



CHAP 7. 메뉴와 대화상자(1)

1

안드로이드의 사용자 인터페이스

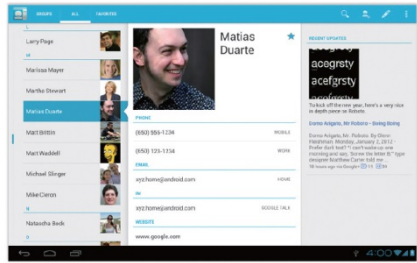
- 네비게이션 바



- 액션바



- 다중 패널 레이아웃



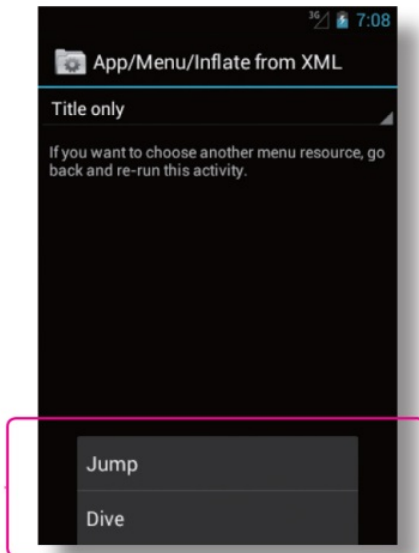
- 제스처



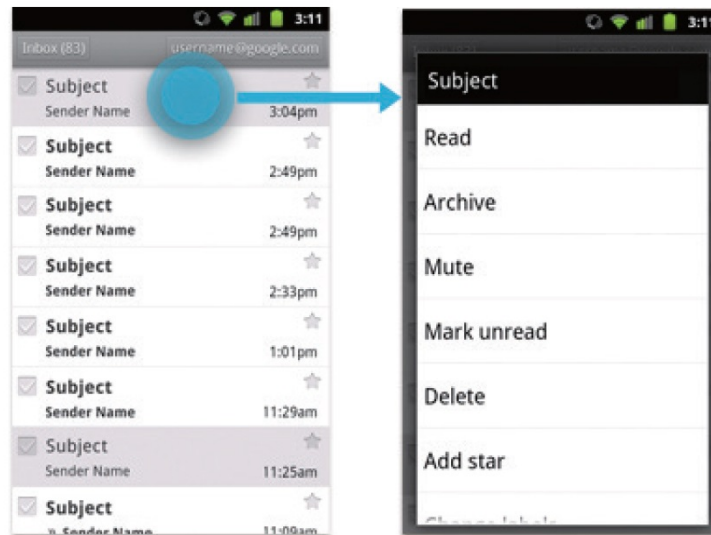
메뉴의 종류

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

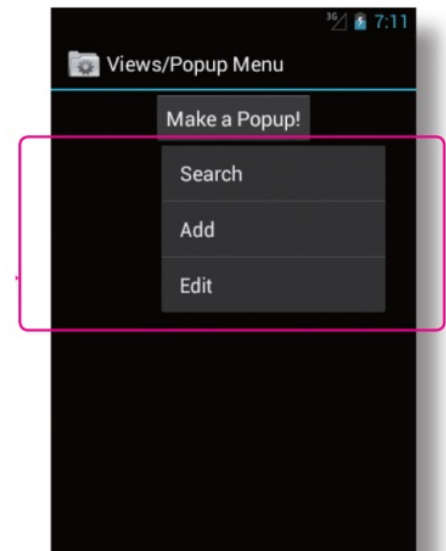
