

다양한 메뉴(Menu) 사용



목차

- ☐ Android의 메뉴 사용
- ☐ 옵션 메뉴의 구성과 실행
- ☐ 컨텍스트 메뉴의 구성과 실행
- ☐ 팝업 메뉴의 구성과 실행
- ☐ 메뉴의 편집

Android의 메뉴 사용

☐ 메뉴 구성: 프로그램의 기능을 **계층적**으로 표현

- ◆ PC 환경 → 프로그램의 **모든 기능**
- ◆ 모바일 환경 → **환경 설정** 등 부가 기능

☐ Android 메뉴의 유형

- ◆ **옵션** 메뉴(Option): **기본** 메뉴, 메뉴 버튼 클릭 시 표시
 - 앱 전체에 영향을 미치는 기능을 배치
- ◆ **컨텍스트** 메뉴(Context): 요소를 **롱탭** 시 화면 중앙에 표시
 - Floating context menu & contextual action mode
- ◆ **팝업** 메뉴: **특정 뷰 항목에 고정**되어 있는 메뉴
 - 상황에 따른 메뉴

↻ 반드시 X

☐ 메뉴 UI 구현 방법

- ◆ **코드**로 직접 구현
- ◆ **XML**로 메뉴 구현 후 Inflation 수행

Option Menu 의 구성 방법

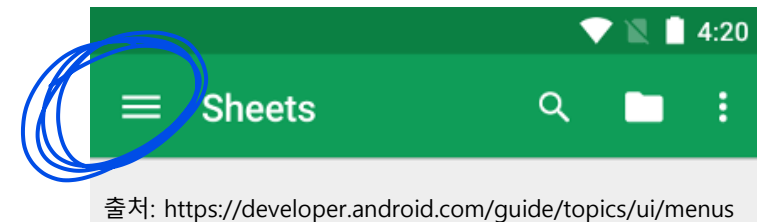
빈 메뉴

 public boolean onCreateOptionsMenu(Menu menu)

- ◆ 메뉴 생성을 위한 Activity_멤버 메소드 Activity가 有
- ◆ 메뉴 생성 필요 시 최초 1회만 호출
 - API level 11 (3.0)부터는 액티비티 시작 시 자동 호출
 - 물리적인 메뉴 버튼 대신 앱바(App Bar)에 메뉴 호출 버튼 생성

메뉴 구성 작성 방법

- ◆ 직접 구현
 - 매개변수 menu 에 메뉴 항목 추가 기능을 메소드 내부에서 직접 구현
- ◆ XML 구현
 - 매개변수 menu 에 메뉴 XML 파일을 inflation 하여 추가
 - 주로 XML 로 메뉴 작성 후 inflation 수행



XML을 사용한 Option Menu 작성 1

메뉴 xml 파일

- ◆ res/menu 폴더 추가 후 menu 관련 xml 파일 추가
- ◆ 예: res/menu/menu_main.xml

res/values/themes/themes.xml 코드 중 .NoActionBar 삭제

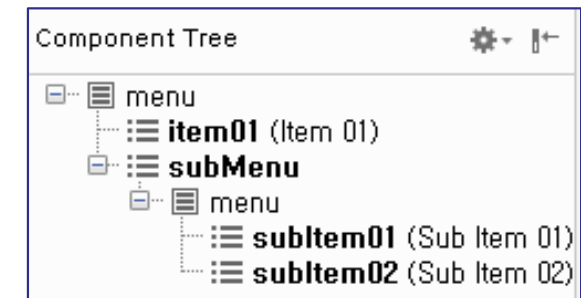
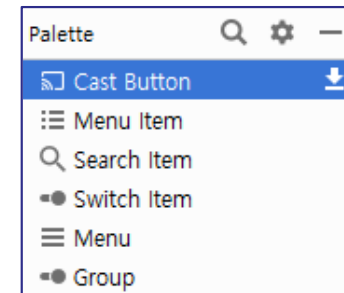
XML 메뉴 기본 구성

