



학습 목표

- 안드로이드의 메뉴 유형을 이해한다.
- 옵션, 컨텍스트, 팝업 메뉴 처리에 대해 익힌다.
- 안드로이드의 앱 바 유형에 대해 이해한다.
- 액션 바와 톨 바 처리에 대해 익힌다.

목차

1. 01 메뉴 살펴보기
2. 02 기본 유형 메뉴 만들기
3. 03 앱 바 다루기
4. [도서 쇼핑몰] 메뉴와 앱 바 만들기

01

메뉴 살펴보기

1. 메뉴 살펴보기

■ 메뉴(Menu)

- 사용자와 상호작용할 수 있는 UI 요소 중 하나
- 앱의 상황에 따라 특정 작업을 실행하도록 유도하며 일부 공통 기능을 처리하는 데 사용
- 메뉴를 사용하면 액티비티에서 메뉴 항목이나 구성 요소에 대한 구체적인 정보를 상황에 맞게 표시하여 사용자 인터페이스를 보다 쉽게 제공할 수 있음

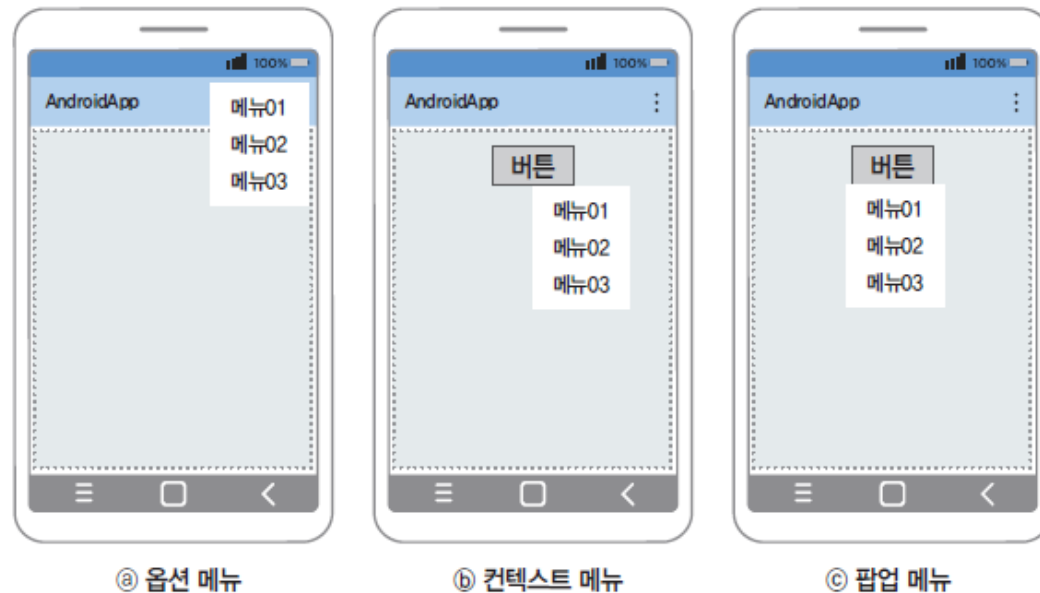
1. 메뉴 살펴보기

■ 메뉴(Menu)

표 7-1 메뉴 유형

유형	설명
옵션 메뉴	메뉴 버튼(:)을 클릭할 때 나타나는 메뉴입니다.
컨텍스트 메뉴	특정 위젯을 길게 클릭할 때 나타나는 메뉴입니다.
팝업 메뉴	명령에 따라 특정 위치에 팝업되어 나타나는 메뉴입니다.

그림 7-1 메뉴 유형 화면 예



1. 메뉴 살펴보기

■ XML로 메뉴 생성

- 메뉴 항목을 포함한 전용 XML 리소스 파일을 생성해야 함

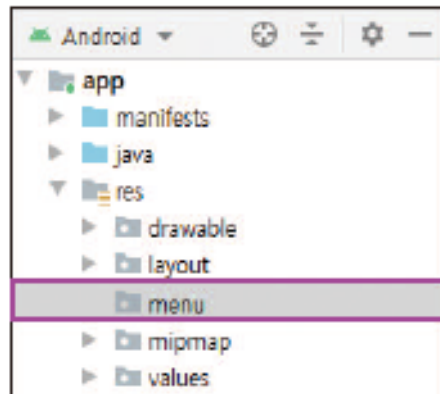
그림 7-2 메뉴 전용 XML 파일 생성 과정



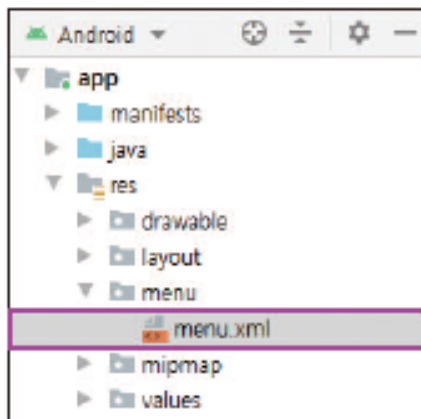
1. 메뉴 살펴보기

■ XML로 메뉴 생성

- menu 폴더 생성하기



- 메뉴 XML 파일 생성하기



1. 메뉴 살펴보기

■ XML로 메뉴 생성

■ 메뉴 항목 정의하기

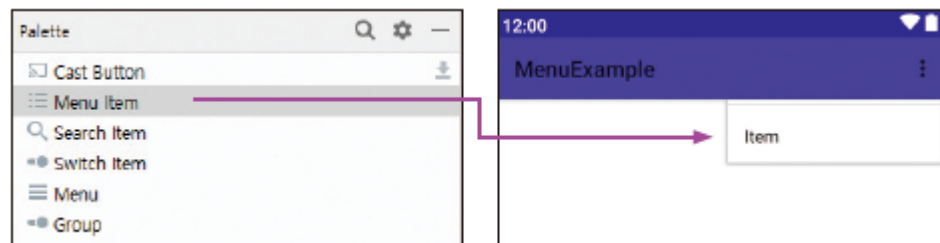


표 7-2 메뉴의 구성 요소

요소	설명
<menu>	메뉴를 정의하는 루트 요소로, 1개 이상의 <item>과 <group> 요소를 포함합니다.
<item>	하나의 메뉴 항목을 나타냅니다. 하위 메뉴를 만들기 위해 <menu> 요소를 포함할 수도 있습니다.
<group>	선택 사항으로, <item> 요소를 묶어 관리할 수 있습니다.

표 7-3 <item> 요소의 속성

요소	설명
id	메뉴 항목을 식별하는 아이디입니다.
icon	메뉴 항목에 표시할 아이콘 이미지를 설정합니다. (Android API 11 버전 이상부터 아이콘 이미지가 표시되지 않습니다.)
title	메뉴 항목에 표시할 문자열을 설정합니다.
showAsAction	메뉴 항목을 앱 바에서 작업 항목으로 표시하는 방법을 설정합니다. showAsAction에 대한 자세한 내용은 《7-3장-앱 바 다루기》를 참고하시기 바랍니다.

1. 메뉴 살펴보기

```
<?xml version="1.0" encoding="utf-8"?>
```

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

```
  <item ①  
    android:id="@+id/edit"  
    android:title="편집하기" />
```

```
  <item ②  
    android:id="@+id/search"  
    android:title="검색하기" />
```

```
  <item ③  
    android:id="@+id/save"  
    android:title="저장하기" />
```

```
</menu>
```



```
<?xml version="1.0" encoding="utf-8"?>
```

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

```
  <item  
    android:id="@+id/edit"  
    android:title="편집하기" />
```

```
  <menu ①
```

```
    <item  
      android:id="@+id/copy"  
      android:title="복사하기" />
```

```
    <item  
      android:id="@+id/cut"  
      android:title="잘라내기" />
```

```
    <item  
      android:id="@+id/paste"  
      android:title="붙여넣기" />
```

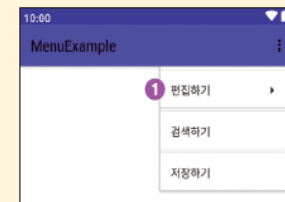
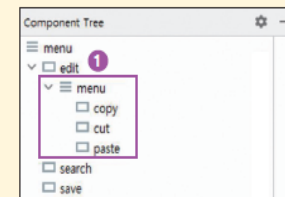
```
  </menu>
```

```
</item>
```

```
<item  
  android:id="@+id/search"  
  android:title="검색하기" />
```

```
<item  
  android:id="@+id/save"  
  android:title="저장하기" />
```

```
</menu>
```



1. 메뉴 살펴보기

■ 코드로 메뉴 생성

- 액티비티 상황에 따라 동적으로 메뉴 항목을 구성해야 할 때 코드에서 메뉴 항목들을 추가하거나 서브 메뉴 항목들을 만들 수 있음
- 이때 메뉴 항목들은 MenuItem 클래스의 add() 메서드를 호출하여 추가 가능

```
MenuItem add (int groupId, int itemId, int order, CharSequence title)
```

```
MenuItem add (int groupId, int itemId, int order, int titleRes)
```

▲ add() 메서드의 형식

표 7-4 add() 메서드의 매개 변수

매개 변수	설명
groupId	메뉴 항목을 그룹으로 구성할 때 지정합니다. 그룹 구성이 아니라 단독 항목일 때는 'NONE(0)'으로 지정합니다.
itemId	메뉴 항목을 식별하는 고유 아이디로, 메뉴 항목을 선택할 때 아이디로 구분합니다.
order	메뉴 항목의 순서를 지정합니다. 'NONE(0)'으로 지정하면 추가된 메뉴 항목 순서대로 나타납니다.
title, titleRes	메뉴 항목의 문자열 또는 리소스의 문자열 ID를 지정합니다.

