



Option 메뉴 실습

배 희호 교수
경북대학교
스마트IT과



Menu



■ OptionMenu(옵션 메뉴)

- KeyPad의 Menu Button을 눌렀을 때 일반적으로 화면 상단이나 하단에 나오는 Menu
- 화면 상단에 사용자가 선택할 수 있는 Menu들이 표시되며, 화면을 초과할 경우 Scroll로 처리할 수 있음

■ ContextMenu(컨텍스트 메뉴)

- 화면을 오래 누르고 있다거나 하는 동작으로 인해 화면 정중앙에 나타나는 Menu
- 보통 Menu 창 맨 위에 제목을 표시하고 그 아래에 메뉴가 표시



Option Menu





Option Menu

- Activity를 위한 Menu Item의 주요한 모임으로 사용자가 단말기 또는 Emulator의 [Menu] 버튼을 누르면 화면에 표시
- 기본적으로 6개의 Item이 표시되는데 **실제적으로 5개의 메뉴만 표시되고** 이후의 메뉴는 6번째에 위치한 [더 보기(More)]를 누르면 표시



Option Menu



- Menu Key 클릭 시 화면 하단이나 상단에 나타나는 Menu
 - Icon Menu(아이콘 메뉴)
 - 메뉴 키를 눌렀을 때 화면 하단에 처음 나타나는 아이콘 메뉴 항목
 - Extended Menu(확장 메뉴)
 - 메뉴 항목이 6개 초과시 나머지 항목 들을 표시하기 위해 사용
 - Submenu(부 메뉴)
 - Menu Item이 또 다른 Menu Item들로 구성
- Option Menu 생성자
 - onCreateOptionsMenu(Menu menu)



Option Menu



- Option Menu를 액션 바(Action Bar)메뉴로 사용
- Android 폰 중에서 하드웨어 키로 Menu나 Home 버튼이 있는 경우는 여전히 아래에서 옵션 메뉴가 나타남
- 반면에 소프트 키로 menu나 Home 버튼이 있는 일부 기기 (LG g3 등)에서는 Action Bar에서 옵션 메뉴가 아래로 펼쳐짐
- 물론 하드웨어 키가 있는 기기에서도 옵션 메뉴를 눌렀을 때 Action Bar에서 나타나게 할 수 있음
- Android에서 옵션 메뉴를 추가하기 위해서는 onCreateOptionsMenu() 메소드를 Override 해야 함
- 기본적으로 프로젝트를 생성하게 되면 추가가 되지 않는다면 Override/Implement methods 대화상자에서 추가하면 됨



Option Menu



■ Option Menu의 정체

- Option Menu는 Activity의 재정의 함수에 의존적으로 동작
- 따라서 마치 Android의 특별한 기능처럼 보이기도 함
- 하지만 내부적으로는 AlertDialog를 이용한 Dialog에 불과하다. 즉 개발자가 AlertDialog 혹은 직접 Dialog를 상속받아 구현할 수도 있다는 것
- Android에서 Activity를 통해 Option Menu를 제공한 이유는 단지, 화면의 일관성과 더 쉽고 편리하게 Menu를 제공하기 위함



Option Menu

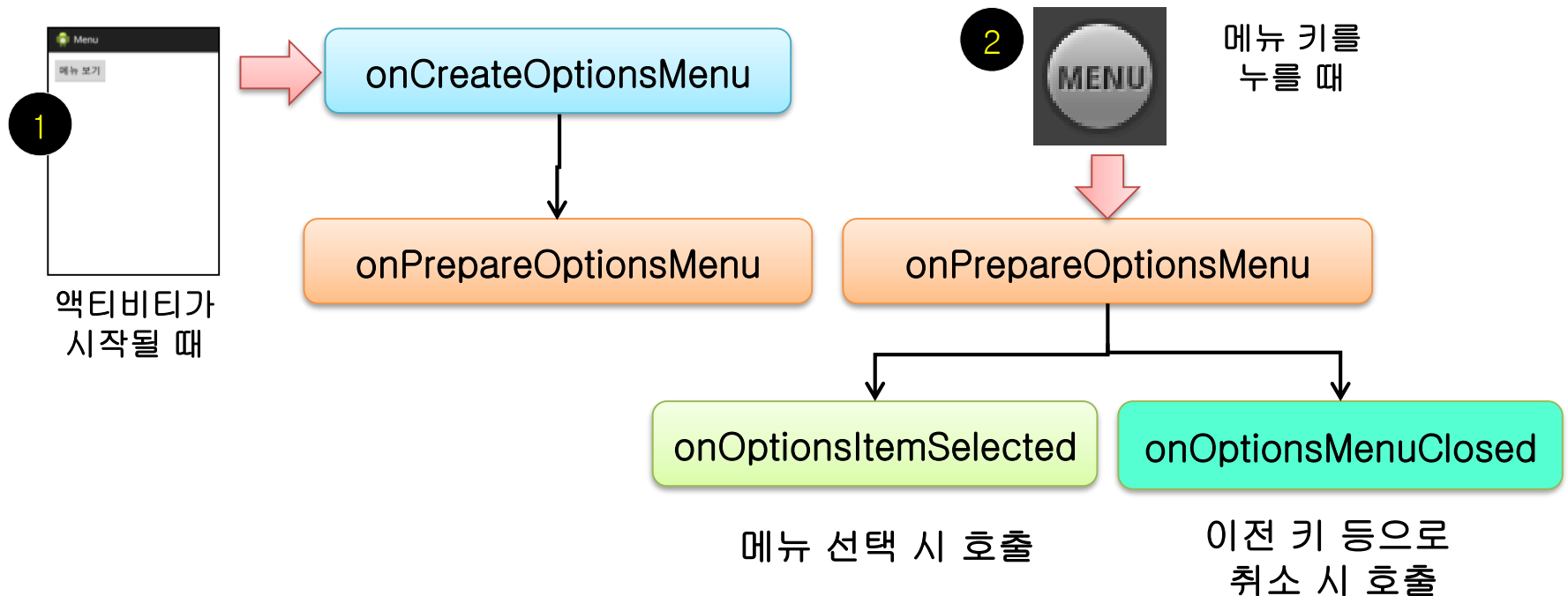
■ 메소드

메소드	설명
onCreateOptionsMenu()	<ul style="list-style-type: none">✓ 옵션메뉴 추가✓ onCreateOptionsMenu가 최초로 생성될 때만 호출되는 메소드✓ 메뉴 생성 이후에는 onCreateOptionsMenu가 화면에 나타날 때 onPrepareOptionsMenu()메소드가 호출✓ Activity가 onDestroy()될 때 onCreateOptionsMenu도 Destroy
onOptionsItemSelected()	<ul style="list-style-type: none">✓ 메뉴의 아이템 클릭에 따른 이벤트 발생에 대한 처리✓ onOptionsItemSelected 아이템 선택 시 자동 호출되는 메소드
onPrepareOptionsMenu()	<ul style="list-style-type: none">✓ 화면의 메뉴 변경 메소드✓ 옵션메뉴가 화면에 나타날 때마다 호출되는 메소드✓ 스크린 화면에 맞는 onOptionsItemSelected를 화면에 표시하기 위해 호출되는 메소드
onOptionsMenuClosed()	<ul style="list-style-type: none">✓ onOptionsItemSelected가 종료될 때 호출
openOptionsMenu()	<ul style="list-style-type: none">✓ onOptionsItemSelected가 강제로 Open될 때 호출
closeOptionsMenu()	<ul style="list-style-type: none">✓ onOptionsItemSelected가 강제로 Close될 때 호출



Option Menu

- Option Menu는 아래와 같이 액티비티의 4 가지 재정의의 메소드를 통해 제어





Option Menu



- boolean onCreateOptionsMenu(Menu menu)
 - 이 메소드는 Option Menu를 초기화 함
 - Option Menu를 맨 처음 띄울 때 호출
 - menu 파라미터의 add() 메소드를 호출하여 Menu Item을 추가
 - menu는 Option Menu를 의미하며, 그것의 add() 메소드는 옵션 메뉴에 항목을 추가
 - 이 메소드가 반환해야 할 값은 다음과 같음
 - true : 옵션 메뉴를 보여주려면 true를 반환
 - false : 옵션 메뉴를 보여주지 않으려면 false를 반환



Option Menu



- boolean onPrepareOptionsMenu(Menu menu)
 - 이 메소드는 옵션 메뉴를 띄울 때마다 호출
 - 초기화 된 옵션 메뉴를 변경할 수 있음
 - 만약 메뉴의 내용을 변경하길 원한다면, 이 메소드에 그것을 구현해 주면 됨. 가능한 변경 유형은 다음과 같음
 - 조건에 따라 메뉴 항목을 비활성화
 - 조건에 따라 메뉴 항목을 추가
 - 조건에 따라 메뉴 항목을 제외
 - 반환값 선택은 onCreateOptionsMenu()와 동일



Option Menu



- boolean onOptionsItemSelected(MenuItem item)
 - 이 메소드는 사용자가 선택한 메뉴 항목을 처리함
 - 선택된 메뉴 항목이 무엇인지 구별
 - item 파라미터는 선택된 항목을 의미. 그것의 getTitle() 메소드를 통해 항목의 이름이 무엇인지 파악할 수 있음
 - 항목의 ID로 구별하는 방법도 있음
 - 선택된 메뉴 항목을 처리
 - 반환값
 - true - 항목 클릭을 이 메소드가 처리하려면 true를 선택
 - false - 일반 메뉴 처리를 진행할 수 있도록 false를 반환



Option Menu



- onOptionsItemSelected(Menu item)
 - Option Menu를 클릭하여 활성화 된 상태에서 이전 버튼을 클릭하거나 액티비티의 다른 영역을 클릭하여 옵션 메뉴를 닫을 때 호출되는 메소드



Option Menu

- Menu 객체는 아래와 같이 옵션 메뉴의 항목을 추가 할 수 있는 메소드를 제공

일반 메뉴 아이템을 추가

```
public MenuItem add(int groupId1, int itemId2, int order3,  
                    CharSequence title4)
```

하위 메뉴가 존재하는 메뉴 아이템을 추가

```
SubMenu addSubMenu(final int groupId1, final int itemId2, int order3,  
                    final CharSequence title4)
```

- SubMenu 객체를 통해 하위 메뉴 아이템들을 추가할 수 있음

하위 메뉴 아이템 추가(일반 메뉴 아이템 추가와 같은 방식)

```
public MenuItem add(int groupId1, int itemId2, int order3,  
                    CharSequence title4)
```

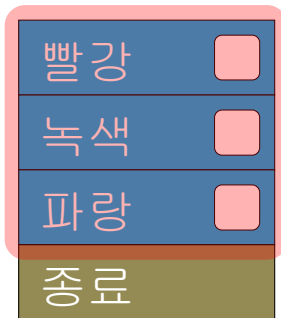
- 모든 메소드의 인자는 동일함



Option Menu

```
public MenuItem add(int groupId1, int itemId2, int order3, CharSequence title4)
```

1 groupId 아이템 그룹 ID



관련된 아이템을 그룹으로 묶기 그룹 ID이며, 이를 이용하여 그룹 관련 처리를 편리하게 할 수 있다.
필요 없다면 0으로 설정

2 itemId 아이템 ID



메뉴 아이템을 구분하기 위한 아이템 ID
특히 특정 아이템이 선택되었을 때, 선택된 아이템을 구분하기 위한 용도로 많이 사용

3 order 아이템 순번



메뉴 아이템이 배치되는 순번
만일 순번을 0으로 설정하면, 아이템을 추가하는 순서대로 배치

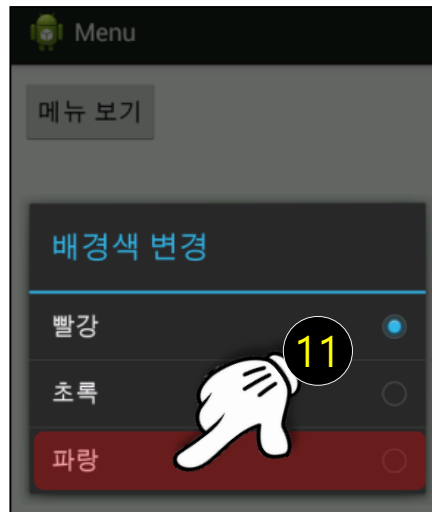
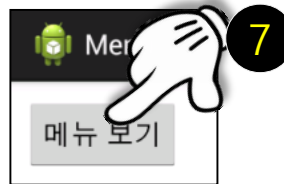
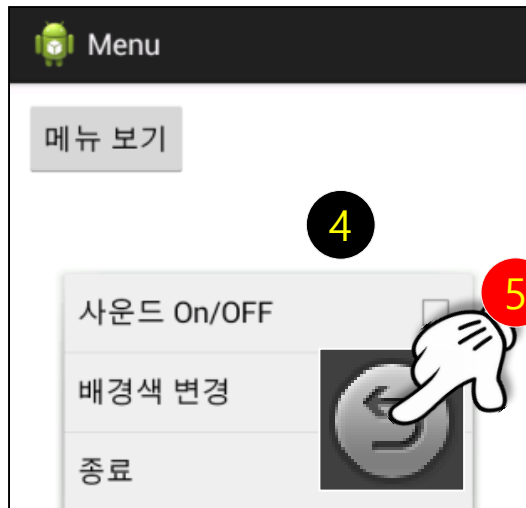
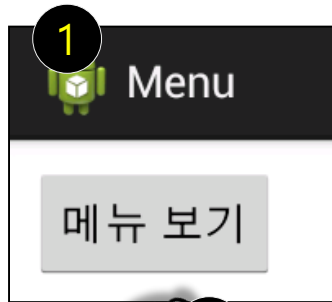
4 title 아이템 명



메뉴 아이템 명을 설정



Option Menu



● 로그

- 2 `onCreateOptionsMenu()`
`onPrepareOptionsMenu()`
- 6 `onOptionsMenuClosed()`
- 8 `onPrepareOptionsMenu()`
- 10 `onOptionsItemSelected()`
- 12 `onOptionsItemSelected()`



Option Menu

- 그런데 일일이 Menu Item을 등록하는 것은 매우 귀찮다. 뿐만 아니라 메뉴 내용이 변경될 때마다 JAVA 소스를 일일이 수정해주어야 한다. 즉 **유지보수 측면에서 불편**
- 이를 위해 Menu Item들을 메뉴 XML 리소스로 구성할 수 있도록 지원



Option Menu



- Menu를 XML 파일로 작성한 Option Menu
 - [res]-[menu] 폴더 생성
 - 메뉴 xml 파일 생성 및 코딩
 - 액티비티(*.java)에 onCreateOptionsMenu() 재정의
 - 메뉴 파일을 자동으로 등록하기 위함
 - 액티비티(*.java)에 onOptionsItemSelected() 재정의
 - 메뉴 선택 시 동작을 위함
- JAVA Code만 이용한 Option Menu
 - onCreateOptionsMenu() 메소드 안에서 Menu XML 파일에 접근하는 대신에 직접 Menu.add() 메소드로 메뉴 항목을 추가



