



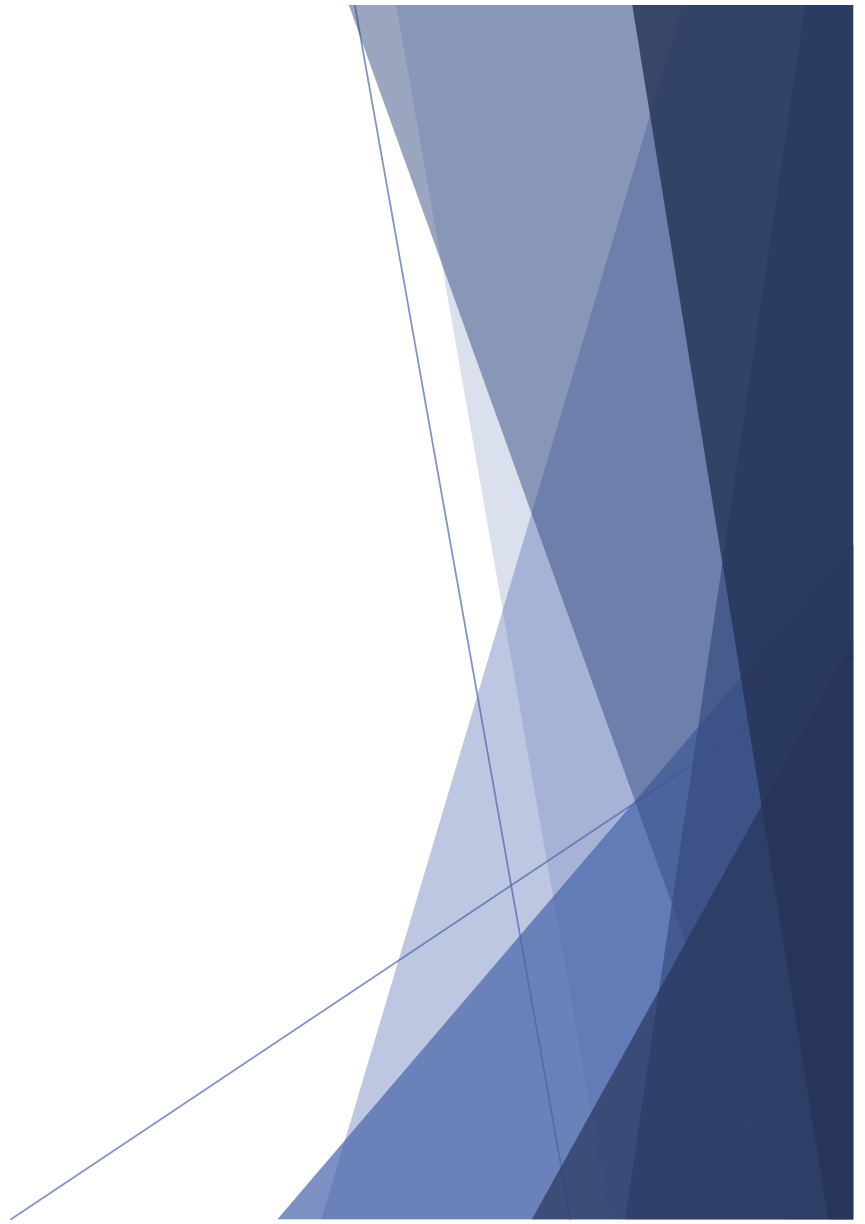
Android Programming

Menu와 ActionBar



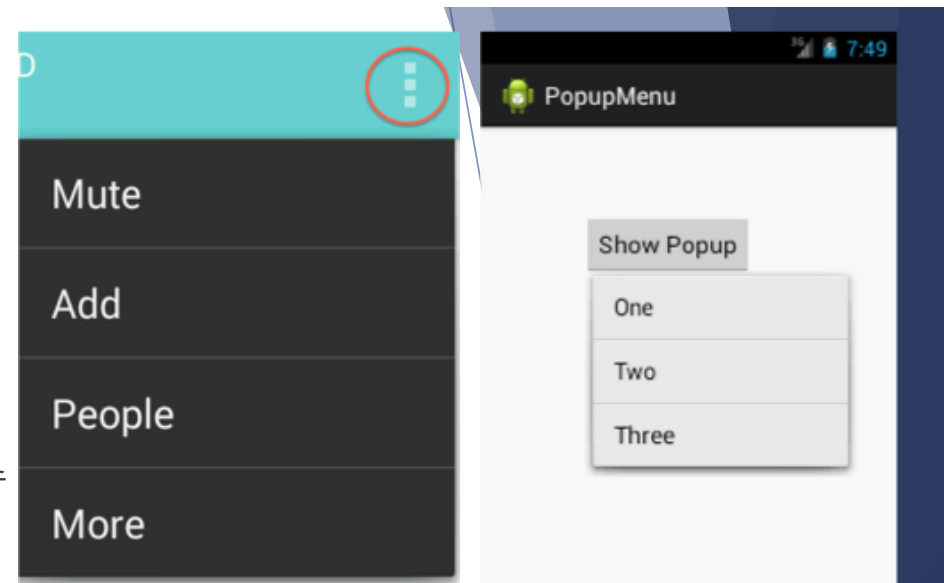
Menu

Options Menu와 Context Menu



Menu

- ▶ 메뉴는 대부분 애플리케이션에 필요한 요소
- ▶ 정형화된 UI를 제공, 사용자가 필요한 기능을 쉽게 이용할 수 있게 함
- ▶ 메뉴의 종류
 - ▶ 옵션 메뉴
: 기본적인 메뉴. 액션바 우측 메뉴 아이콘을 터치하면 나타남
 - ▶ 컨텍스트 메뉴
: 특정 뷰나 위젯을 길게 눌렀을 때 표시되는 메뉴
 - ▶ 서브 메뉴
: 옵션 메뉴나 컨텍스트 메뉴에 추가될 수 있는 보조 메뉴
특정 주제에 따라 메뉴를 그룹화 할 때 유용
- ▶ 메뉴 구현 방법
 - ▶ Activity 클래스 내에서 코드로 메뉴를 생성
 - ▶ XML 리소스를 활용 (/res/menu)



Options Menu

