

대화상자 (Dialog)



목차

 대화상자의 사용

 대화상자의 이벤트 처리

 목록 표시 대화상자

- ◆ 단순 목록
- ◆ 단일 선택 목록
- ◆ 다중 선택 목록

 커스텀 대화상자의 작성

대화상자

☐ 개요

- ◆사용자에게 정보를 전달하거나 선택을 입력 받는 용도
- ◆Toast와 달리 정보 전달의 경우에도 확인 필요 → 확인버튼

☐ Dialog 클래스

- ◆안드로이드의 기본 대화상자 클래스
- ◆대화상자의 모든 기능을 제공하나 세세한 제어 필요 → 사용 지양

☐ AlertDialog 클래스

- ◆Dialog 하위 클래스로 간단한 설정을 통해 대화상자 생성
- ◆내부 클래스인 AlertDialog.Builder 클래스를 활용하여 대화상자 객체 생성

상속

☐ DatePickerDialog 또는 TimePickerDialog 클래스

☐ DialogFragment 클래스 Activity같은 화면 제어 역할

- ◆다이얼로그 관리를 위해 사용하는 Fragment

AlertDialog의 사용

다이어로그 생성 및 표시

• 객체.apply { ... } 범위지정함수
- 객체를 생성하며 필요setter 작업 수행 시 활용

```
mainBinding.button.setOnClickListener { it: View!
    val builder : AlertDialog.Builder = AlertDialog.Builder(this).apply {
        title = "대화상자 제목"
        setMessage("대화상자 메시지")
        setIcon(R.mipmap.ic_launcher_round)
        setPositiveButton("확인", null)
        setNeutralButton("대기", null)
        setNegativeButton("취소", null)
        setCancelable(false)
    }
    // val dialog: Dialog = builder.create()
    // dialog.show()
    builder.show()
}
```

• builder.title ...
• builder.setMessage(...)
• ...

• 클릭 매개변수에 null을 전달할 경우 대화상자 종료

• 대화상자 외부 또는 BACK 버튼을 클릭해도 대화상자 유지

• Dialog 객체 생성 후 표시

• Dialog 표시

AlertDialog 버튼 – 이벤트 처리

대화상자 버튼 클릭 시 이벤트 리스너

```
val onOkClick = DialogInterface.OnClickListener {
    dialogInterface: DialogInterface?, whichButton: Int ->
        Toast.makeText(this@MainActivity, "확인!", Toast.LENGTH_SHORT).show()
}

mainBinding.button.setOnClickListener { it: View!
    val builder : AlertDialog.Builder = AlertDialog.Builder(this).apply { this: AlertDialog.Builder
        title = "대화상자 제목"
        setMessage("대화상자 메시지")
        setIcon(R.mipmap.ic_launcher_round)
        setPositiveButton("확인", onOkClick)
    }
    builder.show()
}
```

• DialogInterface.OnClickListener
- 대화상자의 버튼 클릭 처리

• 람다함수 적용
• 매개변수의 마지막이 람다함수일 경우
() 안에서 () {} 형식으로 변경

```
setPositiveButton("확인") { dialogInterface: DialogInterface?, i: Int ->
    Toast.makeText(this@MainActivity, "확인!", Toast.LENGTH_SHORT).show()
}
```

대화상자 처리 시 유의점

Modal 대화상자

- ◆ 대화상자를 표시하면 대화상자가 닫히기 까지 프로그램은 실행 대기 → 대화상자를 닫은 이후에 동작
- ◆ 파일 열기 등의 일반적인 대화상자 방식

