
        // Convert to also  
        // Press Ctrl+Shift+I to open preview  
        // ...  
    }  
)  
return false  
}
```

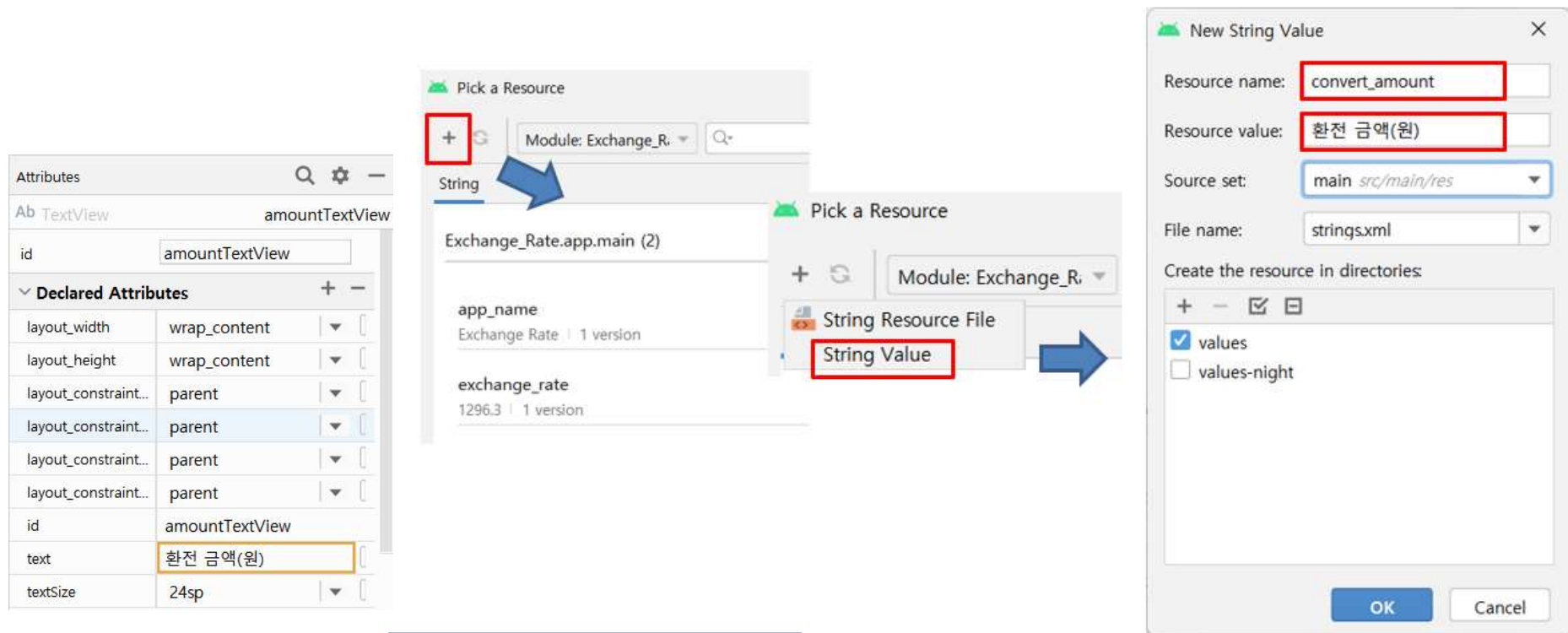
```
var wonEditText: EditText = findViewById(R.id.wonEditText)  
wonEditText.setOnEditorActionListener { v, actionId, _ ->  
    if (actionId == EditorInfo.IME_ACTION_DONE) {  
        exchangeCurrency(v?.text.toString())  
        true  
    }  
    false ^setOnEditorActionListener  
}
```

'\_' → 파라미터를 사용하지 않을 때.

# What to do next?

- ExchangeRate 프로젝트
  - EditText, Button 추가 및 속성 변경
- 원화를 달러로 변환
  - 변환 과정에서 에러 처리
  - EditText 에서 직접 이벤트 처리
- 문자열 추가 및 XML 코드에서 참조
- 단위 리소스 및 색상 리소스
  - 정적 레이아웃
  - 동적 레이아웃
- 강의 노트에 포함된 코드: 5-3.소스코드.hwp

# Adding the string resource in the Attributes window



res/values/strings.xml

