



<8장> 대화상자 만들기

학습 목표

- 안드로이드의 대화상자 유형을 이해한다.
- 기본 대화상자를 만드는 과정에 대해 익힌다.
- 날짜/시간 선택 대화상자를 만드는 과정에 대해 익힌다.
- 커스텀 대화상자를 만드는 과정에 대해 익힌다.

목차

01 대화상자 살펴보기

02 기본 대화상자 만들기

03 날짜/시간 선택 대화상자 만들기

04 사용자 정의 대화상자 만들기

[도서 쇼핑몰] 로그인과 장바구니 등록 대화상자 만들기

01

대화상자 살펴보기

1. 대화상자 살펴보기

■ 대화상자

- 전체 화면을 채우지 않고 일부 화면만 가리는 창
- 주로 사용자에게 결정을 내리거나 추가 정보를 입력하도록 유도하는 메시지 표시
- 일반적으로 사용자가 응답하기 전까지는 화면에서 사라지지 않고 대기 상태로 유지
- 앱을 계속 진행하기 위한 특정 작업을 수행하는 메시지를 사용자에게 표시하거나 화면을 전환할 때, 화면 전체를 변경하지 않고 일부분만 변경하여 보여줄 때 사용

1. 대화상자 살펴보기

■ 대화상자

그림 8-1 대화상자의 유형



표 8-1 대화상자의 클래스 유형

유형	설명
AlertDialog	제목과 최대 3개의 버튼, 선택 가능한 항목의 목록 또는 사용자 정의 레이아웃을 표시할 수 있는 대화 상자입니다.
DatePickerDialog	사용자가 날짜를 선택할 수 있도록 미리 정의된 UI가 있는 대화상자입니다.
TimePickerDialog	사용자가 시간을 선택할 수 있도록 미리 정의된 UI가 있는 대화상자입니다.

02

기본 대화상자 만들기

2. 기본 대화상자 만들기

■ 기본 대화상자

- 일반적으로 기본 대화상자는 작은 창을 표시하는 경고 대화상자로, 사용자에게 어떤 액션을 취할지(예 : YES 또는 NO 옵션 버튼) 결정하도록 유도
- 주로 앱에서 뒤로 가기 버튼과 팝업 닫기 등을 확인하기 위한 알림 대화상자로도 활용
- 기본 대화상자는 보통 3개의 영역으로 구성되며, AlertDialog 클래스를 사용하여 다양한 디자인으로 구성 가능

2. 기본 대화상자 만들기

■ 기본 대화상자

그림 8-2 기본 대화상자의 레이아웃



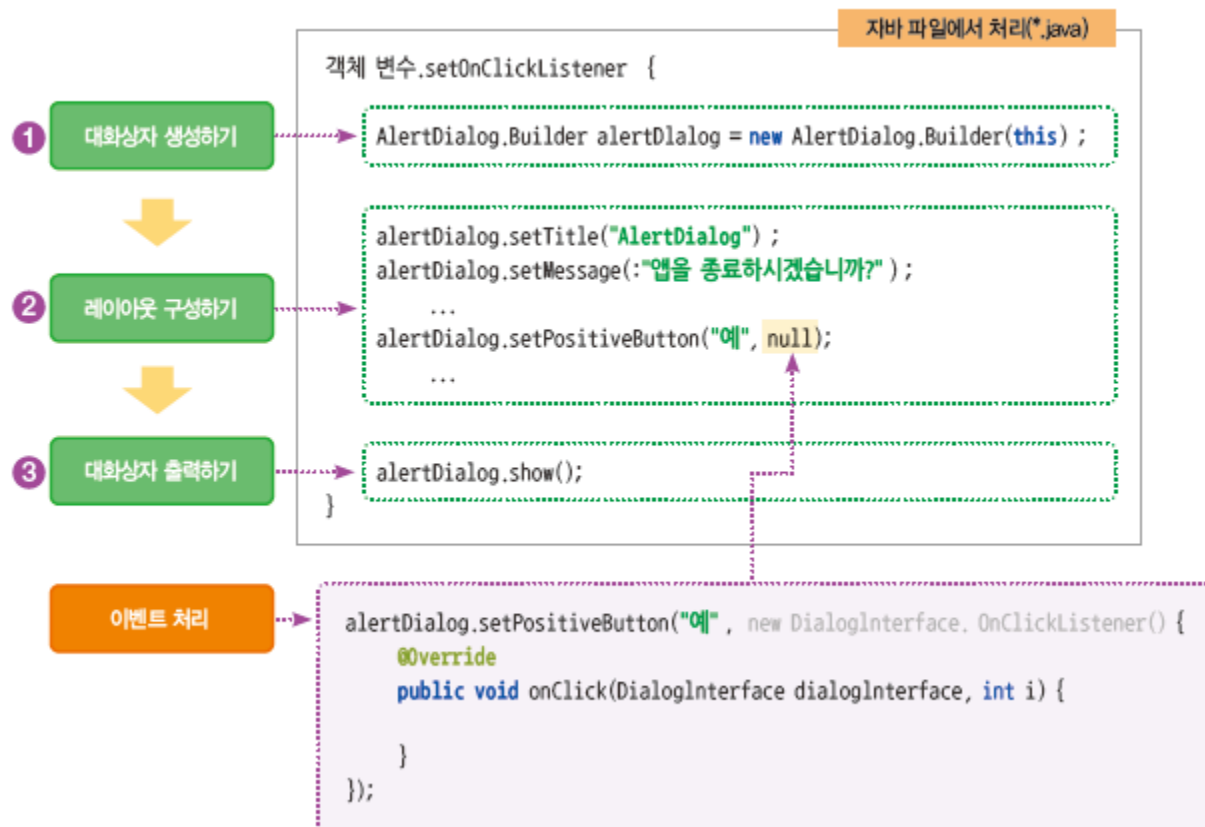
표 8-2 기본 대화상자의 구성 요소

영역	설명
① 제목 영역	선택 사항으로, 요구 사항에 따라 제목을 표시하거나 아이콘을 설정합니다.
② 내용 영역	요구 사항에 따라 메시지나 라디오 버튼, 체크박스 등과 같은 선택 목록 또는 기타 사용자 정의 레이아웃을 표시합니다.
③ 버튼 영역	사용자와 상호작용하기 위한 버튼을 표시합니다. 예, 아니오, 취소 등의 작업 버튼을 최대 3개까지 사용할 수 있습니다.

