

Ch07. 메뉴와 대화상자 1

모바일앱프로그래밍
김지심교수

학습목표



메뉴를 구현할 수 있다.
토스트를 다양하게 구현할 수 있다.
대화상자를 구현할 수 있다.

목차

1. 메뉴
2. 토스트
3. 대화상자



01



메뉴



1. 메뉴 | 옵션메뉴

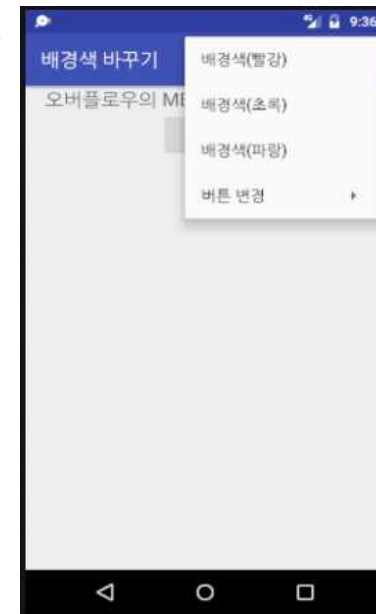
메뉴

안드로이드의 기본 메뉴: 옵션 메뉴(OptionsMenu), 컨텍스트 메뉴(ContextMenu)

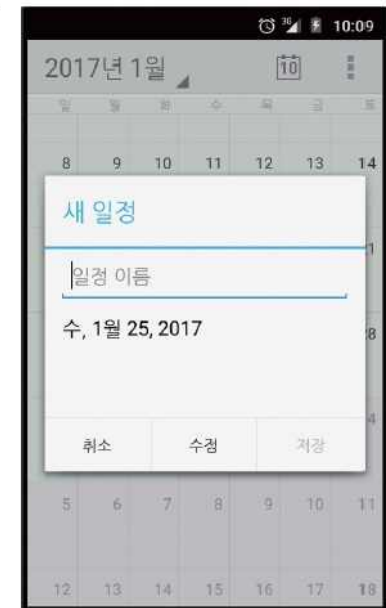
옵션 메뉴를 구현하는 방법

이전 버전에서는 하단에, 최신 버전에서는 오버플로우 영역을 누르면 화면 상단에 표시됨

- 1) 메뉴 XML 파일을 정의한 후 Java에서 바인딩
- 2) 클래스만으로 메뉴를 생성



(a) 옵션 메뉴



(b) 컨텍스트 메뉴(날짜를 톱클릭한 화면)

1. 메뉴 | 옵션메뉴

1) 메뉴 XML 파일을 이용한 방식



1. 메뉴 | 옵션메뉴

메뉴 XML 파일 형식

```
<menu>
    <item
        android:id="@+id/항목1아이디"
        android:title="항목1 제목"/>
    <item
        android:id="@+id/항목2아이디"
        android:title="항목2 제목"/>
</menu>
```

1. 메뉴 | 옵션메뉴

onCreateOptionsMenu() 메소드 기본 형식

앱이 실행되면 메뉴 내용을 XML에서 읽어와 메뉴가 등록되도록
인플레이션해주어야 함

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

* 싱글톤(Singleton Pattern): 단 하나의 유일한 객체를 만들어 사용
직접 객체를 생성하지 않고 생성된 객체를 얻어옴

method chaining 가능
getMenuInflater().inflate(int menuRes, Menu menu);

1. 메뉴 | 옵션메뉴

* 인플레이터(Inflater) p.291

원의미: ‘부풀리는 장치’, ‘자전거 등의 공기펌프’

개념: 풍선에 바람을 넣어서 부풀려 풍선답게 만들듯이
실제 객체로 만들어 사용한다는 의미로,
정적으로 정의된 XML파일을 Java코드에서 접근하여 실제
객체로 만들어 사용하는 것