```
<resources>  
  <string name="app_name">Exchange Rate</string>  
  <string name="exchange_rate">1296.3</string>  
  <string name="convert_amount">환전 금액(원)</string>  
</resources>
```

# Referring the string resources in the XML Layout

## res/values/strings.xml

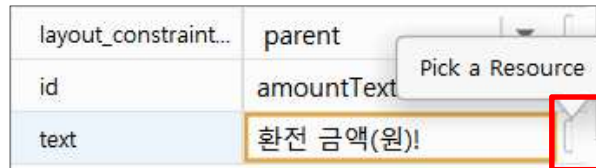
```
<resources>
  <string name="app_name">Exchange Rate</string>
  <string name="exchange_rate">1296.3</string>
  <string name="convert_amount">환전 금액(원)</string>
  <string name="convert_button">환전</string>
</resources>
```

## activity\_main.xml

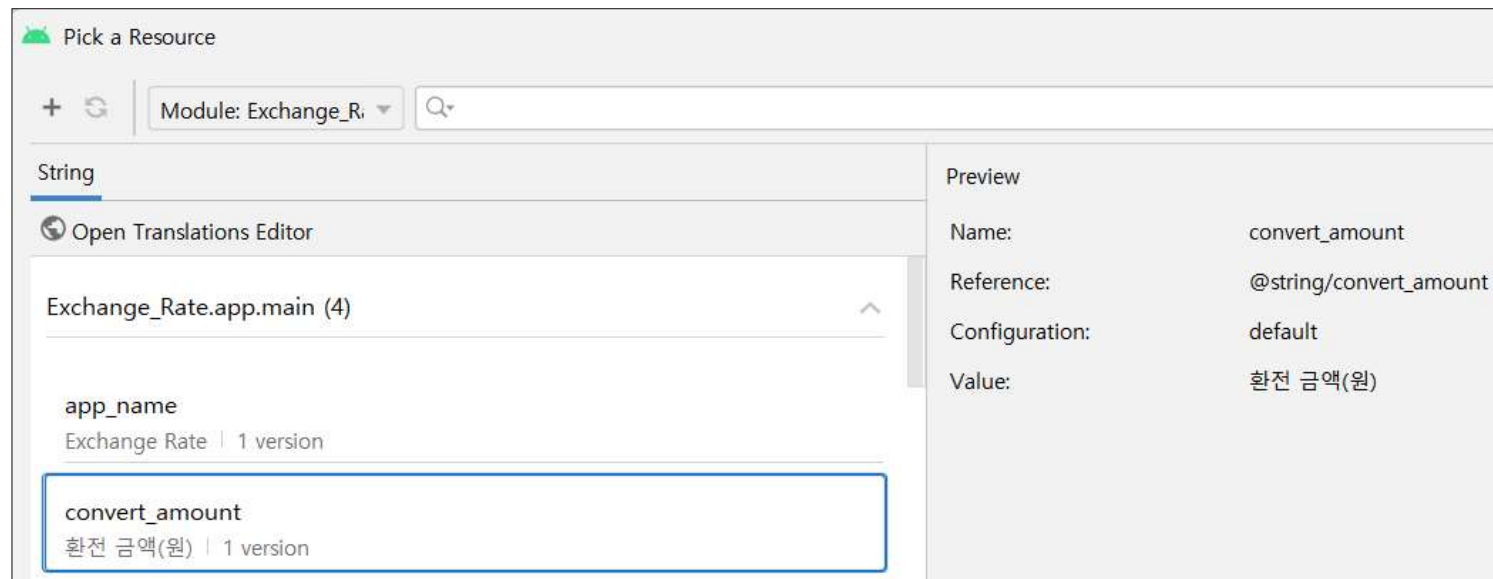
```
<TextView
  android:id="@+id/amountTextView"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:text="@string/convert_amount"
  android:textSize="24sp"
  app:layout_constraintBottom_toBottomOf="parent"
  app:layout_constraintEnd_toEndOf="parent"
  app:layout_constraintHorizontal_bias="0.5"
  app:layout_constraintStart_toStartOf="parent"
  app:layout_constraintTop_toTopOf="parent" />