2. 기본 대화상자 만들기

■ 기본 대화상자

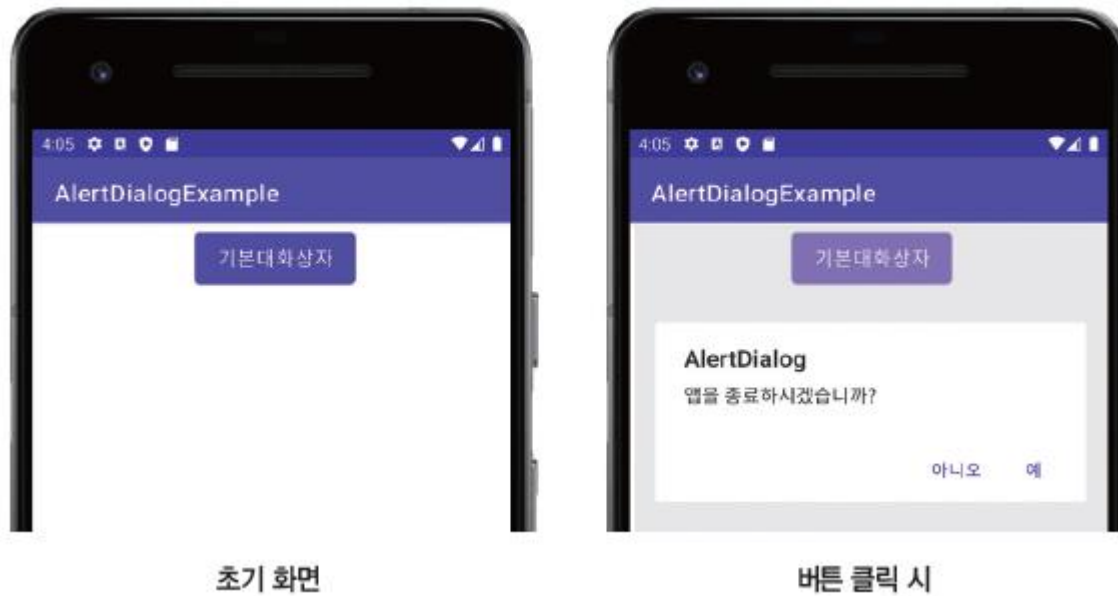
그림 8-3 기본 대화상자 생성 과정



2. 기본 대화상자 만들기

■ 기본 대화상자

그림 8-4 기본 대화상자 생성 결과



2. 기본 대화상자 만들기

■ 기본 대화상자

■ 대화상자 생성하기

▲ AlertDialog.Builder 클래스를 사용한 기본 구문

```
AlertDialog.Builder 객체 변수(인스턴스 변수) = AlertDialog.Builder(Context context);  
AlertDialog.Builder 객체 변수(인스턴스 변수) = AlertDialog.Builder(Context context, int  
themeResId);
```

```
import androidx.appcompat.app.AlertDialog;  
...  
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        Button button = findViewById(R.id.button);  
  
        button.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                AlertDialog.Builder alertDialog = new AlertDialog.Builder(MainActivity.  
this); // 인스턴스 변수 생성  
            }  
        });  
    }  
}
```

2. 기본 대화상자 만들기

■ 기본 대화상자

■ 레이아웃 구성하기

표 8-3 대화상자 레이아웃 구성 요소의 메서드

메서드	설명
<code>setTitle(int titleId)</code> <code>setTitle(CharSequence title)</code>	선택적 구성 요소로, 기본 대화상자의 제목을 설정합니다.
<code>setIcon(Drawable icon)</code> <code>setIcon(int iconId)</code>	제목 앞에 아이콘을 설정합니다.
<code>setMessage(int messageId)</code> <code>setMessage(CharSequence message)</code>	기본 대화상자에 표시할 메시지를 설정합니다.
<code>setCancelable(boolean cancelable)</code>	<code>true/false</code> 를 설정해서 대화상자 영역 외부를 클릭하여 기본 대화상자를 취소(닫기)합니다.
<code>setPositiveButton(...)</code>	기본 대화상자의 예/확인 버튼을 설정하고 버튼의 클릭 이벤트를 구현할 수 있습니다.
<code>setNegativeButton(...)</code>	기본 대화상자의 아니요/취소 버튼을 설정하고 버튼의 클릭 이벤트를 구현할 수 있습니다.
<code>setNeutralButton(...)</code>	기본 대화상자의 중립 버튼(Neutral Button)을 설정하고 버튼의 클릭 이벤트를 구현할 수 있습니다.

2. 기본 대화상자 만들기

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);
```

```
        Button button = findViewById(R.id.button);  
  
        button.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View view) {  
                AlertDialog.Builder alertDialog = new AlertDialog.Builder(MainActivity.  
this);  
  
                alertDialog.setTitle("AlertDialog"); // 제목 표시  
                alertDialog.setMessage("앱을 종료하시겠습니까?"); // 메시지 내용 표시  
                alertDialog.setPositiveButton("예", null); // '예' 버튼 표시  
                alertDialog.setNegativeButton("아니요", null); // '아니요' 버튼 표시  
            }  
        });  
    }  
}
```



2. 기본 대화상자 만들기

■ 기본 대화상자

- 대화상자 출력하기

객체 변수(인스턴스 변수).show()

```
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Button button = findViewById(R.id.button);

        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                AlertDialog.Builder alertDialog = new AlertDialog.Builder(MainActivity.
this);
                ...
                alertDialog.show() // 대화상자 화면 표시 ③
            }
        });
    }
}
```

2. 기본 대화상자 만들기

■ 기본 대화상자

- 버튼 이벤트 처리하기