Option Menu

■ item 속성

android:id	✓ 각 메뉴 아이템의 고유한 ID를 설정
android:icon	✓ 메뉴 아이템에 보일 아이콘을 설정
android:title	✓ 메뉴 아이템에 보일 타이틀을 설정
android:showAsAction	✓ 메뉴를 어떤 방식으로 표시할지를 설정
app:orderInCategory	<ul style="list-style-type: none">✓ 여러 액션 아이템이 나열되는 순서를 정의하는 속성✓ 0이 우선순위가 가장 높음✓ 숫자가 낮을 수록 위에 위치하게 됨



Option Menu



■ app:showAsAction 속성값

always	✓ 항상 액션 바에 아이템을 추가하여 표시
never	✓ 기본값 ✓ 액션 바에 아이템을 추가하고 표시하지 않음
ifRoom	✓ 액션 바에 여유공간이 있을 때만 아이템을 표시
withText	✓ 설정된 제목을 함께 표시
collapseActionView	✓ 아이템에 설정한 뷰(actionViewLayout으로 설정한 뷰)의 아이콘만 표시



Option Menu

■ Menu.add() 메소드의 매개변수

Menu.add(groupId, itemId, order, title)

■ groupId

- Menu에서 제공하는 그룹 속성을 이용하기 위해 설정
- Menu:setGroupCheckable(), setGroupEnabled(), setGroupVisible() 등을 적용 가능
- 그룹 지정을 하지 않으려면 pre-define된 상수 Menu:NONE이나 숫자 0을 전달하는 것이 일반적

■ itemId

- Menu에서 관리하는 고유한 item ID

■ order

- item의 순서 (0 ~ 10) 10이 제일 먼저 표시
- 10이상 추가 가능

■ title

- item의 title



Option Menu



■ Option Menu 추가방법

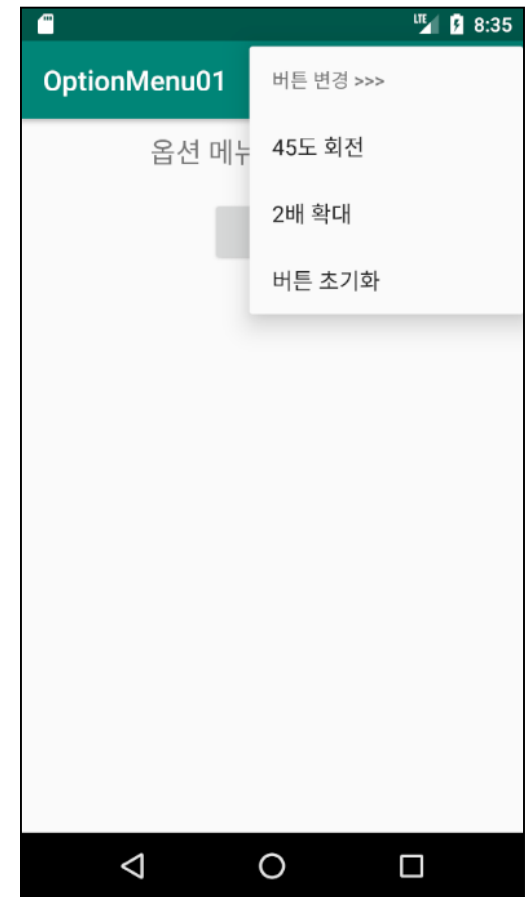
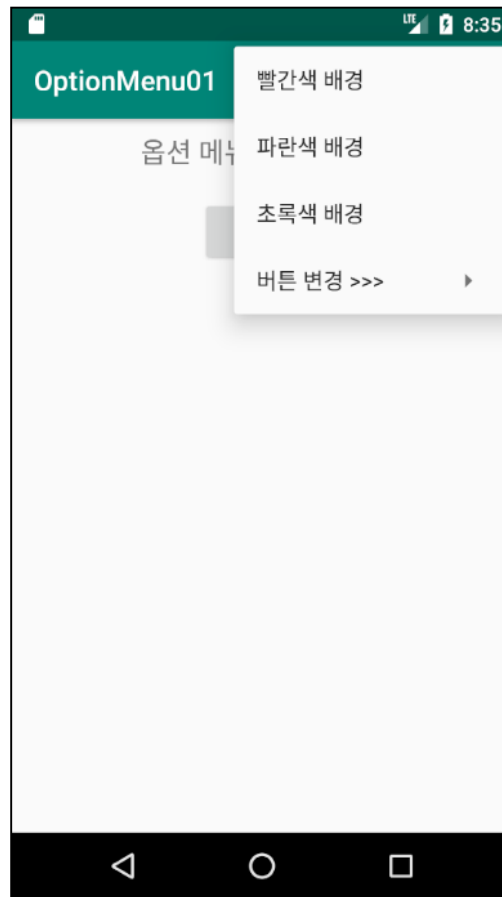
- Option Menu를 추가하기 위해서는 2개의 메소드를 재정 의함으로써 메뉴 아이টে을 추가할 수 있음

단계	설명
1	Menu 폴더 생성 Menu 폴더에 XML 파일 생성
2	onCreateOptionsMenu() 메소드 오버라이딩 메뉴 생성
3	onOptionsItemSelected() 메소드 오버라이딩 메뉴 항목 선택에 대한 이벤트 처리



Option Menu 예제 1

- Option Menu를 클릭하면 Layout의 배경색이 바뀌게 하고 Sub Menu로 버튼을 조작하는 애플리케이션을 만들어 보자





Option Menu 예제 1

■ 메뉴 XML 파일을 이용한 방식

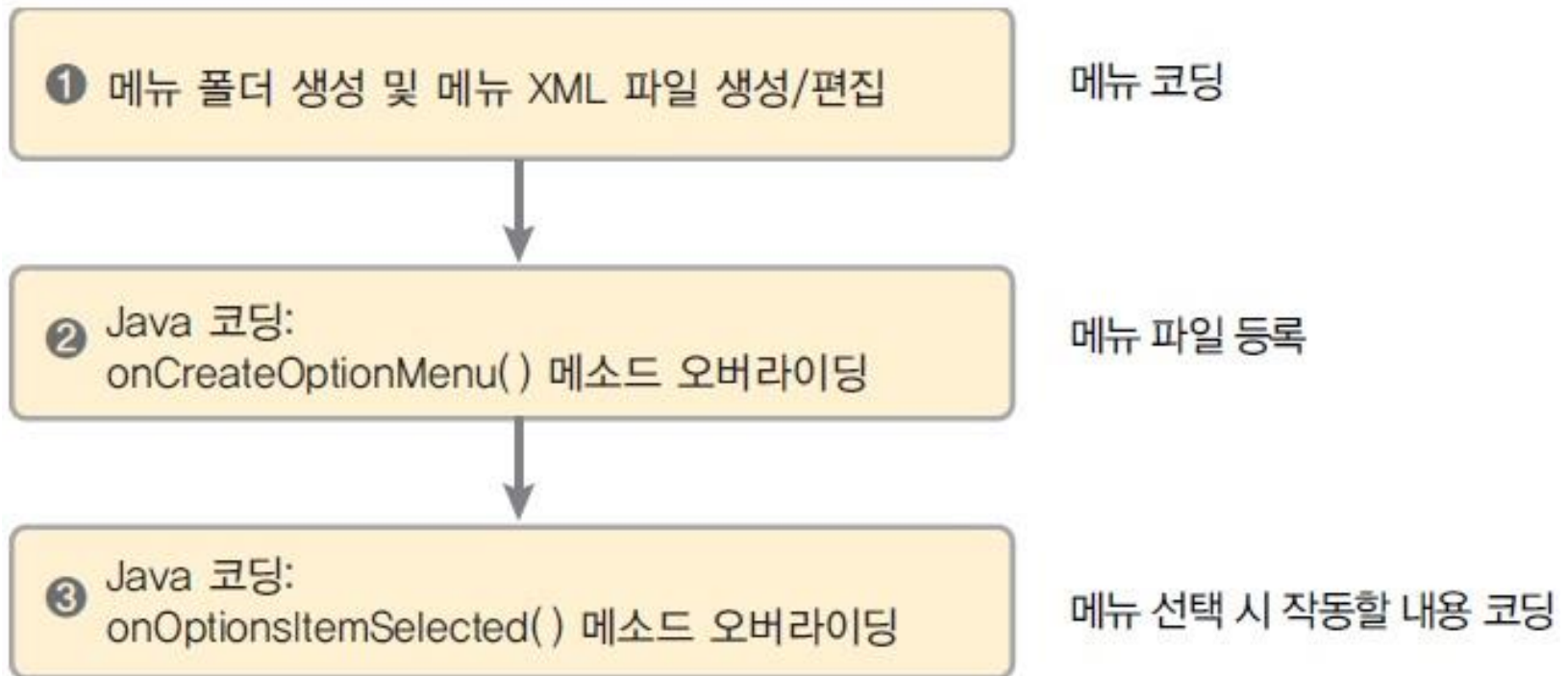


그림 7-2 옵션 메뉴 설정 순서(XML 파일 이용)



Option Menu 예제 1



■ Menu 생성

- onCreateOptionsMenu(Menu menu)
- Menu가 '최초로' 화면에 표시되기 전에 호출
- Menu 항목을 만들어주는 역할을 함

■ Menu 준비

- onPrepareOptionsMenu(Menu menu)
- 현재 상태에 따라 특정 메뉴 항목을 활성화/비활성화시켜야 할 때 사용
- 예) 메시지가 없을 때는 '메시지 전체 삭제' 메뉴가 비활성화되어야 함
- 여기서는 상태에 따라 동적으로 메뉴 항목을 영향을 주는 작업을 구현



Option Menu 예제 1



■ Menu 표시

- 앞의 과정들을 통해 실제로 메뉴가 화면에 표시되고, 사용자가 메뉴를 선택하여 원하는 동작을 수행할 수 있게 됨

■ Menu 선택 처리

- Menu가 제 기능을 발휘하려면 메뉴 항목을 선택했을 때의 동작을 구현해야 함
- Option Menu의 항목 선택 처리는
onOptionsItemSelected(MenuItem) 메소드에서 담당
- 인자로 넘겨받은 MenuItem 인스턴스를 통해 사용자가 어떤 메뉴를 선택했는지 알아내고, 각 항목에 알맞은 동작을 구현



Option Menu 예제 1



- 응용 실행 도중 메뉴 버튼(“Menu”)을 누르면 나타나는 응용 고유의 메뉴
- 응용에 대한 추가적인 제어나 기타 설정 수단으로 쓰임
- Option Menu 생성
 - Activity 클래스의 onCreateOptionsMenu() 메소드 재정의
 - onCreateOptionsMenu 파라미터 menu에 원하는 만큼의 옵션 메뉴(메뉴 이름, 아이콘, 인텐트)를 추가
- Menu 선택 시 실행
 - 각 메뉴 항목에 이미 인텐트를 지정하였으므로 자동으로 실행 됨
 - 추가적으로 수행하고자 하는 코드는 onOptionsItemSelected() 메소드 재정의



