

이벤트 처리

코틀린 안드로이드 앱 개발

강사: 배정만

이벤트 처리 프로세스

1. 레이아웃 연동 프로세스 (선언적 처리 프로세스)

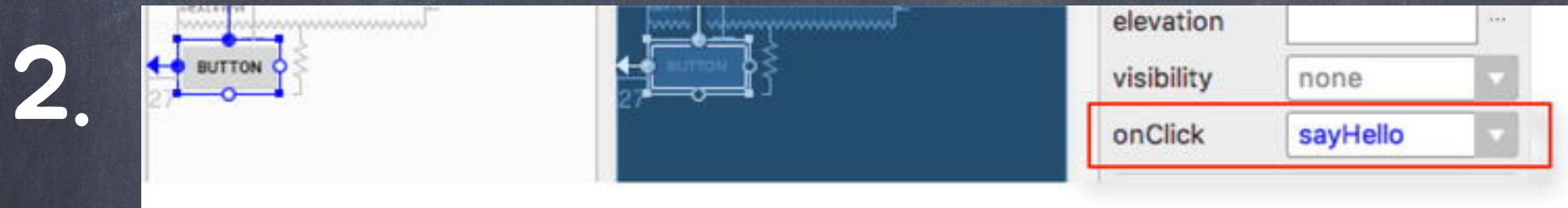
- 1.뷰 xml에 onClick 속성에 함수이름 지정
- 2.해당액티비티에 동일이름 이벤트의 기능함수작성
- 3.두개의 속성 연결

2.프로그램(코드) 이벤트 핸들러 (프로그래밍 프로세스)

- 1.findViewById로 xml로 작성된 뷰의 인스턴스를 가져옴
- 2.뷰 인스턴스에 onClickListner 를 지정
- 3.onClick 이벤트 함수를 오버라이딩
- 4.뷰 xml의 아이디로 onClick 함수에 기능작성

실습 - 뷰 연동 프로세스

1. activity_main.xml 레이아웃 디자인모드에서 버튼 뷰(Button) 선택



3. sayHello 함수 작성

- Hello Kotlin Android! 출력

실습 - 프로그래밍 방식

1. 버튼을 findViewById 로 가져옴

- `Button objButton = (Button) findViewById(R.id.button);`

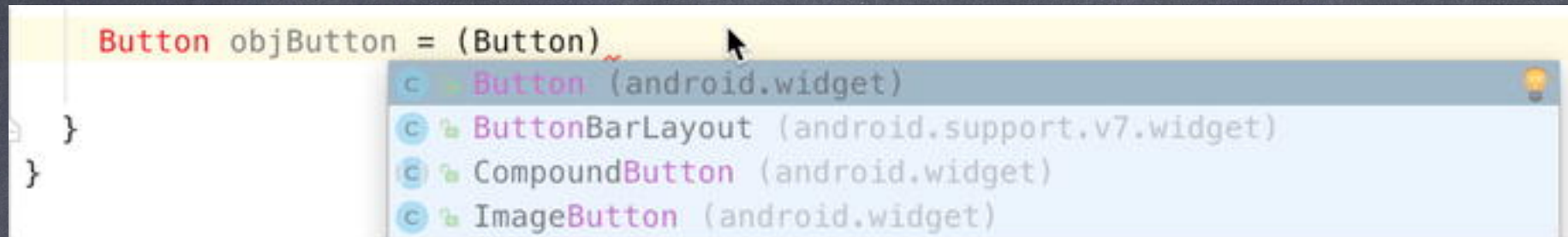
2. 해당 버튼뷰 변수에 리스너 등록

- `objButton.setOnClickListener(...)`

3. 리스너 오버라이드 함수 작성

- `Hello Kotlin Android! 출력`

버튼 참조 값 가져오기



```
import android.view.Button;
```

```
var mButton: Button = findViewById(R.id.button)
```

- MainActivity 클래스 정의 앞에 `import` 문을 이용하여 `android.view.Button` 추가

클릭 이벤트 리스너

```
objButton.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        // 여기에 코드 작성  
    }  
});
```

```
mButton.setOnClickListener(object: View.OnClickListener {  
    override fun onClick(v: View?) {  
  
    }  
}  
);
```