```
setPositiveButton(int textId, DialogInterface.OnClickListener listener)
setPositiveButton(CharSequence text, DialogInterface.OnClickListener listener)

setNegativeButton(CharSequence text, DialogInterface.OnClickListener listener)
setNegativeButton(int textId, DialogInterface.OnClickListener listener)

setNegativeButton(int textId, DialogInterface.OnClickListener listener)
setNeutralButton(CharSequence text, DialogInterface.OnClickListener listener)
setNeutralButton(int textId, DialogInterface.OnClickListener listener)
```

▲ setPositiveButton(), setNegativeButton(), setNeutralButton() 메서드 기본 구문

2. 기본 대화상자 만들기

■ 기본 대화상자

■ 버튼 이벤트 처리하기

```
setPositiveButton(int textId, DialogInterface.OnClickListener listener)
setPositiveButton(CharSequence text, DialogInterface.OnClickListener listener)

setNegativeButton(CharSequence text, DialogInterface.OnClickListener listener)
setNegativeButton(int textId, DialogInterface.OnClickListener listener)

setNegativeButton(int textId, DialogInterface.OnClickListener listener)
setNeutralButton(CharSequence text, DialogInterface.OnClickListener listener)
setNeutralButton(int textId, DialogInterface.OnClickListener listener)
```

▲ setPositiveButton(), setNegativeButton(), setNeutralButton() 메서드 기본 구문

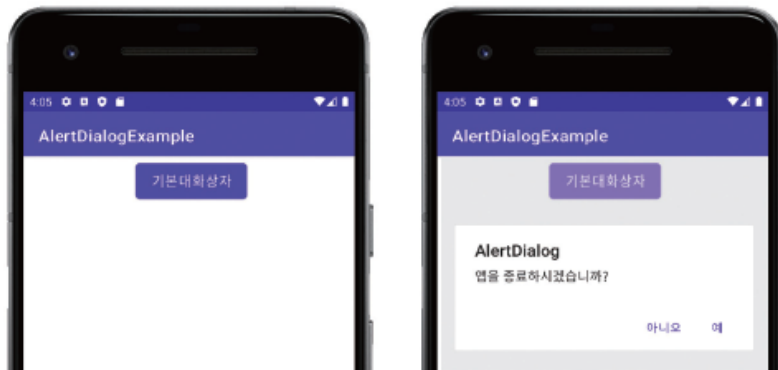
표 8-4 버튼 메서드의 매개 변수

인수	설명
textId	버튼에 표시할 텍스트의 리소스 아이디입니다.
text	버튼에 표시할 텍스트입니다.
listener	버튼을 클릭하면 호출되는 이벤트 리스너입니다.

2. 기본 대화상자 만들기

```
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        ...  
        Button button = findViewById(R.id.button);  
        button.setOnClickListener(new View.OnClickListener() { // 버튼 클릭 이벤트 처리  
            @Override
```

```
                public void onClick(View view) {  
                    AlertDialog.Builder alertDialog = new AlertDialog.Builder(MainActivity.  
this);  
  
                    ...  
                    alertDialog.setPositiveButton("예", new DialogInterface.  
OnClickListener() {  
                        @Override  
                        public void onClick(DialogInterface dialogInterface, int i) {  
                            dialogInterface.dismiss();  
                        }  
                    });  
  
                    alertDialog.setNegativeButton("아니오", new DialogInterface.  
OnClickListener() {  
                        @Override  
                        public void onClick(DialogInterface dialogInterface, int i) {  
                            Toast.makeText(getApplicationContext(), "종료를 취소합니다.", Toast.  
LENGTH_LONG).show();  
                            dialogInterface.cancel();  
                        }  
                    });  
                    alertDialog.show();  
                }  
            });  
        }  
    }  
}
```



2. 기본 대화상자 만들기

■ 목록이 있는 대화상자

그림 8-5 목록이 있는 대화상자의 유형 예

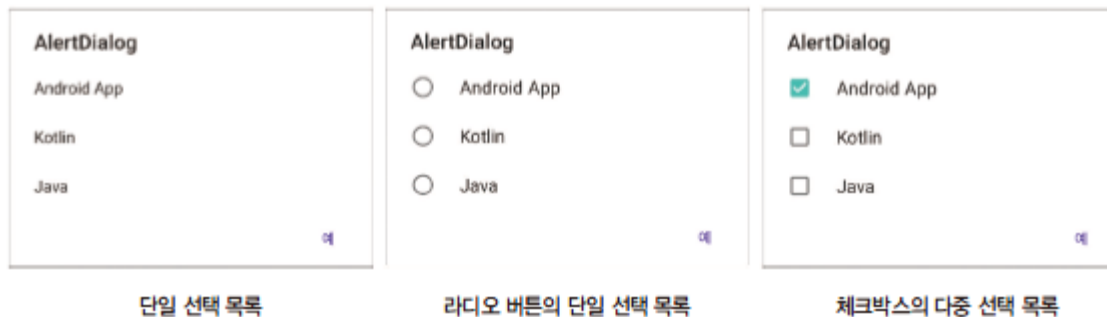


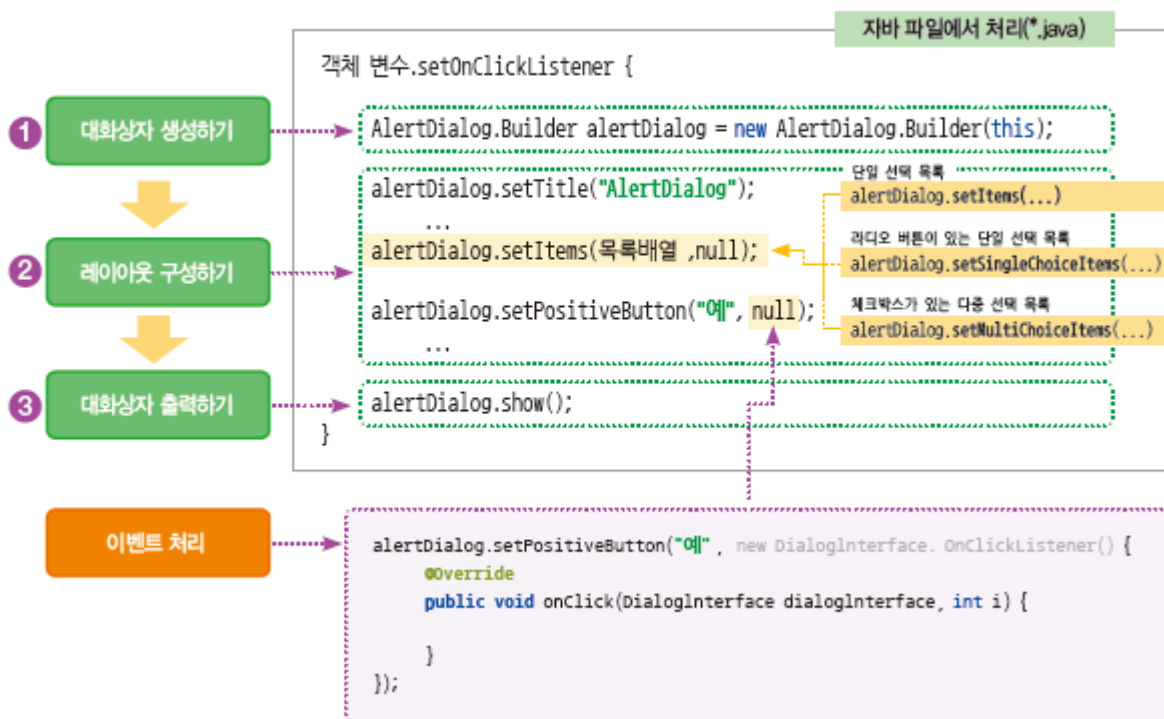
표 8-5 목록이 있는 대화상자의 메서드

유형	메서드
단일 선택 목록	<code>setItems()</code>
라디오 버튼이 있는 단일 선택 목록	<code>setSingleChoiceItems()</code>
체크박스가 있는 다중 선택 목록	<code>setMultiChoiceItems()</code>

2. 기본 대화상자 만들기

■ 목록이 있는 대화상자

그림 8-6 목록이 있는 대화상자 생성 과정



2. 기본 대화상자 만들기

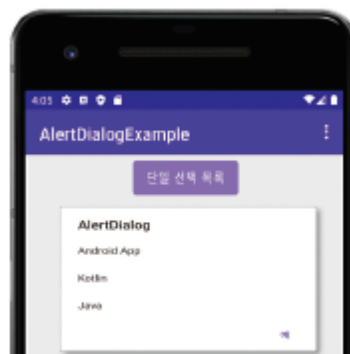
■ 목록이 있는 대화상자

■ 단일 선택 목록 대화상자

