



Shared Preferences(실습)

배 희호 교수
경북대학교
스마트IT과



데이터 저장 기법

■ Android에서 제공하는 데이터 저장 기법

- Preference(프리퍼런스)
- 내부/외부 저장소를 사용한 파일
- SQLite 데이터베이스
- Network를 사용한 Web 상에서의 데이터 저장 방법이 있음





데이터 저장 기법



- Shared Preferences(공유 프리퍼런스)
 - 키-값 쌍(key-value pair)으로 **사적이고 기초적인 데이터**를 저장
- Internal Storage(내부 저장소)
 - 사적인 데이터를 장치 메모리에 저장
- External Storage (외부 저장소)
 - 공유 데이터를 공유 외부 저장소에 저장
- SQLite Databases(SQLite 데이터베이스)
 - 구조화된 데이터를 사적인 데이터베이스에 저장
- Network Connection(네트워크 연결)
 - 데이터를 네트워크 Server를 사용하여서 Web에 저장



Preference

- Application에 간단한 설정 Option을 추가하고 사용자가 이를 변경할 때, 변경된 데이터를 저장하고 관리하게 하려면 어떻게 해야 할까요?
 - 파일을 따로 만들어야 할까요?
- Android에서는 App이나 그 Component(Activity, Service 등)의 환경 설정 정보를 영구적으로 저장하고 관리할 수 있게끔 '공유 프리퍼런스(Shared Preference)'라는 장치를 제공
- Preference는 DataBase처럼 복잡한 자료가 아닌 간단한 정보를 저장하는 일에 사용
 - 예) 인사말, 폰트, 소리같은 설정 정보를 저장하고 복원하는 용도로 사용
- Windows 환경의 레지스트리나 LINUX 환경의 Setting 파일 정도에 대응되는 개념임



Preference

- App을 개발하다 보면 각종 **환경 설정 값들을 보관**해야 할 경우가 있음
 - 예) 게임 앱인 경우 사운드 on/off, 진동 on/off, 화면 밝기 설정 등을 보관 해둬야 함
- 이를 위해 Android에서는 공유 프리퍼런스(Shared Preferences)라는 매우 편리한 클래스를 제공
- 해당 클래스는 마치 HashMap 클래스를 이용하여 원하는 데이터를 **키와 값 형태로** 손쉽게 추가 및 제거할 수 있음
- 이뿐만 아니라 **내장 메모리에 파일로 저장되기 때문에 영구히 보관**할 수도 있음



Preference

- Preference는 /data/data에 위치하며 각 Application마다 /data/data [애플리케이션의 패키지명]/shared_prefs 폴더 내에 저장되어 있음
- Preference는 키 이름(Keyname)-저장된 값(value)의 쌍(pair)으로 저장
- XML 포맷의 텍스트 파일에 정보를 저장
=> Setting 파일이나 INI 파일에 더 가까움
- 애플리케이션에서 정보를 유지하기 위해 Preference를 사용할 때는 Preference를 저장하는 위치와 저장된 Preference를 읽어오는 위치 및 Preference를 사용하는 방법을 알아야 함



Preference



■ Preference 용도

- Application의 환경 설정 정보를 영구적으로 저장
 - 사용자의 옵션 선택 사항이나 프로그램 자체의 구성 정보를 주로 저장
- 하나의 Application내에서도 정보를 유지하기 위해 Activity간의 정보 교환 및 공유
 - 한쪽 Activity에서 Preference의 정보를 수정하면 다른 Activity도 수정된 값이 적용
 - 그러나 Preference는 Application의 고유 정보이기 때문에 외부에서는 읽을 수 없음
- UI 화면 정보를 저장했다가 복원하는 용도로 사용 하기도 함



Preference

■ Preference 장점

- Preference는 상대적으로 용량이 작고 Activity간 정보 교환에도 활용이 가능함
- 데이터 보관이 가장 손쉽고 간편하며 다른 프로그램에서는 접근할 수 없다는 장점이 있음
- Preference스는 “응용 프로그램의 환경 설정 정보를 저장”하는 용도와 “액티비티 간의 정보 교환”에 사용됨
- UI 화면 정보를 저장했다가 복원하는 용도로 사용하기도 함

■ Preference 단점

- Preference는 DB나 파일에 비해 그 사용법이 매우 간단하고 쉬움. 하지만 Preference의 가장 큰 단점은 “**설정값을 XML 형태로 저장하므로 처리속도가 느림**”



Preference



■ Shared Preference를 얻는 방법

- ✓ `getSharedPreferences(name, mode)`
- ✓ `getPreferences(mode)`
- ✓ `PreferenceManager.getDefaultSharedPreferences(Context context)`

■ Preference 파일에서 값을 읽거나 쓰는 메소드

- ✓ `getBoolean(String key, boolean defValue)`
- ✓ `getInt(String key, int defValue)`
- ✓ `getString(String key, String defValue)`

- ✓ `putBoolean(String key, boolean value)`
- ✓ `putInt(String key, int value)`
- ✓ `putString(String key, String value)`



Preference



- Preference를 사용하는 3가지 방법
 - PreferenceActivity를 상속받지 않고 Preference를 사용할 경우
 - PreferenceActivity를 상속받아 Preference를 사용할 경우
 - 허니콤(3.0)이후 버전에서 Preference를 사용할 경우



Preference

- PreferenceActivity를 상속받지 않고 Preference를 사용할 경우
 - Preference 가져오기
 - Preference 읽어 오기
 - Preference에 기록하기



Preference



- Preference 가져오기
 - Preference는 SharedPreferences 클래스를 사용해서 구현
 - SharedPreferences 클래스는 기본 데이터 타입(primitive data type)의 키와 값의 쌍으로 이루어진 형태로 정보를 저장하고 얻어내는 것을 제공
 - SharedPreferences 클래스는 boolean, float, int, long, 그리고 strings 타입의 데이터를 사용할 수 있음



Preference

- SharedPreferences를 통해 데이터를 저장하기 위해, 우선 SharedPreferences의 인스턴스를 얻어와야 하는데 SharedPreferences 인스턴스를 얻는 3가지 방법
 - getPreferences(int mode)
 - 하나의 Activity에서만 사용하는 SharedPreferences를 생성
 - 생성되는 SharedPreferences 파일은 해당 Activity의 이름으로 생성
 - 하나의 Activity에서 사용할 목적으로 생성하였지만, 생성한 SharedPreferences의 이름을 getSharedPreferences() 메소드에 인자값으로 전달하면 다른 Activity에서도 저장된 데이터에 접근 가능



Preference

- `getSharedPreferences("String name", int mode)`
 - 특정 이름을 가지는 `SharedPreferences`를 생성

```
SharedPreferences prefs  
    = getSharedPreferences("userPrefs", MODE_PRIVATE);
```



Preference



- getSharedPreferences의 인수
 - 첫번째 인수는 Preference를 저장할 XML 파일의 이름
 - mode 인수 (이 파일은 공유 모드)

mode 인수	설명
MODE_PRIVATE	자기 앱 내에서 사용 외부 앱에서 접근 불가
0	외부 앱에서 읽기, 쓰기 가능
MODE_WORLD_READABLE	외부 앱에서 읽기 공유
MODE_WORLD_WRITEABLE	외부 앱에서 쓰기 공유

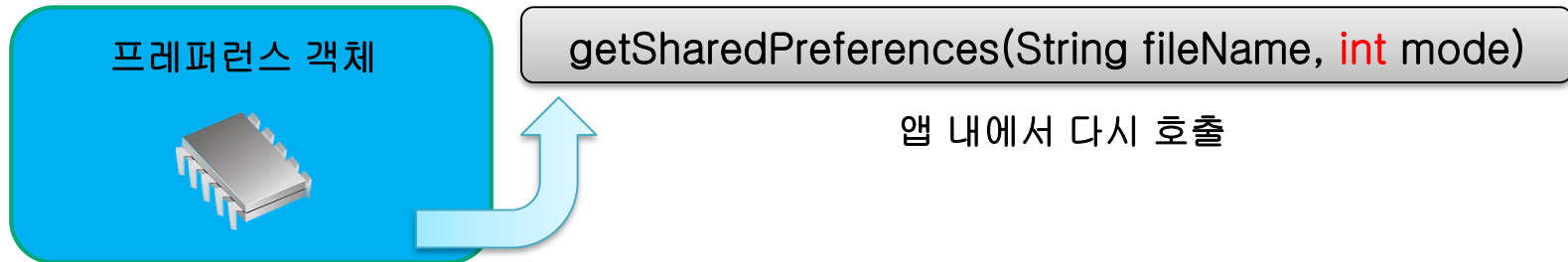
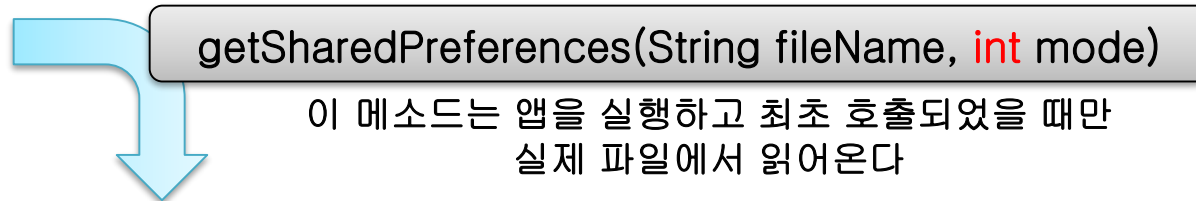


