**■ 단위 변환 (dp -> px)**

**======================= MainActivity.kt ===========================**

package com.example.eventhandlerpractice1

import android.os.Bundle

import android.util.Log

import androidx.appcompat.app.AppCompatActivity

class MainActivity : AppCompatActivity() {

private val TAG = "Unit Conversion>>>"

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

val density: Float = resources.displayMetrics.density

val densityDpi = resources.displayMetrics.densityDpi

Log.i(TAG, "density = $density") // screen density

Log.i(TAG, "densityDpi = $densityDpi") // dpi:dot per inch

var scale: Float = densityDpi/160F

Log.i(TAG, "scale = $scale") // scale

val dp = 16

val px = (scale\*dp + 0.5).toInt()

Log.i(TAG, "dp = $dp, px = $px")

}

}

**■ 실습 1: 다양한 단위 변환 방법**

**======================= activity\_main.xml ===========================**

<?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:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="dp를 px로 변환"

android:textSize="24sp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<Button

android:id="@+id/button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="64dp"

android:text="method 1"

app:layout\_constraintEnd\_toStartOf="@+id/button2"

app:layout\_constraintHorizontal\_bias="0.5"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView" />

<Button

android:id="@+id/button2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="64dp"

android:text="method 2"

app:layout\_constraintEnd\_toStartOf="@+id/button3"

app:layout\_constraintHorizontal\_bias="0.5"

app:layout\_constraintStart\_toEndOf="@+id/button"

app:layout\_constraintTop\_toBottomOf="@+id/textView" />

<Button

android:id="@+id/button3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="64dp"

android:text="method 3"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.5"

app:layout\_constraintStart\_toEndOf="@+id/button2"

app:layout\_constraintTop\_toBottomOf="@+id/textView" />

</androidx.constraintlayout.widget.ConstraintLayout>

**======================= MainActivity.kt ===========================**

package com.example.eventhandlerpractice1

import android.os.Bundle

import android.util.TypedValue

import android.widget.Button

import android.widget.TextView

import androidx.appcompat.app.AppCompatActivity

import com.example.eventhandlerpractice1.databinding.ActivityMainBinding

import kotlin.math.roundToInt

class MainActivity : AppCompatActivity() {

private lateinit var binding: ActivityMainBinding

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

binding = ActivityMainBinding.inflate(layoutInflater)

setContentView(binding.root)

val dp = 16

val scale: Float = resources.displayMetrics.density

binding.button.setOnClickListener {

val px:Int = (scale \* dp + 0.5).toInt() // method 1

val msg = "method 1>> dp = $dp, px = $px"

binding.textView.text = msg

}

binding.button2.setOnClickListener {

val px:Int = (scale \* dp).roundToInt() // method 2

val msg = "method 2>> dp = $dp, px = $px"

binding.textView.text = msg

}

binding.button3.setOnClickListener {

val px:Int = TypedValue.applyDimension( // method 3

TypedValue.COMPLEX\_UNIT\_DIP, dp.toFloat(),

resources.displayMetrics).toInt()

val msg = "method 3>> dp = $dp, px = $px"

binding.textView.text = msg

}

}

}

**■ 실습 2: 한 개 view에 대한 2개 이벤트 처리**

**======================= activity\_main.xml ===========================**

<?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:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Hello World!"

android:textSize="24sp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<Button

android:id="@+id/myButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="64dp"

android:text="Button"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView" />

</androidx.constraintlayout.widget.ConstraintLayout>

**======================= MainActivity.kt ===========================**

package com.example.eventhandlerpractice1

import android.os.Bundle

import android.view.View

import androidx.appcompat.app.AppCompatActivity

import com.example.eventhandlerpractice1.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

private lateinit var binding: ActivityMainBinding

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

binding = ActivityMainBinding.inflate(layoutInflater)

setContentView(binding.root)

binding.myButton.setOnClickListener (object : View.OnClickListener {

override fun onClick(v: View?) {

binding.textView.text = "Button clicked"

}

})

binding.myButton.setOnLongClickListener(object : View.OnLongClickListener {

override fun onLongClick(v: View?): Boolean {

binding.textView.text = "Long button click"

return false

}

})