<Button
  android:id="@+id/convertButton"
  android:layout_width="wrap_content"
  android:layout_height="wrap_content"
  android:layout_marginTop="64dp"
  android:text="@string/convert_button"
  app:layout_constraintEnd_toEndOf="parent"
  app:layout_constraintHorizontal_bias="0.5"
  app:layout_constraintStart_toStartOf="parent"
  app:layout_constraintTop_toBottomOf="@+id/amountTextView" />
```

# Referring to the string resource



XML 코드에서 입력하는 대신  
속성 창에서 문자열 리소스를 선택

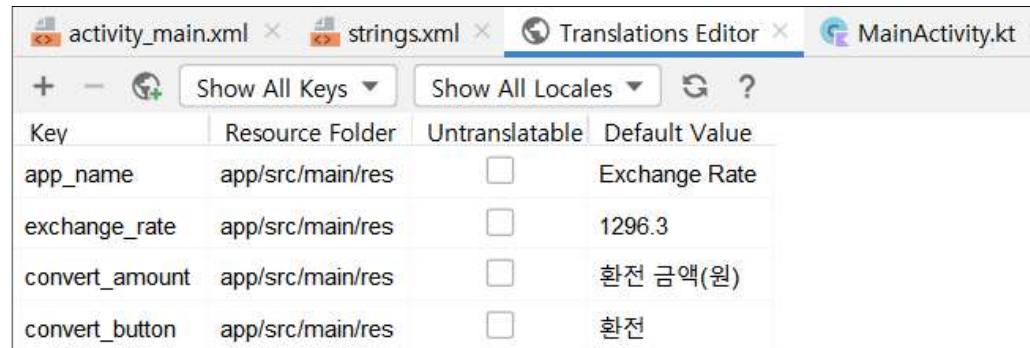


# How to reference a resource

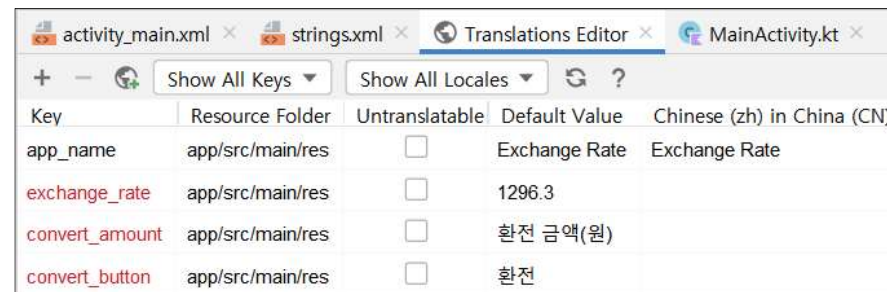
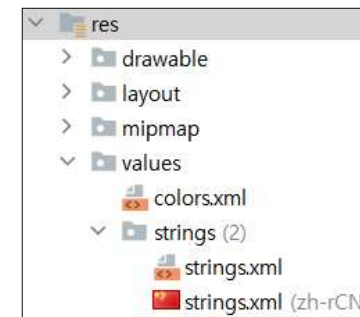
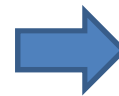
- @리소스타입/리소스이름
  - XML 문서에서 리소스를 참조할 때
    - strings.xml → @string/xxx
    - dimens.xml → @dimen/xxx
- R.리소스타입.리소스이름
  - Kotlin 코드에서 리소스를 참조할 때
    - res/layout/activity\_main.xml  
