



출처: [maxpixel.net](http://maxpixel.net)

## CHAP 7. 메뉴, 대화 상자, 알림



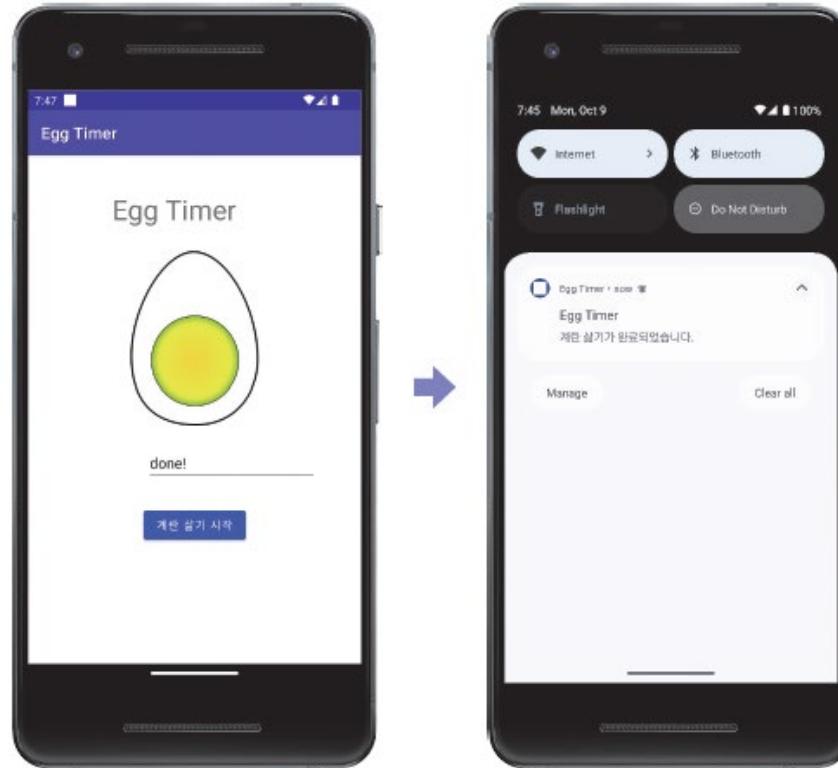
# 메뉴, 대화상자, 알림





# 7장의 목표

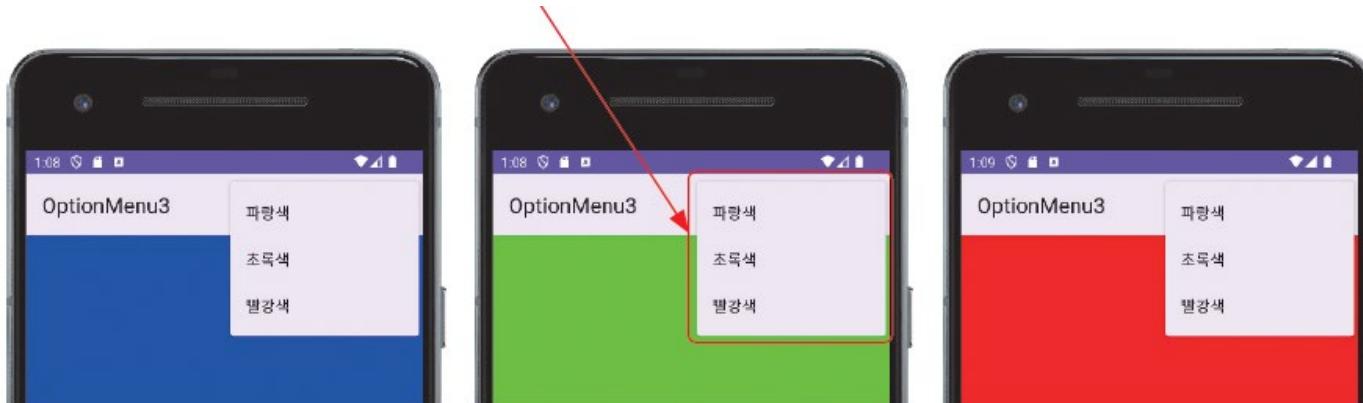
- 사용자가 다양한 계란 스타일에 대해 서로 다른 요리 시간 설정을 지정할 수 있는 작동하는 계란 타이머 앱을 만든다. 타이머는 선택한 시간 간격부터 카운트 다운하고 계란이 준비되면 알림을 보낸다.





# 메뉴의 종류

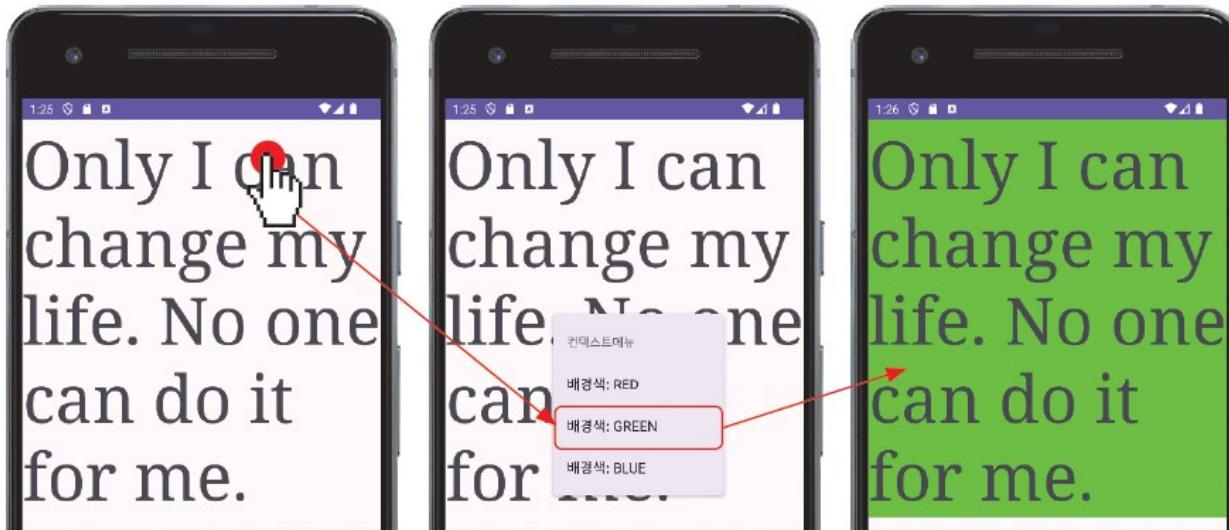
- 옵션 메뉴: 액션바에 표시되는 메뉴





# 메뉴의 종류

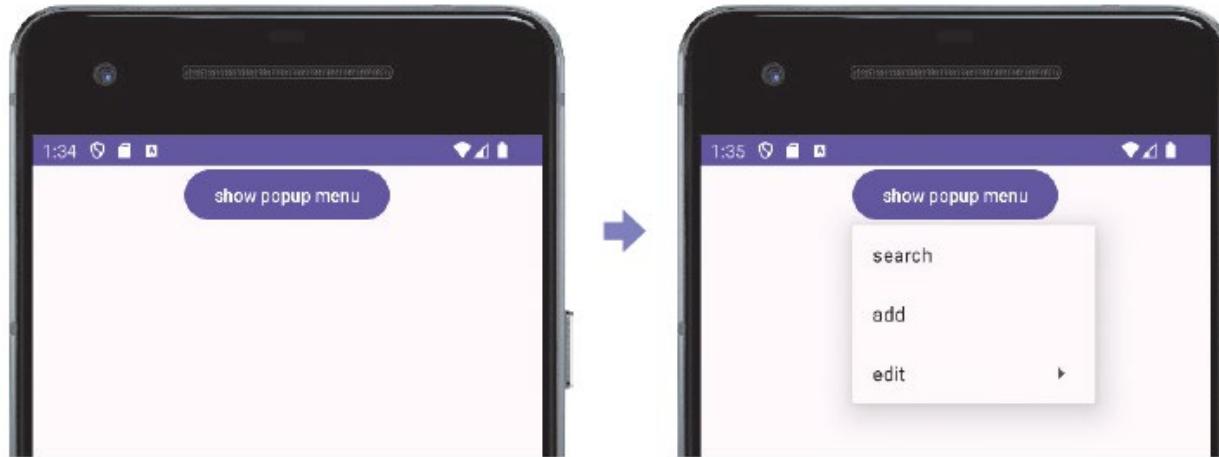
- 컨텍스트 메뉴: 컨텍스트 메뉴는 사용자가 화면을 일정 시간 이상으로 길게 누르면 나타나는 메뉴이다.





# 메뉴의 종류

- 팝업 메뉴: 액션바에 표시되는 메뉴





# 옵션 메뉴

- 옵션 메뉴를 생성하는 절차는 다음과 같다.

메뉴 생성

- 메뉴 XML 파일 생성



메뉴 파일 등록

- onCreateOptionMenu()  
메소드 오버라이딩



메뉴 항목 이벤트 처리

- onOptionItemSelected()  
메소드 오버라이딩



# 메뉴도 XML로 정의

mymenu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/blue"
        android:icon="@android:drawable/btn_star_big_on"
        android:title="파랑색"
        app:showAsAction="always" />
    <item
        android:id="@+id/green"
        android:icon="@android:drawable/ic_btn_speak_now"
        android:title="초록색"
        app:showAsAction="ifRoom" />
</menu>
```

〈menu〉 요소는 Menu를 생성하고 이것은 메뉴 항목들을 담는 컨테이너가 된다. 반드시 루트 노드이어야 하며, 〈item〉이나 〈group〉을 하나 이상 포함한다.

〈item〉 요소는 MenuItem을 생성한다. 이것은 하나의 메뉴 항목을 나타낸다. 서브 메뉴를 작성하려면 〈menu〉를 포함할 수 있다.



# 메뉴 XML 파일

- <menu>: 메뉴를 나타내며 메뉴 항목을 저장하는 컨테이너이다. Menu 객체를 생성한다.
- <item>: 하나의 메뉴 항목을 나타낸다. MenuItem 객체를 생성한다. <menu>를 내부에 가질 수 있어서 서브 메뉴를 생성할 수 있다.

```
<item
```

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

메뉴 항목을 나타내는 리소스 식별자로서 사용자가 메뉴 항목을 선택하였을 경우에 애플리케이션이 식별하기 위하여 필요하다.

```
    android:icon="@android:drawable/btn_star_big_on"
```

사용자에게 보이는 아이콘 이미지이다.

```
    android:title="파랑색"
```

사용자에게 보이는 메뉴 항목의 타이틀로서 문자열 리소스로 정의된다.

