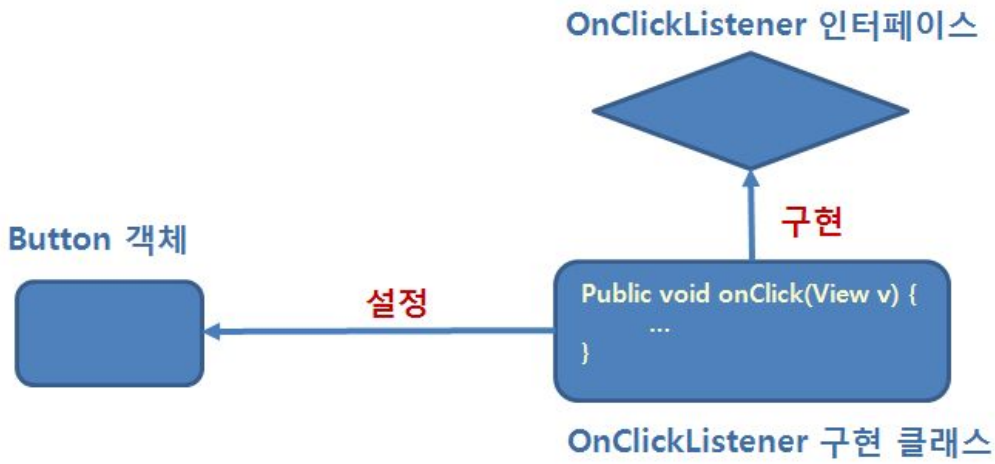
# 1. 이벤트 처리

# 뷰의 이벤트 처리하기




[버튼에 OnClickListener를 설정할 때의 패턴]

뷰를 상속할 때 이벤트 처리를 위한 메소드 재정의

boolean onTouchEvent (MotionEvent event)

boolean onKeyDown (int keyCode, KeyEvent event)

boolean onKeyUp (int keyCode, KeyEvent event)

뷰 객체에 전달되는 이벤트를 처리하기 위한 리스너 설정

View.OnTouchListener : boolean onTouch (View v, MotionEvent event)

View.OnKeyListener : boolean onKey (View v, int keyCode, KeyEvent event)

View.OnClickListener : void onClick (View v)

View.OnFocusChangeListener : void onFocusChange (View v, boolean hasFocus)

## 대표적인 이벤트

- **터치 이벤트**
  - 화면을 손가락으로 누를 때 발생하는 이벤트
- **키 이벤트**
  - 키패드나 하드웨어 버튼을 누를 때 발생하는 이벤트
- **제스처 이벤트**
  - 터치 이벤트 중에서 일정 패턴을 만들어 내는 이벤트
- **포커스**
  - 뷰마다 순서대로 주어지는 포커스
- **화면 방향 변경**
  - 화면의 방향이 가로/세로로 바뀜에 따라 발생하는 이벤트

## 제스처를 통해 처리할 수 있는 이벤트

| 메소드 | 이벤트 유형 |
|---|---|
| **onDown()** | - 화면이 눌렸을 경우 |
| **onShowPress()** | - 화면이 눌렸다 떼어지는 경우 |
| **onSingleTapUp()** | - 화면이 한 손가락으로 눌렸다 떼어지는 경우 |
| **onSingleTapConfirmed()** | - 화면이 한 손가락으로 눌려지는 경우 |
| **onDoubleTap()** | - 화면이 두 손가락으로 눌려지는 경우 |
| **onDoubleTapEvent()** | - 화면이 두 손가락으로 눌려진 상태에서 떼거나 이동하는 등 세부적인 액션을 취하는 경우 |
| **onScroll()** | - 화면이 눌린 채 일정한 속도화 방향으로 움직였다 떼는 경우 |
| **onFling()** | - 화면이 눌린 채 가속도를 붙여 손가락을 움직였다 떼는 경우 |
| **onLongPress()** | - 화면을 손가락으로 오래 누르는 경우 |

**1.** 이벤트 처리

# GestureDetector를 이용한 이벤트 처리 예제

대표적인 이벤트 처리 예제

-손가락으로 화면 터치시의 이벤트 처리

-텍스트뷰의 이벤트를 받아 처리

메인 액티비티의
**XML** 레이아웃

-간단한 레이아웃 코드 작성

메인 액티비티 코드 작성

-이벤트 처리 코드 작성

# 메인 액티비티 코드 만들기

```
public boolean onTouchEvent(MotionEvent event) {
  if (mGestures != null) {
       return mGestures.onTouchEvent(event);
  } else {
      return super.onTouchEvent(event);
  }
}
…
public void onCreate(Bundle savedInstanceState) {
  …
  mGestures = new GestureDetector(new GestureDetector.SimpleOnGestureListener() {
  …
```

1 터치 이벤트를 제스처와 구분하여 처리

2 GestureDetector 객체 정의

Continued..

