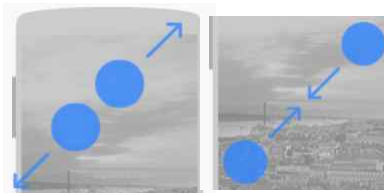
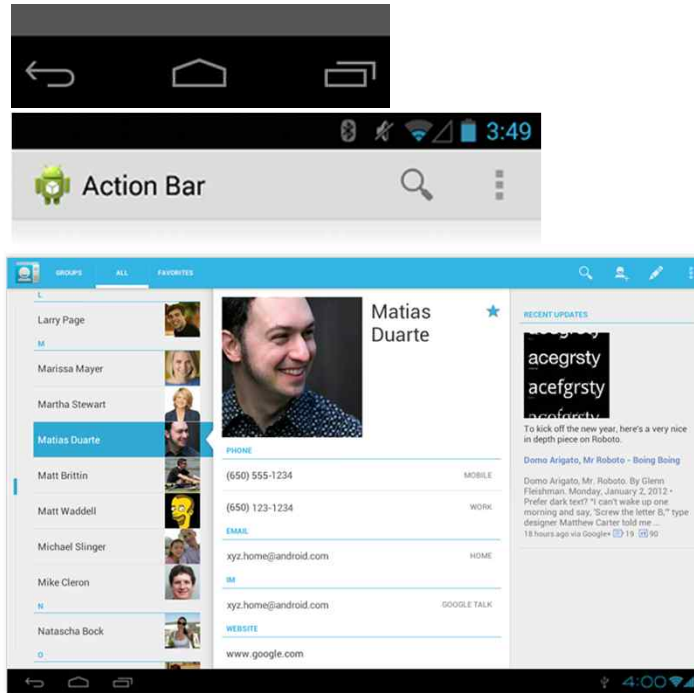


CHAP 5. 메뉴와 대화상자

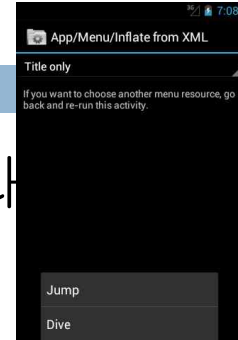
안드로이드의 사용자 인터페이스

- 네비게이션바
- 액션바
- 다중 패널 레이아웃
- 제스처

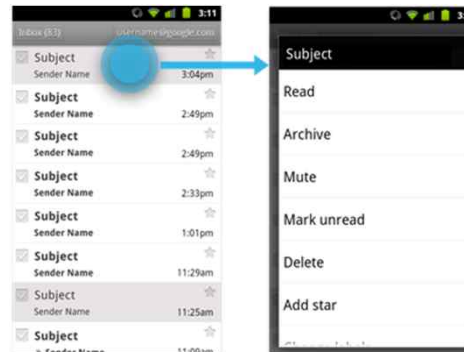
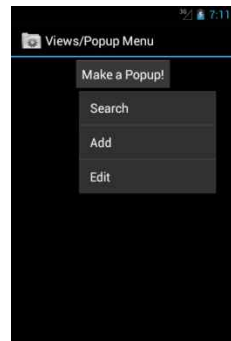


메뉴의 종류

- 옵션 메뉴: 사용자가 MENU 키를 누를 때 나타난다
- 컨텍스트 메뉴: 컨텍스트 메뉴는 사용자가 화면을 일정 시간 이상으로 길게 누르면 나타나는 메뉴이다.



- 팝업메뉴



메뉴를 생성하는 방법

- 코드로 메뉴 생성하기
- **XML**로 메뉴 생성하기

메뉴도 XML로 정의

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.

메뉴 팽창

- 메뉴 리소스를 팽창(**inflate**)하면 실제 메뉴가 생성

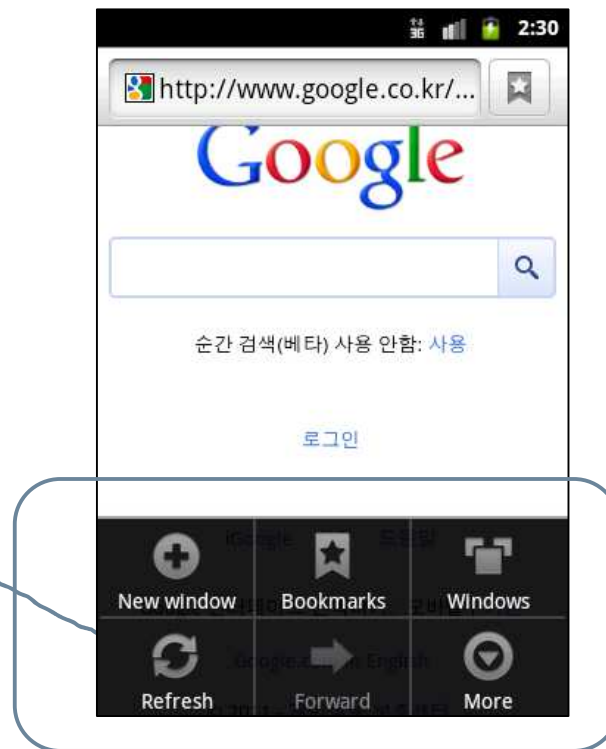


메뉴 리소스 팽창

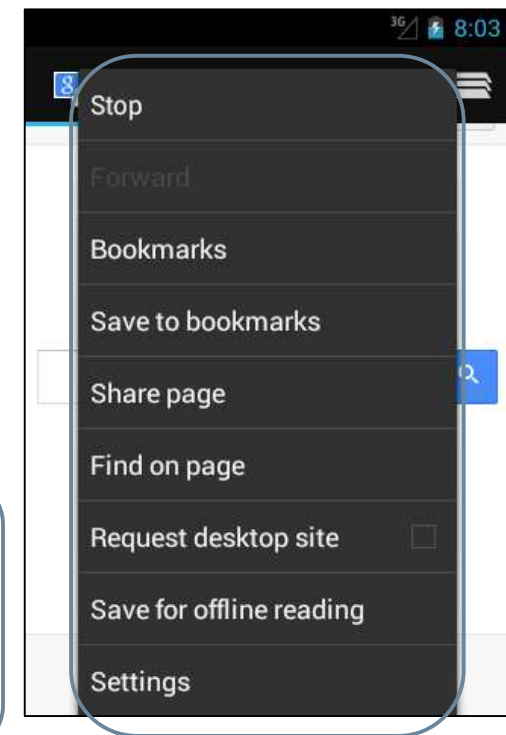
```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.mymenu, menu);
    return true;
}
```

옵션 메뉴

- 현재 액티비티와 관련된 동작



2.3이하



2.3이상

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 항목이 클릭되면 실행

"도움말" 메뉴 항목이 클릭되면 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_background"
    android:title="사과"/>
```

```
  <item
    android:id="@+id/grape"
    android:icon="@drawable/ic_launcher_backgrou"
    android:title="포도"/>
```

```
  <item
    android:id="@+id/banana"
    android:icon="@drawable/ic_launcher_background"
    android:title="바나나"/>
```

```
</menu>
```



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 app bar.

옵션 메뉴 생성

- 사용자가 옵션 키를 누르면 다음의 메소드가 호출된다.