옵션 메뉴



컨텍스트 메뉴

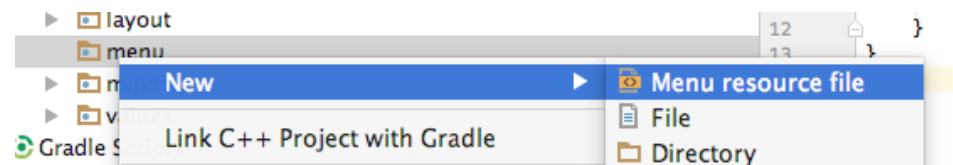
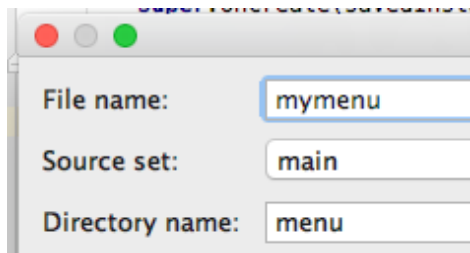
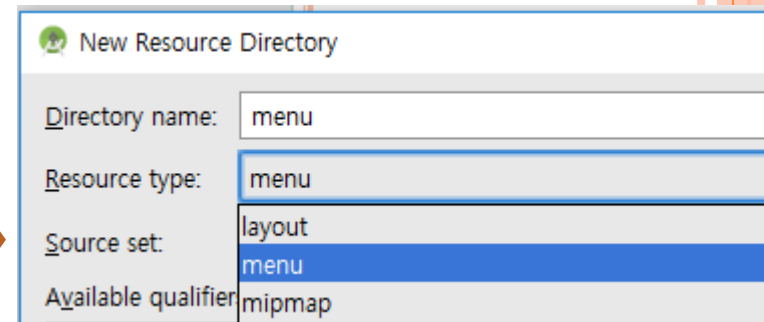
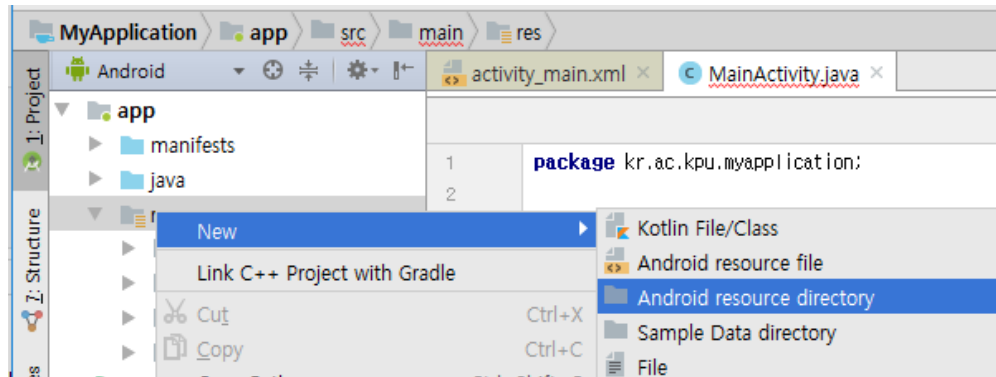


팝업메뉴



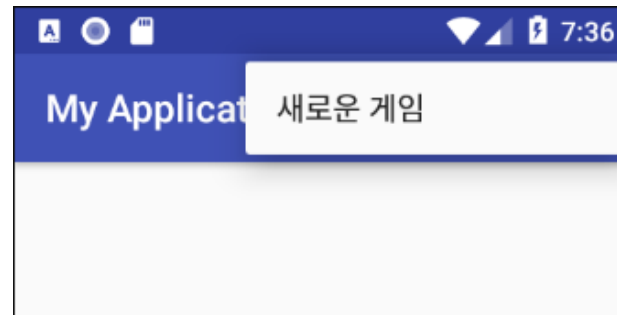
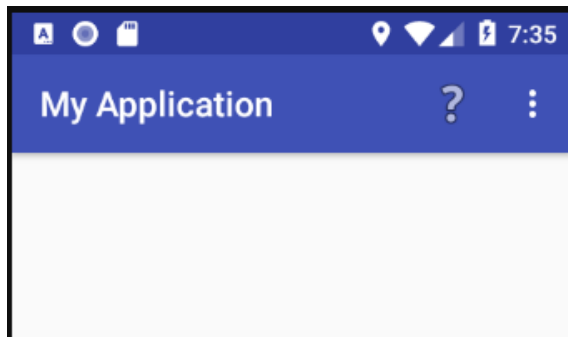
메뉴를 생성하는 방법