1. 메뉴 살펴보기

```
import android.view.Menu;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        ...
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {

        menu.add(Menu.NONE, Menu.FIRST+0, Menu.NONE, "편집하기");
        menu.add(Menu.NONE, Menu.FIRST+1, Menu.NONE, "검색하기");
        menu.add(Menu.NONE, Menu.FIRST+2, Menu.NONE, "저장하기");

        return super.onCreateOptionsMenu(menu);
    }
}
```



1. 메뉴 살펴보기

■ 코드로 메뉴 생성

- 메뉴 항목에 하위 메뉴를 추가할 때 SubMenu 클래스의 addSubMenu() 메서드 호출

```
SubMenu addSubMenu(int titleRes); // 항목이 매우 많을 때 별도로 하위 메뉴 추가
```

```
import android.view.Menu;
import android.view.SubMenu;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        ...
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        SubMenu sub = menu.addSubMenu("편집하기");
        sub.add(Menu.NONE, Menu.FIRST+0, Menu.NONE, "복사하기");
        sub.add(Menu.NONE, Menu.FIRST+10, Menu.NONE, "잘라내기");
        sub.add(Menu.NONE, Menu.FIRST+20, Menu.NONE, "붙여넣기");

        menu.add(Menu.NONE, Menu.FIRST+1, Menu.NONE, "검색하기");
        menu.add(Menu.NONE, Menu.FIRST+2, Menu.NONE, "저장하기");
        return super.onCreateOptionsMenu(menu);
    }
}
```



02

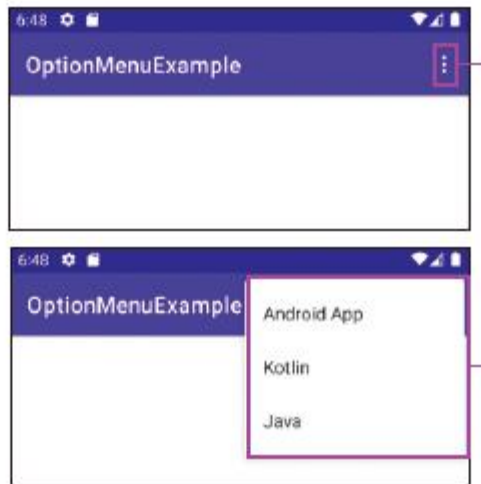
기본 유형 메뉴 만들기

2. 기본 유형 메뉴 만들기

■ 옵션 메뉴

- 기본 메뉴로 액티비티와 관련된 메뉴 항목을 모아둔 것
- 앱 전체에 영향을 미치는 작업인 검색, 로그인/로그아웃, 알림, 설정 등을 구현하는 데 사용
- 메뉴 버튼을 누르면 상단 툴 바에 메뉴 항목 표시