→ R.layout.activity\_main
    - res/values/strings.xml  
→ R.string.xxx

# String resource utilization

strings.xml > 오른쪽 버튼 > Open Translations Editor



Key	Resource Folder	Untranslatable	Default Value
app_name	app/src/main/res	<input type="checkbox"/>	Exchange Rate
exchange_rate	app/src/main/res	<input type="checkbox"/>	1296.3
convert_amount	app/src/main/res	<input type="checkbox"/>	환전 금액(원)
convert_button	app/src/main/res	<input type="checkbox"/>	환전



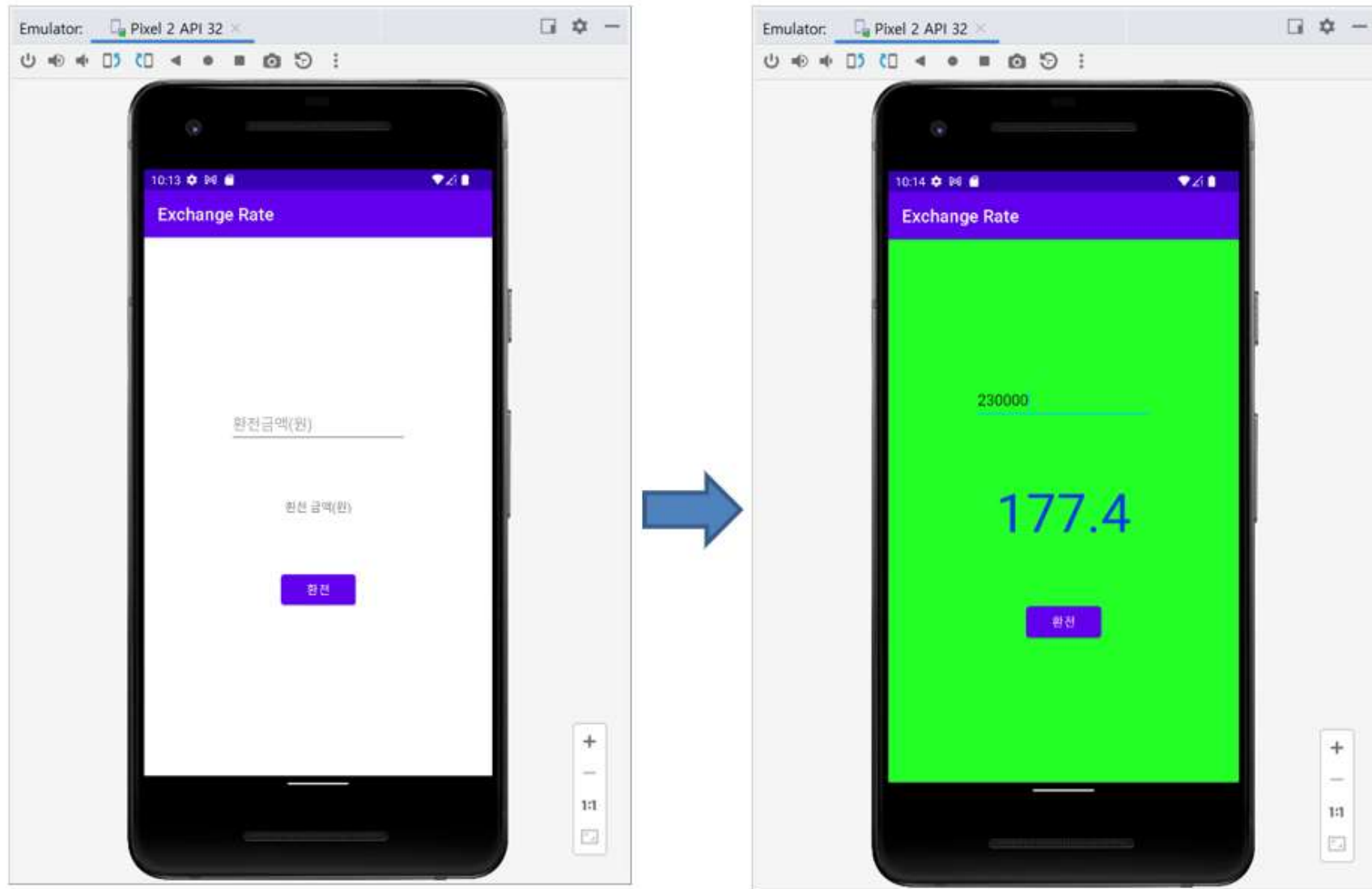
Key	Resource Folder	Untranslatable	Default Value	Chinese (zh) in China (CN)
app_name	app/src/main/res	<input type="checkbox"/>	Exchange Rate	Exchange Rate
exchange_rate	app/src/main/res	<input type="checkbox"/>	1296.3	
convert_amount	app/src/main/res	<input type="checkbox"/>	환전 금액(원)	
convert_button	app/src/main/res	<input type="checkbox"/>	환전	

# What to do next?

- ExchangeRate 프로젝트
