

MODULE 9

PREFERENCES

TABLE OF CONTENT

- Persistence of key-value pairs
- Types of Shared Preferences
- Activity Preferences
- Shared Preferences
- Type of Permissions
- Writing Values in SharedPreferences
- Reading Data from SharedPreferences

TABLE OF CONTENT

- User Settings
- Preference Fragment
- Webography

Persistence of key-value pairs

- ▶ To save and retrieve persistent key-value pairs
- ▶ In xml files
- ▶ To save primitive data
 - boolean
 - float
 - int
 - long
 - String
- ▶ Data will persist across user sessions
 - Even if the application is killed
 - Data accessed by different components of the same application

Types of Shared Preferences

- ▶ 2 types
 - Activity Preferences
 - Data accessed only by the activity
 - Shared Preferences
 - Data accessed by different components (activities or services) of the same application

Activity Preferences

- ▶ Preferences accessed only by one activity
- ▶ Create an object of SharedPreferences
 - By using the **getPreferences** method
 - Argument : type of permission (int)

Shared Preferences

- ▶ Preferences shared by different activities of the same application
- ▶ Create an object of SharedPreferences
 - By using the **getSharedPreferences** method
 - If multiple files of preferences
 - Arguments
 - Name of file of preferences (String)
 - Type of permission (int)
 - By using the **PreferenceManager.getDefaultSharedPreferences** method
 - If only one file of preferences
 - Argument
 - Context

Type of Permissions

- ▶ 3 modes of permission
 - `MODE_PRIVATE`
 - By default
 - Preferences accessible by the same application
 - `MODE_WORLD_READABLE`
 - Deprecated
 - Too dangerous (likely to cause security holes in application)
 - `MODE_WORLD_WRITEABLE`
 - Obsolete
 - Too dangerous (likely to cause security holes in application)

Writing Values in SharedPreferences

- ▶ Get a SharedPreferences.Editor
 - Call edit()
- ▶ Add values
 - Call methods such as putBoolean(), putString(), ...
 - According to the type of values
- ▶ Commit the new values
 - Call commit()

Writing Values in SharedPreferences

► E.g,

```
import android.content.SharedPreferences;
import android.content.SharedPreferences.Editor;
import android.preference.PreferenceManager;

public class MainActivity extends Activity {

    private SharedPreferences preferences;

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

        preferences = PreferenceManager.getDefaultSharedPreferences(this);
        Editor editor = preferences.edit();
        editor.putString("color", "red");
        editor.commit();
    }
}
```

Or getApplicationContext()

Retrieving shared preferences

Saving data in preferences

Commit changes in preferences

Reading Data from SharedPreferences

- ▶ No need of editor
- ▶ E.g,

```
public class SecondActivity extends Activity {  
  
    private SharedPreferences preferences;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_second);  
  
        preferences = PreferenceManager.getDefaultSharedPreferences(this);  
        String color = preferences.getString("color", "white");  
    }  
}
```

Reading data from
shared preferences

Default value if no corresponding data found in shared preferences

User Settings

- ▶ Settings that allow users to modify app features and behaviors
- ▶ Use Android's Preference APIs
 - To build an interface that's consistent with the user experience in other Android apps
 - Including the system settings
- ▶ Settings are built using subclasses of **Preference** class
 - E.g, CheckBoxPreference, ListPreference, EditTextPreference

User Settings

- ▶ Each Preference has a corresponding key-value pair saved by the system in a default SharedPreferences
 - When the user changes a setting, the system updates the corresponding value in the SharedPreferences file
- ▶ Settings declared in XML file
 - Each preference element should include attributes
 - android:key
 - Key in the SharedPreferences file
 - android:title
 - android:defaultValue
- ▶ Create preference xml file in /res/xml

User Settings

- ▶ E.g, xml file of settings

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://schemas.android.com/apk/res/android" >
  <EditTextPreference
    android:key="prefTextId"
    android:title="..."
    android:dialogTitle="..."
    android:dialogMessage="..."
    android:defaultValue="@string/welcomeDefaultValueID"/>
  <CheckBoxPreference
    android:key="prefCheckBoxId"
    android:title="..."
    android:defaultValue="true" />
</PreferenceScreen>
```

Preference Fragment

- ▶ For Android 3.0 (API level 11) and higher
 - Use a **PreferenceFragment** to display list of Preference objects
 - Create subclass of PreferenceFragment
 - Add a PreferenceFragment to any activity

Preference Fragment

- ▶ E.g, creating preference fragment