```
    app:showAsAction="always" />
```

액션바에 어떻게, 언제 메뉴 항목을 표시할 것인지를 지정한다.



# 액션바에 메뉴 표시 옵션

- `showAsAction`은 액션바에 메뉴 항목을 어떻게 나타내느냐를 지정한다.
  - `app:showAsAction="always"`: 메뉴 항목을 항상 보이게 표시한다.
  - `app:showAsAction="never"`: 메뉴 항목을 오버플로우 영역에 숨긴다.
  - `app:showAsAction="ifRoom"`: 액션바에 공간이 있는 경우에만 표시한다.



# 액티비티에 메뉴 등록하기

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



전체  
구조

메뉴 리소스 평창



# 메뉴 팽창

- 메뉴 리소스를 팽창(**inflate**)하면 실제 메뉴가 생성





# 옵션 메뉴 클릭 이벤트 처리



전체  
구조

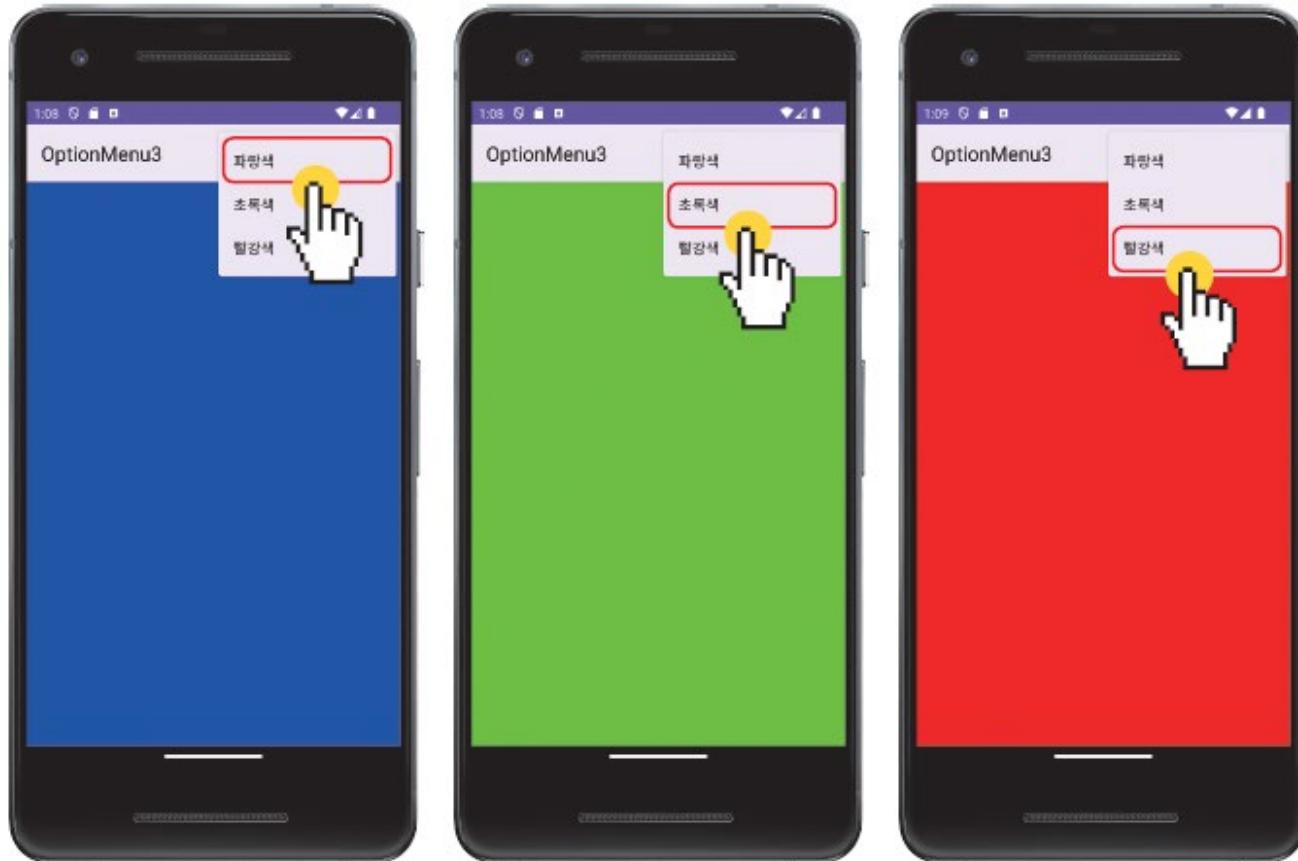
```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    if( id == R.id.blue){
        view1.setBackgroundColor(Color.BLUE);
        return true;
    } else if( id == R.id.green){
        view1.setBackgroundColor(Color.GREEN);
        return true;
    }
    return super.onOptionsItemSelected(item);
}
```

"파랑색" 메뉴 항목이 클릭되면 실행

"초록색" 메뉴 항목이 클릭되면 실행



# 예제: 옵션 메뉴 생성





# 예제: 옵션 메뉴 생성

activity\_main.xml

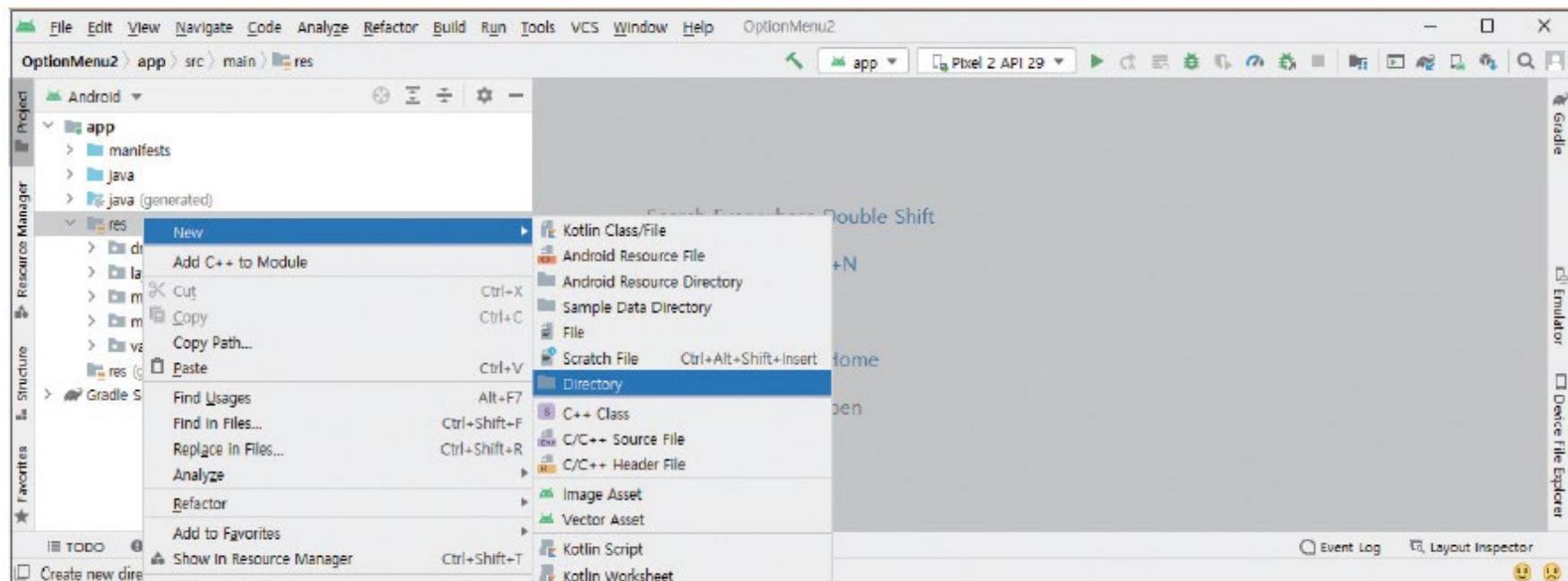
```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout >

    <LinearLayout
        android:id="@+id/layout"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal"></LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>
```

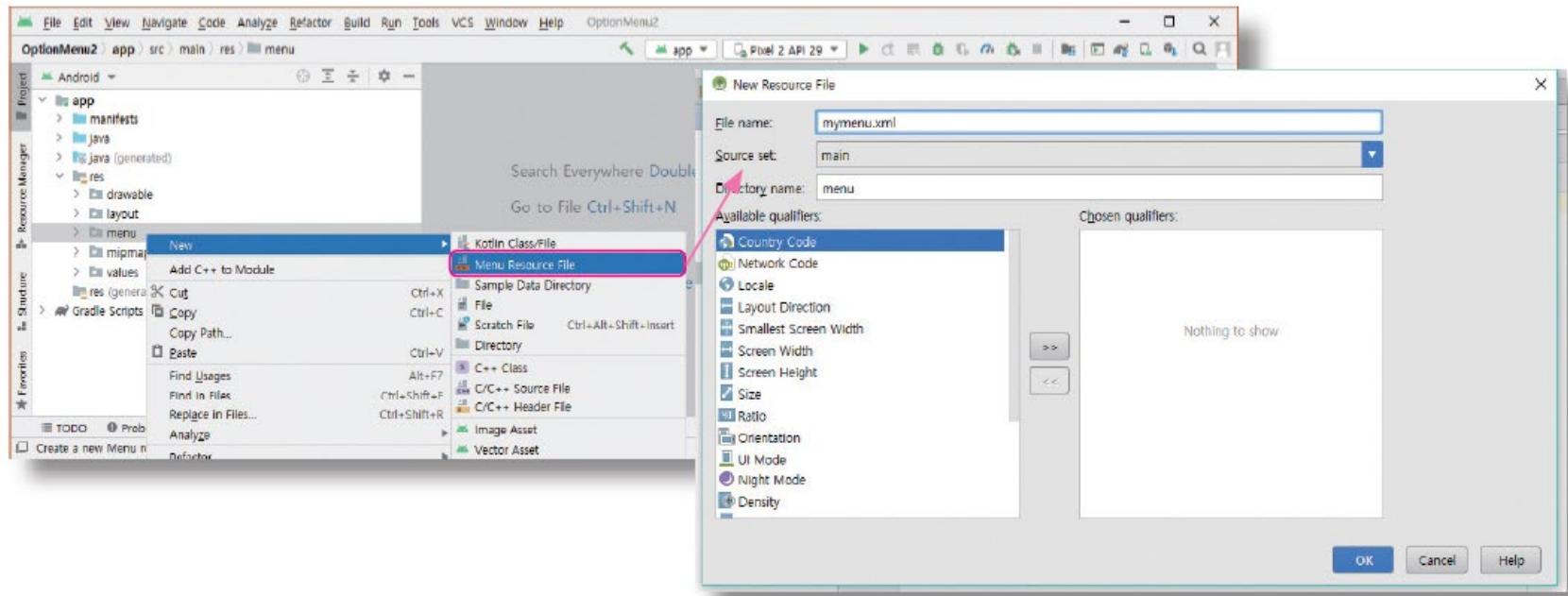


# 예제: 옵션 메뉴 생성





# 예제: 옵션 메뉴 생성





# 예제: 옵션 메뉴 생성

mymenu.xml

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

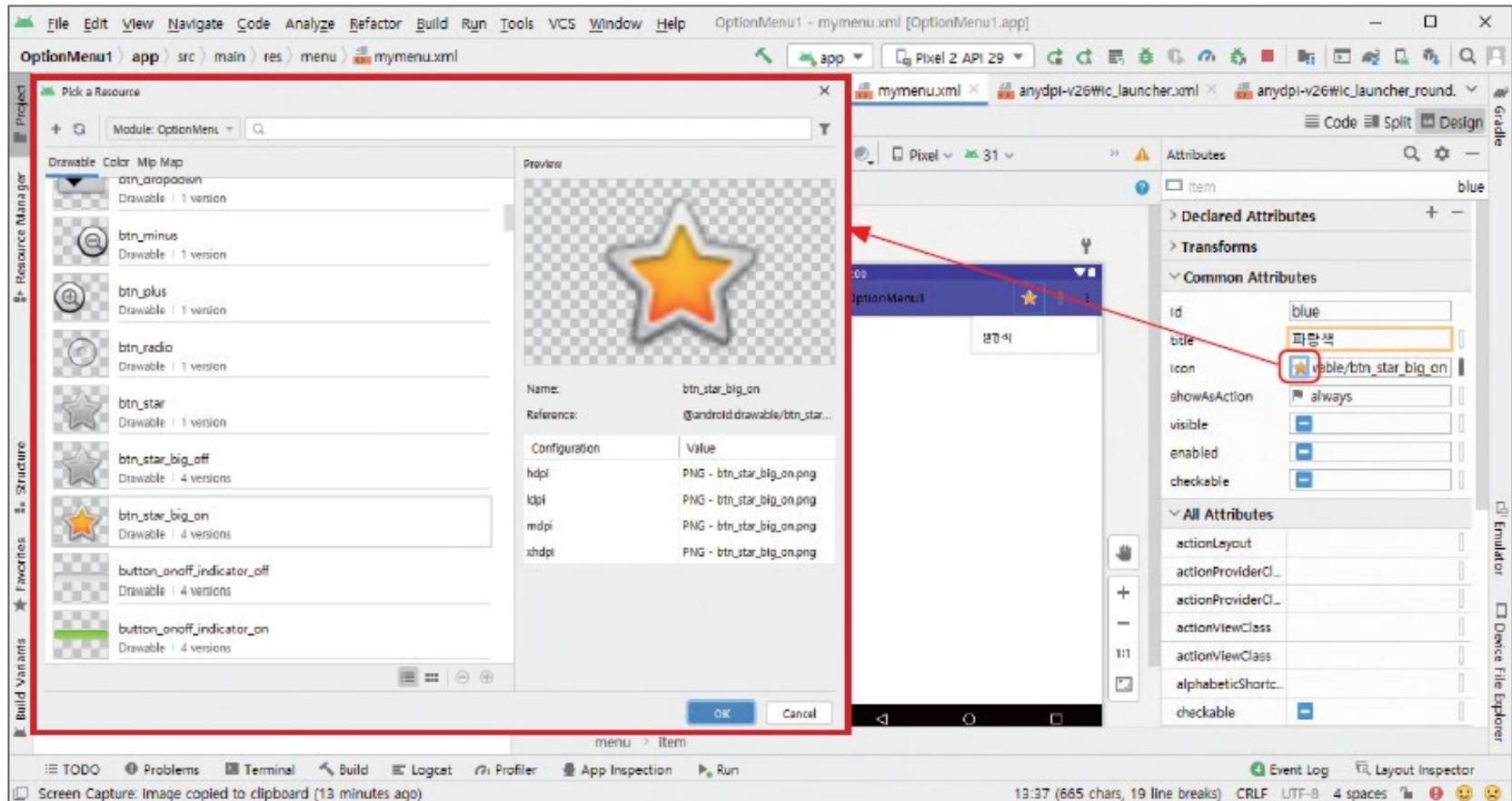
    <item
        android:id="@+id/blue"
        android:icon="@android:drawable/btn_star_big_on"
        android:title="파랑색"
        app:showAsAction="never" />

    <item
        android:id="@+id/green"
        android:icon="@android:drawable/ic_btn_speak_now"
        android:title="초록색"
        app:showAsAction="never" />

    <item
        android:id="@+id/red"
        android:icon="@android:drawable/checkbox_on_background"
        android:title="빨강색"
        app:showAsAction="never" />

</menu>
```

# 예제: 옵션 메뉴 아이콘 설정





# 예제: 옵션 메뉴 생성

MainActivity.java

```
package kr.co.company.optionmenu1;

...
public class MainActivity extends AppCompatActivity {
    View view1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        view1 = findViewById(R.id.layout);
    }
}
```



# 예제: 옵션 메뉴 생성

```
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    getMenuInflater().inflate(R.menu.mymenu, menu);  
    return true;  
}  
  
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    int id = item.getItemId();  
    if( id == R.id.blue ){  
        view1.setBackgroundColor(Color.BLUE);  
        return true;  
    }  
    else if( id == R.id.green ){  
        view1.setBackgroundColor(Color.GREEN);  
        return true;  
    }  
    else if( id == R.id.red ){  
        view1.setBackgroundColor(Color.RED);  
        return true;  
    }  
    return super.onOptionsItemSelected(item);  
}
```





# 매니페스트 파일 수정

- /res/values/themes/themes.xml 파일에서 “Theme.Material3.DayNight.NoActionBar”를 “Theme.Material3.DayNight”로 수정한다.  
“Theme.Material3.DayNight.NoActionBar”로 설정되면 액션바가 나타나지 않는다.

themes.xml

```
<resources xmlns:tools="http://schemas.android.com/tools">  
    <!-- Base application theme. -->  
    <style name="Base.Theme.OptionMenu3" parent="Theme.Material3.DayNight">  
        <!-- Customize your light theme here. -->  
        <!-- <item name="colorPrimary">@color/my_light_primary</item> -->  
    </style>  
  
    <style name="Theme.OptionMenu3" parent="Base.Theme.OptionMenu3" />  
</resources>
```

여기 수정한다.



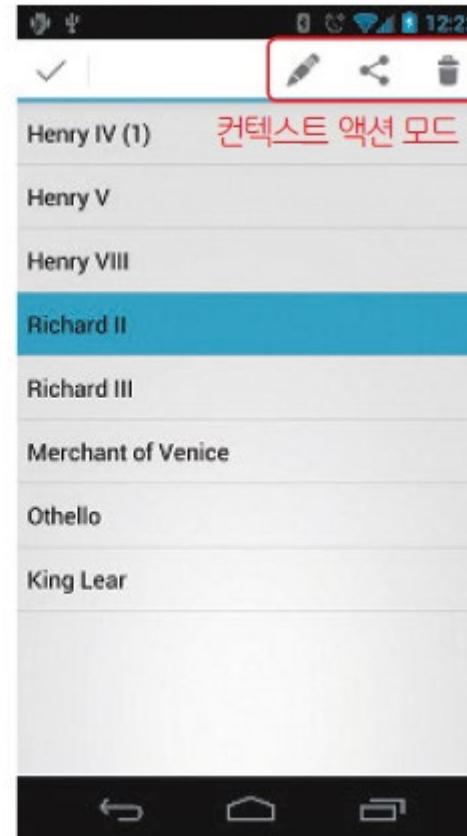
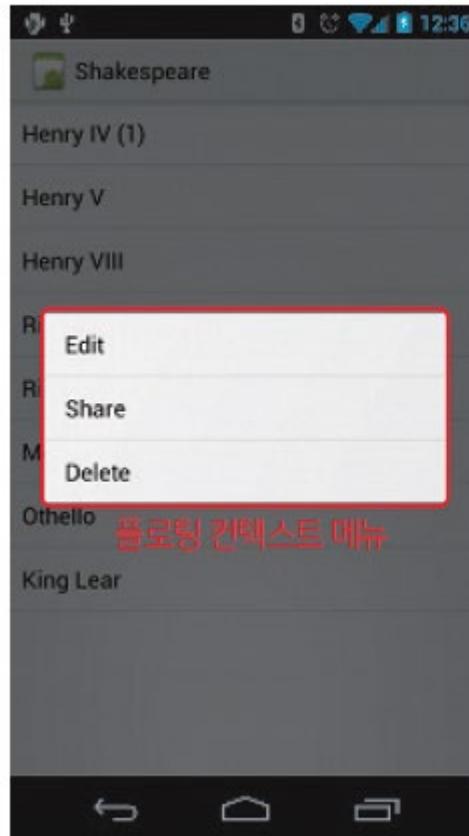
# TIP

- Android 3.0부터 `android:onClick` 특성을 사용하여 XML에 있는 메뉴 항목에 대한 온-클릭 동작을 정의하는 기능이 추가되었음.
- 특성 값은 메뉴를 사용하여 액티비티가 정의한 메서드의 이름이어야 한다.



