

# 이벤트(Event)

# 콜백 동작

---

## 콜백 메서드

- 이벤트를 처리하는 가장 쉬운 방법
- 특정 이벤트가 발생했을 경우 시스템에 의해 자동으로 호출되는 메서드
- 메서드에 코드를 작성하여 이벤트 발생 시의 동작을 정의

사용자와 상호 작용하는 주체는 뷰이며, 이벤트 콜백은 주로 뷰가 제공함  
뷰(View)를 상속받아서 `onTouchEvent` 메서드를 메서드를 재정의

## ➤ 대표적인 콜백 메소드

`boolean onTouchEvent (MotionEvent event)`

`boolean onKeyDown (int keyCode, KeyEvent event)`

`boolean onKeyUp (int keyCode, KeyEvent event)`

`boolean onTrackballEvent (MotionEvent event)`

# 콜백 작성

```
class MyView extends View {  
    public MyView(Context context) {  
        super(context);  
    }  
    public boolean onTouchEvent(MotionEvent event) {  
        super.onTouchEvent(event);  
        if (event.getAction() == MotionEvent.ACTION_DOWN) {  
            Toast.makeText(  
                MainActivity.this, "터치 이벤트 발생함",  
                Toast.LENGTH_SHORT  
            ).show();  
        }  
        return true;  
    }  
}
```

# 콜백 작성

---

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    View vw = new MyView(this);  
    setContentView(vw);  
}
```

# 리스너 인터페이스 구현

---

리스너(Listener) : 특정 이벤트를 처리하는 인터페이스.

View 클래스의 이너 인터페이스로 정의되어 있으며, 대응되는 이벤트를 받는 단 하나의 메서드가 선언되어 있음

## ➤ 대표적인 리스너 인터페이스

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

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

**OnClickListener** : void onClick (View v)

**OnLongClickListener** : boolean onLongClick (View v)

# 리스너 작성

```
class TouchListener implements View.OnTouchListener {
```

```
    @Override
```

```
    public boolean onTouch(View v, MotionEvent event) {
```

```
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
```

```
            Toast.makeText(
```

```
                MainActivity.this, "터치 이벤트 발생", Toast.LENGTH_SHORT
```

```
            ).show();
```

```
            return true;
```

```
        }
```

```
        return false;
```

```
    }
```

```
}
```

# 리스너 인터페이스 객체 생성 및 등록

```
public void onCreate(Bundle savedInstanceState) {
```

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    // 객체 생성
```

```
    TouchListener touchListener = new TouchListener();
```

```
    Button btn = (Button) findViewById(R.id.btn);
```

```
    // 리스너 등록
```

```
    btn.setOnTouchListener(touchListener);
```

```
}
```

# 리스너 인터페이스 구현 – 익명 클래스 활용

```
Button btn = (Button) findViewById(R.id.btn);
```

```
// 1. 리스너 등록
```

```
btn.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public boolean onTouch(View view, MotionEvent event) {  
        if (event.getAction() == MotionEvent.ACTION_DOWN) {  
            Toast.makeText(  
                MainActivity.this, "익명 터치 이벤트 발생", Toast.LENGTH_SHORT  
            ).show();  
            return true;  
        }  
        return false;  
    }  
});
```



# 이벤트 처리 우선순위

---

## ➤ 1 순위

뷰의 리스너

## ➤ 2 순위

뷰의 onTouchEvent 콜백 메서드

## ➤ 3 순위

액티비티의 onTouchEvent 콜백 메서드

# 리스너 - 1순위

```
View vw = new MyView(this);
vw.setOnTouchListener(new View.OnTouchListener() {
    @Override
    public boolean onTouch(View v, MotionEvent event) {
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
            Toast.makeText(
                MainActivity.this, "리스너에서 처리", Toast.LENGTH_SHORT).show();
            return true;
        }
        return false;
    }
});
setContentView(vw);
```

## 뷰의 콜백 - 2순위

```
protected class MyView extends View {  
    public MyView(Context context) {  
        super(context);  
    }  
    @Override  
    public boolean onTouchEvent(MotionEvent event) {  
        super.onTouchEvent(event);  
        if (event.getAction() == MotionEvent.ACTION_DOWN) {  
            Toast.makeText(MainActivity.this, "뷰의 콜백 메서드",  
                Toast.LENGTH_SHORT).show();  
            return true;  
        }  
        return false;  
    }  
}
```

# 액티비티의 콜백 - 3순위

```
public class MainActivity extends Activity {  
    public boolean onTouchEvent(MotionEvent event) {  
        super.onTouchEvent(event);  
        if (event.getAction() == MotionEvent.ACTION_DOWN) {  
            Toast.makeText(MainActivity.this, "액티비티의 콜백 메서드",  
                Toast.LENGTH_SHORT).show();  
            return true;  
        }  
        return false;  
    }  
}
```