Option Menu 예제 1

■ activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/baseLayout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="10dp"
    tools:context="com.example.bae.optionmenu0.MainActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:textSize="20dp"
        android:text="옵션 메뉴를 누르세요"/>
```



Option Menu 예제 1

■ activity_main.xml

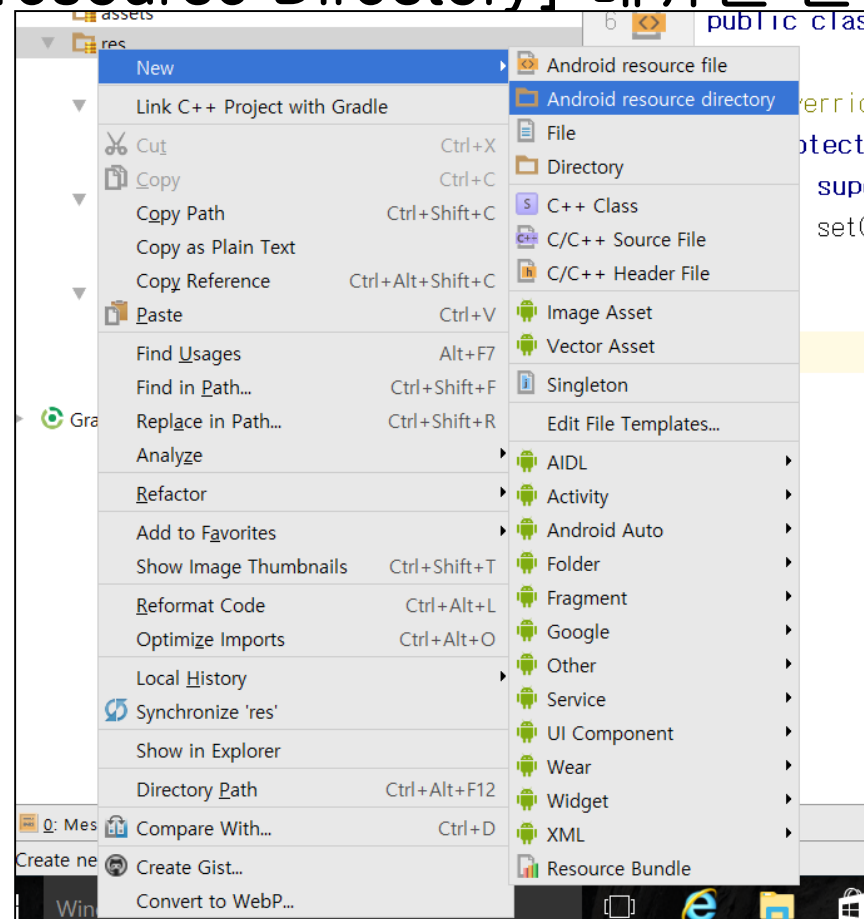
```
<Button
    android:id="@+id/button"
    android:layout_marginTop="20dp"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="버튼"/>
</LinearLayout>
```



Option Menu 예제 1 (XML 사용)

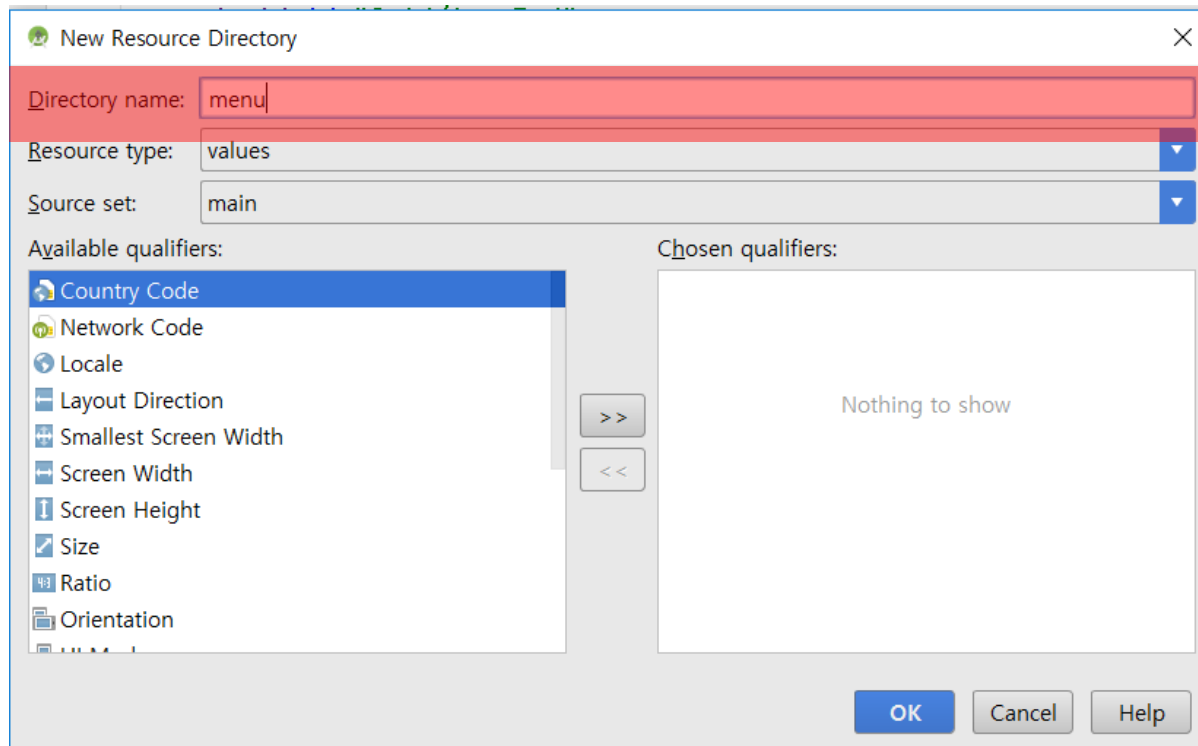
- [res]-[menu] 폴더 생성

- [res] 폴더에서 마우스 오른쪽 버튼을 클릭한 뒤, [new]-[Android resource Directory] 메뉴를 클릭



Option Menu 예제 1 (XML 사용)

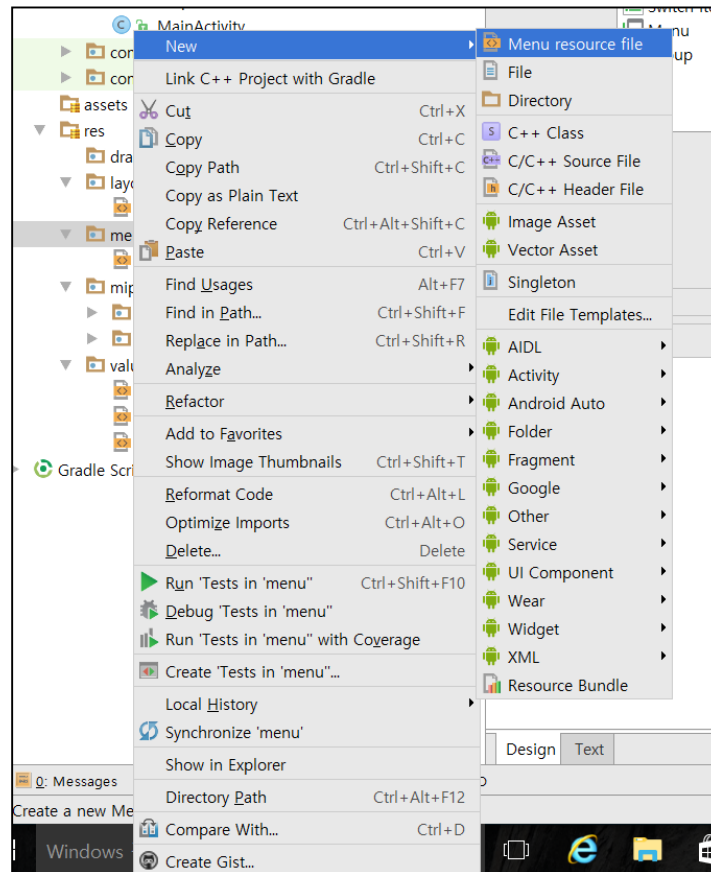
- [res]-[menu] 폴더 생성
 - menu라는 폴더를 만듦
 - 그림과 같이 [res]폴더의 하위 폴더 [menu] 폴더가 생성





Option Menu 예제 1 (XML 사용)

- 메뉴 XML 파일 생성 및 코딩
 - [menu] 폴더에서 마우스 오른쪽 버튼을 클릭한 뒤, [new]-[Menu resource file] 메뉴를 클릭





Option Menu 예제 1 (XML 사용)

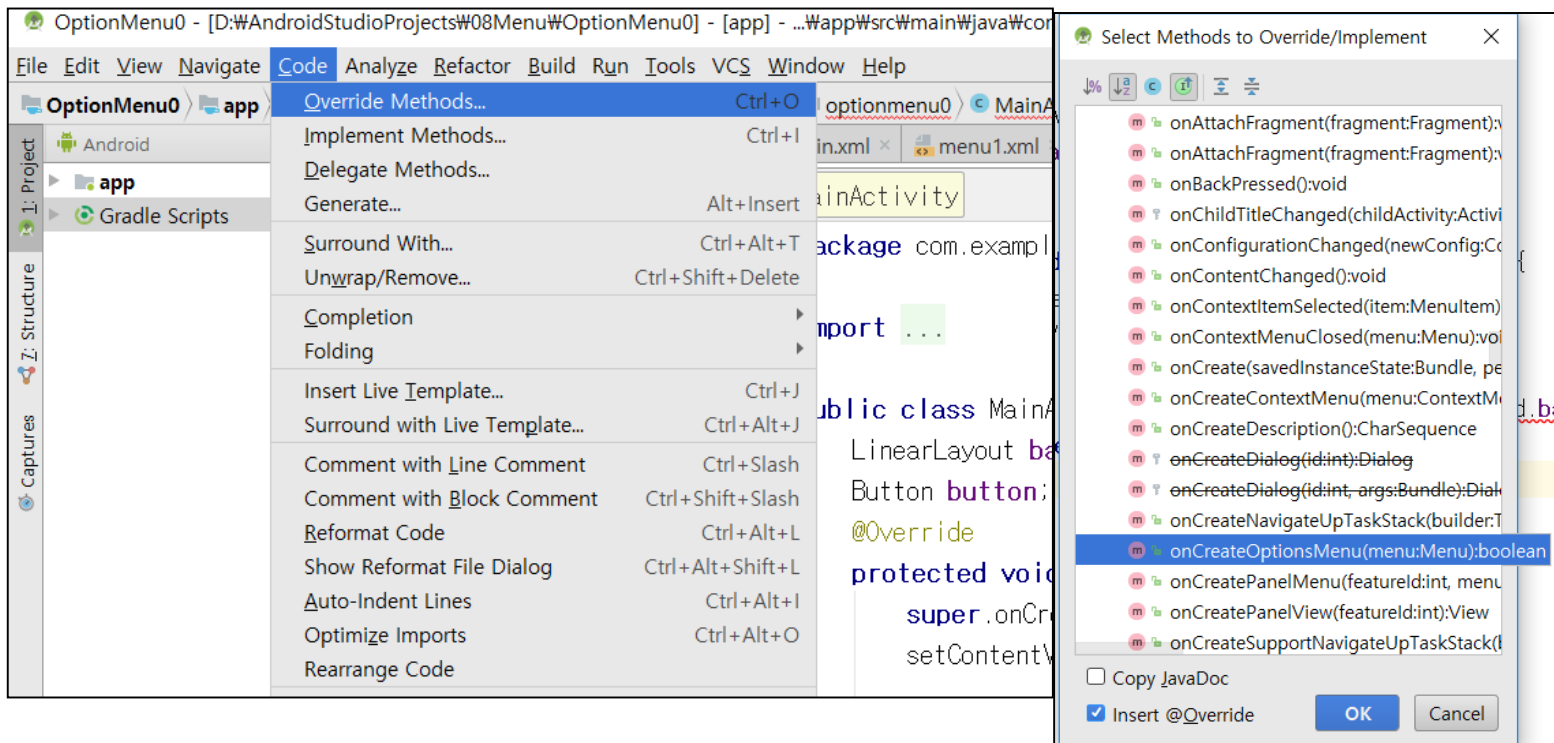
■ 메뉴 XML 파일 생성 및 코딩

```
<menu>
  <item
    android:id="@+id/subrotate"
    android:title="45도 회전" />
  <item
    android:id="@+id/subsize"
    android:title="2배 확대" />
  <item
    android:id="@+id/subNormal"
    android:title="버튼 초기화" />
</menu>
</item>
</menu>
```



Option Menu 예제 1 (XML 사용)

- Android Studio 상단의 메뉴에서 [Code]→[Override Methods...]를 클릭한 뒤 onCreateOptionsMenu() 메소드를 검색해서 오버라이딩
- 이 메소드는 앱이 실행되면 **메뉴의 내용을 XML 파일에서 자동으로 읽어 옴**





Option Menu 예제 1 (XML 사용)

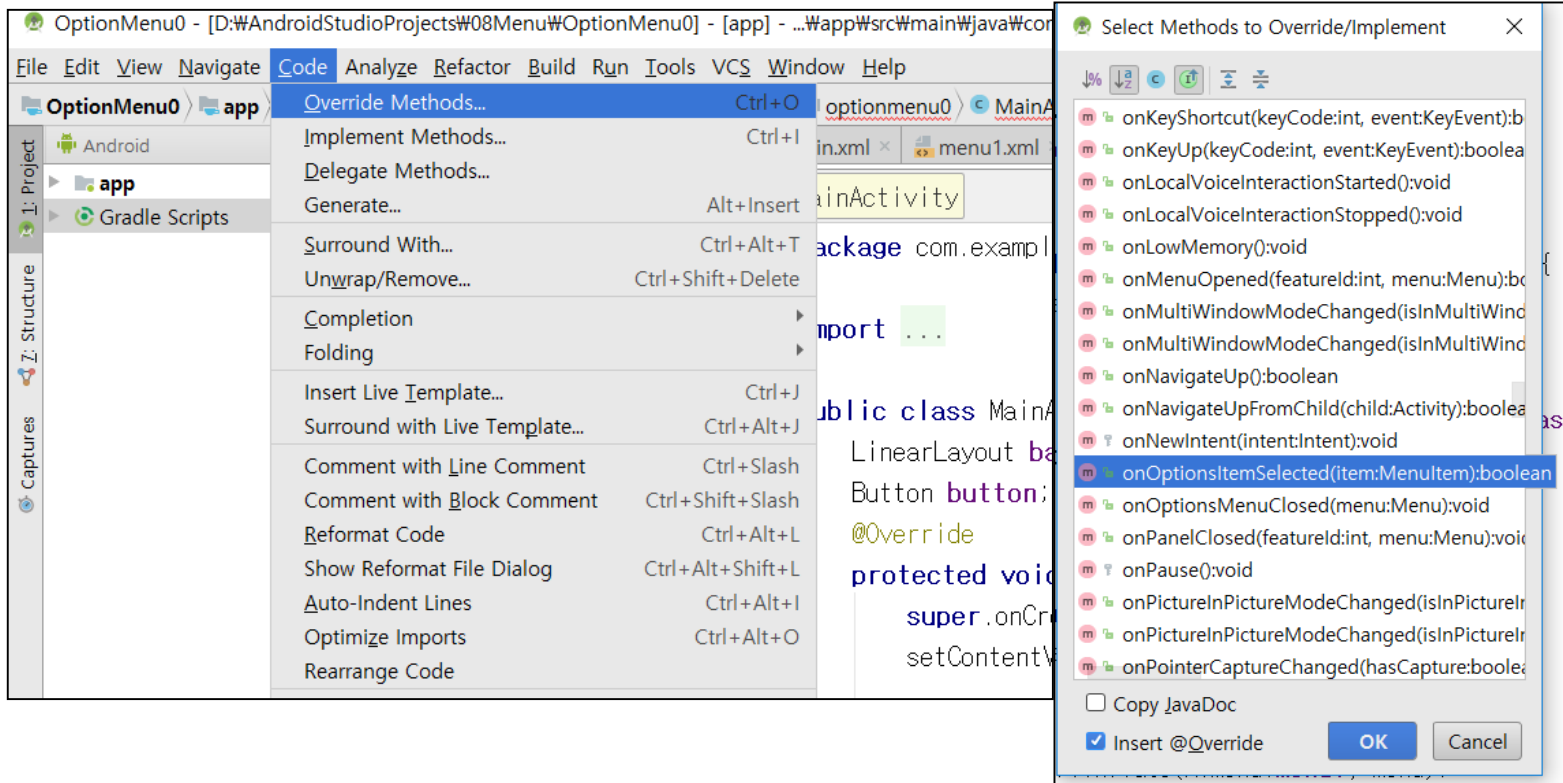
■ MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
    LinearLayout baseLayout;  
    Button button;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        baseLayout = findViewById(R.id.baseLayout);  
        button = findViewById(R.id.button);  
    }  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu) {  
        MenuInflater inflater = getMenuInflater();  
        inflater.inflate(R.menu.menu1, menu);  
        return true;  
    }  
}
```



Option Menu 예제 1 (XML 사용)

- Android Studio 상단의 메뉴에서 [Code]→[Override Methods...]를 클릭한 뒤 **onOptionsItemSelected()** 메소드를 검색해서 오버라이딩





Option Menu 예제 1 (XML 사용)

■ Menu관련 멤버 메소드

메소드	설명
onCreateOptionsMenu()	<ul style="list-style-type: none">• 옵션 메뉴 추가• onCreateOptionsMenu가 최초로 생성될 때만 호출되는 메소드• 메뉴 생성 이후에는 onCreateOptionsMenu가 화면에 나타날 때 onPrepareOptionsMenu() 메소드가 호출• Activity가 onDestroy()될 때 onCreateOptionsMenu도 Destory
onOptionsItemSelected()	<ul style="list-style-type: none">• 메뉴의 아이템 클릭에 따른 이벤트 발생에 대한 처리• onOptionsItemSelected 아이템 선택 시 자동호출되는 메소드
onPrepareOptionsMenu()	<ul style="list-style-type: none">• 화면의 메뉴 변경 메소드• 옵션메뉴가 화면에 나타날 때마다 호출되는 메소드• 스크린화면에 맞는 onOptionsItemSelected를 화면에 표시하기 위해 호출되는 메소드
onOptionsMenuClosed()	<ul style="list-style-type: none">• onOptionsItemSelected가 종료될 때 호출
openOptionsMenu()	<ul style="list-style-type: none">• onOptionsItemSelected가 강제로 Open될 때 호출
closeOptionsMenu()	<ul style="list-style-type: none">• onOptionsItemSelected가 강제로 Close될 때 호출