- 위는 자바 아래는 코틀린
- 안드로이드 스튜디오 자동완성으로 생성되는 클릭 이벤트 객체
- 추상 클래스를 재정의 하는 형태의 코드 블록

실습 해설 - Java

```
import ...
class MainActivity : AppCompatActivity() {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button objButton = (Button) findViewById(R.id.button); ❶
        assert objButton != null; ❷
        objButton.setOnClickListener(new View.OnClickListener() { ❸
            @Override ❹
            public void onClick(View view) { ❺
                System.out.println("Hello World");
            }
        });
    }
}
```


실습 해설 - Kotlin

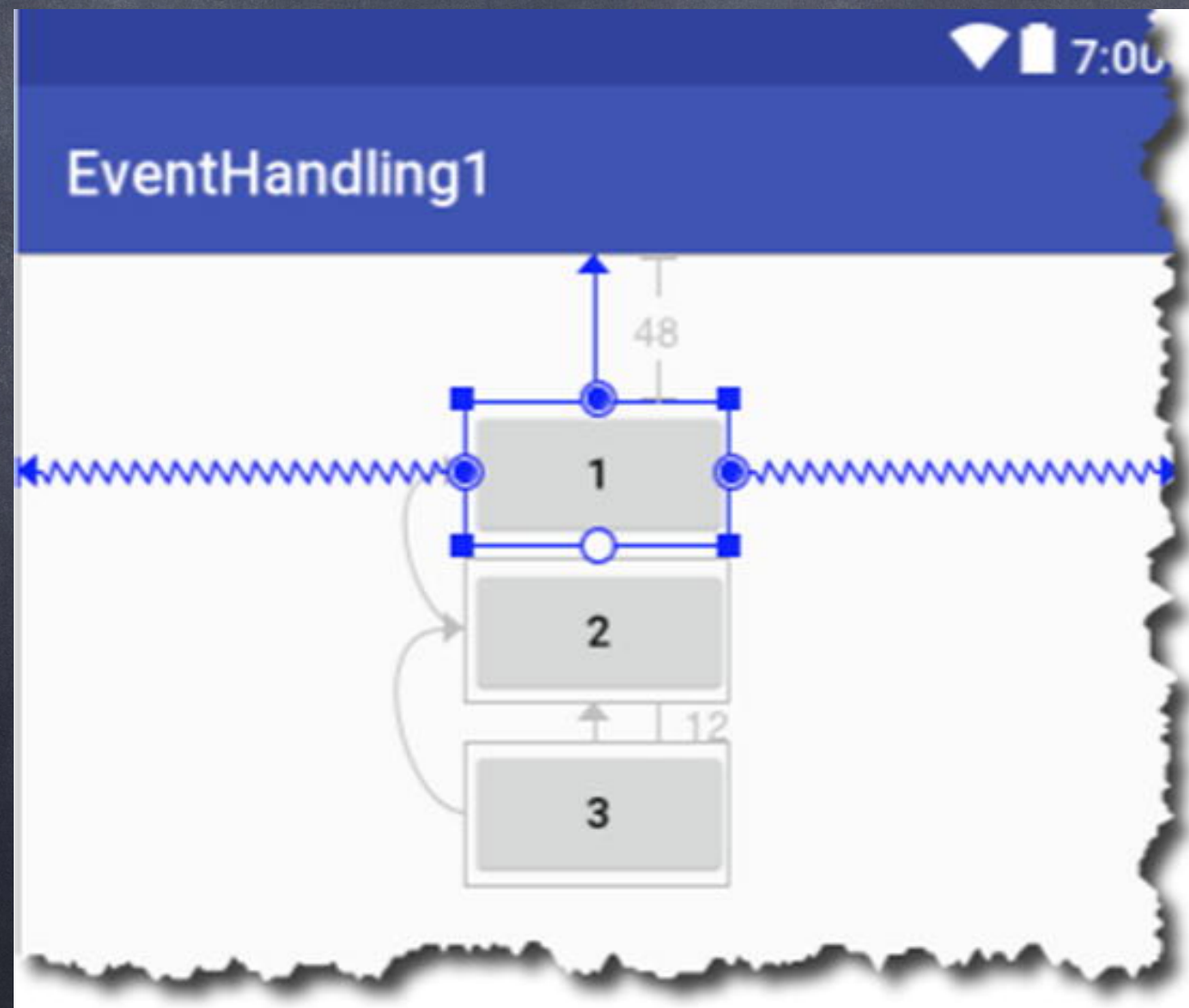
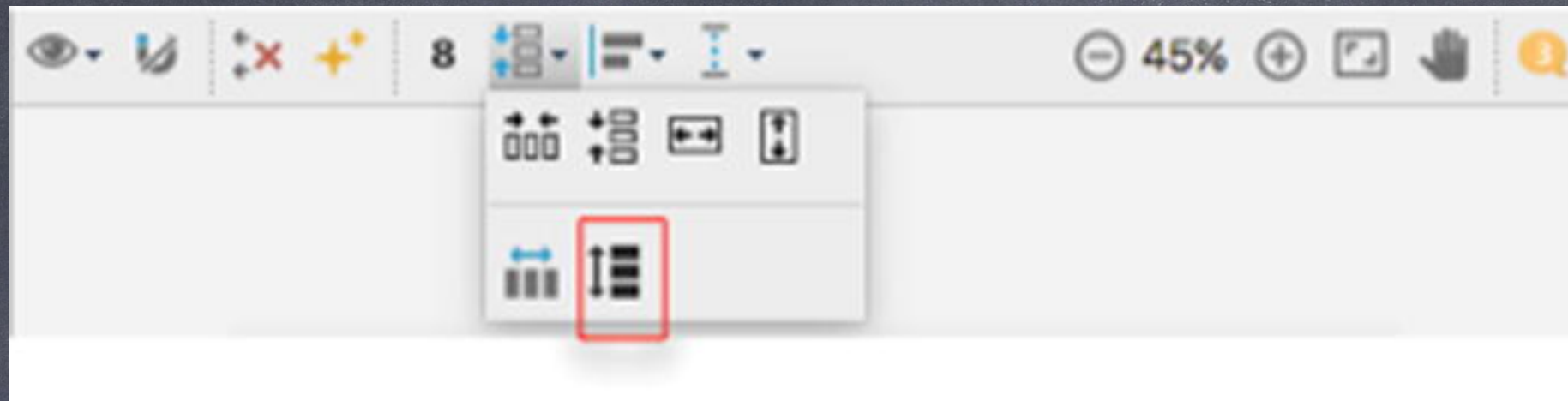
```
class MainActivity : AppCompatActivity() {  
    lateinit var mButton: Button  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
  
        mButton = findViewById(R.id.button)  
  
        mButton.setOnClickListener(object: View.OnClickListener {  
            override fun onClick(v: View?) {  
                println("Hello click")  
            }  
        })  
    }  
}
```