- 코드로 메뉴 생성하기
- XML로 메뉴 생성하기



메뉴도 XML로 정의

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
  <item
    android:id="@+id/new_game"
    android:icon="@android:drawable/ic_menu_gallery"
    android:title="새로운 게임" />
  <item
    android:id="@+id/help"
    android:icon="@android:drawable/ic_menu_help"
    app:showAsAction="ifRoom"
    android:title="도움말" />
</menu>
```



메뉴 팽창

- 메뉴 리소스를 팽창(inflate)하면 실제 메뉴가 생성



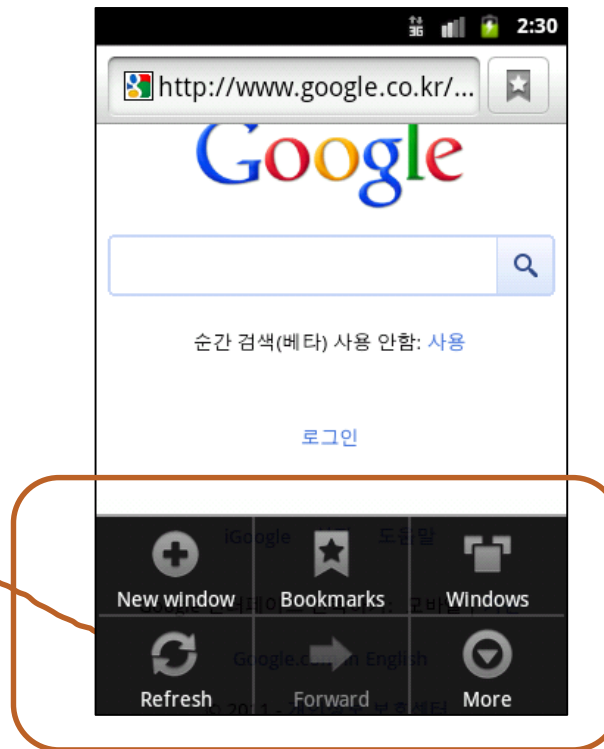
메뉴 리소스 팅창

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

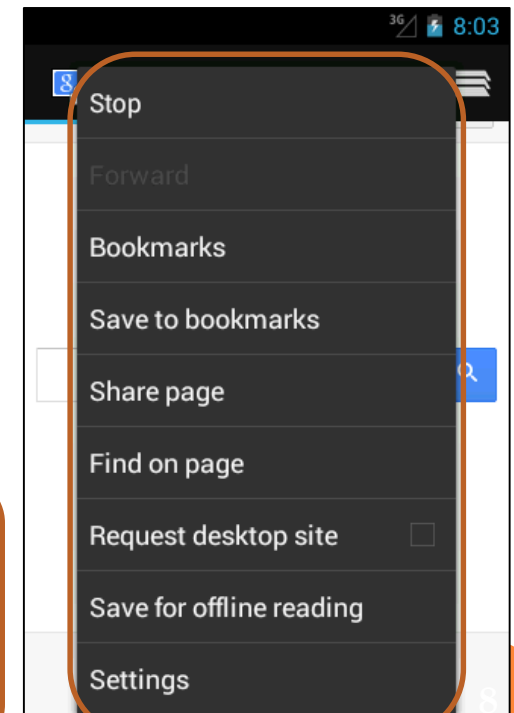
← 메뉴 리소스 팅창

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 항목
이 클릭되면 실행

← "도움말" 메뉴 항목이 클
릭되면 실행

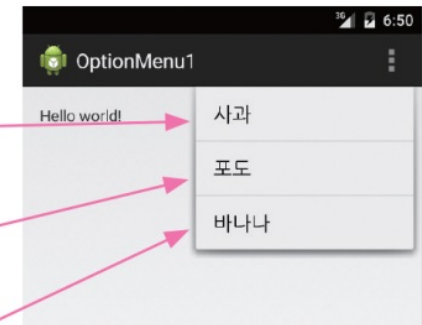
← 이벤트를 상위 클래스로
전달

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

- 사용자가 옵션 키를 누르면 다음의 메소드가 호출된다.

OptionsMenu2Activity.java

```
package kr.co.company.optionmenu2;  
// 소스만 입력하고 Alt+Enter를 눌러서 import 문장을 자동으로 생성한다.
```