Context Menu

Menu

: 메뉴 구현 실습 (By Code)

[실습] OptionMenuExample

1) 옵션 메뉴 구현

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {

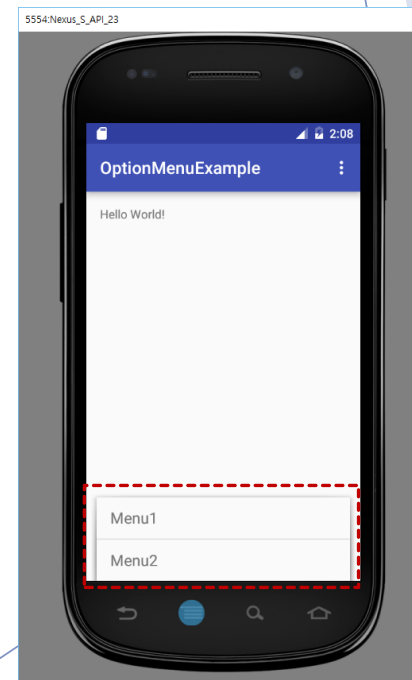
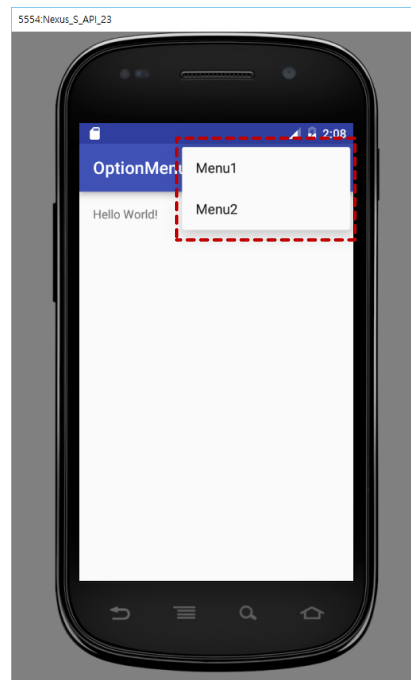
    menu.add(Menu.NONE, 0, 0, "Menu 1");
    menu.add(Menu.NONE, 1, 1, "Menu 2");

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

- onCreateOptionsMenu(Menu menu) 오버라이딩

- add 메서드 매개변수

- 1 : 그룹 아이디
- 2 : 아이템 아이디
- 3 : 순서
- 4 : 메뉴이름



Menu

: 메뉴 구현 실습 (By Code)