Modeless 대화상자

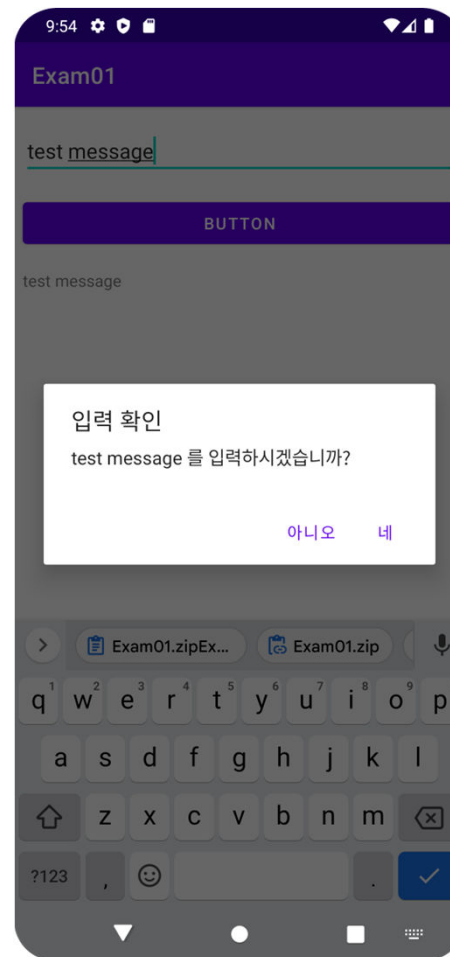
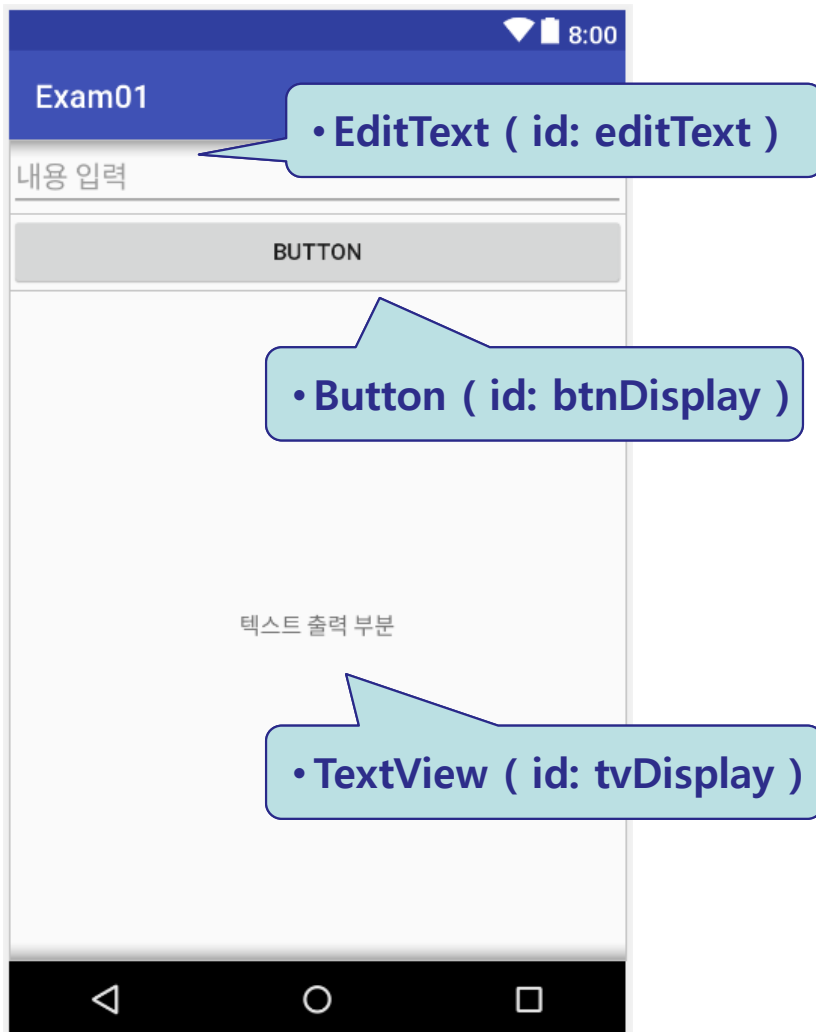
- ◆ 대화상자를 표시하여도 프로그램은 계속 실행 → 대화상자를 닫지 않아도 동작 가능
- ◆ 찾기 대화상자와 같은 대화상자 방식

 Android의 대화상자는 실행 흐름을 막지 않으므로 사용 시 실행 흐름을 잘 고려하여야 함

실습 01 (Exam01.zip)

다음과 같은 화면을 구성한 후 버튼을 눌렀을 때 다이얼로그를 출력하여 동작을 수행하도록 구현하시오.