# 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.



# 플로팅 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 설정

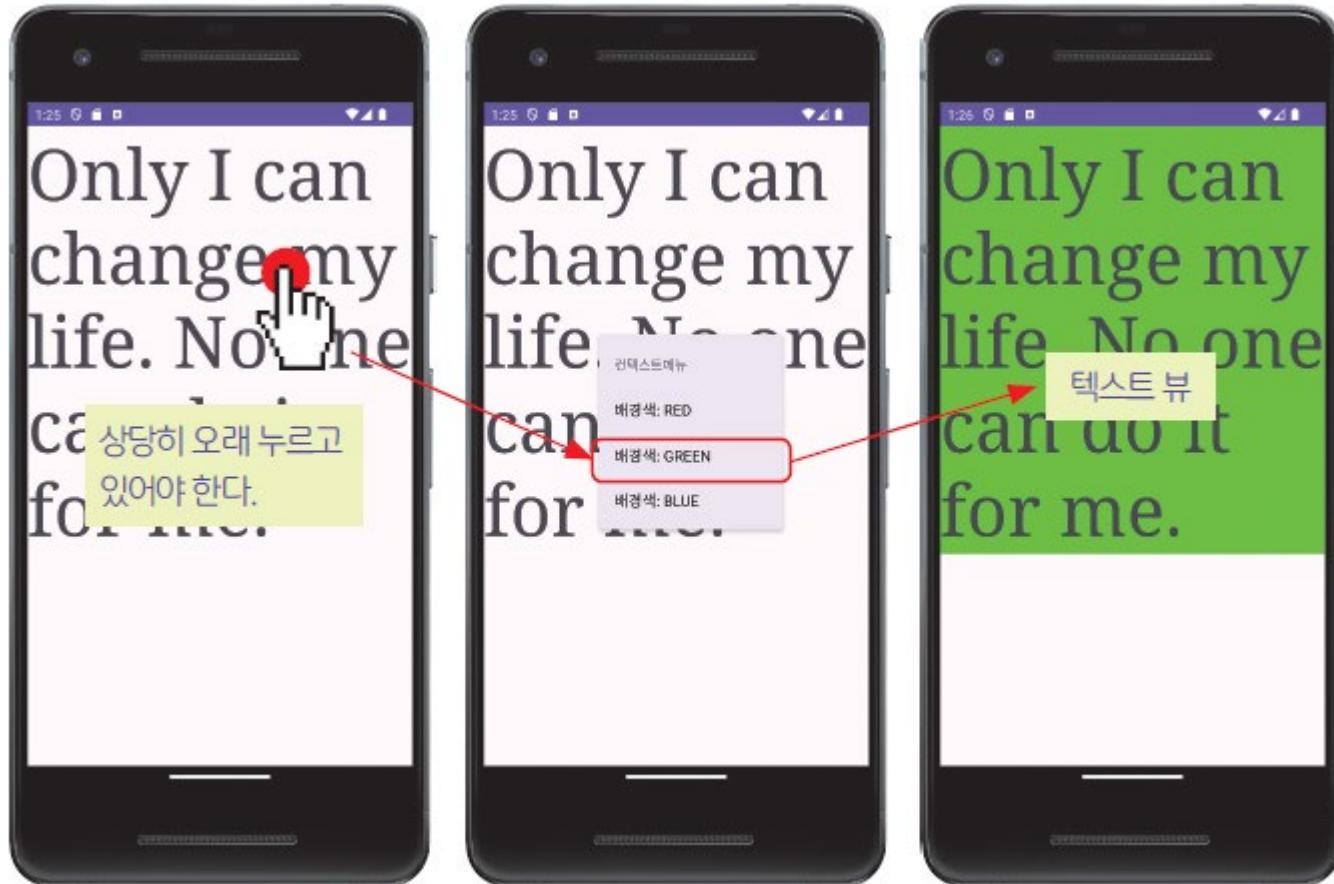


전체  
구조

```
@Override  
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo menuInfo) {  
  
    super.onCreateContextMenu(menu, v, menuInfo); ← 부모 클래스 메소드 호출  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.context_menu, menu); ← 메뉴 리소스를 팽창한다.  
}
```



# 예제: 컨텍스트 메뉴 예제





# 예제: 컨텍스트 메뉴 예제

activity\_main.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/LinearLayout01"
    android:layout_width="match_parent"
    android:layout_height="match_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>
```

← 텍스트 뷰에 컨텍스  
메뉴를 등록한다.



# 예제: 컨텍스트 메뉴 예제

MainActivity.java

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

← 텍스트 뷰에 컨텍스트 메뉴를 등록한다.



# 예제: 컨텍스트 메뉴 예제

```
@Override  
public void onCreateContextMenu(ContextMenu menu, View v,  
        ContextMenuInfo menuInfo) {  
    super.onCreateContextMenu(menu, v, menuInfo);  
    menu.setHeaderTitle("컨텍스트메뉴");  
    menu.add(0, 1, 0, "배경색: RED"); ← add(int groupId, int  
    menu.add(0, 2, 0, "배경색: GREEN");      itemId, int order,  
    menu.add(0, 3, 0, "배경색: BLUE");       CharSequence title)  
}  
}
```

코드로 메뉴를 생성할 수도 있다.

```
@Override  
public boolean onContextItemSelected(MenuItem item) {  
    int id = item.getItemId();
```



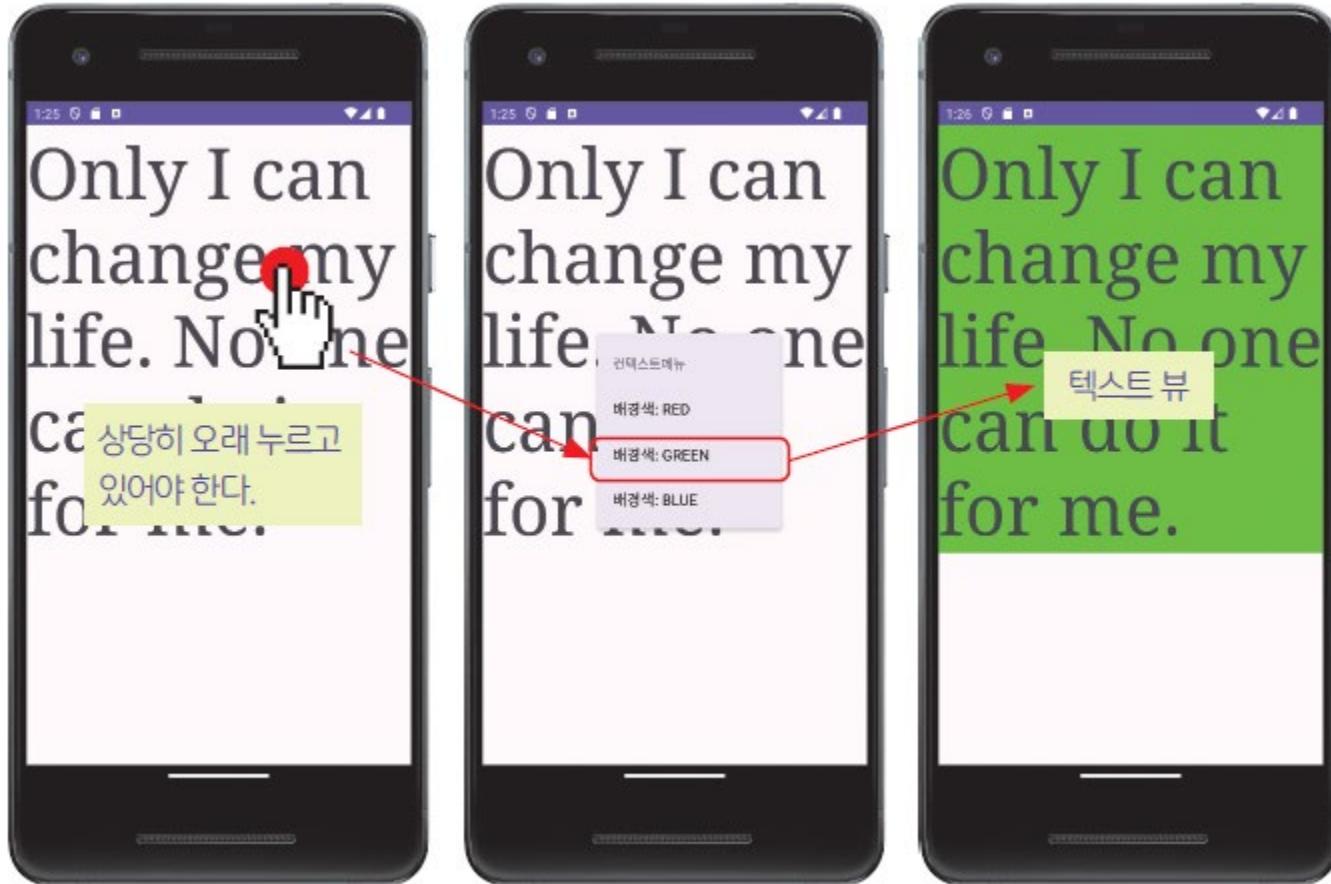
# 예제: 컨텍스트 메뉴 예제

```
if (id == 1) {  
    text.setBackgroundColor(Color.RED);  
    return true;  
} else if (id == 2) {  
    text.setBackgroundColor(Color.GREEN);  
    return true;  
} else if (id == 3) {  
    text.setBackgroundColor(Color.BLUE);  
    return true;  
}  
return super.onContextItemSelected(item);  
}
```

컨텍스트 메뉴가 클릭되면 텍스트 뷰의 배경색이 변경된다.



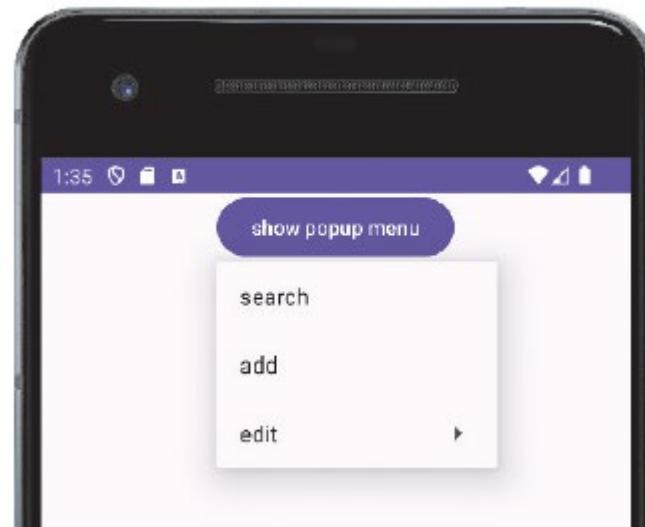
# 예제: 실행 결과





# 팝업 메뉴

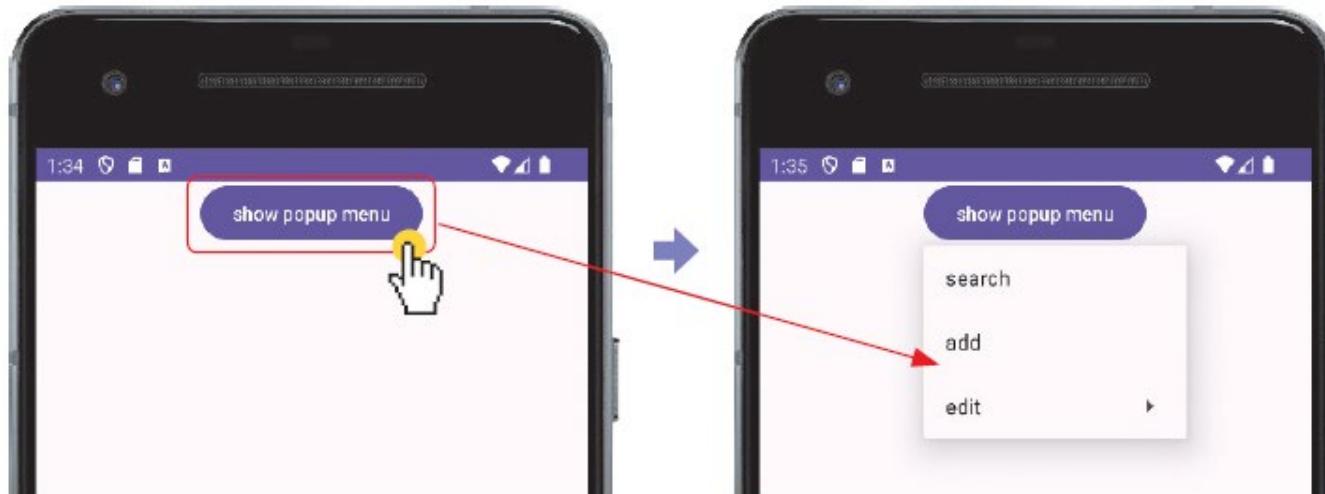
- 뷰에 부착된 모달 메뉴(modal menu)
- API 레벨 11부터 제공
- 팝업 메뉴의 용도
  - 오버플로우 스타일 메뉴 제공
  - 서브 메뉴의 역할
  - 드롭다운 메뉴





# 예제: 팝업 메뉴

- 간단한 팝업 메뉴를 가지고 있는 애플리케이션을 작성한다.





# 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" <-- 버튼을 누르면 onClick()이 호출된다.  
        android:text="show popup menu" />  
  
</LinearLayout>
```