Three buttons

- 앱이름: Three buttons
- Company domain: 아무거나
(예: kotlin.study)
- Project location: 프로젝트 폴더 설정
- Include Kotlin 체크
- Minimum SDK: 23 (6.0)
- Empty Activity 선택
- MainActivity / Layout 생성 체크
- Backward Compatibility 체크

프로젝트 개요

1. 버튼 3개 추가
2. 수직 정렬
3. 화면 중앙 정렬
4. Inner Class 를 이용한 클릭 리스너 구현



레이아웃 xml

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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="com.thelogbox.eventhandling1.MainActivity">
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="48dp"
        android:text="1"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"/>
    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="4dp"
        android:text="2"
        app:layout_constraintStart_toStartOf="@+id/button1"
        app:layout_constraintTop_toBottomOf="@+id/button1"/>
    <Button
        android:id="@+id/button3"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="12dp"
        android:text="3"
        app:layout_constraintStart_toStartOf="@+id/button2"
        app:layout_constraintTop_toBottomOf="@+id/button2"/>
</android.support.constraint.ConstraintLayout>
```


Inner 클래스 정의

```
class MainActivity : AppCompatActivity() {  
    override fun onCreate(savedInstanceState: Bundle? { . . . }  
    private class ButtonHandler implements View.OnClickListener {  
  
    }  
}
```


ButtonHandler 구현

```
private inner class ButtonHandler :  
View.OnClickListener {  
    override fun onClick(view: View) {  
        when (view.id) {  
            R.id.button1 -> show("Button One")  
            R.id.button2 -> show("Button Two")  
            R.id.button3 -> show("Button 3")  
            else -> show("This should not happen")  
        }  
    }  
}
```


show() 메서드

```
internal fun show(message: String) {  
    Toast.makeText(this, message, Toast.LENGTH_LONG).show()  
    Log.i(javaClass.name, message)  
}
```

- Toast 는 작은 팝업 형태로 문자 피드백을 제공
- 현재 액티비티에 오버레이형태로 나타남
- 일정 시간동안 노출되고 사라짐
- 로그 클래스는 logcat 도구 창에서 info, debug, error 타입으로 보임