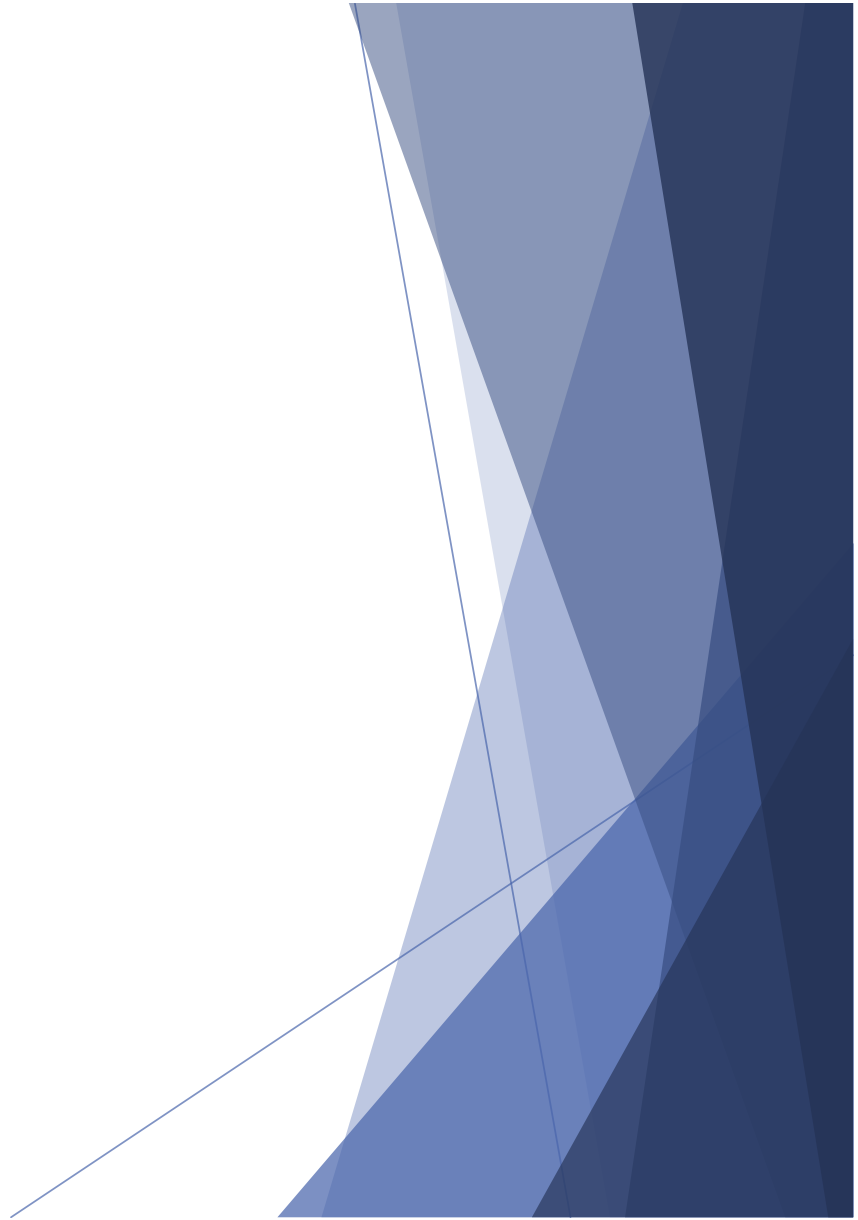




Android Programming

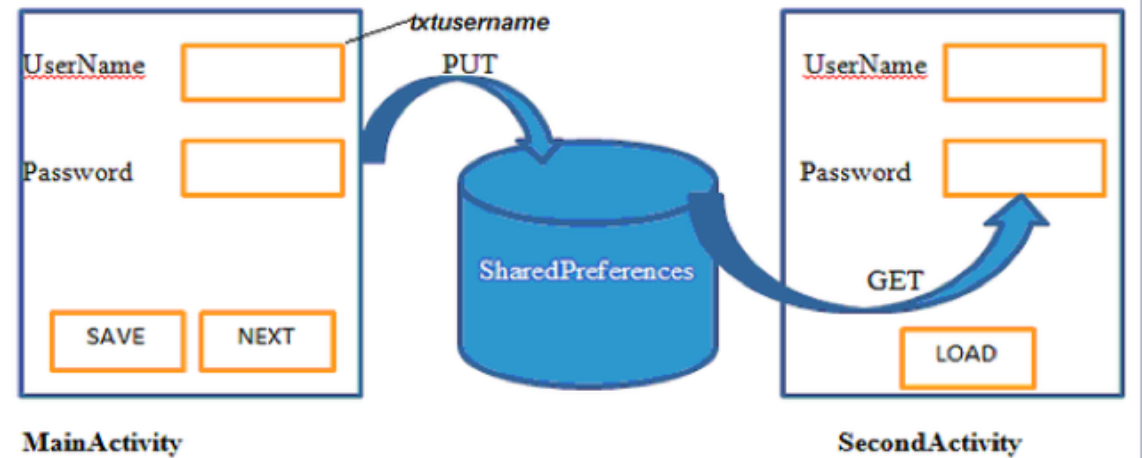
Extra: SharedPreferences

SharedPreferences



SharedPreferences

- ▶ SharedPreferences : 앱의 데이터를 영속적으로 저장하기 위한 클래스
 - ▶ 데이터를 간단하게 키-값(key-value) 쌍으로 저장
 - ▶ Database 처럼 복잡한 데이터는 다룰 수 없으나 앱 내에서 생성되고 사용되는 간단한 형태의 데이터를 다루기에는 적합
 - ▶ 내부적으로 데이터는 XML 형태로 저장되나, 데이터를 읽고 쓰는 메서드를 제공하므로 개발자가 직접 해당 로직을 작성할 필요 없음



SharedPreferences

: 객체의 획득

Mode 값

- MODE_PRIVATE
- MODE_WORLD_READABLE
- MODE_WORLD_WRITABLE

▶ Case 1: 액티비티 전용 - getPreferences

- ▶ 파일명을 지정하지 않으므로, 자동으로 액티비티 이름이 파일명이 됨
- ▶ 하나의 액티비티만을 이용한 저장 공간이 됨

```
SharedPreferences sharedPref = getPreferences(Context.MODE_PRIVATE);
```

▶ Case 2: 컴포넌트간 공유 - getSharedPreferences

- ▶ 파일명을 직접 지정 가능
- ▶ 데이터가 많을 때 각각의 파일로 나누어 구분 저장하고자 할 때 주로 이용

```
SharedPreferences sharedPref = getSharedPreferences("settings", Context.MODE_PRIVATE);
```

▶ Case 3: PreferenceManager.getDefaultSharedPreferences

- ▶ 패키지 명이 파일명이 됨

```
SharedPreferences sharedPref = PreferenceManager  
.getDefaultSharedPreferences(this);
```

SharedPreferences

: 데이터의 저장

- ▶ SharedPreferences로 데이터를 저장시 Editor 클래스의 setter 메서드를 이용
 - ▶ putBoolean(String key, boolean value)
 - ▶ putFloat(String key, boolean value)
 - ▶ putInt(String key, int value)
 - ▶ putLong(String key, long value)
 - ▶ putString(String key, String value)
- ▶ 데이터 저장 시작 전에는 edit 메서드, 데이터를 최종 저장할 때는 commit 메서드 호출

```
SharedPreferences sharedPref =  
PreferenceManager.getDefaultSharedPreferences(this);  
SharedPreferences.Editor editor = sharedPref.edit();  
editor.putString("username", "hong");  
editor.putInt("age", 24);  
editor.commit();
```