```
package kr.co.company.optionmenu2;  
// 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);  
    }
```

```
    @Override
```

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

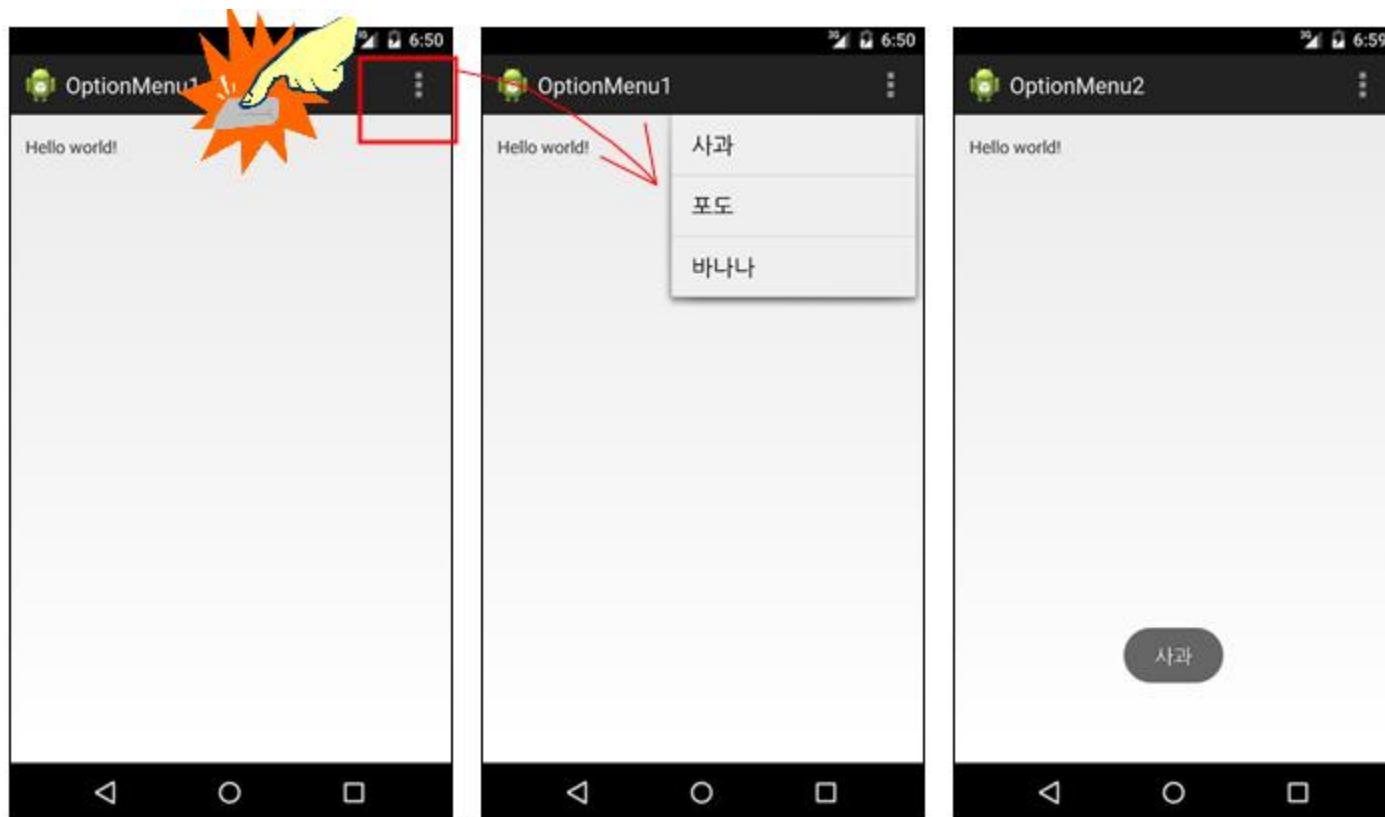
메뉴 mymenu.xml을 팅칭하여
메뉴 객체로 만든다.

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);
    }
}
```

각 메뉴항목이 눌러지면 토스트 메시지를 표시한다.

실행 결과



코드로 옵션 메뉴 생성하기

MainActivity.java

...

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {  
    super.onCreateOptionsMenu(menu);
```

```
    MenuItem item1 = menu.add(0, 1, 0, "사과");  
    item1.setIcon(R.drawable.ic_launcher);  
    item1.setAlphabeticShortcut('a');
```

메뉴 항목의 아이디는
1

```
    menu.add(0, 2, 0, "포도").setIcon(R.drawable.ic_launcher);  
    menu.add(0, 3, 0, "바나나").setIcon(R.drawable.ic_launcher);
```

반환되는 값을 이용하
여서 이렇게 하여도
된다.

```
    return true;
```

```
}
```

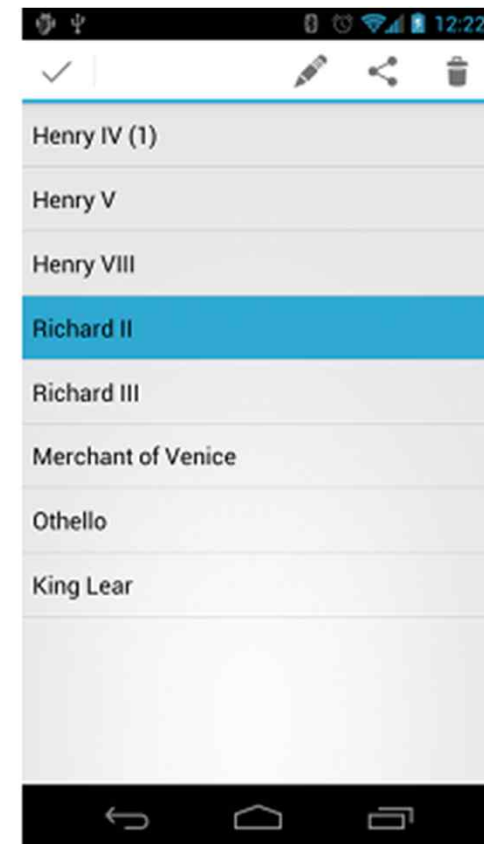
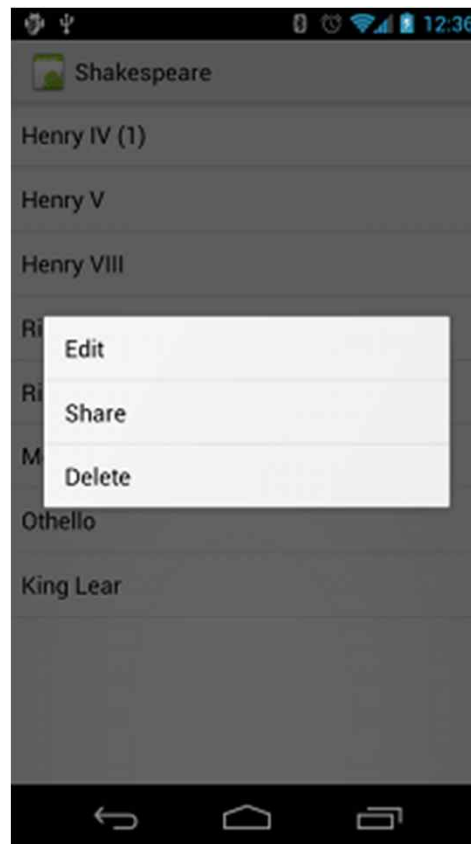
...

TIP

- Android 3.0에는 `android:onClick` 특성을 사용하여 XML에 있는 메뉴 항목에 대한 온-클릭 동작을 정의하는 기능이 추가되었습니다. 특성 값은 메뉴를 사용하여 액티비티가 정의한 메서드의 이름이어야 합니다. 메서드는 공개여야 하며 하나의 **MenuItem** 매개변수를 수락해야 합니다 —시스템이 이 메서드를 호출하면 메서드가 선택한 메뉴 항목을 전달합니다. 자세한 정보와 예시는 메뉴 리소스 문서를 참조하세요.

컨텍스트 메뉴

- UI의 어떤 항목과 관련된 동작



컨텍스트 메뉴 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="200px"
        android:typeface="serif" />
</LinearLayout>
```