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <menu xmlns:tools="http://schemas.android.com/tools"
3        xmlns:app="http://schemas.android.com/apk/res-auto"
4        xmlns:android="http://schemas.android.com/apk/res/android">
5      <item
6        android:id="@+id/item01"
7        android:title="Item 01" />
8      <item
9        android:title="SubMenu" >
10         <menu >
11           <item
12             android:id="@+id/subItem01"
13             android:title="Sub Item 01" />
14           <item
15             android:id="@+id/subItem02"
16             android:title="Sub Item 02" />
17         </menu>
18      </item>
19 </menu>

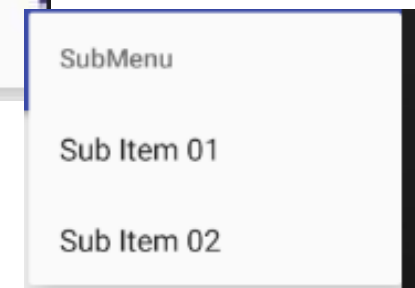
```

<item> 태그 내부에
<menu> 태그를 추가할
경우 하위 메뉴 생성



• 메뉴 클릭

• 하위 메뉴 클릭

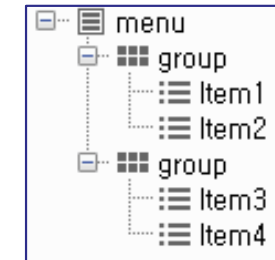


XML을 사용한 Option Menu 작성 2

XML 그룹 메뉴 구성

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <menu xmlns:tools="http://schemas.android.com/tools"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:android="http://schemas.android.com/apk/res/android">
5     <group android:checkableBehavior="all">
6         <item android:title="Item1" />
7         <item android:title="Item2" />
8     </group>
9     <group android:checkableBehavior="single">
10        <item android:title="Item3" android:checked="true" />
11        <item android:title="Item4" />
12    </group>
13 </menu>
    
```



• checkableBehavior
: none | single | all

체크
버튼

라디오버튼

all

single

checked

<item> 의 showAsAction 속성

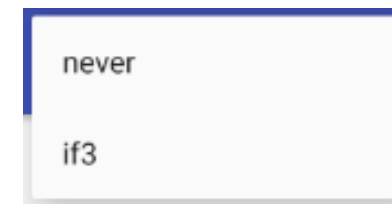
- ◆ App Bar 에 메뉴 항목을 언제/어떻게 표시할 것인지 지정
- ◆ ifRoom | withText | never | always | collapseActionView

```

<item
    android:title="always"
    app:showAsAction="always" />
<item
    android:title="never"
    app:showAsAction="never" />
<item
    android:title="if3"
    app:showAsAction="ifRoom" />
<item
    android:title="if4"
    app:showAsAction="ifRoom" />
    
```

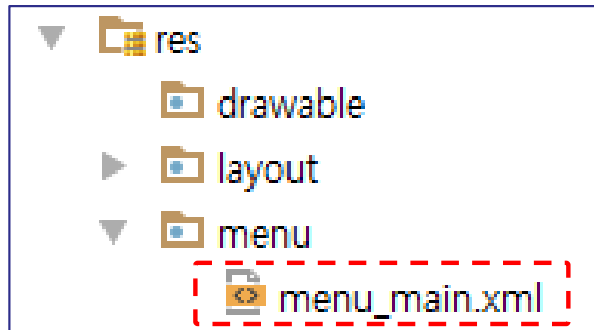


• App Bar 부분



XML을 사용한 Option Menu 작성 3

XML로 메뉴 항목 구성 후 Inflation 수행



```

1  <menu xmlns:android="http://schemas.android.com/apk/res/android"
2      xmlns:app="http://schemas.android.com/apk/res-auto"
3      xmlns:tools="http://schemas.android.com/tools"
4      tools:context="mobile.test.mymenuapplication.MainActivity">
5      <item
6          android:id="@+id/action_settings"
7          android:orderInCategory="100"
8          android:title="Settings"
9          app:showAsAction="never" />
10 </menu>
    
```

```

14  override fun onCreateOptionsMenu(menu: Menu?): Boolean {
15      menuInflater.inflate(R.menu.menu_main, menu)
16      return super.onCreateOptionsMenu(menu)
17  }
    
```