[실습] OptionMenuExample

2) 옵션 메뉴 선택 이벤트 처리

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case 0:
            Toast.makeText(this, "Menu 1 선택", Toast.LENGTH_LONG).show();
            break;
        case 1:
            Toast.makeText(this, "Menu 2 선택", Toast.LENGTH_LONG).show();
            break;
        default:
    }
}
```

– onOptionsItemSelected(MenuItem item) 오버라이딩

Android의 Inflater

: 정적으로 존재하는 XML을 실제 객체로 만들어 내는 것

Menu

: 메뉴 구현 실습 (By XML)

[실습] OptionMenuExample

3) XML를 이용한 메뉴 구현

- /res/menu 디렉터리에 main_menu.xml 생성

```
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item
        android:id="@+id/menu_refresh"
        android:title="refresh" />
    <item
        android:id="@+id/menu_search"
        android:title="search" />
</menu>
```

- onCreateOptionsMenu 및 onOptionsItemSelected 수정

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.main_menu, menu);

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

Menu

: 컨텍스트 메뉴 구현 실습

[실습] ContextMenuExample

1) 컨텍스트 메뉴 구현

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

    // Root View를 위한 Context Menu의 등록
    registerForContextMenu(findViewById(android.R.id.content));
}
```

- Activity의 onCreate 메서드에서 registerForContextMenu(View) 를 호출하여 컨텍스트 메뉴를 등록할 View를 설정한다.
- view 는 해당 컨텐스트 메뉴를 보여줄 위젯(View)를 넣어주면 된다.
- findViewById(android.R.id.content) 는 Root View 즉, 레이아웃을 찾는다.

Menu

: 컨텍스트 메뉴 구현 실습

[실습02] ContextMenuExample

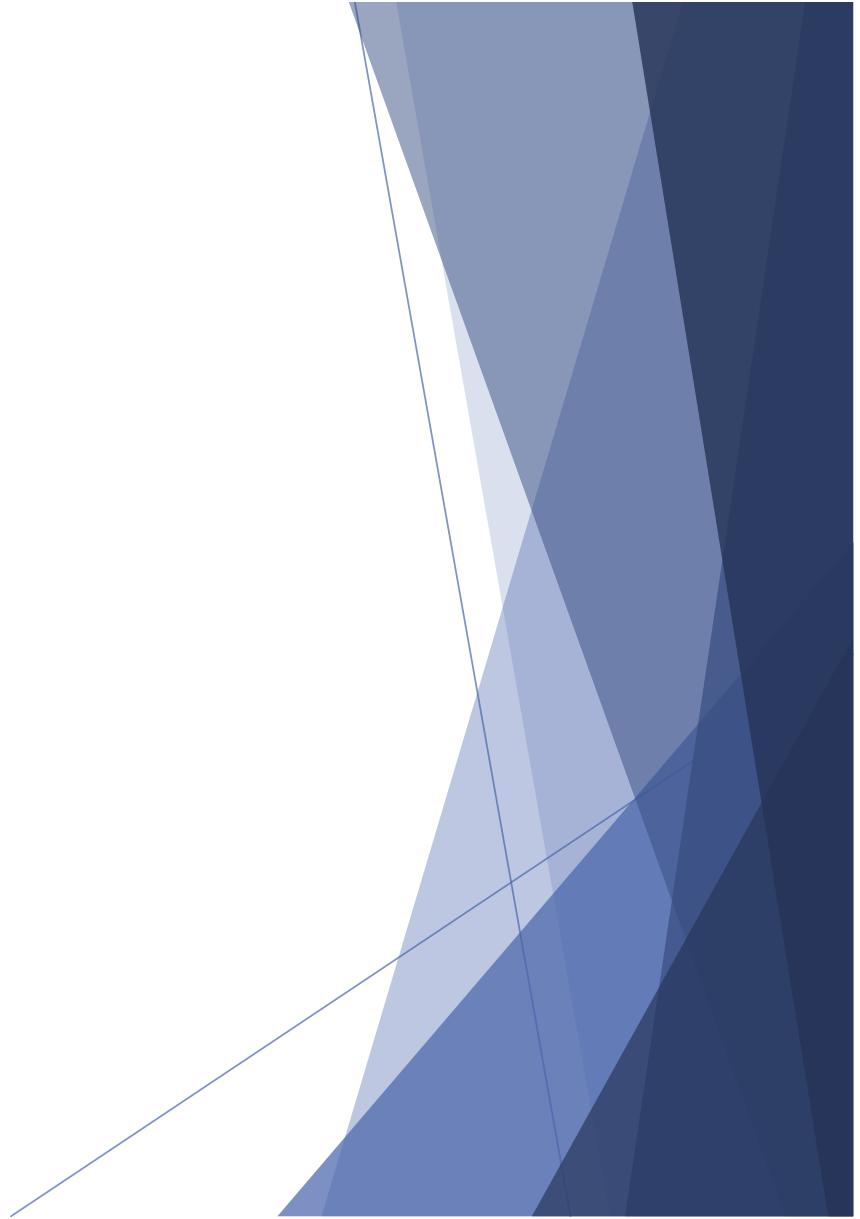
2) 컨텍스트 메뉴 생성

```
@Override  
public void onCreateContextMenu(ContextMenu menu,  
                                View v, ContextMenu.ContextMenuInfo menuInfo) {  
    getMenuInflater().inflate(R.menu.menu_context, menu);  
}
```