```
setItems(int itemId, DialogInterface.OnClickListener listener)  
setItems(CharSequence[] items, DialogInterface.OnClickListener listener)
```

표 8-6 setItems() 메서드의 매개 변수

매개 변수	설명
itemId	배열의 리소스 ID입니다.
items	대화상자에 표시하는 항목을 나타내는 문자열 배열입니다.
listener	목록에서 항목을 선택하는지 감지하는 이벤트 리스너입니다.



단일 선택 목록 대화상자

2. 기본 대화상자 만들기

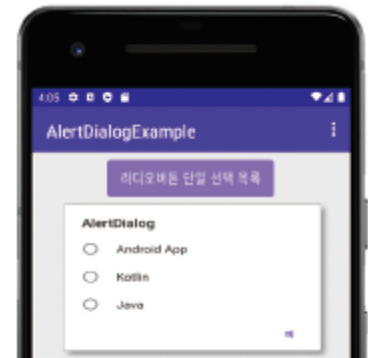
■ 목록이 있는 대화상자

- 라디오 버튼이 있는 단일 선택 목록 대화상자

```
setSingleChoiceItems(int itemsId, int checkedItem, DialogInterface.OnClickListener listener)
setSingleChoiceItems(ListAdapter adapter, int checkedItem, DialogInterface.OnClickListener listener)
setSingleChoiceItems(Cursor cursor, int checkedItem, String labelColumn, DialogInterface.OnClickListener listener)
setSingleChoiceItems(CharSequence[] items, int checkedItem, DialogInterface.OnClickListener listener)
```

표 8-7 setSingleChoiceItems() 메서드의 매개 변수

매개 변수	설명
itemsId	배열의 리소스 아이디입니다.
items	라디오 버튼으로 표시하는 항목을 나타내는 문자열 배열입니다.
checkedItem	선택된 항목의 인덱스 번호(인덱스 번호는 0부터 시작하고, 선택되지 않으면 -1)를 나타냅니다.
listener	목록에서 항목을 선택하는지를 감지하는 이벤트 리스너입니다.
cursor	항목을 검색할 커서입니다.
labelColumn	레이블에 표시할 문자열을 포함하는 커서의 열 이름입니다.
adapter	항목 목록을 제공하는 ListAdapter입니다.



라디오 버튼이 있는 대화상자

2. 기본 대화상자 만들기

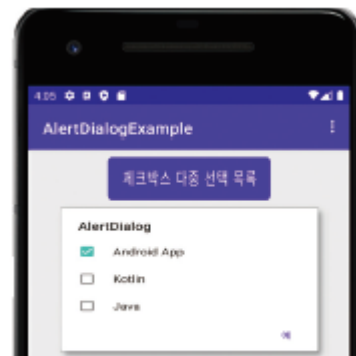
■ 목록이 있는 대화상자

■ 체크박스가 있는 다중 선택 목록 대화상자

