



បណ្ណបណ្ណាលក្នុង សម្បត្តិ អេច អ ឌី

Korea Software HRD Center

Dialog

ណែនាំដោយ : Dr. Kim Tae Kyung



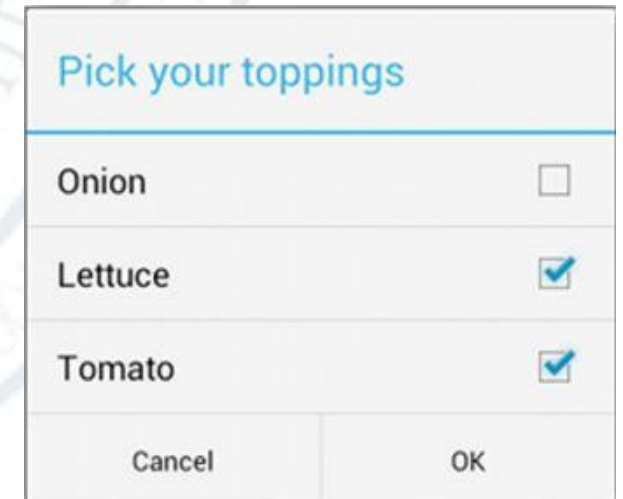
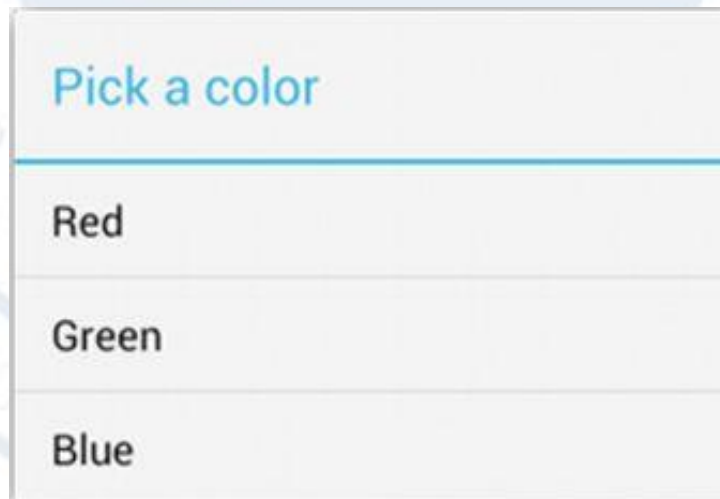
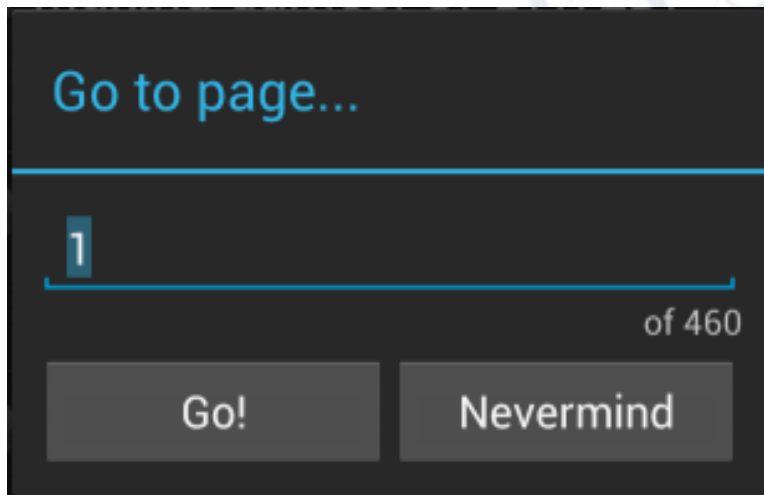
<http://www.kshrd.com.kh>

