

04

04-2. 터치, 키 이벤트

*Event Programming*

# 터치와 키 이벤트

---

## 터치 이벤트

- 터치 이벤트의 콜백 함수인 onTouchEvent()를 선언
- 매개변수는 MotionEvent 객체이며, 이 객체에 터치의 종류와 발생 지점(좌표값)이 담깁니다.

### • 터치 이벤트 처리

```
class MainActivity : AppCompatActivity() {  
    (... 생략 ...)  
    override fun onTouchEvent(event: MotionEvent?): Boolean {  
        return super.onTouchEvent(event)  
    }  
}
```

# 터치와 키 이벤트

---

- 터치 이벤트의 종류
  - ACTION\_DOWN: 화면을 손가락으로 누른 순간의 이벤트
  - ACTION\_UP: 화면에서 손가락을 떼는 순간의 이벤트
  - ACTION\_MOVE: 화면을 손가락으로 누른 채로 이동하는 순간의 이벤트

## • 터치 이벤트 처리

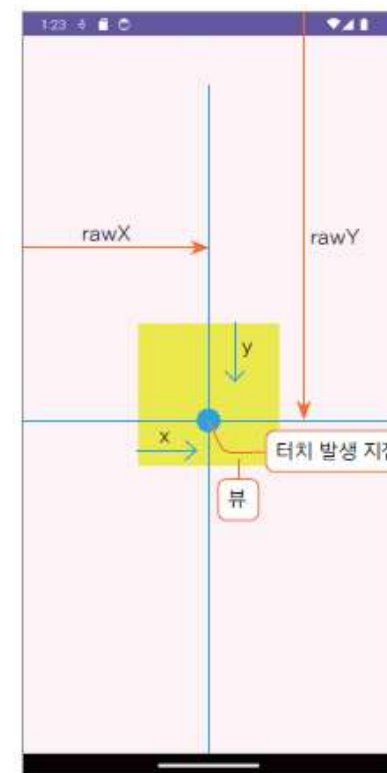
```
override fun onTouchEvent(event: MotionEvent?): Boolean {  
    when (event?.action) {  
        MotionEvent.ACTION_DOWN -> {  
            Log.d("kkang", "Touch down event")  
        }  
        MotionEvent.ACTION_UP -> {  
            Log.d("kkang", "Touch up event")  
        }  
    }  
    return super.onTouchEvent(event)  
}
```

# 터치와 키 이벤트

- 터치 이벤트 발생 좌표 얻기
- onTouchEvent() 함수의 매개변수인 MotionEvent 객체로 획득
  - x: 이벤트가 발생한 뷰의 X 좌표
  - y: 이벤트가 발생한 뷰의 Y 좌표
  - rawX: 화면의 X 좌표
  - rawY: 화면의 Y 좌표

• 터치 이벤트가 발생한 좌표 얻기

```
override fun onTouchEvent(event: MotionEvent?): Boolean {  
    when (event?.action) {  
        MotionEvent.ACTION_DOWN -> {  
            Log.d("kkang",  
                "Touch down event x: ${event.x}, rawX: ${event.rawX}")  
        }  
    }  
    return super.onTouchEvent(event)  
}
```



# 터치와 키 이벤트

---

## 키 이벤트

- 사용자가 폰의 키를 누르는 순간에 발생
- 콜백 함수
  - onKeyDown: 키를 누른 순간의 이벤트
  - onKeyUp: 키를 떼는 순간의 이벤트
  - onKeyLongPress: 키를 오래 누르는 순간의 이벤트

### • 키 이벤트 처리

```
class MainActivity2 : AppCompatActivity() {  
    (... 생략 ...)  
    override fun onKeyDown(keyCode: Int, event: KeyEvent?): Boolean {  
        Log.d("kkang", "onKeyDown")  
        return super.onKeyDown(keyCode, event)  
    }  
    override fun onKeyUp(keyCode: Int, event: KeyEvent?): Boolean {  
        Log.d("kkang", "onKeyUp")  
        return super.onKeyUp(keyCode, event)  
    }  
}
```