```
setMultiChoiceItems(CharSequence[] items, boolean[] checkedItems, DialogInterface.  
OnMultiChoiceClickListener listener)  
setMultiChoiceItems(Cursor cursor, String isCheckedColumn, String labelColumn,  
DialogInterface.OnMultiChoiceClickListener listener)  
setMultiChoiceItems(int itemId, boolean[] checkedItems, DialogInterface.  
OnMultiChoiceClickListener listener)
```

표 8-8 setMultiChoiceItems() 메서드의 매개 변수

매개 변수	설명
itemId	배열의 리소스 아이디입니다.
items	라디오 버튼으로 표시하는 항목을 나타내는 문자열 배열입니다.
checkedItem	항목의 선택 여부를 true 또는 false(선택 시 true)로 나타냅니다.
listener	목록에서 어떤 항목을 선택하는지 감지하는 이벤트 리스너입니다.
cursor	항목을 제공하는 데 사용되는 커서입니다.
labelColumn	레이블에 표시할 문자열을 포함하는 커서의 열 이름입니다.

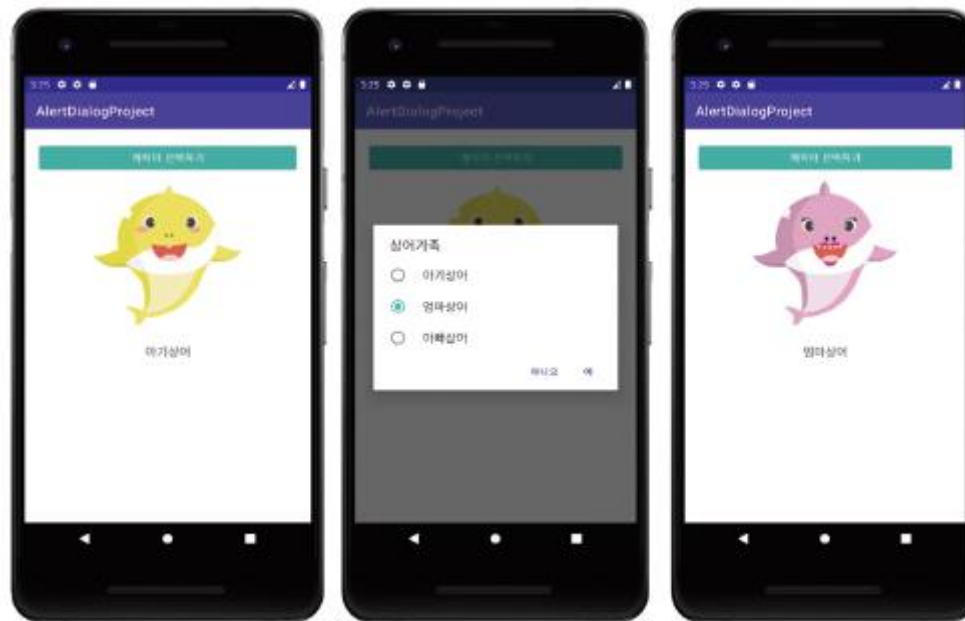


체크박스가 있는 대화상자

2. 기본 대화상자 만들기

■ [실습 예제 8-1] 라디오 버튼이 있는 단일 선택 목록 대화상자 만들기

- 구현 내용 : 액티비티에서 버튼을 클릭하면 캐릭터를 선택할 수 있는 대화상자를 표시합니다. 대화상자에서 캐릭터는 라디오 버튼으로 고를 수 있습니다. 캐릭터를 선택하면 이미지 뷰에 해당 캐릭터를 표시합니다.



03

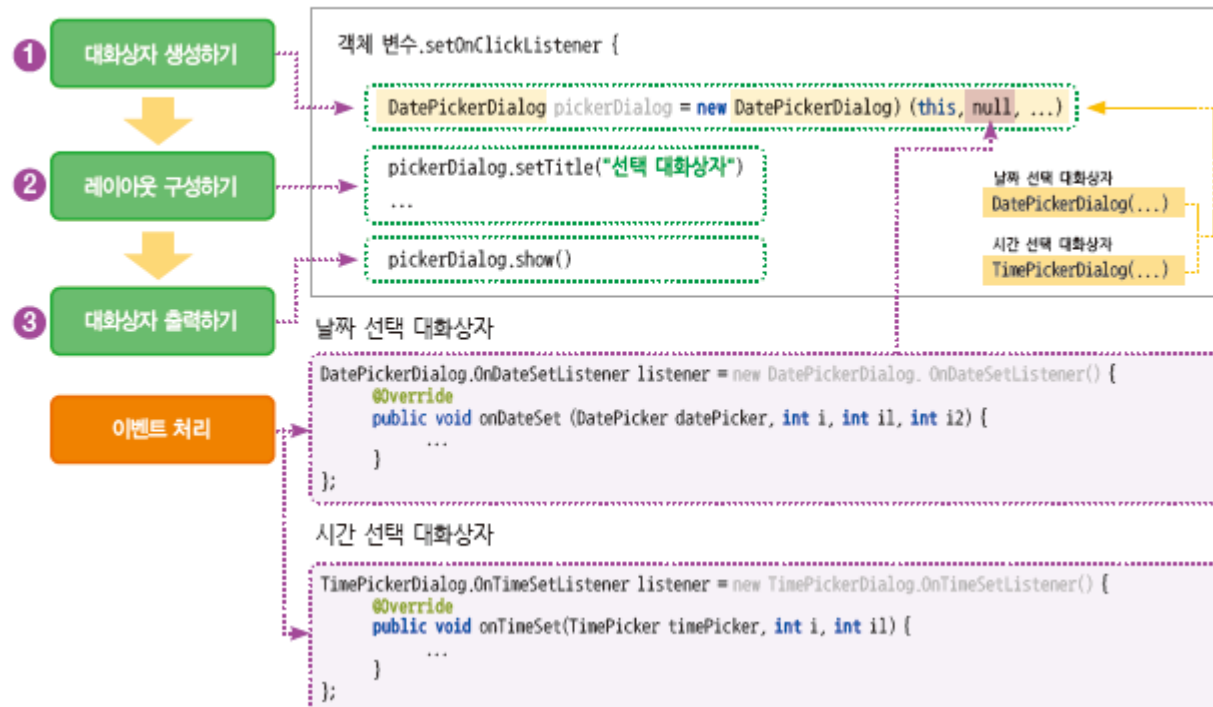
날짜/시간 선택 대화상자 만들기

3. 날짜/시간 선택 대화상자 만들기

■ 날짜/시간 선택 대화상자

- 날짜 또는 시간 선택 대화상자 : 날짜를 선택하는 DatePicker 위젯이나 시간을 선택하는 TimePicker 위젯을 대화상자에 표시

그림 8-8 날짜/시간 선택 대화상자 생성 과정



3. 날짜/시간 선택 대화상자 만들기

■ 날짜/시간 선택 대화상자

그림 8-9 날짜/시간 선택 대화상자 생성 결과



3. 날짜/시간 선택 대화상자 만들기

■ 날짜 선택 대화상자

- 일, 월, 연으로 구성된 날짜를 사용자가 선택할 수 있도록 하는데 사용
- 예를 들어, 철도 승차권이나 항공권 예약 앱, 호텔 숙박 예약 앱 등에서 사용자가 날짜를 선택할 때 사용
- DatePicker 위젯을 사용하여 날짜를 선택하며, 이를 대화상자에 표시하려면 DatePickerDialog 클래스 사용

3. 날짜/시간 선택 대화상자 만들기

■ 날짜 선택 대화상자