មាតិកា

1. Overview
2. Method សំខាន់ៗ របស់ Dialog Box
3. ការបង្កើត Dialog Box
4. List dialog
5. Dialog fragment
6. Custom Dialog

1. Overview

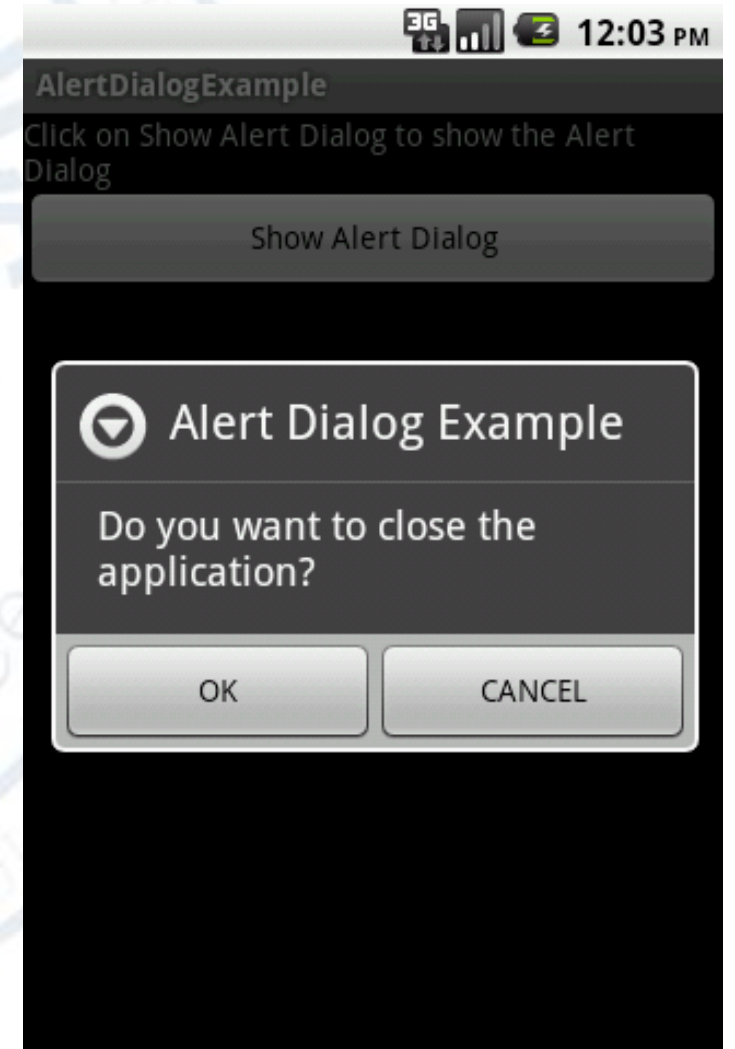
- Dialog ត្រូវបានប្រើសម្រាប់បង្ហាញផ្ទាំងនៃលក្ខខណ្ឌដល់ User ឬ សម្រាប់ទទួលយកនូវជម្រើសរបស់ User ឬ ការបញ្ចូលនូវទិន្នន័យពី User។ ផ្ទាំងនេះគឺបង្ហាញនៅក្នុង Activity ដដែល មិនបានបើកនូវ Activity ផ្សេងទៀតឡើយ។
- Dialog នេះ នឹងដំណើរការរហូតទាល់តែ User បញ្ចប់នូវ Action របស់វា ។



1. Overview (ត)

ឧទាហរណ៍៖

នៅពេលដែលយើងចង់សួរ User អោយធ្វើការសម្រេចចិត្ត
រវាង OK or Cancel ដើម្បីឆ្លើយតបទៅនឹង សកម្មភាពជាក់លាក់
ណាមួយរបស់ User ដោយមិនចាំបាច់ធ្វើការផ្លាស់ប្តូរ Screen
(Activity) នោះយើងអាចប្រើប្រាស់ Alert Dialog ។



2. Method សំខាន់ៗ របស់ Dialog Box

Method	Description
setIcon(Drawable icon)	ដាក់ Icon អោយ Alert Dialog Box
setCancelable(boolean cancelable)	កំណត់ថា តើ Dialog អាច cancel ឬ អត់
setMessage(CharSequence msg)	កំណត់ message ដើម្បីបង្ហាញក្នុង Content Area
setMultiChoiceItems(CharSequence[] items, boolean[] checkedItems, DialogInterface.OnMultiChoiceClickListener listener)	បញ្ចូលនូវ Check Box List ទៅក្នុង Content Area។ Selected Option នឹងត្រូវបានប្រាប់ដោយ listener
setTitle(CharSequence title)	កំណត់ Title ទៅអោយ Dialog Box

2. Method សំខាន់ៗ របស់ Dialog Box (ត)

Method	Description
<code>setOnCancelListener(DialogInterface. OnCancelListener onCancelListener)</code>	កំណត់ callback ដែលត្រូវហៅប្រសិនបើ dialog ត្រូវបាន canceled

3. ကလေးဘွဲ့ Dialog Box

// 1. Instantiate an [AlertDialog.Builder](#) with its constructor

```
AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
```