SharedPreferences

: 데이터의 획득

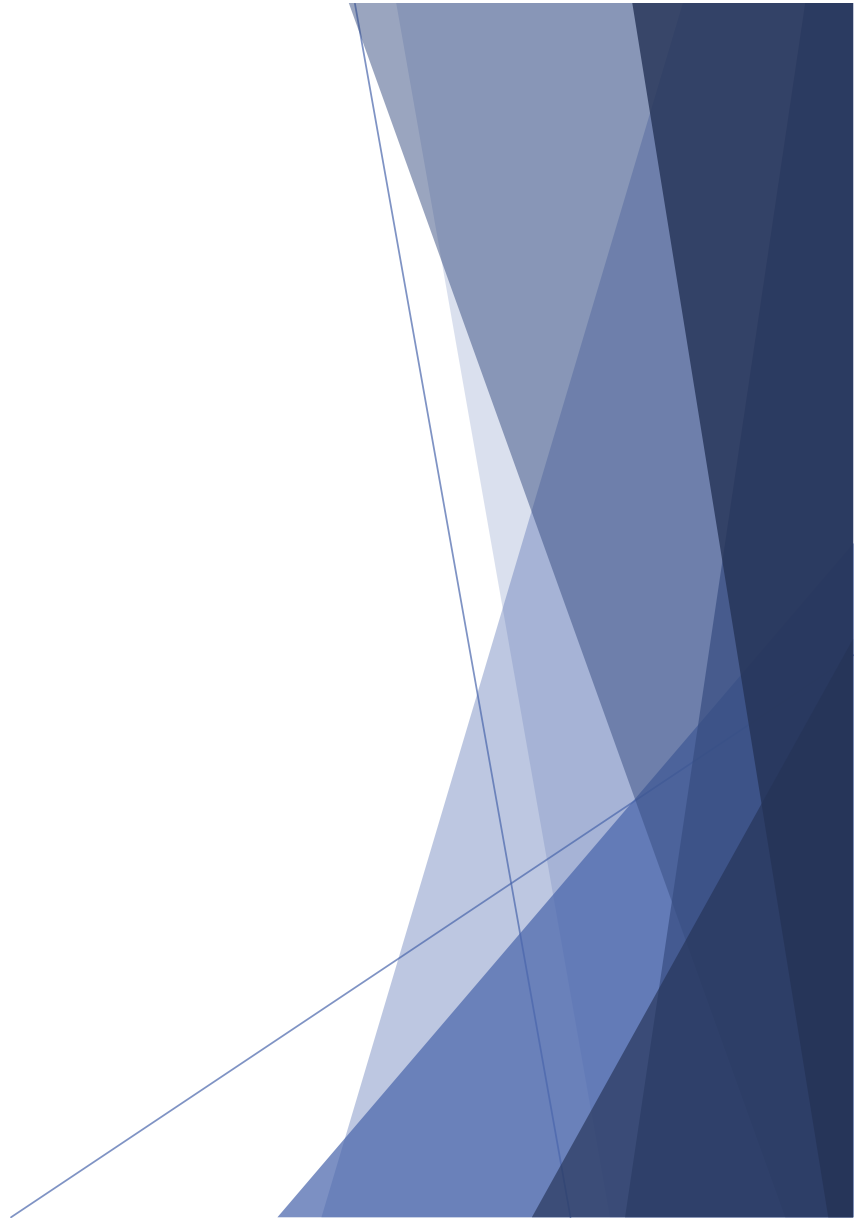
- ▶ SharedPreferences에서 데이터를 획득할 때는 getter 메서드를 활용
 - ▶ `getBoolean(String key, boolean defaultValue)`
 - ▶ `getFloat(String key, float defaultValue)`
 - ▶ `getInt(String key, int defaultValue)`
 - ▶ `getLong(String key, long defaultValue)`
 - ▶ `getString(String key, String defaultValue)`

```
SharedPreferences sharedPref =  
PreferenceManager.getDefaultSharedPreferences(this);  
String username = sharedPref.getString("username", "");  
int age = sharedPref.getInt("age", 0);
```