Preference

■ Preference 파일 읽어 오기



프리퍼런스
파일



따라서 최초 `getSharedPreferences` 메소드가 호출될 때 저장된 데이터가
많다면 느리다

`getSharedPreferences` 메소드는 최초 호출 시에만 느릴 수 있기 때문에
앱이 실행될 때 앱이 실행될 때 로딩 화면에서 미리 호출해 두면 좋다



Preference

■ Preference 읽어 오기

- Preference는 키와 값의 쌍으로 데이터를 저장
- 키는 정보의 이름이며 값엔 정보의 default 값을 저장
- 여러 타입의 자료형이 존재하는데, 웬만한 기본 타입에 대해서는 읽기 메소드가 모두 제공
- 가장 자주 사용되는 타입은 정수, 문자열, 논리형

프리퍼런스 읽기/쓰기	설명
<code>int getInt("String key", int defValue)</code>	정수형 읽기
<code>String getString ("String key", String defValue)</code>	문자열 읽기
<code>boolean getBoolean("String key", boolean defValue)</code>	논리형 읽기
<code>void putInt("String key", int defValue)</code>	정수형 쓰기
<code>void putString ("String key", String defValue)</code>	문자열 쓰기
<code>void putBoolean("String key", boolean defValue)</code>	논리형 쓰기



Preference

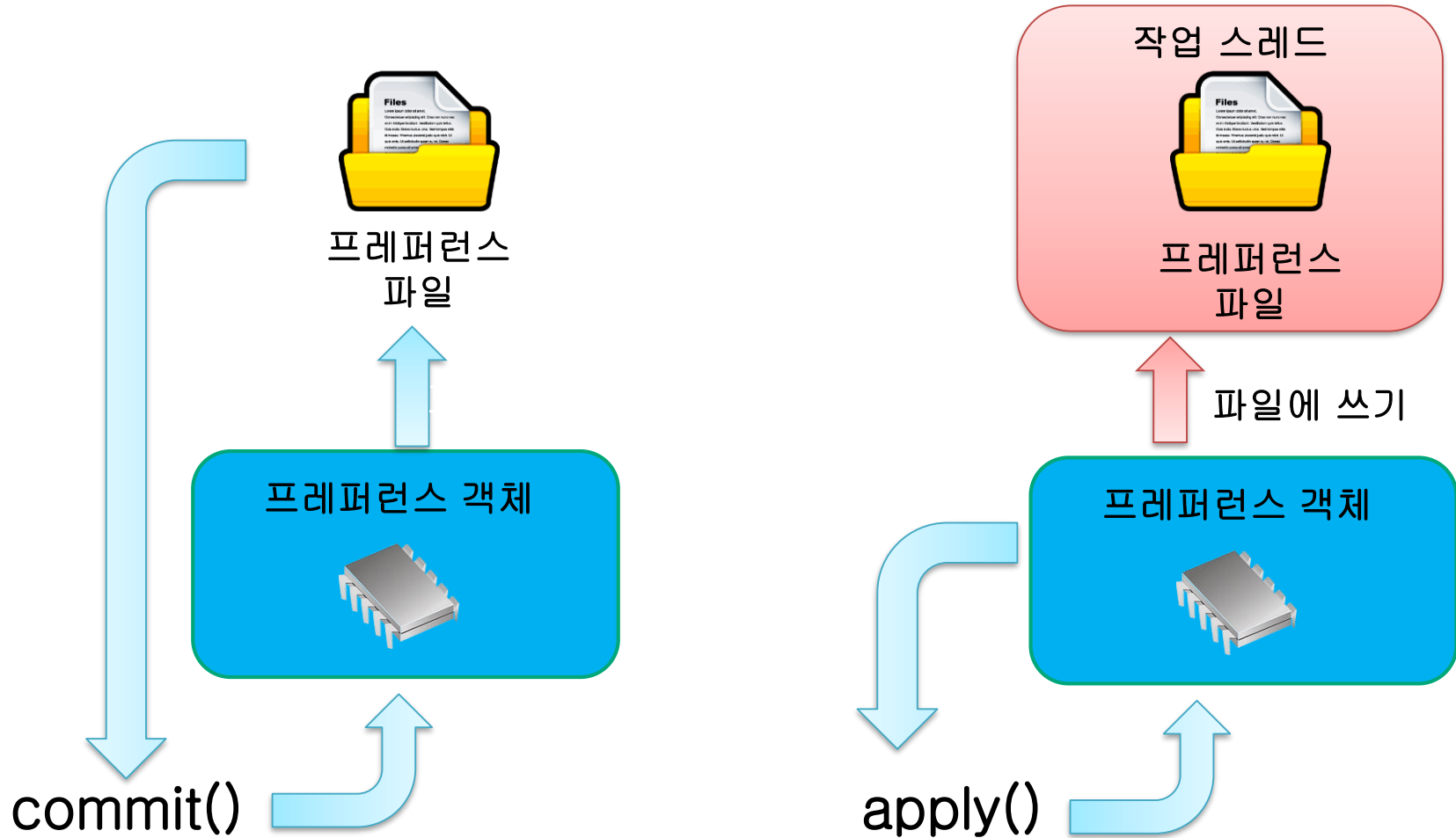


- Preference 읽어 오기
 - 읽어올 데이터의 타입이 다를 뿐이며 사용하는 방법은 모두 같음
 - key 인수로 데이터의 이름을 지정하고 defValue 인수로 값이 없을 때 적용할 디폴트를 지정
 - Preference에 키가 존재하면 해당 Preference에 기록되어 있는 값이 반환되고 키가 없을 때에는 두 번째 인수로 지정한 defValue가 반환
 - 최초 실행될 때에는 Preference가 생성되기 전이므로 디폴트가 반환



Preference

■ Preference 저장하는 함수 commit과 apply



주의 : apply 함수는 안드로이드 API 9 진저브레드부터 추가었다





Preference

- 그렇다면 commit과 apply 중 어떤 것을 사용해야 하나?
 - apply는 작업 스레드에서 실제 파일에 저장하기 때문에 속도가 매우 빠름
 - 아직 작업 스레드가 파일에 저장하는 것을 마치지 않은 상태에서 Preference를 참조해도 전혀 문제가 되지 않음
 - Preference의 데이터 참조는 이미 생성된 Preference 객체를 참조하기 때문
 - apply 단점 한 가지



Preference

- Preference를 저장하는 위치와 저장된 Preference를 읽어오는 위치
 - Preference를 저장하는 위치
 - 액티비티의 onPause() 메소드
 - Preference에 정보를 저장할 때
 - 액티비티가 포그라운드(foreground)에서 백그라운드(background)로 교체되는 지점인 onStop() 메소드에 기술되어야 함
 - 저장된 Preference를 읽어오는 위치
 - 액티비티의 onCreate() 메소드
 - 저장된 Preference의 정보를 얻어내려고 할 때
 - 액티비티가 생성 지점인 onCreate() 메소드에서 함



Preference

- Preference는 SharedPreferences 클래스를 사용해서 구현
- SharedPreferences 클래스는 기본 데이터 타입(primitive data type)의 키와 값의 쌍으로 이루어진 형태로 정보를 저장하고 읽어내는 것을 제공

getSharedPreferences()메소드의 사용 예

```
SharedPreferences prefs  
    = getSharedPreferences("userPrefs", MODE_PRIVATE);
```

- SharedPreferences 클래스는 boolean, float, int, long, 그리고 strings 타입의 데이터를 사용할 수 있음



Preference

- SharedPreferences 객체를 얻어내기
 - Preference에 정보를 저장하거나 저장된 정보를 얻어낼 때는 공통적으로 getSharedPreferences() 메소드 또는 getPreferences() 메소드 사용
 - getPreferences() 메소드
 - 액티비티내에서 오직 한 개의 preference 파일만 필요한 경우에 사용
 - 이 경우 이름은 정의 하지 않음

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



Preference

- SharedPreferences 객체를 얻어내기
 - getSharedPreferences() 메소드
 - 액티비티내에서 여러 개의 preference 파일이 필요할 경우 이 메소드를 사용
 - Preference에 이름을 부여해 preference 파일을 식별
 - 일반적으로 이 메소드를 사용해 preference 객체를 얻어냄

```
SharedPreferences prefs =  
getSharedPreferences("userPrefs", MODE_PRIVATE);
```




