● 액티비티가 출력되는 창(Window)의 다양한 설정

▶ 키보드 보이기와 숨김: 안드로이드 폰에서 제공하는 키보드는 하드웨어 키보드와 소프트 키보드로 구분된다. 하드웨어 키보드는 스마트폰에서 물리적으로 제공하는 키보드로 우리 앱에서 창을 제어하는 것과는 관련이 없다. 소프트웨어로 제공하는 소프트 키보드는 화면에서 사용자 입력을 위한 EditText나 AutoCompelteTextView 등이 포커스를 받는 순간 자동으로 아래에서 올라온다. 자바 코드에서 특정 순간에 키보드를 나타나게 하거나 사라지게 할 수 있으며, 이 기능을 제공하는 클래스가 InputMethodManager이다. InputMethodManager 클래스에서 제공하는 showSoftInput() 함수를 이용해 키보드를 보이게 하고, hideSoftInputFromWindow() 함수로 숨긴다.

```
InputMethodManager manager = (InputMethodManager) getSystemService(INPUT_METHOD_SERVICE)
```

showSoftInput(View view, int flags)	키보드 보임. 첫 번째 매개변수로 글이 입력될 뷰를 지칭하는데, 입력 대상이 되는 뷰에 포커스가 없는 상태라면 키보드가 나타나지 않는다. 따라서 현재 포커스가 없는 EditText에 입력되게 키보드를 보이려면 requestFocus() 함수를 이용하여 포커스를 요청한 후 showSoftInput() 함수로 키보드가 나타나게 하면 된다.
hideSoftInputFromWindow(IBinder windowToken, int flags)	키보드 숨김

주의할 점은 showSoftInput() 함수의 첫 번째 매개변수로 글이 입력될 뷰를 지칭하는데, 입력 대상이 되는 뷰에 포커스가 없는 상태라면 키보드가 나타나지 않는다. 따라서 현재 포커스가 없는 EditText에 입력되게 키보드를 보이려면 requestFocus() 함수를 이용하여 포커스를 요청한 후 showSoftInput() 함수로 키보드가 나타나게 하면 된다.

```
binding.showButton.setOnClickListener(view -> {
    binding.editText.requestFocus();
    manager.showSoftInput(binding.editText, InputMethodManager.SHOW_IMPLICIT);
});

binding.hideButton.setOnClickListener(view -> {
    manager.hideSoftInputFromWindow(getCurrentFocus().getWindowToken(), InputMethodManag
});
```

코드에서 특정 순간에 키보드를 보이거나 사라지게 하는 방법으로 toggleSoftInput() 함수를 이용하는 방법도 있다. 이 함수는 현 상황과 반대로 키보드를 제어한다. 현 상황에서 키보드가 안 보이면 보이게 하고, 보이면 사라지게 한다.

```
toggleSoftInput(int showFlags, int hideFlags);
```

▶ 키보드로 액티비티 화면 조정: 액티비티 화면에서 키보드가 보이면 액티비티가 차지하던 영역에 변화가 생긴다. 키보드에 일정 정도의 면적을 할당해야 하므로, 액티비티 자체를 어떻게 변경할 것인지를 설정해야 한다.

기본(설정 안 된 경우)	adjustUnspecified와 stateUnspecificed가 적용
adjustPan	키보드가 올라올 때 입력 EditText에 맞춰 화면을 위로 올림 (화면이 스크롤 업 된다)
adjustResize	키보드가 올라올 때 액티비티의 크기 조정. 액티비티 전체 크기가 shrink 한다.
adjustUnsepcified	시스템이 알아서 상황에 맞는 옵션 설정
stateHidden	액티비티 실행 시 키보드가 자동으로 올라오는 것 방지
stateVisible	액티비티 실행 시 키보드가 자동으로 올라옴
stateUnspecified	시스템이 적절한 키보드 상태를 설정하거나 테마에 따라 설정

```
<activity
    android:name=".KeyboardActivity"
    android:exported="true"
    android:windowSoftInputMode="adjustResize|stateHidden">
// 초기에 키보드가 자동으로 올라오지 않고,
// EditText에 포커스를 주는 순간 액티비티 화면의 크기가 재조정된다.
```

▶ 화면 방향: 스마트폰은 기기를 회전함에 따라 화면이 자동으로 회전하는데, 때로는 앱 화면을 세로나 가로 방향으로 고정해야 할 때가 있다. 액티비티의 screenOrientation 속성만으로 쉽게 액티비티의 방향을 고정할 수 있다.

```
<activity
    android:name=".KeyboardActivity"
    android:exported="true"
    android:screenOrientation="portrait">
```