// 2. Chain together various setter methods to set the dialog characteristics

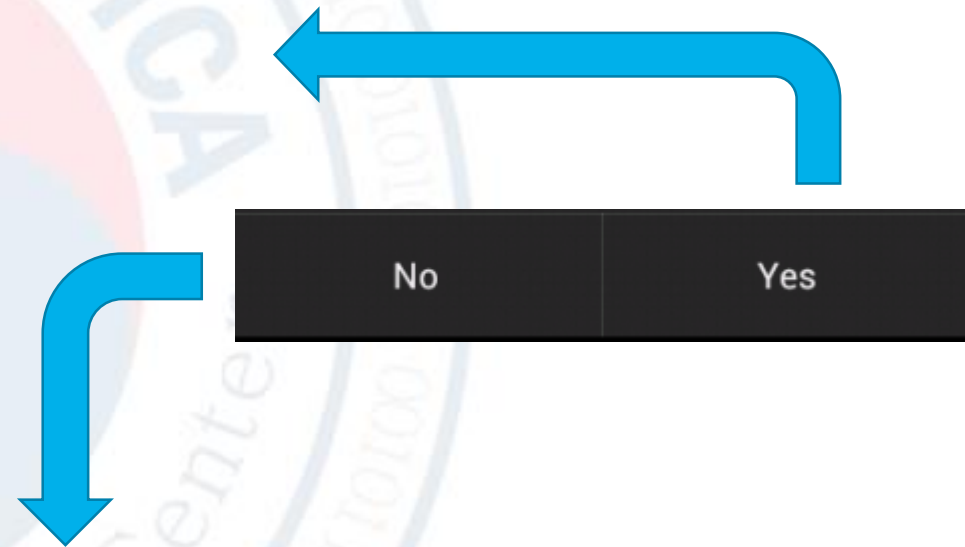
```
builder.setMessage(R.string.dialog_message)  
    .setTitle(R.string.dialog_title);
```

// 3. Get the [AlertDialog](#) from [create\(\)](#)

```
AlertDialog dialog = builder.create();  
dialog.show();
```

3. ကာယဗွေ့န Dialog Box (ခ)

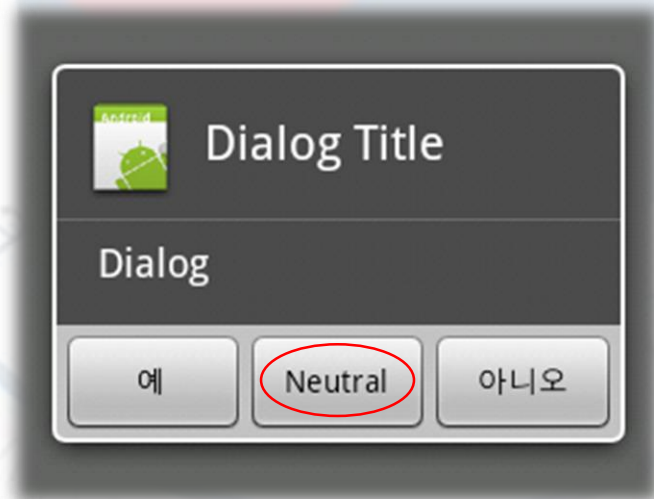
```
builder.setPositiveButton(R.string.ok, new DialogInterface.OnClickListener() {  
    public void onClick(DialogInterface dialog, int id) {  
        // User clicked OK button  
    }  
});
```



```
builder.setNegativeButton(R.string.cancel, new DialogInterface.OnClickListener() {  
    public void onClick(DialogInterface dialog, int id) {  
        // User cancelled the dialog  
    }  
});
```


3. **နာမပစ္စေိန္** Dialog Box (၈)

```
builder.setNeutralButton(R.string.cancel, new DialogInterface.OnClickListener() {  
    public void onClick(DialogInterface dialog, int id) {  
        // User cancelled the dialog  
    }  
});
```



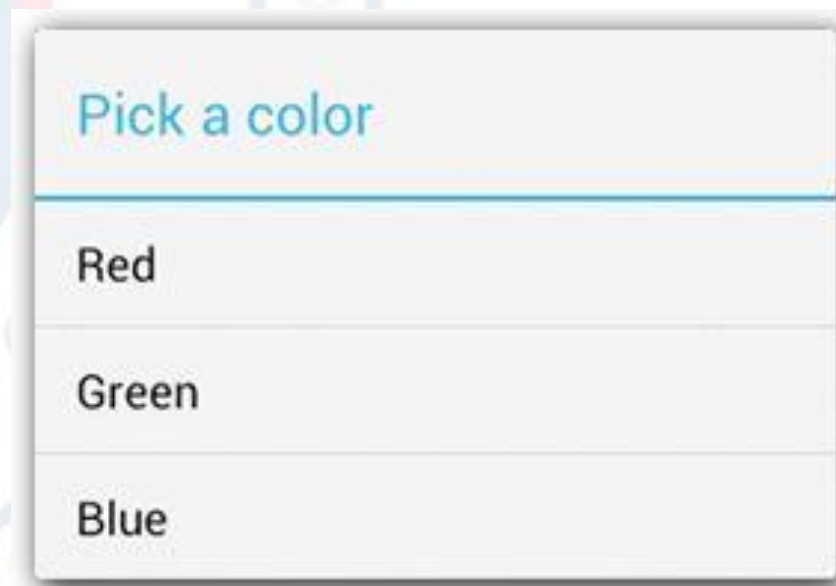
4. List Dialog

អ្វីទៅជា **List Dialog**?