Preference

- SharedPreferences 객체를 얻어내기
 - PreferenceManager.getDefaultSharedPreferences(Context context)
 - 전달된 context가 기본으로 사용하는 preference로부터 SharedPreferences에 저장된 데이터들을 가져와 사용하려고 할 때 사용

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



Preference



- Preference에 정보를 저장
 - SharedPreferences 클래스의 edit() 메소드를 호출해 SharedPreferences.Editor 객체를 얻어냄
 - SharedPreferences.Editor 객체가 프로퍼티를 사용해서 preference에 정보를 저장할 수 있음

```
SharedPreferences.Editor prefsEditor = prefs.edit();
```



Preference

- Preference에 정보를 저장

- SharedPreferences.Editor 클래스의 putBoolean() 또는 putString()과 같은 putXxx(key, value) 메소드를 사용해 key와 value의 쌍으로 정보를 저장

- 이때 key가 프로퍼티의 이름, value가 프로퍼티의 값

```
prefsEditor.putString("id", id);
```

- 추가 또는 변경된 프로퍼티를 commit() 메소드를 사용해 preference에 한 번에 반영

```
prefsEditor.commit();
```



Preferences

- Preference에 정보를 얻어냄
 - SharedPreferences 클래스의 edit() 메소드를 호출해 SharedPreferences.Editor 객체를 얻어낸 후
 - getBoolean() 또는 getString()과 같은 getXxx(key, defValue) 메소드를 사용해서 정보를 얻어냄
 - 이때 Xxx는 저장된 preference의 정보의 데이터 타입으로 int 타입일 경우 getInt() 메소드를 String 타입일 경우 getString() 메소드를 사용
 - key 저장된 프로퍼티의 이름이고, defValue는 해당 preference의 key가 존재하지 않을 경우 리턴되는 값

```
String id = pref.getString("id", "");
```



Preference

■ 프리퍼런스 사용

```
//저장된 프리퍼런스의 정보를 읽어옴
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    ...생략...
    //SharedPreferences 객체를 얻어냄
    SharedPreferences prefs = getSharedPreferences(PREFS_NAME,0);
    //저장된 정보를 읽어옴
    currentViewMode = prefs.getInt("view_mode", DAY_VIEW_MODE);
}
```



Preference 예제 0

- 다음과 같이 기억할 데이터를 입력하고 저장하면 종료했다가 다시 실행하여도 저장되어 있는 앱을 만들어보자

Preference00

결과 :

변수 자료형	데이터 입력	저장	불러오기
String	저장할 문자열 입력	저장	불러오기
Boolean	<input checked="" type="radio"/> ture <input type="radio"/> false	저장	불러오기
Int	저장할 정수 입력	저장	불러오기
Float	저장할 실수 입력	저장	불러오기
Long	저장할 정수 입력	저장	불러오기

CLEAR



Preference 예제 0

■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:background="#88ff0000">
```



Preference 예제 0

■ 사용자 인터페이스

```
<TextView
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_weight="1"
    android:gravity="center"
    android:paddingLeft="5dp"
    android:text="결과 : " />
```

```
<TextView
    android:id="@+id/result"
    android:layout_width="match_parent"
    android:layout_height="50dp"
    android:layout_weight="1"
    android:background="#88cccccc"
    android:gravity="center"
    android:paddingLeft="5dp"
    android:text="" />
```

```
</LinearLayout>
```




Preference 예제 0

■ 사용자 인터페이스

<LinearLayout

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_marginTop="10dp">
```

<TextView

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_weight="1"  
android:gravity="center"  
android:paddingLeft="5dp"  
android:text="변수 자료형" />
```

<TextView

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_weight="1"  
android:gravity="center"  
android:text="데이터 입력" />
```



Preference 예제 0

■ 사용자 인터페이스

```
<TextView  
    android:layout_width="80dp"  
    android:layout_height="wrap_content"  
    android:gravity="center"  
    android:text="저장" />
```

```
<TextView  
    android:layout_width="80dp"  
    android:layout_height="wrap_content"  
    android:layout_marginLeft="2dp"  
    android:gravity="center"  
    android:text="불러오기" />
```

```
</LinearLayout>
```



Preference 예제 0



■ 사용자 인터페이스

<LinearLayout

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_marginTop="10dp">
```

<TextView

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_weight="1.2"  
android:paddingLeft="2dp"  
android:text="String" />
```

<EditText

```
android:id="@+id/edit01"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_weight="0.8"  
android:hint="저장할 문자열 입력" />
```



Preference 예제 0

■ 사용자 인터페이스

```
<Button
    android:id="@+id/save01"
    android:layout_width="80dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="2dp"
    android:text="저장" />
```

```
<Button
    android:id="@+id/load01"
    android:layout_width="80dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="2dp"
    android:text="불러오기" />
```

```
</LinearLayout>
```



Preference 예제 0



■ 사용자 인터페이스

<LinearLayout

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_marginTop="10dp">
```

<TextView

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_weight="1.2"  
android:paddingLeft="2dp"  
android:text="Boolean" />
```

<RadioGroup

```
android:id="@+id/radioGroup"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_gravity="center"  
android:layout_weight="0.9"  
android:orientation="horizontal">
```



Preference 예제 0

■ 사용자 인터페이스

```
<RadioButton
    android:id="@+id/radio1"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:checked="true"
    android:text="ture" />

<RadioButton
    android:id="@+id/radio2"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="false" />
</RadioGroup>
```



Preference 예제 0

■ 사용자 인터페이스

```
<Button  
    android:id="@+id/save02"  
    android:layout_width="80dp"  
    android:layout_height="wrap_content"  
    android:text="저장" />
```

```
<Button  
    android:id="@+id/load02"  
    android:layout_width="80dp"  
    android:layout_height="wrap_content"  
    android:layout_marginLeft="2dp"  
    android:text="불러오기" />
```

```
</LinearLayout>
```



Preference 예제 0

■ 사용자 인터페이스

<LinearLayout

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_marginTop="10dp">
```

<TextView

```
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_weight="1.2"  
android:paddingLeft="5dp"  
android:text="Int" />
```

<EditText

```
android:id="@+id/edit03"  
android:layout_width="match_parent"  
android:layout_height="wrap_content"  
android:layout_weight="0.8"  
android:hint="저장할 정수 입력"  
android:inputType="number" />
```




Preference 예제 0

■ 사용자 인터페이스

```
<Button  
    android:id="@+id/save03"  
    android:layout_width="80dp"  
    android:layout_height="wrap_content"  
    android:text="저장" />
```

```
<Button  
    android:id="@+id/load03"  
    android:layout_width="80dp"  
    android:layout_height="wrap_content"  
    android:layout_marginLeft="2dp"  
    android:text="불러오기" />
```

```
</LinearLayout>
```



Preference 예제 0

■ 사용자 인터페이스

<LinearLayout

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="10dp">
```

<TextView

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_weight="1.2"  
    android:paddingLeft="5dp"  
    android:text="Float" />
```

<EditText

```
    android:id="@+id/edit04"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_weight="0.8"  
    android:hint="저장할 실수 입력"  
    android:inputType="numberDecimal" />
```



Preference 예제 0

■ 사용자 인터페이스

```
<Button  
    android:id="@+id/save04"  
    android:layout_width="80dp"  
    android:layout_height="wrap_content"  
    android:text="저장" />
```

```
<Button  
    android:id="@+id/load04"  
    android:layout_width="80dp"  
    android:layout_height="wrap_content"  
    android:layout_marginLeft="2dp"  
    android:text="불러오기" />
```

```
</LinearLayout>
```



Preference 예제 0

■ 사용자 인터페이스

<LinearLayout

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="10dp">
```

<TextView

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_weight="1.2"  
    android:paddingLeft="5dp"  
    android:text="Long" />
```

<EditText

```
    android:id="@+id/edit05"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_weight="0.8"  
    android:hint="저장할 정수 입력"  
    android:inputType="number" />
```



Preference 예제 0



■ 사용자 인터페이스

```
<Button
    android:id="@+id/save05"
    android:layout_width="80dp"
    android:layout_height="wrap_content"
    android:text="저장" />
```

```
<Button
    android:id="@+id/load05"
    android:layout_width="80dp"
    android:layout_height="wrap_content"
    android:layout_marginLeft="2dp"
    android:text="불러오기" />
```

```
</LinearLayout>
```

```
<Button
    android:id="@+id/clear"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:text="clear"/>
```

```
</LinearLayout>
```



Preference 예제 0

■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {  
    final String PREFERENCE = "Sharepreference";  
    final String key01 = "key01";  
    final String key02 = "key02";  
    final String key03 = "key03";  
    final String key04 = "key04";  
    final String key05 = "key05";  
    boolean isBoolean = true;  
    EditText inputString, inputInt, inputFloat, inputLong;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        final TextView result = findViewById(R.id.result);  
    }  
}
```