* 온전한 Activity 단위가 아닌 (xml로 정의된) 메뉴, 대화상자용 화면,
부분화면 등은 수동으로 inflating해야 함

예시

MenuInflater 클래스: 메뉴 XML 파일을 클래스와 바인딩하여
인플레이션할 수 있도록 해줌

1. 메뉴 | 옵션메뉴

onOptionsItemSelected() 메소드 기본 형식
메뉴 선택 시 동작 코딩

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.항목1아이디:  
            항목1을 선택했을 때 실행할 코드  
            return true;  
        case R.id.항목2아이디:  
            항목2를 선택했을 때 실행할 코드  
            return true;  
    }  
    return false;  
}
```

1. 메뉴 | 옵션메뉴

예제7-1: Ex07_01

오버플로우 클릭 시 나타나는 메뉴로
배경색을 변경하는 앱 만들기



1. 메뉴 | 옵션메뉴

activity_main.xml

리니어레이아웃의 id를 baseLayout으로 지정

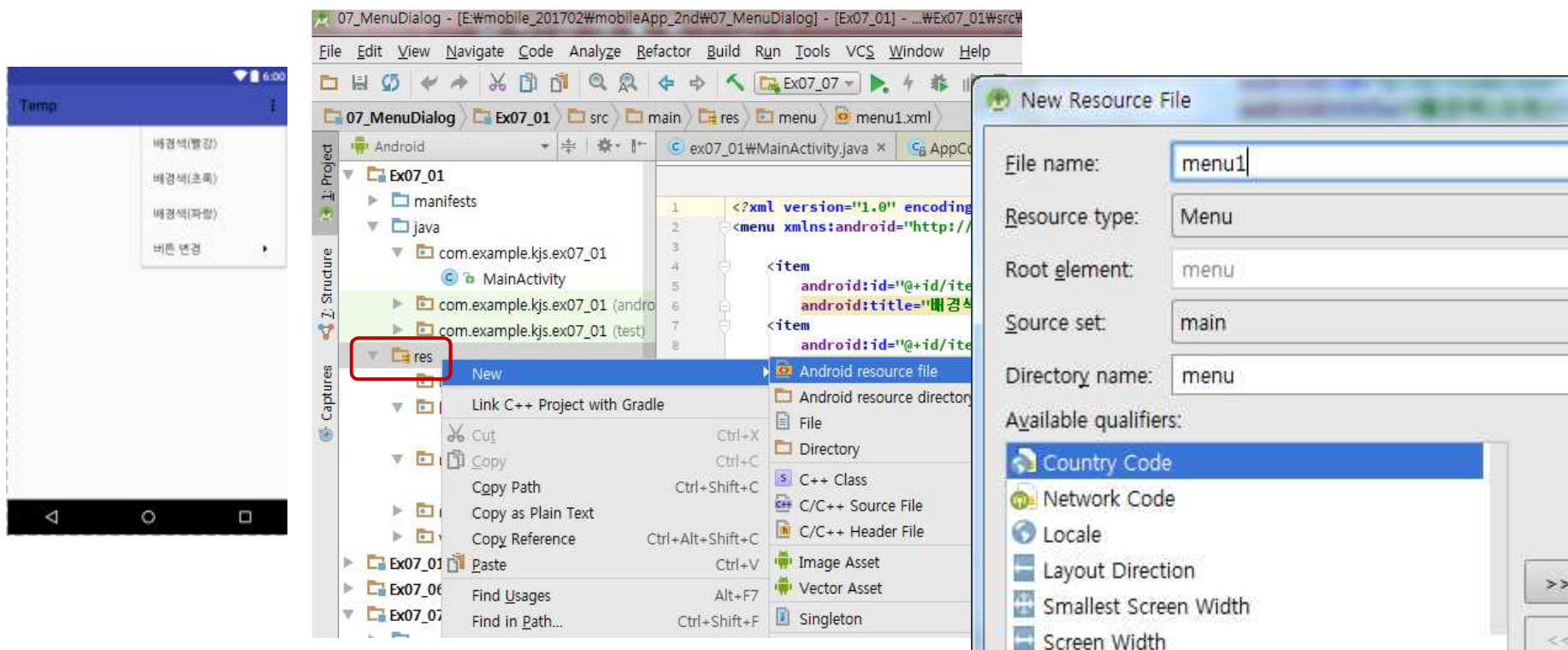
```
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      android:id="@+id/baseLayout" 클래스에서 레이아웃 객체에 접근하여 배경색을 바꾸기 위해 id 지정
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:gravity="center_horizontal"
7      android:orientation="vertical" >
8
9      <TextView
10         android:layout_width="wrap_content"
11         android:layout_height="wrap_content"
12         android:text="오버플로우의 MENU 버튼을 누르세요."
13         android:textSize="20sp" />
14
15         <Button
16             android:id="@+id/btnMenu"
17             android:layout_width="wrap_content"
18             android:layout_height="wrap_content"
19             android:text="버튼" />
20
21 </LinearLayout>
```

1. 메뉴 | 옵션메뉴

menu1.xml

프로젝트 트리에서 [app] > [res] > [New] > [Android Resource file] > menu 생성 > [res] 하위에 [menu] 폴더가 생성되고 menu1.xml이 추가됨

File name: menu1, Resource Type: menu 선택



1. 메뉴 | 옵션메뉴

menu1.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <menu xmlns:android="http://schemas.android.com/apk/res/android">
3
4     <item
5         android:id="@+id/itemRed"
6         android:title="배경색(빨강)" />
7     <item
8         android:id="@+id/itemGreen"
9         android:title="배경색(초록)" />
10    <item
11        android:id="@+id/itemBlue"
12        android:title="배경색(파랑)" />
13    <item
14        android:title="버튼 변경">
15        <menu>
16            <item
17                android:id="@+id/subRotate"
18                android:title="버튼 45도 회전" />
19            <item
20                android:id="@+id/subSize"
21                android:title="버튼 2배 확대" />
22        </menu>
23    </item>
24 </menu>
25
```