```
public class OptionsMenu2Activity extends ActionBarActivity {
```

```
@Override
```

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

```
@Override
```

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

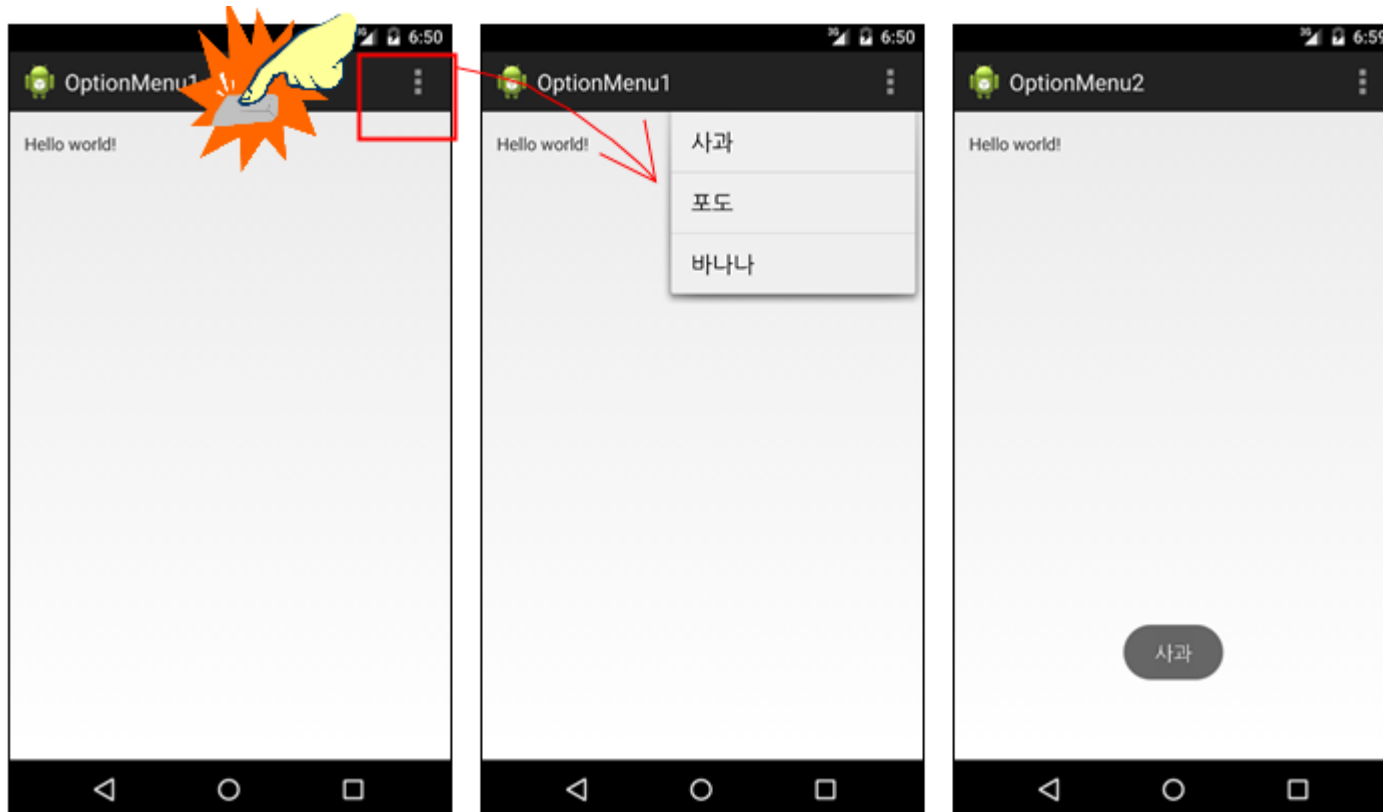
← 각 메뉴항목이 눌러지면 토스트 메시지를 표시한다.

옵션 메뉴 이벤트 처리

메뉴 mymenu.xml을 팽창하여 메뉴 객체로 만든다.

```
@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

OptionMenu1Activity.java

```
package kr.co.company.optionmenu1;  
// 소스만 입력하고 Ctrl-Shift-O를 눌러서 import 문장을 자동으로 생성한다.
```

```
public class OptionMenu1Activity extends Activity {  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.main);  
    }  
  
    @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');  
  
        menu.add(0, 2, 0, "포도").setIcon(R.drawable.ic_launcher);  
        menu.add(0, 3, 0, "바나나").setIcon(R.drawable.ic_launcher);  
  
        return true;  
    }  
}
```

“사과”라는 메뉴 항목을 추가하고 아이콘과 단축키도 설정한다.

메뉴 항목의 아이디는 1

반환되는 값을 이용하여서 이렇게 하어도 된다.

코드로 옵션 메뉴 생성하기 #2 (실습)