Option Menu 예제 1 (XML 사용)

■ MainActivity.xml

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch(item.getItemId()){  
        case R.id.itemRed :  
            baseLayout.setBackgroundColor(Color.RED);  
            return true;  
        case R.id.itemBlue :  
            baseLayout.setBackgroundColor(Color.BLUE);  
            return true;  
        case R.id.itemGreen :  
            baseLayout.setBackgroundColor(Color.GREEN);  
            return true;  
        case R.id.subrotate :  
            button.setRotation(45);  
            return true;  
    }  
}
```



Option Menu 예제 1 (XML 사용)

■ MainActivity.xml

```
case R.id.subsize :  
    button.setScaleX(2);  
    return true;  
case R.id.subNormal :  
    button.setScaleX(1);  
    button.setRotation(0);  
    return true;  
}  
return false;  
}
```



Option Menu 예제 1



- 메뉴 작성하기 - 차이의 이유
 - Android 버전 2.3 이하
 - 하단의 사각형 메뉴를 사용함
 - Android 버전 3.0 이상 (현재 모습)
 - Action Bar가 추가됨
 - 따라서 메뉴가 상위에 배치됨
- 상위 버전에서도 하단메뉴를 배치하려면?
 - Theme을 구 버전으로 바꿈
 - manifest 파일에서 target SDK 버전을 적절히 낮춤 (API Level 8 이하)



Option Menu 예제 1 (JAVA 코드)

- JAVA 코드만 이용한 옵션 메뉴
 - onCreateOptionsMenu() 메소드 안에서 메뉴 XML 파일에 접근하는 대신에 직접 Menu.add() 메소드로 메뉴 항목을 추가

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {  
    menu.add(0,1,0, "빨간색 배경");  
    menu.add(0,2,0, "파란색 배경");  
    menu.add(0,3,0, "초록색 배경");  
    SubMenu subMenu = menu.addSubMenu("버튼 변경 >>>");  
    subMenu.add(0, 4, 0, "45도 회전");  
    subMenu.add(0, 5,0, "2배 확대");  
    subMenu.add(0, 6, 0,"버튼 초기화");  
    return true;  
}
```



Option Menu 예제 1 (JAVA 코드)

- 유동적으로 JAVA 코드를 이용해서 메뉴를 추가해야 하는 경우 add() 메소드를 사용

```
Menu.add(groupId, itemId, order, title)
```

- groupId
 - 그룹 ID를 지정하며, Menu에서 사용할 수 있는 그룹 옵션을 사용할 때 사용
- itemId
 - Menu 각각의 item의 ID를 지정
- order
 - item의 순서이며, android:orderInCategory와 같음
- title
 - item의 title



Option Menu 예제 1 (JAVA 코드)

- SubMenu란 옵션 메뉴에서 하나의 Menu item을 그 하위의 sub item들과 묶어서 하나의 Group 형태로 활용하는 방법
- SubMenu를 설정하고 SubMenu가 선택이 되었을 때는 안드로이드 ContextMenu와 동일한 형태의 팝업 창을 보여 줌



Option Menu 예제 1 (JAVA 코드)

■ JAVA 코드만 이용한 옵션 메뉴

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch(item.getItemId()){  
        case 1 :  
            baseLayout.setBackgroundColor(Color.RED);  
            return true;  
        case 2 :  
            baseLayout.setBackgroundColor(Color.BLUE);  
            return true;  
        case 3 :  
            baseLayout.setBackgroundColor(Color.GREEN);  
            return true;  
        case 4 :  
            button.setRotation(45);  
            return true;  
    }  
}
```



Option Menu 예제 1 (JAVA 코드)

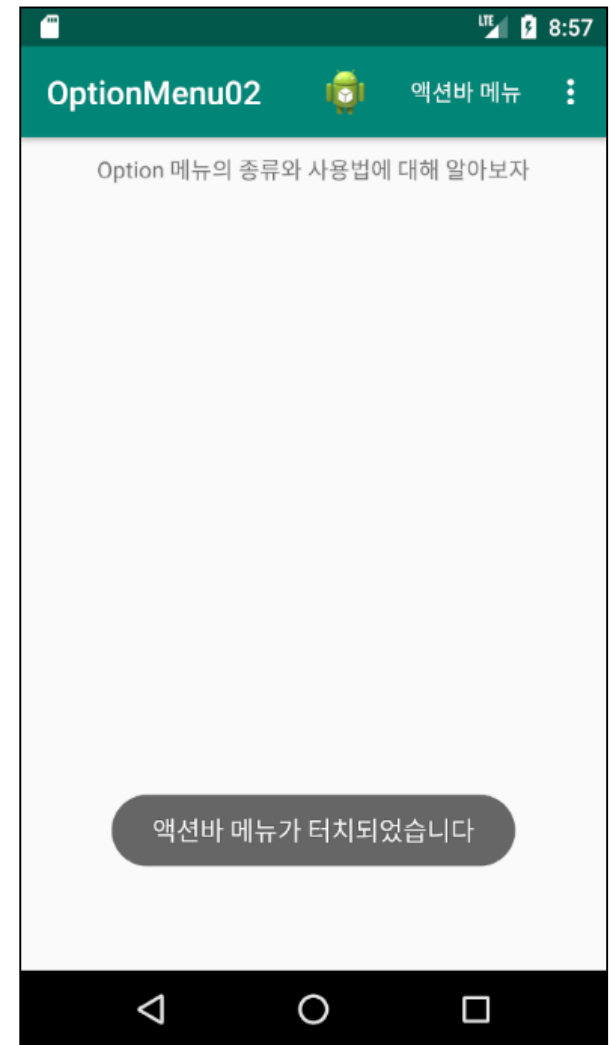
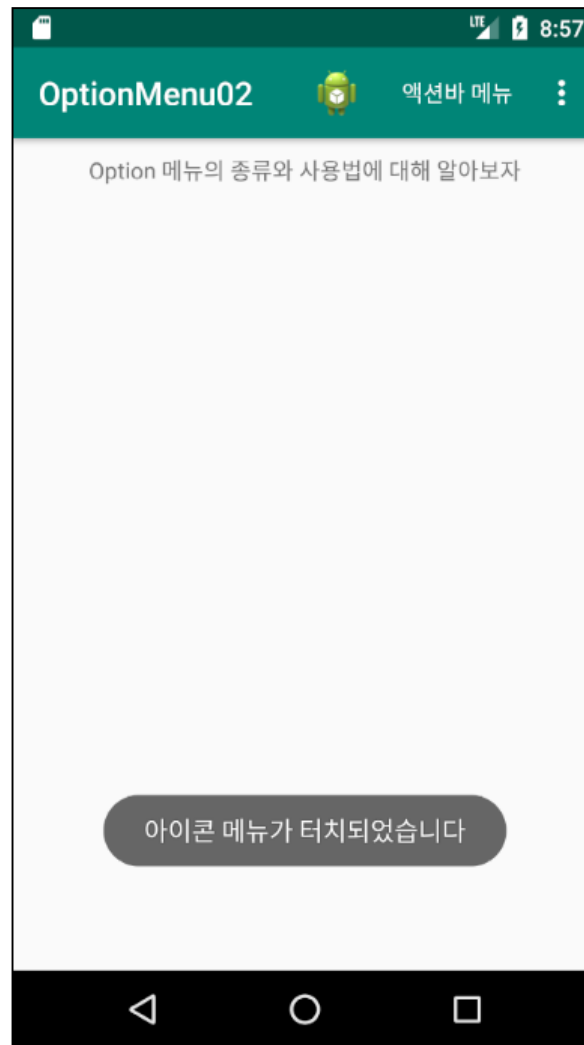
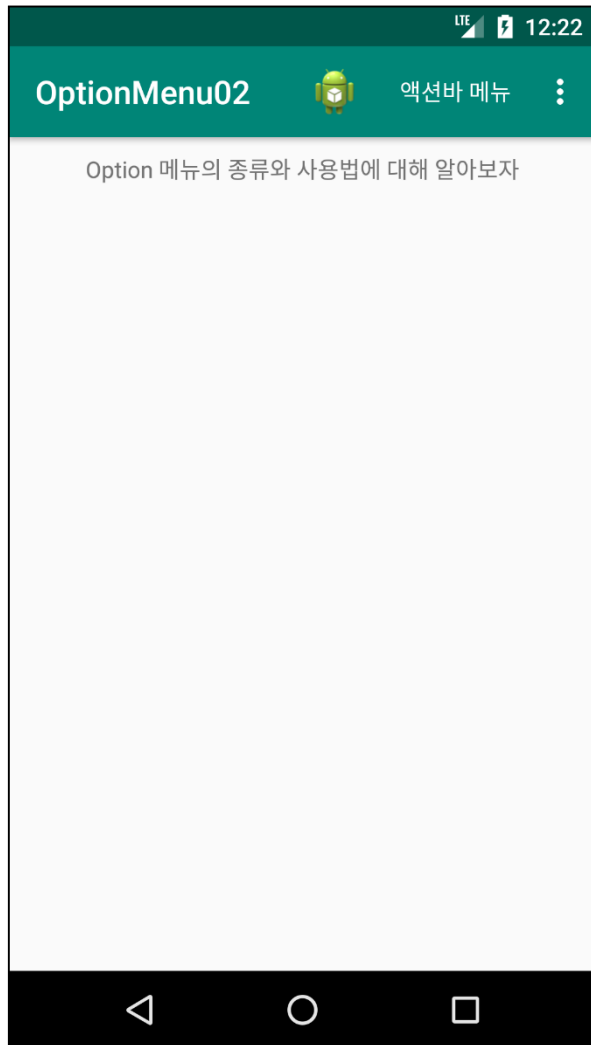
■ JAVA 코드만 이용한 옵션 메뉴

```
case 5 :  
    button.setScaleX(2);  
    return true;  
case 6 :  
    button.setScaleX(1);  
    button.setRotation(0);  
    return true;  
}  
return false;  
}
```



Option Menu 예제 2

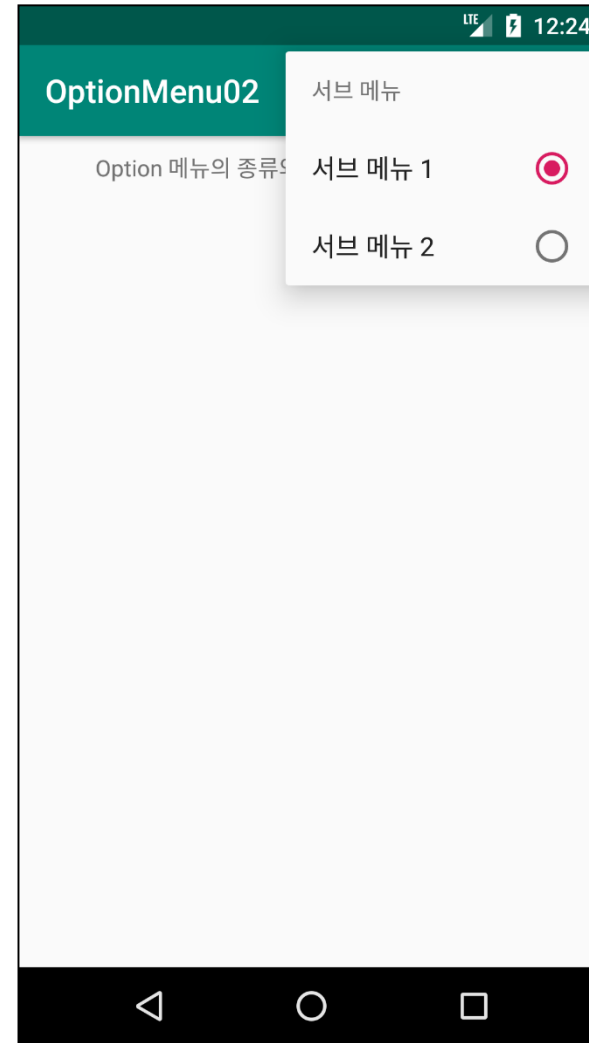
■ 다음과 같은 OptionMenu를 만들어 보자





Option Menu 예제 2

■ 실행 화면





Option Menu 예제 2

■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8" ?>
<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:padding="10dp"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:text="Option 메뉴의 종류와 사용법에 대해 알아보자" />
</LinearLayout>
```




Option Menu 예제 2



```
<?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/SimpleMenu"
    android:orderInCategory="1"
    app:showAsAction="never"
    android:title="일반 메뉴"/>
  <item
    android:id="@+id/clickAbleMenu"
    android:orderInCategory="2"
    app:showAsAction="never"
    android:checkable="true"
    android:title="선택 메뉴"/>
  <item
    android:id="@+id/IconMenu"
    android:orderInCategory="3"
    app:showAsAction="ifRoom"
    android:icon="@drawable/ic_launcher"
    android:title="아이콘 메뉴"/>
```



Option Menu 예제 2

```
<item
  android:id="@+id/subMenu_1"
  android:orderInCategory="4"
  android:title="서브 메뉴"
  app:showAsAction="never">
  <menu>
    <group
      android:checkableBehavior="single">
        <item
          android:id="@+id/subMenu_2"
          android:checked="true"
          android:title="서브 메뉴 1"
          app:showAsAction="never" />
        <item
          android:id="@+id/subMenu_3"
          android:title="서브 메뉴 2"
          app:showAsAction="never" />
      </group>
    </menu>
  </item>
```



Option Menu 예제 2

■ menu.xml

```
<item
    android:id="@+id/actionBarMenu"
    android:orderInCategory="3"
    app:showAsAction="always"
    android:title="액션바 메뉴"/>
</menu>
```



Option Menu 예제 2

■ JAVA 프로그램

```
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.menu, menu);  
        return true;  
    }  
}
```



Option Menu 예제 2

■ JAVA 프로그램

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    if(item.getItemId()==R.id.SimpleMenu){  
        Toast.makeText(this, "기본 메뉴가 터치되었습니다",  
            Toast.LENGTH_SHORT).show();  
    }  
    if(item.getItemId()==R.id.clickAbleMenu){  
        if(item.isChecked()){  
            item.setChecked(false);  
            Toast.makeText(this, "체크가 해제되었습니다",  
                Toast.LENGTH_SHORT).show();  
        }else{  
            item.setChecked(true);  
            Toast.makeText(this, "체크 되었습니다",  
                Toast.LENGTH_SHORT).show();  
        }  
    }  
}
```



Option Menu 예제 2

```
if(item.getItemId()==R.id.IconMenu){
    Toast.makeText(this, "아이콘 메뉴가 터치되었습니다",
                    Toast.LENGTH_SHORT).show();
}
if(item.getItemId()==R.id.subMenu_2){
    Toast.makeText(this, "서브메뉴 1가 터치되었습니다",
                    Toast.LENGTH_SHORT).show();