컨텍스트 메뉴 예제

```
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

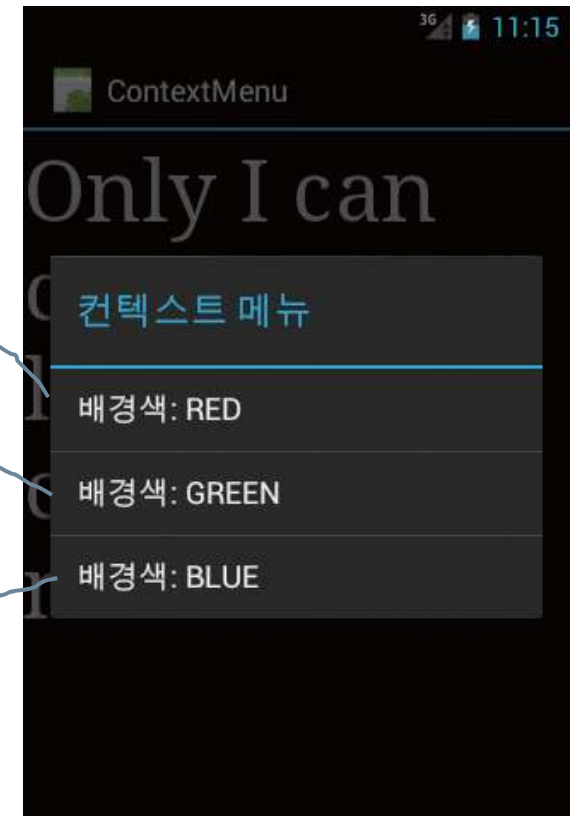
배경색: RED

배경색: GREEN

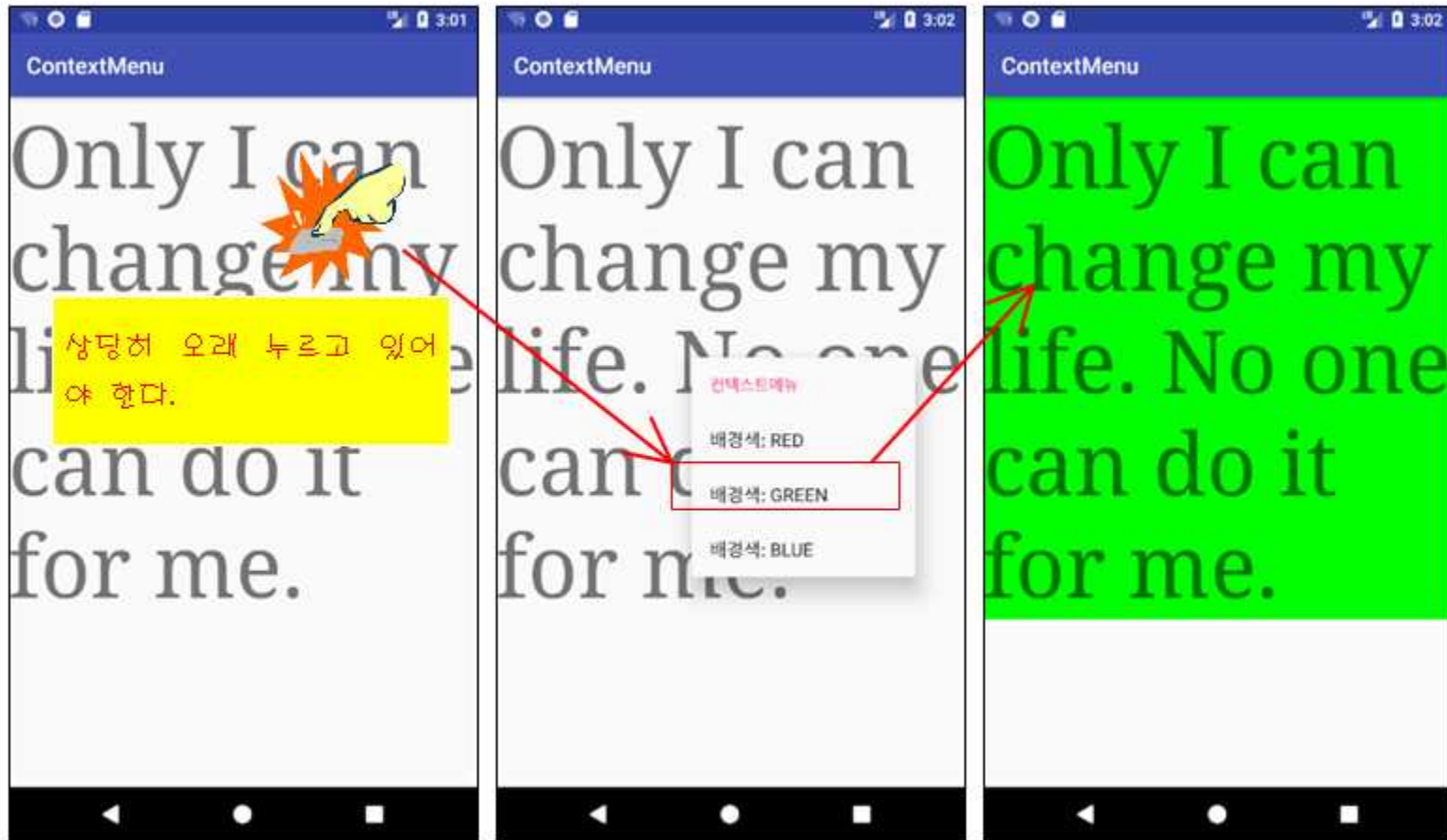
배경색: BLUE

컨텍스트 메뉴 이벤트 처리