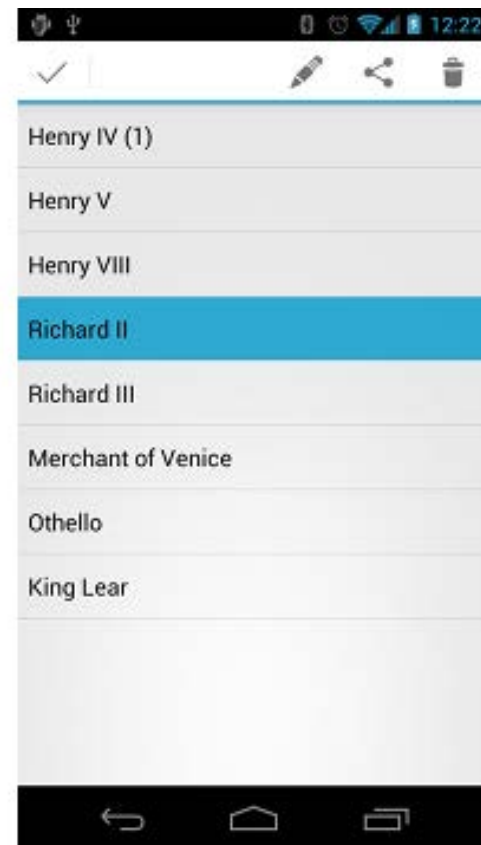
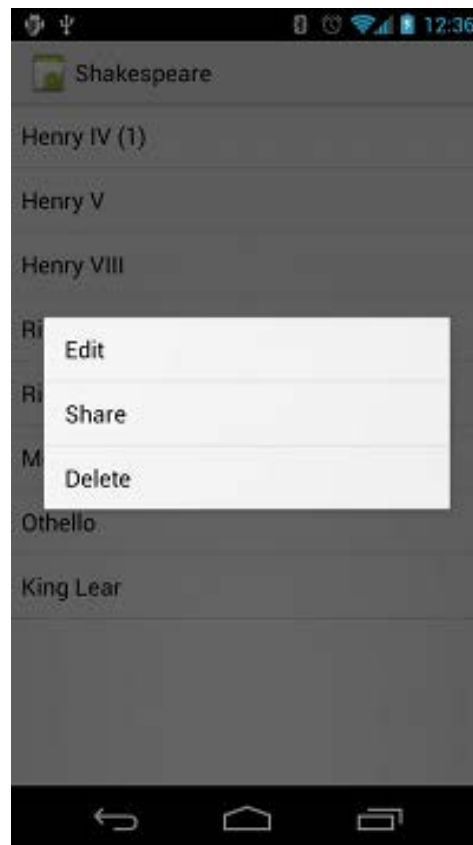
@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case 1:  
            Toast.makeText(this, "사과", Toast.LENGTH_SHORT).show();  
            return true;  
        case 2:  
            Toast.makeText(this, "포도", Toast.LENGTH_SHORT).show();  
            return true;  
        case 3:  
            Toast.makeText(this, "바나나", Toast.LENGTH_SHORT).show();  
            return true;  
        default:  
            return super.onOptionsItemSelected(item);  
    }  
}
```

메뉴 항목이 선택되면
호출된다.

컨텍스트 메뉴

- UI의 어떤 항목과 관련된 동작



컨텍스트 메뉴 종류

○ 플로팅 컨텍스트 메뉴:

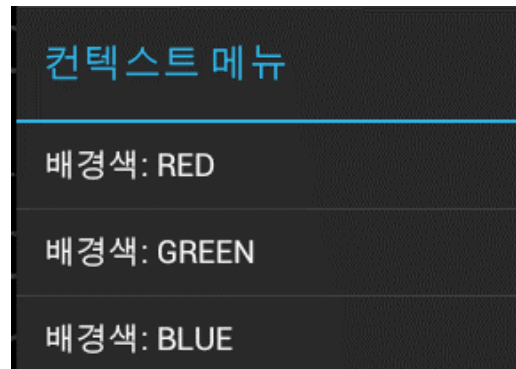
- 사용자가 항목 위에서 오래 누르기(long click)를 하면 메뉴가 대화 상자처럼 떠서 표시된다.

○ 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="50px"
        android:typeface="serif" />
</LinearLayout>
```