# 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 서  
브 메뉴이다.



# 예제: 팝업 메뉴

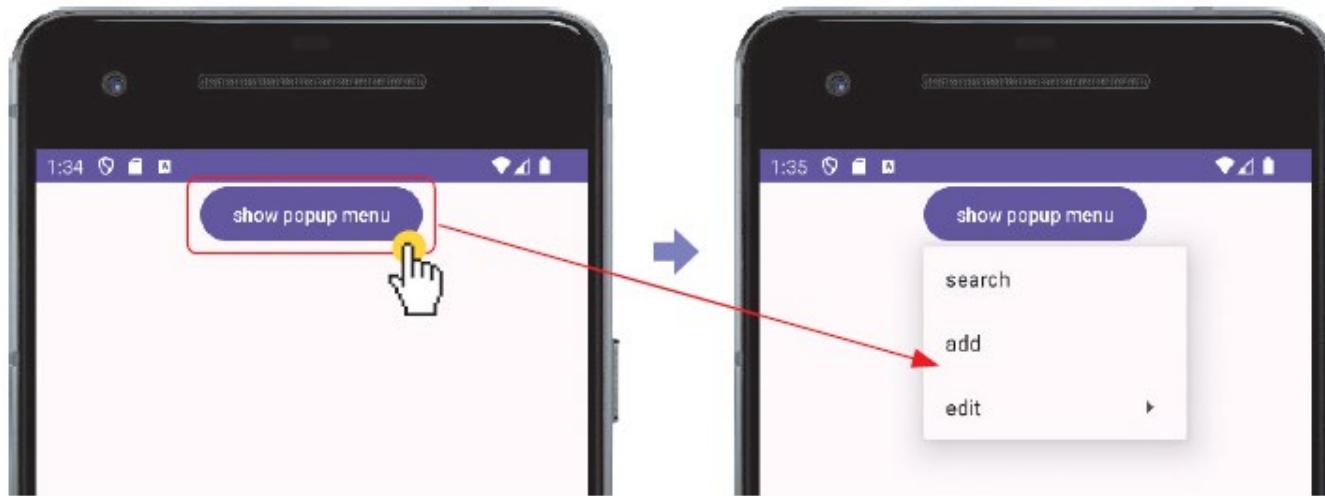
MainActivity.java

```
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.



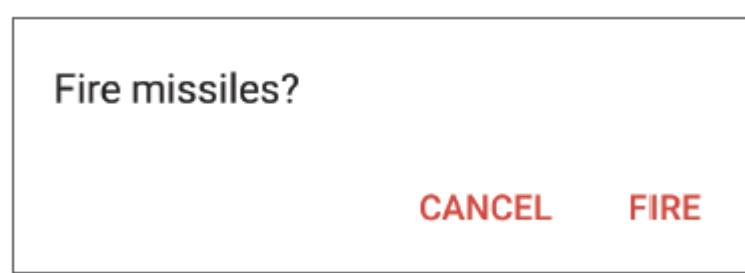
# 실행 결과





# 대화 상자

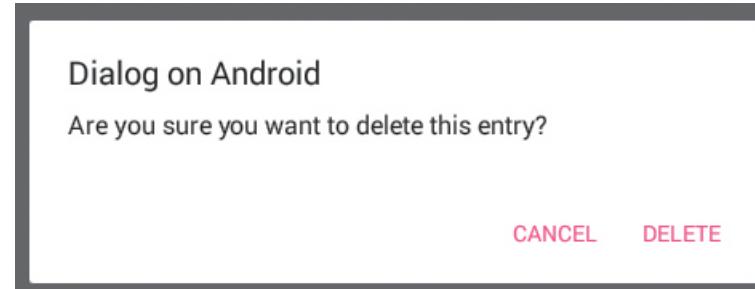
- 대화 상자(dialog)는 사용자에게 메시지를 출력하고 사용자로부터 입력을 받아들이는 아주 보편적인 사용자 인터페이스



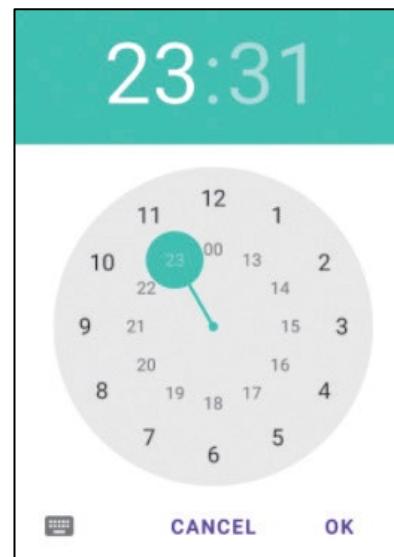


# 대화 상자의 조각

- AlertDialog

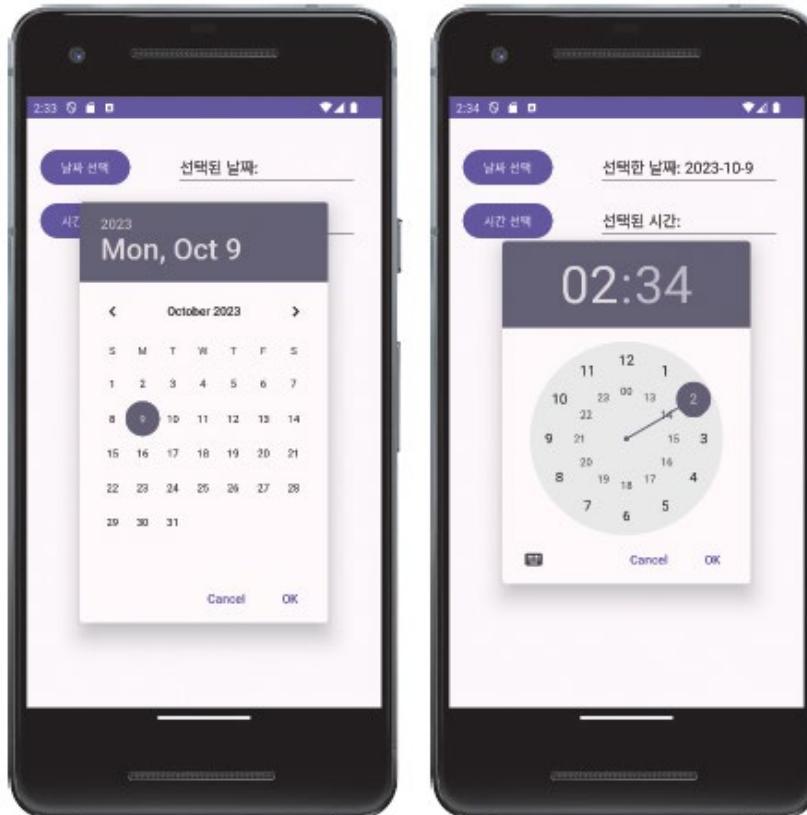


- DatePickerDialog
  - TimePickerDialog



# 날짜 / 시간 선택 대화 상자

- 안드로이드 앱에서 날짜와 시간을 선택하는 대화 상자를 구현하려면 **DatePickerDialog**(날짜 선택)와 **TimePickerDialog**(시간 선택)를 사용 할 수 있다.





# 날짜 / 시간 선택 대화 상자

(1) 다음 코드를 사용하여 Calendar 객체를 초기화한다.

```
Calendar calendar = Calendar.getInstance();
int year = calendar.get(Calendar.YEAR);
int month = calendar.get(Calendar.MONTH);
int dayOfMonth = calendar.get(Calendar.DAY_OF_MONTH);
```



# 날짜 / 시간 선택 대화 상자

(2) DatePickerDialog 객체를 생성하고 초기 날짜를 설정한다.

```
DatePickerDialog datePickerDialog = new DatePickerDialog(  
    MainActivity.this,  
    new DatePickerDialog.OnDateSetListener() {  
        @Override  
        public void onDateSet(DatePicker view, int selectedYear, int selectedMonth, int  
        selectedDayOfMonth) {  
            // 사용자가 선택한 날짜 처리  
            String selectedDate = selectedYear + "—" + (selectedMonth + 1) + "—" +  
            selectedDayOfMonth;  
            // 선택한 날짜를 사용하거나 표시할 곳에 적용한다.  
        }  
    },  
    year, month, dayOfMonth  
);
```



# 날짜 / 시간 선택 대화 상자

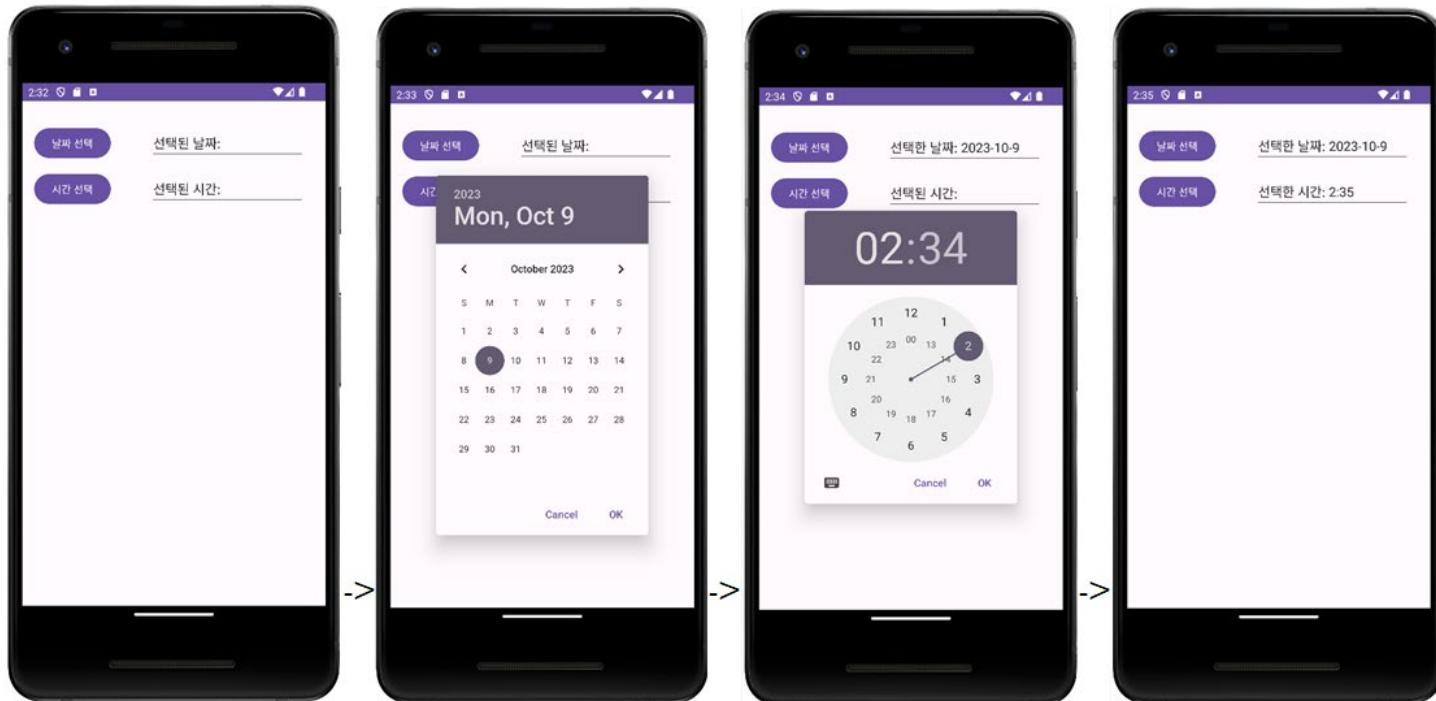
(3) DatePickerDialog를 표시한다. show( ) 메소드를 호출하여 대화 상자를 보여준다.

