

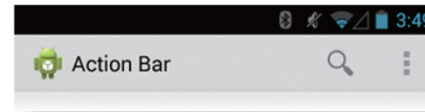
# CHAP 5. 메뉴와 대화상자

# 안드로이드의 사용자 인터페이스

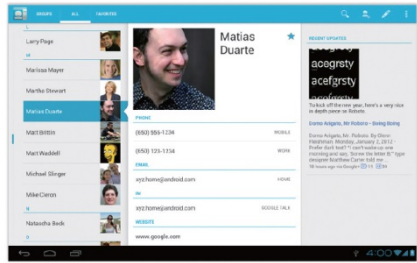
- 네비게이션 바



- 액션바



- 다중 패널 레이아웃



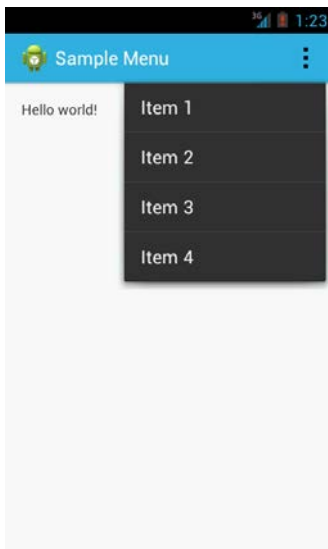
- 제스처



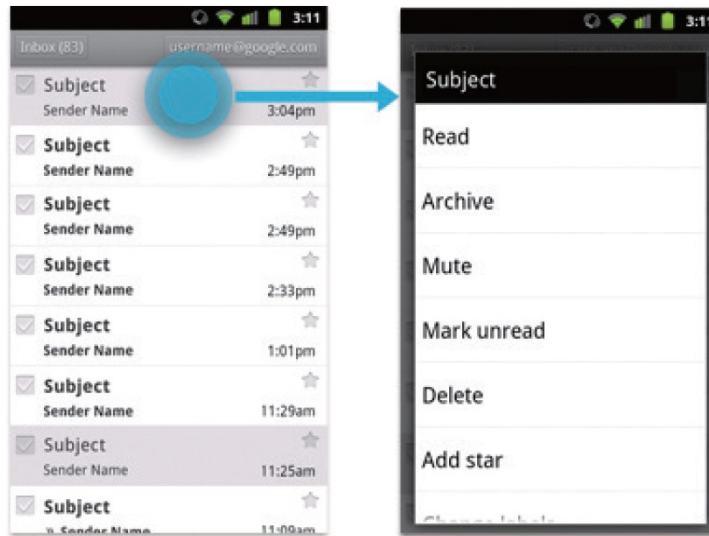
# 메뉴의 종류

- **Option Menu:** 사용자가 MENU 키를 누를 때 나타난다.
- **Context Menu:** Context 메뉴는 사용자가 화면을 일정 시간 이상으로 길게 누르면 나타나는 메뉴이다.
- **Popup Menu**

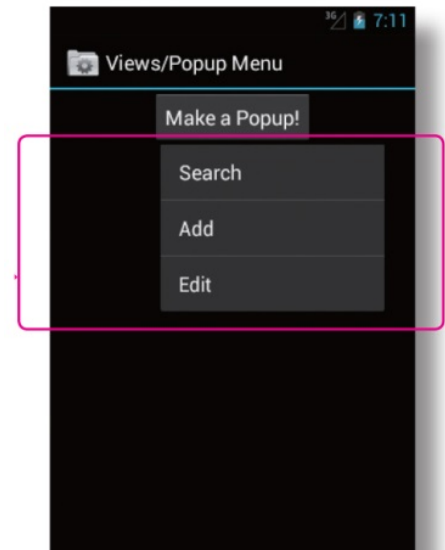
옵션 메뉴



컨텍스트 메뉴



팝업메뉴



# 메뉴를 생성하는 방법

- 코드로 메뉴 생성하기
- XML로 메뉴 생성하기

# 메뉴도 XML로 정의

- xml 정의의 장점
  - 메뉴 구조의 시각화
  - 코드와 메뉴의 분리로 수정/확장 용이
  - 실행 환경에 적응적인 메뉴 구성

*mymenu.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >

  <item
    android:id="@+id/apple"
    android:icon="@drawable/ic_launcher"
    android:title="사과"/>
  <item
    android:id="@+id/grape"
    android:icon="@drawable/ic_launcher"
    android:title="포도"/>
  <item
    android:id="@+id/banana"
    android:icon="@drawable/ic_launcher"
    android:title="바나나"/>

</menu>
```

<menu> 엘리먼트는 Menu를 생성하고 이것은 메뉴 항목들을 담는 컨테이너가 된다. 반드시 루트 노드이어야 하며 <item>이나 <group>을 하나 이상 포함한다.

<item> 엘리먼트는 MenuItem을 생성한다. 이것은 하나의 메뉴 항목을 나타낸다. 서브 메뉴를 작성하려면 <menu>를 포함할 수 있다.

# showAsAction

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+id/new_game"
        android:icon="@drawable/ic_new_game"
        android:title="@string/new_game"
        android:showAsAction="ifRoom"/>
  <item android:id="@+id/help"
        android:icon="@drawable/ic_help"
        android:title="@string/help" />
</menu>
```

## **android:showAsAction**

Specifies when and how this item should appear as an action item in the menu bar

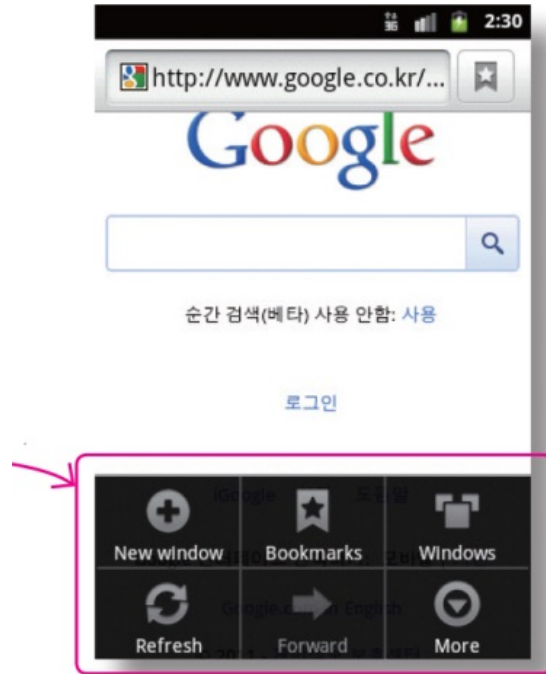
# 메뉴 팽창

- 메뉴 리소스를 **팽창(inflate)**하면 실제 메뉴가 생성

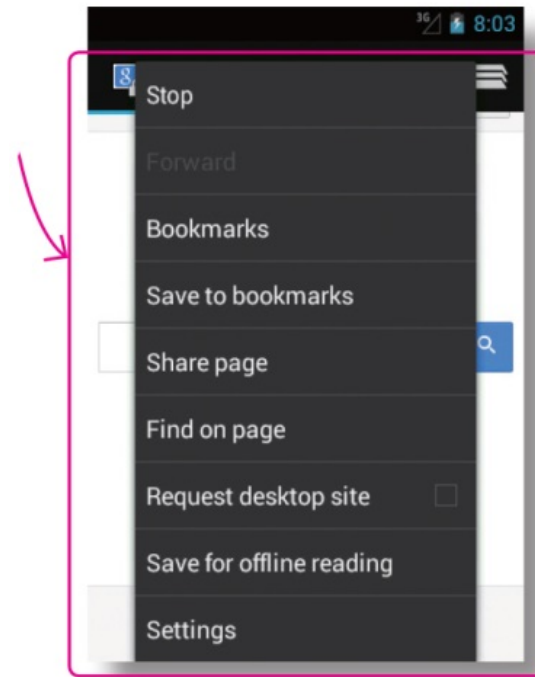


# Option Menu

- 현재 액티비티와 관련된 동작



2.3이하



2.3이상



# 메뉴 리소스 팽창

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.mymenu, menu);
    return true;
}
```

# 옵션 메뉴 클릭 이벤트 처리

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.new_game:
            newGame();
            return true;
        case R.id.help:
            showHelp();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```

← "새로운 게임" 메뉴 항목  
이 클릭되면 실행

← "도움말" 메뉴 항목이 클  
릭되면 실행

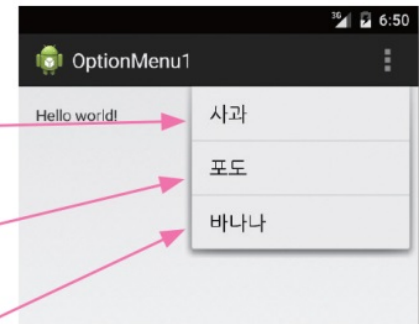
← 이벤트를 상위 클래스로  
전달

# 예제: 옵션 메뉴 생성

- 메뉴를 나타내는 XML 파일 생성

*mymenu.xml*

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
    <item
        android:id="@+id/apple"
        android:icon="@drawable/ic_launcher"
        android:title="사과" />
    <item
        android:id="@+id/grape"
        android:icon="@drawable/ic_launcher"
        android:title="포도" />
    <item
        android:id="@+id/banana"
        android:icon="@drawable/ic_launcher"
        android:title="바나나" />
</menu>
```