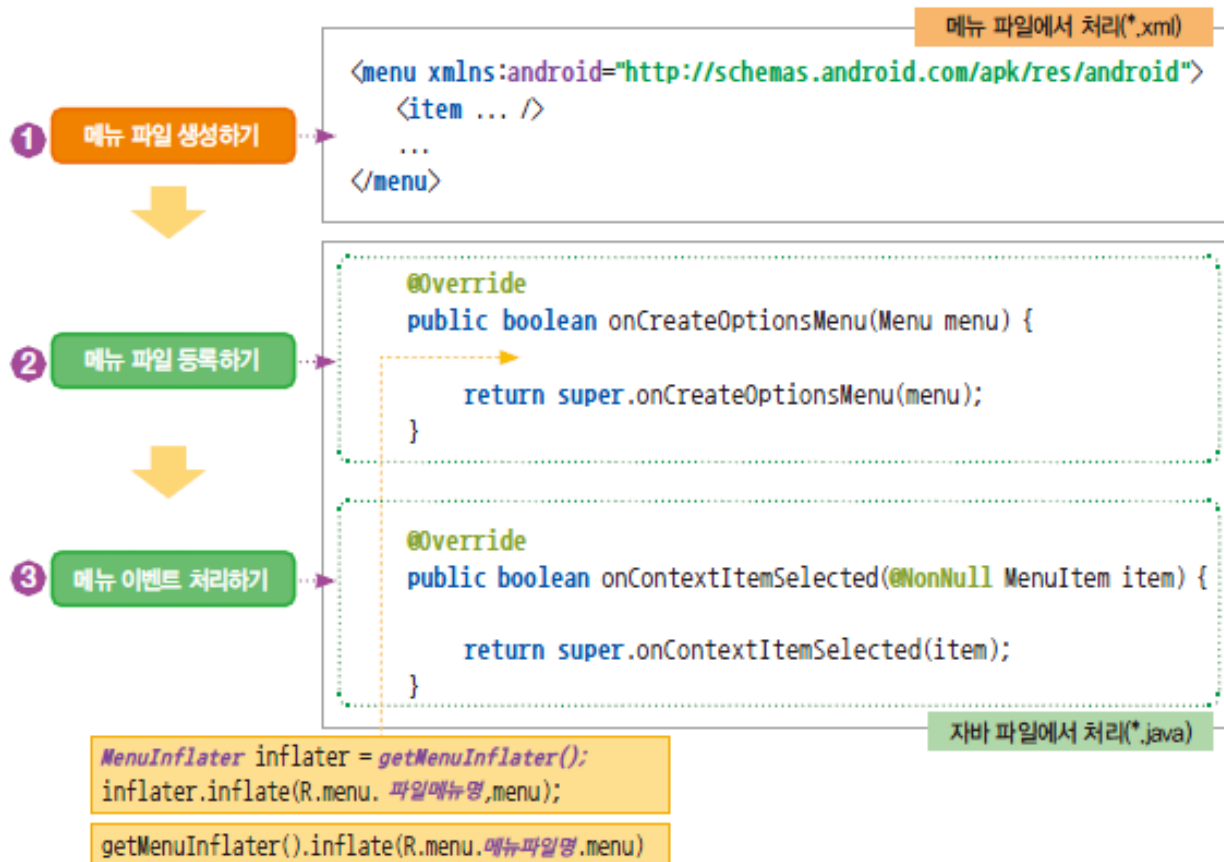
그림 7-3 옵션 메뉴 화면 예



2. 기본 유형 메뉴 만들기

■ 옵션 메뉴

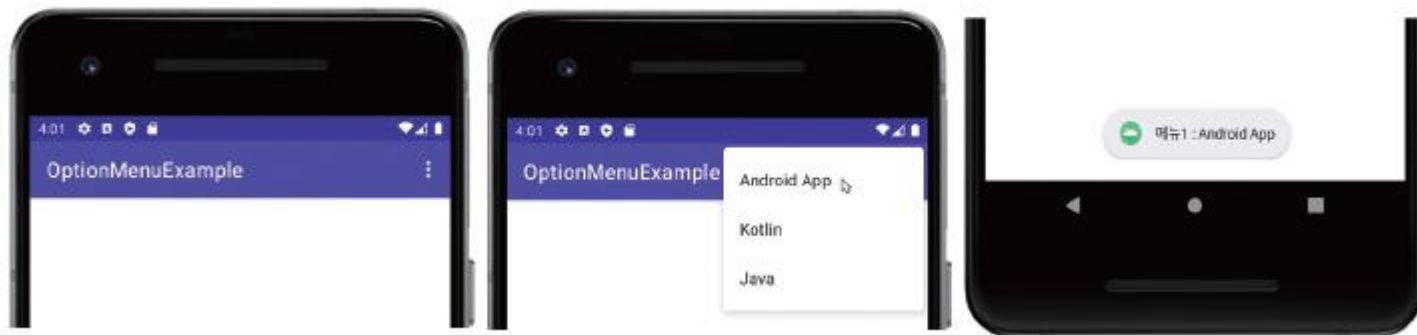
그림 7-4 옵션 메뉴 생성 과정



2. 기본 유형 메뉴 만들기

■ 옵션 메뉴

그림 7-5 옵션 메뉴 생성 결과

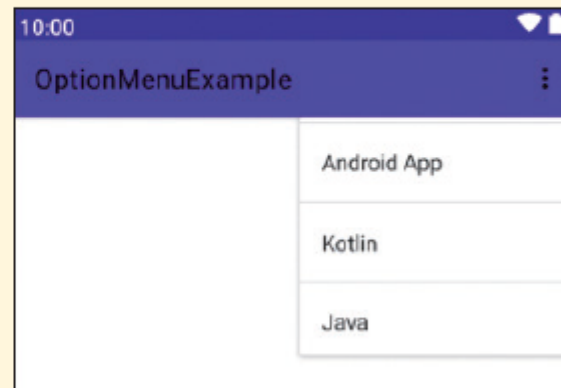


2. 기본 유형 메뉴 만들기

■ 옵션 메뉴

- 메뉴 XML 파일 생성하고 메뉴 항목 정의하기

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/menu01"
        android:title="Android App" />
    <item
        android:id="@+id/menu02"
        android:title="Kotlin" />
    <item
        android:id="@+id/menu03"
        android:title="Java" />
</menu>
```

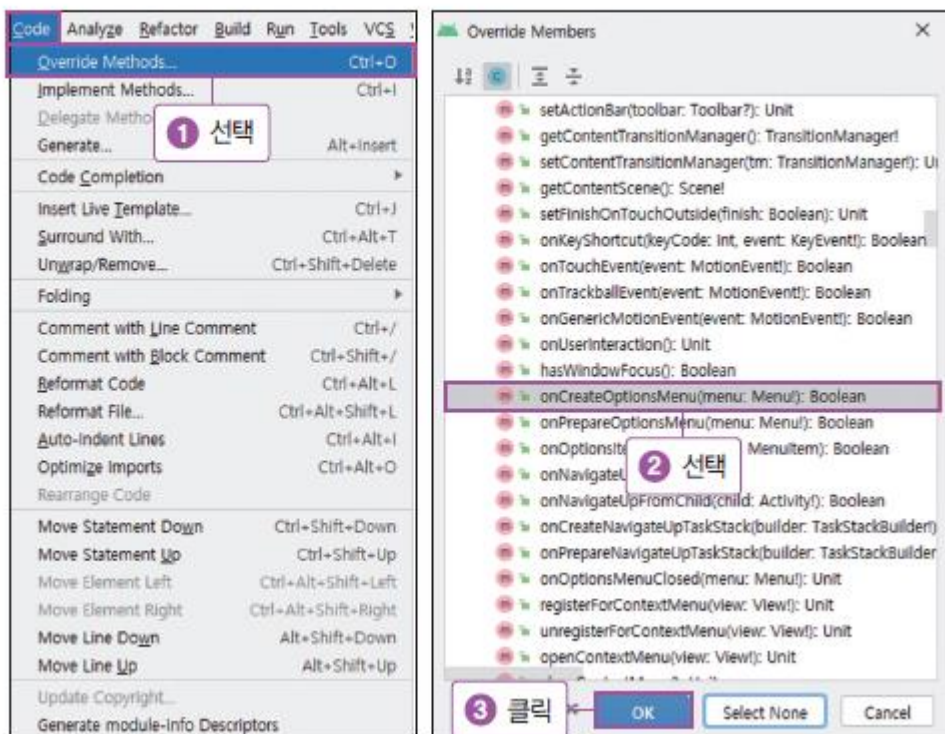


2. 기본 유형 메뉴 만들기

■ 옵션 메뉴

- 옵션 메뉴 XML 파일 등록하기

그림 7-6 onCreateOptionsMenu() 메서드 재정의



2. 기본 유형 메뉴 만들기

■ 옵션 메뉴

- 옵션 메뉴 XML 파일 등록하기

```
import android.view.MenuInflater;

public class MainActivity extends AppCompatActivity {
    ...

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

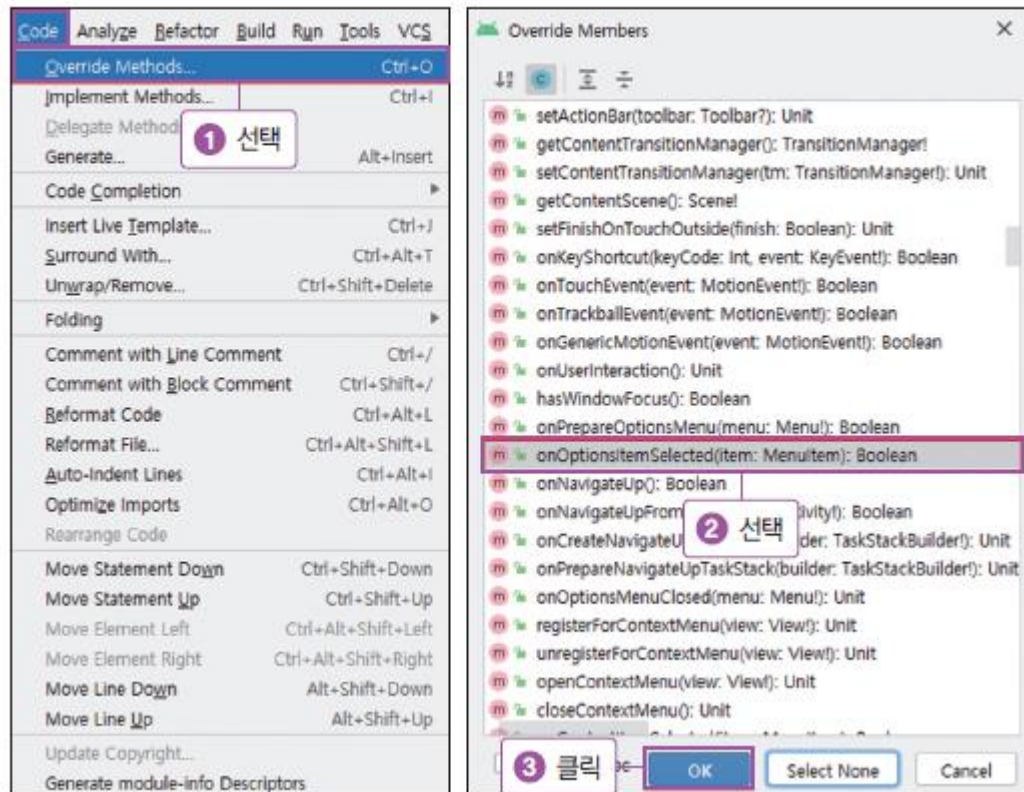


2. 기본 유형 메뉴 만들기

■ 옵션 메뉴

- 옵션 메뉴의 클릭 이벤트 처리하기

그림 7-7 onOptionsItemSelected() 메서드 재정의



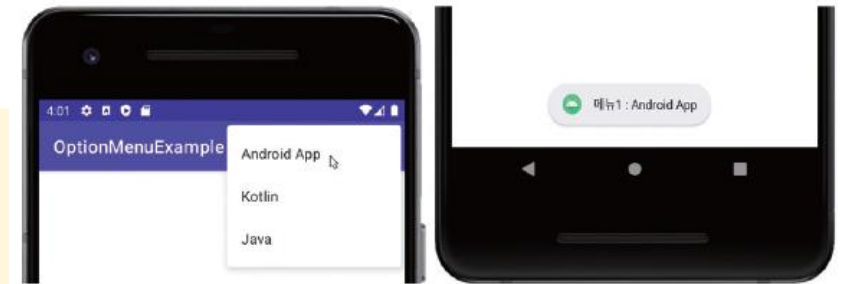
2. 기본 유형 메뉴 만들기

■ 옵션 메뉴

- 옵션 메뉴의 클릭 이벤트 처리하기

```
import android.view.MenuItem;

public class MainActivity extends AppCompatActivity {
    ...
    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        switch (item.getItemId()) {
            case R.id.menu01 :
                Toast.makeText(getApplicationContext(), " 메뉴1 : " + item.getTitle(),
                    Toast.LENGTH_SHORT).show();
                break;
            case R.id.menu02 :
                Toast.makeText(getApplicationContext(), " 메뉴2 : " + item.getTitle(),
                    Toast.LENGTH_SHORT).show();
                break;
            case R.id.menu03 :
                Toast.makeText(getApplicationContext(), " 메뉴3 : " + item.getTitle(),
                    Toast.LENGTH_SHORT).show();
                break;
        }
        return super.onOptionsItemSelected(item);
    }
}
```



2. 기본 유형 메뉴 만들기

■ 컨텍스트 메뉴

- 사용자가 특정 위젯을 길게 클릭하면 화면에 표시되는 플로팅 메뉴
- 선택한 콘텐츠나 컨텍스트 프레임에 영향을 주는 작업을 구현할 때 유용
- 윈도우 환경에서 마우스 오른쪽 버튼을 클릭하면 표시되는 메뉴와 비슷한 역할