}

}

**■ 실습 2: 한 개 view에 대한 2개 이벤트 처리 (concise version)**

**========================= MainActivity.kt ===========================**

package com.example.eventhandlerpractice1

import android.os.Bundle

import androidx.appcompat.app.AppCompatActivity

import com.example.eventhandlerpractice1.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

private lateinit var binding: ActivityMainBinding

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

binding = ActivityMainBinding.inflate(layoutInflater)

setContentView(binding.root)

binding.myButton.setOnClickListener {

binding.textView.text = "Button clicked"

}

binding.myButton.setOnLongClickListener {

binding.textView.text = "Long button click"

false

}

}

}

**■ 버튼을 클릭하면 Toast 창에 메시지 출력 (람다 식 사용)**

**========================= MainActivity.kt ===========================**

package com.example.eventhandlerpractice1

import android.os.Bundle

import android.widget.Toast

import androidx.appcompat.app.AppCompatActivity

import com.example.eventhandlerpractice1.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

private lateinit var binding: ActivityMainBinding

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

binding = ActivityMainBinding.inflate(layoutInflater)

setContentView(binding.root)

binding.myButton.setOnClickListener {

Toast.makeText(

applicationContext,"Button is pressed", Toast.LENGTH\_SHORT)

.show()

}

binding.myButton.setOnLongClickListener {

Toast.makeText(

applicationContext,"Long Button click", Toast.LENGTH\_SHORT)

.show()

false

}

}

}

**■ 실습 3: Single Touch Event**

**======================= activity\_main.xml ===========================**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout

xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

tools:context=".MainActivity" >

<View

android:id="@+id/myView"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_weight="2"

android:background="@android:color/holo\_blue\_bright" />

<ScrollView

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:layout\_weight="1">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<TextView

android:id="@+id/textView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:text="show message" />

</LinearLayout>

</ScrollView>

</LinearLayout>

**======================= MainActivity.kt ===========================**

package com.example.eventhandlerpractice2

import android.os.Bundle

import android.view.MotionEvent

import android.view.View

import androidx.appcompat.app.AppCompatActivity

import com.example.eventhandlerpractice2.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

private lateinit var binding:ActivityMainBinding

private val sb = StringBuilder()

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

binding = ActivityMainBinding.inflate(layoutInflater)

setContentView(binding.root)

binding.myView.setOnTouchListener(object : View.OnTouchListener {

override fun onTouch(v: View?, event: MotionEvent?): Boolean {

handleTouch(event)

return true

}

})

}

private fun handleTouch(m: MotionEvent?) {

var x:Float? = m?.x

var y:Float? = m?.y

when (m?.action) {

MotionEvent.ACTION\_DOWN -> sb.append("손가락 누름 $x, $y\n")

MotionEvent.ACTION\_MOVE -> sb.append("손가락 이동 $x, $y\n")

MotionEvent.ACTION\_UP -> sb.append("손가락 뗌 $x, $y\n")

else -> sb.append("\n")

}

binding.textView.text = sb.toString()

}

}

**■ 잠깐! : safe call**

**======================= MainActivity.kt ===========================**

package com.example.eventhandlerpractice2

import android.os.Bundle

import android.view.MotionEvent

import android.view.View

import androidx.appcompat.app.AppCompatActivity

import com.example.eventhandlerpractice2.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

private lateinit var binding:ActivityMainBinding

private val sb = StringBuilder()

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

binding = ActivityMainBinding.inflate(layoutInflater)

setContentView(binding.root)

binding.myView.setOnTouchListener(object : View.OnTouchListener {

override fun onTouch(v: View?, event: MotionEvent?): Boolean {

handleTouch(event)

return true

}

})

}

private fun handleTouch(m: MotionEvent?) {

if (m == null) return

var x:Float = m.x

var y:Float = m.y

when (m.action) {

MotionEvent.ACTION\_DOWN -> sb.append("손가락 누름 $x, $y\n")

MotionEvent.ACTION\_MOVE -> sb.append("손가락 이동 $x, $y\n")