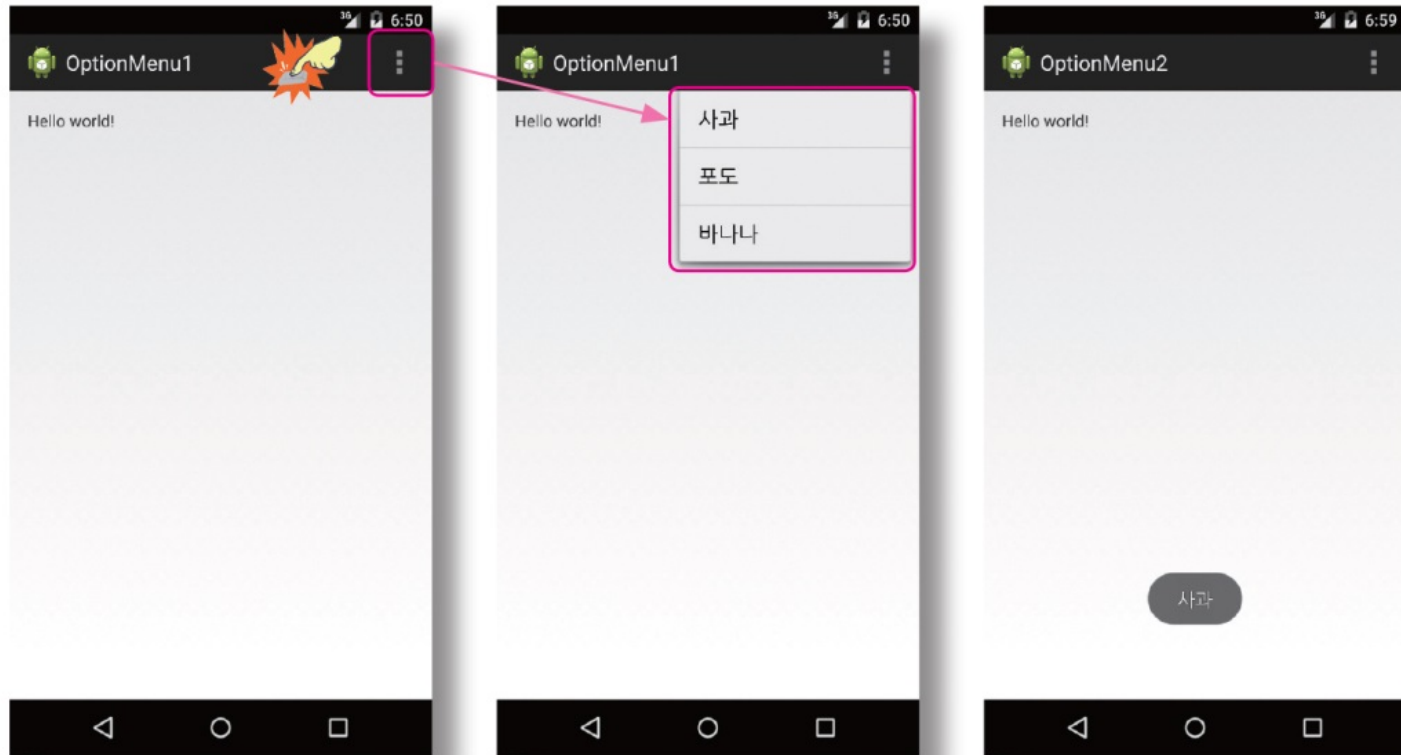
# 옵션 메뉴 생성

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu) {  
        MenuInflater inflater = getMenuInflater();  
        inflater.inflate(R.menu.options, menu);  
        return super.onCreateOptionsMenu(menu);  
    }  
}
```

# 옵션 메뉴 이벤트 처리

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.apple:
            Toast.makeText(this, "사과", Toast.LENGTH_SHORT).show();
            return true;
        case R.id.grape:
            Toast.makeText(this, "포도", Toast.LENGTH_SHORT).show();
            return true;
        case R.id.banana:
            Toast.makeText(this, "바나나", Toast.LENGTH_SHORT).show();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```

# 실행 결과



# 코드로 옵션 메뉴 생성하기 #1

```
Menu::add(int groupId, int itemId, int order, CharSequence title);
```

```
@Override
```

```
public boolean onCreateOptionsMenu(Menu menu) {  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.options, menu);
```

```
    MenuItem item = menu.add(0, 1, 0, "복숭아");  
    item.setIcon(R.drawable.ic_launcher);
```

```
    return super.onCreateOptionsMenu(menu);
```

```
}
```

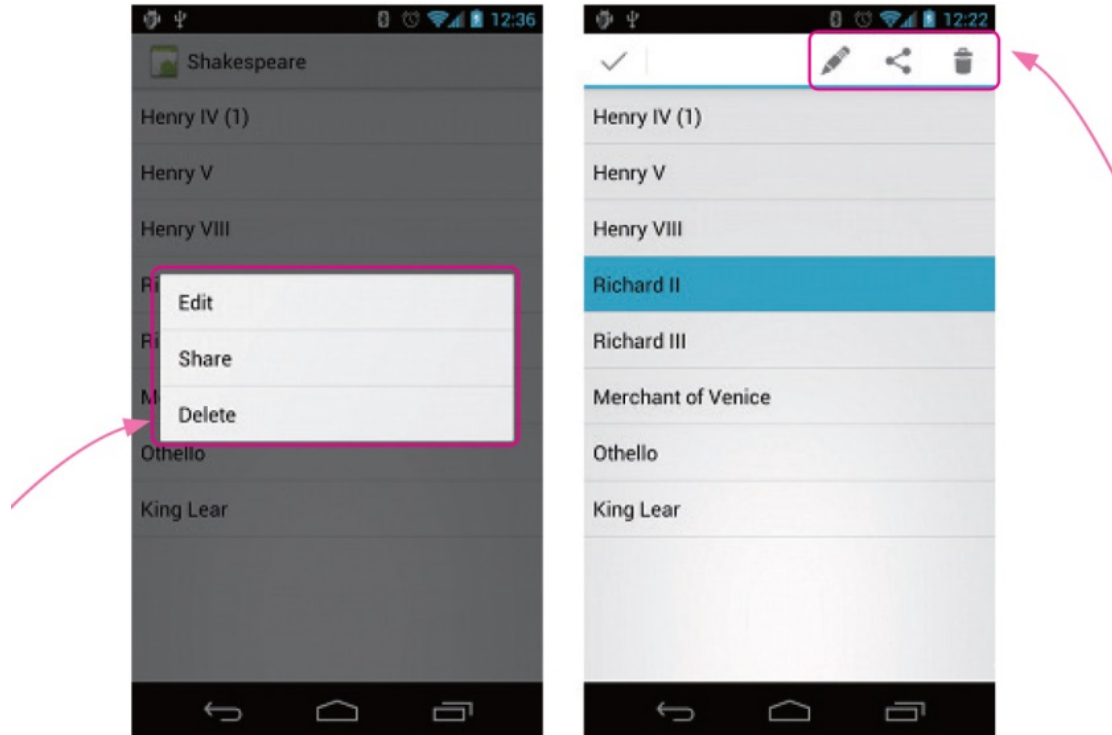
## 코드로 옵션 메뉴 생성하기 #2

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case 1:
            Toast.makeText(this, "복숭아", Toast.LENGTH_SHORT).show();
            return true;
        case R.id.apple:
            Toast.makeText(this, "사과", Toast.LENGTH_SHORT).show();
            return true;
        case R.id.grape:
            Toast.makeText(this, "포도", Toast.LENGTH_SHORT).show();
            return true;
        case R.id.banana:
            Toast.makeText(this, "바나나", Toast.LENGTH_SHORT).show();
            return true;
        default:
            return super.onOptionsItemSelected(item);
    }
}
```



# Context Menu

- UI의 어떤 항목과 관련된 동작



# Context Menu 종류

- **플로팅 컨텍스트 메뉴:**

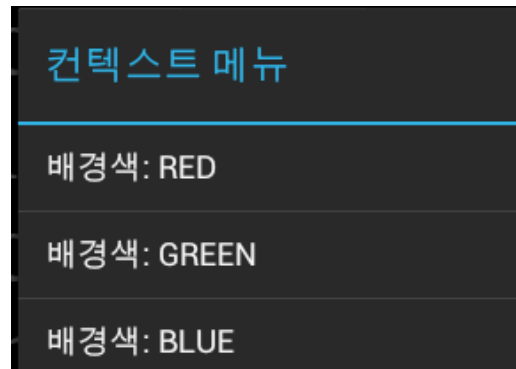
- 사용자가 항목 위에서 오래 누르기(long click)를 하면 메뉴가 대화 상자처럼 떠서 표시된다.

- **컨텍스트 액션 모드:**

- 현재 선택된 항목에 관련된 메뉴가 액션바에 표시된다. 여러 항목을 선택하여 특정한 액션을 한꺼번에 적용할 수 있다.

# 컨텍스트 메뉴

- 사용자가 항목 위에서 “오래 누르기”(long-press)를 하면 컨텍스트 메뉴가 표시
- PC에서 마우스 오른쪽 버튼을 눌렀을 때 나오는 메뉴와 개념적으로 유사



# 컨텍스트 메뉴 예제

- 사용자 인터페이스 작성

```
<?xml version="1.0" encoding="UTF-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/LinearLayout01"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/TextView01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Only I can change my life. No one can do it for me."
        android:textSize="50px"
        android:typeface="serif" />
</LinearLayout>
```

# 컨텍스트 메뉴 예제

- Context menu에 연동될 view 객체 등록 (하나 이상 가능)

```
public class MainActivity extends AppCompatActivity {  
    TextView text;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        text = (TextView) findViewById(R.id.TextView01);  
        registerForContextMenu(text);  
    }  
}
```

# 컨텍스트 메뉴

- 컨텍스트 메뉴는 다음과 같은 메소드가 호출되면서 생성된다.

