



# 메뉴와 대화상자

≡ 태그

## 1. 옵션 메뉴

- 키패드 메뉴 버튼 또는 화면 오른쪽 위의 메뉴 아이콘을 눌렀을 때 화면 하단에 나오는 메뉴

## 2. 컨텍스트 메뉴

- 위젯 등을 롱클릭하면 나오는 메뉴
  - 제목과 함께 화면의 중앙에 출력

## 3. XML을 이용한 옵션 메뉴

1. 메뉴 코딩
2. 메뉴 파일 등록
  - onCreateOptionsMenu() 메소드 오버라이딩
3. 메뉴 선택시 동작할 내용 코딩
  - onOptionsItemSelected() 메소드 오버라이딩

### ▼ 실습 7-1) 배경색 변경 앱

▼ Kotlin 코드 - menuInflater, onCreateOptionsMenu, onOptionsItemSelected, when문

```
class MainActivity : AppCompatActivity() {  
  
    override fun onCreate(savedInstanceState: Bundle?) {  
        super.onCreate(savedInstanceState)  
        setContentView(R.layout.activity_main)  
    }  
}
```

```

    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        super.onCreateOptionsMenu(menu)
        var mInflater = menuInflater
        mInflater.inflate(R.menu.menu1, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        super.onOptionsItemSelected(item)
        when (item.itemId) {
            R.id.itemRed-> {
                baseLayout.setBackgroundColor(Color.RED)
                return true
            }
            R.id.subSiz-> {
                button1.scaleX= 2f
                return true
            }
        }
        return false
    }
}

```

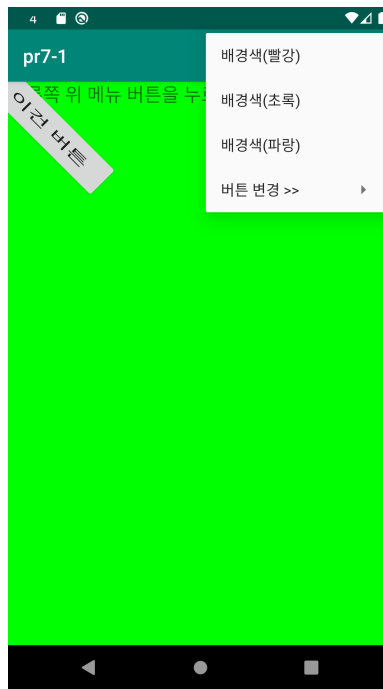
## ▼ Menu resource file

```

<menu>
    <item
        android:id="@+id/itemRed"
        android:title="배경색(빨강)">
    </item>
    <item android:title="버튼 변경 >> ">
        <menu>
            <item
                android:id="@+id/subRotate"
                android:title="버튼 45도 회전"/>
            </item>
        </menu>
    </item>
</menu>

```

## ▼ 결과



## ▼ 직접 풀어보기 7-1) RelativeLayout

### ▼ XML 코드

```
<RelativeLayout>

    <TextView
        android:id="@+id/View1"
        android:text="각도 입력" />

    <EditText
        android:id="@+id/Edit1"
        android:layout_toRightOf="@+id/View1"
        android:layout_alignBaseline="@+id/View1" />

    <ImageView
        android:id="@+id/imgView"
        android:visibility="invisible"
        android:scaleType="fitXY"
        android:layout_centerInParent="true" />
</RelativeLayout>
```

### ▼ Kotlin 코드 - setChecked, setImageResource

```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
```

```

        setContentView(R.layout.activity_main)
    }

    override fun onCreateOptionsMenu(menu: Menu?): Boolean {
        super.onCreateOptionsMenu(menu)
        var mInflater = menuInflater
        mInflater.inflate(R.menu.menu1, menu)
        return true
    }

    override fun onOptionsItemSelected(item: MenuItem): Boolean {
        super.onOptionsItemSelected(item)
        when (item.itemId) {
            R.id.rotate-> {
                var num : String = Edit1.text.toString()
                imgView.rotation= num.toFloat()
                return true
            }
            R.id.item1-> {
                item.setChecked(true)
                imgView.setImageResource(R.drawable.item1)
                imgView.visibility= View.VISIBLE
                return true
            }
        }
        return false
    }
}