```
datePickerDialog.show();
```



# 예제: 날짜와 시간 선택

- 사용자가 버튼을 누르면 **DatePickerDialog**(날짜 선택 대화 상자)와 **TimePickerDialog**(시간 선택 대화 상자)를 표시하고 선택된 날짜와 시간을 화면에 표시하는 간단한 앱을 만들어보자.





# 예제: 날짜와 시간 선택

activity\_main.xml

```
<androidx.constraintlayout.widget.ConstraintLayout >  
    <Button  
        android:id="@+id/select_date_button"  
        android:text="날짜 선택" />  
  
    <EditText  
        android:id="@+id/selected_date_text"  
        android:text="선택된 날짜: " />  
  
    <Button  
        android:id="@+id/select_time_button"  
        android:text="시간 선택" />  
  
    <EditText  
        android:id="@+id/selected_time_text"  
        android:text="선택된 시간: " />  
</androidx.constraintlayout.widget.ConstraintLayout>
```



# 예제: 날짜와 시간 선택

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    private TextView selectedDateText, selectedTimeText;

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

        selectedDateText = findViewById(R.id.selected_date_text);
        selectedTimeText = findViewById(R.id.selected_time_text);

        Button selectDateButton = findViewById(R.id.select_date_button);
        selectDateButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showDatePickerDialog();
            }
        });
    }
}
```

날짜 선택 버튼 생성



# 예제: 날짜와 시간 선택

```
Button selectTimeButton = findViewById(R.id.select_time_button);
selectTimeButton.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        showTimePickerDialog();
    }
});
```

시간 선택 버튼 생성

```
private void showDatePickerDialog() {
    Calendar calendar = Calendar.getInstance();
    int year = calendar.get(Calendar.YEAR);
    int month = calendar.get(Calendar.MONTH);
    int dayOfMonth = calendar.get(Calendar.DAY_OF_MONTH);

    DatePickerDialog datePickerDialog = new DatePickerDialog(
        this,
        new DatePickerDialog.OnDateSetListener() {
            @Override
            public void onDateSet(DatePicker view, int selectedYear, int selectedMonth,
            int selectedDayOfMonth) {
                String selectedDate = selectedYear + "-" + (selectedMonth + 1) + "-" +
                selectedDayOfMonth;
            }
        }
    );
}
```

DatePickerDialog  
대화상자 생성



## 예제:

```
        selectedDateText.setText("선택한 날짜: " + selectedDate);
    }
},
year, month, dayOfMonth
);

datePickerDialog.show();
}

private void showTimePickerDialog() {
    Calendar calendar = Calendar.getInstance();
    int hour = calendar.get(Calendar.HOUR_OF_DAY);
    int minute = calendar.get(Calendar.MINUTE);

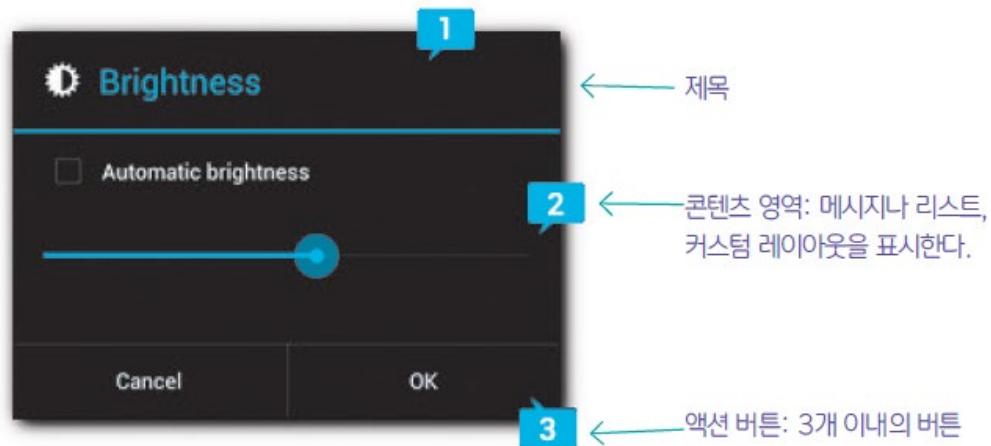
    TimePickerDialog timePickerDialog = new TimePickerDialog(
        this,
        new TimePickerDialog.OnTimeSetListener() {
            @Override
            public void onTimeSet(TimePicker view, int selectedHour, int selectedMinute) {
                String selectedTime = selectedHour + ":" + selectedMinute;
                selectedTimeText.setText("선택한 시간: " + selectedTime);
            }
        }, hour, minute, true);

    timePickerDialog.show();
}
```



# AlertDialog

- AlertDialog는 Dialog 클래스를 상속받은 클래스이다. AlertDialog만 사용하여도 사용자 인터페이스에서 사용되는 대부분의 대화 상자를 구현할 수 있다.



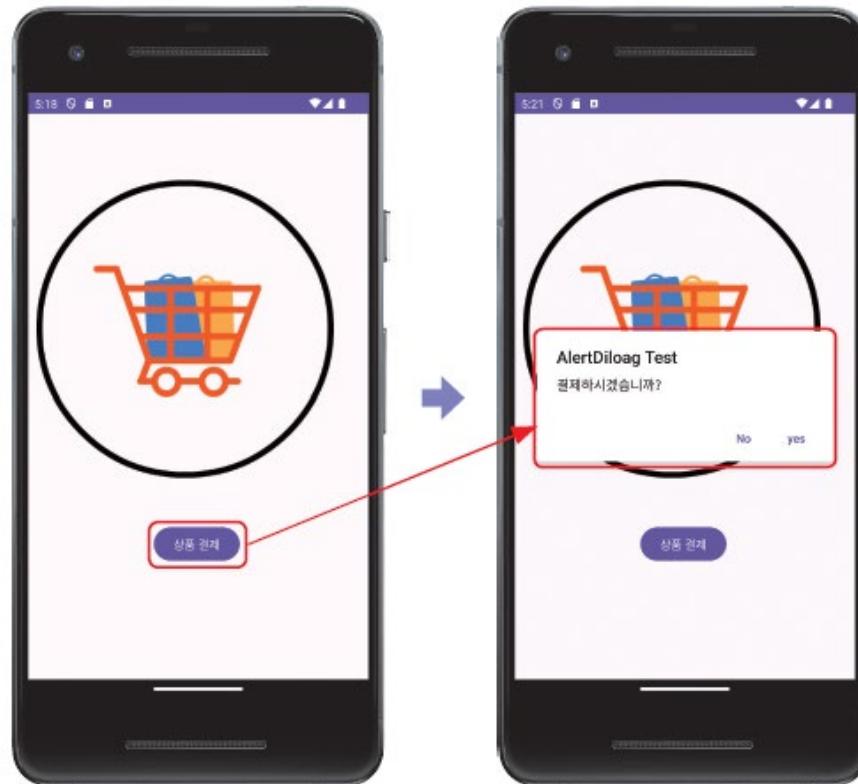
AlertDialog는 사용자 인터페이스에서 필요한 대부분의 대화 상자를 구현할 수 있어요.





# 예제: AlertDialog

- 버튼이 눌러지면 AlertDialog 대화 상자를 화면에 표시하는 애플리케이션을 작성하여 보자.





# 예제: AlertDialog

activity\_main.xml

```
<androidx.constraintlayout.widget.ConstraintLayout >  
    <Button  
        android:id="@+id/button"  
        android:onClick="open"  
        android:text="상품 결제" />  
  
    <ImageView  
        android:id="@+id/imageView"  
        app:srcCompat="@drawable/img" />  
/</androidx.constraintlayout.widget.ConstraintLayout>
```



# 예제: AlertDialog

MainActivity.java

```
package kr.co.company.alertdialogtest;  
// 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 open(View view){  
        AlertDialog.Builder alertDialogBuilder = new AlertDialog.Builder(this);  
        alertDialogBuilder.setTitle("AlertDialog Test");  
        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();  
                }  
            });  
    }  
}
```

빌더 객체를 생성한다.

대화 상자의 메시지 부분 설정

대화 상자의 "yes" 버튼 부분 설정



# 예제: AlertDialog

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

대화상자의 "no" 버튼  
부분 설정

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



# 목록을 사용하는 AlertDialog

```
final CharSequence[] items = { "Red", "Green", "Blue" };

AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setTitle("색상을 선택하시오");
builder.setItems(items, new DialogInterface.OnClickListener(){
    public void onClick(DialogInterface dialog, int item) {
        Toast.makeText(getApplicationContext(), items[item],
        Toast.LENGTH_SHORT).show();}
});
AlertDialog alert = builder.create();
```

색상을 선택하시오

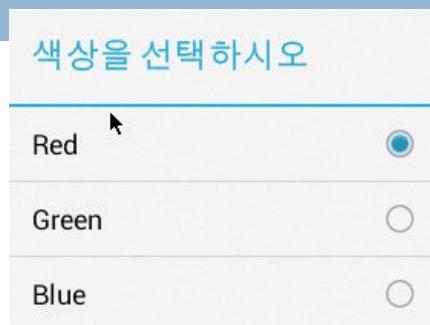
Red

Green

Blue



# 목록을 사용하는 AlertDialog



```
final CharSequence[] items = { "Red", "Green", "Blue" }; ← 빌더 객체를 생성한다.
```

```
AlertDialog.Builder builder = new AlertDialog.Builder(this); ← 대화 상자의 메시지 부분  
builder.setTitle("색상을 선택하시오"); ← 설정
```