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v,
                                ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    menu.setHeaderTitle("컨텍스트 메뉴");
    menu.add(0, 1, 0, "배경색: RED");
    menu.add(0, 2, 0, "배경색: GREEN");
    menu.add(0, 3, 0, "배경색: BLUE");
}
```

컨텍스트 메뉴

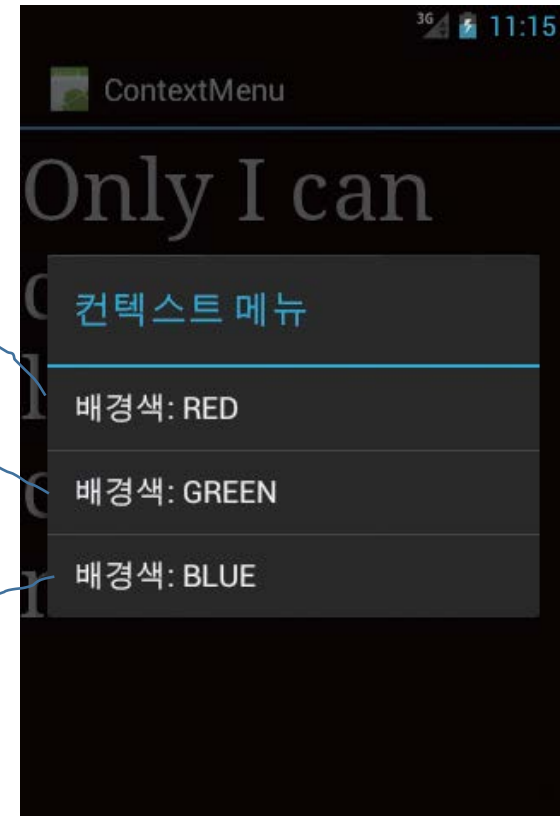
배경색: RED

배경색: GREEN

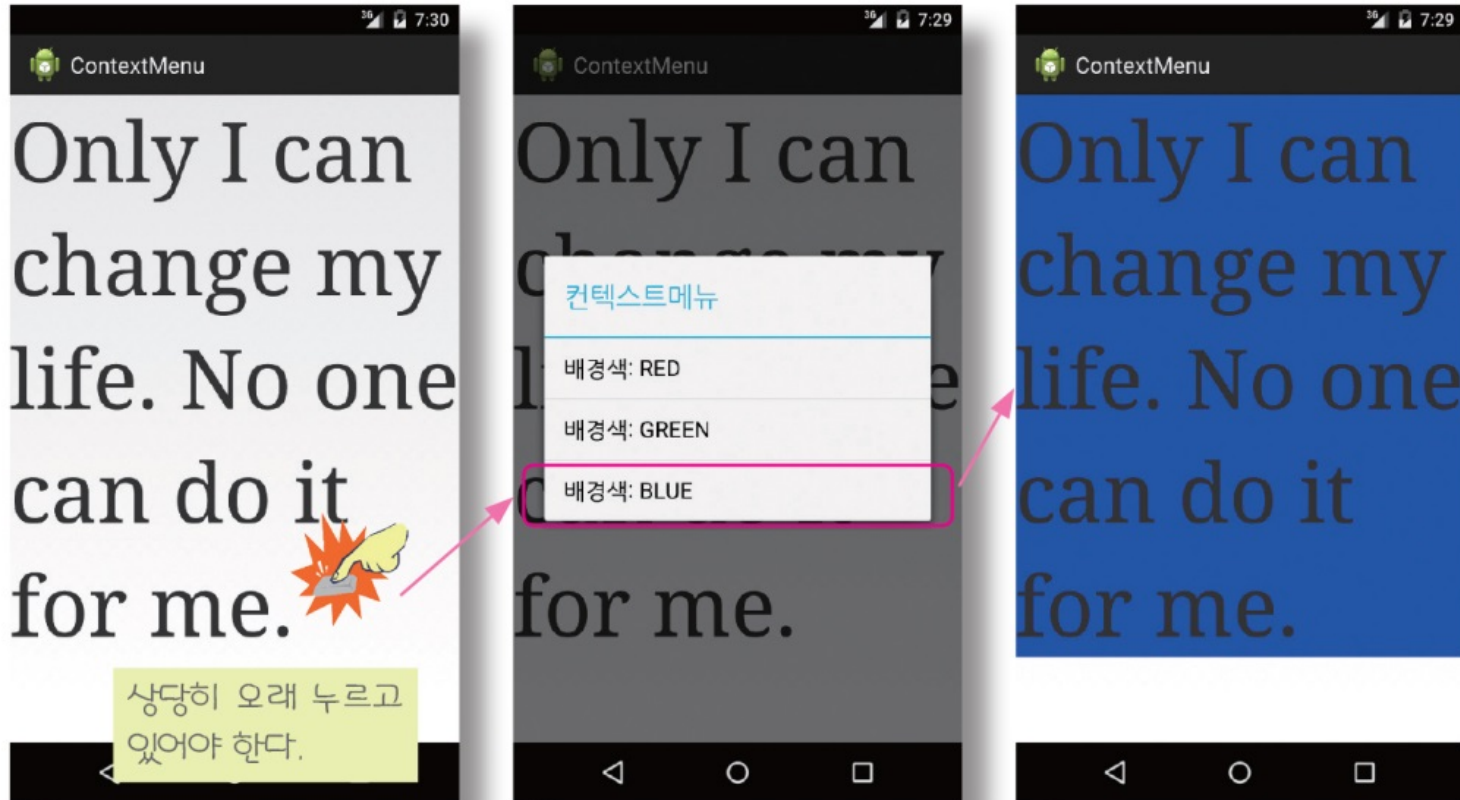
배경색: BLUE

# 컨텍스트 메뉴 이벤트 처리

```
public boolean onContextItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case 1:  
            text.setBackgroundColor(Color.RED);  
            return true;  
        case 2:  
            text.setBackgroundColor(Color.GREEN);  
            return true;  
        case 3:  
            text.setBackgroundColor(Color.BLUE);  
            return true;  
        default:  
            return super.onContextItemSelected(item);  
    }  
}
```



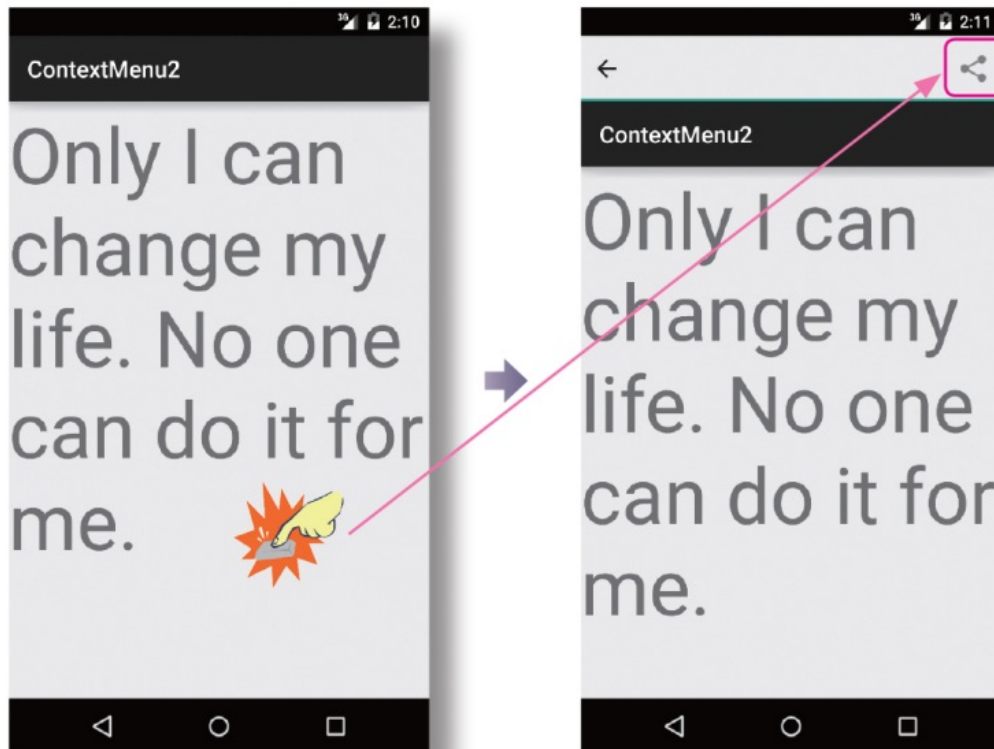
# 실행 결과





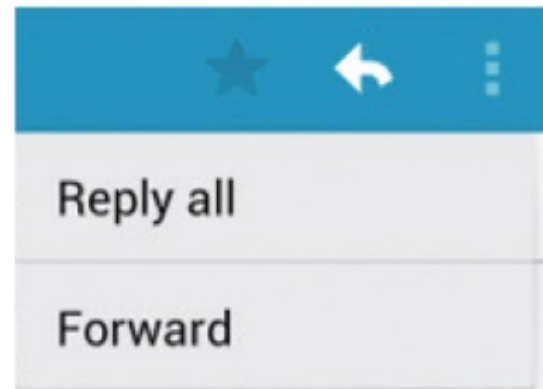
# 컨텍스트 액션 모드

- 현재 선택된 항목에 대하여 수행할 수 있는 액션들을 제공하는 컨텍스트 액션바가 화면의 상단에 표시된다.



# 팝업 메뉴

- 뷰에 부착된 모달 메뉴(modal menu)
- API 레벨 11부터 제공
- 팝업 메뉴의 용도
  - 오버플로우 스타일 메뉴 제공
  - 서브 메뉴의 역할
  - 드롭다운 메뉴



# 예제: 팝업 메뉴

activity\_main.xml

---

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <Button android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:onClick="onClick"
        android:text="show popup menu" />
</LinearLayout>
```

버튼을 누르면 `onClick()`이 호출된다.

# 예제: 팝업 메뉴

popup.xml

---

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android" >

    <item
        android:id="@+id/search"
        android:icon="@android:drawable/ic_menu_search"
        android:title="search"/>

    <item
        android:id="@+id/add"
        android:icon="@android:drawable/ic_menu_add"
        android:title="add"/>

    <item
        android:id="@+id/edit"
        android:icon="@android:drawable/ic_menu_edit"
        android:title="edit">
        <menu>
            <item
                android:id="@+id/share"
                android:icon="@android:drawable/ic_menu_share"
                android:title="share"/>
        </menu>
    </item>

</menu>
```

메뉴 안에 위치하는 서브  
메뉴이다.

# 예제: 팝업 메뉴

MainActivity.java

```
package kr.co.company.popupmenu;
```

```
// 소스만 입력하고 Alt+Enter를 눌러서 import 문장을 자동으로 생성한다.
```

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

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
    }
```

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

```
        PopupMenu popup = new PopupMenu(this, button);
```

```
        popup.getMenuInflater().inflate(R.menu.popup, popup.getMenu());
```

```
        popup.setOnMenuItemClickListener(
```

```
            new PopupMenu.OnMenuItemClickListener() {
```

```
                public boolean onMenuItemClick(MenuItem item) {
```

```
                    Toast.makeText(getApplicationContext(), "클릭된 팝업 메뉴: "
```

```
                        + item.getTitle(),
```

```
                        Toast.LENGTH_SHORT).show();
```

```
                    return true;
```

```
                }
```

```
            });
```