만약 사용자에 의한 액티비티의 화면 전환이 가능한 상황이라면, 화면이 회전하는 순간을 자바 코드에서 감지할 수 있도록 해 주는 속성이 configChanges이다. 이 속성은 액티비티의 각종 환경이 변경되었을 때, 액티비티의 콜백 함수를 호출하기 위해 사용된다. 속성값으로 orientation|screenSize를 지정하면, 화면이 회전할 때 콜백 함수가 호출되게 하고, 이 함수를 코드로 작성해 주면 된다. configChanges 속성에 의해 등록된 환경이 바뀌는 순간 호출되는 액티비티의 콜백 함수는 onConfigurationChanged()이다.

```
<activity
    android:name=".KeyboardActivity
    android:exported="true"
    android:configChanges="orientation|screenSize">
```

```
@Override
public void onConfigurationChanged(@NonNull Configuration newConfig) {
    super.onConfigurationChanged(newConfig);
    if (newConfig.oreintation = Configuration.ORIENTATION_PORTRAIT) {
        Log.d("kkang", "portrait....");
    } else {
        Log.d("kkang", "landscape....");
    }
}
```

▶ 전체 화면: 액티비티를 전체 화면으로 표시한다는 것은 액티비티의 콘텐츠 영역을 화면 전체에 나오게 한다는 의미이다. 액션바는 출력되지 않으며, 액티비티의 내용이 상태바(status bar)의 영역까지 차지하도록 화면을 구성한다. 액티비티를 전체화면으로 설정하기 위해서는 먼저 액티비티의 액션바가 출력되지 않도록 테마를 적용해야 한다.

```
<activity
    android:name=".KeyboardActivity"
    android:exported="true"
    android:theme="@style/Theme.MaterialComponents.DayNight.NoActionBar">
```

그리고 화면 상단의 시간이 나오는 상태바 영역과 하단의 시스템 버튼이 출력되는 네비게이션바 영역에 윈도우 내용이 나오도록 코드에서 제어해 주어야 한다. 이 제어 방법은 API Level 29까지는 Window 객체의 setFlags() 함수를 이용하였지만 deprecated 되었으며, API Level 30버전부터 WindowInsetsController 객체에 의해 설정된다. WindowInsetsController 객체의 hide() 함수로 윈도우에 기본으로 출력되는 상태바와 네비게이션바가 나오지 않도록 매개변수에 지정하여 설정했으며, 원한다면 이 중 하나만 지정해도 된다.

```
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.R) {
    Window window = getWindow();
    WindowInsetsController controller = window.getInsetsController();
    if (controller != null) {
        controller.hide(WindowInsets.Type.statusBars()|WindowInsets.Type.navigationBars(
    } else {
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowManager.L
    }
}
```

```
// activity_main.xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <Button
        android:id="@+id/lab3Button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="toggle keyboard"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="#FF0000"/>
    <TextView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1"
        android:background="#0000FF"/>
    <EditText
        android:id="@+id/lab3EditText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

</LinearLayout>
```

```java
// MainActivity.java

package com.android.practiceactivitysettings;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Build;
import android.os.Bundle;
import android.view.Window;
import android.view.WindowInsets;
import android.view.WindowInsetsController;
import android.view.WindowManager;
import android.view.inputmethod.InputMethodManager;

import com.android.practiceactivitysettings.databinding.ActivityMainBinding;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        ActivityMainBinding binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        // 키보드가 보이거나 숨겨지게 하는 기능을 제공하는 클래스
        InputMethodManager manager = (InputMethodManager) getSystemService(INPUT_METHOD_

        binding.lab3Button.setOnClickListener(view -> {
            // 현상황과 반대로 키보드를 제어한다. 안 보이면 보이게 하고, 보이면 사라지게 한다.
            manager.toggleSoftInput(InputMethodManager.SHOW_IMPLICIT, 0);
        });

        // API30부터는 WindowInsetsController 객체에 의해 상태바와 네비게이션 바가 나오지 않
        // 그 전 버전에서는 Windows 객체의 setFlag() 함수를 이용한다.
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.R) {
            Window window = getWindow();
            WindowInsetsController controller = window.getInsetsController();
            if (controller != null) {
                controller.hide(WindowInsets.Type.statusBars()|WindowInsets.Type.navigat
            } else {
                getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN, WindowM
            }
        }
    }
}
```

```
// AndroidManifest.xml

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.android.practiceactivitysettings">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.MaterialComponents.DayNight.DarkActionBar"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity"
            android:exported="true"
            android:theme="@style/Theme.MaterialComponents.DayNight.NoActionBar"
            android:windowSoftInputMode="adjustResize">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Resources: 깡쌤의 안드로이드 프로그래밍