클래스에서 참조할 필요가 없다면
하위메뉴의 컨테이너 메뉴는 id 생략



1. 메뉴 | 옵션메뉴

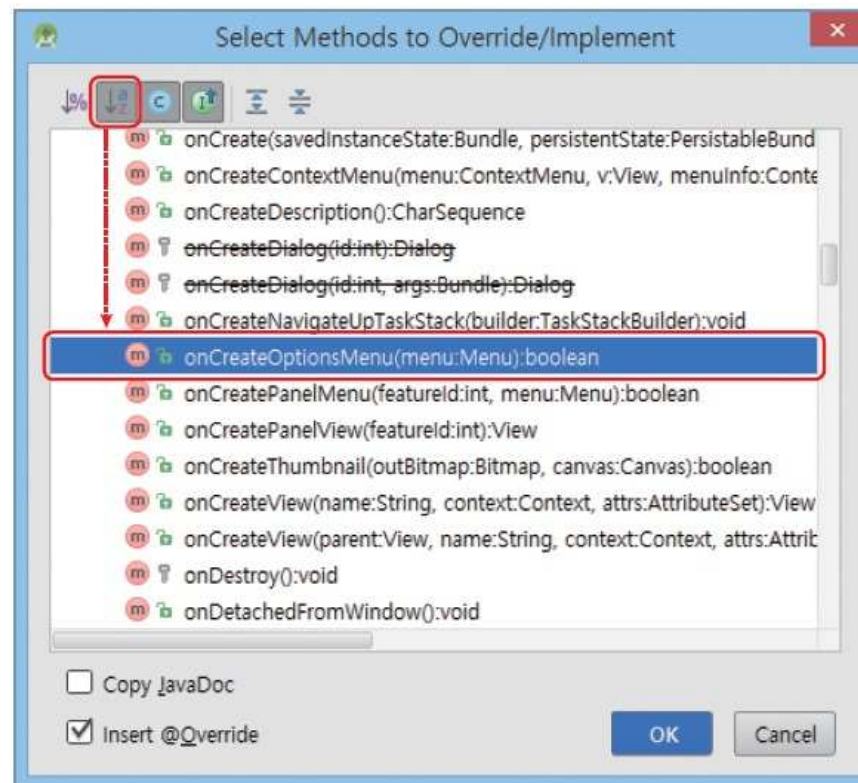
MainActivity.java

```
12 public class MainActivity extends AppCompatActivity {  
13  
14     LinearLayout baseLayout;  
15     Button btnChange;  
16  
17  
18     @Override  
19     protected void onCreate(Bundle savedInstanceState) {  
20         super.onCreate(savedInstanceState);  
21         setContentView(R.layout.activity_main);  
22  
23         baseLayout = (LinearLayout) findViewById(R.id.baseLayout);  
24         btnChange = (Button) findViewById(R.id.btnMenu);  
25     }  
26
```