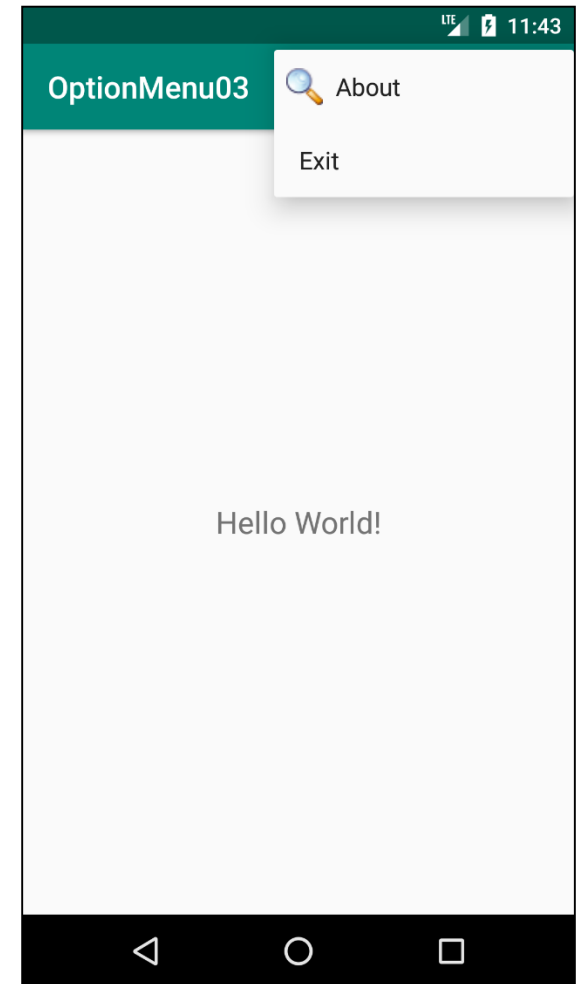
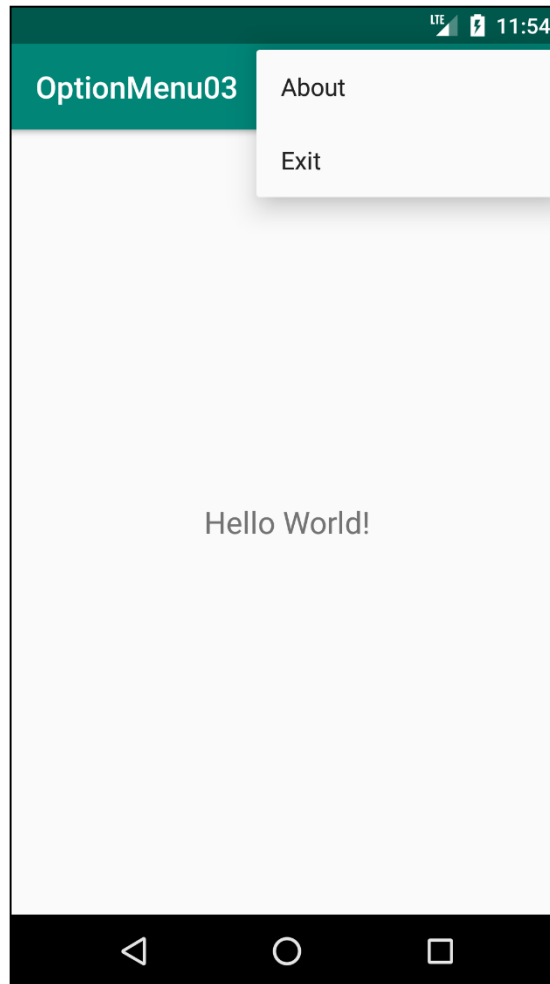
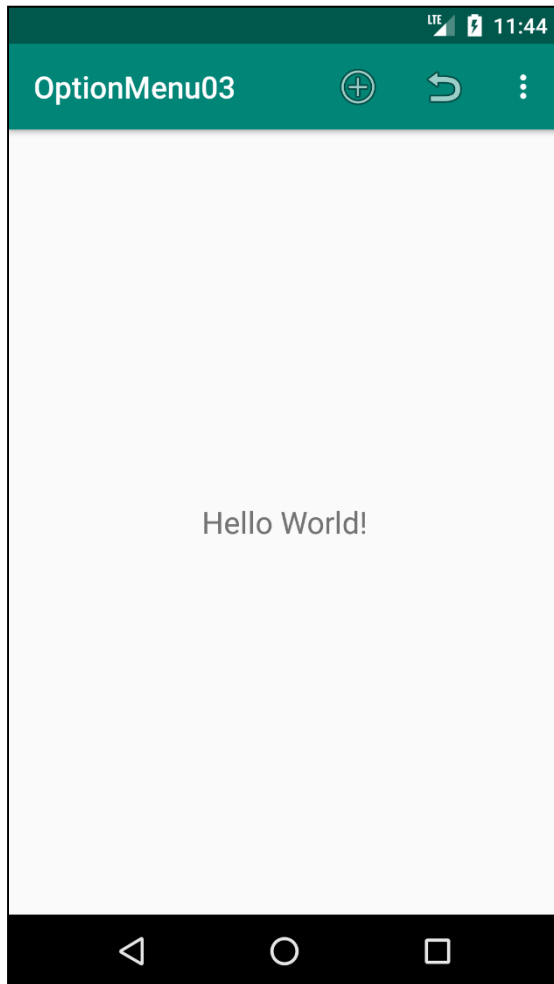
    item.setChecked(true);
} else if(item.getItemId()==R.id.subMenu_3){
    Toast.makeText(this, "서브메뉴 2이 터치되었습니다",
                    Toast.LENGTH_SHORT).show();

    item.setChecked(true);
}
if(item.getItemId()==R.id.actionBarMenu){
    Toast.makeText(this, "액션바 메뉴가 터치되었습니다",
                    Toast.LENGTH_SHORT).show();
}
return super.onOptionsItemSelected(item);
}
```



Option Menu 예제 3

■ 다음과 같이 ActionBar에 3개의 옵션 메뉴를 추가해보자





Option Menu 예제 3



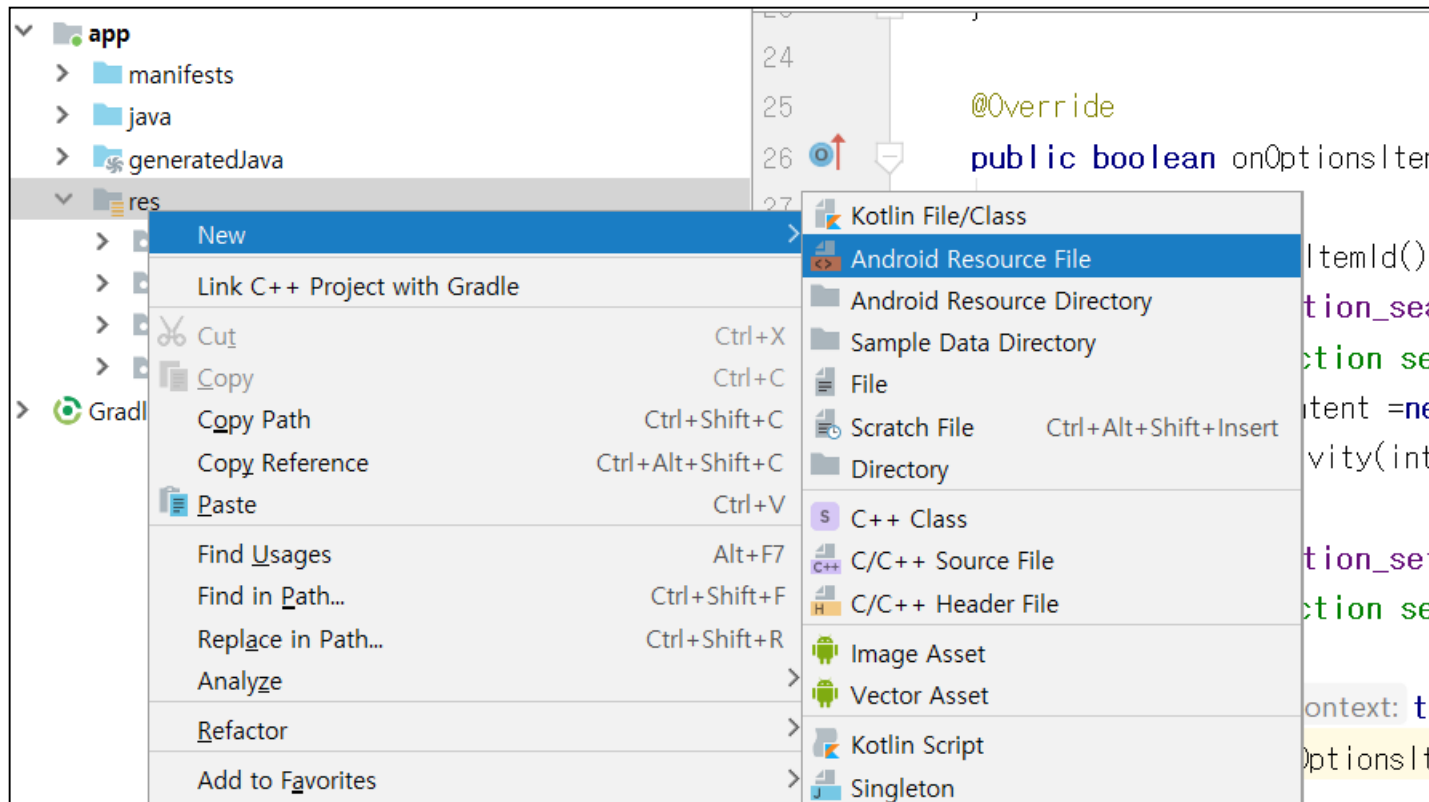
- Action Bar에 Option Menu를 추가하는 방법
 - res/menu 폴더 생성
 - Menu들이 정의된 XML 정의
 - Menu 리소스 XML은 "/res/menu/"에 생성하는데, XML 파일을 만든 다음, "<menu>" 요소 아래에 액션당 하나의 "<item>"을 사용하여 App Bar에 표시될 액션을 추가
- JAVA Code 수정
 - onCreateOptionsMenu() 메소드를 통한 메뉴 inflate
 - onOptionsItemSelected() 메소드를 통해 사용자가 메뉴를 클릭했을 때 처리



Option Menu 예제 3

■ menu 폴더 만들기

- Project의 app밑에 있는 res를 선택 후 마우스 오른쪽쪽을 클릭하여 팝업 메뉴에서 [New]-[Android resource file]을 클릭

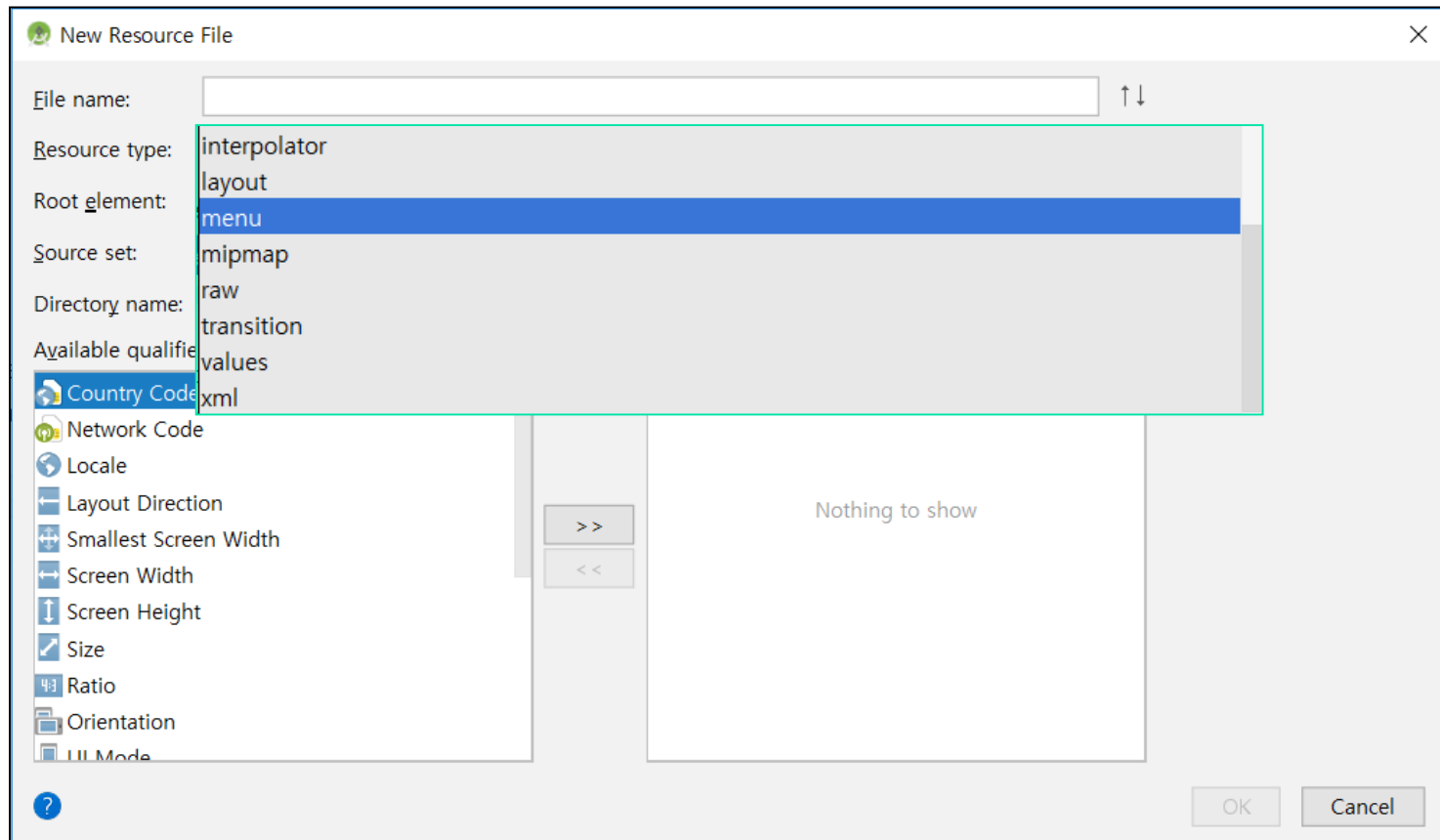




Option Menu 예제 3

■ menu 폴더 만들기

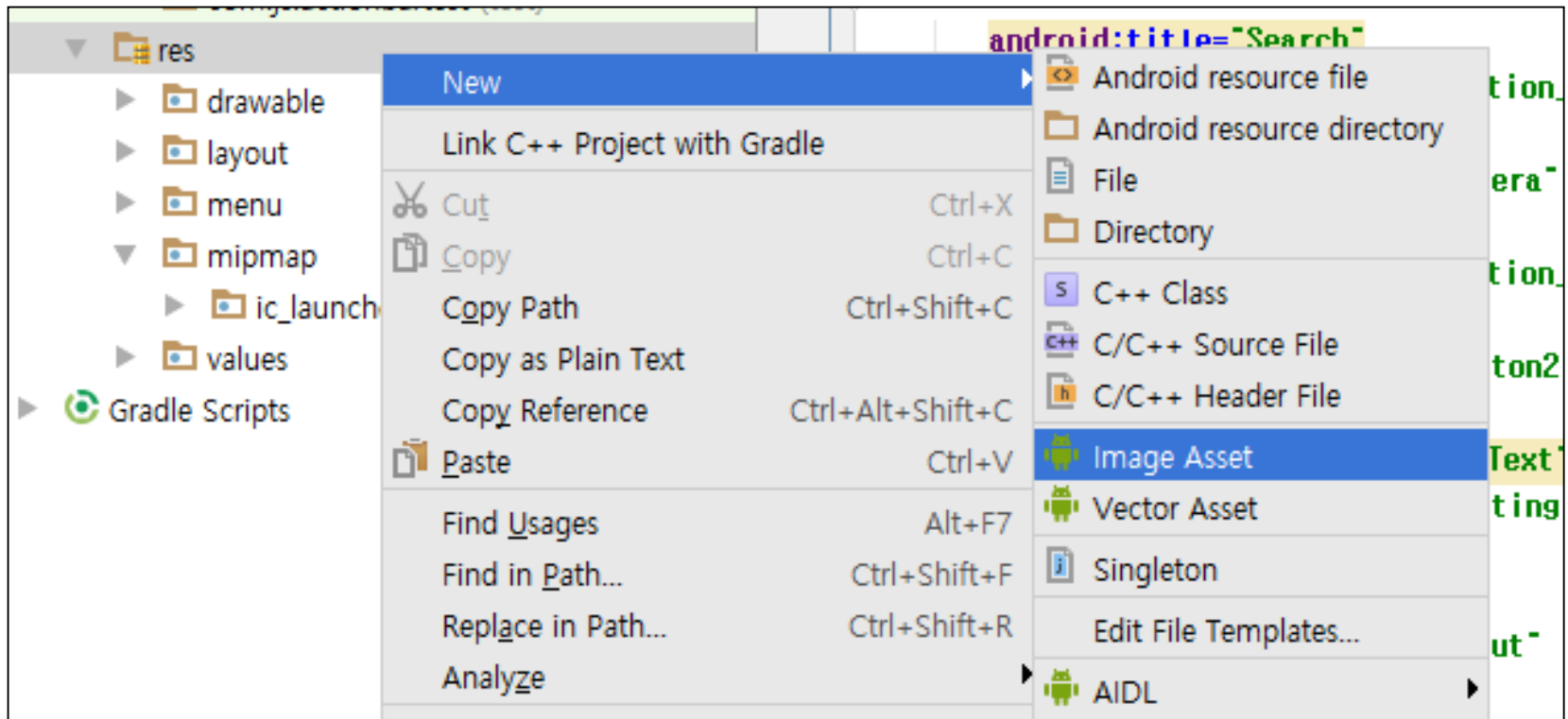
- 아래와 같은 대화상자가 표시되면, [Resource type]에서 [menu]를 선택하고 [OK] 버튼을 클릭





Option Menu 예제 3

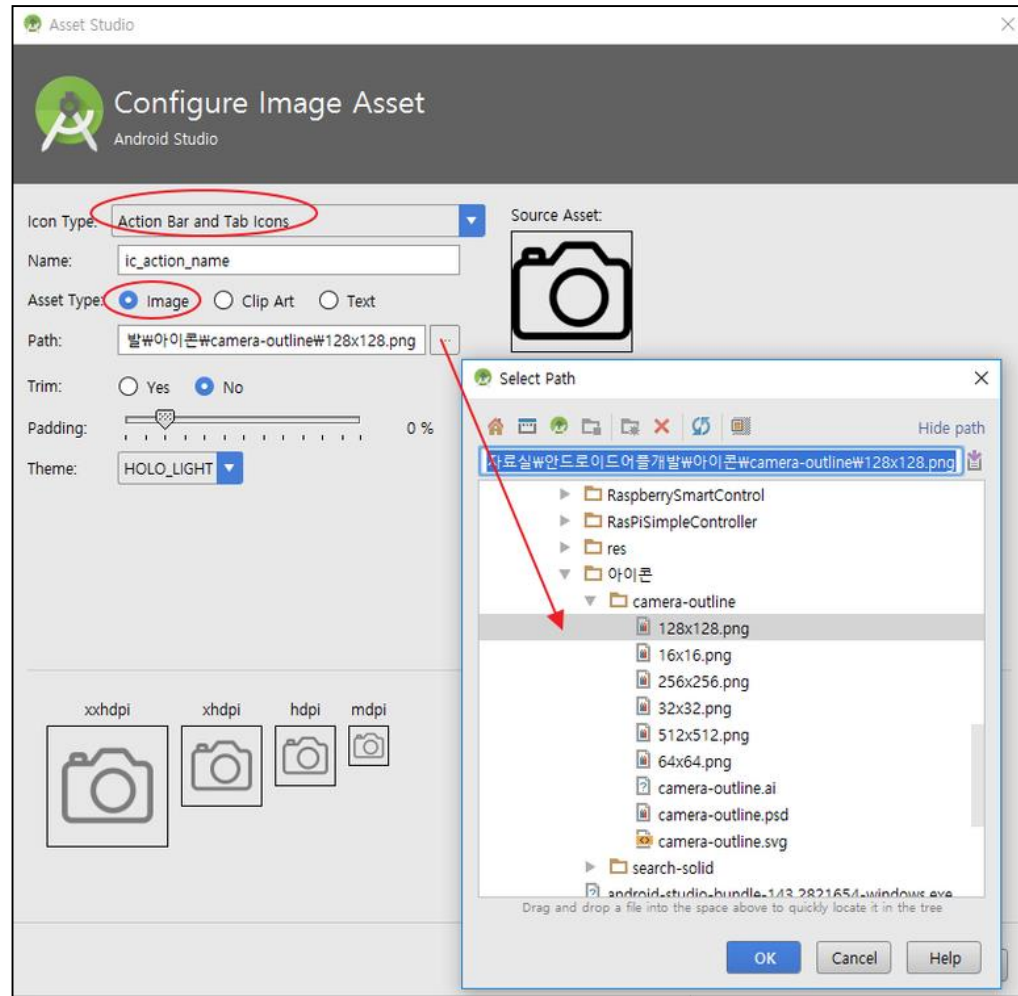
- 아이콘 리소스를 쉽게 추가하는 방법
 - 원하는 이미지 파일을 준비
 - res 폴더를 오른쪽 마우스 클릭해서 image Asset를 실행





Option Menu 예제 3

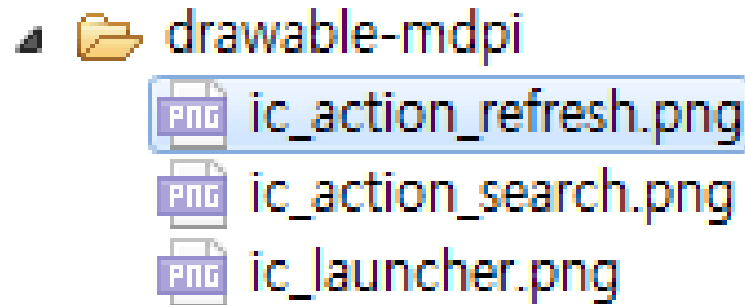
■ 아이콘 리소스를 쉽게 추가하는 방법





Option Menu 예제 3

- Minimum Required SDK를 API 11로 함
- /res/drawable 폴더에 다음과 같은 아이콘을 복사





Option Menu 예제 3

■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8" ?>
<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:gravity="center"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textSize="20dp"
        android:text="Hello World!"/>

</LinearLayout>
```