```
public boolean onOptionsItemSelected(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.onOptionsItemSelected(item);  
    }  
}
```

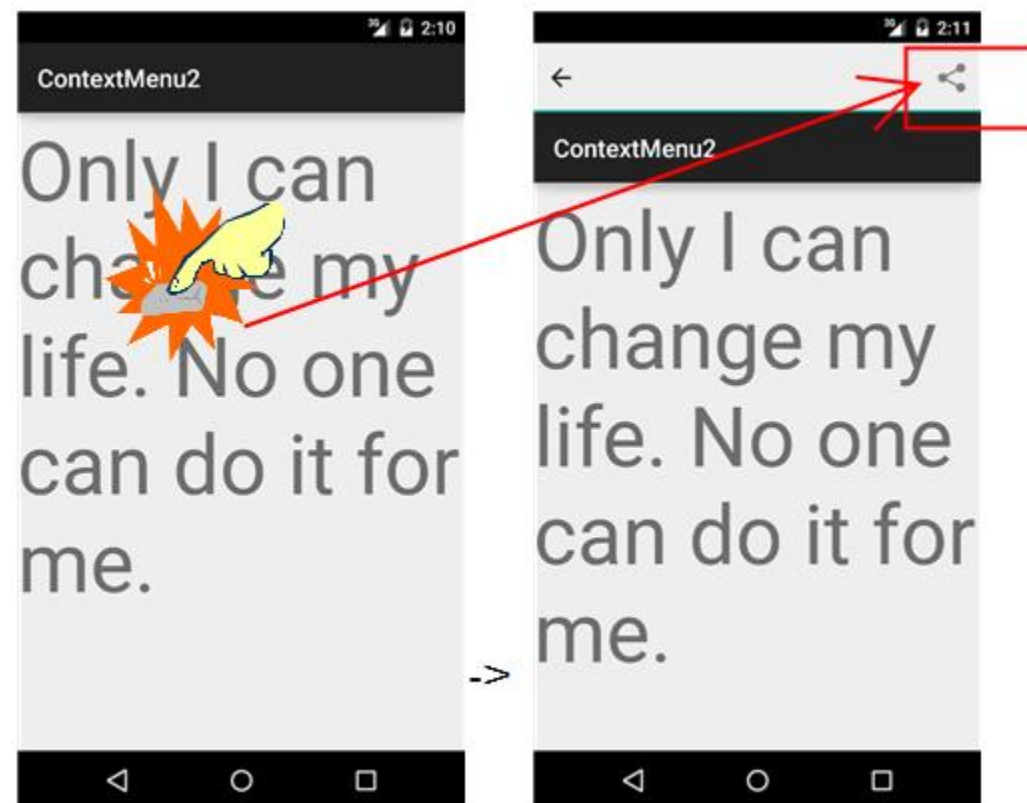


실행 결과



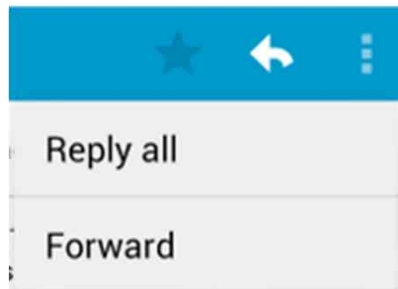
컨텍스트 액션 모드

- 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);
    }
```

팝업 메뉴 항목이 눌러지면 여기서 처리한다.

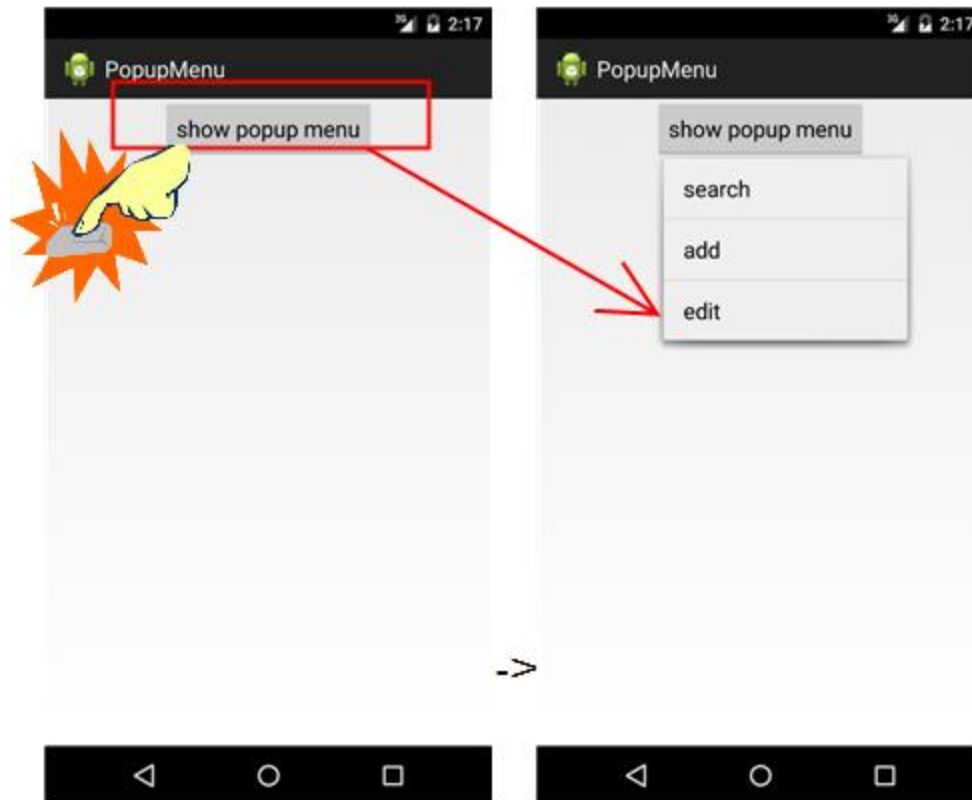
```
    public void onClick(View button) {
        PopupMenu popup = new PopupMenu(this, button);
        popup.getMenuInflater().inflate(R.menu.popup, popup.getMenu());
```

```
        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();
```