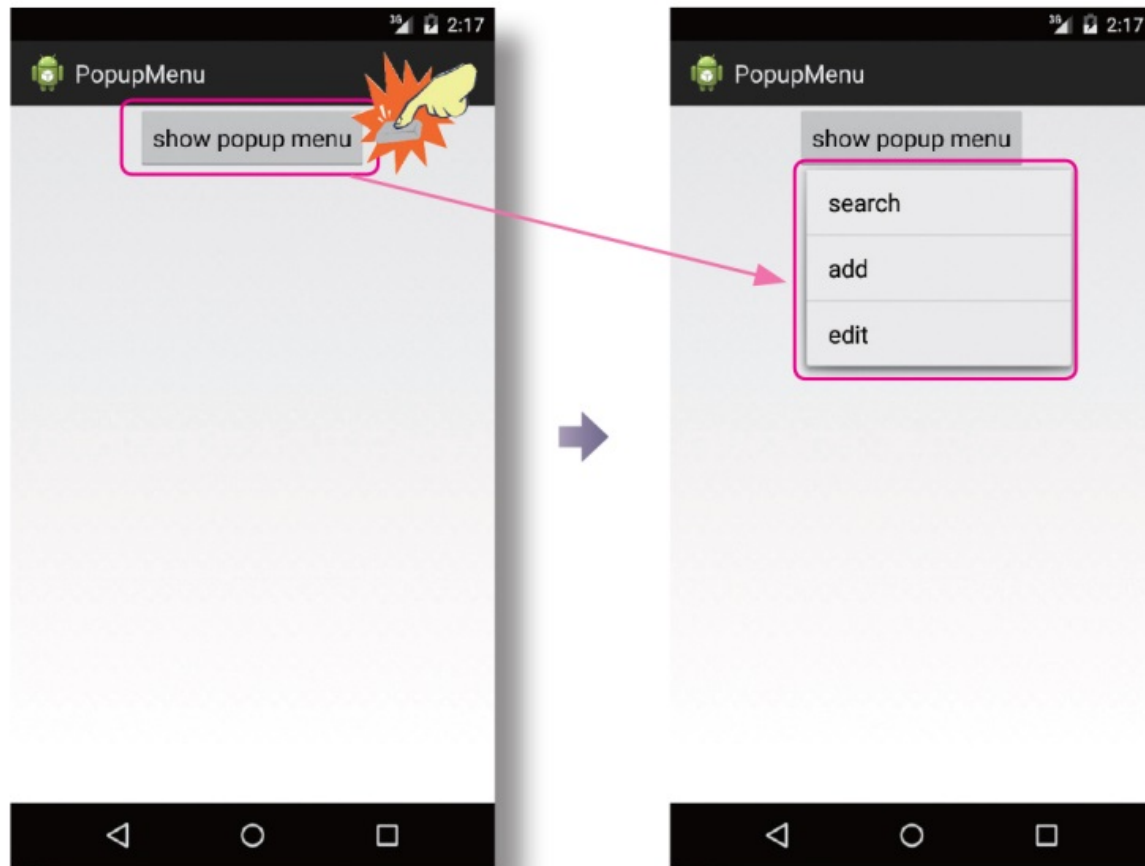
```
        popup.show();
```

```
    }
```

```
}
```

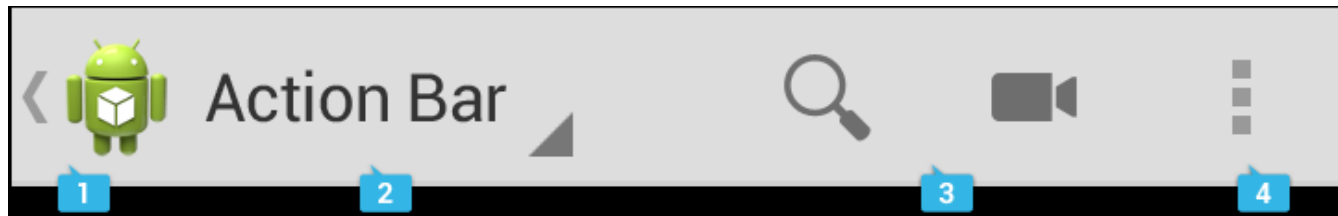
팝업 메뉴 항목이 눌러지면 여기서 처리한다.

# 실행 결과



# 액션바

- 애플리케이션의 로고 표시
- 탭이나 드롭다운 리스트를 이용하여서 내비게이션 메뉴와 뷰 전환 기능을 제공한다.



앱아이콘

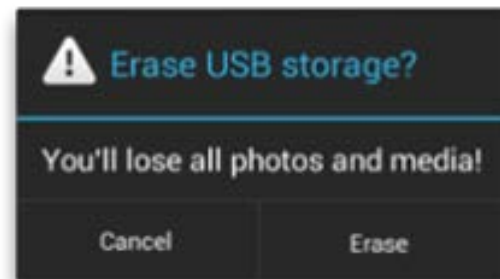
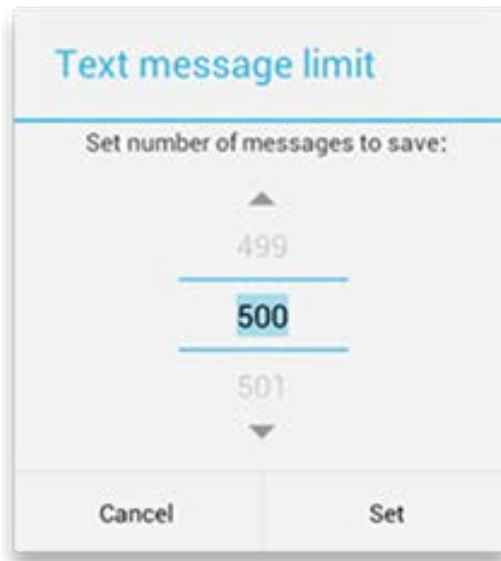
뷰컨트롤

액션버튼

액션 오버플로우

# 대화 상자

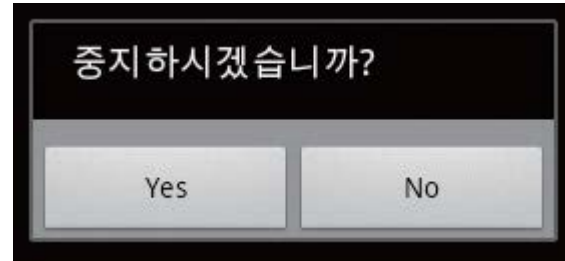
- **대화 상자(dialog)**는 사용자에게 메시지를 출력하고 사용자로 부터 입력을 받아들이는 아주 보편적인 사용자 인터페이스



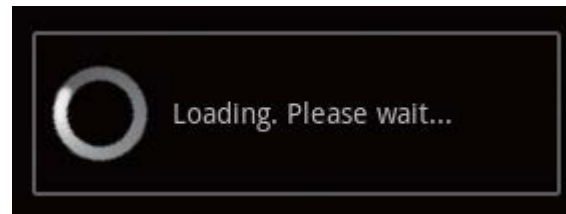


# 대화 상자의 종류

- AlertDialog



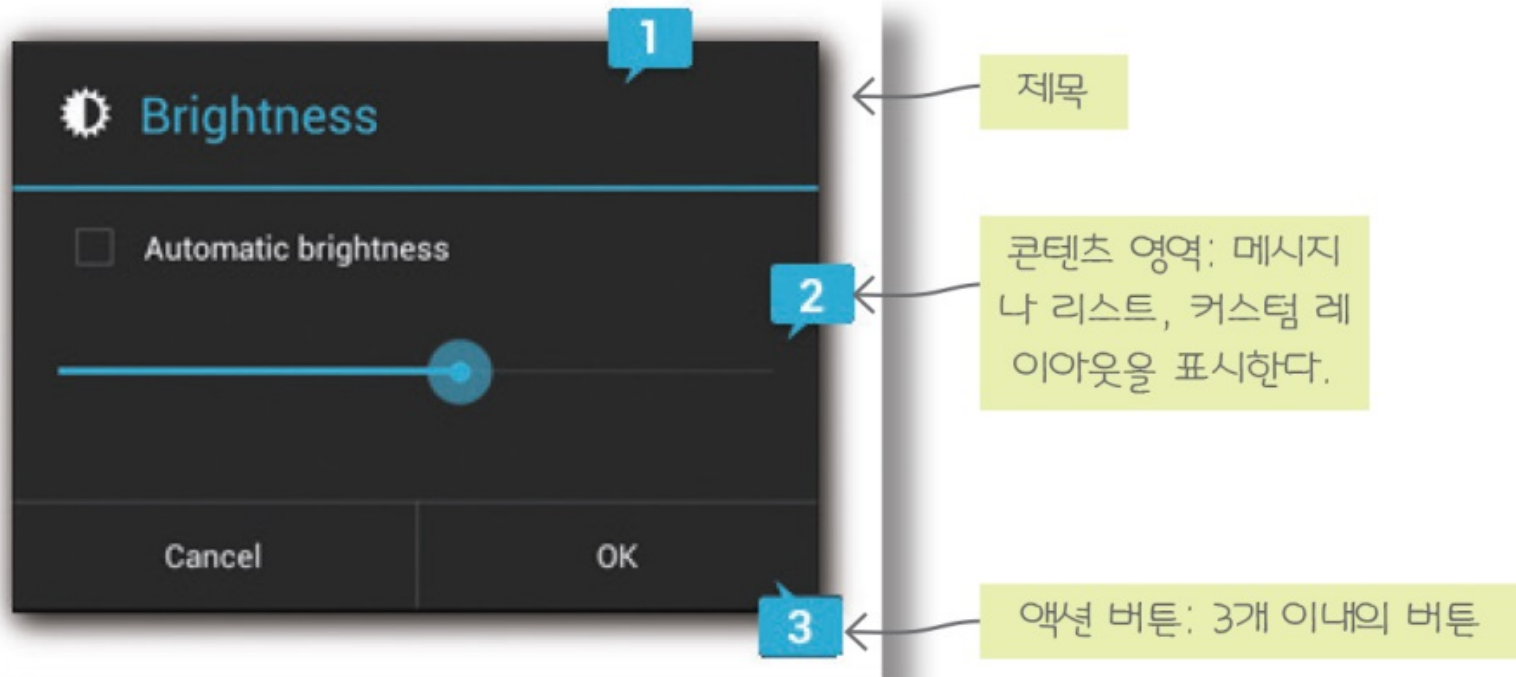
- ProgressDialog



- DatePickerDialog
- TimePickerDialog



# AlertDialog

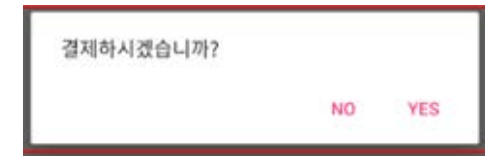


```

public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

