



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

Lập trình Android

Bài 12: *Dialog Boxes & Toast*

Phòng LT & Mạng

<http://csc.edu.vn/lap-trinh-va-csdl>





Nội dung

1. Toast
2. AlertDialog
3. Tùy chỉnh Dialog
4. Một số dialog đặc biệt
 - TimePickerDialog
 - DatePickerDialog
5. FragmentDialog

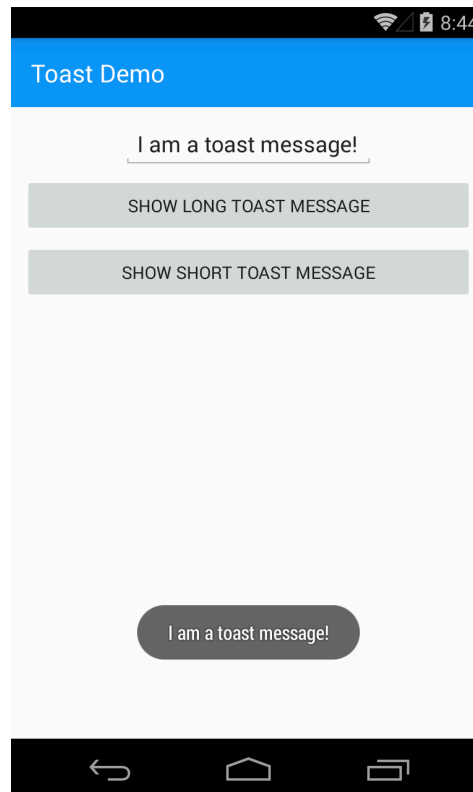


Android Dialog Boxes

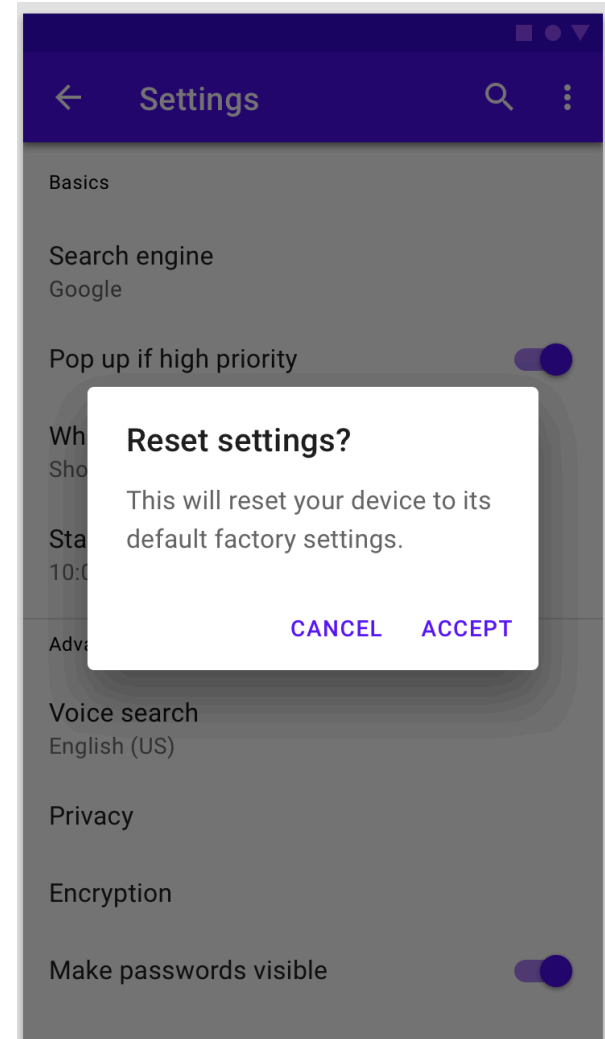
- Dialog là một cửa sổ nhỏ có chức năng nhắc người dùng đưa ra một quyết định hoặc nhập thông tin bổ sung
- Android cung cấp 2 loại dialog chính:

- Dialog

- Toast



Toast là 1 hộp thoại thông báo, hiển thị nhanh khoảng vài giây lên màn hình





Toast

- Cách đơn giản nhất để hiển thị một thông điệp
- Thường là thông điệp ngắn, thông báo nhanh đến người dùng và xuất hiện trong vài giây
- Cú pháp:

```
Toast.makeText(context, message, duration).show();
```