- **List Dialog** ត្រូវបានគេប្រើសម្រាប់បង្ហាញ list of items នៅក្នុង dialog box មួយ។
- List Dialog ចែកចេញជា 3 ប្រភេទគឺ៖
 - Single-choice list
 - Radio button list
 - Checkbox list

Single-choice list dialog

- **Single-choice list dialog**: ជា text list ធម្មតា ឬជា list dialog មួយដែលអនុញ្ញាតឱ្យអ្នកប្រើប្រាស់ជ្រើសរើស item មួយ។
- List របស់វានឹងបាត់នៅពេលដែល item ណាមួយត្រូវបាន select។

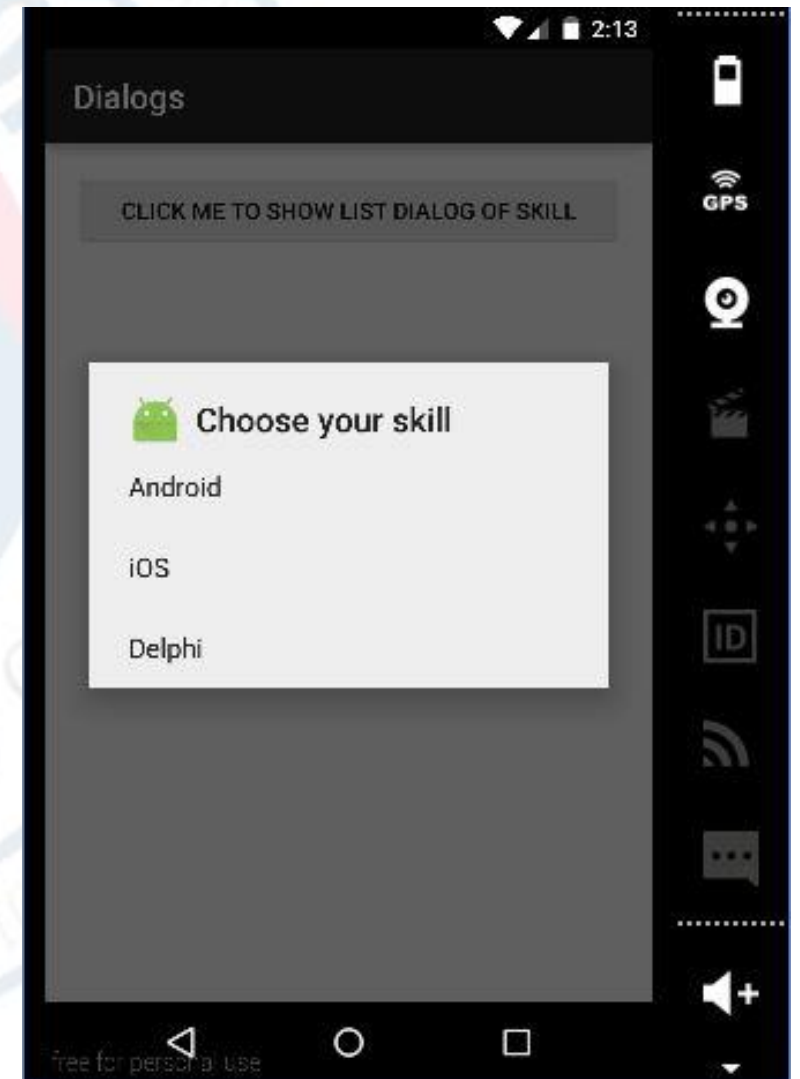


Single-choice list dialog (ឆ)

Example: បង្កើត a single-choice list dialog

1. ប្រកាស array មួយដែលមានឈ្មោះនៃ available classes

```
private String[ ] items = { "Android", "iOS", "Delphi" };
```