Option Menu 예제 3



■ optionmenu.xml

```
<?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/add"
        android:icon="@android:drawable/ic_menu_add"
        android:title="@string/add"
        app:showAsAction="always" />

    <item
        android:id="@+id/reset"
        android:icon="@android:drawable/ic_menu_revert"
        android:title="@string/reset"
        app:showAsAction="always|withText"/>
```



Option Menu 예제 3

■ optionmenu.xml

```
<item
    android:id="@+id/about"
    android:icon="@android:drawable/ic_dialog_info"
    android:title="@string/about"
    app:showAsAction="never"/>

<item
    android:id="@+id/exit"
    android:title="@string/exit"
    app:showAsAction="never"/>

</menu>
```




Option Menu 예제 3

■ optionmenu1.xml (icon 표시)

```
<?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/add"
    android:icon="@android:drawable/ic_menu_add"
    android:title="@string/add"
    app:showAsAction="always" />

  <item
    android:id="@+id/reset"
    android:icon="@android:drawable/ic_menu_revert"
    android:title="@string/reset"
    app:showAsAction="always|withText" />
```



Option Menu 예제 3

■ optionmenu1.xml (icon 표시)

```
<item
    android:icon="@drawable/ic_more_vert_white_24dp"
    android:title=""
    app:showAsAction="always">
    <menu>
        <item
            android:id="@+id/about"
            android:icon="@drawable/untitled"
            android:title="@string/about"
            app:showAsAction="never" />

        <item
            android:id="@+id/exit"
            android:title="@string/exit"
            app:showAsAction="never" />
    </menu>
</item>
</menu>
```



Option Menu 예제 3

■ ic_more_vert_white_24dp.xml

```
<?xml version="1.0" encoding="utf-8"?>
<vector xmlns:android="http://schemas.android.com/apk/res/android"
    android:width="24dp"
    android:height="24dp"
    android:viewportWidth="24.0"
    android:viewportHeight="24.0">
    <path
        android:fillColor="#ffffff"
        android:pathData="M12,8c1.1,0 2,-0.9 2,-2s-0.9,-2 -2,-2 -2,0.9 -2,
            2 0.9,2 2,2zM12,10c-1.1,0 -2,0.9 -2,2s0.9,2 2,2 2,-0.9 2,
            -2 -0.9,-2 -2,-2zM12,16c-1.1,0 -2,0.9 -2,2s0.9,2 2,2 2,-0.9 2,
            -0.9 2,-2 -0.9,-2 -2,-2z"/>
</vector>
```



Option Menu 예제 3

■ strings 파일

```
<resources>
  <string name="app_name">OptionMenu03</string>

  <string name="count">Nothing is selected</string>
  <string name="action_settings">Settings</string>
  <string name="reset">Speak</string>
  <string name="about">About</string>
  <string name="exit">Exit</string>
  <string name="add">Add</string>

</resources>
```



Option Menu 예제 3



■ JAVA 코드

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



Option Menu 예제 3



@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()){  
        case R.id.add:  
            textView.setText("ADD가 클릭 되었음");  
            break;  
        case R.id.reset:  
            textView.setText("Reset이 클릭 되었음");  
            break;  
        case R.id.about:  
            textView.setText("OverFlow Icon이 클릭 되었음");  
            Toast.makeText(this, "About is 클릭되었다",  
                Toast.LENGTH_LONG).show();  
            break;  
        case R.id.exit:  
            finish();  
    }  
    return super.onOptionsItemSelected(item);  
}
```



Option Menu 예제 4

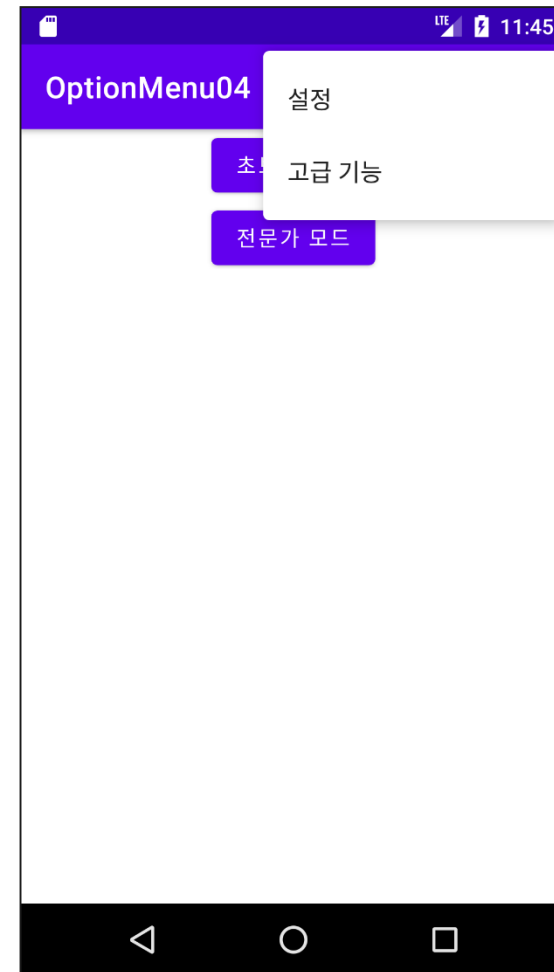
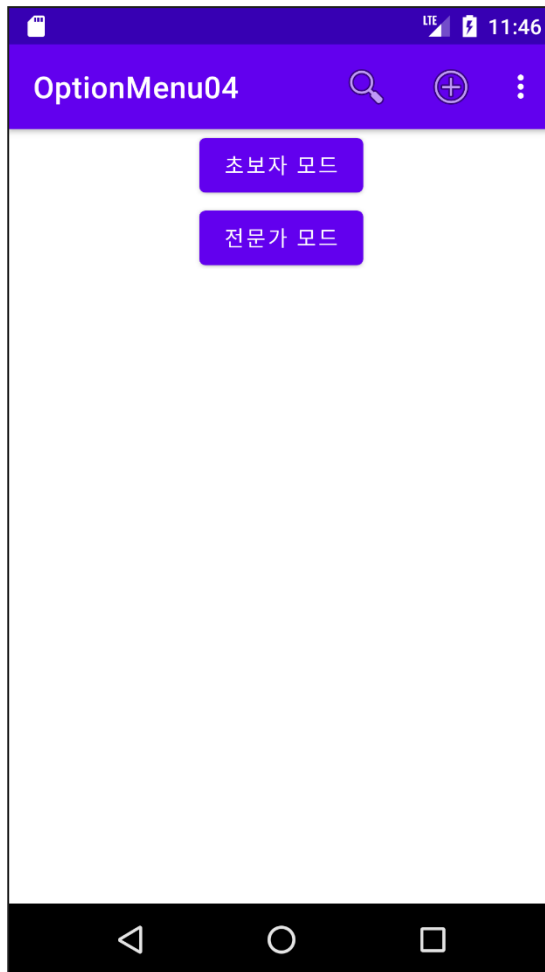
■ 초보자 모드





Option Menu 예제 4

■ 전문가 모드





Option Menu 예제 4

■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<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"
    android:gravity="center|top"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/btnbeginner"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="mOnClick"
        android:text="초보자 모드" />
```