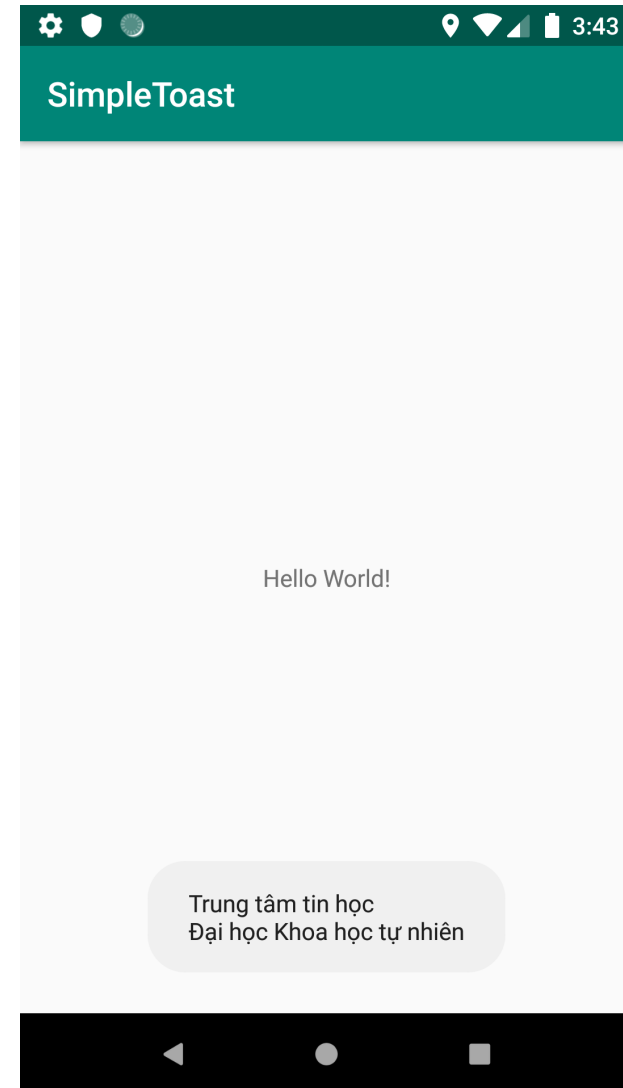
- Context: cho biết nơi Toast được hiển thị (màn hình nào...)
- Message: thông điệp được hiển thị
- Duration: **Toast.LENGTH_SHORT** – khoảng 2 giây
Toast.LENGTH_LONG – khoảng 3.5 giây



Toast – Ví dụ

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        Toast.makeText(getApplicationContext(),  
            "Trung tâm tin học\nĐại học Khoa học tự nhiên",  
            Toast.LENGTH_SHORT).show();  
    }  
}
```

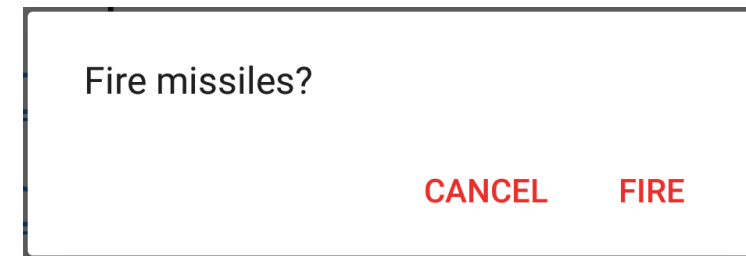
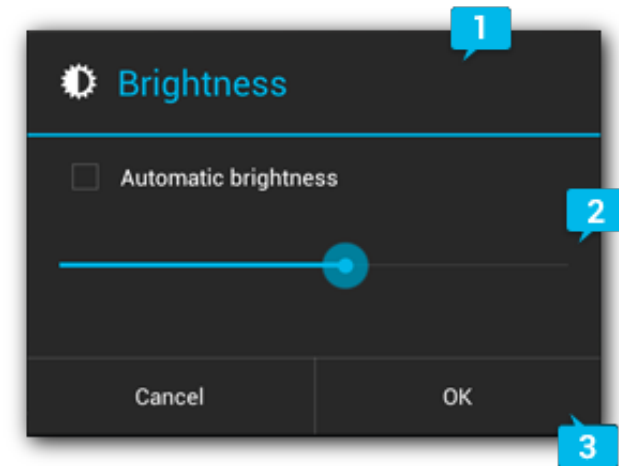
Trong trường hợp này, **context** có thể là `getApplicationContext()`, `MainActivity.this` hoặc đơn giản là `this`





AlertDialog

- Class **AlertDialog** cho phép xây dựng nhiều dialog khác nhau. Một AlertDialog gồm có 3 vùng:
 - (1) **Title**: tiêu đề
 - không bắt buộc phải có tiêu đề
 - (2) **Content**: vùng nội dung
 - có thể bố trí một thông báo (text), danh sách (list) hoặc tùy chỉnh (custom)
 - (3) **Button**: các nút nhấn
 - có không quá 3 nút nhấn trong 1 dialog



Dialog không có tiêu đề



Xây dựng AlertDialog

// 1. Khởi tạo 1 đối tượng AlertDialog.Builder

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

// 2a. Thiết lập các thuộc tính cho dialog

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

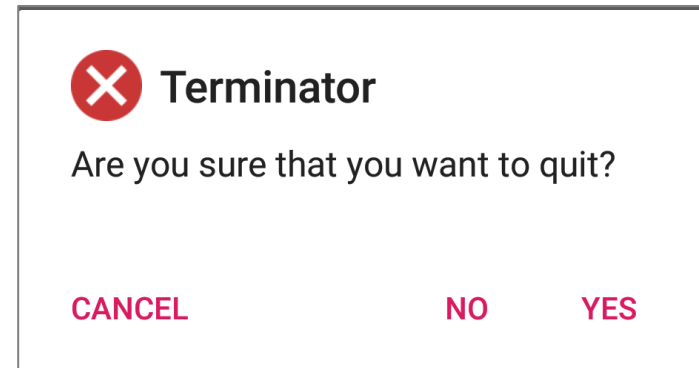
// tiêu đề dialog
// văn bản thông báo
// icon của dialog

// 3. Tạo ra đối tượng AlertDialog

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

// 4. Hiển thị AlertDialog