1. 메뉴 | 옵션메뉴




























옵션메뉴를 등록하는 onCreateOptionsMenu()를 Activity 클래스로부터 오버라이드

onCreate() 바깥에 커서 두고, [Generate] > [Override Methods] > 메소드 선택 혹은 Ctrl + O



1. 메뉴 | 옵션메뉴

onCreateOptionsMenu()을 오버라이딩하여 옵션메뉴 등록

```
26  
27  
28                             
29  
30  
31  
32  
33
```

```
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    super.onCreateOptionsMenu(menu);  
    MenuInflater inflater = getMenuInflater();  
    inflater.inflate(R.menu.menu1, menu);  
    return true; 옵션메뉴를 등록하여 출력  
}
```

1. 메뉴 | 옵션메뉴

메뉴 클릭 이벤트를 처리하기 위해 메소드 `onOptionsItemSelected()`를 오버라이딩
onCreate 메소드 밖에 커서를 두고, [Generate] > [Override Methods] > 메소드 선택
혹은 Ctrl + O

```
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
  
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
  
    switch (item.getItemId()) {  
        case R.id.itemRed:  
            baseLayout.setBackgroundColor(Color.RED);  
            return true;  
        case R.id.itemGreen:  
            baseLayout.setBackgroundColor(Color.GREEN);  
            return true;  
        case R.id.itemBlue:  
            baseLayout.setBackgroundColor(Color.BLUE);  
            return true;  
        case R.id.subRotate:  
            btnChange.setRotation(45);  
            return true;  
        case R.id.subSize:  
            btnChange.setScaleX(2);  
            return true;  
    }  
    return false; defalut(정상동작을 막을 경우 true)  
}
```

1. 메뉴 | 옵션메뉴

예제7-6: Ex07_06

2) xml 없이 클래스로 생성

menu1.xml 삭제 후, onCreateOptionsMenu() 메소드 수정

`Menu.add(int groupId, int itemId, int order, CharSequence charSequence)`

메소드로 메뉴 항목을 추가

SubMenu 객체로 서브 메뉴 및 항목 추가(순서는 연속적)

```
27
28
29
30
31 menu.add(0,1,0,"배경색(빨강)");
32 menu.add(0,2,0,"배경색(초록)"); order를 3,2,1 순으로 변경해보자.
33 menu.add(0,3,0,"배경색(파랑)");
34
35 SubMenu subMenu = menu.addSubMenu("버튼 변경 >>");
36 subMenu.add(0,4,0,"버튼 45도 회전");
37 subMenu.add(0,5,0,"버튼 2배 확대");
38
39 return true;
40 }
```