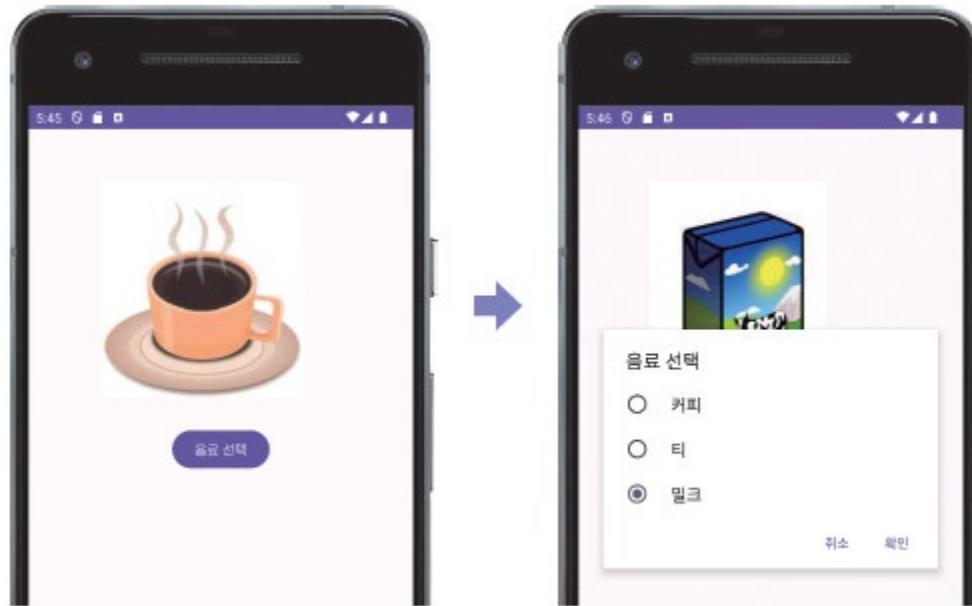
```
builder.setSingleChoiceItems(items, -1, new  
    DialogInterface.OnClickListener() {  
        public void onClick(DialogInterface dialog, int item) {  
            Toast.makeText(getApplicationContext(), items[item],  
                Toast.LENGTH_SHORT).show();  
        }  
    });
```

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

setSingleChoiceItems()의 두 번째 매개변수는 초기에 선택된 항목을 나타내는 정수로, -1은 아직 선택된 항목이 없음을 나타낸다.



# 예제: 라디오 버튼이 있는 AlertDialog 대화 상자





# 예제: 라디오 버튼이 있는 AlertDialog 대화 상자

activity\_main.xml

```
<androidx.constraintlayout.widget.ConstraintLayout>
    <ImageView
        android:id="@+id/imageView"
        app:srcCompat="@drawable/coffee" />
    <Button
        android:id="@+id/show_dialog_button"
        android:text="음료 선택" />
</androidx.constraintlayout.widget.ConstraintLayout>
```



# 예제: 라디오 버튼이 있는 AlertDialog 대화 상자

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    private String selectedDrink = "커피"; // 기본 선택값
    ImageView imageview;

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

        imageview = findViewById(R.id.imageView);
        Button showDialogButton = findViewById(R.id.show_dialog_button);
        showDialogButton.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                showDrinkSelectionDialog();
            }
        });
    }
}
```

버튼이 눌리면 라디오 버튼이 있는 대화 상자를 생성한다.



# 예제: 라디오 버튼이 있는 AlertDialog 대화 상자

```
private void showDrinkSelectionDialog() {  
    final String[] drinks = {"커피", "티", "밀크"};  
  
    AlertDialog.Builder builder = new AlertDialog.Builder(this);  
    builder.setTitle("음료 선택")  
        .setSingleChoiceItems(drinks, -1, new DialogInterface.OnClickListener() {  
  
            @Override  
            public void onClick(DialogInterface dialog, int which) {  
                selectedDrink = drinks[which];  
                if( which==0) imageview.setImageResource(R.drawable.tea);  
                else if(which==1) imageview.setImageResource(R.drawable.tea);  
            }  
        });  
    builder.setPositiveButton("선택", new DialogInterface.OnClickListener() {  
        public void onClick(DialogInterface dialog, int which) {  
            //선택한 음료를 처리하는 로직  
        }  
    });  
    builder.setNegativeButton("취소", new DialogInterface.OnClickListener() {  
        public void onClick(DialogInterface dialog, int which) {  
            dialog.cancel();  
        }  
    });  
    builder.show();  
}
```

빌더 객체를 먼저 생성하여 여러 가지 속성을 설정한다.



# 예제: 라디오 버튼이 있는 AlertDialog 대화 상자

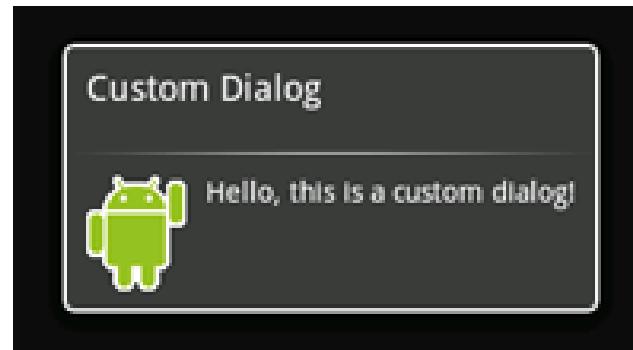
```
        else imageView.setImageResource(R.drawable.milk);
    }
})
.setPositiveButton("확인", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        // 선택한 음료에 대한 추가 작업 수행
        // 실제 액션을 수행하거나 화면에 표시하는 방식으로 변경 가능
        dialog.dismiss();
    }
})
.setNegativeButton("취소", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        dialog.dismiss();
    }
});

builder.create().show();
}
```



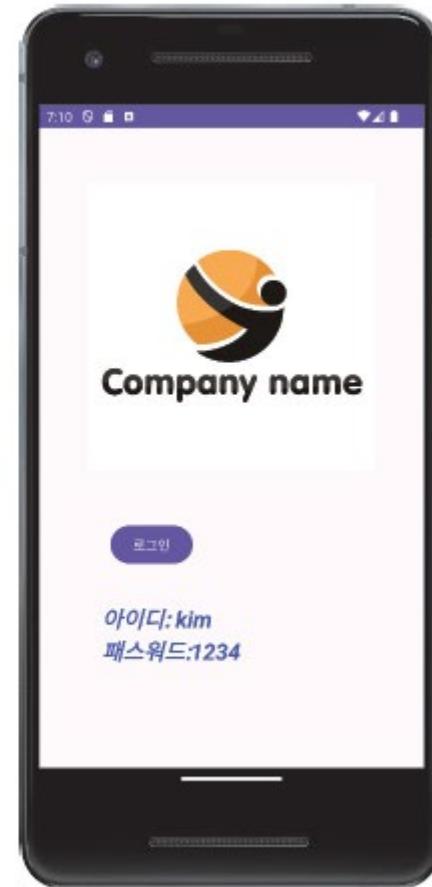
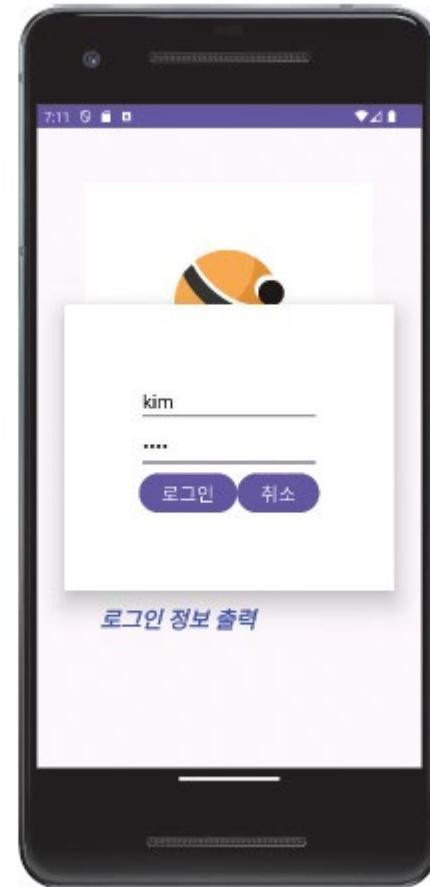
# 커스텀 대화상자

- 커스텀 대화 상자(custom dialog)란 사용자가 마음대로 대화 상자의 내용을 디자인할 수 있는 대화 상자이다.





# 예제: 커스텀 대화상자





# 예제: 커스텀 대화상자

activity\_main.xml

```
<androidx.constraintlayout.widget.ConstraintLayout >  
    <ImageView  
        android:id="@+id/imageView"  
        app:srcCompat="@drawable/img" />  
  
    <Button  
        android:id="@+id/button"  
        android:onClick="onClick"  
        android:text="로그인" />  
  
    <TextView  
        android:id="@+id/textView"  
        android:text="로그인 정보 출력" />  
</androidx.constraintlayout.widget.ConstraintLayout>
```



예 7

custom\_dialog.xml

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical"  
    android:padding="8dp">
```

```
    <EditText  
        android:id="@+id/username"  
        android:hint="username"  
        android:textSize="20sp">  
        <requestFocus></requestFocus>  
    </EditText>
```

```
    <EditText  
        android:id="@+id/password"  
        android:hint="password"  
        android:inputType="textPassword"  
        android:textSize="20sp"></EditText>
```

```
    <LinearLayout>  
        <Button  
            android:id="@+id/login"  
            android:layout_weight="0.5"  
            android:text="로그인"  
            android:textSize="20sp"></Button>
```

```
        <Button  
            android:id="@+id/cancel"  
            android:layout_weight="0.5"  
            android:text="취소"  
            android:textSize="20sp"></Button>  
    </LinearLayout>  
</LinearLayout>
```





# 예제. 커스터 대화상자

MainActivity.java

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

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

커스텀 대화 상자를 생성한다.