프로젝트 생성 시 Blank Activity
로 생성 시 관련 메소드를 자동
으로 생성해줌

◆menuInflater

- Menu 관련 XML Inflation을 수행하는 객체
- menuInflater.inflate(menu_xml_file, menu) 을 사용하여 menu_xml_file 의 xml 리소스에서 메뉴 객체 생성 후 시스템에서 전달한 menu 객체에 지정

Option Menu 의 사용

메뉴 실행 중 메뉴 구성의 변경이 필요할 경우

- ◆ 옵션 메뉴 변경 메소드로 메뉴 변경

```
override fun onPrepareOptionsMenu(menu: Menu?): Boolean
```

- ◆ Activity 클래스의 멤버 메소드
- ◆ 메뉴가 보여질 때마다 호출
- ◆ 실행 중 변경할 메뉴 구성을 구현

Option Menu 항목 선택 처리

- ◆ 메뉴를 선택했을 때 호출하는 메소드

```
override fun onOptionsItemSelected(item: MenuItem): Boolean
```

- ◆ Activity 클래스의 멤버 메소드
- ◆ item.getItemId 로 메뉴 항목 구분 → 항목 추가 시 ID
- ◆ 메뉴 항목의 이벤트 처리를 완료하였을 경우 true 반환

Option Menu 항목 선택 처리 (XML에 구현)

XML 메뉴 Layout 에 메뉴 선택 시의 실행 함수 지정

◆ onClick 속성 → onOptionsItemSelected 추가

```
<item
    android:id="@+id/action_settings"
    android:onClick="onMenuItemClick"
    android:title="Settings" />
```

Activity의 멤버함수로 해당 메소드 선언

```
32 fun onMenuItemClick (item: MenuItem) {
33     when (item.itemId) {
34         R.id.action_settings ->
35             Toast.makeText(this, "Menu Click!", Toast.LENGTH_SHORT).show()
36     }
37 }
```