```
dialog.show();
```





Xây dựng AlertDialog

- Tại **bước 2**, có thể thêm các button cho dialog (có thể thêm 0-3 button):

// 2b. Thêm các button

```
builder.setPositiveButton(R.string.yes, new DialogInterface.OnClickListener() {  
    public void onClick(DialogInterface dialog, int id) {  
        // Xử lý trường hợp người dùng chọn button "Yes"  
    }  
});
```

dùng khi chấp nhận và tiếp tục với hành động (hành động "Yes")

```
builder.setNegativeButton(R.string.no, new DialogInterface.OnClickListener() {  
    public void onClick(DialogInterface dialog, int id) {  
        // Xử lý trường hợp người dùng chọn button "No"  
    }  
});
```

dùng khi huỷ bỏ hành động

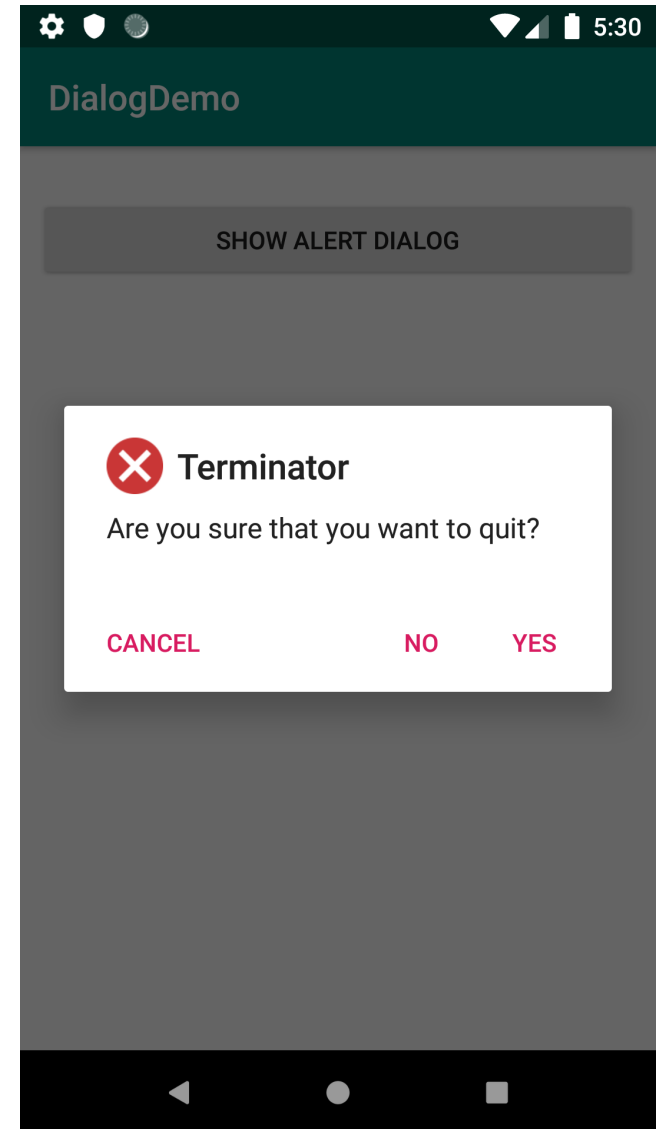
```
builder.setNeutralButton(R.string.cancel, new DialogInterface.OnClickListener() {  
    public void onClick(DialogInterface dialog, int id) {  
        // Xử lý trường hợp người dùng chọn button "Cancel"  
    }  
});
```

dùng khi không chấp nhận với hành động, cũng không muốn huỷ bỏ



Ví dụ AlertDialog

- Bắt đầu với **project-start** được cung cấp
- Giáo viên hướng dẫn học viên tiếp tục để đạt được kết quả như **project-final**





Tuỳ chỉnh Dialog

Có thể tạo ra một dialog với giao diện tùy chỉnh như sau:

// 1. Khởi tạo 1 dialog