- onCreateContextMenu(...) 를 오버라이딩 한다.
- 옵션 메뉴와 마찬가지로 리소스에서 메뉴를 인플레이션 시킨다.
- 메뉴 아이템 선택한 대한 처리는 옵션메뉴와 동일하다.



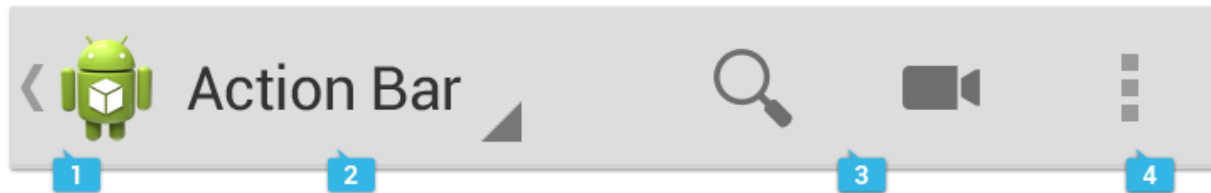
ActionBar



ActionBar

: ActionBar 기초 사용법

- ▶ HoneyComb(3.0) 이전의 타이틀바(TitleBar)를 대체하는 컴포넌트
- ▶ 메뉴 제공, 액션 아이템(ActionItem)을 이용한 단축 메뉴, 탭 지원 등 다양한 기능 제공
- ▶ 일반적인 구성

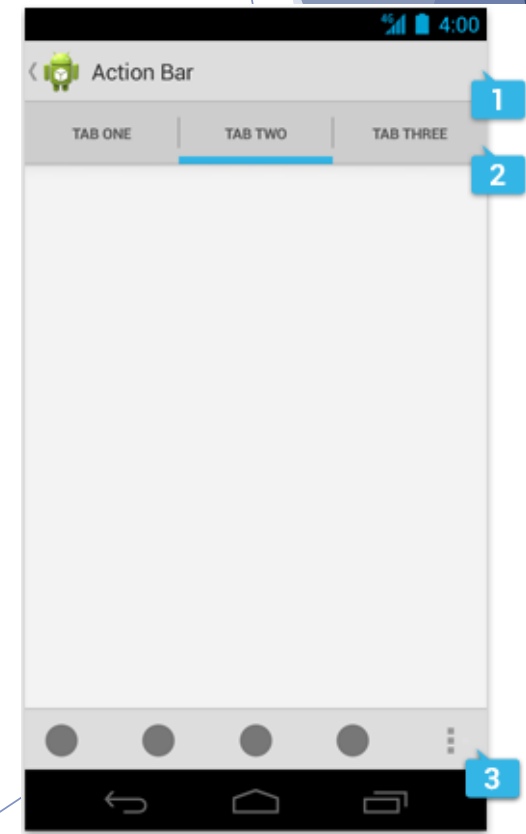


- ▶ 1) 애플리케이션 아이콘
- ▶ 2) 뷰 컨트롤 - 탭, 스피너 등 사용
- ▶ 액션 버튼
- ▶ 액션 오버플로우

ActionBar

: ActionBar 기초 사용법

- ▶ 분할 액션바
 - ▶ 메인 액션바
 - ▶ 상단바
 - ▶ 하단바
- ▶ 분할 액션바 고려사항
 - ▶ 만약 사용자가 현재 화면에서 상위 계층으로 이동할 수 있다면 메인 액션바에는 적어도 상위 이동 기호는 표시해야 한다
 - ▶ 사용자가 빠르게 애플리케이션에서 제공하는 뷰 사이를 전환할 수 있도록 상단 바에 탭이나 스피너를 배치
 - ▶ 필요할 경우 액션과 액션 오버플로우를 표시하려면 하단 바를 배치