◆ onOptionsItemSelected() 메소드 보다 높은 우선순위

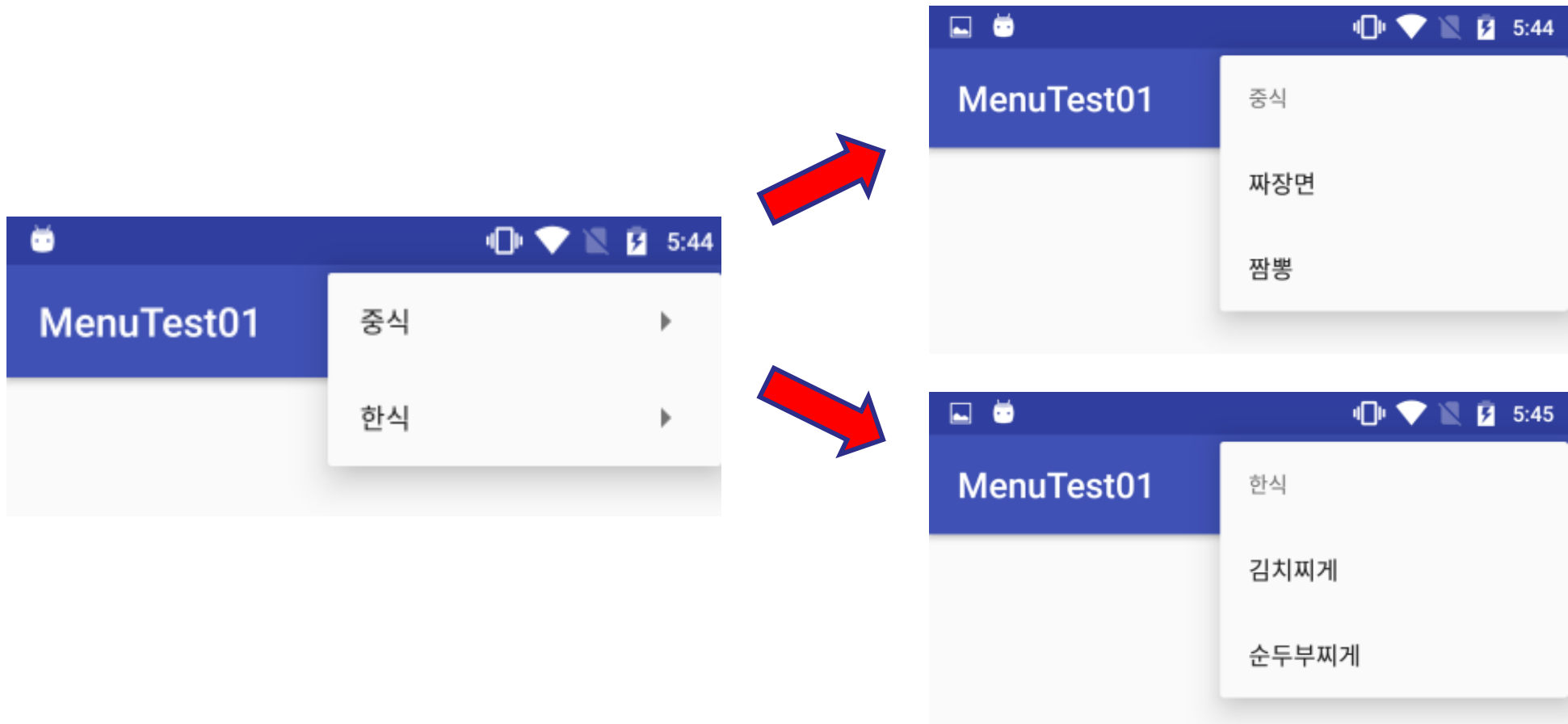
◆ 호환성 고려가 필요

◆ ActionBar 적용 시 개별 항목 사용이 많으므로 적용이 유리

실 습

☞ 다음과 같은 Option Menu를 구성하시오.

- ◆ Code 직접 구현
- ◆ Menu XML 사용 구현
- ◆ Menu Item 클릭 시 해당 음식 평가를 Toast 로 출력



Context Menu

 특정 뷰를 길게 클릭할 때 관련 메뉴 출력

- ◆ Floating Context Menu
- ◆ Contextual Action Mode

 Context Menu 구현 및 사용 절차

1. Context Menu를 사용 View를 Activity에 등록 → `registerForContextMenu()`
2. `onCreateContextMenu()` 구현 : XML로 정의한 메뉴를 inflation
3. `onContextItemSelected()` 구현: 메뉴 기능 실행

1. Context Menu 사용 View를 Activity에 등록

• `Activity.registerForContextMenu (view: View) : Unit`

- ◆ 특정 View 를 길게 클릭하면 Activity 에서 Context Menu 를 등록한 View인지 확인 후 `onCreateContextMenu()` 실행
- ◆ 일반적으로 Activity의 `onCreate()` 에서 등록 수행

컨텍스트 메뉴의 구현 및 실행

여러 뷰가 공유할 수 있으므로 뷰 식별을 위해 현재 선택한 뷰 객체 전달

onCreateContextMenu() 구현

```
// Activity 안에서 여러 뷰가 공유하는 컨텍스트 메뉴 생성 시
override fun onCreateContextMenu(menu: ContextMenu?, v: View?,
                                menuInfo: ContextMenu.ContextMenuInfo? )

// 커스텀 뷰 사용 시 뷰 클래스 내부에 자신이 사용할 컨텍스트 메뉴 생성 시
• override fun onCreateContextMenu (menu ContextMenu)
```

- ◆ 메뉴를 실행할 때마다 호출됨
- ◆ v: 여러 View 에서 같은 Context Menu를 사용하는 경우 View의 아이디로 메뉴를 사용하는 View 구분
- ◆ menuInfo: 메뉴에 대한 세부 정보
 - 예: List View를 롱클릭했을 경우 → 항목의 순서

onContextItemSelected() 구현

```
override fun onContextItemSelected(item: MenuItem): Boolean
```

- ◆ item의 getItemId()를 사용하여 id 확인 후 구분 (R.id.*)
- ◆ 메뉴 항목의 이벤트 처리를 완료하였을 경우 true 반환 → 처리하지 않았을 경우 Super class 에 전달