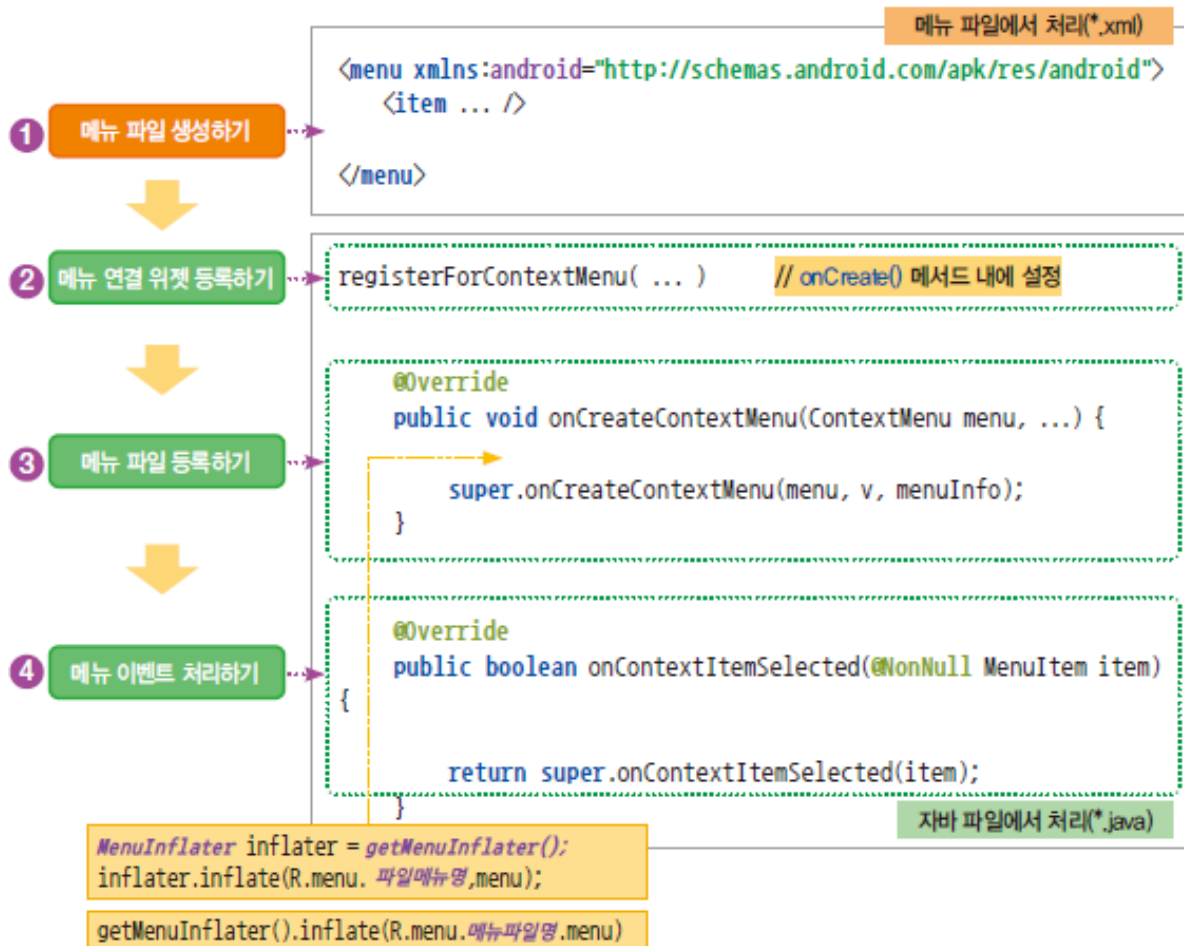
그림 7-8 컨텍스트 메뉴의 화면 예



2. 기본 유형 메뉴 만들기

■ 컨텍스트 메뉴

그림 7-9 컨텍스트 메뉴 생성 과정



2. 기본 유형 메뉴 만들기

■ 컨텍스트 메뉴

그림 7-10 컨텍스트 메뉴 생성 결과

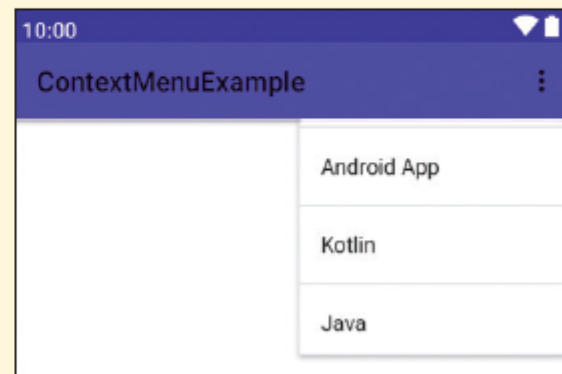


2. 기본 유형 메뉴 만들기

■ 컨텍스트 메뉴

- 메뉴 XML 파일 생성하고 메뉴 항목 만들기

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/menu01"
        android:title="Android App" />
    <item
        android:id="@+id/menu02"
        android:title="Kotlin" />
    <item
        android:id="@+id/menu03"
        android:title="Java" />
</menu>
```



2. 기본 유형 메뉴 만들기

■ 컨텍스트 메뉴

- 메뉴 연결을 위한 위젯 등록하기

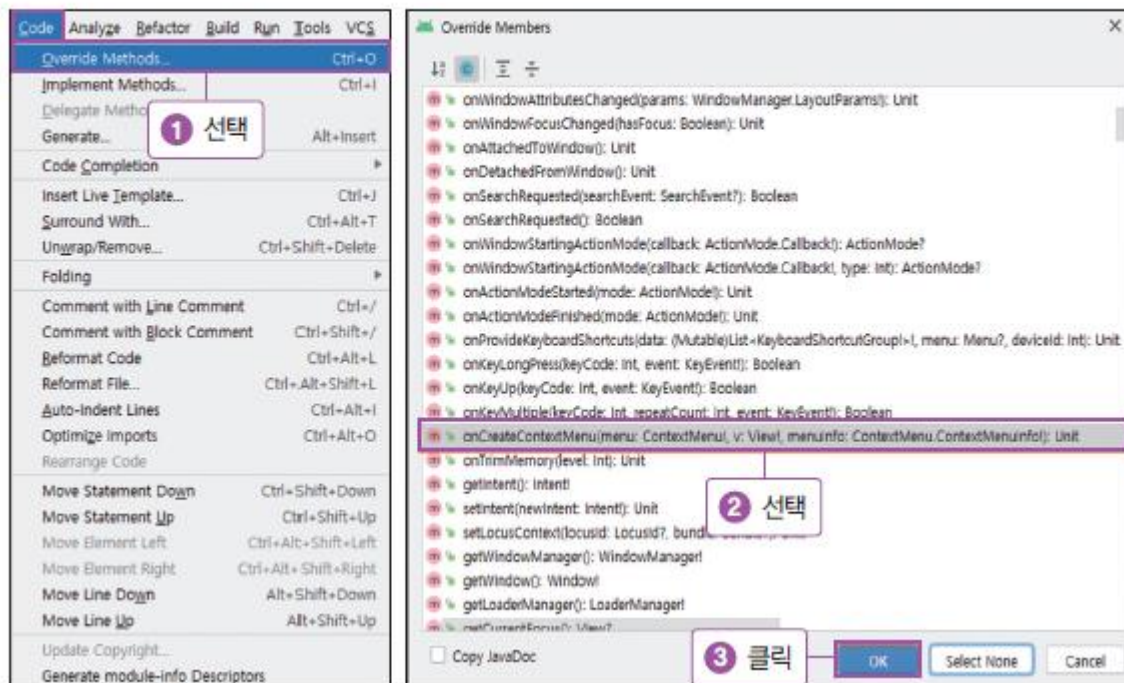
```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        Button btn = (Button) findViewById(R.id.button);  
        registerForContextMenu(btn);  
    }  
}
```

2. 기본 유형 메뉴 만들기

■ 컨텍스트 메뉴

- 컨텍스트 메뉴 XML 파일 등록하기

그림 7-11 onCreateContextMenu() 메서드 재정의

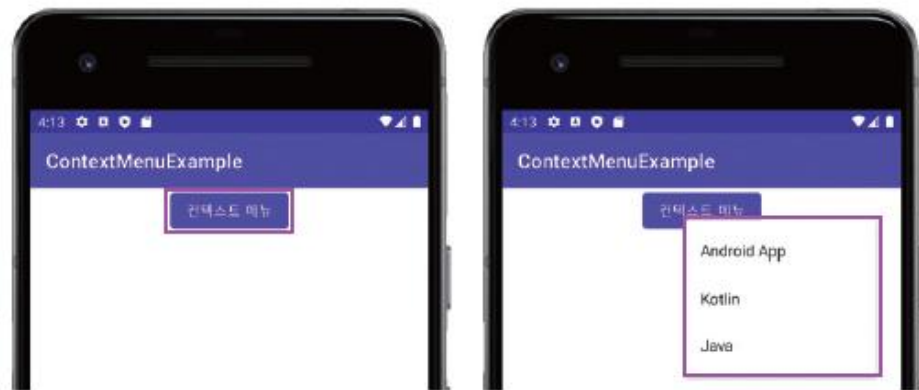


2. 기본 유형 메뉴 만들기

```
import android.view.MenuInflater;

public class MainActivity extends AppCompatActivity {
    ...

    @Override
    public void onCreateContextMenu(ContextMenu menu, View v, ContextMenu.
ContextMenuInfo menuInfo) {
        MenuInflater inflater = getMenuInflater();
        inflater.inflate(R.menu.context_menu, menu);
        super.onCreateContextMenu(menu, v, menuInfo);
    }
}
```



2. 기본 유형 메뉴 만들기

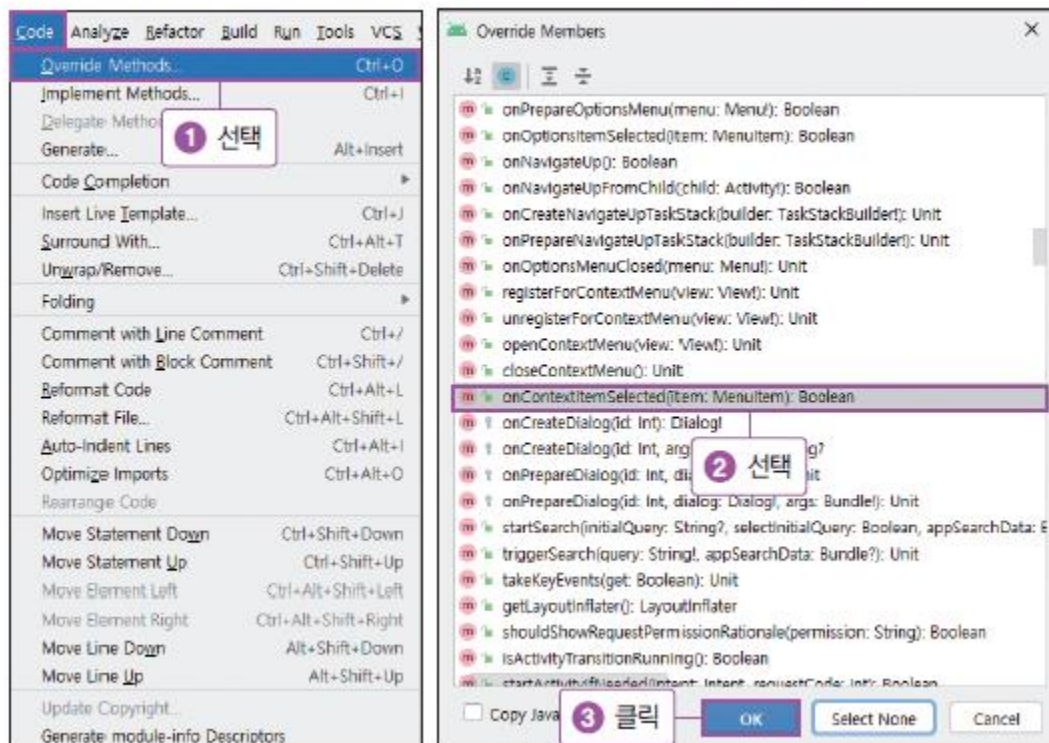
■ 컨텍스트 메뉴

- 컨텍스트 메뉴의 클릭 이벤트 처리하기

그림 7-12

onContextItemSelected()

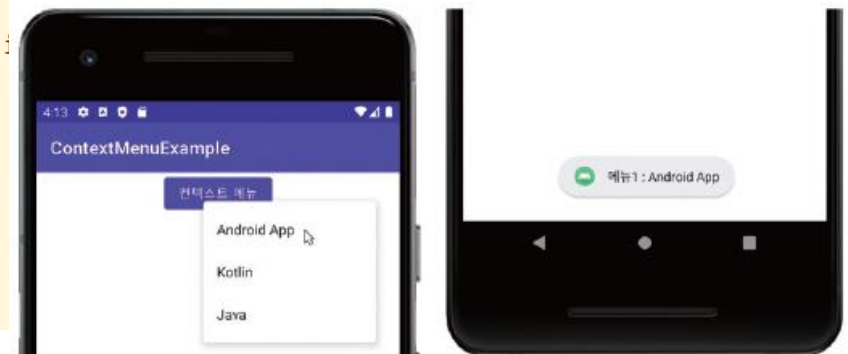
메서드 재정의



2. 기본 유형 메뉴 만들기