```

대화 상자의 메시지 부분 설정



```

    public void open(View view){
        AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);
        alertDialogBuilder.setMessage("결제하시겠습니까?");
        alertDialogBuilder.setPositiveButton("yes",
            new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface arg0, int arg1) {
                    Toast.makeText(MainActivity.this, "결제가 완료되었습니다.",
                        Toast.LENGTH_LONG).show();
                }
            });

```

대화 상자의 "yes" 버튼 부분 설정

```

        alertDialogBuilder.setNegativeButton("No",
            new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    Toast.makeText(MainActivity.this, "결제가 취소되었습니다.",
                        Toast.LENGTH_LONG).show();
                    finish();
                }
            });

```

대화 상자의 "no" 버튼 부분 설정

```

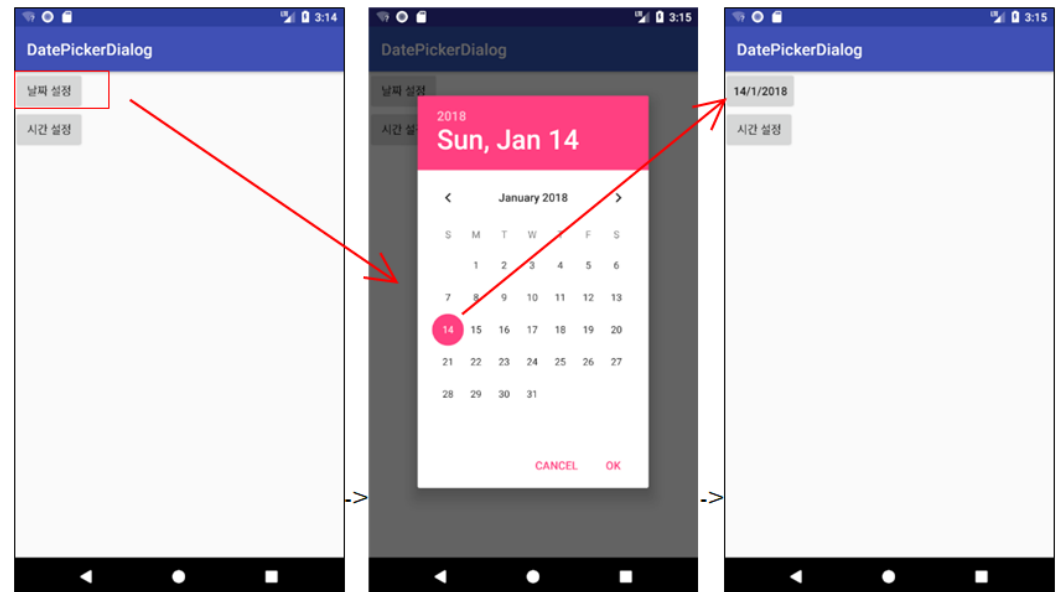
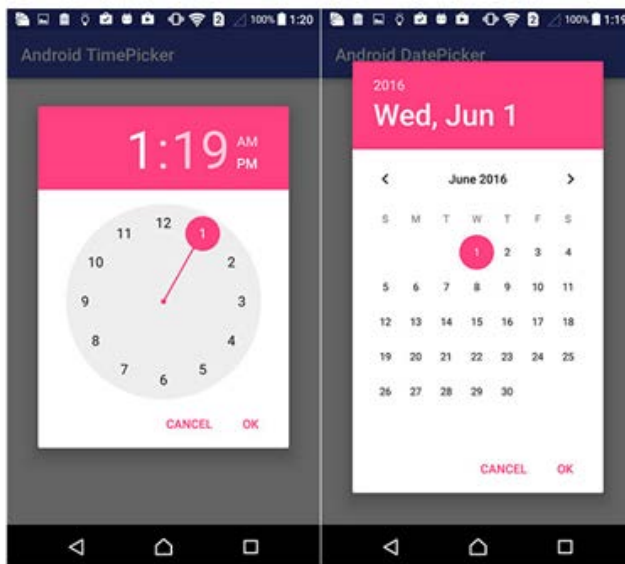
        AlertDialog alertDialog = alertDialogBuilder.create();
        alertDialog.show();
    }
}

```

}

# DatePickerDialog

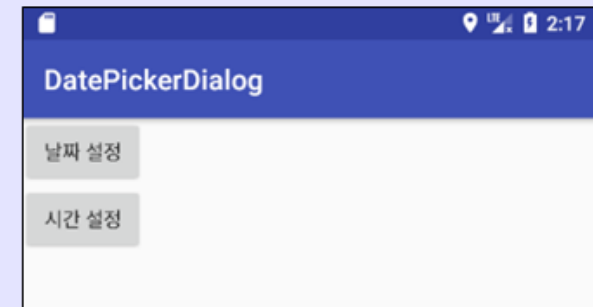
- 날짜와 시간을 입력받는 대화 상자



# DatePickerDialog

activity\_main.xml

```
<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" >
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onClick"
        android:text="날짜 설정" />
    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="onClick"
        android:text="시간 설정" />
</LinearLayout>
```



# DatePickerDialog

```
public class MainActivity extends AppCompatActivity {  
    Button btnSelectDate, btnSelectTime;  
    DatePickerDialog datePickerDialog;  
    TimePickerDialog timePickerDialog;
```

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
    btnSelectDate=(Button)findViewById(R.id.button1);  
    btnSelectTime=(Button)findViewById(R.id.button2);  
}
```

```
public void onClick(View v) {  
    if (v == btnSelectDate) {  
        final Calendar c = Calendar.getInstance();  
        int mYear = c.get(Calendar.YEAR);  
        int mMonth = c.get(Calendar.MONTH);  
        int mDay = c.get(Calendar.DAY_OF_MONTH);
```

// 현재 시간을 얻는다.

```

DatePickerDialog datePickerDialog = new DatePickerDialog(this,
    new DatePickerDialog.OnDateSetListener() {
        @Override
        public void onDateSet(DatePicker view, int year,
                                int monthOfYear, int dayOfMonth) {
            btnSelectDate.setText(dayOfMonth + "/" + (monthOfYear +
1) + "/" + year);
        }
    }, mYear, mMonth, mDay);
datePickerDialog.show();
}
if (v == btnSelectTime) {
    final Calendar c = Calendar.getInstance();
    int mHour = c.get(Calendar.HOUR_OF_DAY);
    int mMinute = c.get(Calendar.MINUTE);

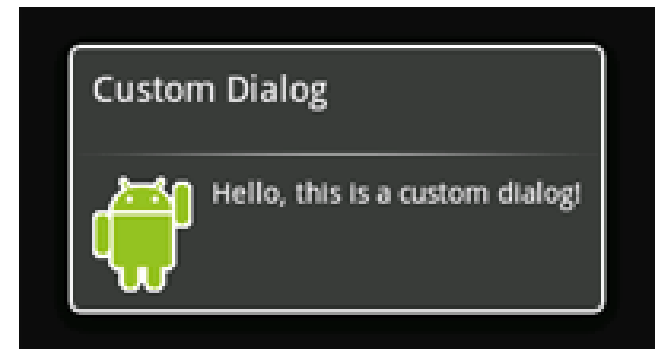
    TimePickerDialog timePickerDialog = new TimePickerDialog(this,
        new TimePickerDialog.OnTimeSetListener() {
            @Override
            public void onTimeSet(TimePicker view, int hourOfDay,
                                    int minute) {
                btnSelectTime.setText(hourOfDay + ":" + minute);
            }
        }, mHour, mMinute, false);
    timePickerDialog.show();
}
}
}

```

---

# 커스텀 대화 상자

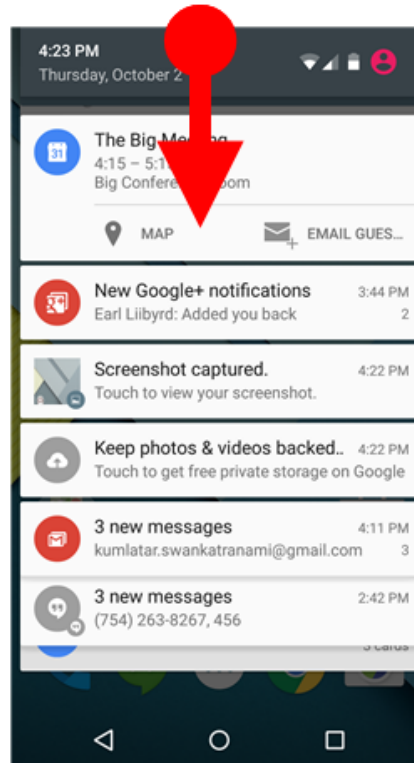
- 사용자가 마음대로 대화 상자의 내용을 디자인할 수 있는 대화 상자
  1. Dialog 인스턴스 생성
  2. setContentView()로 레이아웃 생성.
  3. Dialog 버튼 이벤트 리스너 설정
  4. Dialog.show() 호출





# Notification

- **알림기능(notification)**은 어떤 이벤트가 발생하였을 때, 앱이 사용자에게 전달하는 메시지이다.



# Notification 절차

## 1. 알림 빌더 생성

```
NotificationCompat.Builder builder = new NotificationCompat.Builder(this)
```

## 2. 알림 속성 설정

```
builder.setSmallIcon(R.drawable.notification_icon);  
builder.setTitle("알려드립니다.");  
builder.setContentText("이것은 시험적인 알림입니다.");
```

## 3. 액션 첨부

```
Intent intent = new Intent(Intent.ACTION_VIEW, Uri.parse("http://www.google.com/"));  
PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent, 0);  
builder.setContentIntent(pendingIntent);
```

## 4. 알림 생성