```
DatePickerDialog(Context context)
DatePickerDialog(Context context, int themeResId)
DatePickerDialog(Context context, DatePickerDialog.OnDateSetListener listener, int
year, int month, int dayOfMonth)
DatePickerDialog(Context context, int themeResId, DatePickerDialog.OnDateSetListener
listener, int year, int monthOfYear, int dayOfMonth)
```

▲ DatePickerDialog 클래스의 생성자

표 8-9 DatePickerDialog 클래스 생성자 메서드의 매개 변수

매개 변수	설명
context	애플리케이션 컨텍스트를 설정합니다.
themeResId	대화상자에 적용할 테마의 리소스 아이디를 설정합니다.
listener	대화상자에 대한 이벤트 리스너를 설정합니다.
year	대화상자에 표시할 연도를 설정합니다.
month	대화상자에 표시할 월을 설정합니다.
dayOfMonth	대화상자에 표시할 일을 설정합니다.

3. 날짜/시간 선택 대화상자 만들기

```
public class MainActivity extends AppCompatActivity {

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

        Button button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Calendar calendar = Calendar.getInstance();
                int day = calendar.get(Calendar.DAY_OF_MONTH);
                int month = calendar.get(Calendar.MONTH);
                int year = calendar.get(Calendar.YEAR);

                DatePickerDialog.OnDateSetListener listener = new DatePickerDialog.
OnDateSetListener() {
                    @Override
                    public void onDateSet(DatePicker datePicker, int i, int i1, int i2) {
                        Toast.makeText(getApplicationContext(), "선택날짜 : "+i+ "년 " +(i1
+ 1)+"월 " + i2+"일", Toast.LENGTH_LONG).show();
                    }
                };

                DatePickerDialog pickerDialog = new DatePickerDialog(MainActivity.this,
listener, year, month, day); ❶

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

이벤트 처리

3. 날짜/시간 선택 대화상자 만들기

■ 시간 선택 대화상자

- 사용자가 AM/PM 모드 또는 24시간 모드로 시간을 선택할 수 있도록 하는 UI 요소
- 안드로이드 기본 앱 알람 등에서 자주 사용
- 시간 선택 : TimePicker 위젯 사용, TimePicker 위젯을 대화상자에 표시하려면 Time PickerDialog 클래스 사용

3. 날짜/시간 선택 대화상자 만들기

■ 시간 선택 대화상자

▼ TimePickerDialog 클래스의 생성자

```
TimePickerDialog(Context context, TimePickerDialog.OnTimeSetListener listener, int  
hourOfDay, int minute, boolean is24HourView)  
TimePickerDialog(Context context, int themeResId, TimePickerDialog.OnTimeSetListener  
listener, int hourOfDay, int minute, boolean is24HourView)
```

표 8-10 TimePickerDialog 클래스 생성자 메서드의 매개 변수

매개 변수	설명
context	애플리케이션 컨텍스트를 설정합니다.
listener	대화상자에 대한 이벤트 리스너를 설정합니다.
hourOfDay	대화상자에 표시할 시간을 설정합니다.
minute	대화상자에 표시할 분을 설정합니다.
is24HourView	false로 설정하면 시간을 24시간 형식으로 표시합니다.
themeResId	대화상자에 적용할 테마의 리소스 아이디를 설정합니다.

3. 날짜/시간 선택 대화상자 만들기

```
public class MainActivity extends AppCompatActivity {

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

        Button button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Calendar calendar = Calendar.getInstance();
                int hour = calendar.get(Calendar.HOUR_OF_DAY);
                int minute = calendar.get(Calendar.MINUTE);

                TimePickerDialog.OnTimeSetListener listener = new TimePickerDialog.
                OnTimeSetListener() {
                    @Override
                    public void onTimeSet(TimePicker timePicker, int i, int i1) {
                        Toast.makeText(getApplicationContext(), "선택시간 : "+i+ "시 " +i1
                        +"분", Toast.LENGTH_LONG).show();
                    }
                };

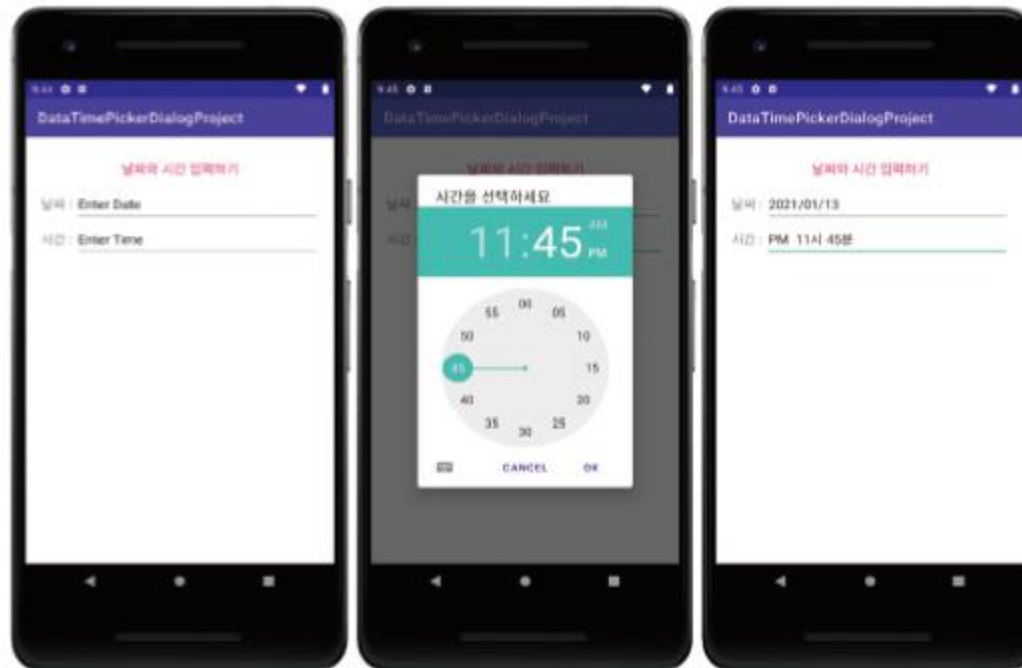
                TimePickerDialog timePickerDialog = new TimePickerDialog(MainActivity.
                this, listener, hour, minute, true); ❶
                timePickerDialog.show(); ❷
            }
        });
    }
}
```