  - EditText, Button 추가 및 속성 변경
- 원화를 달러로 변환
  - 변환 과정에서 에러 처리
  - EditText 에서 직접 이벤트 처리
- 문자열 추가 및 XML 코드에서 참조
- 단위 리소스 및 색상 리소스
  - 정적 레이아웃
  - 동적 레이아웃
- 강의 노트에 포함된 코드: 5-3.소스코드.hwp



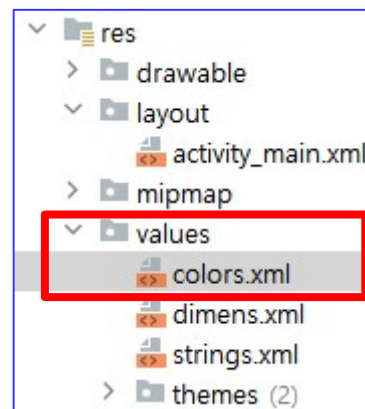
# Change the attributes of the ConstraintLayout and the TextView



# Adding the color resources (**colors.xml**)



컬러 사각형을 클릭하면  
색상 조절이 가능한  
Pop-up window가 나타남



색상은 #AARRGGBB 로 표현

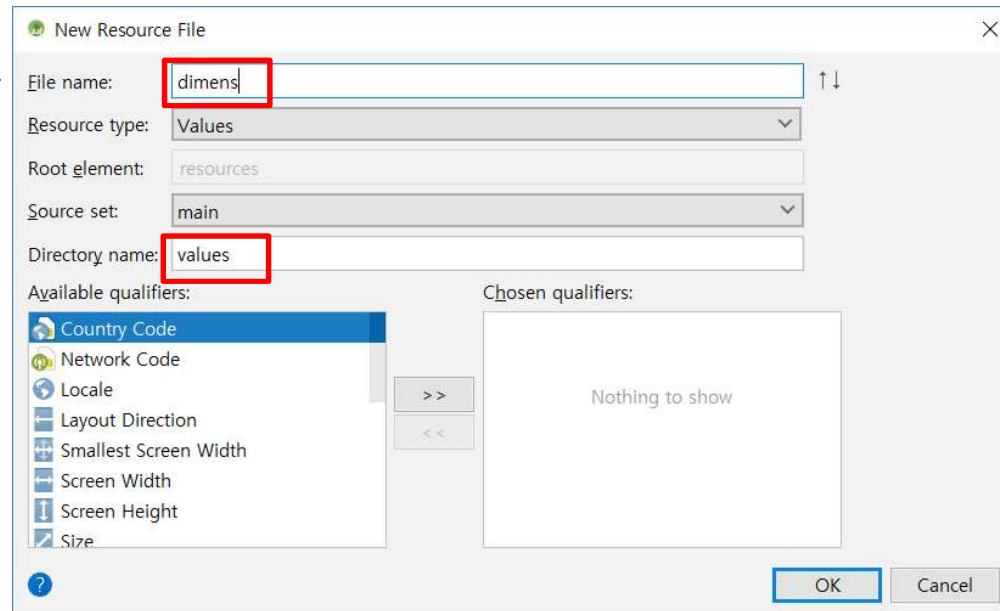
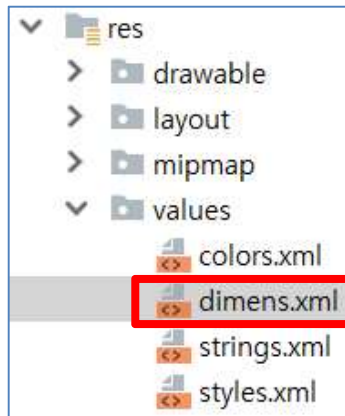
- #기호는 반드시 사용.
- 2자리 16진수 숫자로 표현(0~255)
- A(alpha), R(red), G(green), B(blue)
- A는 투명도(0:투명, 255:반투명)
- A는 생략 가능

# Creates the dimension resource

**res**

> 오른쪽 버튼 > New  
> Android Resource File

res/values/dimens.xml



**dimens.xml**

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <dimen name="textview_size">24dp</dimen>
</resources>
```