```
final Dialog dialog = new Dialog(this);
```

// 2. Gán content view cho dialog là một layout tự định nghĩa

```
dialog setContentView(R.layout.dialog_custom);
```

// 3. Kết nối XML layout và Java code

```
final EditText txtUsername = dialog.findViewById(R.id.txtUsername);
```

```
final EditText txtPassword = dialog.findViewById(R.id.txtPassword);
```

```
Button btnClose = dialog.findViewById(R.id.btnClose);
```

// 4. Xử lý sự kiện trên các view con của dialog (nếu có)

```
btnClose.setOnClickListener(new View.OnClickListener() {
```

```
    @Override
```

```
    public void onClick(View v) {
```

```
        String username = txtUsername.getText().toString();
```

```
        String password = txtPassword.getText().toString();
```

```
        dialog.dismiss(); // Tắt dialog
```

```
    }
```

```
});
```

// 5. Thiết lập layout_width và layout_height cho dialog

```
dialog.getWindow().setLayout(WindowManager.LayoutParams.MATCH_PARENT,
```

```
WindowManager.LayoutParams.WRAP_CONTENT);
```

// 6. Hiện thị dialog

```
dialog.show();
```



Tuỳ chỉnh dialog

```
<?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="wrap_content"
    android:orientation="vertical">
```

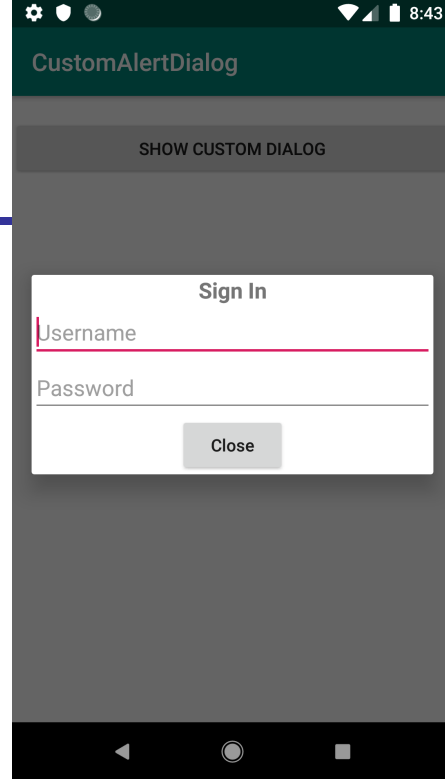
```
<TextView
    android:id="@+id/tvTitle"
    android:text="@string/sign_in"
    android:gravity="center"
    android:textStyle="bold"
    android:textSize="@dimen/largeFontSize"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

```
<EditText
    android:id="@+id/txtUsername"
    android:hint="@string/username"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
```

```
<EditText
    android:id="@+id/txtPassword"
    android:hint="@string/password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPassword" />
```

```
<Button
    android:id="@+id/btnClose"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:text="@string/close"
    android:textAllCaps="false" />
```