Preference 예제 0

■ MainActivity.JAVA

```
inputString = findViewById(R.id.edit01);  
RadioGroup group = findViewById(R.id.radioGroup);  
inputInt = findViewById(R.id.edit03);  
inputFloat = findViewById(R.id.edit04);  
inputLong = findViewById(R.id.edit05);
```

```
Button saveString = findViewById(R.id.save01);  
Button saveBoolean = findViewById(R.id.save02);  
Button saveInt = findViewById(R.id.save03);  
Button saveFloat = findViewById(R.id.save04);  
Button saveLong = findViewById(R.id.save05);
```

```
Button loadString = findViewById(R.id.load01);  
Button loadBoolean = findViewById(R.id.load02);  
Button loadInt = findViewById(R.id.load03);  
Button loadFloat = findViewById(R.id.load04);  
Button loadLong = findViewById(R.id.load05);  
Button clear = findViewById(R.id.clear);
```



Preference 예제 0

■ MainActivity.JAVA

```
group.setOnCheckedChangeListener(  
    new RadioGroup.OnCheckedChangeListener() {  
        @Override  
        public void onCheckedChanged(RadioGroup group, int checkedId) {  
            switch (checkedId) {  
                case R.id.radio1 :  
                    isBoolean = true;  
                    break;  
                case R.id.radio2 :  
                    isBoolean = false;  
            }  
        }  
    });
```




Preference 예제 0

■ MainActivity.JAVA

```
saveString.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        String test = inputString.getText().toString();  
        if (test.equals(""))  
            Toast.makeText(getApplicationContext(), "입력해주세요",  
                                Toast.LENGTH_SHORT).show();  
        else  
            setPreference(key01, test);  
    }  
});  
  
saveBoolean.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        setPreference(key02, isBoolean);  
    }  
});
```



Preference 예제 0

```
saveInt.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        String test = inputInt.getText().toString();  
        if (test.equals(""))  
            Toast.makeText(getApplicationContext(), "입력해주세요",  
                                Toast.LENGTH_SHORT).show();  
        else  
            setPreference(key03, Integer.parseInt(test));  
    }  
});  
saveFloat.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        String test = inputFloat.getText().toString();  
        if (test.equals(""))  
            Toast.makeText(getApplicationContext(), "입력해주세요",  
                                Toast.LENGTH_SHORT).show();  
        else  
            setPreference(key04, Float.parseFloat(test));  
    }  
});
```



Preference 예제 0

■ MainActivity.JAVA

```
saveLong.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String test = inputLong.getText().toString();
        if (test.equals(""))
            Toast.makeText(getApplicationContext(), "입력해주세요",
                                   Toast.LENGTH_SHORT).show();
        else
            setPreference(key05, Long.parseLong(test));
    }
});

loadString.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        result.setText("String : " + getPreferenceString(key01));
    }
});
```



Preference 예제 0

■ MainActivity.JAVA

```
loadBoolean.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        result.setText("Boolean : " + getPreferenceBoolean(key02));  
    }  
});  
loadInt.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        result.setText("Int : " + getPreferenceInt(key03));  
    }  
});  
loadFloat.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        result.setText("Float : " + getPreferenceFloat(key04));  
    }  
});
```



Preference 예제 0

■ MainActivity.JAVA

```
loadLong.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        result.setText("Long : " + getPreferenceLong(key05));  
    }  
});  
  
clear.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        setPreferenceClear();  
    }  
});  
}
```



Preference 예제 0

■ MainActivity.JAVA

```
public void setPreference(String key, boolean value){
    SharedPreferences pref = getSharedPreferences(
        PREFERENCE, MODE_PRIVATE);
    SharedPreferences.Editor editor = pref.edit();
    editor.putBoolean(key, value);
    editor.apply();
    editor.commit();
}

public void setPreference(String key, String value){
    SharedPreferences pref = getSharedPreferences(
        PREFERENCE, MODE_PRIVATE);
    SharedPreferences.Editor editor = pref.edit();
    editor.putString(key, value);
    editor.apply();
    editor.commit();
}
```



Preference 예제 0

■ MainActivity.JAVA

```
public void setPreference(String key, int value){  
    SharedPreferences pref = getSharedPreferences(  
                                                PREFERENCE, MODE_PRIVATE);  
    SharedPreferences.Editor editor = pref.edit();  
    editor.putInt(key, value);  
    editor.apply();  
    editor.commit();  
}
```

```
public void setPreference(String key, float value){  
    SharedPreferences pref = getSharedPreferences(  
                                                PREFERENCE, MODE_PRIVATE);  
    SharedPreferences.Editor editor = pref.edit();  
    editor.putFloat(key, value);  
    editor.apply();  
    editor.commit();  
}
```



Preference 예제 0

■ MainActivity.JAVA

```
public void setPreference(String key, long value){
    SharedPreferences pref = getSharedPreferences(
        PREFERENCE, MODE_PRIVATE);
    SharedPreferences.Editor editor = pref.edit();
    editor.putLong(key, value);
    editor.apply();
    editor.commit();
}

public boolean getPreferenceBoolean(String key){
    SharedPreferences pref = getSharedPreferences(
        PREFERENCE, MODE_PRIVATE);
    return pref.getBoolean(key, true);
}

public String getPreferenceString(String key){
    SharedPreferences pref = getSharedPreferences(
        PREFERENCE, MODE_PRIVATE);
    return pref.getString(key, "");
}
```




Preference 예제 0

■ MainActivity.JAVA

```
public int getPreferenceInt(String key){  
    SharedPreferences pref = getSharedPreferences(  
        PREFERENCE, MODE_PRIVATE);  
    return pref.getInt(key, 0);  
}  
  
public float getPreferenceFloat(String key){  
    SharedPreferences pref = getSharedPreferences(  
        PREFERENCE, MODE_PRIVATE);  
    return pref.getFloat(key, 0.0f);  
}  
  
public long getPreferenceLong(String key){  
    SharedPreferences pref = getSharedPreferences(  
        PREFERENCE, MODE_PRIVATE);  
    return pref.getLong(key, 0L);  
}
```



Preference 예제 0

■ MainActivity.JAVA

```
public void setPreferenceClear(){
    SharedPreferences pref = getSharedPreferences(
                                                PREFERENCE, MODE_PRIVATE);

    SharedPreferences.Editor editor = pref.edit();
    editor.clear();
    editor.apply();
    editor.commit();
    inputString.setText("");
    inputInt.setText("");
    inputFloat.setText("");
    inputLong.setText("");
}
}
```



Preference 예제 0

- 비교적 소규모의 키(Key) - 값(Value)의 자료구조의 데이터는 SharedPreferences API를 사용하여 데이터를 저장
- SharedPreferences 사용하여 저장되는 데이터는 XML 파일 형식으로 저장이 되며 앱이 삭제되기 전까지 보존이 됨
- 데이터베이스를 사용하는 것처럼 key-value 쌍의 데이터를 저장하고 불러오고 삭제가 가능
- SharedPreferences에는 수치 값이나 간단한 텍스트 또는 자동 로그인 여부와 같은 간단한 사용자 설정 값들을 저장하는 용도로 사용
- 저장 가능한 데이터 타입으로는 boolean, int, float, long, String이 있음



Preference 예제 0

■ 처리 방법

Shared Preferences 열기

```
SharedPreferences sp =  
getSharedPreferences("myFile", Activity.MODE_PRIVATE);
```

쓰기

```
SharedPreferences.Editor editor = sp.edit();  
editor.putString("keyword", keyword);  
editor.commit();
```