◆ EditText에 문자열 입력 후 버튼 클릭 시 다이얼로그 출력



1. EditText에 문자열 입력 후 Button을 누르면 다이얼로그 표시
2. 다이얼로그에는 입력한 메시지 출력
3. [네]를 누를 경우 TextView 에 입력 메시지 출력 [아니오]를 누를 경우 다이얼로그 닫힘

목록을 갖고 있는 대화상자

목록을 대화상자에 표시하기

- AlertDialogBuilder.setItems (items: CharSequence[], listener: DialogInterface.OnClickListener)
- AlertDialogBuilder.setItems (itemArrayId: Int, listener: DialogInterface.OnClickListener)

```
val builder : AlertDialog.Builder = AlertDialog.Builder(this).apply { this: AlertDialog.Builder
    setItems(R.array.foods) {
        DialogInterface: DialogInterface?, idx: Int ->
        val foods = resources.getStringArray(R.array.foods)
        Toast.makeText(this@MainActivity, "선택:" + foods[idx], Toast.LENGTH_SHORT).show()
    }
}
```

• setMessage() 대신 setItems 사용

• XML resource의 배열 가져오기

```
<resources>
    <string-array name="foods">
        <item>김치찌개</item>
        <item>마라탕</item>
        <item>초밥</item>
    </string-array>
</resources>
```

- res/values/arrays.xml

김치찌개
마라탕
초밥

확인

단일선택 목록을 갖고 있는 대화상자

대화상자에 라디오 버튼을 갖고 있는 목록 출력

- ◆ 목록의 항목 선택 시 선택 인덱스를 보관한 후 필요한 위치에서 활용

```
var selectedIdx = 0
```

- 선택한 항목의 index 를 저장하기 위한 변수(멤버변수로 선언)
- 현재 선택한 항목이 아닌 이전 선택 항목 index 가 저장됨

```
val builder : AlertDialog.Builder = AlertDialog.Builder(this).apply { this: AlertDialog.Builder
    setSingleChoiceItems(R.array.foods, selectedIdx) {
        dialogInterface: DialogInterface?, idx: Int ->
            selectedIdx = idx
    }
    setPositiveButton("확인") {
        dialogInterface: DialogInterface?, idx: Int ->
            val foods = resources.getStringArray(R.array.foods)
            Toast.makeText(this@MainActivity, "선택: " + foods[selectedIdx], Toast.LENGTH_SHORT)
                .show()
    }
}
```

- 선택 항목의 Index 를 멤버변수에 저장

- 현재 선택한 항목의 index

다중선택 목록을 갖고 있는 대화상자

대화상자에 체크박스를 갖고 있는 목록 출력

```
var selectedItems = booleanArrayOf(false, false, false)
```

• 항목 선택여부를 저장하는
배열 멤버변수 선언

```
val builder : AlertDialog.Builder = AlertDialog.Builder(this).apply { this: AlertDialog.Builder
    setMultiChoiceItems(R.array.foods, selectedItems) {
        dialogInterface: DialogInterface?, idx: Int, isChecked: Boolean ->
            selectedItems[idx] = isChecked
    }
    setPositiveButton("확인") {
        dialogInterface: DialogInterface?, idx: Int ->
            val foods = resources.getStringArray(R.array.foods)
            var result = "선택: "
            for (i in 0 ≤ until < selectedItems.size) {
                if (selectedItems[i]) result += foods[i] + " "
            }
            Toast.makeText(this@MainActivity, result, Toast.LENGTH_SHORT)
                .show()
    }
}
```

• 현재 항목의 선택 여부를
true/false 로 저장

• 배열 index 위치의
체크 여부 판단

커스텀 대화상자

대화상자에 보여줄 Layout XML 작성

제품
제품명: etProduct
수량
수량: etQuantity
<input type="checkbox"/> 착불결제 (cbPayment)

• res/layout/dialog_layout.xml

레이아웃을 Inflation 한 후 대화상자에 설정

```
mainBinding.button.setOnClickListener { it: View!
    val dialogLayout = DialogLayoutBinding.inflate(layoutInflater)

    val builder : AlertDialog.Builder = AlertDialog.Builder(this).apply { this: AlertDialog.Builder
        setView(dialogLayout.root)
        setPositiveButton("확인") {
            dialogInterface: DialogInterface?, idx: Int ->
            val product = dialogLayout.etProduct.text.toString()
            Toast.makeText(this@MainActivity, product, Toast.LENGTH_SHORT)
                .show()
        }
    }
```

