

안드로이드 SDK 개발 플랫폼 활용하기

토스트 사용하기

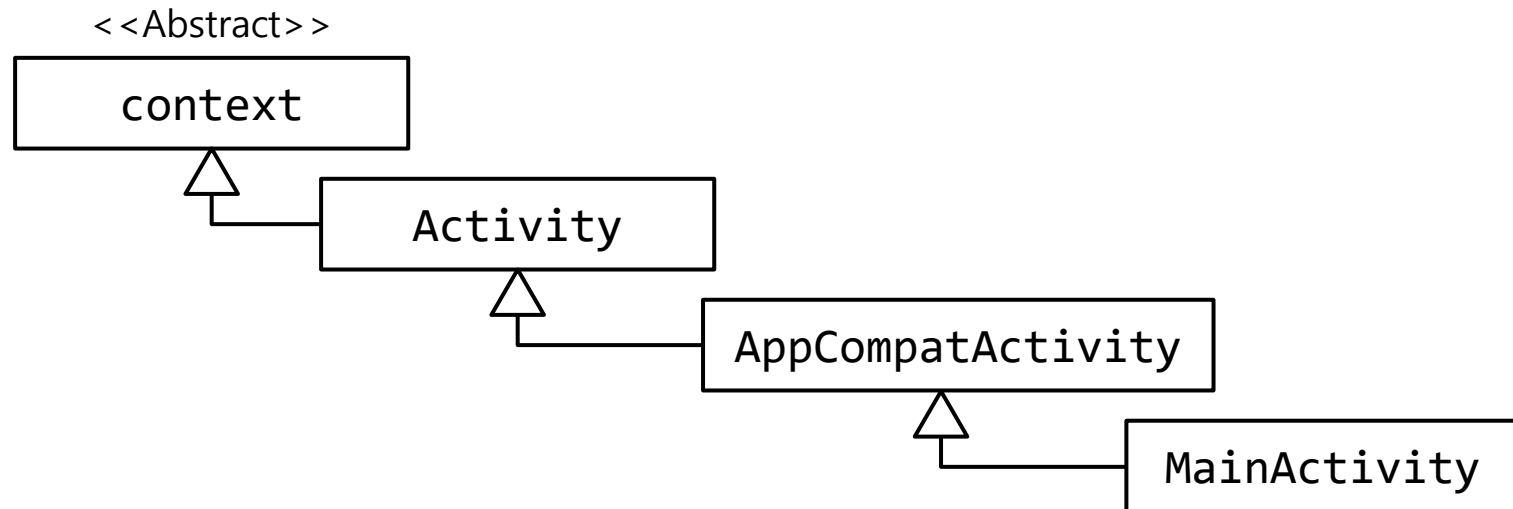
- 토스트는 사용자에게 임의의 안내 메시지를 표시하고 싶을 때 사용
 - 간단한 메시지 박스로 표시할 때
 - 디버그를 위한 메시지를 확인하고 싶을 때



- 네트워크 이벤트 등이 발생했을 때 토스트 주로 사용
 - `onClickListener`와 같은 인터페이스를 구현해서 사용
 - `Toast`는 첫번째 인자로 `Context`를 요구
- `Activity`가 `Context`의 자식이므로 `Activity`를 첫 번째 인자로 넣어주기 위해 `this`를 사용
 - 클래스 안에 포함되어있는 Inner Class나 또는 익명의 객체 인 경우 `this`로 가리키면 Outer Class인 `MainActivity`를 가리킬 수 없음
 - 이런 경우 `this` 키워드 앞에 명시적으로 클래스 이름을 지정해 주면 접근이 가능
 - `this` 일 경우 오류를 표시하지만 `MainActivity.this`로 코딩할 경우 외부의 클래스 객체에 직접 접근이 가능
- `Toast`를 생성할 때 사용한 `make` 메소드는 `static` 메소드
 - 정적 메소드의 경우에는 인스턴스를 생성하지 않고도 사용할 수 있음

- 토스트 생성 시 첫 번째 인자로 **context** 가 필요함
 - Activity가 바로 context를 구현하고 있으므로 Activity를 인자로 전달
- Toast를 생성할 때 사용한 *make()* 메소드는 static 메소드
 - 정적 메소드의 경우에는 인스턴스를 생성하지 않고도 사용할 수 있음
 - new 를 안해도 됨(간편함)

```
Toast.makeText( MainActivity.this , "Hello", Toast.LENGTH_SHORT).show();
```



- MainActivity가 상속받는 AppCompatActivity의 족보
 - "Context의 일종이다."
 - 출처 :
<https://developer.android.com/reference/android/support/v7/app/AppCompatActivity?hl=en>

```
public class AppCompatActivity
extends FragmentActivity implements AppCompatActivity,
TaskStackBuilder.SupportParentable, ActionBarDrawerToggle.DelegateProvider

java.lang.Object
└─ android.content.Context
    └─ android.content.ContextWrapper
        └─ android.view.ContextThemeWrapper
            └─ android.app.Activity
                └─ android.support.v4.app.FragmentActivity
                    └─ android.support.v7.app.AppCompatActivity
```