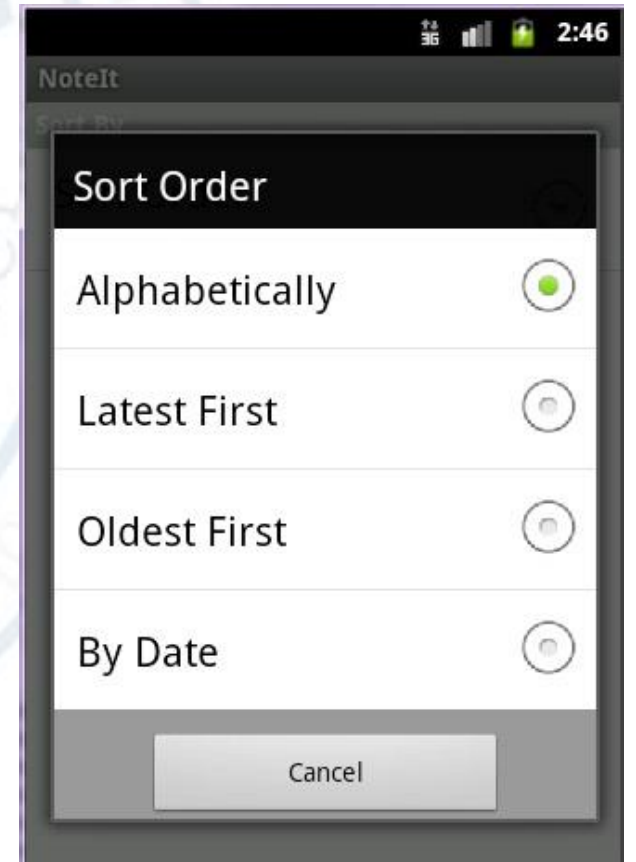
Single-choice list dialog (ត)

2. ប្រើ `setItems()` method នៃ builder ដើម្បី apply the array និង set a click listener ឲ្យវា៖

```
AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);
builder.setIcon(R.mipmap.ic_launcher);
builder.setTitle("Choose your skill");
builder.setItems(items, new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        Toast toast = Toast.makeText(getApplicationContext(), "Selected: "+items[which], Toast.LENGTH_SHORT);
        toast.show();
    }
});
builder.show();
```

Radio Button List Dialog

- **Radio button list:** ជា list dialog មួយ ដែលមានបង្ហាញនូវជម្រើសច្រើន ក៏ប៉ុន្តែ User អាចជ្រើសរើស Item បានតែមួយប៉ុណ្ណោះ ដោយវាបង្ហាញជាចំនួននៃ Radio Buttons។



Radio Button List Dialog (ត)

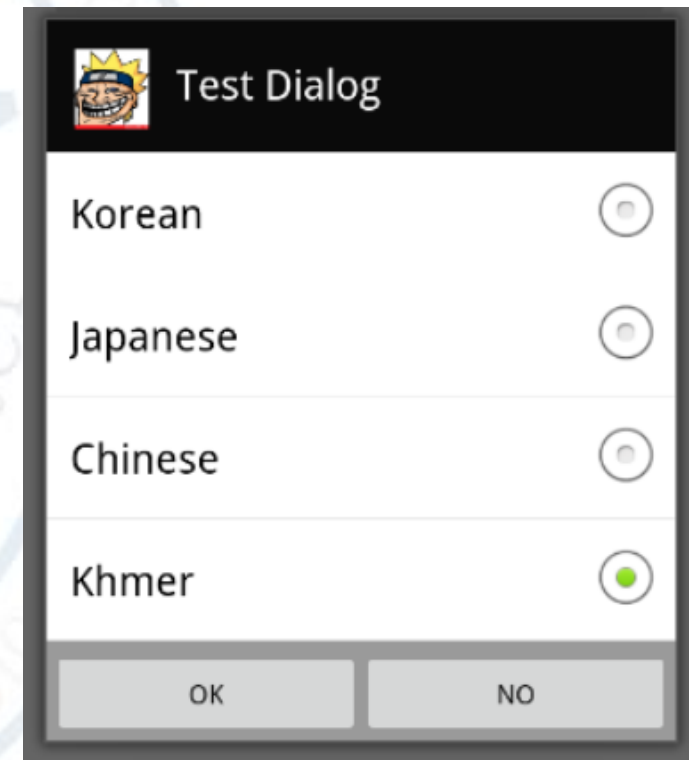
Example: បង្កើត Radio Button List Dialog

1. ប្រកាស array មួយដែលផ្ទុកនូវ ធាតុដូចខាងក្រោម៖

```
private String[ ] values = {  
    "Korean", "Japanese", "Chinese", "Khmer"  
};
```