```

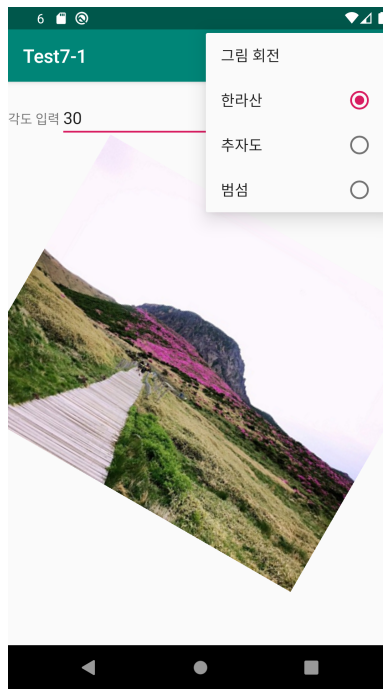
#### ▼ Menu- <group android:checkableBehavior="single">

```

<menu>
    <item
        android:id="@+id/rotate"
        android:title="그림 회전 " />
    <group
        android:checkableBehavior="single">
        <item
            android:id="@+id/item1"
            android:title="한라산" />
        </group>
</menu>

```

#### ▼ 결과



## 4. XML을 이용한 컨텍스트 메뉴

1. 메뉴 코딩
2. 메뉴를 사용할 위젯 등록
  - registerForContextMenu() 등록
3. 메뉴 파일 등록
  - onCreateContextMenu() 메소드 오버라이딩
4. 메뉴 선택 시 동작할 내용 코딩
  - onOptionsItemSelected() 메소드 오버라이딩

### ▼ 실습 7-2)

#### ▼ Kotlin 코드 - registerForContextMenu

- onCreateContextMenu : View v 가 롱클릭한 위젯!!

```
class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
```

```

        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        registerForContextMenu(btn1)
        registerForContextMenu(btn2)
    }

    override fun onCreateContextMenu(
        menu: ContextMenu?,
        v: View?,
        menuInfo: ContextMenu.ContextMenuInfo?
    ) {
        super.onCreateContextMenu(menu, v, menuInfo)

        var mInflater = this.menuInflater
        if (v == btn1) {
            mInflater.inflate(R.menu.menu1, menu)
        }
        if (v == btn2) {
            mInflater.inflate(R.menu.menu2, menu)
        }
    }
}

```

## 5. Toast.makeText(Context context, String message, int duration).show()

- context
  - 현재 액티비티를 표시하기 위해 this를 주로 사용
  - 내부 클래스에서 토스트를 출력하기 위해 applicationContext를 사용
- duration
  - 화면에 나타는 시간
  - Toast.LENGTH\_LONG, Toast.LENGTH\_SHORT
- show()
  - 생성된 토스트를 화면에 보여주기 위함

## 6. setGravity() 메소드

- Toast.setGravity(int gravity, int xOffset, int yOffset)
- ex) setGravity(Gravity.TOP or Gravity.LEFT, xOffset, yOffset)

## 7. 대화상자 설정 순서

### 1. 대화상자 생성

- AlertDialog.Builder 클래스로 생성

### 2. 용도에 따른 설정

setTitle()	제목 설정	setMessage()	내용 입력
setIcon()	아이콘 설정	setPositiveButton()	<OK> 버튼
setNegativeButton()	<Cancel> 버튼	setItems()	목록 출력
setSingleChoiceItems	라디오버튼 목록 출력	setMultiChoiceItems	체크박스 목록 출력

### 3. 대화상자 화면 출력

- show()