**dimens** → dimension  
단위(unit) resource를 말함

**dp**: 픽셀 기준 단위  
device independent pixel

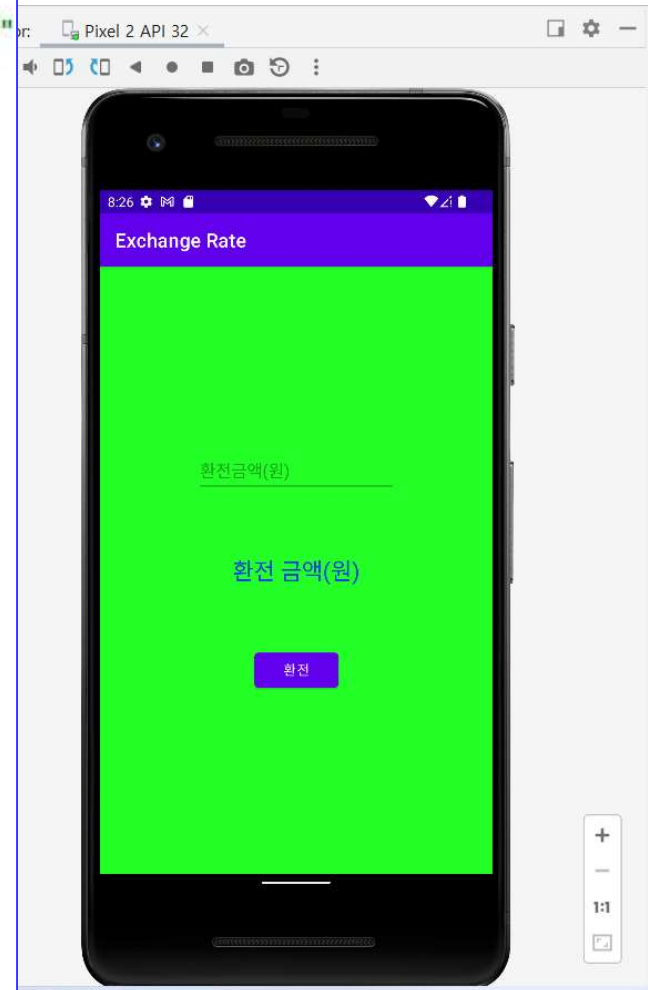
# Static layout

```
<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"
    android:background="@color/color_background"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/amountTextView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/convert_amount"
        android:textSize="@dimen/textview_size"
        android:textColor="@color/color_textview"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.5"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button...>

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



# 코드 (MainActivity.kt)에서 속성 설정 예