• 대화상자에 사용할 layout을 inflation

• layout을 대화 상자에 설정

• 다이얼로그의 layout에 배치한 뷰를 접근

DialogFragment 사용

개요

- ◆ 대화상자 컨테이너
 - `Androidx.fragment.app.DialogFragment`
- ◆ 다이얼로그의 생성/수정/삭제 등 수명주기와 관련한 이벤트를 관리
- ◆ 대화상자의 크기를 바꾸어 표시하는 등 유연성 제공

사용 방법

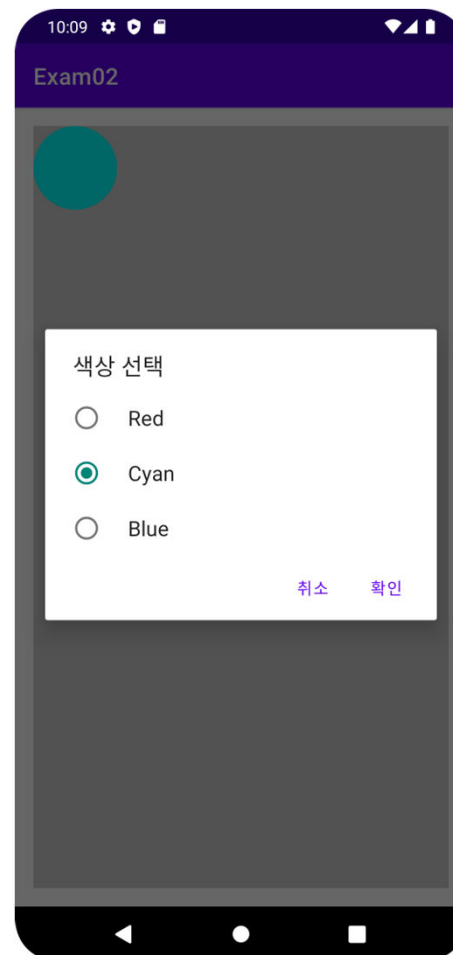
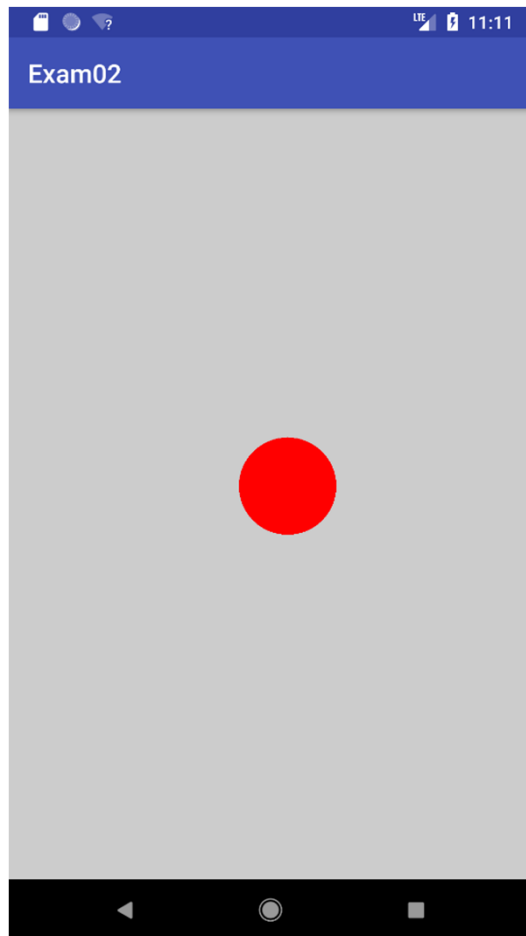
- ◆ DialogFragment 상속 클래스 구현 후 `onCreateDialog()` 를 재정의 → 다이얼로그 생성 코드 작성
- ◆ 다이얼로그를 표시하고자 할 때 DialogFragment 객체 생성 후 `show()` 호출

참고 링크

- ◆ <https://developer.android.com/guide/topics/ui/dialogs>

실습 02 (Exam02.zip)

📱 커스텀 뷰인 MyCustomView 를 롱클릭하면 색상 변경 다이얼로그를 출력하고 선택한 색상으로 원의 색상을 변경하도록 구현하시오.



1. myCustomView를 롱클릭하면 다이얼로그 표시
2. SingleChoice 에서 색상 선택 후 확인을 누를 경우 원의 색상 변경
 - 현재 선택한 색상을 기록하는 멤버변수를 선언하여 사용
 - 확인을 누를 때 위의 멤버변수를 확인하여 선택한 색상으로 원의 색상 변경 수행