이벤트 처리

3. 날짜/시간 선택 대화상자 만들기

■ [실습 예제 8-2] 에디트 텍스트에 입력된 텍스트 읽어오기

- 구현 내용 : 날짜와 시간을 입력하는 에디트 텍스트를 만들고 날짜 입력 에디트 텍스트를 클릭하면 날짜 선택 대화상자, 시간 입력 에디트 텍스트를 클릭하면 시간 선택 대화상자를 표시합니다. 날짜 선택 대화상자에서 날짜를 선택한 후 <OK> 버튼을 클릭하면 날짜 입력 에디트 텍스트에 지정한 날짜가 입력되고, 시간 선택 대화상자에서 시간을 선택한 후 <OK> 버튼을 클릭하면 시간 입력 에디트 텍스트에 선택한 시간이 입력됩니다.



04

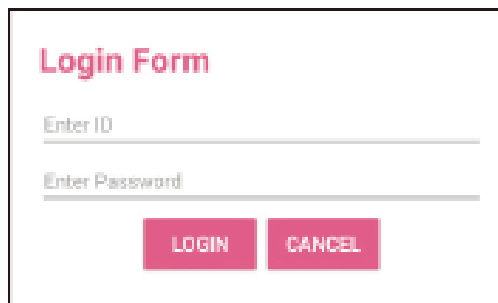
사용자 정의 대화상자 만들기

4. 사용자 정의 대화상자 만들기

■ 사용자 정의 대화상자

- 기본 대화상자에 필요한 디자인이나 구성 요소를 사용자가 원하는 대로 넣을 수 있는 기능
- AlertDialog 클래스에서는 기본 대화상자를 구성하기 위한 메서드들로 제목, 메시지, 버튼 등을 제공. 이를 활용하여 레이아웃을 구성하거나 레이아웃 XML 파일에 직접 구성 요소들을 추가 가능

그림 8-10 커스텀 대화상자의 레이아웃 구성 예



The image shows a custom login dialog box. It has a title bar with the text "Login Form" in red. Below the title bar, there are two input fields. The first input field is labeled "Enter ID" and the second input field is labeled "Enter Password". Both labels are in a light gray font. Below the input fields, there are two buttons: "LOGIN" and "CANCEL". Both buttons are red with white text.

4. 사용자 정의 대화상자 만들기

■ 사용자 정의 대화상자

```
View inflate(int resource, ViewGroup root)
View inflate(XmlPullParser parser, ViewGroup root)
View inflate(XmlPullParser parser, ViewGroup root, boolean attachToRoot)
```