컨텍스트 메뉴 예제

```
...  
public class ContextMenuActivity extends Activity {  
    TextView text;  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.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

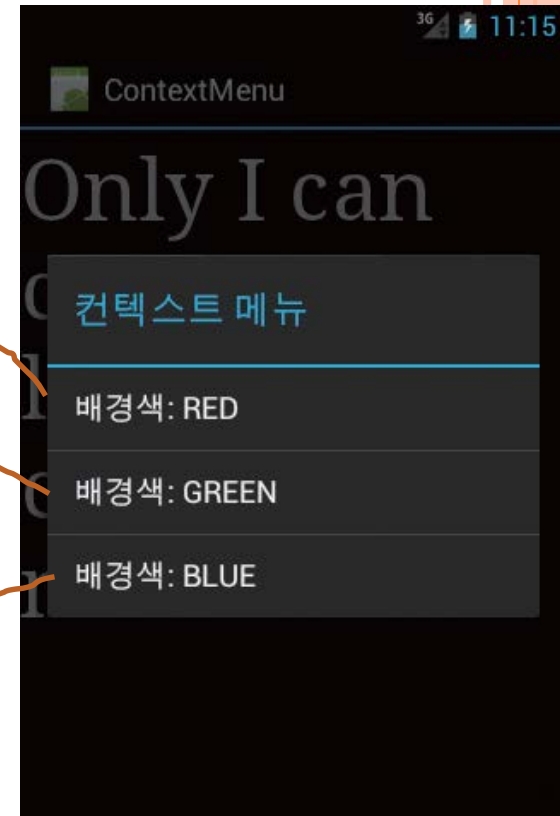
배경색: RED

배경색: GREEN

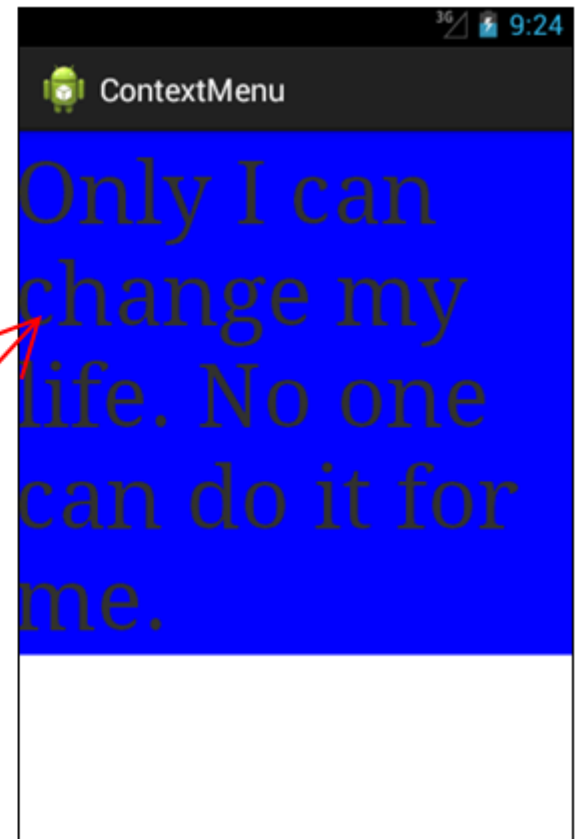
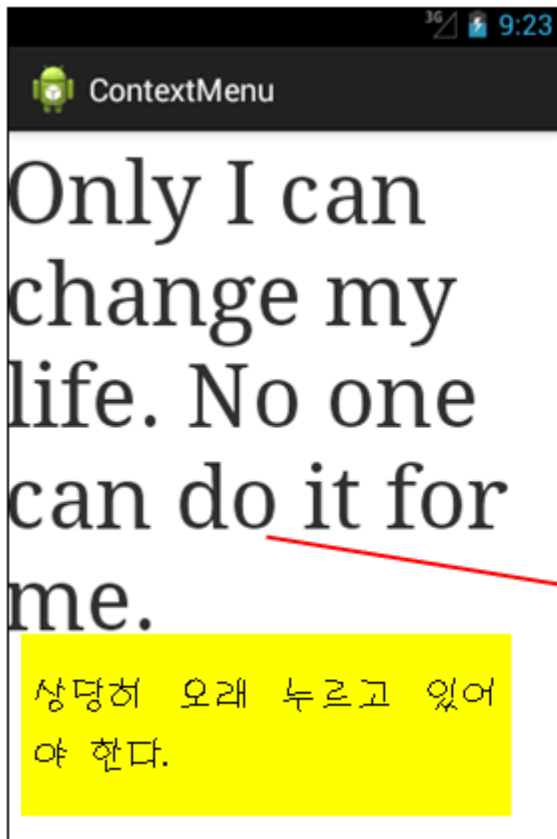
배경색: BLUE

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);  
    }  
}
```

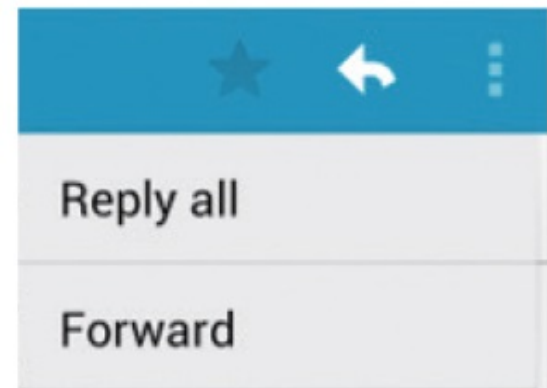


실행 결과



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

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 모든
메뉴이다.