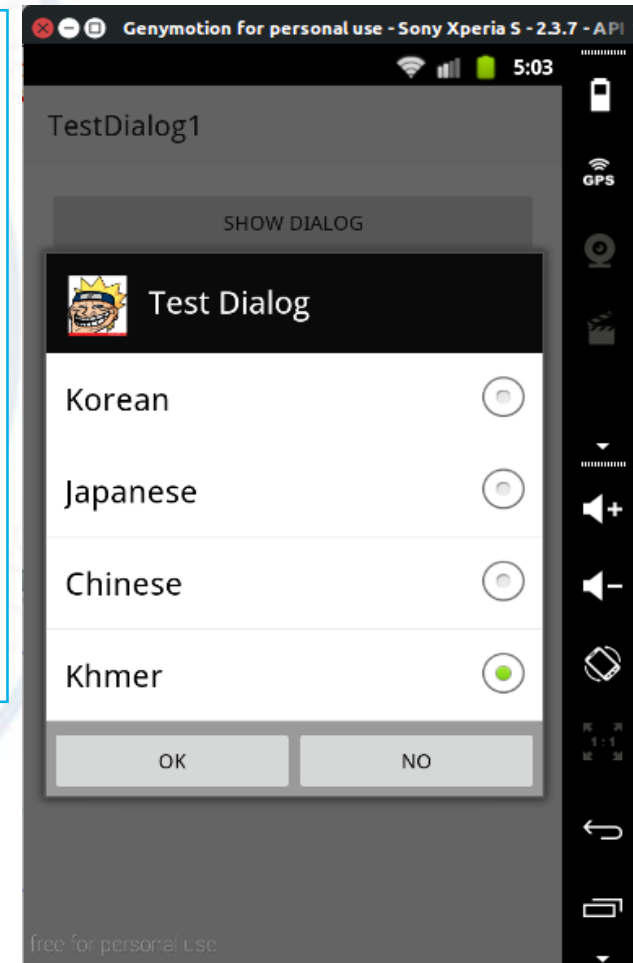
2. ប្រើនូវ Method របស់ AlertDialog.Builder:

```
setSingleChoiceItems(CharSequence[] items, int checkedItem,  
    onClickListener listener)
```



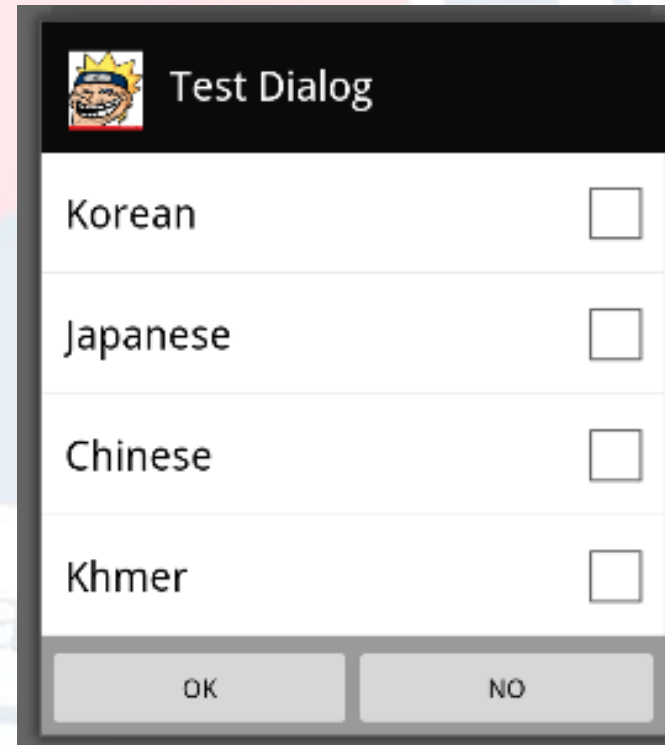
Radio Button List Dialog (👤)

```
String[] values = new String[] {  
    "Korean", "Japanese", "Chinese", "Khmer"  
};  
AlertDialog.Builder builder = new AlertDialog.Builder(this);  
builder  
    .setTitle("Test Dialog")  
    .setSingleChoiceItems(values, new DialogInterface.OnClickListener() {  
        @Override  
        public void onClick(DialogInterface dialogInterface, int i) {  
            // Do something here  
        }  
    });  
builder.show();
```



Checkbox List Dialog

- **Checkbox List Dialog:** ជា list dialog មួយដែលអាចអោយយើងធ្វើការជ្រើសរើស Items បានច្រើន ហើយ វាបង្ហាញជាចំនួននៃ Checkbox។



Checkbox List Dialog (ត)