# 터치와 키 이벤트

---

- 첫 번째 매개변수는 키의 코드이며 이 값으로 사용자가 어떤 키를 눌렀는지 식별

• 어떤 키를 눌렀는지 식별

```
override fun onKeyDown(keyCode: Int, event: KeyEvent?): Boolean {  
    when (keyCode) {  
        KeyEvent.KEYCODE_0 -> Log.d("kkang", "0 키를 눌렀네요")  
        KeyEvent.KEYCODE_A -> Log.d("kkang", "A 키를 눌렀네요")  
    }  
    return super.onKeyDown(keyCode, event)  
}
```

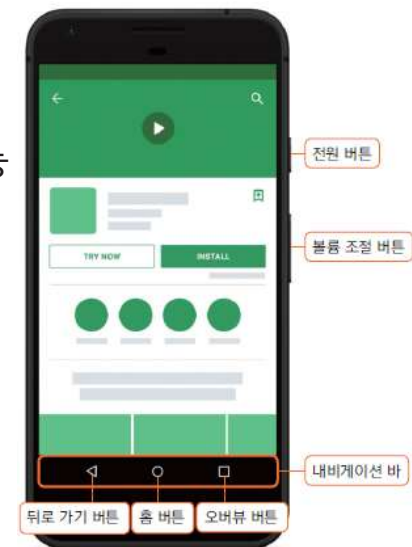
# 터치와 키 이벤트

- 키 이벤트가 발생하는 키는 폰에서 제공하는 소프트 키보드의 키를 의미하지 않습니다.
- 안드로이드 시스템 버튼도 키로 취급하므로 이 버튼의 이벤트를 처리
- 뒤로가기 버튼 이벤트에는 앞에서 살펴본 `onKeyDown()`이나 `onKeyUp()` 함수를 이용할 수도 있지만 `onBackPressed()` 함수를 이용할 수도 있습니다.
- 제스처 내비게이션을 사용하고 있다면 사용자 제스처로 뒤로 가기를 하면 <뒤로 가기> 버튼을 누른 것과 같습니다.

그러므로 사용자 제스처 뒤로 가기도 `onKeyDown()` 함수를 선언해 이벤트 처리가 가능

## • 뒤로 가기 버튼과 볼륨 조절 버튼의 이벤트 처리

```
override fun onKeyDown(keyCode: Int, event: KeyEvent?): Boolean {  
    when (keyCode) {  
        KeyEvent.KEYCODE_BACK -> Log.d("kkang", "BACK Button을 눌렀네요")  
        KeyEvent.KEYCODE_VOLUME_UP -> Log.d("kkang", "Volume Up 키를 눌렀네요")  
        KeyEvent.KEYCODE_VOLUME_DOWN -> Log.d("kkang", "Volume Down 키를 눌렀네요")  
    }  
    return super.onKeyDown(keyCode, event)  
}
```



# 터치와 키 이벤트

---

- 뒤로가기 버튼 이벤트에는 `onBackPressed()` 함수를 이용할 수도 있습니다.

```
• 뒤로 가기 버튼의 이벤트 처리  
  
override fun onBackPressed() {  
    Log.d("kkang", "Back Button을 눌렀네요")  
}
```

- API Level 33에서 뒤로 가기 버튼 이벤트 처리 함수인 `onBackPressed()` 함수는 deprecated
- `androidx.activity.OnBackPressedCallback()` 함수 이용을 권장

```
• 뒤로 가기 버튼의 이벤트 처리  
  
onBackPressedDispatcher.addCallback(this, object : OnBackPressedCallback(true) {  
    override fun handleOnBackPressed() {  
  
    }  
})
```





# 감사합니다

단단히 마음먹고 떠난 사람은  
산꼭대기에 도착할 수 있다.  
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어  
William Shakespeare