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



# 예제: 커스텀 대화상자

```
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();
            text.setText("아이디: " + username.getText() + "\n패스워드: " + password.
                getText());
            loginDialog.dismiss();
        } else {
            Toast.makeText(getApplicationContext(), "다시 입력하시오",
                Toast.LENGTH_LONG).show();
        }
    }
});
```

커스텀 대화상자의 로그인  
버튼이 클릭되면

커스텀 대화 상자의 취소 버  
튼이 클리리되어



# 예제: 커스텀 대화상자

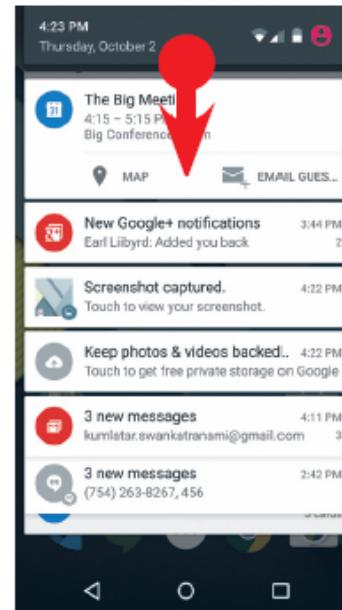
```
cancel.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        loginDialog.dismiss();  
    }  
});  
  
loginDialog.show();  
}  
}
```

커스텀 대화 상자의 취소 버튼이 클릭되면



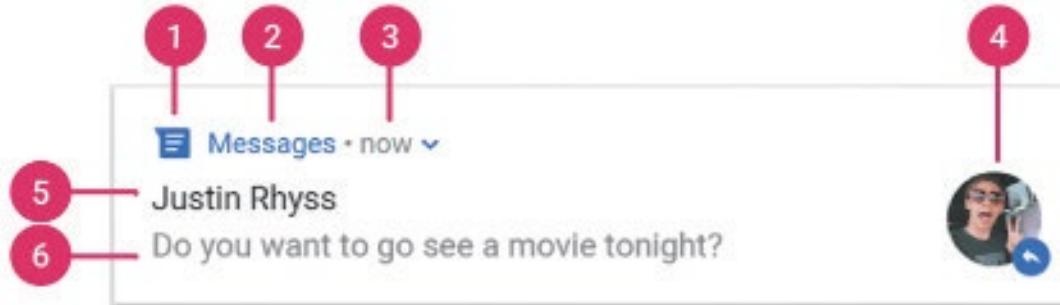
# 알림 기능

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





# 알림 구조



- ① 작은 아이콘: 이것은 필수적으로 있어야 한다. `setSmallIcon()`으로 설정한다.
- ② 앱 이름: 안드로이드에서 제공한다.
- ③ 알림 발생 시간: 안드로이드에서 제공하지만, `setWhen()`으로 재정의할 수 있다.
- ④ 큰 아이콘: 선택 항목이며, `setLargeIcon()`으로 설정할 수 있다.
- ⑤ 제목: 선택 항목이며, `setContentTitle()`로 설정한다.
- ⑥ 텍스트: 선택 항목이며, `setContentText()`로 설정한다.



# 알림을 만드는 절차

알림 채널  
생성하기

알림 빌더를  
생성한다.

알림 속성을  
설정한다.

액션을 첨부한  
다.(선택 사양)

알림 객체 생성  
하며 보내기



# 알림 채널 생성하기

- 안드로이드 버전 8.0(API 레벨 26)부터 모든 알림은 채널에 할당되어야 한다. 채널을 사용하지 않으면 알림이 나타나지 않는다. 알림을 채널로 분류하면 사용자가 앱의 특정 알림 채널을 비활성화할 수 있고 각 채널마다 여러 가지 옵션을 제어할 수 있다.

```
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);
    }
}
```



# 알림 빌더를 생성한다.

- 첫 번째 단계는 `NotificationCompat.Builder.build()`를 사용하여 알림 빌더를 만드는 것이다. `Notification Builder`를 사용하여 큰 아이콘, 작은 아이콘, 제목, 우선순위 등과 같은 다양한 `Notification` 속성을 설정 한다.

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



# 알림 속성을 설정한다.

- Builder 객체가 생성되면 필요에 따라 Builder 객체를 사용하여 Notification 속성을 설정할 수 있다. 다음의 3가지 속성은 반드시 설정하여야 한다.
  - setSmallIcon( ): 작은 아이콘
  - setContentTitle( ): 알림의 제목
  - setContentText( ): 알림의 상세 텍스트

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



# 액션을 첨부한다.

- 이것은 선택적 요소이며, 알림과 함께 어떤 조치를 첨부하려는 경우에 필요하다. 액션을 첨부하면 사용자가 알림에서 애플리케이션의 액티비티로 바로 갈 수 있게 한다.

```
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

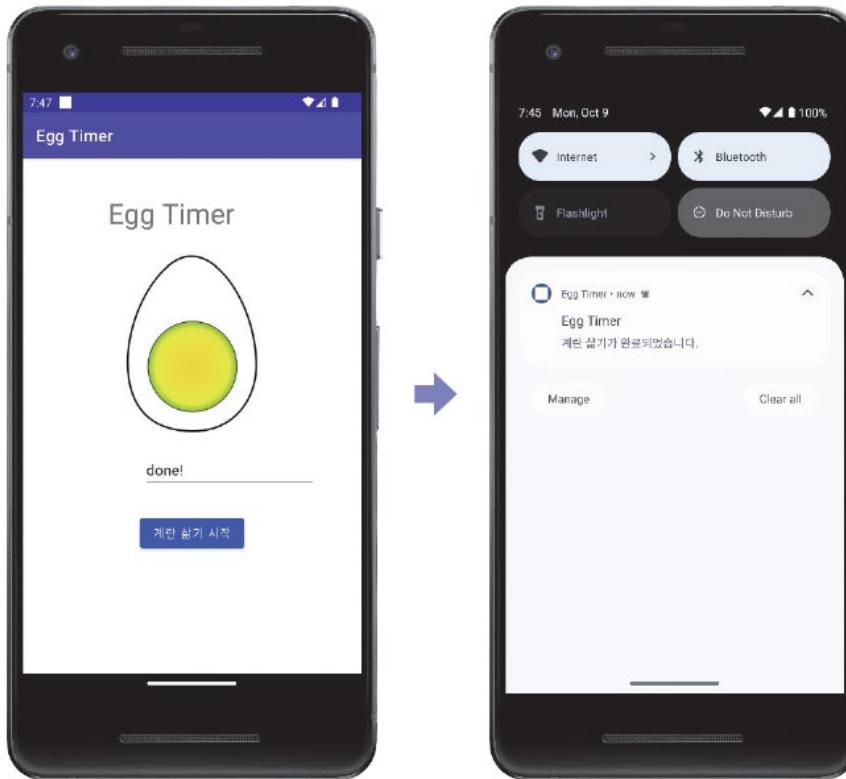
- 알림을 생성하려면 `NotificationManager.notify( )`를 호출하여 시스템에 Notification 객체를 전달한다.

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



# 예제: 버튼을 누르면 알림을 보내는 앱

- 계란 삶는 것을 도와주는 앱을 만들어보자. 계란 삶기는 간단하지만, 시간을 추적하지 않으면 어려운 작업이 될 수 있다.





# 예제: 버튼을 누르면 알림을 보내는 앱

activity\_main.xml

```
<androidx.constraintlayout.widget.ConstraintLayout >  
    <ImageView  
        android:id="@+id/imageView"  
        app:srcCompat="@drawable/boiled_egg"  
        tools:srcCompat="@drawable/boiled_egg" />  
  
    <EditText  
        android:id="@+id/edit"  
        android:text="01:00" />  
  
    <Button  
        android:id="@+id/button"  
        android:onClick="startTimer"  
        android:text="계란 삶기 시작" />  
  
    <TextView  
        android:id="@+id/textView"  
        android:text="Egg Timer"/>  
</androidx.constraintlayout.widget.ConstraintLayout>
```



# 예제. 자바 소스

MainActivity.java

```
public class MainActivity extends AppCompatActivity {

    private EditText mEditText;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        mEditText = (EditText) findViewById(R.id.edit);
        createNotificationChannel();
    }

    String NOTIFICATION_CHANNEL_ID = "my_channel_id_01";
```

알림 채널을 생성한다.

```
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);
    }
}
```



## 예제: 자바 소스

```
public void sendNotification() {  
    NotificationCompat.Builder notificationBuilder = new NotificationCompat.Builder(this,  
        NOTIFICATION_CHANNEL_ID);  
  
    // 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, PendingIntent.FLAG_IMMUTABLE);  
    디자인 패턴 중에서 빌더 패턴을 사용하고 있다.  
  
    notificationBuilder.setSmallIcon(R.drawable.ic_launcher_background) ←  
        .setContentTitle("Egg Timer")  
        .setContentText("계란 삶기가 완료되었습니다.")  
        .setContentIntent(pendingIntent);  
  
    NotificationManager notificationManager = (NotificationManager)  
        getSystemService(Context.NOTIFICATION_SERVICE);  
    notificationManager.notify(/*notification id*/1, notificationBuilder.build());  
}
```



# 예제: 자바 소스

```
public void startTimer (View view){  
    String s = mEditText.getText().toString();  
    int min = Integer.parseInt(s.substring(0, 2));  
    int sec = Integer.parseInt(s.substring(3, 5));  
    new CountDownTimer(min * 60 * 1000 + sec * 1000, 1000) {  
        public void onTick(long millisUntilFinished) {  
            mEditText.setText(":" + (int) (millisUntilFinished / (1000)) + "초");  
        }  
        ← 타이머가 끝나면 알림을 보낸다.  
        public void onFinish() {  
            mEditText.setText("done!");  
            sendNotification();  
        }  
    }.start();  
}
```



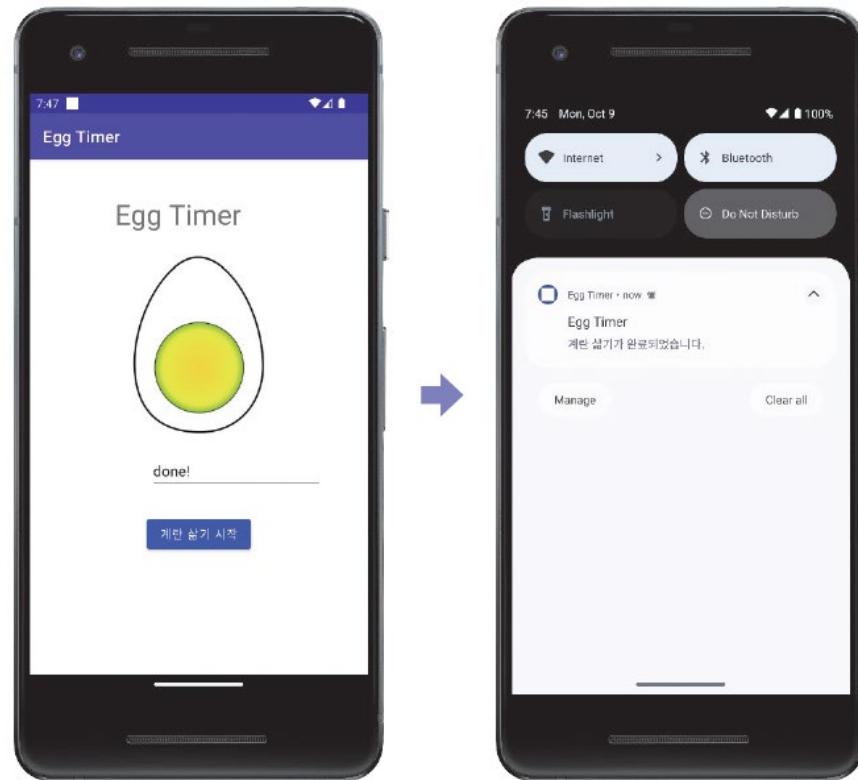
# 메니페스트 파일 수정

AndroidManifest.xml

```
<uses-permission android:name="android.permission.POST_NOTIFICATIONS" />
```



# 실행 결과





# Q & A