Option Menu 예제 4

■ 사용자 인터페이스

```
<Button
    android:id="@+id/btnprofessional"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:onClick="mOnClick"
    android:text="전문가 모드" />
</LinearLayout>
```



Option Menu 예제 4

■ beginner.xml

```
<?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:icon="@android:drawable/ic_menu_help"
    android:title="도움말"
    app:showAsAction="always|withText" />
  <item
    android:title="프로그램 소개" />
</menu>
```



Option Menu 예제 4

■ professional.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:icon="@android:drawable/ic_menu_search"
    android:title="검색"
    app:showAsAction="always|withText" />
  <item
    android:icon="@android:drawable/ic_menu_add"
    android:title="추가"
    app:showAsAction="always|withText" />
  <item
    android:title="설정" />
  <item
    android:title="고급 기능" />
</menu>
```



Option Menu 예제 4



■ MainActivity.JAVA

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



Option Menu 예제 4

■ MainActivity.JAVA

```
public void mOnClick(View v) {  
    switch (v.getId()) {  
        case R.id.btnbeginner:  
            beginner = true;  
            invalidateOptionsMenu();  
            break;  
        case R.id.btnprofessional:  
            beginner = false;  
            invalidateOptionsMenu();  
            break;  
    }  
}
```



Option Menu 예제 5(XML)

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch(item.getItemId()) {  
        case R.id.menu_refresh:  
            Toast.makeText(this, "새로고침 메뉴가 선택되었습니다.",  
                           Toast.LENGTH_SHORT).show();  
  
            break;  
        case R.id.menu_search:  
            Toast.makeText(this, "검색 메뉴가 선택되었습니다.",  
                           Toast.LENGTH_SHORT).show();  
  
            break;  
        case R.id.menu_settings:  
            Toast.makeText(this, "설정 메뉴가 선택되었습니다.",  
                           Toast.LENGTH_SHORT).show();  
  
            break;  
        default:  
    }  
    return super.onOptionsItemSelected(item);  
}
```



Option Menu 예제 5(JAVA)

■ Java 프로그램

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




Option Menu 예제 5(JAVA)

■ Java 프로그램

@Override

```
public boolean onCreateOptionsMenu(Menu menu) {  
    menu.add(0, 1, 0, "새로고침").  
        setIcon(R.drawable.menu_refresh).  
        setShowAsAction(MenuItem.SHOW_AS_ACTION_ALWAYS);  
    menu.add(0, 2, 0, "검색").  
        setIcon(R.drawable.menu_search).  
        setShowAsAction(MenuItem.SHOW_AS_ACTION_ALWAYS);  
    menu.add(0, 3, 0, "설정").  
        setIcon(R.drawable.menu_settings).  
        setShowAsAction(MenuItem.SHOW_AS_ACTION_ALWAYS);  
    return super.onCreateOptionsMenu(menu);  
}
```



Option Menu 예제 5(JAVA)



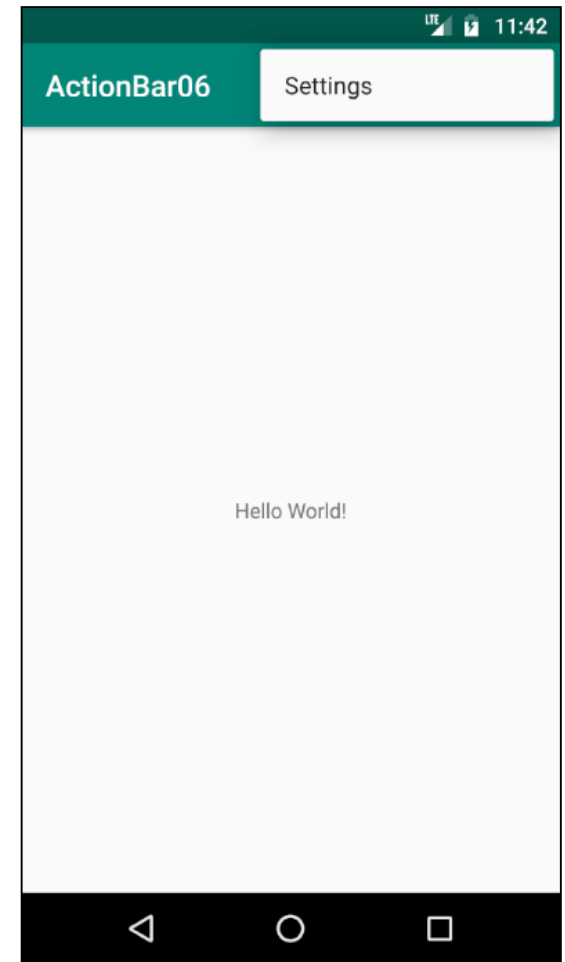
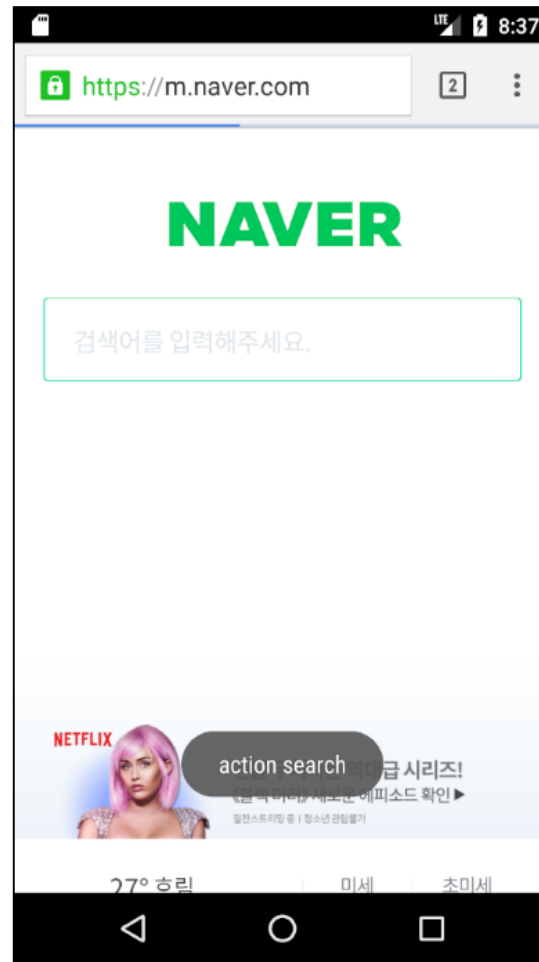
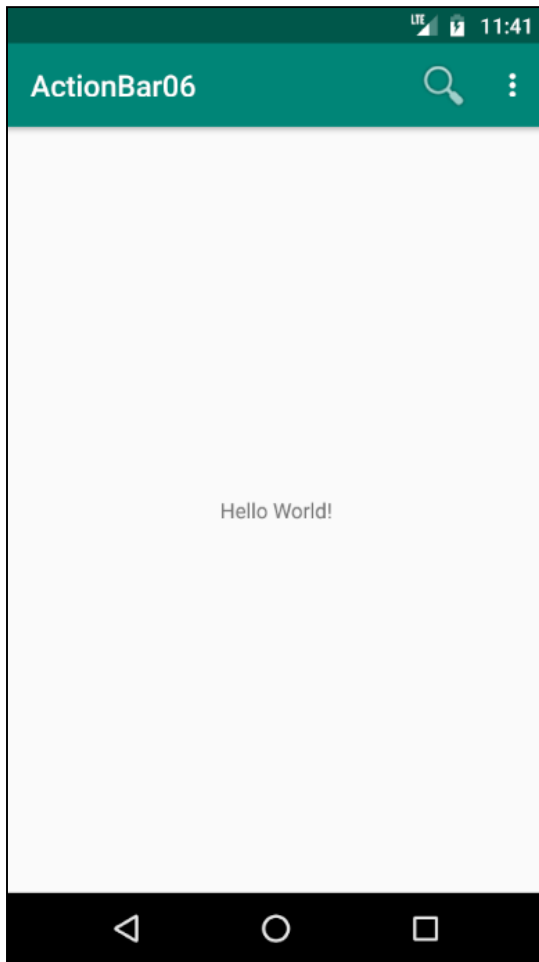
@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch(item.getItemId()) {  
        case 1 :  
            Toast.makeText(this, "새로고침 메뉴가 선택되었습니다.",  
                           Toast.LENGTH_SHORT).show();  
            break;  
        case 2 :  
            Toast.makeText(this, "검색 메뉴가 선택되었습니다.",  
                           Toast.LENGTH_SHORT).show();  
            break;  
        case 3 :  
            Toast.makeText(this, "설정 메뉴가 선택되었습니다.",  
                           Toast.LENGTH_SHORT).show();  
            break;  
        default:  
    }  
    return super.onOptionsItemSelected(item);  
}
```



Option Menu 예제 6

■ 다음과 같이 ActionBar에 2개의 옵션 메뉴를 추가해보자





Option Menu 예제 6



■ actionBar.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
  <item
    android:id="@+id/action_search"
    android:icon="@drawable/ic_search"
    app:showAsAction="ifRoom"
    android:title="@string/action_search" />

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



Option Menu 예제 6



■ strings.xml

```
<resources>
  <string name="app_name">OptionMenu04</string>

  <string name="action_settings">Settings</string>
  <string name="action_search">Search</string>
</resources>
```



Option Menu 예제 6

■ JAVA 프로그램

```
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) {  
        getMenuInflater().inflate(R.menu.actionbar, menu);  
        return super.onCreateOptionsMenu(menu);  
    }  
}
```



Option Menu 예제 6

■ JAVA 프로그램

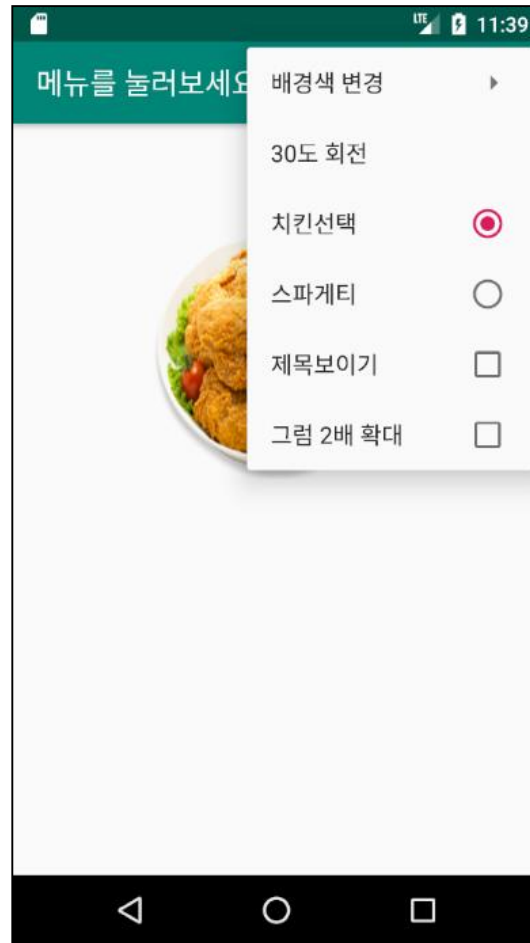
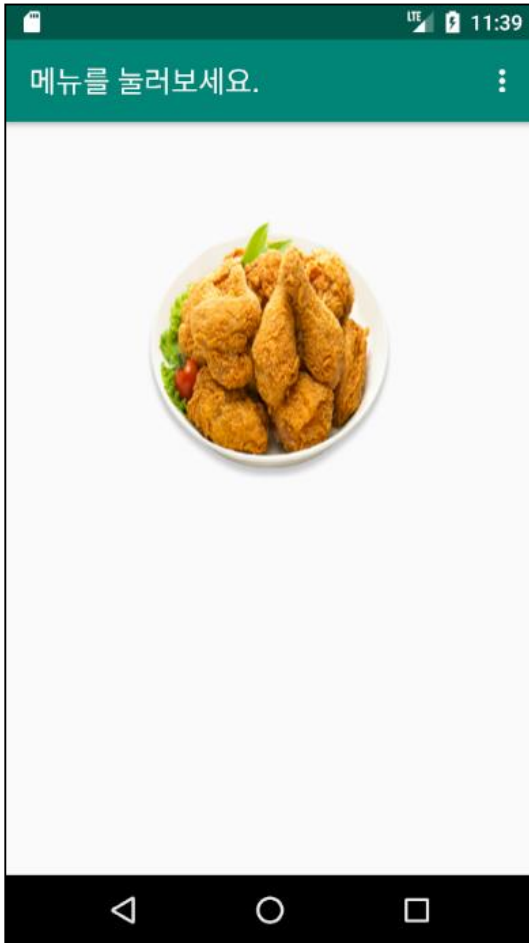
@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    String txt = "";  
    switch (item.getItemId()) {  
        case R.id.action_search :  
            txt = "action search";  
            Intent intent = new Intent(Intent.ACTION_VIEW,  
                                       Uri.parse("http://m.naver.com"));  
            startActivity(intent);  
            break;  
        case R.id.action_settings :  
            txt = "action settings";  
    }  
    Toast.makeText(this, txt, Toast.LENGTH_LONG).show();  
    return super.onOptionsItemSelected(item);  
}
```



Option Menu 예제 7

■ 다음과 같은 OptionMenu를 작성하여라





Option Menu 예제 7

■ Image 준비



chicken



spaghetti



Option Menu 예제 7

■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/activity_main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:context=".MainActivity">

    <FrameLayout
        android:layout_width="match_parent"
        android:layout_height="300dp"
        android:layout_gravity="center">
```



Option Menu 예제 7



■ 사용자 인터페이스

```
<ImageView  
    android:id="@+id/image_chicken"  
    android:layout_width="200dp"  
    android:layout_height="200dp"  
    android:layout_gravity="center"  
    android:visibility="visible"  
    android:background="@drawable/chicken" />
```

```
<ImageView  
    android:id="@+id/image_spaghetti"  
    android:layout_width="200dp"  
    android:layout_height="200dp"  
    android:layout_gravity="center"  
    android:visibility="gone"  
    android:background="@drawable/spaghetti" />
```

```
</FrameLayout>
```



Option Menu 예제 7



■ 사용자 인터페이스

```
<TextView
    android:id="@+id/chicken"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="치킨"
    android:textSize="30dp"
    android:visibility="gone" />
```

```
<TextView
    android:id="@+id/spaghetti"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:text="스파게티"
    android:textSize="30dp"
    android:visibility="gone" />
```

```
</LinearLayout>
```



Option Menu 예제 7

■ menu/menu.xml

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:title="배경색 변경">
    <menu>
      <item
        android:id="@+id/item_red"
        android:title="빨간색" />
      <item
        android:id="@+id/item_yellow"
        android:title="노란색" />
      <item
        android:id="@+id/item_blue"
        android:title="파란색" />
      <item
        android:id="@+id/item_white"
        android:title="흰색" />
    </menu>
  </item>
```