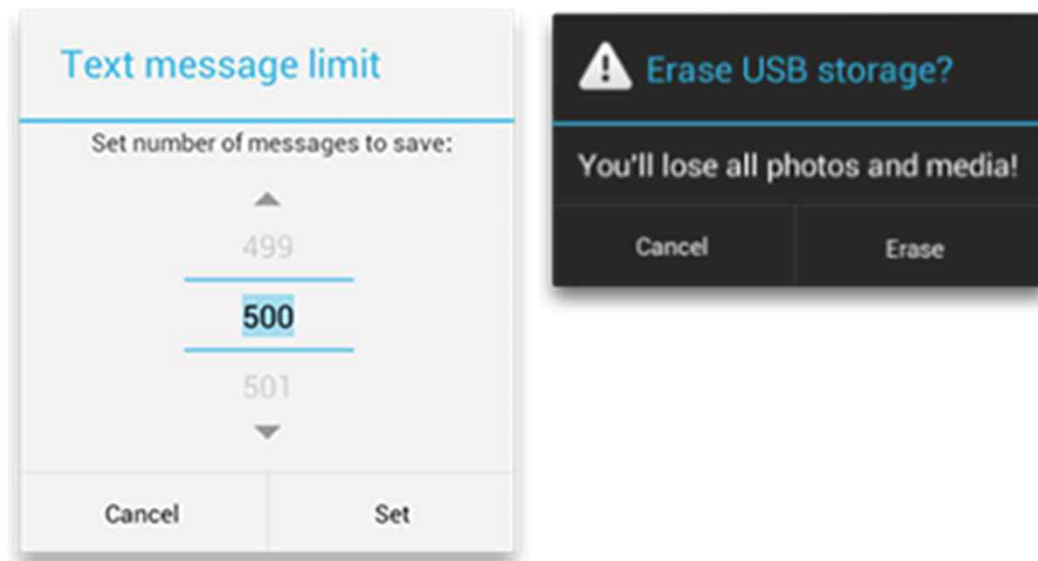
```
    }
}
```

실행 결과



대화 상자

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

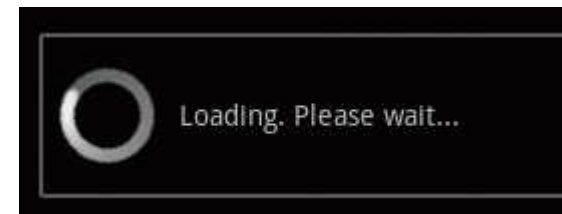


대화 상자의 종류

□ AlertDialog



□ ProgressDialog



□ DatePickerDialog

□ TimePickerDialog



AlertDialog



AlertDialog 예제

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    >
    <Button
        android:text="결제하려면 버튼을 누르세요."
        android:id="@+id/Button01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
    </Button>
</LinearLayout>
```



```

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();
        }
    });
```

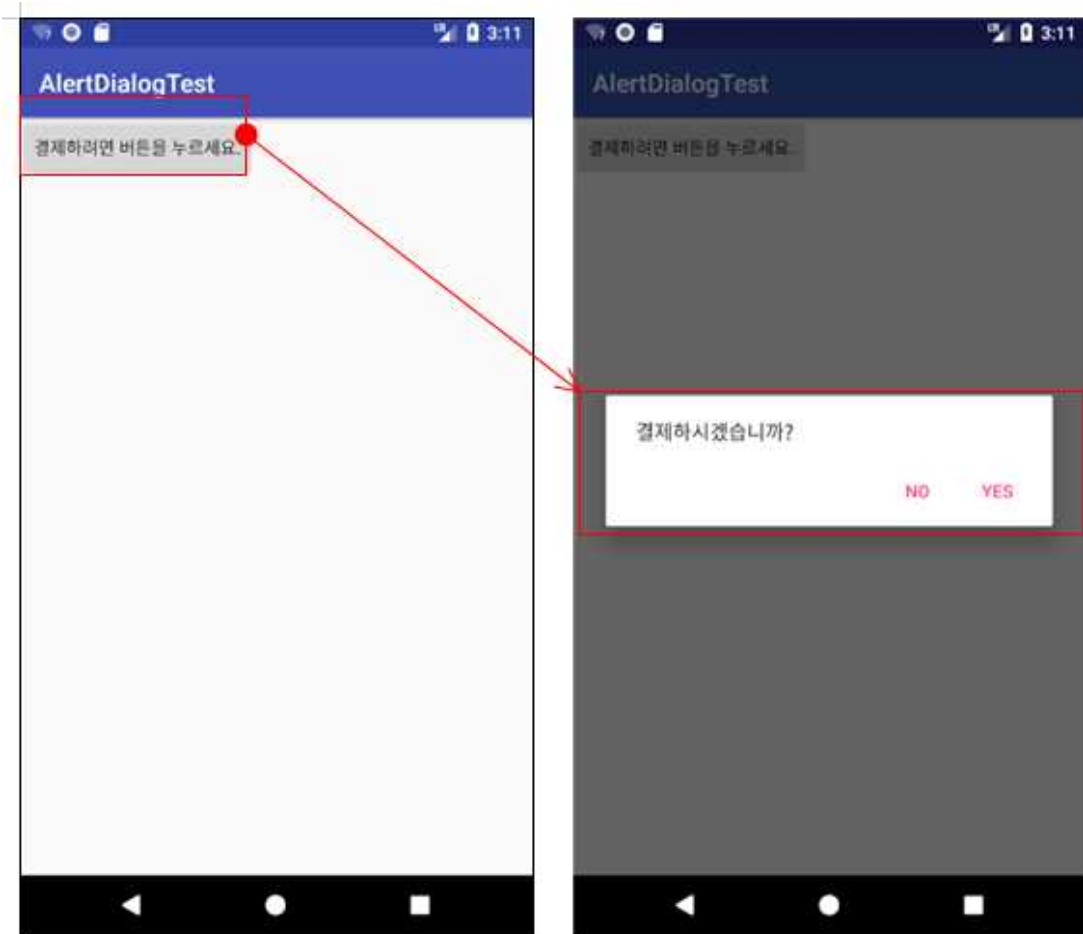
대화 상자의 "no" 버튼 부분
설정

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