```
import android.view.MenuItem;

public class MainActivity extends AppCompatActivity {
    ...
    @Override
    public boolean onOptionsItemSelected(@NonNull MenuItem item) {
        switch (item.getItemId()) {
            case R.id.menu01 :
                Toast.makeText(getApplicationContext(), " 메뉴1 : " + item.getTitle(),
                    Toast.LENGTH_SHORT).show();
                break;
            case R.id.menu02 :
                Toast.makeText(getApplicationContext(), " 메뉴2 : " + item.getTitle(),
                    Toast.LENGTH_SHORT).show();
                break;
            case R.id.menu03 :
                Toast.makeText(getApplicationContext(), " 메뉴3 : " + item.getTitle(),
                    Toast.LENGTH_SHORT).show();
                break;
        }
        return super.onOptionsItemSelected(item);
    }
}
```



2. 기본 유형 메뉴 만들기

■ 팝업 메뉴

- 호출한 위젯 영역에 고정된 수직 목록 형태로 메뉴를 표시하는 것
- 특정 내용과 관련된 오버플로우(overflow) 작업을 구현할 때 유용
- 화면 공간이 있으면 : 호출한 위젯 아래에 표시, 공간이 부족하면 : 위젯 위에 표시
- 팝업 메뉴 외부를 클릭하면 자동으로 닫힘

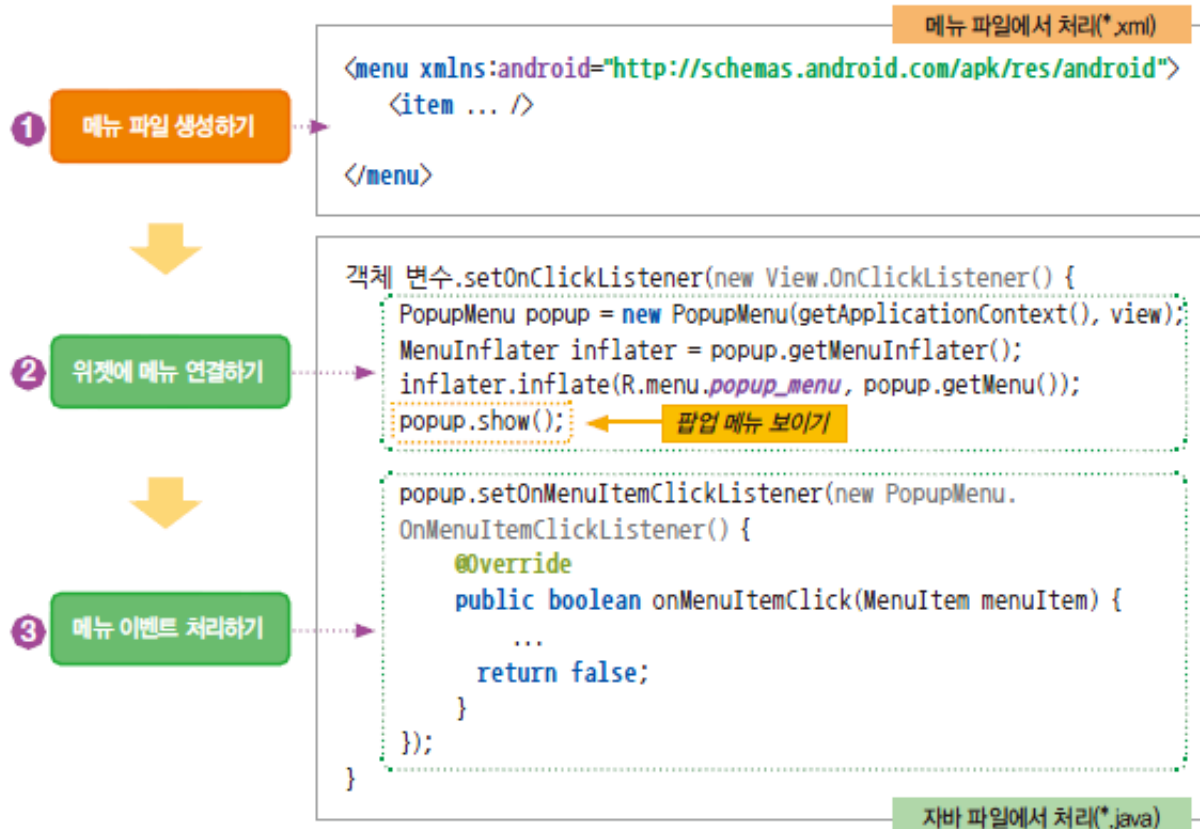
그림 7-13 팝업 메뉴 화면 예



2. 기본 유형 메뉴 만들기

■ 팝업 메뉴

그림 7-14 팝업 메뉴 생성 과정



2. 기본 유형 메뉴 만들기

■ 팝업 메뉴

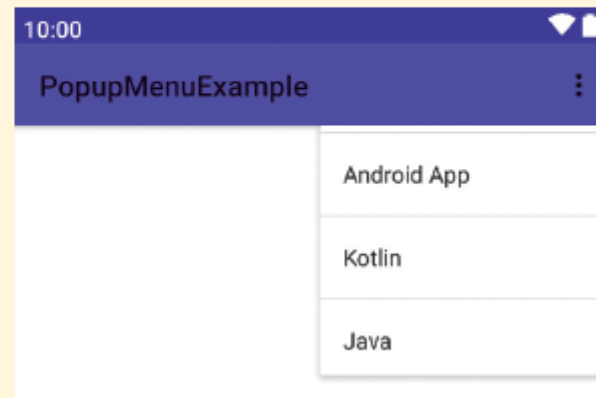


2. 기본 유형 메뉴 만들기

■ 팝업 메뉴

- 메뉴 XML 파일 생성하고 메뉴 항목 만들기

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/menu01"
        android:title="Android App" />
    <item
        android:id="@+id/menu02"
        android:title="Kotlin" />
    <item
        android:id="@+id/menu03"
        android:title="Java" />
</menu>
```



2. 기본 유형 메뉴 만들기

■ 팝업 메뉴

- 버튼에 메뉴 연결하기

```
import android.widget.PopupMenu;

public class MainActivity extends AppCompatActivity {

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

        Button btn = (Button) findViewById(R.id.button);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                PopupMenu popup = new PopupMenu(getApplicationContext(), btn); ❶
                popup.inflate(R.menu.popup_menu); ❷

                popup.show(); ❸
            }
        });
    }
}
```



2. 기본 유형 메뉴 만들기

■ 팝업 메뉴

- 팝업 메뉴 클릭 이벤트 처리하기

```
import android.widget.MenuItem;

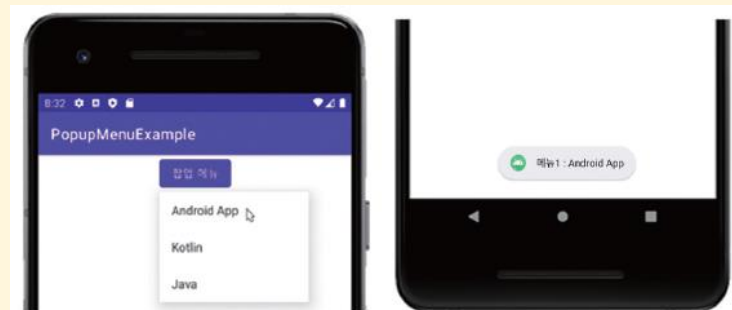
public class MainActivity extends AppCompatActivity {

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

        Button btn = (Button) findViewById(R.id.button);
        btn.setOnClickListener(new View.OnClickListener() {

            @Override
            public void onClick(View view) {
                PopupMenu popup = new PopupMenu(getApplicationContext(), btn);
                popup.inflate(R.menu.popup_menu);
                popup.show();
                popup.setOnMenuItemClickListener(new PopupMenu.OnMenuItemClickListener() {
                    @Override
                    public boolean onMenuItemClick(MenuItem menuItem) {
```