예제: 팝업 메뉴

PopupMenuActivity.java

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

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

```
public class PopupMenuActivity extends ActionBarActivity {
```

```
    @Override
```

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

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

```
        setContentView(R.layout.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;
```

```
                }
```

```
            });
```

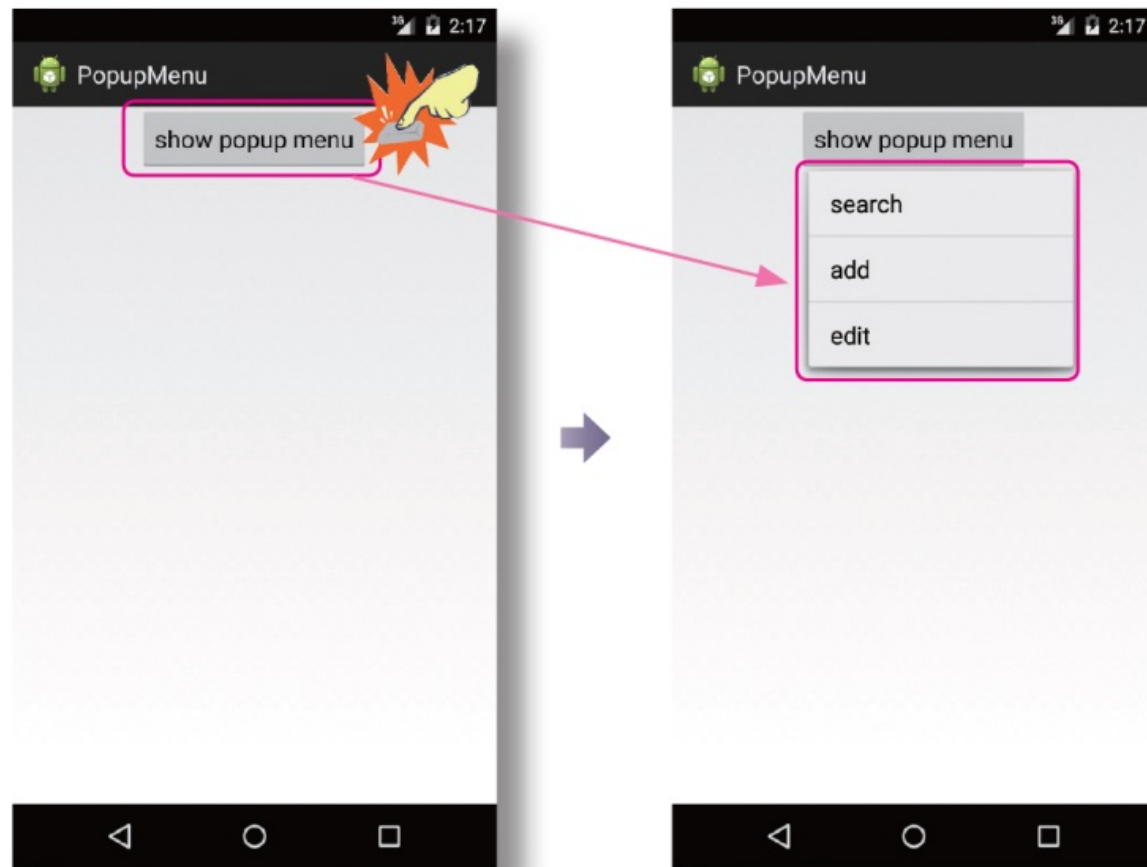
```
        popup.show();
```

```
    }
```

```
}
```

팝업 메뉴 항목이 눌리지
면 여기서 처리한다. |

실행 결과



액션바

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



XML 파일

/menu/action_bar.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >
```

오타: android → app

```
<item
  android:id="@+id/action_refresh"
  android:icon="@drawable/ic_action_refresh"
  android:showAsAction="always"
  android:title="refresh" />
```

← refresh 항목

← 항상 액션바에 표시

```
<item
  android:id="@+id/action_search"
  android:icon="@drawable/ic_action_search"
  android:showAsAction="ifRoom"
  android:title="search" />
```

search 항목 →

← 액션바에 남는 공간이 있는 경우에만 표시

```
<item
  android:id="@+id/action_settings"
  android:orderInCategory="100"
  android:showAsAction="never"
  android:title="@string/action_settings" />
```

← 액션바에는 표시하지 않는다.