# 메인 액티비티 코드 만들기 (계속)

```
  public boolean onFling(MotionEvent e1, MotionEvent e2,
                         float velocityX, float velocityY) {
    textView01.append("\nonFling \n\tx = " + velocityX + "\n\ty=" + velocityY);
    return super.onFling(e1, e2, velocityX, velocityY);
  }

  public boolean onScroll(MotionEvent e1, MotionEvent e2,
                          float distanceX, float distanceY) {
    textView01.append("\nonScroll \n\tx = " + distanceX + "\n\ty =" + distanceY);
    return super.onScroll(e1, e2, distanceX, distanceY);
  }
});
```

3 Fling 이벤트 처리

4 Scroll 이벤트 처리

Continued..

# 메인 액티비티 코드 만들기 (계속)

```
  textView01.setOnLongClickListener(new View.OnLongClickListener() {
    public boolean onLongClick(View v) {
      textView01.append("\nonLongClick: " + v.toString());
      return true;
    }
  });

  textView01.setOnFocusChangeListener(new View.OnFocusChangeListener() {
        public void onFocusChange(View v, boolean hasFocus) {
          if (hasFocus) {
            textView01.append("\nonFocusChange, hasFocus : " + hasFocus);
          } else {
            textView01.append("\nonFocusChange, hasFocus : " + hasFocus);
          }
        }
    });
  }
}
```

6 LongClick 이벤트 처리

7 FocusChange 이벤트 처리

## 실행 화면


11:23
SampleEvents
오른쪽 영역을 터치하세요.
터치 모드가 바뀌었습니다. : true
onScroll
distanceX = -22.402893
distanceY = -25.823761
onScroll
distanceX = -13.228821
distanceY = -18.215912
onScroll
distanceX = -7.156494
distanceY = -33.737854
onScroll
distanceX = -5.843506
distanceY = -13.705383
onScroll
distanceX = 0.0
distanceY = -2.5567627
onFling
velocityX = 37.855404
velocityY=105.70049
onScroll
distanceX = 0.0
distanceY = -54.152985
onScroll
distanceX = 0.0
distanceY = -135.58743
onScroll
distanceX = -2.0
distanceY = -73.25958
onFling
velocityX = 55.10711

## 1. 이벤트 처리

# 키 입력 이벤트 처리하기

## 1. 이벤트 처리

뷰를 상속할 때 키 이벤트 처리를 위한 메소드 재정의

**boolean onKeyDown (int keyCode, KeyEvent event)**

**boolean onKey (View v, int keyCode, KeyEvent event)**

**[키를 눌렀을 때 전달되는 대표적인 키값]**

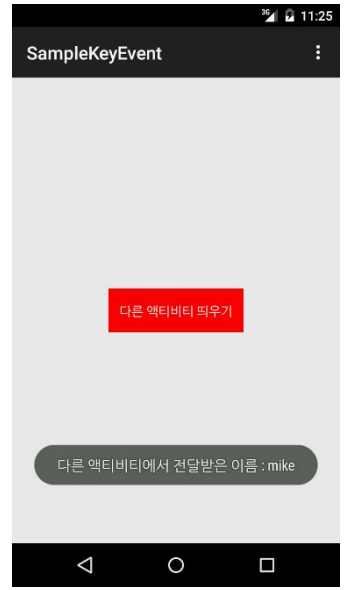
| 키 코드 | 설 명 |
|---|---|
| KEYCODE_DPAD_LEFT | - 왼쪽 화살표 |
| KEYCODE_DPAD_RIGHT | - 오른쪽 화살표 |
| KEYCODE_DPAD_UP | - 위쪽 화살표 |
| KEYCODE_DPAD_DOWN | - 아래쪽 화살표 |
| KEYCODE_DPAD_CENTER | - [중앙] 버튼 |
| KEYCODE_CALL | - [통화] 버튼 |
| KEYCODE_ENDCALL | - [통화 종료] 버튼 |
| KEYCODE_HOME | - [홈] 버튼 |
| KEYCODE_BACK | - [뒤로 가기] 버튼 |
| KEYCODE_VOLUME_UP | - [소리 크기 증가] 버튼 |
| KEYCODE_VOLUME_DOWN | - [소리 크기 감소] 버튼 |
| KEYCODE_0 ~ KEYCODE_9 | - 숫자 0부터 9까지의 키값 |