팝업 메뉴의 구현

☐ 컨텍스트 메뉴와의 비교

- ◆ 화면 상에 메뉴 표시 위치가 해당 뷰에 연결되어 표시 → 컨텍스트 메뉴도!
- ◆ 롱탭 이외의 경우에도 호출 가능

```

fun showPopup(view: View) {
    val popup = PopupMenu(this, view)
    val inflater: MenuInflater = popup.menuInflater
    inflater.inflate(R.menu.popup_menu, popup.menu)
    popup.show()

    popup.setOnMenuItemClickListener { it: MenuItem?
        when(it.itemId) {
            R.id.menu01 ->
                Toast.makeText(this, "Popup!!!", Toast.LENGTH_SHORT).show()
        }
        true ^setOnMenuItemClickListener
    }
}
    
```

Popup 을 띄울 View 객체

popup.getMenuInflater() 대체

popup.getMenu() 대체

XML

Popup 메뉴 항목 클릭 리스너 연결

대체 변수 하나만 생각 가능

Popup 을 표시할 view 객체

```

view.setOnClickListener { it: View!
    showPopup(view)
}
    
```

메뉴의 편집

☞ 실행 중 Option Menu 항목 변경이 필요한 경우

- ◆ onPrepareOptionsMenu() 에 변경 내용 추가
- ◆ removeItem(), clear(), findItem(), getItem() 등

☞ 메뉴 체크 항목 사용

- ◆ <group> 태그 checkableBehavior 속성 지정

■ none | all | single

default

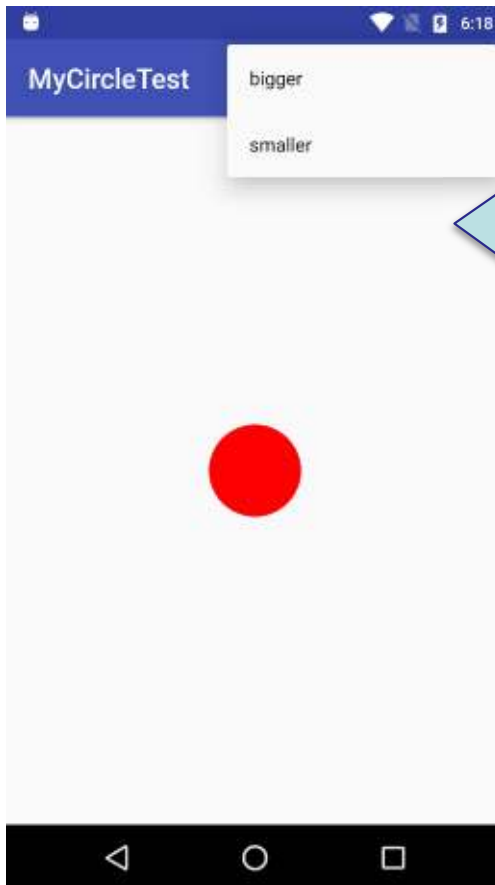
- ◆ fun isChecked() : boolean 를 사용하여 체크 여부 확인
- ◆ fun setChecked(checked: boolean) : MenuItem 을 사용하여 체크 상태 변경

☞ 메뉴의 교체

- ◆ 기존 메뉴를 삭제하고 새로운 메뉴로 대체할 때
- ◆ Activity.invalidateOptionsMenu() 사용 → onPrepareOptionsMenu 자동 호출

Exam02 프로젝트에 다음 메뉴를 추가하시오.

- ◆ 원의 크기를 변경하는 Option Menu
- ◆ 원의 색상을 변경하는 Context Menu → ⊕ 라디오 버튼



Hint

- MyCircle 클래스에 원의 크기를 결정하는 부분을 찾아 해당 부분을 멤버변수로 만든 후 setter 추가
- 메뉴 항목 선택 시 추가한 setter 값을 사용하여 원의 크기 변경 후 화면 다시 그리기 수행



Hint

- MyCircle 클래스에 원의 색상부분을 결정하는 부분을 찾아 해당 부분을 멤버변수로 만든 후 setter 추가
- 메뉴 항목 선택 시 추가한 setter 값을 사용하여 원의 색상 변경 후 화면 다시 그리기 수행