```
NotificationManager notificationManager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);  
notificationManager.notify(NOTIFICATION_ID, builder.build());
```

# 예제: 버튼을 누르면 알림을 보내는 앱

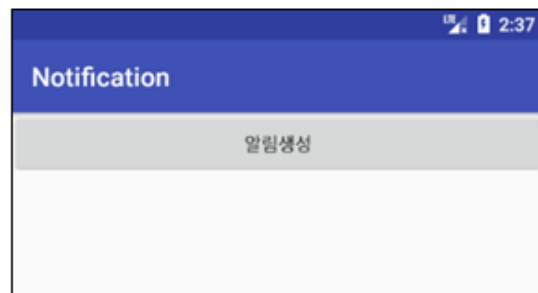
activity\_main.xml

---

```
<?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="match_parent"
    android:orientation="vertical">
    |
    <Button
        android:id="@+id/button"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="sendNotification"
        android:text="알림 생성" />

</LinearLayout>
```

---



# 코드

```
public class MainActivity extends AppCompatActivity {  
  
    // 각 알림을 식별하는 정수  
    public static final int NOTIFICATION_ID = 1;  
  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
  
    public void sendNotification(View view) {  
  
        // NotificationCompat.Builder를 사용하여 알림을 설정한다.  
        NotificationCompat.Builder builder = new NotificationCompat.Builder(  
            this);  
  
        // 알림에 나타나는 아이콘  
        builder.setSmallIcon(R.drawable.ic_launcher);  
  
        // 알림이 클릭되면 이 인텐트가 보내진다.  
        Intent intent = new Intent(Intent.ACTION_VIEW,  
            Uri.parse("http://www.google.com/"));  
        PendingIntent pendingIntent = PendingIntent.getActivity(this, 0,  
            intent, 0);  
    }  
}
```

```
// 사용자가 알림을 터치하면 인텐트가 보내진다.
builder.setIntent(pendingIntent);

// 알림에 표시되는 큰 아이콘
builder.setLargeIcon(BitmapFactory.decodeResource(getResources(),
    R.drawable.ic_launcher));

// 알림 제목
builder.setTitle("알려드립니다.");

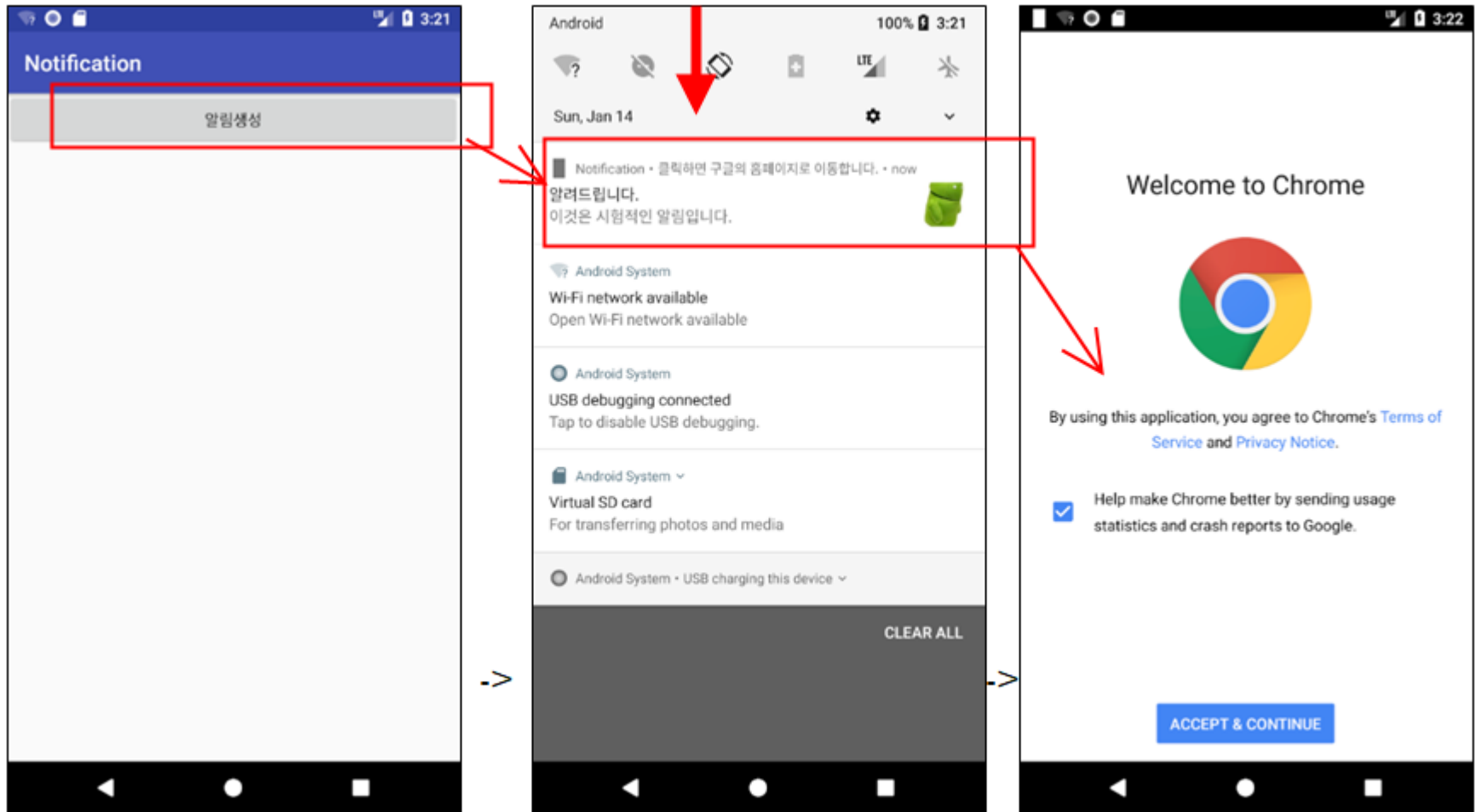
// 알림 콘텐츠
builder.setText("이것은 시험적인 알림입니다.");

// 4.2 이상인 경우에 보여지는 서브 텍스트
builder.setSubText("클릭하면 구글의 홈페이지로 이동합니다.");

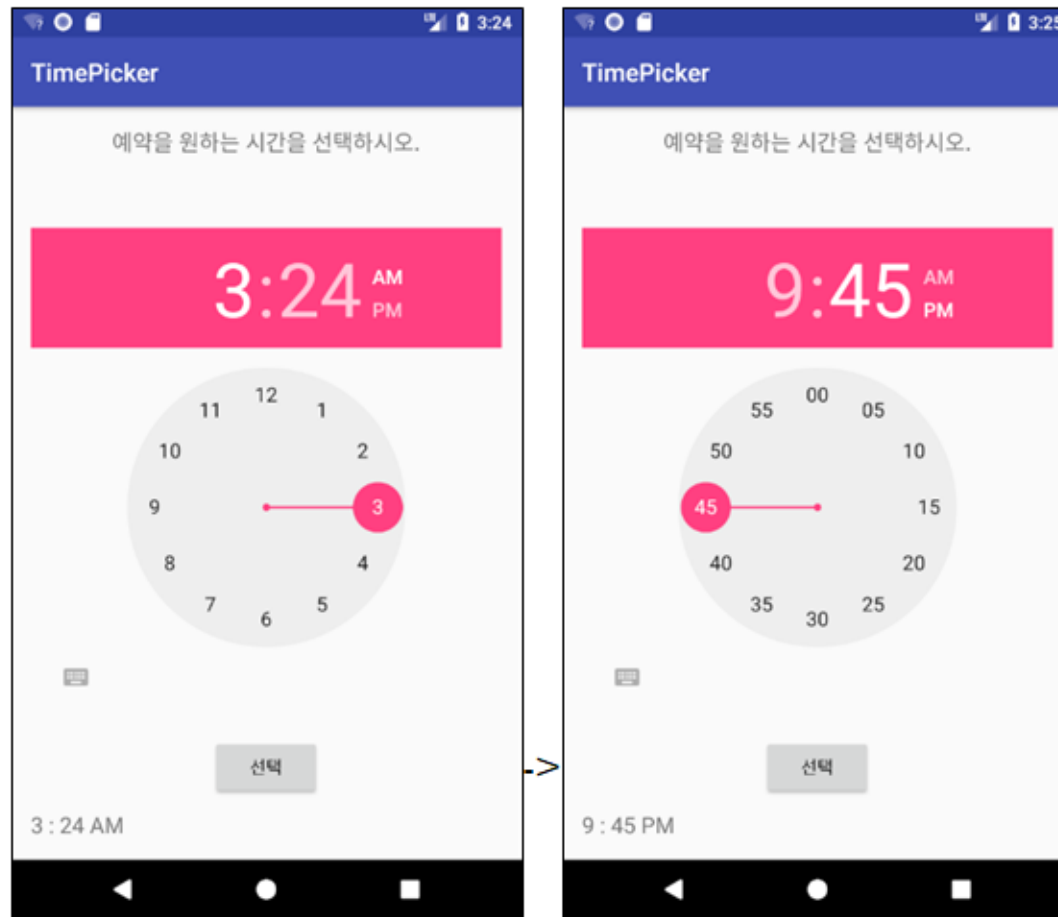
NotificationManager notificationManager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);

// 알림바에 알림을 표시한다.
notificationManager.notify(NOTIFICATION_ID, builder.build());
}
}
```

# 실행 결과



# Lab: 예약 앱 작성