1. 메뉴 | 옵션메뉴

onOptionsItemSelected() 메소드 수정

```
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62
```

```
@Override  
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case 1: item id 사용  
            baseLayout.setBackgroundColor(Color.RED);  
            return true;  
        case 2:  
            baseLayout.setBackgroundColor(Color.GREEN);  
            return true;  
        case 3:  
            baseLayout.setBackgroundColor(Color.BLUE);  
            return true;  
        case 4:  
            button1.setRotation(45);  
            return true;  
        case 5:  
            button1.setScaleX(2);  
            return true;  
    }  
    return false;  
}
```


1. 메뉴 | 옵션메뉴

실습(직접 풀어보기)7-1: Test07_01

▶ 직접 풀어보기 7-1

[실습 7-1]을 다음과 같이 변경하자.

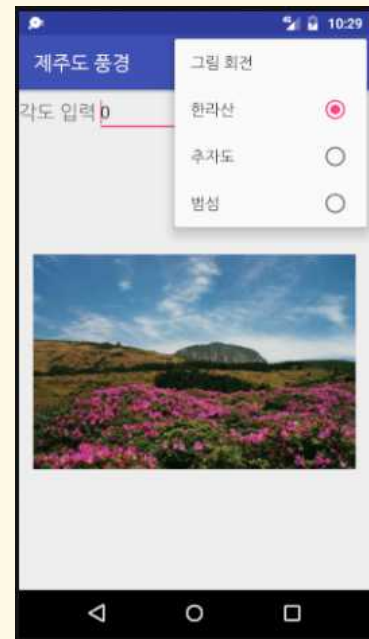
- 레이아웃은 RelativeLayout을 사용하고 텍스트뷰, 에디트텍스트, 이미지뷰 3개를 적절히 배치한다.
- 에디트텍스트에 각도를 입력하고 옵션 메뉴의 [그림 회전]을 선택하면 해당 각도만큼 이미지뷰가 회전된다.
- 한라산/추자도/범섬 옵션 메뉴는 라디오버튼과 같이 3개 중 하나만 체크되도록 하고, 선택하면 이미지뷰가 해당 이미지로 바뀌도록 한다(이미지나 이미지 이름은 아무거나 사용해도 된다).

HINT 1 메뉴 XML에서 여러 항목을 묶으려면 <group> ~ </group>을 이용하면 된다. 이 예는 다음과 같다.

```
<group
    android:checkableBehavior="single">
    <item
        android:id="@+id/item1"
        android:title="한라산"
        android:checked="true">
    </item>
    ~~~ 중간 생략 ~~~
</group>
```

라디오 버튼으로 구현하기 위해
(생략시 일반 메뉴로 구현됨)

HINT 2 이미지뷰의 이미지를 변경하려면 setImageResource() 메소드를 사용하면 된다.



[실행]

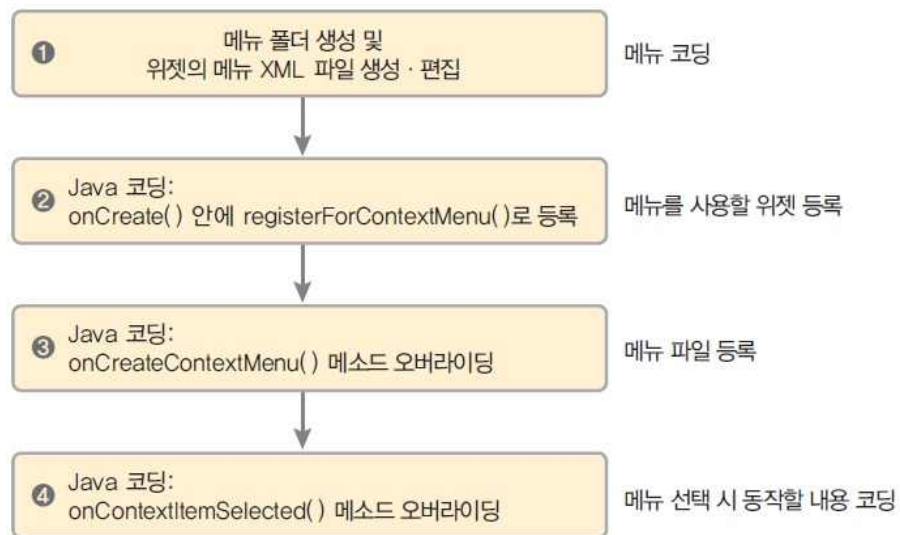
메뉴에서 이미지 선택 >
각도 입력 후 '그림 회전' >
해당 그림 바뀜