- 본 실습에서는 아래와 같은 순으로 실습을 진행한다.
 1. 기본 프로젝트 만들기
 2. 세로 비율을 8:2로 나누고 위쪽에 사진 표시
 3. 아래쪽 레이아웃을 가로로 8:2로 나누고 글자 입력창과 버튼 표시
 4. 버튼은 클릭 이벤트 처리
 5. 그림은 터치 이벤트 처리
 6. 각 이벤트 마다 토스트 띄우기

1. 세로 비율을 8:2로 나누기
 - 위쪽에 사진 표시
2. 아래쪽 레이아웃을 가로로 8:2로 나누기
 - 글자 입력창과 버튼 표시



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="안드로이드 APK 파일의 스키마 경로는 생략합니다."
    android:layout_width="match_parent" // 부모를 안벗어나게 최대화
    android:layout_height="match_parent" // 부모를 안벗어나게 최대화
    android:orientation="vertical"
    android:weightSum="10"
>

<!-- Top Image Layout -->
<LinearLayout
    android:layout_width="match_parent" // 부모를 안벗어나게 최대화
    android:layout_height="0dp"
    android:background="#000000"
    android:layout_weight="8"
>
```



```
<ImageView
    android:id="@+id/imageView"
    android:layout_width="match_parent" // 부모를 안벗어나게 최대화
    android:layout_height="match_parent" // 부모를 안벗어나게 최대화
    android:src="@drawable/momoland"
/>
</LinearLayout>

<!-- Bottom EditText Layout -->

<LinearLayout
    android:layout_width="match_parent" // 부모를 안벗어나게 최대화
    android:layout_height="0dp"
    android:layout_weight="2"
    android:background="#FFFF00"
    android:weightSum="10"
>
```

```
<EditText
    android:id="@+id/editText"
    android:layout_width="0dp"
    android:layout_height="match_parent" // 부모를 안벗어나게 최대화
    android:layout_weight="8"
    android:hint="Input Message"
/>
<Button
    android:id="@+id/button"
    android:layout_width="0dp"
    android:layout_height="match_parent" // 부모를 안벗어나게 최대화
    android:layout_weight="2"
    android:text="Push"
    android:textSize="22dp"
    android:textColor="#000000"
    android:gravity="center"
/>
```

```
</LinearLayout>
```

```
</LinearLayout>
```

- 화면 구현 완료 확인



- 리스너를 구현한 후 onTouch 이벤트에서 토스트 생성
 - Activity가 외부에 있는 클래스이므로 this 키워드 앞에 액티비티 클래스명 지정
- Inner Class 내에서 this로 가리키면 Outer Class인 MainActivity를 가리킬 수 없음
 - this 키워드 앞에 명시적으로 클래스 이름을 지정해 주면 접근 가능

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        imageView.setOnTouchListener(new View.OnTouchListener() {  
            @Override  
            public boolean onTouch(View v, MotionEvent event) {  
                Toast.makeText( MainActivity.this, "Hello", Toast.LENGTH_SHORT).show();  
                return false;  
            }  
        });  
    }  
}
```

- 터치 리스너를 이용해 터치 시 토스트를 띄워보는 예제
- Push Button을 누르면, EditText에 입력된 메시지를 토스트로 표시
 - 상단의 이미지를 터치해도 토스트를 표시하는 예제를 구현하자.
- 이미지를 터치해도 토스트가 표시
 - 아래쪽에 문자열을 입력하고 버튼을 눌러도 토스트가 표시



```
package com.iot.toasttest;

import android.support.v7.app.AppCompatActivity; // 액티비티 부모 클래스
import android.os.Bundle; // 액티비티 생성 번들
import android.view.MotionEvent;
import android.view.View; // 뷰계열 최상위 클래스
import android.widget.Button; // 버튼 뷰 사용
import android.widget.EditText; // 글자 입력창
import android.widget.ImageView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity { // 메인 화면
    private ImageView imageView;
    private EditText editText;
    private Button button;
    @Override // 부모 메소드 재정의
    protected void onCreate(Bundle savedInstanceState) { // 화면생성 이벤트
        super.onCreate(savedInstanceState); // 부모 생성자 호출
        setContentView(R.layout.activity_main); // 메인 화면 표시
    }
}
```

```
imageView = (ImageView)findViewById(R.id.imageView);
editText = (EditText)findViewById(R.id.editText);
button = (Button)findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() { // 클릭리스터 생성
    @Override // 부모 메소드 재정의
    public void onClick(View v) { // 클릭 이벤트 처리
        String msg = editText.getText().toString();
        editText.setText("");
        Toast.makeText(MainActivity.this, msg, Toast.LENGTH_SHORT).show();
    }
});
imageView.setOnTouchListener(new View.OnTouchListener() {
    @Override // 부모 메소드 재정의
    public boolean onTouch(View v, MotionEvent event) {
        Toast.makeText(MainActivity.this, "Hello",
            Toast.LENGTH_SHORT).show();
        return false;
    }
});
}
```

- 동작 확인
 - 버튼을 눌러도 토스트 표시
 - 사진을 터치해도 토스트 표시



- 떠있는 토스트 없애는 방법?
 - Toast 클래스에서 제공하는 cancel() 메소드
 - 현재 표시중인 토스트의 객체를 필드로 보관
 - 취소하고 싶을 때 cancel()호출
 - 그러면 표시되던 토스트를 제거 할 수 있음