MotionEvent.ACTION\_UP -> sb.append("손가락 뗌 $x, $y\n")

else -> sb.append("\n")

}

binding.textView.text = sb.toString()

}

}

**■ 실습 3(추가): 출력 서식 지정 (1/2)**

**======================= MainActivity.kt ===========================**

private fun handleTouch(m: MotionEvent?) {

if (m == null) return

var x:Float = m.x

var y:Float = m.y

when (m.action) {

MotionEvent.ACTION\_DOWN -> outMessage("손가락 누름", x, y)

MotionEvent.ACTION\_MOVE -> outMessage("손가락 이동", x, y)

MotionEvent.ACTION\_UP -> outMessage("손가락 뗌", x, y)

else -> outMessage("액션 없음", x, y)

}

}

private fun outMessage(msg: String, x1:Float, y1:Float) {

sb.append("%-10s: %.2f, %.2f \n".format(msg, x1, y1))

binding.textView.text = sb.toString()

}

**■ 실습 3(추가): 출력 서식 지정 (2/2)**

**======================= MainActivity.kt ===========================**

import java.math.RoundingMode

import java.text.DecimalFormat

private fun handleTouch(m: MotionEvent?) {

if (m == null) return

var x:Float = m.x

var y:Float = m.y

when (m.action) {

MotionEvent.ACTION\_DOWN -> outMessage("손가락 누름", x, y)

MotionEvent.ACTION\_MOVE -> outMessage("손가락 이동", x, y)

MotionEvent.ACTION\_UP -> outMessage("손가락 뗌", x, y)

else -> outMessage("액션 없음", x, y)

}

}

private fun outMessage(msg: String, x: Float, y: Float) {

val df = DecimalFormat("#.##")

df.roundingMode = RoundingMode.CEILING

sb.append("$msg : ${df.format(x)}, ${df.format(y)} \n")

binding.textView.text = sb.toString()

}

**■ 실습 4: Multi-Touch Event**

**======================= activity\_main.xml ===========================**

<?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/activity\_motion\_event"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<TextView

android:id="@+id/textView"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="64dp"

android:text="Touch One Status"

android:textSize="16sp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.5"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<TextView

android:id="@+id/textView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="16dp"

android:text="Touch Two Status"

android:textSize="16sp"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.5"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView"

app:layout\_constraintVertical\_bias="0.091" />

</androidx.constraintlayout.widget.ConstraintLayout>

**======================= MainActivity.kt ===========================**

package com.example.eventhandlerpractice2

import android.os.Bundle

import android.view.MotionEvent

import androidx.appcompat.app.AppCompatActivity

import com.example.eventhandlerpractice2.databinding.ActivityMainBinding

class MainActivity : AppCompatActivity() {

private lateinit var binding: ActivityMainBinding

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

binding = ActivityMainBinding.inflate(layoutInflater)

setContentView(binding.root)

binding.activityMotionEvent.setOnTouchListener { v, event ->

handleTouch(event)

true

}

}

private fun handleTouch(m: MotionEvent) {

val pointerCount = m.pointerCount

val actionIndex:Int = m.actionIndex

for (i in 0 until pointerCount) {

val x:Float = m.getX(i)

val y:Float = m.getY(i)

val id:Int = m.getPointerId(i)

var actionString = when (m.action) {

MotionEvent.ACTION\_DOWN -> "DOWN"

MotionEvent.ACTION\_UP -> "UP"

MotionEvent.ACTION\_POINTER\_DOWN -> "PNTR DOWN"

MotionEvent.ACTION\_POINTER\_UP -> "PNTR UP"

MotionEvent.ACTION\_MOVE -> "MOVE"

else -> ""

}

var message = "Index: $actionIndex, Action: $actionString, ID: $id"

if (id == 0) binding.textView.text = outMessage(message, x, y)

if (id == 1) binding.textView2.text = outMessage(message, x, y)

}

}

private fun outMessage(msg: String, x1: Float, y1: Float) =

"%s : %.2f, %.2f \n".format(msg, x1, y1)

}