읽기

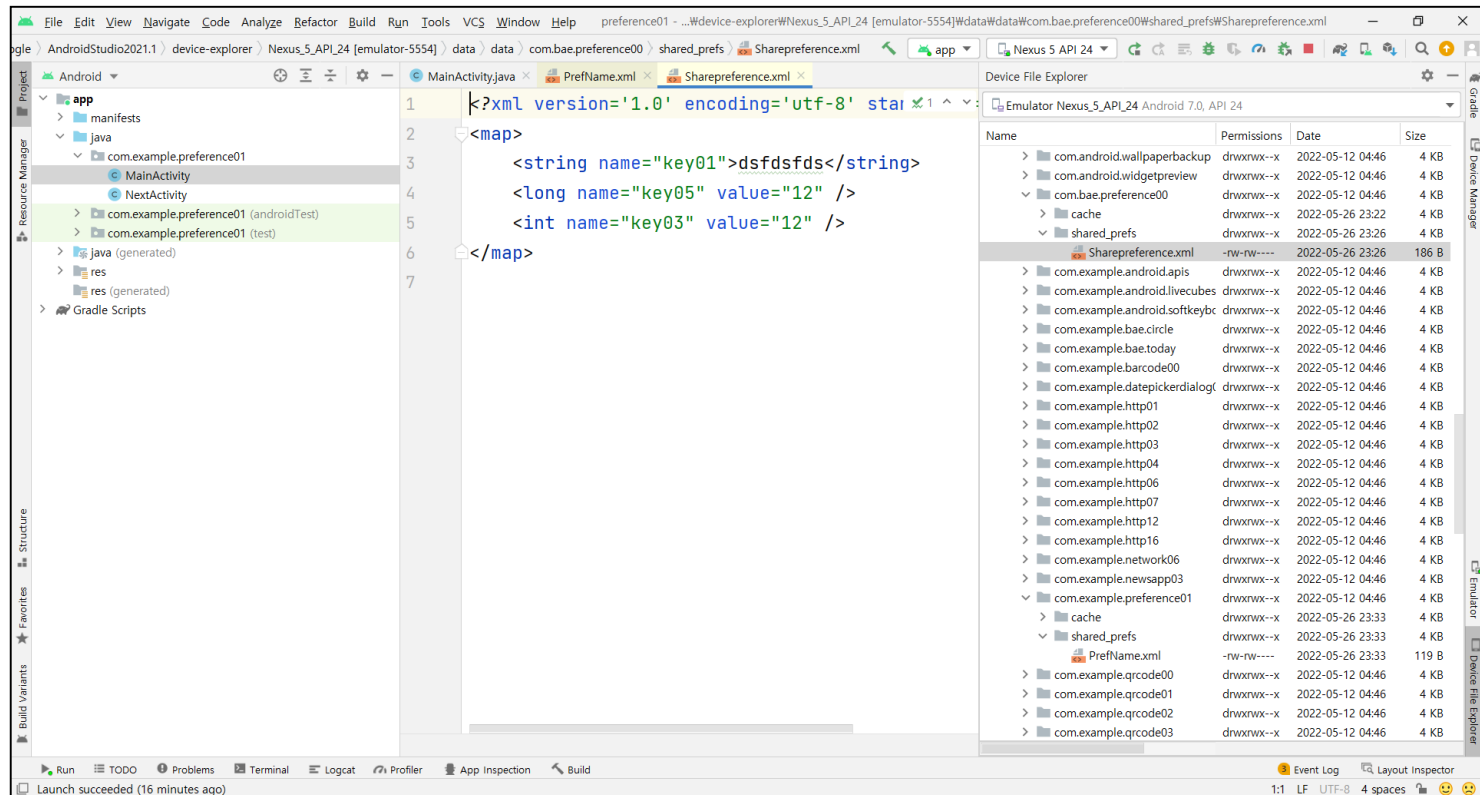
```
String keyword = sp.getString("keyword", "");
```



Preference 예제 0

■ 실습 방법

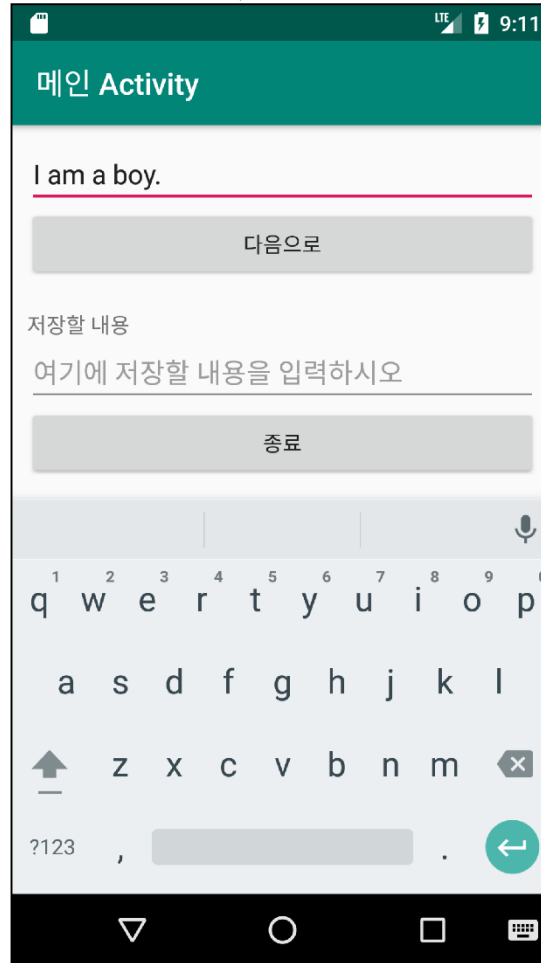
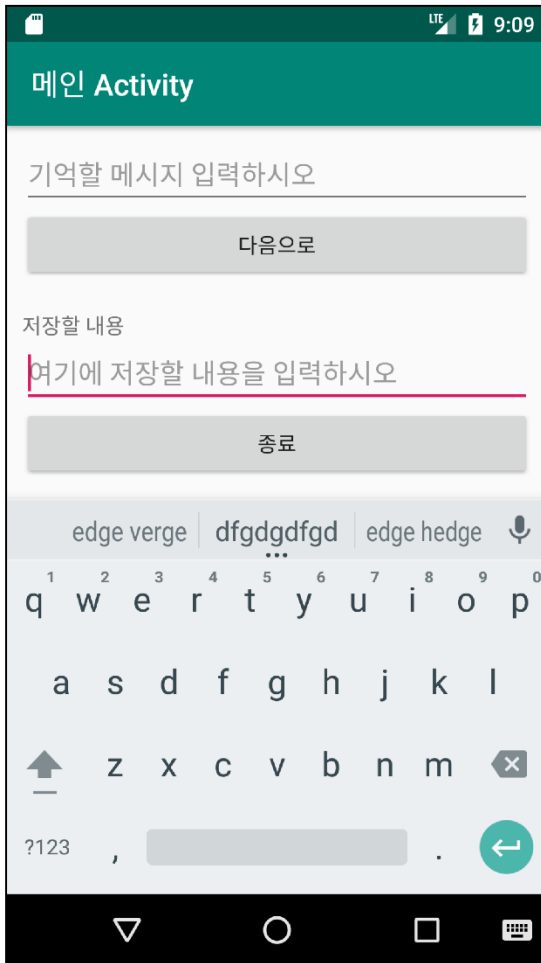
- App을 종료 후 다시 실행해보면 이전에 저장된 문자열이 표시됨
- 기억 공간 확인





Preference 예제 1

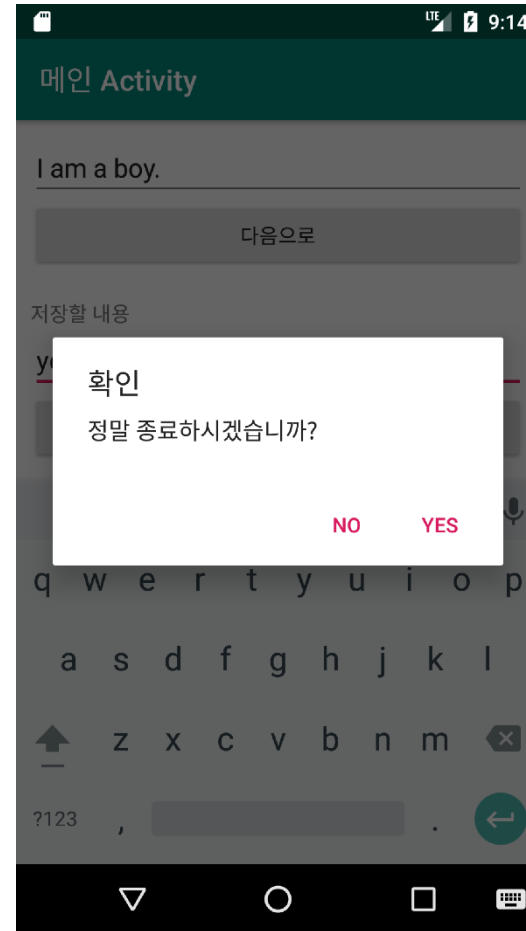
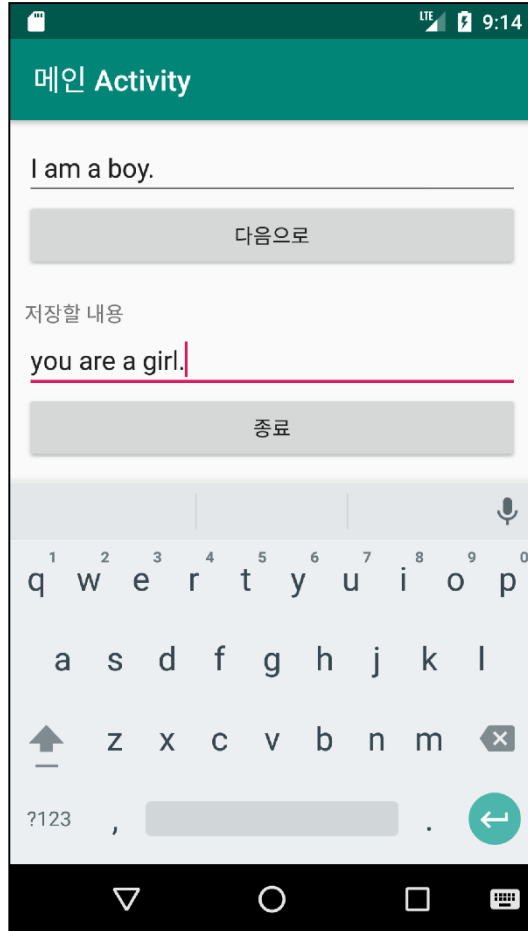
- 다음과 같이 기억할 메시지를 입력하고 저장하면 종료했다가 다시 실행하여도 저장되어 있는 앱을 만들어보자