```
public class MainActivity extends AppCompatActivity {  
    private TimePicker timePicker1;  
    private TextView time;  
    private Calendar calendar;  
    private String format = "";
```

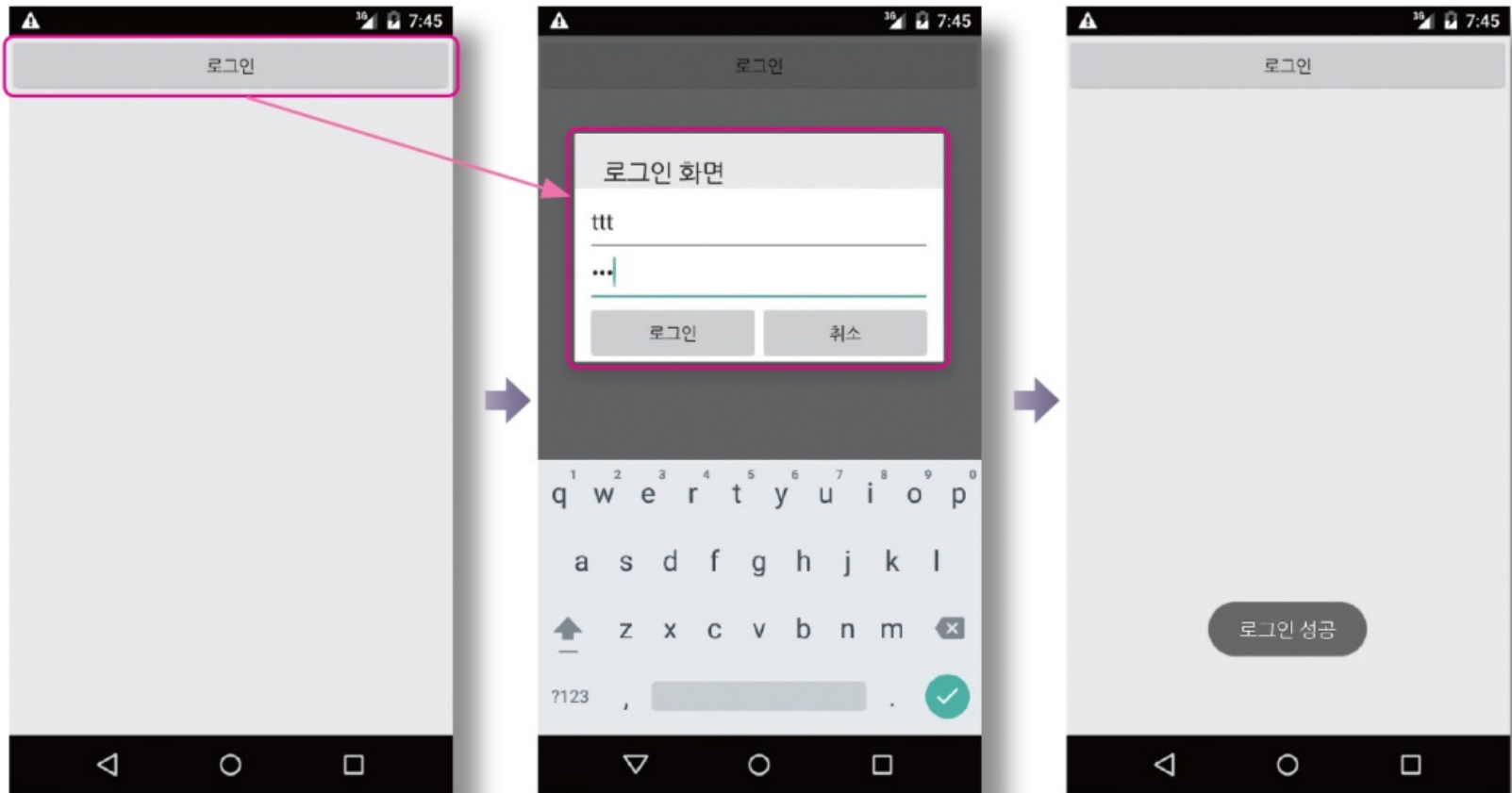
@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    timePicker1 = (TimePicker) findViewById(R.id.timePicker1);  
    time = (TextView) findViewById(R.id.textView3);  
    calendar = Calendar.getInstance();  
  
    int hour = calendar.get(Calendar.HOUR_OF_DAY);  
    int min = calendar.get(Calendar.MINUTE);  
    showTime(hour, min);  
}  
  
public void displayTime(View view) {  
    int hour = timePicker1.getHour();  
    int min = timePicker1.getMinute();  
    showTime(hour, min);  
}
```



```
public void showTime(int hour, int min) {  
    if (hour == 0) {  
        hour += 12;  
        format = "AM";  
    } else if (hour == 12) {  
        format = "PM";  
    } else if (hour > 12) {  
        hour -= 12;  
        format = "PM";  
    } else {  
        format = "AM";  
    }  
  
    time.setText(new StringBuilder().append(hour).append(" : ").append(min)  
        .append(" ").append(format));  
}  
}
```

# 예제: 로그인 대화 상자



# 대화 상자를 XML로 정의

*custom\_dialog.xml*

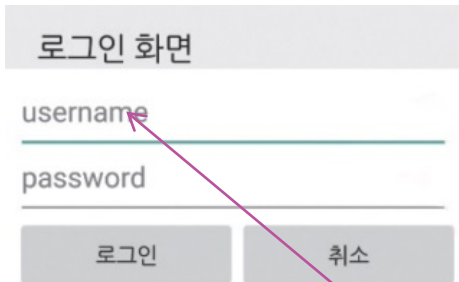


```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="300dp"
    android:layout_height="match_parent"
    android:background="#fff"
    android:orientation="vertical" >
```

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

    <requestFocus>
    </requestFocus>
</EditText>
```

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



```
<LinearLayout  
    android:id="@+id/linearLayout1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content" >
```

```
<Button  
    android:id="@+id/login"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_weight="0.5"  
    android:text="로그인" >  
</Button>
```

```
<Button  
    android:id="@+id/cancel"  
    android:layout_width="wrap_content"  
  
    android:layout_height="wrap_content"  
    android:layout_weight="0.5"  
    android:text="취소" >
```

```
</Button>
```

```
</LinearLayout>
```

```
</LinearLayout>
```

# 코드

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

```
@Override
public void onClick(View v) {
    final Dialog loginDialog = new Dialog(this);
    loginDialog.setContentView(R.layout.custom_dialog);
    loginDialog.setTitle("로그인 화면");

    Button login = (Button) loginDialog.findViewById(R.id.login);
    Button cancel = (Button) loginDialog.findViewById(R.id.cancel);
```

버튼이 눌려지면 커스텀 대화 상자를 생성한다.

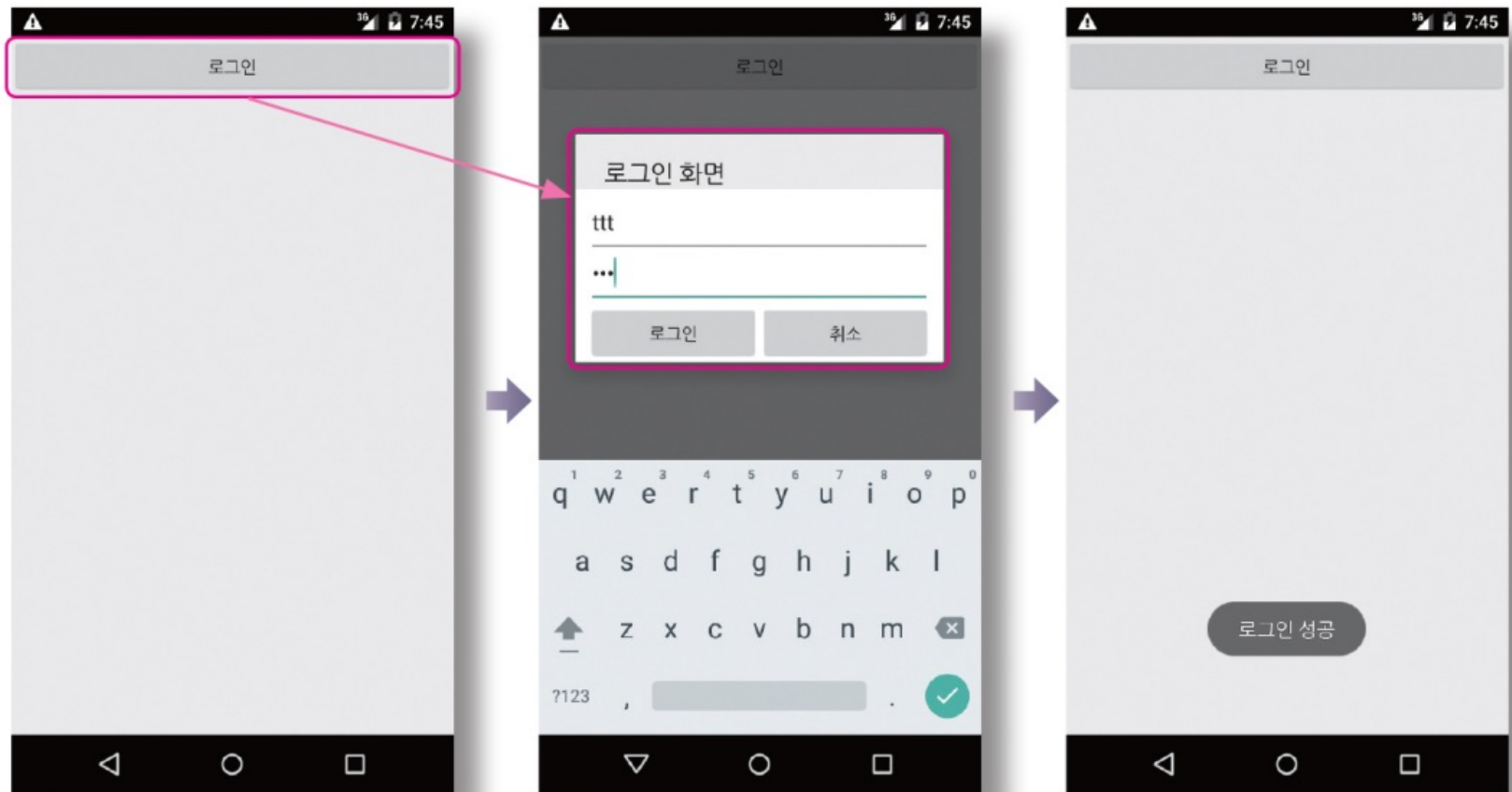
커스텀 대화 상자의 화면이 custom\_dialog.xml에 저장되어 있음

```
final EditText username = (EditText) loginDialog
    .findViewById(R.id.username);
final EditText password = (EditText) loginDialog
    .findViewById(R.id.password);
```

```
login.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (username.getText().toString().trim().length() > 0
            && password.getText().toString().trim().length() > 0) {
            Toast.makeText(getApplicationContext(), "로그인 성공",
                Toast.LENGTH_LONG).show();