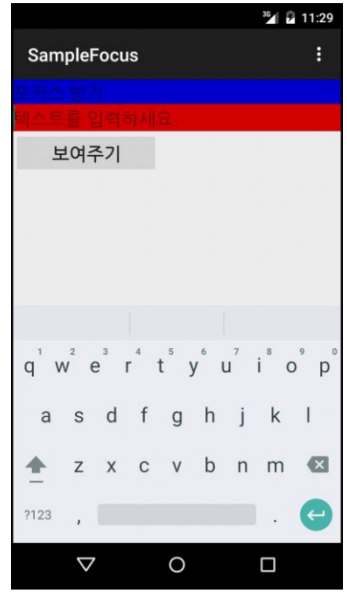
# BACK 버튼과 포커스 이벤트 처리 예제

## BACK 버튼과 포커스 예제

-BACK 버튼을 눌렀을 때의 이벤트 처리

-입력상자가 포커스를 받았을 때의 이벤트 처리




메인 액티비티의
코드 수정

-BACK 버튼 이벤트 처리하도록 수정

포커스 처리 XML로 정의

-포커스 처리를 위한 XML 정의

EditText의 속성 설정

-입력상자의 속성으로 설정

1. 이벤트 처리

# BACK 버튼 처리를 위한 액티비티 코드 만들기

```
public boolean onKeyDown(int keyCode, KeyEvent event) {
if(keyCode == KeyEvent.KEYCODE_BACK) {
close();
return true;
}
return false;
}

private void close() {
  Intent resultIntent = new Intent();
  resultIntent.putExtra("name", "mike");
  setResult(1, resultIntent);
  finish();
  }
}
```

1 하드웨어 [BACK] 버튼이 눌렸을 경우 새로 정의한 close() 메소드 호출

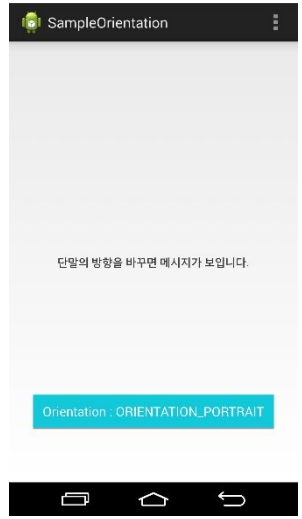
2 호출한 액티비티로 결과값 전송

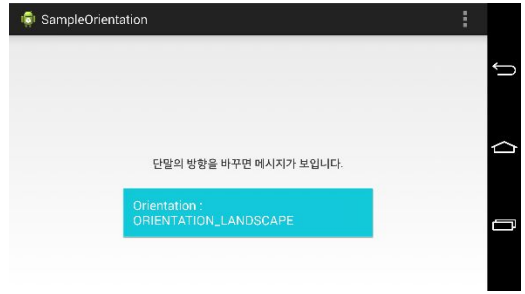
3 액티비티 없애기

# 단말 방향전환 이벤트 처리 예제

## 단말 방향전환 이벤트 예제

-단말 방향이 가로와 세로로 바뀌었을 때 이벤트 처리




## 매니페스트 속성 추가

-매니페스트의 액티비티 속성 추가

## 메인 액티비티 코드 작성

-가로와 세로 방향으로 바뀌었을 때 처리 코드 작성

1. 이벤트 처리

## 매니페스트 파일에 속성 추가

```
<activity android:name=".MainActivity"
          android:label="@string/app_name"
          android:configChanges="orientation|screenSize|layoutDirection"
          android:screenOrientation="sensor">
```

```
public void onConfigurationChanged(Configuration newConfig) {
  super.onConfigurationChanged(newConfig);

  if (newConfig.orientation == Configuration.ORIENTATION_LANDSCAPE) {
    Toast.makeText(this, "Orientation : ORIENTATION_LANDSCAPE",
          Toast.LENGTH_SHORT).show();
  } else if (newConfig.orientation == Configuration.ORIENTATION_PORTRAIT) {
    Toast.makeText(this, "Orientation : ORIENTATION_PORTRAIT",
          Toast.LENGTH_SHORT).show();
  }
}
```

❸ 가로 방향 전환 시 처리

❹ 세로 방향 전환 시 처리

액티비티를 가로 또는 세로 방향으로 고정

[Code]

```
<activity
   android:name=".MainActivity"
   android:theme="@android:style/Theme.NoTitleBar"
   android:screenOrientation="landscape"
   android:configChanges="orientation"
   >
</activity>
```

## [ References]


• 기본 서적

2016, 정재곤, “Do it! 안드로이드 앱 프로그래밍(개정3판)”, 이지스퍼블리싱(주)

• Android Website

http://www.android.com/

• Google Developer’s Conference

http://code.google.com/events/io/

• Android SDK Documentation