Preference 예제 1

■ 실행 화면





Preference 예제 1

■ activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/editText1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="기억할 메시지 입력하십시오" />

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="다음으로" />
```




Preference 예제 1

■ activity_main.xml

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:text="저장할 내용" />

<EditText
    android:id="@+id/editText2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="여기에 저장할 내용을 입력하십시오"/>

<Button
    android:id="@+id/button2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="종료" />
</LinearLayout>
```



Preference 예제 1

■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {  
    EditText editText1, editText2;  
    SharedPreferences prefs;  
    String text = "";  
    String message = "";
```

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    setTitle("메인 Activity");
```

```
    editText1 = findViewById(R.id.editText1);  
    editText2 = findViewById(R.id.editText2);
```



Preference 예제 1

■ MainActivity.JAVA

```
Button button1 = findViewById(R.id.button1);
button1.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        text = editText1.getText().toString();
        if (text.equals("")) {
            Toast.makeText(getApplicationContext(), "입력해주세요",
                Toast.LENGTH_SHORT).show();
        } else {
            Intent intent = new Intent(MainActivity.this, NextActivity.class);
            intent.putExtra("msg", text);
            startActivity(intent);
        }
    }
});
```



Preference 예제 1

■ MainActivity.JAVA

```
Button button2 = findViewById(R.id.button2);
button2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        message = editText2.getText().toString();
        if (message.equals("")) {
            Toast.makeText(getApplicationContext(), "입력해주세요",
                Toast.LENGTH_SHORT).show();
        } else {
            AlertDialog.Builder builder =
                new AlertDialog.Builder(MainActivity.this);
            builder.setTitle("확인");
            builder.setMessage("정말 종료하시겠습니까?");
```



Preference 예제 1

■ MainActivity.JAVA

```
builder.setPositiveButton("Yes",
                           new DialogInterface.OnClickListener() {
                               @Override
                               public void onClick(DialogInterface dialogInterface, int i) {
                                   finish();
                               }
                           });
builder.setNegativeButton("No", null);
builder.create();
builder.show();
    }
});
prefs = getSharedPreferences("Preference", MODE_PRIVATE);
String temp = prefs.getString("next_page", "");
editText1.setText(temp);
editText2.setText(prefs.getString("store_message", ""));
}
```



Preference 예제 1

■ MainActivity.JAVA

```
protected void onStop() {  
    super.onStop();  
    SharedPreferences.Editor editor = prefs.edit();  
    editor.putString("next_page", text);  
    editor.putString("store_message", message);  
    editor.apply();  
    editor.commit();  
}  
}
```



Preference 예제 1

■ activity_next.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context=".NextActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="받은 내용 " />
```



Preference 예제 1

■ activity_next.xml

```
<TextView
    android:id="@+id/reseult"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:paddingLeft="10dp"
    android:text="다음 액티비티"
    android:textSize="20dp" />

<Button
    android:id="@+id/button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="이전으로" />

</LinearLayout>
```




Preference 예제 1

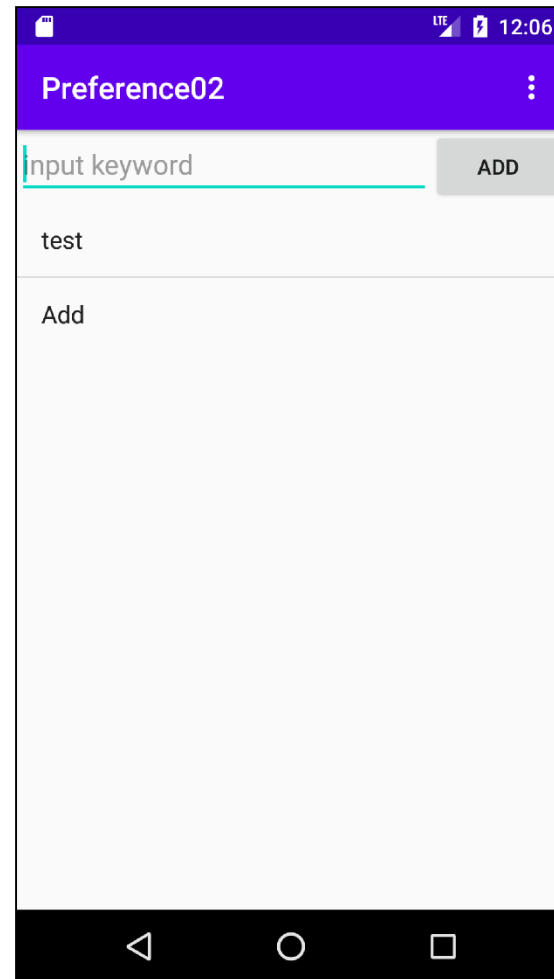
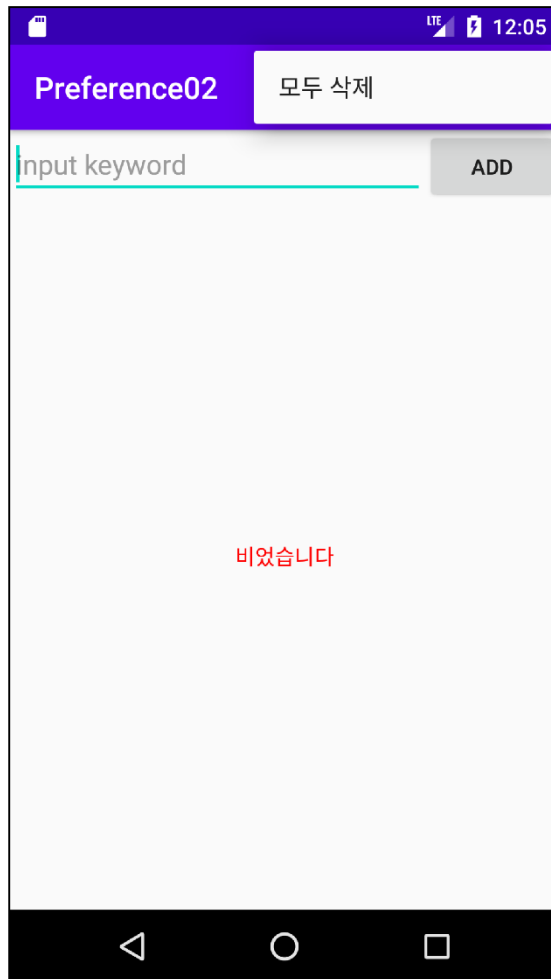
■ NextActivity.JAVA

```
public class NextActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_next);
        setTitle("서브 액티비티");
        TextView textView = findViewById(R.id.result);
        Intent intent = getIntent();
        textView.setText(intent.getStringExtra("msg"));
        Button button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                finish();
            }
        });
    }
}
```



Preference 예제 2





Preference 예제 2



■ activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <EditText
            android:id="@+id/editText"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:hint="input keyword"
            android:inputType="textPersonName" />
    </LinearLayout>
</LinearLayout>
```



Preference 예제 2



■ activity_main.xml

```
<Button
    android:id="@+id/button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="add" />
</LinearLayout>
<ListView
    android:id="@android:id/list"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
<TextView
    android:id="@android:id/empty"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:gravity="center"
    android:text="비었습니다"
    android:textColor="#FF0000" />
</LinearLayout>
```



Preference 예제 2

■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {  
    ArrayAdapter adapter;  
    ArrayList<String> array = new ArrayList<>();  
    static final String SETTINGS_PLAYER_JSON = "settings_item_json";  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        final EditText editText = findViewById(R.id.editText);  
  
        TextView textView = findViewById(android.R.id.empty);  
        ListView listView = findViewById(android.R.id.list);  
        listView.setEmptyView(textView);  
    }  
}
```



Preference 예제 2

■ MainActivity.JAVA

```
array = getStringArrayPref(this, SETTINGS_PLAYER_JSON);
adapter = new ArrayAdapter(this, android.R.layout.simple_list_item_1, array);
listView.setAdapter(adapter);
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view,
                           int position, long l) {

        String str = array.get(position);
        Toast.makeText(getBaseContext(), str,
                       Toast.LENGTH_SHORT).show();
    }
});
```



Preference 예제 2

■ MainActivity.JAVA

```
Button button = findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String str = editText.getText().toString();
        if (str.length() != 0) {
            array.add(str + "");
            adapter.notifyDataSetChanged();
            editText.setText("");
        } else {
            Toast.makeText(getApplicationContext(), "데이터 입력",
                Toast.LENGTH_SHORT).show();
        }
    }
});
}
```



Preference 예제 2

■ MainActivity.JAVA