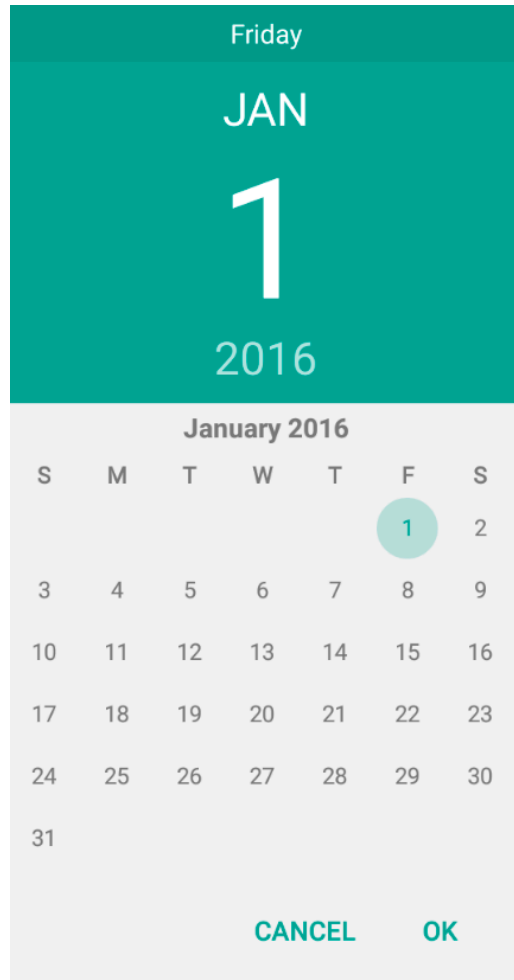
```
</LinearLayout>
```



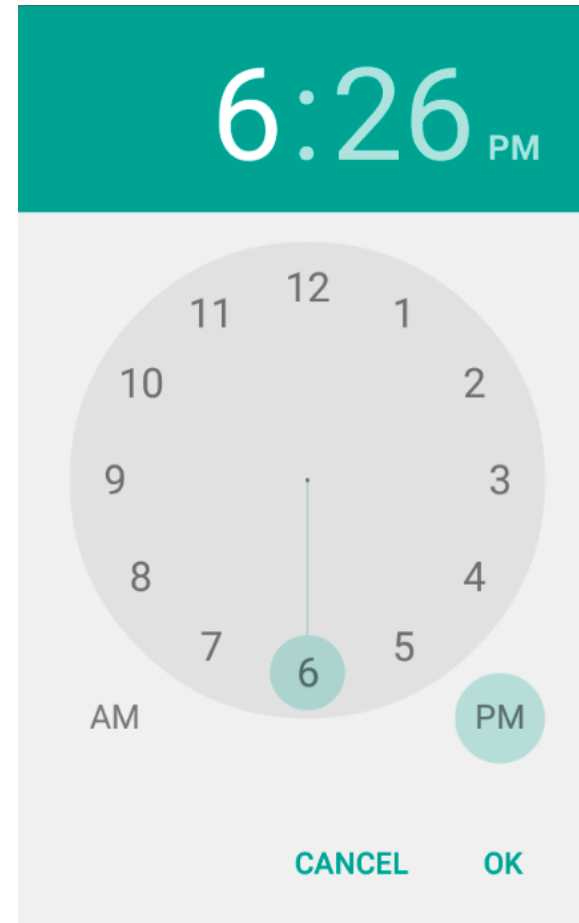
res/layout/dialog_custom.xml



Một số dialog đặc biệt



DatePickerDialog



TimePickerDialog



DatePickerDialog

- **DatePickerDialog** – cho phép người dùng chọn ngày
- Các bước xây dựng 1 **DatePickerDialog** như sau:

// 1. Lấy thông tin ngày hiện tại

```
final Calendar c = Calendar.getInstance();  
int year = c.get(Calendar.YEAR);  
int month = c.get(Calendar.MONTH);  
int day = c.get(Calendar.DAY_OF_MONTH);
```

// 2. Tạo ra một date picker dialog với các tham số: context, date set listener, ngày hiện tại

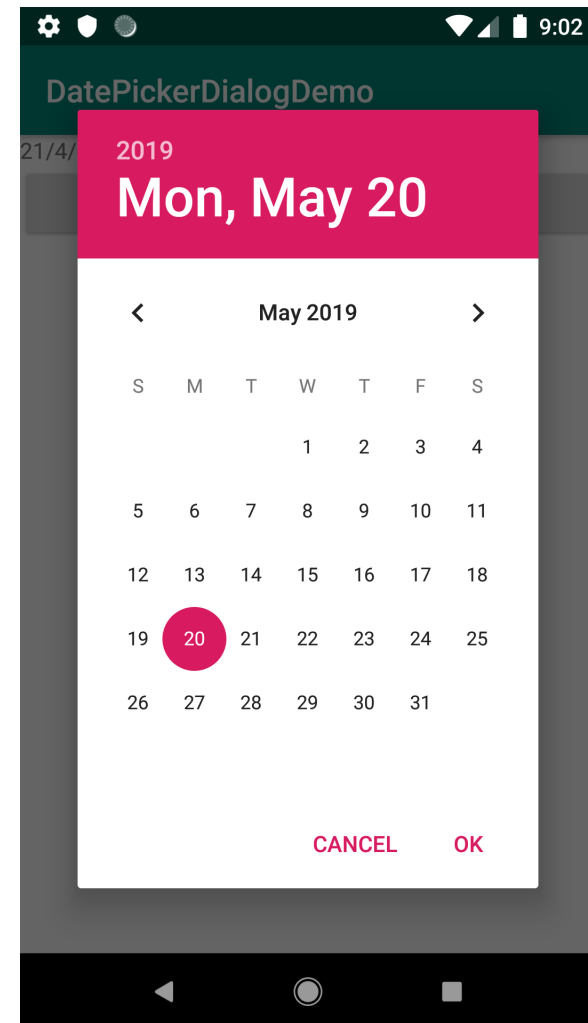
```
DatePickerDialog dialog = new DatePickerDialog(context,  
dateSetListener, year, month, day);
```