▲ LayoutInflater 클래스의 inflate() 메서드로 새로운 뷰를 생성하는 구문

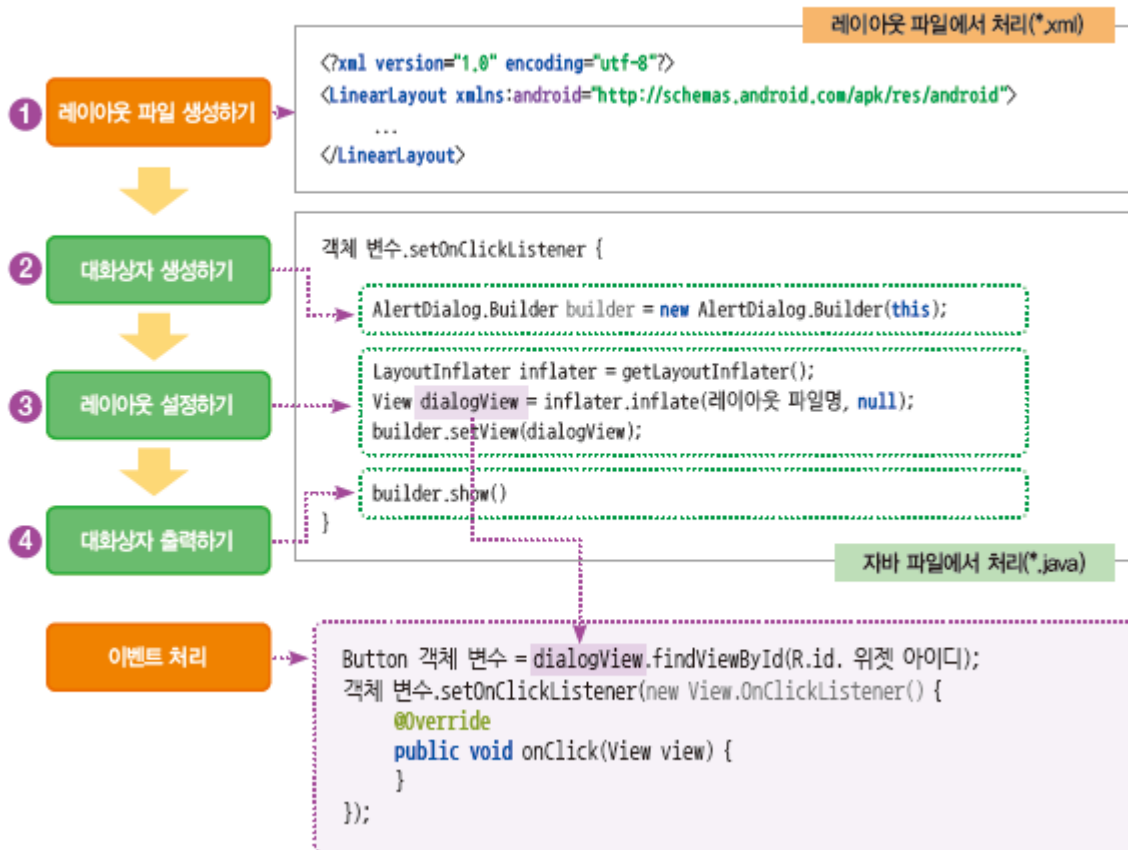
표 8-11 메서드의 매개 변수

매개 변수	설명
resource	블러우 레이아웃 파일명(예 : R.layout.레이아웃 파일명)입니다.
root	선택 사항으로 블러우 상위 뷰를 설정합니다. 이 값은 null일 수 있습니다.
parser	블러우 로드 파일명(XmlPullParser 객체)입니다.
attachToRoot	생성한 뷰를 root의 자식으로 만들지를 설정합니다. false인 경우 root는 생성하는 뷰의 Layout-Params을 생성하는 데 사용됩니다. 이 값은 null일 수 없습니다.

4. 사용자 정의 대화상자 만들기

■ 사용자 정의 대화상자

그림 8-11 사용자 정의 대화상자 생성 과정



4. 사용자 정의 대화상자 만들기

■ 사용자 정의 대화상자



4. 사용자 정의 대화상자 만들기

■ 사용자 정의 대화상자

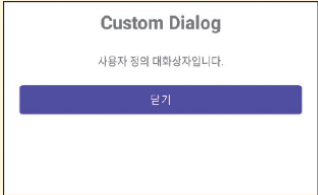
1 dialog_layout.xml

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

    <TextView
        android:id="@+id/textView"
        android:text="Custom Dialog"
        ... />

    <TextView
        android:id="@+id/textView2"
        android:text="사용자 정의 대화상자입니다."
        ... />

    ... />
    <Button
        android:id="@+id/close"
        android:text="닫기"
        ... />
</LinearLayout>
```



MainActivity.java

```
public class MainActivity extends AppCompatActivity {

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

        Button button = findViewById(R.id.button);
        button.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity.this);

                LayoutInflater inflater = getLayoutInflater();
                View dialogView = inflater.inflate(R.layout.dialog_layout, null);

                builder.setView(dialogView);

                AlertDialog dialog = builder.show();

                Button close = dialogView.findViewById(R.id.close);

                close.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View view) {
                        TextView tv = dialogView.findViewById(R.id.textView2);
                        dialog.dismiss();
                        Toast.makeText(MainActivity.this, tv.getText(), Toast.LENGTH_
LONG).show();
                    }
                });
            }
        });
    }
}
```

2 대화상자 생성

3 레이아웃 설정

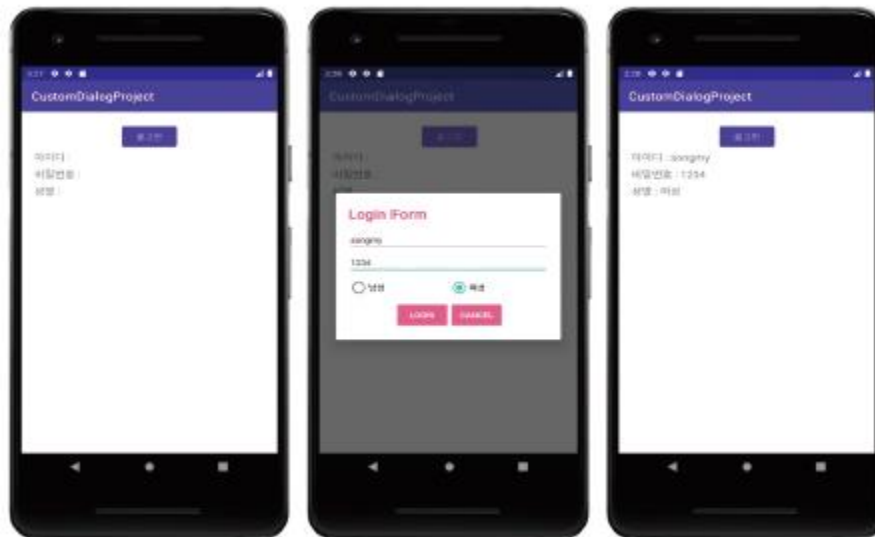
4 대화상자 출력

버튼 클릭 이벤트 처리

4. 사용자 정의 대화상자 만들기

■ [실습 예제 8-3] 사용자 정의 대화상자를 이용해 로그인 정보 출력하기

- 구현 내용 : 아이디, 비밀번호, 성별로 구성된 사용자 정의 대화상자를 만들고, 액티비티에서 버튼을 클릭하면 사용자 정의 대화상자를 표시합니다. 대화상자에 로그인 정보를 입력하고 <OK> 버튼을 클릭하면 입력한 로그인 정보를 화면에 출력합니다.



PROJECT. [도서 쇼핑몰] 로그인과 장바구니 등록 대화상자 만들기

장바구니 등록 대화상자 만들기 결과 화면

도서 쇼핑몰 메인 화면의 내비게이션 바 메뉴 중 <로그인> 버튼을 누르면 로그인 대화상자가 나타나고 로그인 정보를 입력하면 입력한 정보가 출력됩니다.

로그인 대화상자 만들기 결과 화면

도서 상세 정보 화면에 <장바구니 담기> 버튼을 만들고 해당 버튼을 누르면 해당 도서를 장바구니에 추가하는지 확인하는 대화상자를 나타냅니다. 대화상자에서 <예> 버튼을 누르면 장바구니 액션 버튼에 상품 개수가 표시됩니다.