SharedPreferences

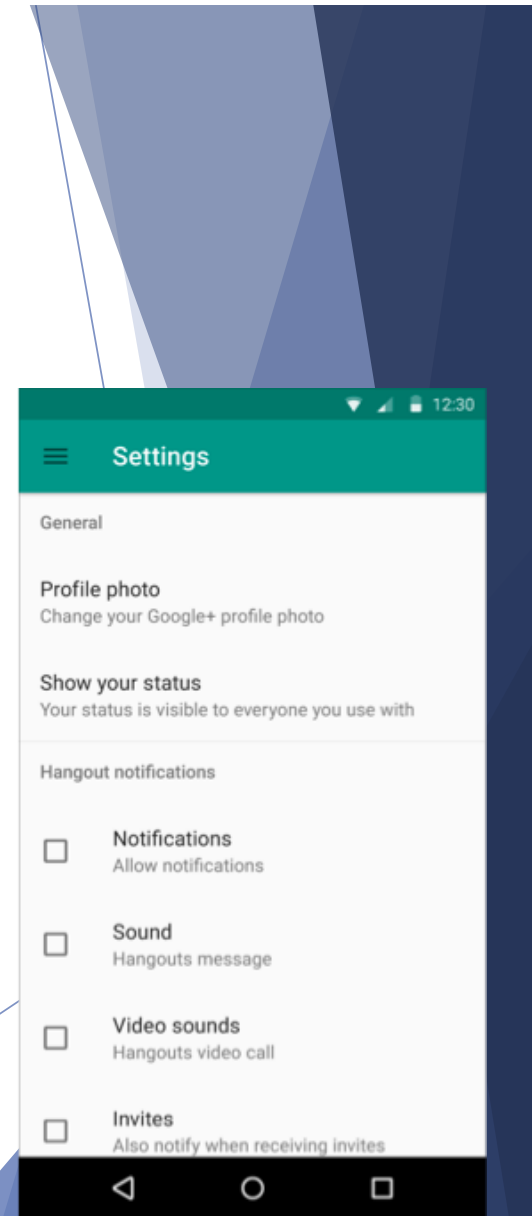
SharedPreferences와 PreferencesFragment



PreferenceFragment

: 앱 설정의 자동화

- ▶ 대부분의 앱에서는 사용자가 어떤 환경으로 앱을 이용할 것인지를 결정하는 설정 기능을 제공
 - ▶ 예) 알림 받기, 사용자 로그인 정보 저장 등
- ▶ 이러한 설정들은 Activity로 화면을 구성하고 그 안에서 발생한 사용자 이벤트들을 처리하여 설정 데이터를 영속적으로 저장해야 함
 - ▶ 이 부분의 작업을 대행해 주는 것이 앱 설정 자동화
- ▶ 안드로이드에서는 표준화된 설정 화면을 구성해주는 PreferenceFragment를 제공
 - ▶ /res/xml 에 설정 화면을 구성할 xml을 만들고, 이를 사용