// 3. Hiển thị dialog

```
dialog.show();
```

context - nơi dialog được hiển thị (màn hình hiện tại)

dateSetListener – đối tượng xử lý sự kiện khi chọn 1 ngày trên dialog





DatePickerDialog

Ví dụ: dùng activity để lắng nghe sự kiện chọn 1 ngày trên **DatePickerDialog**

```
public class MainActivity extends AppCompatActivity implements DatePickerDialog.OnDateSetListener {
```

```
    private void showDatePickerDialog() {  
        final Calendar c = Calendar.getInstance();  
        int year = c.get(Calendar.YEAR);  
        int month = c.get(Calendar.MONTH);  
        int day = c.get(Calendar.DAY_OF_MONTH);  
        DatePickerDialog dialog = new DatePickerDialog(this, this, year, month, day);  
        dialog.show();  
    }
```

@Override

```
    public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {  
        String date = dayOfMonth + "/" + month + "/" + year;  
    }  
}
```

OnDateSetListener

Hàm callback **onDateSet** cho biết thông tin ngày, tháng, năm được chọn



TimePickerDialog

- **TimePickerDialog** – cho phép người dùng chọn giờ
- Các bước xây dựng 1 **TimePickerDialog** như sau:

// 1. Lấy thông tin giờ hiện tại

```
Calendar calendar = Calendar.getInstance();  
int hour = calendar.get(Calendar.HOUR_OF_DAY);  
int minute = calendar.get(Calendar.MINUTE);
```

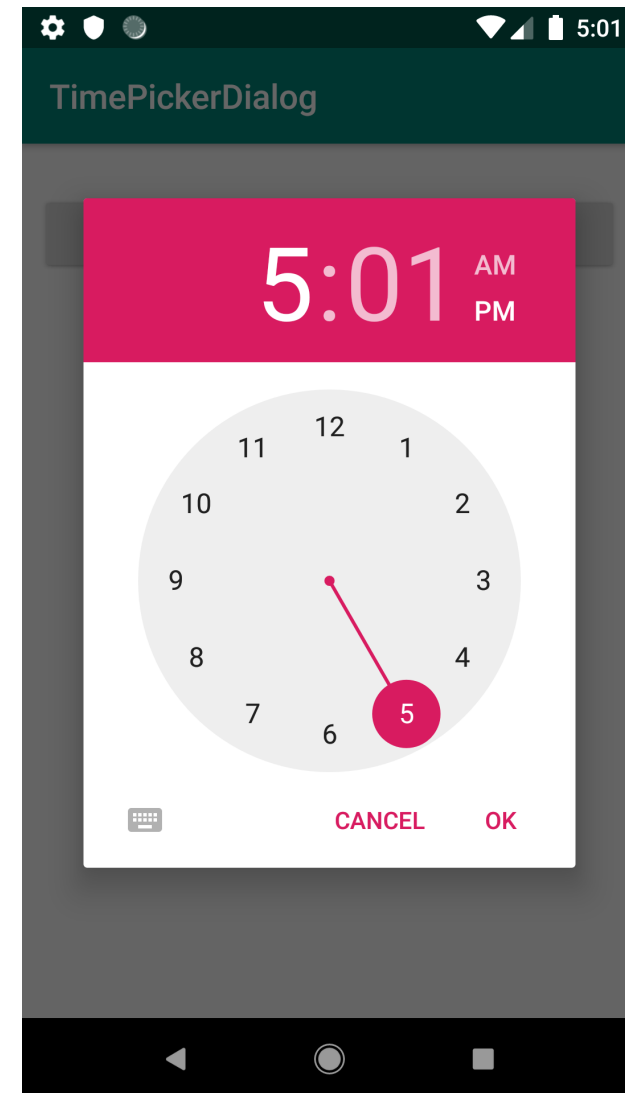
// 2. Tạo dialog với thông tin: context, time set listener, giờ hiện tại, định dạng giờ

```
TimePickerDialog dialog = new TimePickerDialog(context,  
timeSetListener, hour, minute, DateFormat.is24HourFormat(this));
```