#### ▼ 예제 7-18)

##### ▼ Kotlin 코드 - setItems

: 대화상자에 리스트 형태의 목록을 출력하고 그중 하나를 선택 가능

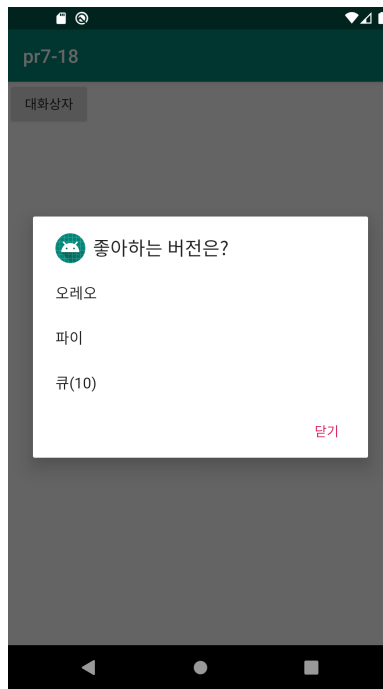
```
class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        button1.setOnClickListener{
            var versionArray =arrayOf("오레오", "파이", "큐(10)")
            var dlg = AlertDialog.Builder(this)
            dlg.setIcon(R.mipmap_launcher)
            dlg.setTitle("좋아하는 버전은?")
            dlg.setPositiveButton("닫기", null)

            dlg.setItems(versionArray){dialog, which->
                button1.text= versionArray[which]
            }
            dlg.show()
        }
    }
}
```

##### ▼ 결과



## 8. setSingleChoiceItems(문자열 배열, 초기 선택 인덱스) vs setItems

- 항목을 선택해도 대화상자가 닫히지 않음
- 라디오버튼과 같은 형태로 출력

## 9. setMultiChoiceItems()

- 여러 개를 동시에 선택할때 사용
- 체크박스 형태로 표시

```
var checkArray = booleanArrayOf(true, false, false)
dlg.setMultiChoiceItems(versionArray, checkArray) { dialog, which, isChecked ->
    button1.text = versionArray[which]
}
```

### ▼ 실습 7-3) 사용자 정보 입력 앱

#### ▼ Kotlin 코드 - inflate, setPositiveButton, setNegativeButton 이벤트 처리



```

class MainActivity : AppCompatActivity() {

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

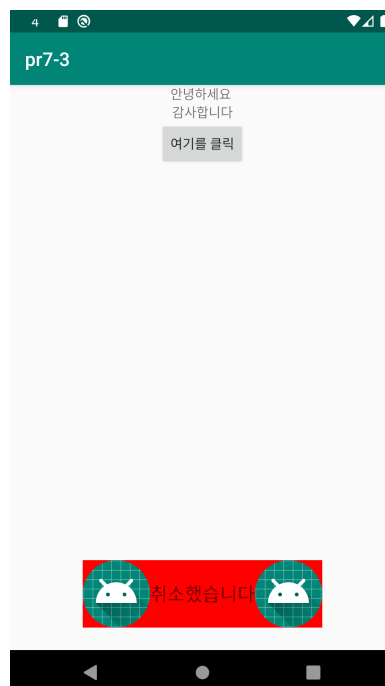
        btn1.setOnClickListener {
            var dialogView = View.inflate(this, R.layout.dialog1, null)
            var dlg = AlertDialog.Builder(this)
            dlg.setTitle("사용자 정보 입력")
            dlg.setIcon(R.mipmap.ic_launcher_round)
            dlg.setView(dialogView)
            dlg.setPositiveButton("확인") { dialog, which ->
                Toast.makeText(applicationContext, "완료했습니다", Toast.LENGTH_
LONG).show()

                text1.text = dialogView.dlgEdt1.text.toString()
                text2.text = dialogView.dlgEdt2.text.toString()
            }

            dlg.setNegativeButton("취소") { dialog, which ->
                var toast = Toast(this)
                var toastView = View.inflate(this, R.layout.toast1, null)
                toastView.toastText1.text = "취소했습니다"
                toast.view = toastView
                toast.show()
            }
            dlg.show()
        }
    }
}

```

## ▼ 결과



## 10. 다른 레이아웃 이용방법 - View.inflate

View.inflate(this, R.layout.dialog1, null)

## 11. 토스트 메시지와 레이아웃 이용

```
var toast = Toast(this)
var toastView = View.inflate(this, R.layout.toast1, null)
toastView.toastText1.text= "취소했습니다"
toast.view= toastView
toast.show()
```