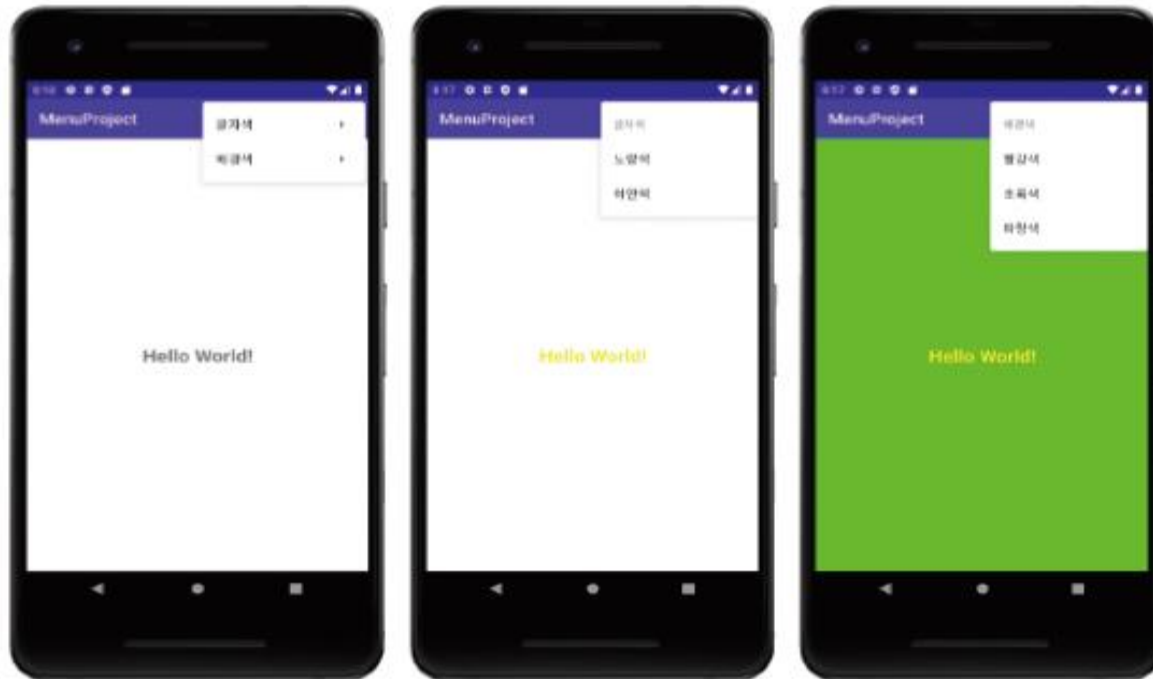
```
                    switch (menuItem.getItemId()) {
                        case R.id.menu01 :
                            Toast.makeText(getApplicationContext(), " 메뉴1 : " + menuItem.
                                getTitle(), Toast.LENGTH_SHORT).show();
                            break;
                        case R.id.menu02 :
                            Toast.makeText(getApplicationContext(), " 메뉴2 : " + menuItem.
                                getTitle(), Toast.LENGTH_SHORT).show();
                            break;
                        case R.id.menu03 :
                            Toast.makeText(getApplicationContext(), " 메뉴3 : " + menuItem.
                                getTitle(), Toast.LENGTH_SHORT).show();
                            break;
                    }
                    return false;
                });
            }
        });
    }
}
```



2. 기본 유형 메뉴 만들기

■ [실습 예제 7-1] 메뉴로 글자색과 배경색 변경하기

- 구현 내용 : 메뉴 항목으로 글자색과 배경색을 넣어 옵션 메뉴를 만듭니다. 글자색 항목은 하위 메뉴로 노란색과 하얀색, 배경색 항목은 하위 메뉴로 빨간색, 초록색, 파란색을 구성합니다. 글자색 항목을 선택하면 글자색을 변경하고, 배경색 항목을 선택하면 레이아웃 배경색을 변경합니다



03

앱 바 다루기

3. 앱 바 다루기

■ 앱 바란

- 액티비티의 제목과 탐색 아이콘 또는 액션 버튼들로 구성된 작업 모음
- 앱 바를 사용하면 사용자에게 통일된 UI를 제공하고, 다른 항목에 좀 더 빠르게 접근 가능
- 앱 바의 주요 기능
 - 앱에 아이디를 설정해 정체성을 부여할 수 있는 전용 공간
 - 중요한 작업을 예측 가능한 방식(예 : 검색)으로 접근
 - 앱 안에서 탐색과 뷰의 전환(탭 또는 드롭다운 목록 포함)을 지원

3. 앱 바 다루기

■ 앱 바란

그림 7-15 앱 바의 레이아웃 구조

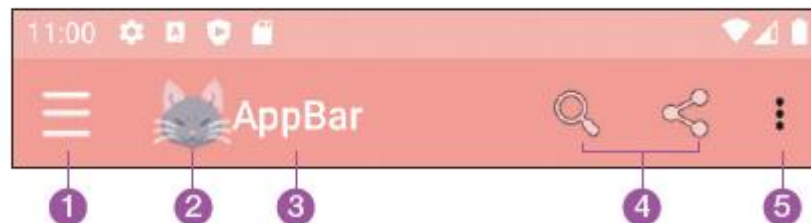


표 7-6 앱 바의 구성 요소

구성 요소	설명
① 탐색 아이콘	상위 액티비티로 이동하는 화살표 또는 열기/닫기로 전환하는 탐색 메뉴 전환 아이콘을 표시합니다.
② 로고 이미지	앱의 브랜드나 로고 이미지를 표시합니다.
③ 제목	앱의 브랜드 이름 또는 액티비티나 사용자 방문 페이지 정보를 표시합니다.
④ 액션 버튼	액션 메뉴 또는 앱의 가장 중요한 작업을 표시합니다.
⑤ 액션 메뉴(메뉴 버튼)	설정, 인쇄, 저장, 프로필 등의 몇 가지 주요 작업을 사용자에게 제공하는 도구 모음입니다.

3. 앱 바 다루기

■ 액션 바

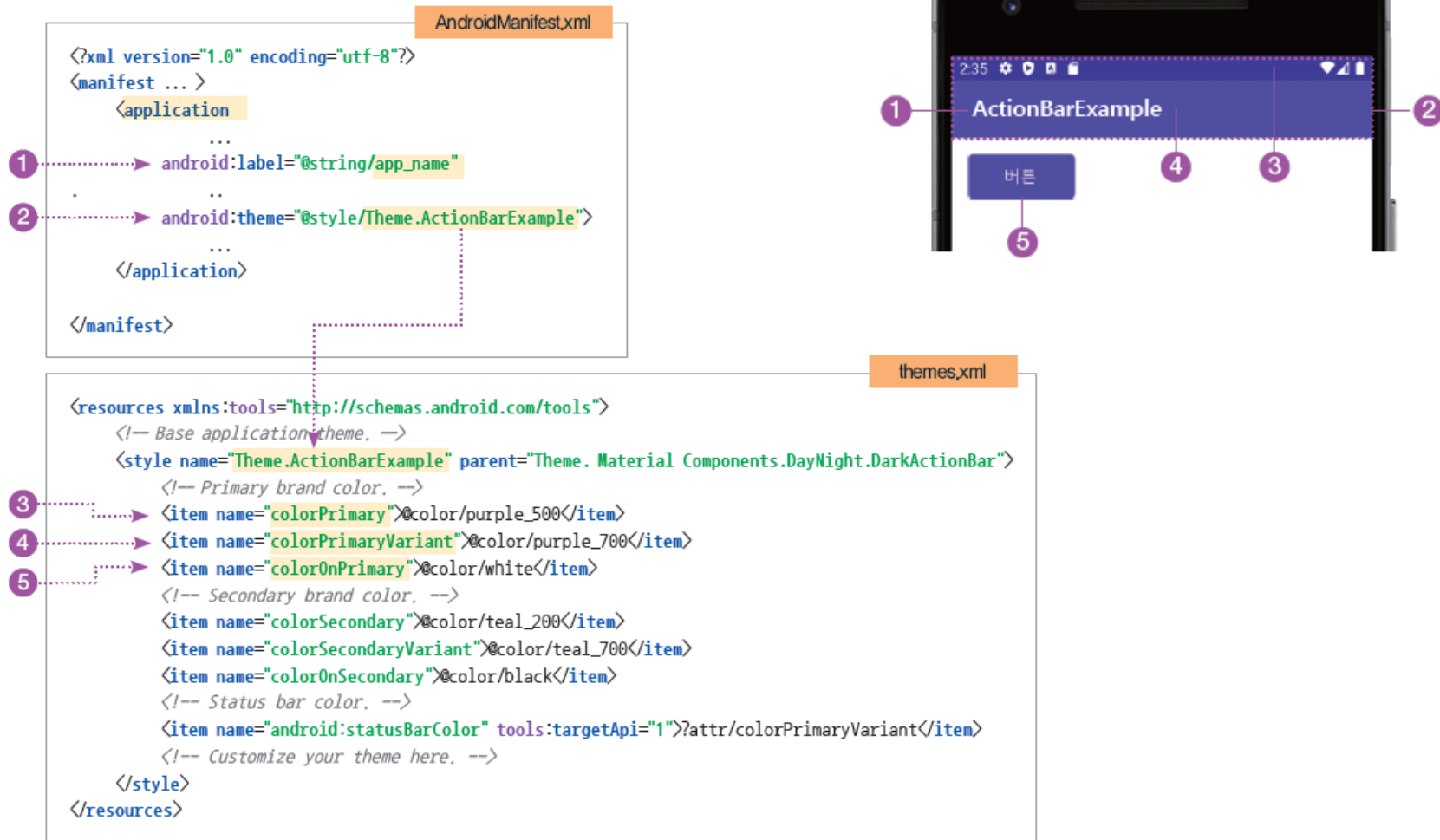
- 기본적으로 안드로이드 액티비티는 액션 바를 기본 테마로 사용
- 따라서 레이아웃 파일에 별도의 위젯 추가 없이도 모든 액티비티에 액션 바 포함
- 앱 바를 액티비티에 표시하려면 앱 바를 표시하는 테마를 설정
- 안드로이드 프로젝트를 생성하면 액티비티 화면 상단에 액션 바 표시