```
import android.os.Bundle;
import android.preference.PreferenceFragment;

public class SettingsFragment extends PreferenceFragment {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        this.addPreferencesFromResource(R.xml.mypreferences);
    }
}
```

→ Associate xml file to fragment

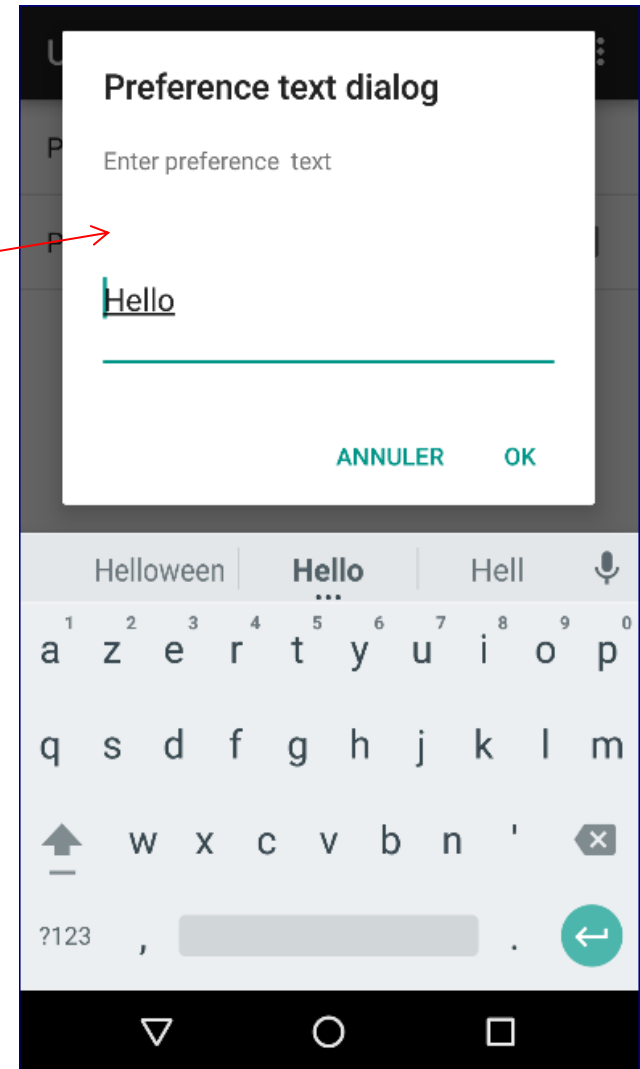
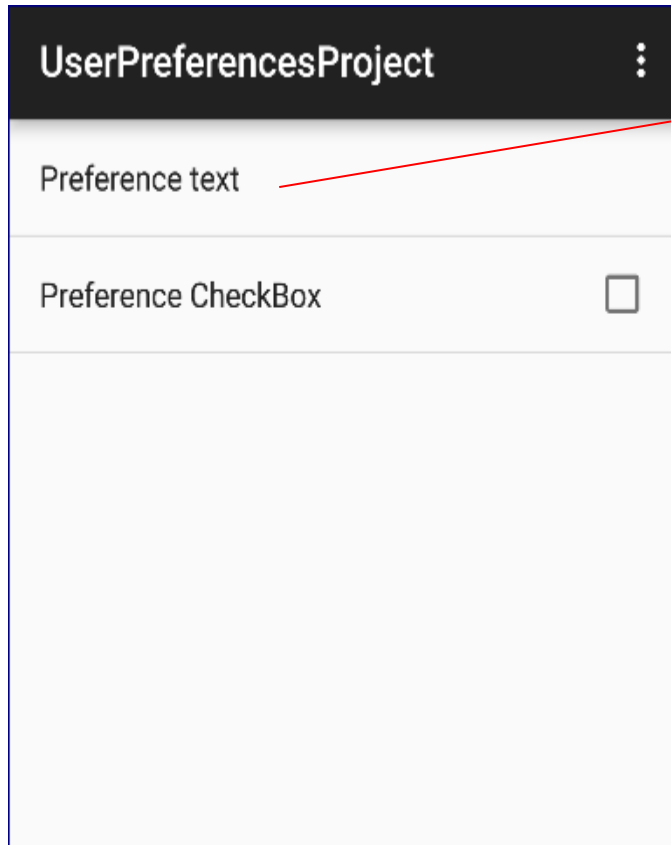
Preference Fragment

- ▶ E.g, adding preference fragment to activity

```
public class MainActivity extends Activity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        getSupportFragmentManager().beginTransaction()  
            .replace(android.R.id.content, new SettingsFragment())  
            .commit();  
    }  
}
```

→ Add fragment to activity

Preference Fragment



Webography

- ▶ <http://developer.android.com/training/basics/data-storage/shared-preferences.html>