```
</menu>
```

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
  <item
    android:id="@+id/action_refresh"
    android:icon="@android:drawable/ic_popup_sync"
    app:showAsAction="always"
    android:title="refresh"/>
  <item
    android:id="@+id/action_search"
    android:icon="@android:drawable/ic_menu_search"
    app:showAsAction="ifRoom"
    android:title="search" />
  <item
    android:id="@+id/action_settings"
    app:showAsAction="never"
    android:title="settings" />
</menu>

```

```
public class ActionBarActivity extends Activity {
```

```
    @Override
```

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

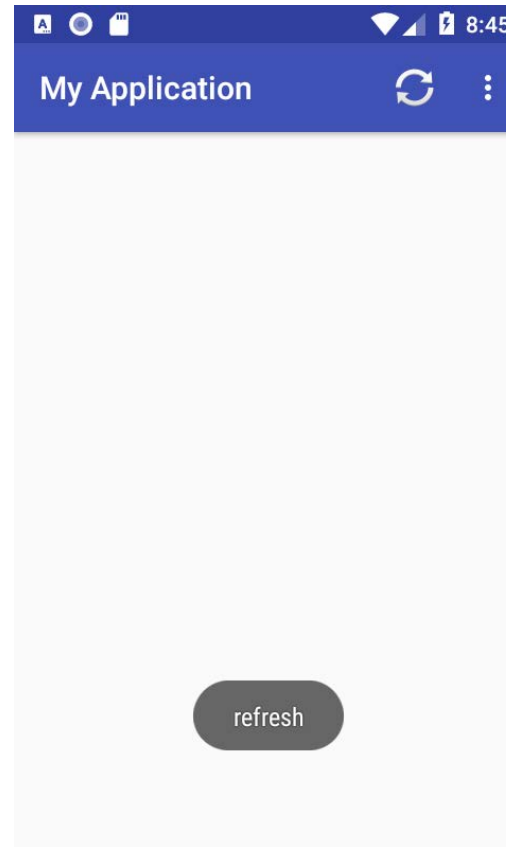
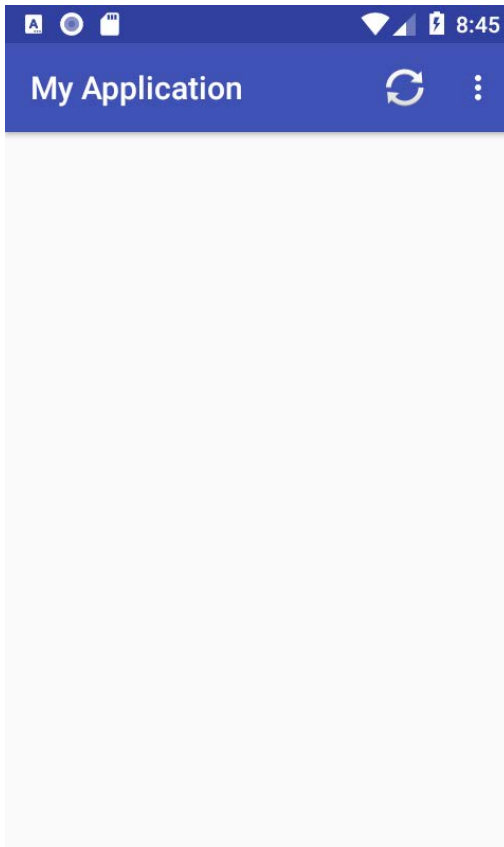
```
    @Override
```

```
    public boolean onCreateOptionsMenu(Menu menu) {  
        // 메뉴 리소스를 팅창한다. 액션바에 항목들을 추가한다.  
        getMenuInflater().inflate(R.menu.action_bar, menu);  
        return super.onCreateOptionsMenu(menu);  
    }
```

```
    @Override
```

```
    public boolean onOptionsItemSelected(MenuItem item) {  
        // 액션바의 항목들이 클릭되면 이벤트를 처리한다.  
        switch (item.getItemId()) {  
            case R.id.action_refresh:  
                Toast.makeText(this, "refresh", Toast.LENGTH_SHORT).show();  
                return true;  
            case R.id.action_search:  
                Toast.makeText(this, "search", Toast.LENGTH_SHORT).show();  
                return true;  
            case R.id.action_settings:  
                Toast.makeText(this, "settings", Toast.LENGTH_SHORT).show();  
                return true;  
            default:  
                return super.onOptionsItemSelected(item);  
        }  
    }  
}
```


실행 결과



실습1