1. 메뉴 | 컨텍스트메뉴

컨텍스트 메뉴(ContextMenu)

옵션메뉴와 달리, 레이아웃 또는 버튼, 에디트텍스트 등의 위젯별로 구현하는 메뉴
롱클릭하면 나타남

컨텍스트 메뉴 설정 순서



`onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo)`

클릭된 item(항목)에 대한 추가 정보

1. 메뉴 | 컨텍스트메뉴

컨텍스트 메뉴

컨텍스트 메뉴가 나오게 할 위젯마다 메뉴를 등록하고 인플레이팅

1) onCreate()에서 컨텍스트 메뉴를 설정할 위젯에 메뉴를 등록

registerForContextMenu(View view);

2) onCreateContextMenu()를 오버라이딩하여 위젯별 컨텍스트 메뉴를 등록

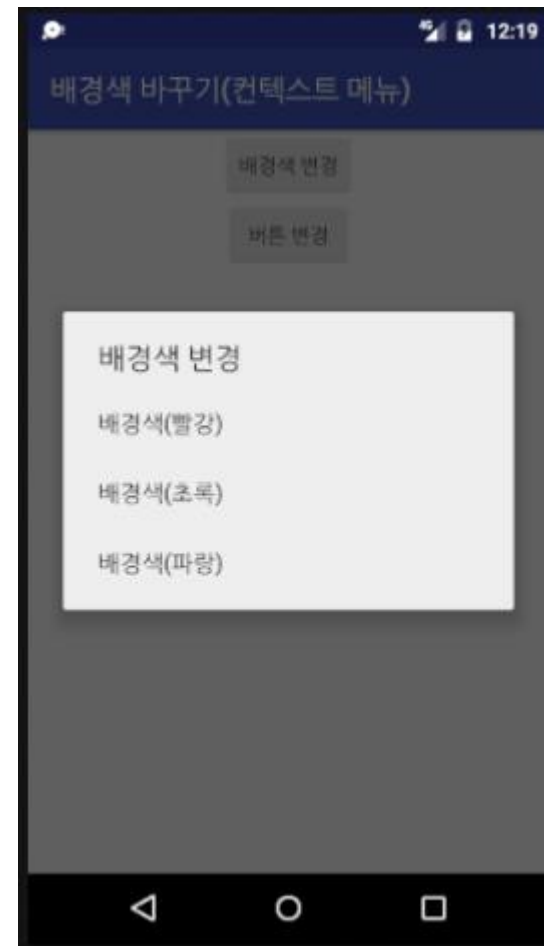
```
public void onCreateContextMenu(ContextMenu menu,
    View v, ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);

    MenuInflater mInflater = getMenuInflater();
    if(v == 위젯1) {
        mInflater.inflate(R.menu.첫번째메뉴XML파일, menu);
    }
    if (v == 위젯2) {
        mInflater.inflate(R.menu.두번째메뉴XML파일, menu);
    }
}
```