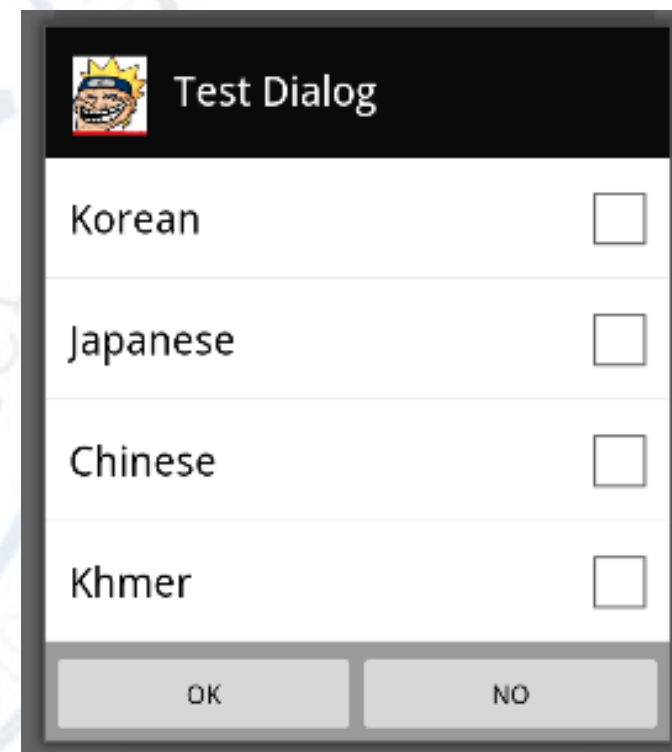
Example: បង្កើត Checkbox List Dialog

1. ប្រកាស array មួយដែលផ្ទុកនូវ ធាតុដូចខាងក្រោម៖

```
private String[ ] items = { "Korean", "Japanese", "Chinese",  
"Khmer"};
```

2. ប្រើនូវ Method របស់ AlertDialog.Builder:

```
setMultiChoiceItems(CharSequence[] items, boolean[] checkedItems,  
onMultiChoiceClickListener listener)
```



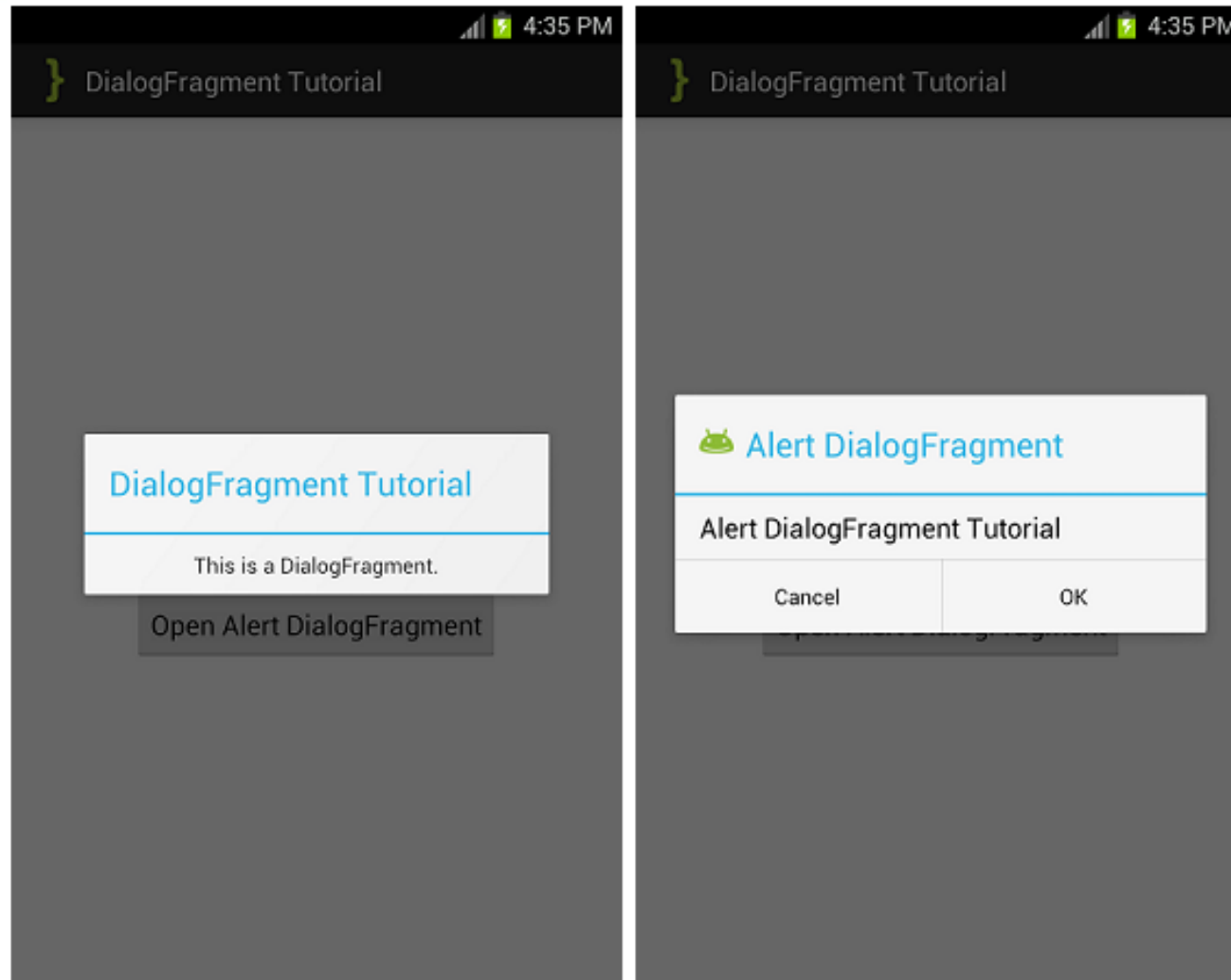
Checkbox List Dialog (☞)

```
final List<String> mSelectedItems = new ArrayList();
final String[] values = new String[] {
    "Korea", "Japanese", "Chinese", "Khmer"
};
AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setTitle("Test Dialog");
builder.setMultiChoiceItems(values, null, new DialogInterface.OnMultiChoiceClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which, boolean isChecked) {
        if (isChecked) {
            // If the user checked the item, add it to the selected items
            mSelectedItems.add(values[which]);
        } else if (mSelectedItems.contains(values[which])) {
            // Else, if the item is already in the array, remove it
            mSelectedItems.remove(values[which]);
        }
    }
})
builder.setPositiveButton("OK", null);
builder.show();
```