ActionBar

: ActionBar 기초 사용법

[실습] ActionBarExample

1. 액션바와 메뉴는 긴밀한 관계가 있다. 메뉴항목은 아이콘과 함께 액션바에 나타나게 할 수 있다. 액션바에 바로 노출되는 항목을 액션 아이템(Action item)이라 한다.
2. XML에서 메뉴 항목이 액션 아이템으로 표시되도록 설정하려면 다음과 같이 **app:showAsAction** 속성을 지정

```
<menu xmlns:android="http://schemas.android.com/apk/res/android"
      xmlns:app="http://schemas.android.com/apk/res-auto"
      xmlns:tools="http://schemas.android.com/tools"
      tools:context=".MainActivity">
    <item android:id="@+id/item1"
          android:icon="@android:drawable/ic_menu_share"
          android:title="Share"
          app:showAsAction="ifRoom|withText" />
</menu>
```

- app:showAsAction 속성으로 지정할 수 있는 항목
 1. ifRoom : 액션 아이템을 표시할 수 있는 공간이 있다면 액션 아이템을 표시
 2. never : 항상 액션 아이템으로 표시하지 않는다 (기본값)
 3. withText : 메뉴 항목의 아이콘과 메뉴 항목의 텍스트를 함께 액션 아이템으로 표시
 4. always : 항상 액션 아이템으로 표시

ActionBar

: ActionBar 기초 사용법

▶ ActionBar 얻어오기

```
ActionBar actionBar = getSupportActionBar();
```

▶ ActionBar의 다양한 구성을 위한 메서드

메서드	설명
<code>setDisplayHomeAsUpEnabled(boolean showHomeAsUp)</code>	아이콘을 Up 이미지로 표시 설정
<code>setDisplayShowCustomEnabled(boolean showCustom)</code>	CustomView 표시 설정
<code>setDisplayShowHomeEnabled(boolean showHome)</code>	홈 아이콘 표시 설정
<code>setDisplayShowTitleEnabled(boolean showTitle)</code>	타이틀 문자열 표시 설정
<code>setDisplayUseLogoEnabled(boolean useLogo)</code>	로고 표시 설정

ActionBar

: ActionBar 기초 사용법

- ▶ ActionBar의 구성요소 내용을 설정하는 메서드

메서드	설명
<code>setLogo(int resId)</code>	리소스로 로고 지정
<code>setLogo(Drawable logo)</code>	이미지 객체로 로고 지정
<code>setTitle(int resId)</code>	리소스로 타이틀 지정
<code>setTitle(CharSequence title)</code>	문자열로 타이틀 지정
<code>setSubtitle(int resId)</code>	리소스로 서브타이틀 지정
<code>setSubtitle(CharSequence title)</code>	문자열로 서브타이틀 지정
<code>setCustomView(int resId)</code>	리소스로 커스텀 뷰 지정
<code>setCustomView(View view)</code>	뷰로 커스텀 뷰 지정

ActionBar

: ActionBar 기초 사용법

[실습] ActionBarExample

3. 액션바를 얻어와서 표시 내용들을 변경해 봅시다

```
actionBar = getSupportActionBar();  
  
actionBar.setDisplayHomeAsUpEnabled(true);  
actionBar.setDisplayHomeAsUpEnabled(true);  
actionBar.setDisplayHomeAsUpEnabled(true);  
actionBar.setDisplayHomeAsUpEnabled(false);  
  
actionBar.setLogo(R.mipmap.ic_launcher_round);
```

4. 각 메서드들을 다르게 적용하면서 표시 내용이 어떻게 바뀌는지 살펴봅니다.

5. Home Up 버튼을 눌렀을 때의 이벤트를 처리해 봅시다.
- Home Up 버튼은 android.R.id.home 입니다.