Option Menu 예제 7



■ menu/menu.xml

```
<item
    android:id="@+id/item_rotate"
    android:title="30도 회전" />
<group
    android:checkableBehavior="single"
    android:enabled="true">
    <item
        android:id="@+id/item_chicken"
        android:checked="true"
        android:enabled="true"
        android:title="치킨선택" />
    <item
        android:id="@+id/item_spaghetti"
        android:enabled="true"
        android:title="스파게티" />
</group>
```



Option Menu 예제 7



■ menu/menu.xml

```
<group
    android:id="@+id/group2"
    android:checkableBehavior="all"
    android:enabled="true"
    android:visible="true">
    <item
        android:id="@+id/item_title"
        android:checkable="true"
        android:checked="false"
        android:enabled="true"
        android:title="제목보이기"
        android:visible="true" />
    <item
        android:id="@+id/item_twice"
        android:checkable="true"
        android:enabled="true"
        android:title="그럼 2배 확대" />
</group>
</menu>
```



Option Menu 예제 7

■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {  
    LinearLayout main;  
    FrameLayout.LayoutParams params_c,params_s;  
    TextView textView1,textView2;  
    ImageView imageView1, imageView2;  
    MenuItem item_c, item_s,item_twice,item_title,item_rotate;  
    int index = 1;  
    float rotate = 0.0f;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        setTitle("메뉴를 눌러보세요.");  
    }  
}
```




Option Menu 예제 7



■ MainActivity.JAVA

```
main = findViewById(R.id.activity_main);
textView1 = findViewById(R.id.chicken);
textView2 = findViewById(R.id.spaghetti);
imageView1 = findViewById(R.id.image_chicken);
imageView2 = findViewById(R.id.image_spaghetti);
}

@Override
public boolean onCreateOptionsMenu(Menu menu){
    getMenuInflater().inflate(R.menu.menu, menu);
    item_c = menu.findItem(R.id.item_chicken);
    item_s = menu.findItem(R.id.item_spaghetti);
    item_title = menu.findItem(R.id.item_title);
    item_twice = menu.findItem(R.id.item_twice);
    item_rotate = menu.findItem(R.id.item_rotate);
    params_c = (FrameLayout.LayoutParams) imageView1.getLayoutParams();
    params_s = (FrameLayout.LayoutParams) imageView2.getLayoutParams();
    return true;
}
```



Option Menu 예제 7

■ MainActivity.JAVA

@Override

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch(item.getItemId()){  
        case R.id.item_chicken :  
            index = 1;  
            item.setChecked(true);  
            visible(imageView1, imageView2, index);  
            if(item_title.isChecked())  
                visible(textView1, textView2, index);  
            break;  
        case R.id.item_spaghetti :  
            index = 2;  
            item.setChecked(true);  
            visible(imageView1, imageView2, index);  
            if(item_title.isChecked())  
                visible(textView1, textView2, index);  
            break;  
    }  
}
```



Option Menu 예제 7



■ MainActivity.JAVA

```
case R.id.item_title:
    if(item.isChecked()) {
        textView1.setVisibility(View.GONE);
        textView2.setVisibility(View.GONE);
        item.setChecked(false);
    } else{
        visible(textView1, textView2, index);
        item.setChecked(true);
    }
    break;
case R.id.item_twice:
    if(item.isChecked()){
        item.setChecked(false);
        changeSize(400);
    }else{
        item.setChecked(true);
        changeSize(800);
    }
    break;
```



Option Menu 예제 7

■ MainActivity.JAVA

```
case R.id.item_rotate:
    rotate += 30.0f;
    imageView1.setRotation(rotate);
    imageView2.setRotation(rotate);
    break;
default :
    if(item.getItemId() == R.id.item_red)
        main.setBackgroundColor(Color.RED);
    else if(item.getItemId() == R.id.item_blue)
        main.setBackgroundColor(Color.BLUE);
    else if(item.getItemId() == R.id.item_yellow)
        main.setBackgroundColor(Color.YELLOW);
    else if (item.getItemId() == R.id.item_white)
        main.setBackgroundColor(Color.WHITE);
}
return true;
}
```



Option Menu 예제 7

■ MainActivity.JAVA

```
public void visible(Imageview chick, Imageview spa, int index) {  
    if (index == 1){  
        chick.setVisibility(View.VISIBLE);  
        spa.setVisibility(View.GONE);  
    } else {  
        chick.setVisibility(View.GONE);  
        spa.setVisibility(View.VISIBLE);  
    }  
}  
  
public void visible(Textview chick, Textview spa, int index) {  
    if (index == 1){  
        chick.setVisibility(View.VISIBLE);  
        spa.setVisibility(View.GONE);  
    } else {  
        chick.setVisibility(View.GONE);  
        spa.setVisibility(View.VISIBLE);  
    }  
}
```



Option Menu 예제 7

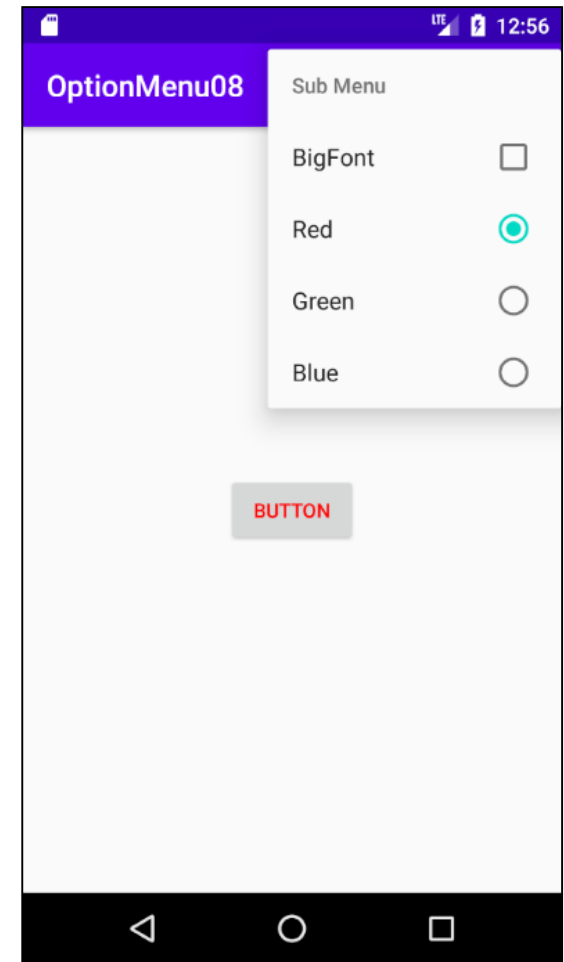
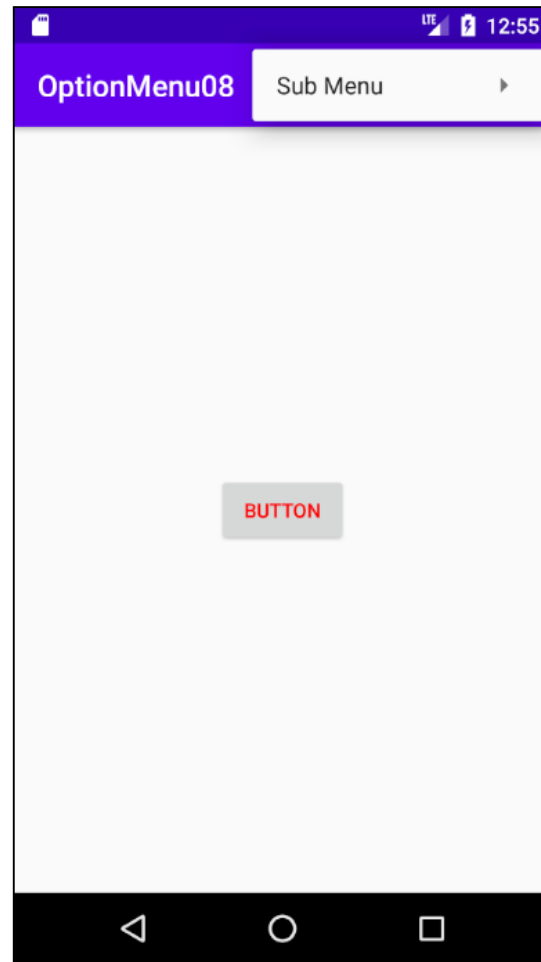
■ MainActivity.JAVA

```
public void changeSize(int size){  
    params_c.width = size;  
    params_c.height = size;  
    params_s.width = size;  
    params_s.height = size;  
    imageView1.setLayoutParams(params_c);  
    imageView2.setLayoutParams(params_s);  
}  
}
```



옵션 메뉴 예제 8

■ 다음과 같은 OptionMenu를 작성하여라





옵션 메뉴 예제 8

■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8" ?>
<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:gravity="center"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#ff0000"
        android:text="Button"
        android:textSize="40px"/>

</LinearLayout>
```




옵션 메뉴 예제 8



■ MainActivity.JAVA

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



옵션 메뉴 예제 8



■ MainActivity.JAVA

```
public boolean onPrepareOptionsMenu(Menu menu) {  
    if (button.getTextSize() == 80)  
        menu.findItem(R.id.bigfont).setChecked(true);  
    else  
        menu.findItem(R.id.bigfont).setChecked(false);  
  
    int color = button.getTextColors().getDefaultColor();  
    if (color == Color.RED)  
        menu.findItem(R.id.red).setChecked(true);  
    if (color == Color.GREEN)  
        menu.findItem(R.id.green).setChecked(true);  
    if (color == Color.BLUE)  
        menu.findItem(R.id.blue).setChecked(true);  
    return true;  
}
```



옵션 메뉴 예제 8

■ MainActivity.JAVA

```
public boolean onOptionsItemSelected(MenuItem item) {  
    switch (item.getItemId()) {  
        case R.id.bigfont:  
            if (item.isChecked()) {  
                button.setTextSize(TypedValue.COMPLEX_UNIT_PX, 40);  
            } else {  
                button.setTextSize(TypedValue.COMPLEX_UNIT_PX, 80);  
            }  
            return true;  
        case R.id.red:  
            button.setTextColor(Color.RED);  
            return true;  
        case R.id.green:  
            button.setTextColor(Color.GREEN);  
            return true;  
    }  
}
```



옵션 메뉴 예제 8

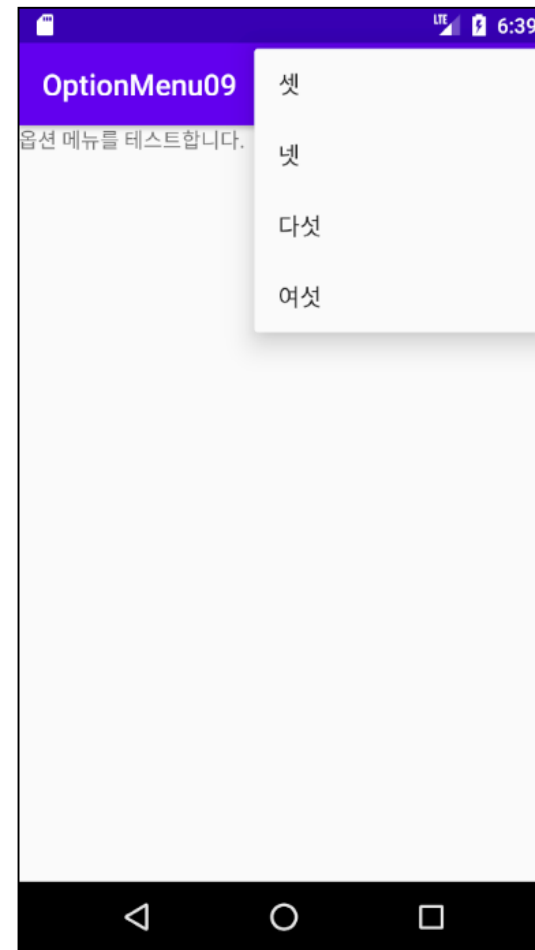
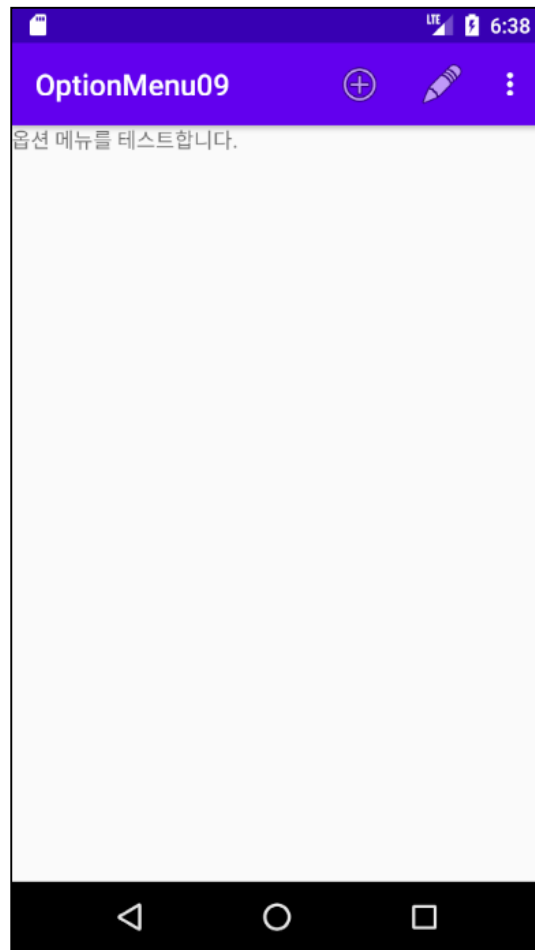
■ MainActivity.JAVA

```
        case R.id.blue:  
            button.setTextColor(Color.BLUE);  
            return true;  
        }  
        return false;  
    }  
}
```



옵션 메뉴 예제 9

- 다음과 같은 OptionMenu를 작성하여라





옵션 메뉴 예제 9

■ menu.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto">
  <item
    android:icon="@android:drawable/ic_menu_add"
    android:title="하나"
    app:showAsAction="always|withText" />
  <item
    android:icon="@android:drawable/ic_menu_edit"
    android:title="둘"
    app:showAsAction="ifRoom" />
  <item
    android:title="셋"
    app:showAsAction="ifRoom" />
  <item
    android:title="넷"
    app:showAsAction="ifRoom" />
</menu>
```



옵션 메뉴 예제 9

■ menu.xml

```
<item
    android:title="다섯"
    app:showAsAction="ifRoom" />
<item android:title="여섯" />
</menu>
```



옵션 메뉴 예제 9



■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
  
        requestWindowFeature(Window.FEATURE_ACTION_BAR);  
        TextView text = new TextView(this);  
        text.setText("옵션 메뉴를 테스트합니다.");  
        setContentView(text);  
    }  
    public boolean onCreateOptionsMenu(Menu menu) {  
        super.onCreateOptionsMenu(menu);  
        MenuInflater inflater = getMenuInflater();  
        inflater.inflate(R.menu.menu, menu);  
  
        return true;  
    }  
}
```