```
}
```

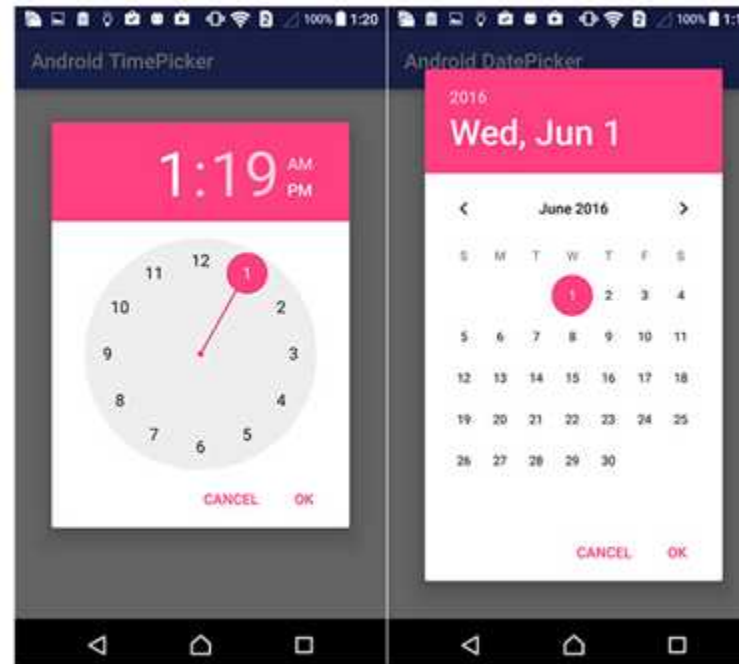
```
}
```

실행 결과

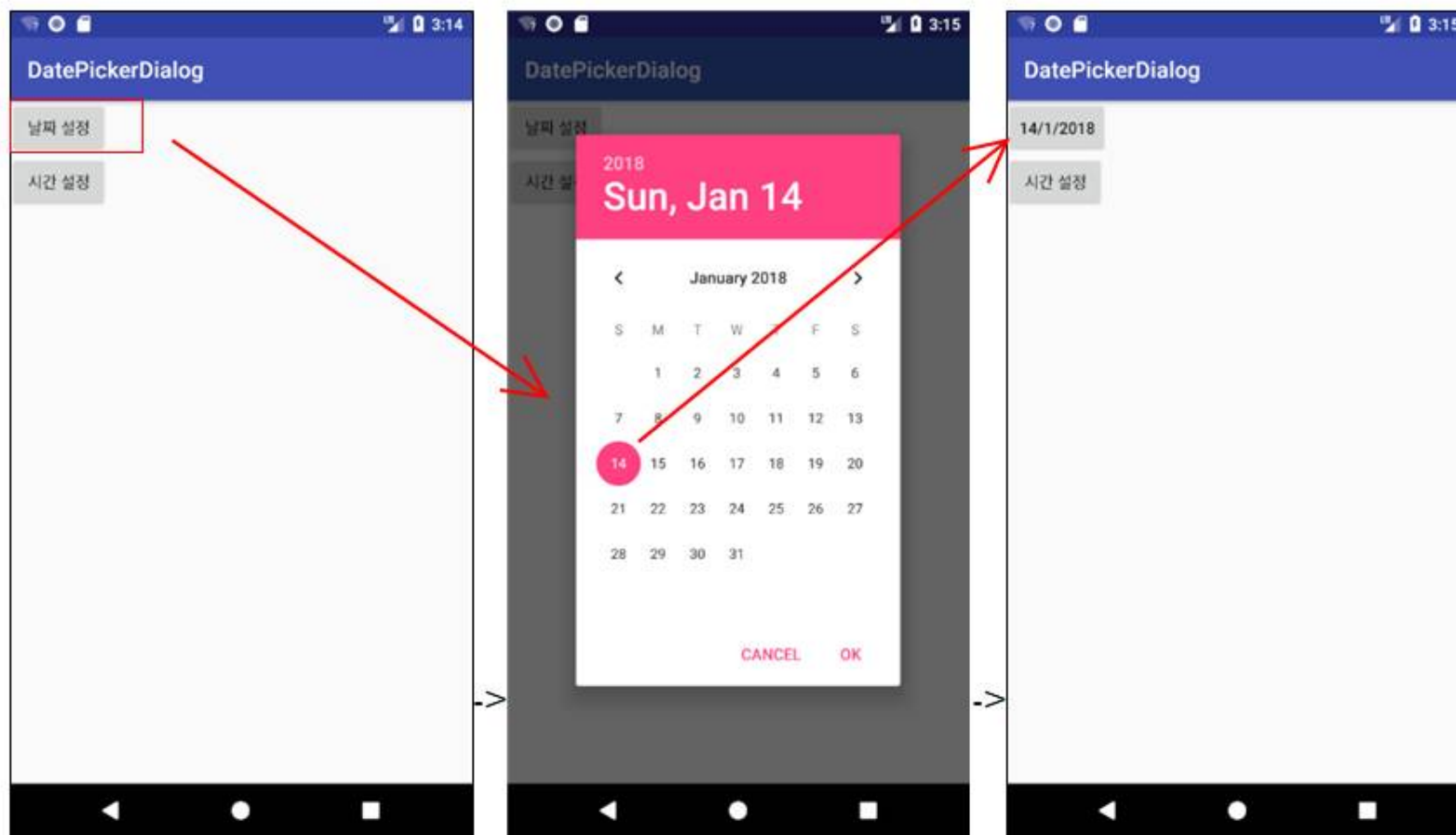


DatePickerDialog

- 날짜와 시간을 입력받는 대화 상자



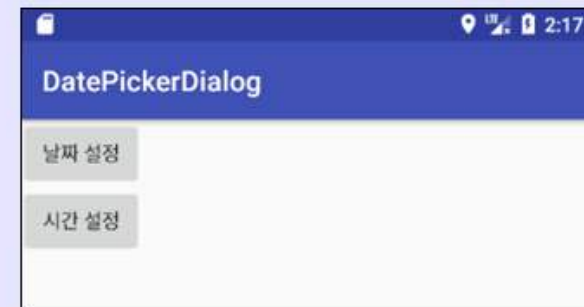
예제



예제

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>
```




```
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.

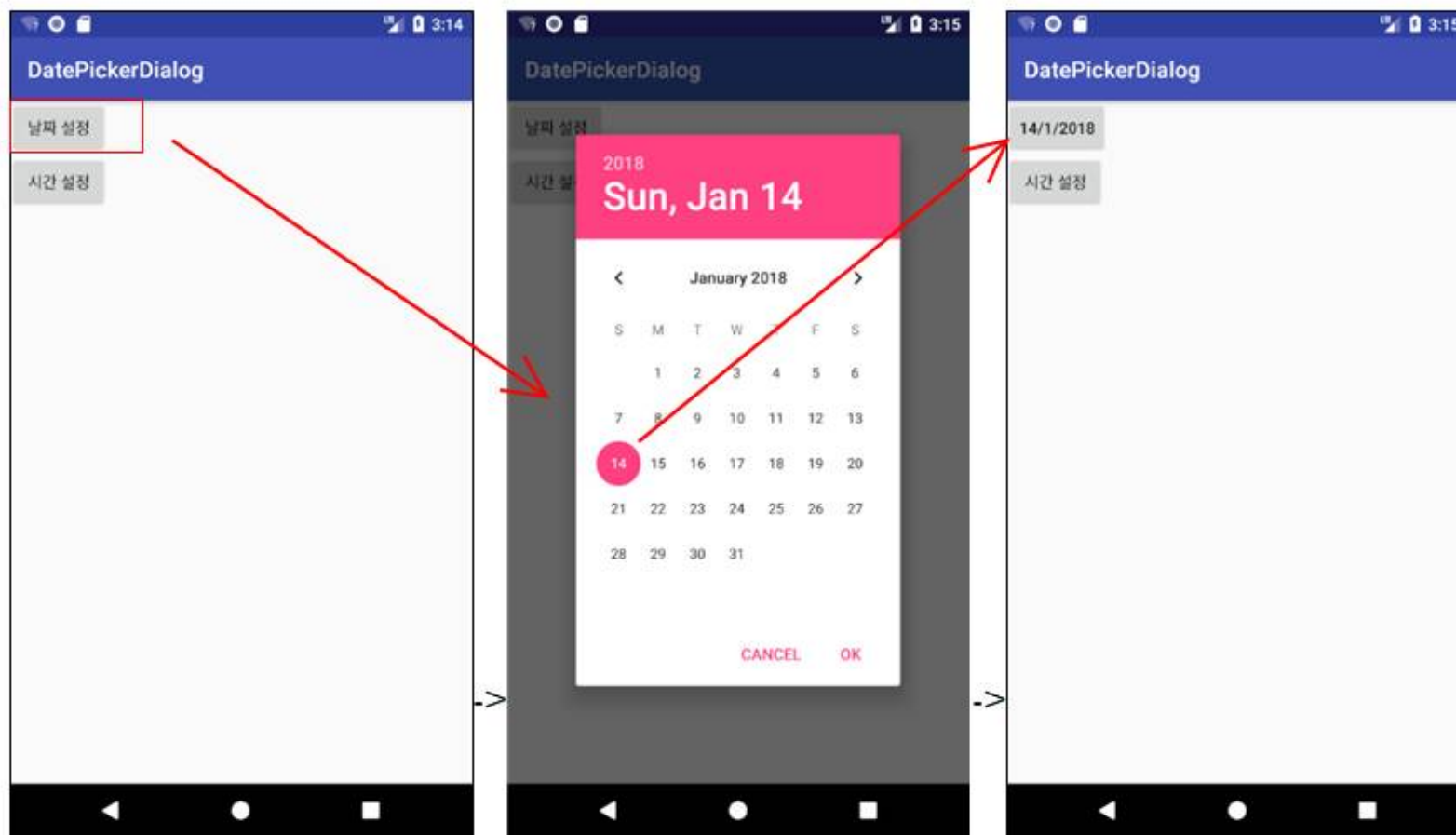
```