```
private void setStringArrayPref(Context context, String key,
                                ArrayList<String> values) {
    SharedPreferences prefs = PreferenceManager.
                                getDefaultSharedPreferences(context);
    SharedPreferences.Editor editor = prefs.edit();
    JSONArray json = new JSONArray();

    for (int i = 0; i < values.size(); i++) {
        json.put(values.get(i));
    }
    if (!values.isEmpty()) {
        editor.putString(key, json.toString());
    } else {
        editor.putString(key, null);
    }
    editor.apply();
    editor.commit();
}
```




Preference 예제 2

■ MainActivity.JAVA

```
private ArrayList<String> getStringArrayPref(MainActivity context, String key) {  
    SharedPreferences prefs = PreferenceManager.  
        getDefaultSharedPreferences(context);  
    String test = prefs.getString(key, null);  
    ArrayList<String> urls = new ArrayList<>();  
    if (test != null) {  
        try {  
            JSONArray json = new JSONArray(test);  
            for (int i = 0; i < json.length(); i++) {  
                String url = json.optString(i);  
                urls.add(url);  
            }  
        } catch (JSONException e) {  
            e.printStackTrace();  
        }  
    }  
    return urls;  
}
```



Preference 예제 2

■ MainActivity.JAVA

@Override

```
protected void onPause() {  
    super.onPause();  
    setStringArrayPref(getBaseContext(), SETTINGS_PLAYER_JSON, array);  
}
```



Preference 예제 2

■ MainActivity.JAVA

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {  
    menu.add(0, 1, 0, "모두 삭제");  
    return true;  
}
```

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case 1:  
            array.clear();  
            adapter.notifyDataSetChanged();  
            return true;  
        }  
    return false;  
}
```



Preference 예제 3

Preference03

아이디

비밀번호

로그인

저장된 프리퍼런스의 정보
저장된 정보가 없음

Preference03

아이디

비밀번호

로그인

저장된 프리퍼런스의 정보
저장된 정보가 없음

입력해주세요

Preference03

아이디 bae

비밀번호

로그인

저장된 프리퍼런스의 정보
아이디 : bae
비밀번호 : 1234

q w e r t y u i o p
a s d f g h j k l
↑ 프리퍼런스에 정보가 저장되었습니다. ×
?123 , . ✓



Preference 예제 3

■ activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
```



Preference 예제 3



■ activity_main.xml

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/idStr"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <EditText
        android:id="@+id/id"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10">
        <requestFocus />
    </EditText>
</LinearLayout>
```



Preference 예제 3



■ activity_main.xml

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:text="@string/passStr"
        android:textAppearance="?android:attr/textAppearanceMedium" />

    <EditText
        android:id="@+id/pass"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:inputType="textPassword" />
</LinearLayout>
```



Preference 예제 3



■ activity_main.xml

```
<Button
    android:id="@+id/login"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="20dp"
    android:text="로그인" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="18dp"
    android:text="@string/resultStr"
    android:textAppearance="?android:attr/textAppearanceMedium" />

<TextView
    android:id="@+id/resultText"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium" />

</LinearLayout>
```




Preference 예제 3

■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {  
    static final String PREFS_NAME = "UserPrefs";  
    EditText idEdit, passEdit;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        requestWindowFeature(Window.FEATURE_NO_TITLE);  
        setContentView(R.layout.activity_main);  
  
        idEdit = findViewById(R.id.id);  
        passEdit = findViewById(R.id.pass);  
        final TextView result = findViewById(R.id.resultText);  
  
        final SharedPreferences prefs = getSharedPreferences(  
                                                    PREFS_NAME, MODE_PRIVATE);  
  
        String id = prefs.getString("id", "");  
        String pass = prefs.getString("pass", "");
```



Preference 예제 3

■ MainActivity.JAVA

```
if(id.equals(""))
    result.setText("저장된 정보가 없음");
else {
    result.setText("아이디 : " + id + "\n" + "비밀번호 : " + pass);
    idEdit.setText(id);
    passEdit.setText(pass);
}
```

```
Button login = findViewById(R.id.login);
login.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String inputId = idEdit.getText().toString();
        String inputPass = passEdit.getText().toString();
        if (inputId.equals("") || inputPass.equals("")) {
            Toast.makeText(getApplicationContext(), "입력해주세요",
                Toast.LENGTH_SHORT).show();
        } else {
```



Preference 예제 3

■ MainActivity.JAVA

```
SharedPreferences.Editor editor = prefs.edit();
editor.putString("id", inputId); //정보저장
editor.putString("pass", inputPass); //정보저장
editor.apply();
editor.commit();

Toast.makeText(getApplicationContext(),
    "프리퍼런스에 정보가 저장되었습니다.",
    Toast.LENGTH_SHORT).show();
result.setText("아이디 : " + inputId + "\n" + "비밀번호 : " +
    inputPass);
    }
    });
}
```



Preference 예제 4

■ Preference를 이용하여 id와 비밀번호를 기억하여보자

Preference04

ID를 입력하세요

ID를 입력

비밀번호를 입력하세요

비밀번호를 입력하세요

☐ 자동 로그인

로그인

로그인 기록 없음

Preference04

ID를 입력하세요

bae

비밀번호를 입력하세요

....

☐ 자동 로그인

로그인

q w e r t y u i o p
a s d f g h j k l
↑ z x c v b n m ×
?123 , . ✓

Preference04

ID를 입력하세요

ID를 입력

비밀번호를 입력하세요

비밀번호를 입력하세요

☐ 자동 로그인

로그인

q w e r t y u i o p
a s d f g h j k l
↑ z x Login ID bae n m ×
?123 , . ←



Preference 예제 4

■ 실행 화면

Preference04

ID를 입력하세요

bae

비밀번호를 입력하세요

....

☒ 자동 로그인

로그인

Preference04

ID를 입력하세요

ID를 입력

비밀번호를 입력하세요

비밀번호를 입력하세요

☐ 자동 로그인

로그인



Preference 예제 4



```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:text="ID를 입력하세요" />
    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:hint="ID를 입력"
        android:ems="10" >
    <requestFocus />
    </EditText>
```



Preference 예제 4

■ 사용자 인터페이스

<TextView

```
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_marginTop="30dp"  
android:text="비밀번호를 입력하세요" />
```

<EditText

```
android:id="@+id/editText2"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:inputType="textPassword"  
android:hint="비밀번호를 입력하세요"  
android:ems="10"/>
```



Preference 예제 4

■ 사용자 인터페이스

<CheckBox

```
    android:id="@+id/checkBox"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_marginTop="10dp"  
    android:text="자동 로그인" />
```

<Button

```
    android:id="@+id/button"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="로그인"/>
```

</LinearLayout>



Preference 예제 4

■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);
```

```
        final EditText editText1 = findViewById(R.id.editText1);
```

```
        final EditText editText2 = findViewById(R.id.editText2);
```

```
        final CheckBox checkBox = findViewById(R.id.checkBox);
```

```
        final SharedPreferences setting = getSharedPreferences("setting",  
                                                                MODE_PRIVATE);
```

```
        editText1.setText(setting.getString("ID", ""));
```

```
        editText2.setText(setting.getString("PW", ""));
```

```
        String id = editText1.getText().toString();
```

```
        String pw = editText2.getText().toString();
```



Preference 예제 4

```
if (id.equals("") && pw.equals("")) {
    checkBox.setChecked(false);
    Toast.makeText(getApplicationContext(), "로그인 기록 없음",
        Toast.LENGTH_SHORT).show();
} else
    checkBox.setChecked(true);
checkBox.setOnCheckedChangeListener(new CompoundButton.
    OnCheckedChangeListener() {

        @Override
        public void onCheckedChanged(CompoundButton buttonView,
            boolean isChecked) {

            if (isChecked) {
                editText1.setText(setting.getString("ID", ""));
                editText2.setText(setting.getString("PW", ""));
            } else {
                editText1.setText("");
                editText2.setText("");
                checkBox.setChecked(false);
            }
        }
    });
```



Preference 예제 4

■ MainActivity.JAVA

```
Button button = findViewById(R.id.button);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if (checkBox.isChecked()) {
            editText1.setText(setting.getString("ID", ""));
            editText2.setText(setting.getString("PW", ""));
        }
        String id = editText1.getText().toString();
        String pw = editText2.getText().toString();
        if (id.equals("") || pw.equals("")) {
            Toast.makeText(getApplicationContext(), "입력해주세요",
                Toast.LENGTH_SHORT).show();
        } else {
```