그림 7-16 액티비티에서 액션 바 형태의 예



3. 앱 바 다루기

그림 7-17 AndroidManifest.xml과 themes.xml 파일의 액션 바 설정 관계

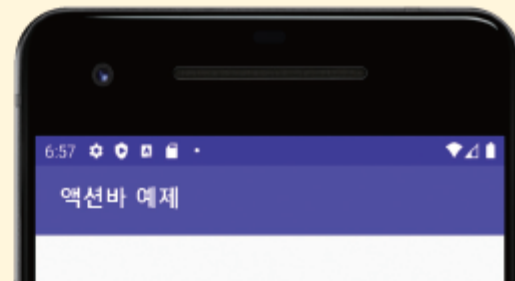


3. 앱 바 다루기

■ 액션 바

- 제목의 텍스트 변경하기

```
<manifest...>  
  <application  
    ...  
    android:label="액션 바 예제"  
    ...  
  </application>  
</manifest>
```



3. 앱 바 다루기

■ 액션 바

■ 액션 버튼 추가하기

```
<?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/edit"
    android:icon="@android:drawable/ic_menu_edit"
    android:title="편집하기" />
  <item
    android:id="@+id/search"
    android:icon="@android:drawable/ic_menu_search"
    android:title="검색하기"
    app:showAsAction="ifRoom" />
  <item
    android:id="@+id/save"
    android:icon="@android:drawable/ic_menu_save"
    android:title="저장하기" />
</menu>
```

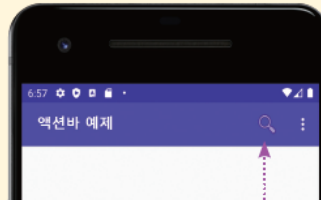


표 7-8 showAsAction 속성값

속성값	설명
always	메뉴 항목을 항상 액션 바에 액션 버튼으로 표시합니다. never와 ifRoom보다 우선이며, 공간이 없으면 표시되지 않습니다.
never	메뉴 항목을 액션 바에 액션으로 표시하지 않고 오버플로우 메뉴에 바로 표시합니다.
ifRoom	액션 바에 공간이 있으면 메뉴 항목을 액션 버튼으로 표시하고, 공간이 없으면 오버플로우 메뉴에 표시합니다.
withText	메뉴 항목을 액션 바에 액션 버튼으로 표시할 때 텍스트와 함께 표시합니다. 단, 아이콘과 텍스트를 함께 표시할 공간이 있는 경우에만 텍스트를 표시합니다.
collapseActionView	메뉴 항목을 접었다 펼쳤다 하면서 위젯을 표시합니다.

```
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) { // [옵션 메뉴] 참고
```

```
        MenuInflater inflater = getMenuInflater();
```

```
        inflater.inflate(R.menu.menu, menu);
```

```
        return super.onCreateOptionsMenu(menu);
```

```
    }
```

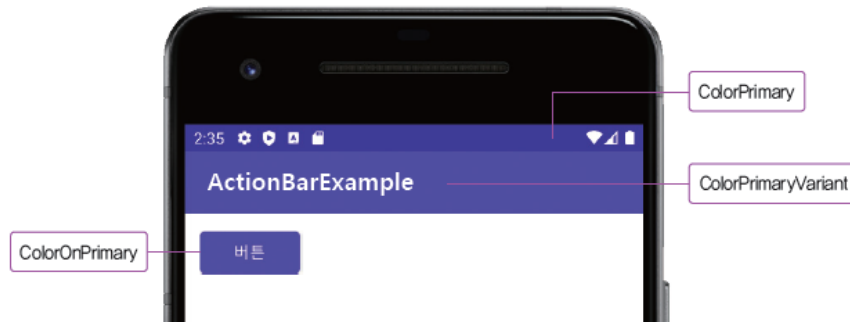
```
}
```

3. 앱 바 다루기

■ 액션 바

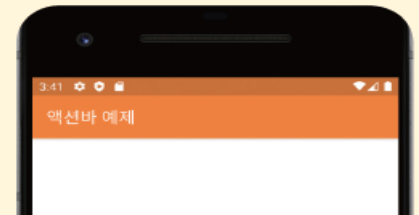
■ 액션 바 색상 변경하기

그림 7-19 액션 바 색상값 적용 위치



```
<resources xmlns:tools="http://schemas.android.com/tools">
  <!-- Base application theme. -->
  <style name="Theme.ActionBarExample" parent="Theme.MaterialComponents.DayNight.
DarkActionBar">
    <!-- Primary brand color. -->

    <item name="colorPrimary">#F6852C</item>
    <item name="colorPrimaryVariant">#D17127</item>
    <item name="colorOnPrimary">@color/white</item>
    <!-- Secondary brand color. -->
    ....
  </style>
</resources>
```



3. 앱 바 다루기

■ 액션 바

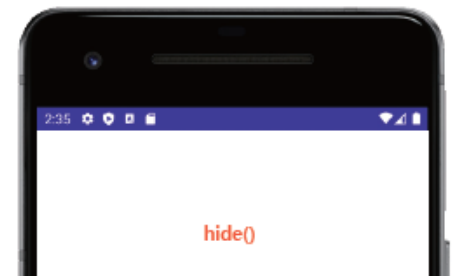
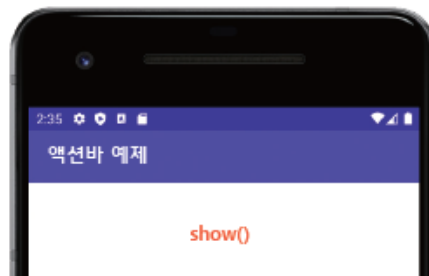
- 액션 바 숨기기와 보이기

```
<resources xmlns:tools="http://schemas.android.com/tools">
    <!-- Base application theme. -->
    <style name="Theme.ActionBarExample" parent="Theme.AppCompat.Light.NoActionBar">
        <item name="colorPrimary">@color/purple_500</item>

        ...
    </style>
</resources>
```

```
getSupportActionBar().hide(); // 액션 바 숨기기
```

```
getSupportActionBar().show(); // 액션 바 보이기
```

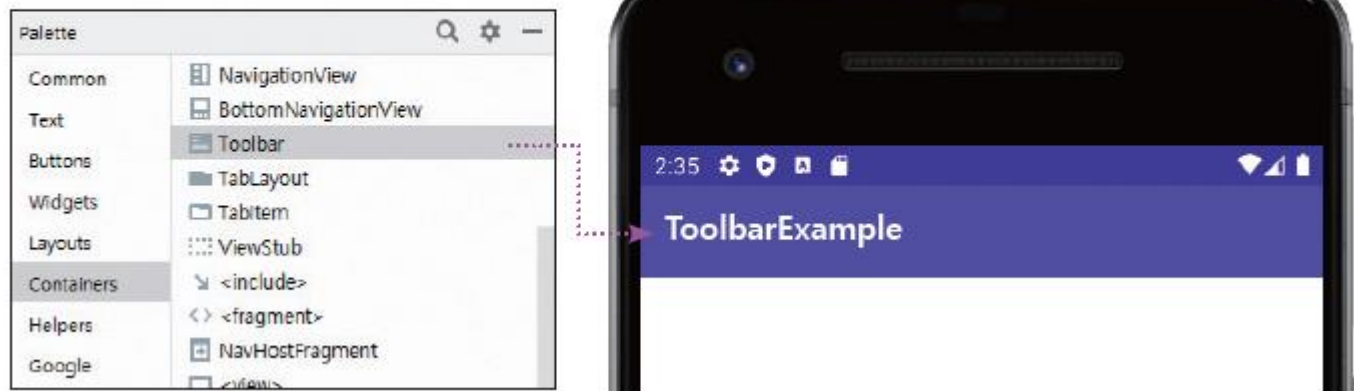


3. 앱 바 다루기

■ 툴 바

- 액션 바와 달리 액티비티 상단의 고정된 위치가 아닌 레이아웃 내 어디든 배치
- 색상, 크기, 위치 등을 더욱 유연하게 수정 가능
- 탐색 아이콘, 로고, 제목, 액션 버튼 등을 추가 가능