// 3. Hiển thị dialog
dialog.show();

context - nơi dialog được hiển thị (màn hình hiện tại)

timeSetListener – đối tượng xử lý sự kiện khi chọn 1 giờ trên dialog





TimePickerDialog

Ví dụ: dùng activity để lắng nghe sự kiện chọn 1 ngày trên TimePickerDialog

```
public class MainActivity extends AppCompatActivity implements TimePickerDialog.OnTimeSetListener {
```

```
    private void showDatePickerDialog() {  
        final Calendar c = Calendar.getInstance();  
        int year = c.get(Calendar.YEAR);  
        int month = c.get(Calendar.MONTH);  
        int day = c.get(Calendar.DAY_OF_MONTH);  
        DatePickerDialog dialog = new DatePickerDialog(this, this, year, month, day);  
        dialog.show();  
    }
```

@Override

```
    public void onTimeSet(TimePicker view, int hourOfDay, int minute) {  
        String time = hourOfDay + ":" + minute;  
    }  
}
```

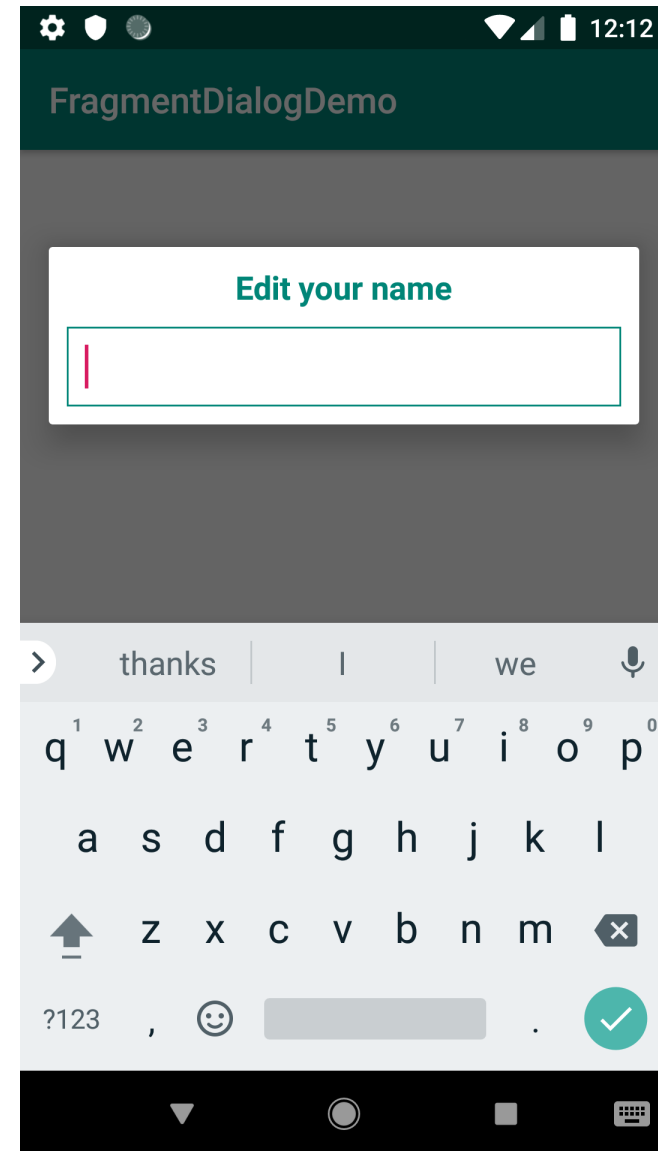
OnTimeSetListener

Hàm callback **onTimeSet** cho biết thông tin giờ, phút được chọn



AlertDialog

- Để xây dựng một dialog với layout tùy chỉnh, có thể sử dụng **AlertDialog**.
- Các bước thực hiện:
 - **Bước 1:** Tạo XML layout file cho dialog
 - **Bước 2:** Tạo class Java gắn với giao diện này, kế thừa từ **AlertDialog**
 - **Bước 3:** Hiển thị dialog trên activity hoặc fragment





FragmentManager

- **Bước 1:** Tạo XML layout file `res/layout/fragment_edit_name.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="wrap_content"
```

```
    android:gravity="center_horizontal"
```

```
    android:padding="10dp"
```

```
    android:orientation="vertical">
```

```
<TextView
```

```
    android:id="@+id/tvTitle"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="Edit your name"
```

```
    android:textColor="@color/colorPrimary"
```

```
    android:textSize="18sp"
```

```
    android:textStyle="bold"/>
```

```
<EditText
```

```
    android:id="@+id/txtName"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content"
```