Preference 예제 4

■ MainActivity.JAVA

```
Toast.makeText(getApplicationContext(), "Login ID " + id,
                                           Toast.LENGTH_SHORT).show();

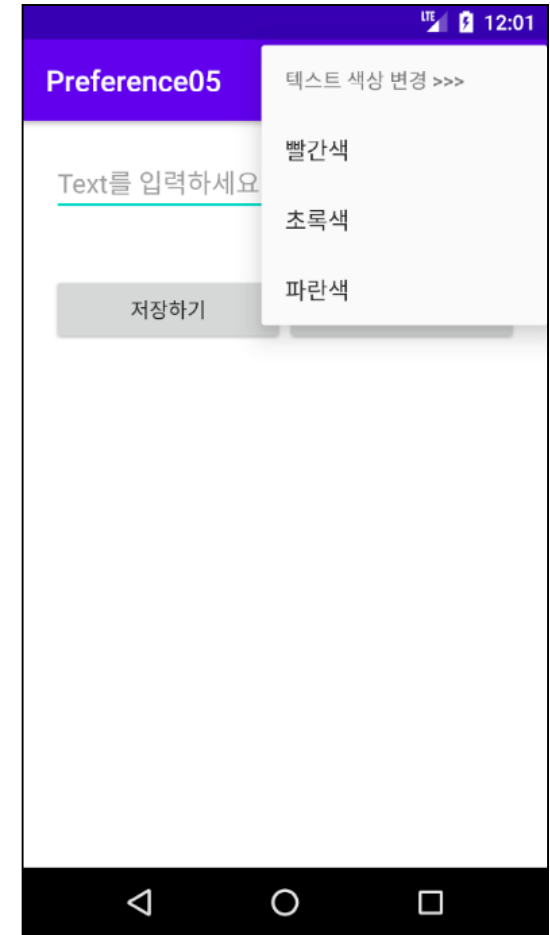
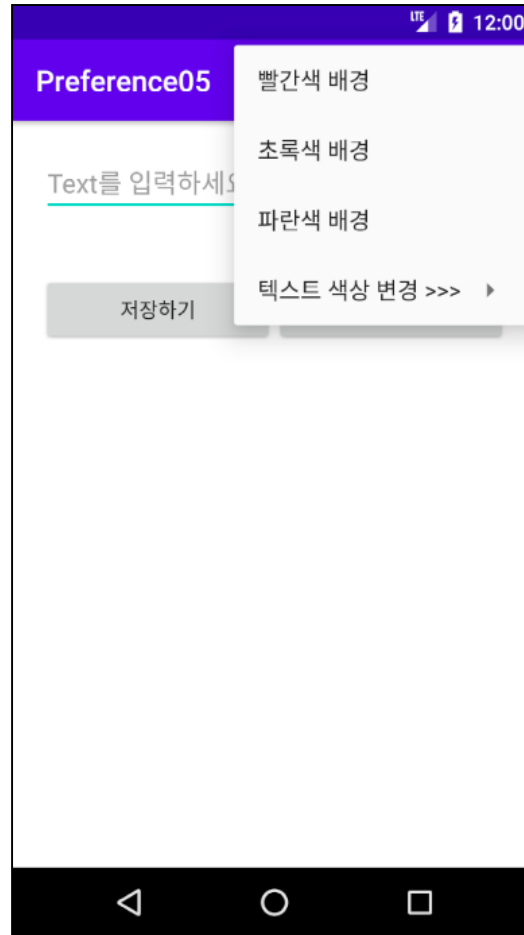
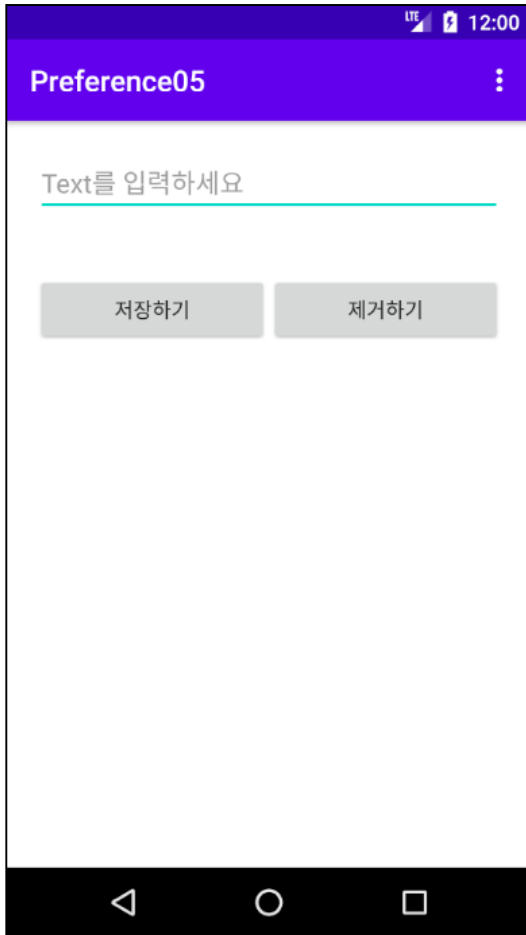
editText1.setText("");
editText2.setText("");
checkBox.setChecked(false);
editText1.requestFocus();

SharedPreferences.Editor editor = setting.edit();
editor.putString("ID", id);
editor.putString("PW", pw);
editor.apply();
editor.commit();
}
}
});
}
}
```



Preference 예제 5

- 옵션메뉴에서 지정한 색상을 저장하는 프로그램을 만들어보자





Preference 예제 5

■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/editText"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="20dp"
        android:hint="Text를 입력하세요"
        android:inputType="text" />
```



Preference 예제 5

■ 사용자 인터페이스

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="20dp">
    <Button
        android:id="@+id/btnSave"
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:text="저장하기"/>
    <Button
        android:id="@+id/btnRemove"
        android:layout_width="0dp"
        android:layout_weight="1"
        android:layout_height="wrap_content"
        android:text="제거하기"/>
</LinearLayout>
</LinearLayout>
```



Preference 예제 5

■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {  
    final String PREFERENCE = "Shareddata";  
    SharedPreferences preferences;  
    SharedPreferences.Editor editor;  
    int layoutColor;  
    int textColor;  
    String text;  
    LinearLayout layout;  
    EditText editText;  
    Button btn_save, btn_remove;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```




Preference 예제 5

■ MainActivity.JAVA

```
preferences = PreferenceManager.getDefaultSharedPreferences(this);  
// preferences = getSharedPreferences(PREFERENCE, MODE_PRIVATE);  
editor = preferences.edit();  
  
layout = findViewById(R.id.layout);  
editText = findViewById(R.id.editText);  
  
btn_save = findViewById(R.id.btnSave);  
btn_remove = findViewById(R.id.btnRemove);  
  
layoutColor = preferences.getInt("layoutcolor", Color.WHITE);  
layout.setBackgroundColor(layoutColor);  
textColor = preferences.getInt("textcolor", Color.BLACK);  
editText.setTextColor(textColor);  
text = preferences.getString("text", "");  
editText.setText(text);
```



Preference 예제 5

■ MainActivity.JAVA

```
btn_save.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        text = editText.getText().toString();  
        editor.putInt("layoutcolor", layoutColor);  
        editor.putInt("textcolor", textColor);  
        editor.putString("text", text);  
        editor.apply();  
    }  
});
```



Preference 예제 5

■ MainActivity.JAVA

```
btn_remove.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        editor.remove("layoutcolor");  
        editor.remove("textcolor");  
        editor.remove("text");  
        editor.apply();  
        editor.commit();  
        layoutColor = Color.WHITE;  
        textColor = Color.BLACK;  
        layout.setBackgroundColor(layoutColor);  
        editText.setTextColor(textColor);  
        editText.setText("");  
    }  
});  
}
```



Preference 예제 5

■ MainActivity.JAVA

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {  
    menu.add(0, 1, 0, "빨간색 배경");  
    menu.add(0, 2, 0, "초록색 배경");  
    menu.add(0, 3, 0, "파란색 배경");  
    SubMenu subMenu = menu.addSubMenu("텍스트 색상 변경 >>>");  
    subMenu.add(0, 4, 0, "빨간색");  
    subMenu.add(0, 5, 0, "초록색");  
    subMenu.add(0, 6, 0, "파란색");  
    return true;  
}
```



Preference 예제 5

■ MainActivity.JAVA

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case 1:  
            layoutColor = Color.RED;  
            layout.setBackgroundColor(layoutColor);  
            return true;  
        case 2:  
            layoutColor = Color.GREEN;  
            layout.setBackgroundColor(layoutColor);  
            return true;  
        case 3:  
            layoutColor = Color.BLUE;  
            layout.setBackgroundColor(layoutColor);  
            return true;  
    }  
}
```



Preference 예제 5

■ MainActivity.JAVA

```
case 4:
    textColor = Color.RED;
    editText.setTextColor(textColor);
    return true;
case 5:
    textColor = Color.GREEN;
    editText.setTextColor(textColor);
    return true;
case 6:
    textColor = Color.BLUE;
    editText.setTextColor(textColor);
    return true;
}
return false;
}
```



Preference 예제 5

■ MainActivity.JAVA

@Override

```
protected void onStop() {  
    super.onStop();  
    String text = editText.getText().toString();  
    editor.putInt("layoutcolor", layoutColor);  
    editor.putInt("textcolor", textColor);  
    editor.putString("text", text);  
    editor.apply();  
    editor.commit();  
}  
}
```