5. Dialog Fragment

- DialogFragment គឺជា Fragment មួយសម្រាប់បង្ហាញនូវ dialog window ឬ ការបង្ហាញរបស់វា នៅពីលើ activity's window។
- DialogFragment inherit ពី Fragment Class
- Support Library : `android.support.v4.app.DialogFragment`

5. Dialog Fragment (🐼)



Creating a Dialog Fragment

```
// AlertDialog.java
public class AlertDialogFragment extends DialogFragment {
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        // Use the Builder class for convenient dialog construction
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        builder.setIcon(R.mipmap.ic_launcher); //set icon
        builder.setTitle("Alert DialogFragment"); //setTitle
        builder.setMessage("Alert DialogFragment Tutorial")
            .setPositiveButton("OK", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    Log.d("msg", "It is OK!");
                }
            })
            .setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int id) {
                    Log.d("msg", "It is Cancel");
                }
            });
        // Create the AlertDialog object and return it
        return builder.create();
    }
}
```

Creating a Dialog Fragment (👤)

```
<!--activity_main.xml-->
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout 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" tools:context=".MainActivity">
    <Button
        android:id="@+id/alertdffragbutton"                android:layout_width="wrap_content"
        android:layout_height="wrap_content"                android:text="Alert Dialog Fragment" />
</RelativeLayout>
```

Creating a Dialog Fragment (👉)

```
/MainActivity.java
public class MainActivity extends AppCompatActivity{

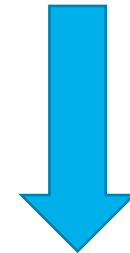
    Button alertdfragbutton;
    FragmentManager fm = getSupportFragmentManager();

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

        // Locate the button in activity_main.xml
        alertdfragbutton = (Button) findViewById(R.id.alertdfragbutton);

        // Capture button clicks
        alertdfragbutton.setOnClickListener(new View.OnClickListener() {
            public void onClick(View arg0) {
                AlertDialog alertdFragment = new AlertDialog();
                // Show Alert DialogFragment
                alertdFragment.show(fm, "Alert Dialog Fragment");
            }
        });
    }
}
```

OPEN ALERT DIALOGFRAGMENT



6. Custom Dialog

1. Create Layout
2. Create class extends from DialogFragment
3. Override method onCreateDialog
4. Set view to builder
5. Call by:

```
SignInFragmentDialog dialog = new SignInFragmentDialog();  
dialog.show(getSupportFragmentManager(),"Sing In");
```

6. Custom Dialog (🔗)

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content">
    <EditText
        android:id="@+id/username"
        android:inputType="textEmailAddress"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="UserName" />
    <EditText
        android:id="@+id/password"
        android:inputType="textPassword"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"

        android:hint="Passwor android:fontFamily="sans-serif"d"/>
</LinearLayout>
```


6. Custom Dialog (🔗)

```
public class SignInFragmentDialog extends DialogFragment {
    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        AlertDialog.Builder builder = new AlertDialog.Builder(getActivity());
        // Get the layout inflater
        LayoutInflater inflater = getActivity().getLayoutInflater();

        // Inflate and set the layout for the dialog
        // Pass null as the parent view because its going in the dialog layout
        builder.setView(inflater.inflate(R.layout.dialog_signin, null))
            // Add action buttons
            .setPositiveButton("Login", new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int id) {
                    // sign in the user ...
                }
            });
        return builder.create();
    }
}
```