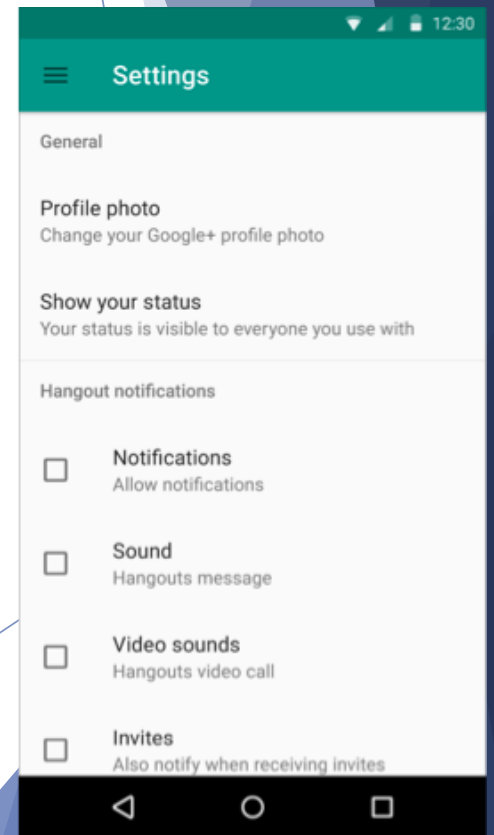


PreferenceFragment

: 앱 설정의 자동화

▶ Preference XML의 구성

- ▶ <PreferenceScreen> : 설정 화면의 단위. 중첩 가능하며 내용은 별도의 화면으로 출력
- ▶ <PreferenceCategory> : 설정 여러 개를 시각적으로 묶기 위함
- ▶ <CheckBoxPreference> : 체크박스가 나오는 설정
- ▶ <EditTextPreference> : 텍스트 입력을 위한 설정
- ▶ <ListPreference> : 항목 선택 다이얼로그를 위한 설정
- ▶ <MultiSelectListPreference> : 다중 선택 다이얼로그를 위한 설정
- ▶ <RingtonePreference> : 알람음 선택을 위한 설정
- ▶ <SwitchPreference> : 스위치 설정



PreferenceFragment

: 앱 설정의 자동화

▶ 공통 속성

- ▶ android:title : 항목의 제목을 설정
- ▶ android:summary : 항목의 설명을 설정
- ▶ android:defaultValue : Preference의 기본값을 위한 설정

android:title

android:summary

자동 로그인
자동 로그인을 설정합니다.



사용자 아이디
사용자 아이디를 설정합니다.

PreferenceFragment

: 앱 설정의 자동화

▶ ListPreference, MultiSelectListPreference

- ▶ android:entry : 목록 표시를 위한 문자열 배열(표시용)
- ▶ android:entryValue : 목록 저장을 위한 문자열 배열(값)

▶ EditTextPreference

- ▶ android:selectAllOnFocus : 입력 대화상자가 뜰 때, 내용 전체를 미리 선택 여부
- ▶ android:singleLine : 한줄 입력 여부

PreferenceFragment

: 앱 설정의 자동화

- ▶ PreferenceFragment를 상속

```
public class SettingsPreferenceFragment extends PreferenceFragment
```

- ▶ onCreate 메서드에 Preference XML을 연결

```
addPreferencesFromResource(R.xml.settings);
```

- ▶ SharedPreferences를 받아옴

```
SharedPreferences pref =  
    PreferenceManager.getDefaultSharedPreferences(getActivity());
```

- ▶ OnSharedPreferencesChangeListener 를 구현후 연결

```
pref.registerOnSharedPreferencesChangeListener(prefListener);
```

PreferenceFragment

: 앱 설정의 자동화

- ▶ PreferenceFragment는 Fragment이므로 Activity에 담아 사용해야 함
 - ▶ 루트 레이아웃은 필요 없고, fragment를 root 엘리먼트로 사용 권장

```
<?xml version="1.0" encoding="utf-8"?>

<fragment
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    class="com.example.android.sharedprefexample.SettingsPreferenceFragment"
    xmlns:android="http://schemas.android.com/apk/res/android" />
```

PreferenceFragment

: 앱 설정의 자동화

[실습] SharedPrefExample

1) 프로젝트를 생성하고 다음의 작업을 수행합니다.

- /res/values 에 arrays.xml 을 복사해 넣습니다.

2) MainActivity에 OptionsMenu를 구성하고, 다음 메뉴를 만듭니다

- Settings : 선택하면 SettingsActivity를 호출합니다.

3) /res/xml 에 settings.xml 을 만들고 예제 화면과 같이 PreferenceScreen을 구성합니다.

4) PreferenceFragment를 상속받은 SettingsPreferenceFragment를 구현하고 다음 작업을 진행합니다.

- addPreferencesFromResource 메서드로 settings.xml을 불러옴
- onCreate 메서드에서 만약 기존에 설정된 값이 있을 경우(사용자 ID, 내 OS) 미리 설정된 값을 Summary에 표시

5) OnSharedPreferenceChanged 리스너를 구현하고 등록합니다.

6) SettingsPreferenceFragment를 SettingsActivity의 레이아웃에 등록합니다.