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();
}
}
}
}

```

실행 결과



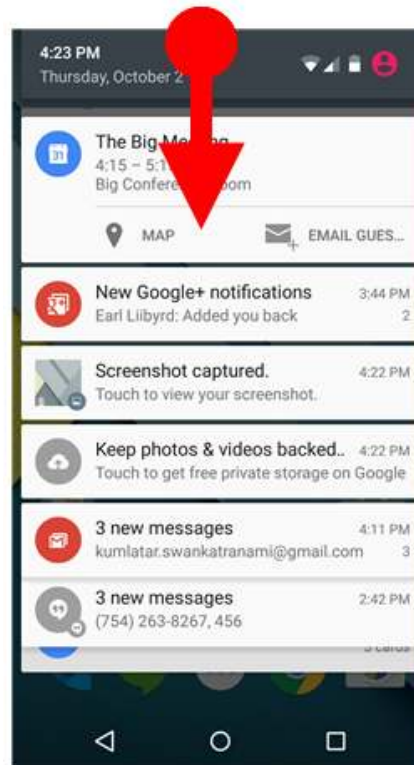
커스텀 대화 상자

- 사용자가 마음대로 대화 상자의 내용을 디자인할 수 있는 대화 상자



알림 기능

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



알림 만드는 절차

1. 알림 채널 생성하기
2. 알림 빌더를 생성한다.
3. 알림 속성을 설정한다.
4. 액션을 첨부한다.(선택 사항)
5. 알림 객체 생성하여 보내기

알림 채널 생성하기

```
String NOTIFICATION_CHANNEL_ID = "my_channel_id_01";

private void createNotificationChannel() {

    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
        NotificationChannel notificationChannel = new
        NotificationChannel(NOTIFICATION_CHANNEL_ID, "My Notifications",
        NotificationManager.IMPORTANCE_DEFAULT);
        notificationChannel.setDescription("Channel description");
        NotificationManager notificationManager = (NotificationManager)
        getSystemService(Context.NOTIFICATION_SERVICE);
        notificationManager.createNotificationChannel(notificationChannel);
    }
}
```

알리 빌더를 생성한다.

- NotificationCompat.Builder builder = new NotificationCompat.Builder(this);

알림 속성을 설정한다.

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

액션을 첨부한다.

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

알림 객체 생성하여 보내기

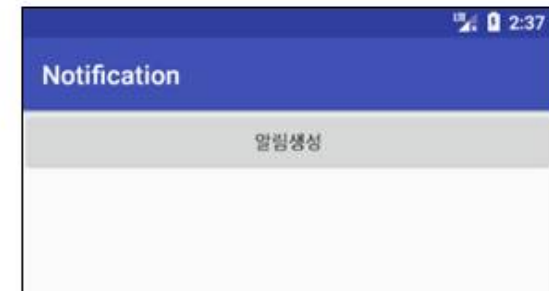
- NotificationManager notificationManager = (NotificationManager) getSystemService(NOTIFICATION_SERVICE);
- notificationManager.notify(NOTIFICATION_ID, builder.build());

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>
```