그림 7-20 툴 바의 화면 표시 예



3. 앱 바 다루기

■ 툴 바

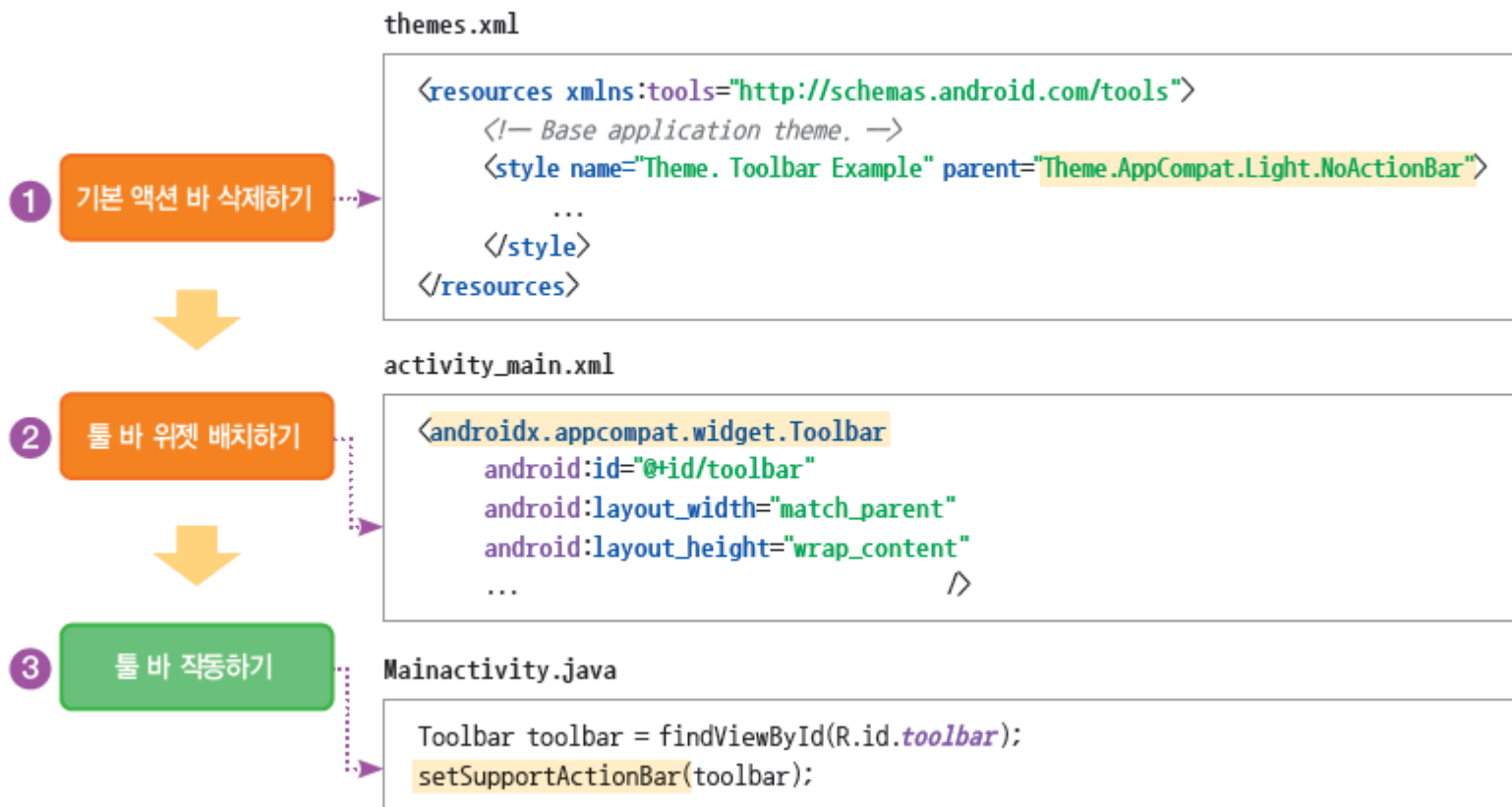
- 툴 바를 사용하는 경우
 - 기존의 액션 바 기능(예 : 메뉴 확장과 선택, ActionBarDrawerToggle 등)을 사용하면서 모양을 제어하고 싶을 때
 - 액션 바를 지원하지 않는 상황에서 앱에서 앱 바를 사용하고 싶을 때
- 액션 바와 다른 툴 바의 특징
 - 툴 바는 기본 위젯과 마찬가지로 레이아웃에 포함된 위젯
 - 툴 바는 일반 위젯으로서 배치, 애니메이션과 제어가 더 쉬움
 - 하나의 앱에서 여러 개의 개별 툴 바를 정의 가능

3. 앱 바 다루기

■ 툴 바

■ 툴 바 만들기

그림 7-21 툴 바 생성 과정



3. 앱 바 다루기

■ 툴 바

- 기본 액션 바 삭제하기

```
<style name="Theme.ToolbarExample" parent="Theme.AppCompat.Light.NoActionBar">
    <!-- Primary brand color. -->
    ...
    <!-- Customize your theme here. -->
</style>
```

- 툴바 위젯 배치하기

```
<androidx.appcompat.widget.Toolbar
    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?attr/colorPrimary"
    android:minHeight="?attr/actionBarSize"
    android:theme="?attr/actionBarTheme"/>
```

3. 앱 바 다루기

■ 툴 바

- 툴바 적용하기

```
import androidx.appcompat.widget.Toolbar;

public class MainActivity extends AppCompatActivity {

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

        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
    }
}
```

3. 앱 바 다루기

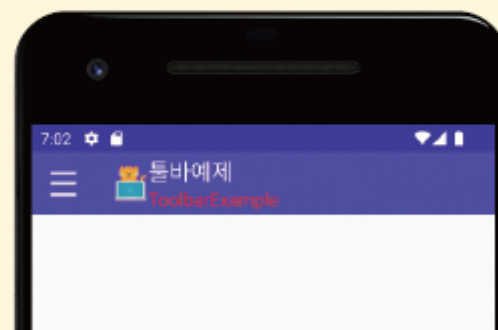
■ 툴 바

- XML로 속성 설정하기

```
<androidx.appcompat.widget.Toolbar
    android:id="@+id/toolbar"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="?attr/colorPrimary"
    android:minHeight="?attr/actionBarSize"
    android:theme="?attr/actionBarTheme"
    app:logo="@drawable/working"
    app:navigationIcon="@drawable/menu"
    app:subtitle="ToolbarExample"
    app:subtitleTextColor="#FF0000"
    app:title="툴바예제"
    app:titleTextColor="#FFFFFF" />
```

표 7-9 툴 바의 속성

속성	설명
id	툴 바를 식별하는 데 사용하는 고유한 아이디를 설정합니다.
logo	툴 바에 표시할 로고 이미지를 설정합니다.
navigationIcon	툴 바에 표시할 탐색 아이콘 이미지를 설정합니다.
title	툴 바에 표시할 제목을 설정합니다.
titleTextColor	툴 바에 표시할 제목의 색상을 설정합니다.
subtitle	툴 바에 표시할 서브 제목을 설정합니다.
subtitleTextColor	툴 바에 표시할 서브 제목 색상을 설정합니다.



3. 앱 바 다루기

■ 툴 바

■ 메서드로 속성 설정하기

표 7-10 툴 바의 주요 속성 메서드

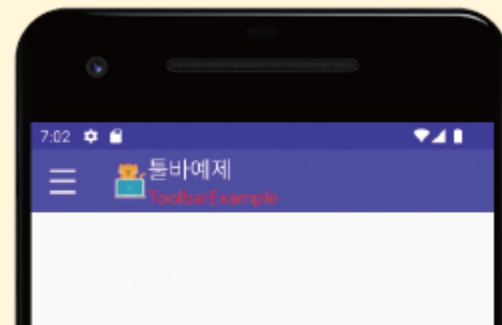
메서드	설명
<code>void setLogo(int resId)</code>	resId로 지정된 이미지를 툴 바의 로고로 설정합니다.
<code>void setLogo(Drawable drawable)</code>	drawable 객체로 지정된 이미지를 툴 바의 로고로 설정합니다.
<code>void getLogo()</code>	툴 바가 표시하는 로고 이미지를 가져옵니다.
<code>void setNavigationIcon(int resId)</code>	resId로 지정된 이미지를 탐색 버튼의 아이콘 이미지로 설정합니다.
<code>void setNavigationIcon(Drawable icon)</code>	Icon 객체로 지정된 이미지를 탐색 버튼의 아이콘 이미지로 설정합니다.
<code>void getNavigationIcon()</code>	툴 바가 표시하는 탐색 버튼의 아이콘을 가져옵니다.
<code>void setTitle(CharSequence title)</code>	title로 지정된 문자열을 툴 바의 제목으로 설정합니다.
<code>void setSubtitle(int resId)</code>	resId로 지정된 문자열(string.xml에 저장된 문자열)을 툴 바의 부제목으로 설정합니다.
<code>void setSubtitle(CharSequence subtitle)</code>	subtitle로 지정된 문자열을 툴 바의 부제목으로 설정합니다.
<code>void getTitle()</code>	툴 바가 표시하는 제목을 가져옵니다.
<code>void getSubtitle()</code>	툴 바가 표시하는 부제목을 가져옵니다.
<code>void setTitleTextColor(int color)</code>	color로 지정된 색상을 툴 바의 제목에 설정합니다.
<code>void setSubtitleTextColor(int color)</code>	color로 지정된 색상을 툴 바의 부제목에 설정합니다.
<code>void setNavigationOnClickListener(View.OnClickListener listener)</code>	툴 바에서 탐색 아이콘 버튼의 클릭 이벤트를 처리하는 데 사용합니다.

3. 앱 바 다루기

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);

        Toolbar toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        toolbar.setLogo(R.drawable.working);           // 로고 이미지 표시
        toolbar.setNavigationIcon(R.drawable.menu);    // 탐색 아이콘 표시
        toolbar.setTitle("툴바예제");                  // 제목 표시
        toolbar.setSubtitle("ToolbarExample");         // 부제목 표시
        toolbar.setTitleTextColor(Color.WHITE);        // 제목 색상 표시
        toolbar.setSubtitleTextColor(Color.RED);       // 부제목 색상 표시
    }
}
```



3. 앱 바 다루기

■ [실습 예제 7-2] 트위터 톨 바 만들기

- 구현 내용 : 다음과 같이 트위터처럼 생긴 톨 바를 만들어 봅시다.