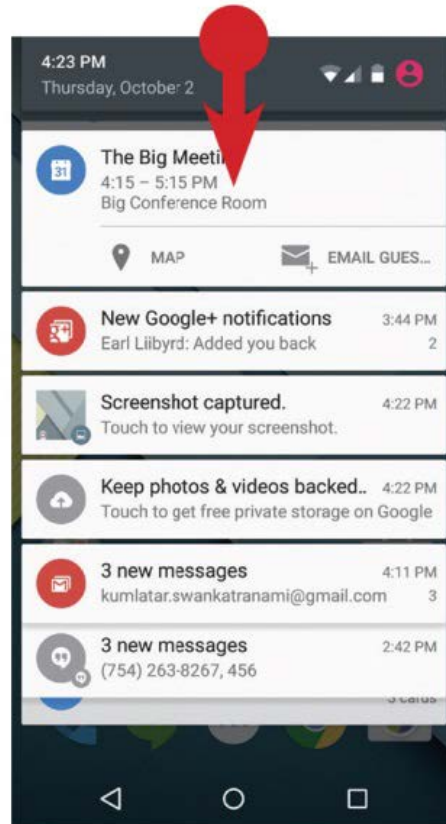
            loginDialog.dismiss();
        } else {
            Toast.makeText(getApplicationContext(), "다시 입력하십시오",
                Toast.LENGTH_LONG).show();
        }
    }
});
```

# 실행 결과



# 알림 기능

- **알림기능(notification)**은 어떤 이벤트가 발생하였을 때, 앱이 사용자에게 전달하는 메시지이다.



# 예제: 버튼을 누르면 알림을 보내는 앱

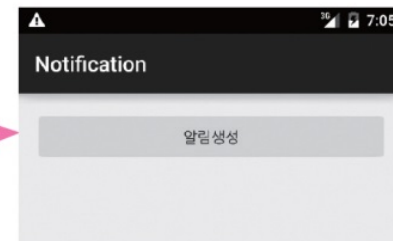
main.xml

<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"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".NotificationActivity" >
```

<Button

```
android:id="@+id/button"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_alignParentTop="true"
android:onClick="sendNotification"
android:text="알림생성" />
```



</RelativeLayout>



# 코드

```
public void sendNotification(View view) {  
  
    // NotificationCompat.Builder를 사용하여 알림을 설정한다.  
    NotificationCompat.Builder builder = new NotificationCompat.Builder(  
        this);  
  
    // 알림에 나타나는 아이콘  
    builder.setSmallIcon(R.drawable.ic_launcher);  
  
    // 알림이 클릭되면 이 인텐트가 보내진다.  
    Intent intent = new Intent(Intent.ACTION_VIEW,  
        Uri.parse("http://www.google.com/"));  
    PendingIntent pendingIntent = PendingIntent.getActivity(this, 0,  
        intent, 0);  
  
    // 사용자가 알림을 터치하면 인텐트가 보내진다.  
    builder.setContentIntent(pendingIntent);  
  
    // 알림에 표시되는 큰 아이콘  
    builder.setLargeIcon(BitmapFactory.decodeResource(getResources(),  
        R.drawable.ic_launcher));  
}
```

# 코드

```
// 알림 제목
builder.setTitle("알려드립니다.");

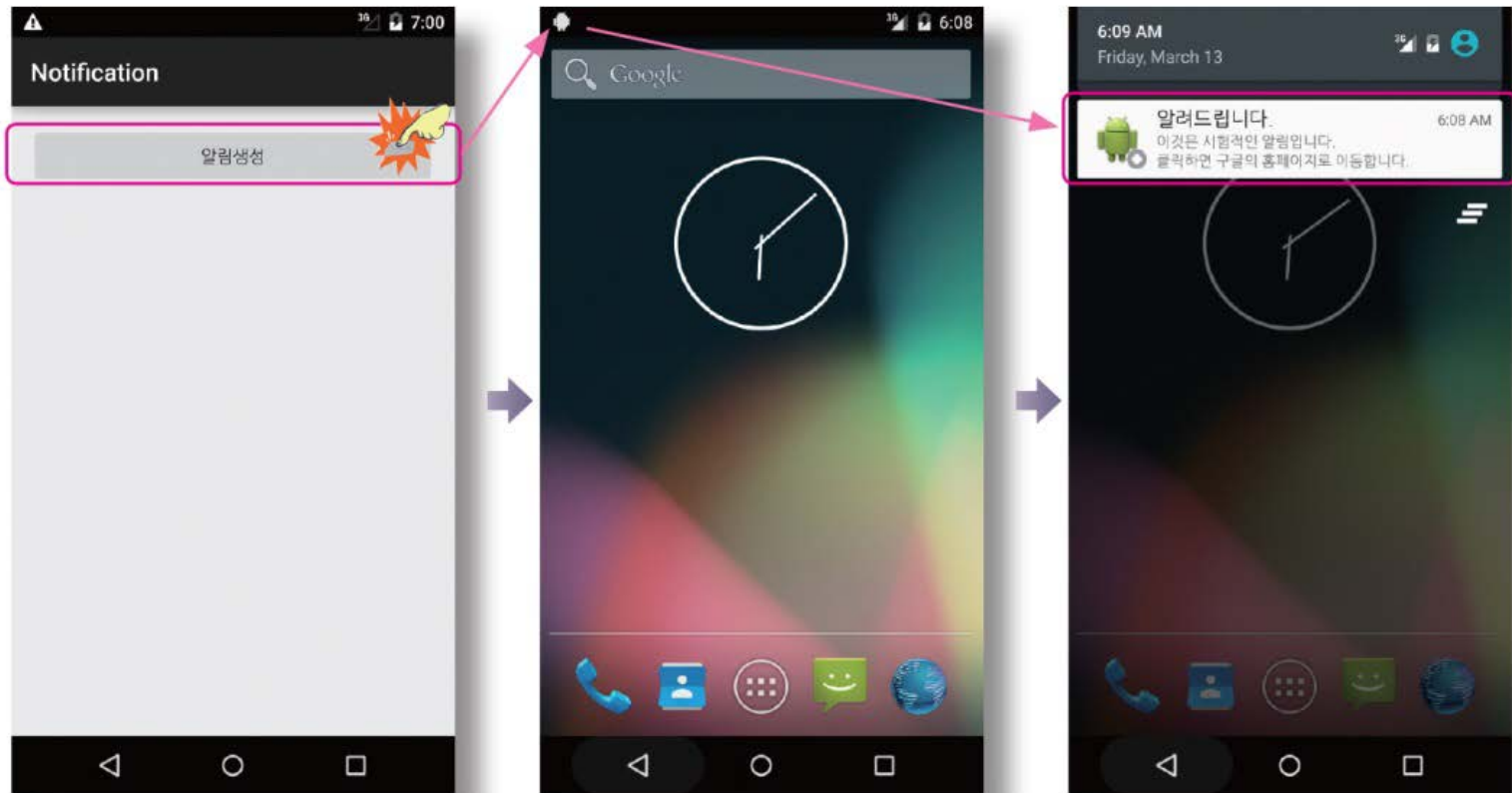
// 알림 콘텐츠
builder.setText("이것은 시험적인 알림입니다.");

// 4.2 이상인 경우에 보여지는 서브 텍스트
builder.setSubText("클릭하면 구글의 홈페이지로 이동합니다.");

NotificationManager notificationManager = (NotificationManager)
getSystemService(NOTIFICATION_SERVICE);

// 알림바에 알림을 표시한다.
notificationManager.notify(NOTIFICATION_ID, builder.build());
}
}
```

# 실행 결과



# Lab: 예약 앱 작성

- 식당 예약 앱

Reservation

식당 예약 애플리케이션

March 2015

	S	M	T	W	T	F	S
10	1	2	3	4	5	6	7
11	8	9	10	11	12	13	14
12	15	16	17	18	19	20	21
13	22	23	24	25	26	27	28
14	29	30	31	1	2	3	4
15	5	6	7	8	9	10	11

5 00

6 : 01 AM

7 02 PM

예약날짜: \_\_\_\_\_

# 사용자 인터페이스

*activity\_main.xml*

<LinearLayout

```
xmlns:android="http://schemas.android.com/apk/res/android"  
android:layout_width="fill_parent"  
android:layout_height="fill_parent"  
android:orientation="vertical" >
```

<TextView

```
android:id="@+id/textView1"  
android:layout_width="wrap_content"  
android:layout_height="wrap_content"  
android:layout_gravity="center"  
  
android:text="식당 예약 애플리케이션"  
android:textAppearance="?android:attr/textAppearanceLarge" />
```

<CalendarView

```
android:id="@+id/calendarView1"  
android:layout_width="match_parent"  
android:layout_height="200dp" />
```

```
<LinearLayout
```

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

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

```
<TextView
```

```
    android:id="@+id/textView2"
```

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

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

```
    android:text="예약날짜:"
```

```
    android:textAppearance="?android:attr/textAppearanceLarge" />
```

```
<EditText
```

```
    android:id="@+id/editText1"
```

```
    android:layout_width="230dp"
```

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

```
    android:layout_weight="1"
```

```
    android:ems="10" />
```

```
<EditText
```

```
    android:id="@+id/editText2"
```

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

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

```
    android:layout_weight="1"
```

```
    android:ems="10" />
```

```
</LinearLayout>
```

```
</LinearLayout>
```



## MainActivity.java

```
package kr.co.company.reservation;
// 소스만 입력하고 Alt+Enter를 눌러서 import 문장을 자동으로 생성한다.

public class MainActivity extends ActionBarActivity {

    private TimePicker tPicker;
    private CalendarView calView;
    private EditText text;
    private EditText text1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        tPicker = (TimePicker) findViewById(R.id.timePicker1);
        calView = (CalendarView) findViewById(R.id.calendarView1);
        text = (EditText) findViewById(R.id.editText1);
        text1 = (EditText) findViewById(R.id.editText2);

        calView.setOnDateChangeListener(new OnDateChangeListener() {
            @Override
            public void onSelectedDayChange(CalendarView view, int year,
                int month, int dayOfMonth) {
                // TODO Auto-generated method stub

                text.setText(" " + year + " : " + month + " : " + dayOfMonth);
            }
        });
        tPicker.setOnTimeChangedListener(new OnTimeChangedListener() {
            public void onTimeChanged(TimePicker view, int hourOfDay, int minute) {
                text1.setText(" " + hourOfDay + " : " + minute);
            }
        });
    }
}
```