자바 소스

MainActivity.java

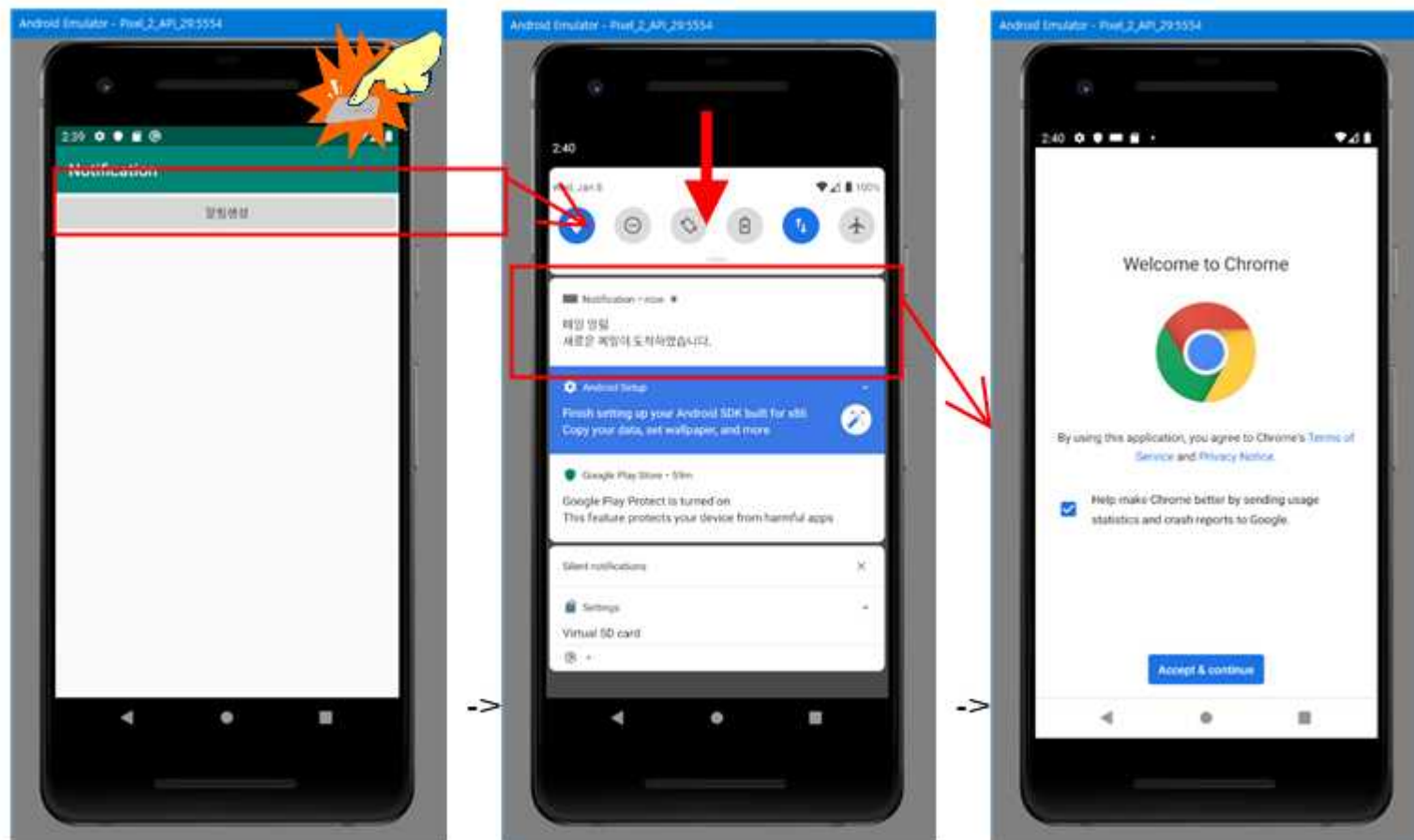
```
public class MainActivity extends AppCompatActivity {
    String NOTIFICATION_CHANNEL_ID = "my_channel_id_01";

    private void createNotificationChannel() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            NotificationChannel notificationChannel = new
            NotificationChannel(NOTIFICATION_CHANNEL_ID, "My Notifications",
            NotificationManager.IMPORTANCE_DEFAULT);
            notificationChannel.setDescription("Channel description");
            NotificationManager notificationManager = (NotificationManager)
            getSystemService(Context.NOTIFICATION_SERVICE);
            notificationManager.createNotificationChannel(notificationChannel);
        }
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        createNotificationChannel();
    }
}
```

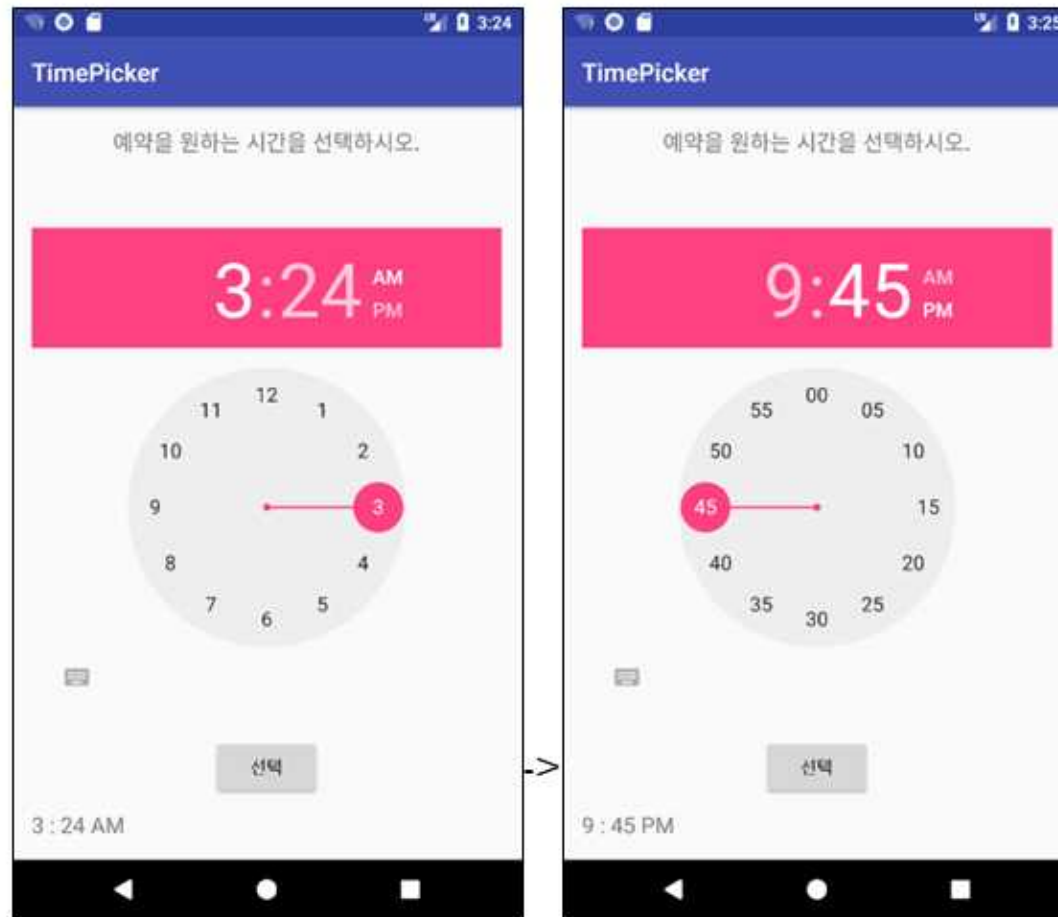
자바 소스



```
public void sendNotification(View view) {  
    NotificationCompat.Builder notificationBuilder = new  
    NotificationCompat.Builder(this, NOTIFICATION_CHANNEL_ID);  
  
    // 알람이 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);  
  
    notificationBuilder.setSmallIcon(R.drawable.android)  
        .setContentTitle("메일 알람")  
        .setContentText("새로운 메일이 도착하였습니다.")  
        .setContentIntent(pendingIntent);  
  
    NotificationManager notificationManager = (NotificationManager)  
    getSystemService(Context.NOTIFICATION_SERVICE);  
    notificationManager.notify(1, notificationBuilder.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));  
}  
}
```