```
<?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:id="@+id/viewgroup_screen"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/color_background"
    tools:context=".MainActivity">

    <TextView...>
    <Button...>
    <EditText...>
</androidx.constraintlayout.widget.ConstraintLayout>
```

< activity\_main.xml >  
Root view인 ConstraintLayout에  
id 속성 설정

this →  
MainActivity 객체

```
class MainActivity : AppCompatActivity() {

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

        val bgId = findViewById<ConstraintLayout>(R.id.screen_background)
        val bgColor = ContextCompat.getColor(context this, R.color.color_background)
        bgId.setBackgroundColor(bgColor)
    }

    fun onBtnClicked(view: View) {...}
}
```

회색 문자는 입력하지 마세요!!!  
Studio에서 파라미터 타입을 알려주는 메시지.



# 잠깐! deprecated

```
val bgColor = resources.getColor(R.color.color_background)
```

'getColor(Int): Int' is deprecated. Deprecated in Java



getColor deprecated android



It looks like the best approach is to use:

1095

```
ContextCompat.getColor(context, R.color.color_name)
```

deprecated

더 이상 사용되지 않음.  
→ 실행은 할 수 있지만  
더 이상 지원하지 않음.

# Dynamic layout (1/2)

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        var wonEditText: EditText = findViewById(R.id.wonEditText)  
        wonEditText.setOnEditorActionListener { v, actionId, _ ->  
            if (actionId == EditorInfo.IME_ACTION_DONE) {  
                exchangeCurrency(v.text.toString())  
                true  
            }  
            false ^setOnEditorActionListener  
        }  
  
        val convertButton: Button = findViewById(R.id.convertButton)  
        convertButton.setOnClickListener { it: View!  
            exchangeCurrency(wonEditText.text.toString())  
        }  
    }  
  
    private fun exchangeCurrency(value:String) {...}  
}
```

# Dynamic layout (2/2)

```
private fun exchangeCurrency(value:String) {
    var s:String = value
    if (s.isNotEmpty()) {
        val bgId:ConstraintLayout = findViewById(R.id.viewgroup_screen)
        val bgColor = ContextCompat.getColor( context: this, R.color.color_background)
        bgId.setBackgroundColor(bgColor)

        val color = ContextCompat.getColor( context: this, R.color.color_textview)
        val textSizeDP = resources.getDimension(R.dimen.textview_size)
        var amountTextView:TextView = findViewById(R.id.amountTextView)
        amountTextView.setTextSize(TypedValue.COMPLEX_UNIT_DIP, textSizeDP)
        amountTextView.setTextColor(color)

        if (s.contains( other: ".")) {
            s = s.replace( oldValue: ".", newValue: "")
        }
        var wonValue:Double = s.toDouble()
        val rate: Double = getString(R.string.exchange_rate).toDouble()
        amountTextView.text = String.format("%.1f", wonValue / rate)
    } else {
        Toast.makeText(applicationContext, text: "금액을 입력하세요",
            Toast.LENGTH_LONG).show()
    }
}
```

Resources:

```
<resources>
    <dimen name="textview_size">20dp</dimen>
</resources>
```

```
10 <color name="colorTextView">#003BEA</color>
11 <color name="color_background">#24FF28</color>
```



# 잠깐! 코딩할 때 유용한 기능

- Java 소스 코드에서
  - Ctrl-Q** : 해당 method에 대한 설명
  - Ctrl-P** : method에서 입력해야 할 parameter
- 코드를 수정하고 나서 indent를 맞추는 때
  - Code > Reformat code**
- 코드 블록을 주석(comment) 처리하고 싶으면
  - Code > Comment with Block Comment**
- 파일 이름을 바꾸고 싶으면
  - 파일 이름 클릭 > 마우스 오른쪽 버튼 > Refactor**
- XML에서 주석 처리하려면
  - <!-- -->**로 주석 처리할 부분을 둘러 씌