```
    android:imeOptions="actionDone"
```

```
    android:background="@drawable/bg_border"
```

```
    android:padding="10dp"
```

```
    android:layout_marginTop="10dp"
```

```
    android:inputType="text" />
```

```
</LinearLayout>
```





AlertDialog

- **Bước 2:** Tạo class Java gắn với giao diện này, kế thừa từ **AlertDialog**

```
import android.support.v4.app.AlertDialog;
```

```
public class EditNameDialogFragment extends AlertDialog {  
    private TextView tvTitle;  
    private EditText txtName;
```

Lưu ý: sử dụng gói support v4 library

```
    public EditNameDialogFragment() {  
    }
```

```
    public static EditNameDialogFragment newInstance(String title) {  
        EditNameDialogFragment frag = new EditNameDialogFragment();  
        Bundle args = new Bundle();  
        args.putString("title", title);  
        frag.setArguments(args);  
        return frag;  
    }
```

Tạo dialog và truyền tham số cho dialog (nếu có)

@Override

```
    public View onCreateView(LayoutInflater inflater, ViewGroup container,  
        Bundle savedInstanceState) {  
        return inflater.inflate(R.layout.fragment_edit_name, container);  
    }
```

Gắn XML layout tương ứng cho dialog



AlertDialog

- **Bước 2:** (tiếp theo)

```
public class EditNameDialogFragment extends DialogFragment {  
    private TextView tvTitle;  
    private EditText txtName;  
    // ...  
    @Override  
    public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {  
        super.onViewCreated(view, savedInstanceState);  
        // Kết nối với XML layout  
        tvTitle = view.findViewById(R.id.tvTitle);  
        txtName = view.findViewById(R.id.txtName);  
        // Lấy giá trị các tham số truyền từ hàm newInstance. Trường hợp này lấy "title", nếu không  
        // có thì mặc định là "Enter your name"  
        String title = getArguments().getString("title", "Enter your name");  
        tvTitle.setText(title);  
    }  
  
    @Override  
    public void onResume() {  
        getDialog().getWindow().setLayout(WindowManager.LayoutParams.MATCH_PARENT,  
        WindowManager.LayoutParams.WRAP_CONTENT);  
        super.onResume();  
    }  
}
```

Thiết lập layout_width, layout_height cho dialog





AlertDialog

- **Bước 3:** hiển thị dialog trên activity (hoặc fragment)

```
import android.support.v4.app.FragmentManager;

public class MainActivity extends AppCompatActivity {

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

    private void showEditDialog() {
        FragmentManager fm = getSupportFragmentManager();
        EditNameDialogFragment editNameDialogFragment =
        EditNameDialogFragment.newInstance("Some Title");
        editNameDialogFragment.show(fm, "fragment_edit_name");
    }
}
```

Sử dụng support fragment manager để hiển thị dialog



AlertDialog

- **Bước 4:** Truyền dữ liệu từ dialog qua activity (hoặc fragment) có thể thực hiện như sau:
 - Định nghĩa 1 interface (listener) có phương thức sẽ được gọi để truyền dữ liệu qua activity
 - Thiết lập sự kiện trên 1 view của dialog, yêu cầu listener gọi phương thức để truyền dữ liệu
 - Implement interface trên trong activity/fragment và định nghĩa hành vi của phương thức tương ứng



AlertDialog

```
public class EditNameDialogFragment extends DialogFragment implements TextView.OnEditorActionListener {  
    public interface EditNameDialogListener {  
        void onFinishEditDialog(String inputText);  
    }  
}
```

Định nghĩa interface có phương thức trả lại dữ liệu

@Override

```
public void onViewCreated(View view, @Nullable Bundle savedInstanceState) {  
    super.onViewCreated(view, savedInstanceState);  
    // ...  
    txtName.setOnEditorActionListener(this); // Callback khi nhấn nút "Done" trên bàn phím  
    // ...  
}
```

@Override

```
public boolean onEditorAction(TextView v, int actionId, KeyEvent event) {  
    if (EditorInfo.IME_ACTION_DONE == actionId) {  
        EditNameDialogListener listener = (EditNameDialogListener) getActivity();  
        listener.onFinishEditDialog(txtName.getText().toString()); // trả dữ liệu đã nhập cho activity  
        dismiss(); // đóng dialog và quay về activity cha  
        return true;  
    }  
    return false;  
}
```

Gọi thực hiện hành vi tương ứng của listener





FragmentManager

public class MainActivity extends AppCompatActivity implements

EditNameDialogFragment.EditNameDialogListener {

// ...

Implement interface có phương thức trả lại dữ liệu

@Override

public void **onFinishEditDialog**(String inputText) {

 Toast.makeText(this, "Hi, " + inputText, Toast.LENGTH_SHORT).show();

}

}

Truy cập dữ liệu được truyền cho activity