3) onOptionsItemSelected(Menuitem item)를 오버라이딩하여 메뉴의 항목별 이벤트를 처리

1. 메뉴 | 컨텍스트메뉴

예제7-2: Ex07_07
버튼 별 컨텍스트 메뉴 구현



1. 메뉴 | 컨텍스트메뉴

activity_main.xml

레이아웃의 id는 baseLayout으로 부여

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3   android:id="@+id/baseLayout"
4   android:layout_width="match_parent"
5   android:layout_height="match_parent"
6   android:orientation="vertical"
7   android:gravity="center_horizontal">
8
9   <Button
10     android:layout_width="wrap_content"
11     android:layout_height="wrap_content"
12     android:id="@+id/button1"
13     android:text="배경색 변경" />
14
15   <Button
16     android:layout_width="wrap_content"
17     android:layout_height="wrap_content"
18     android:id="@+id/button2"
19     android:text="버튼 변경" />
20 </LinearLayout>
```

1. 메뉴 | 컨텍스트메뉴

activity_main.xml

menu1.xml, menu2.xml 생성 및 디자인

menu1.xml: 배경색 변경과 관련된 메뉴아이템

menu2.xml: 버튼 제어와 관련된 2개의 메뉴아이템

Menu1.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <menu xmlns:android="http://schemas.android.com/apk/res/android">
3     <item
4         android:id="@+id/itemRed"
5         android:title="배경색(빨강)" />
6     <item
7         android:id="@+id/itemGreen"
8         android:title="배경색(초록)" />
9     <item
10        android:id="@+id/itemBlue"
11        android:title="배경색(파랑)" />
12 </menu>
```

Menu2.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <menu xmlns:android="http://schemas.android.com/apk/res/android">
3
4     <item
5         android:id="@+id/subRotate"
6         android:title="버튼 45도 회전" />
7
8     <item
9         android:id="@+id/subSize"
10        android:title="버튼 2개 확대" />
11 </menu>
```


1. 메뉴 | 컨텍스트메뉴

MainActivity.java

registerForContextMenu()로 버튼에 컨텍스트 메뉴를 등록

```
13 public class MainActivity extends AppCompatActivity {  
14  
15     LinearLayout baseLayout;  
16     Button button1, button2;  
17  
18     @Override  
19     protected void onCreate(Bundle savedInstanceState) {  
20         super.onCreate(savedInstanceState);  
21         setContentView(R.layout.activity_main);  
22  
23         baseLayout = (LinearLayout) findViewById(R.id.baseLayout);  
24         button1 = (Button) findViewById(R.id.button1);  
25         button2 = (Button) findViewById(R.id.button2);  
26  
27         setTitle("배경색 바꾸기(컨텍스트 메뉴)");  
28  
29         registerForContextMenu(button1);  
30         registerForContextMenu(button2);  
31     }  
32
```

1. 메뉴 | 컨텍스트메뉴

onCreateContextMenu 메소드를 오버라이딩하여 ContextMenu 생성

```
33  @Override
34  public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.ContextMenuInfo menuInfo) {
35      super.onCreateContextMenu(menu, v, menuInfo);
36
37      MenuInflater mInflater = getMenuInflater();
38
39      if (v==button1) {
40          menu.setHeaderTitle("배경색 변경");
41          mInflater.inflate(R.menu.menu1, menu);
42      }
43      if (v==button2) {
44          menu.setHeaderTitle("버튼 변경");
45          mInflater.inflate(R.menu.menu2, menu);
46      }
47  }
```

컨텍스트 메뉴의 제목을 설정하고 인플레이션

1. 메뉴 | 컨텍스트메뉴

옵션메뉴의 onOptionsItemSelected 메소드 처리와 동일하니
onContextItemSelected 메소드 오버라이딩을 완성해보자!

```
49      @Override
50      public boolean onOptionsItemSelected(MenuItem item) {
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69      }
70  }
```

완성 후 MarshMallow에서
실행하기

1. 메뉴 | 컨텍스트메뉴

실습(직접 풀어보기) 7-2: Test07_02

▶ 직접 풀어보기 7-2

[실습 7-2]에서 실습한 컨텍스트 메뉴를 메뉴 XML 파일 없이 Java 코드로만 완성하자. [예제 7-6]을 참조한다.