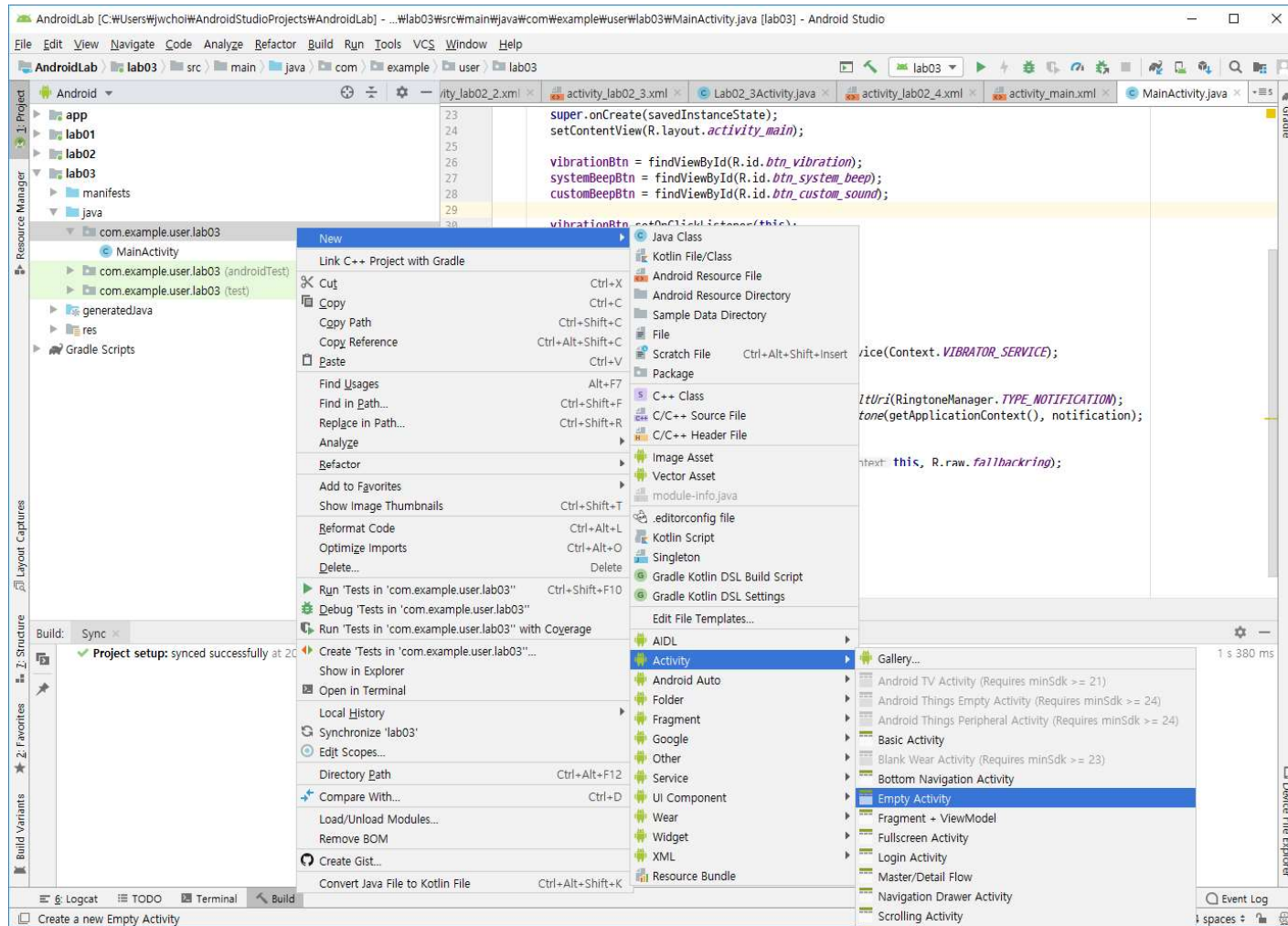




다이얼로그 띄우기

Step 1 _ 액티비티 생성




New Android Activity

 **Configure Activity**
Android Studio



Creates a new empty activity



Activity Name:

Lab03_2Activity

☒ Generate Layout File

Layout Name:

activity_lab03_2

☒ Launcher Activity

Package name:

com.example.user.lab03

Source Language:

Java

If true, this activity will have a CATEGORY_LAUNCHER intent filter, making it visible in the launcher

Previous


Next

Cancel

Finish

Step 2 _ 커스텀 다이얼로그 레이아웃 XML 작성



 New Resource File ✕

File name:

Root element:

Source set:

Directory name:

Available qualifiers:

Country Code

Network Code

Locale

Layout Direction

Smallest Screen Width

Screen Width

Screen Height

Size

Ratio

Orientation

UI Mode

>>

<<

Chosen qualifiers:

Nothing to show

OK

Cancel

Help

Step 3 _ dialog_layout.xml 작성

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/text1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="USB 디버깅을 허용하시겠습니까?"
        android:textStyle="bold"
        android:textSize="15dp"
        android:layout_marginLeft="32dp"
        android:layout_marginTop="32dp" />

    <TextView
        android:id="@+id/text2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="컴퓨터 RSA키 지문: "
        android:layout_below="@id/text1"
        android:layout_alignLeft="@id/text1" />
```

```
<TextView
    android:id="@+id/text3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="17:AA:BB:77:88:DD:98:7E"
    android:layout_below="@id/text2"
    android:layout_alignLeft="@id/text2" />
```

```
<CheckBox
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="이 컴퓨터에서 항상 허용"
    android:layout_below="@id/text3"
    android:layout_alignLeft="@id/text3" />
```

```
</RelativeLayout>
```

Step 4 _ activity_lab03_2.xml 작성

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">

    <Button
        android:id="@+id/btn_alert"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="alert dialog" />

    <Button
        android:id="@+id/btn_list"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="list dialog" />

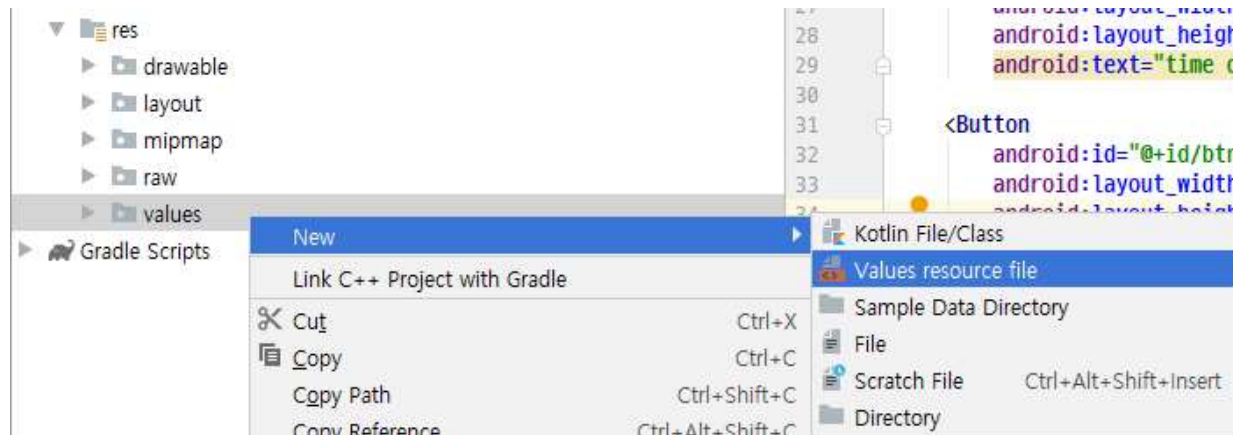
    <Button
        android:id="@+id/btn_date"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="date dialog" />
```

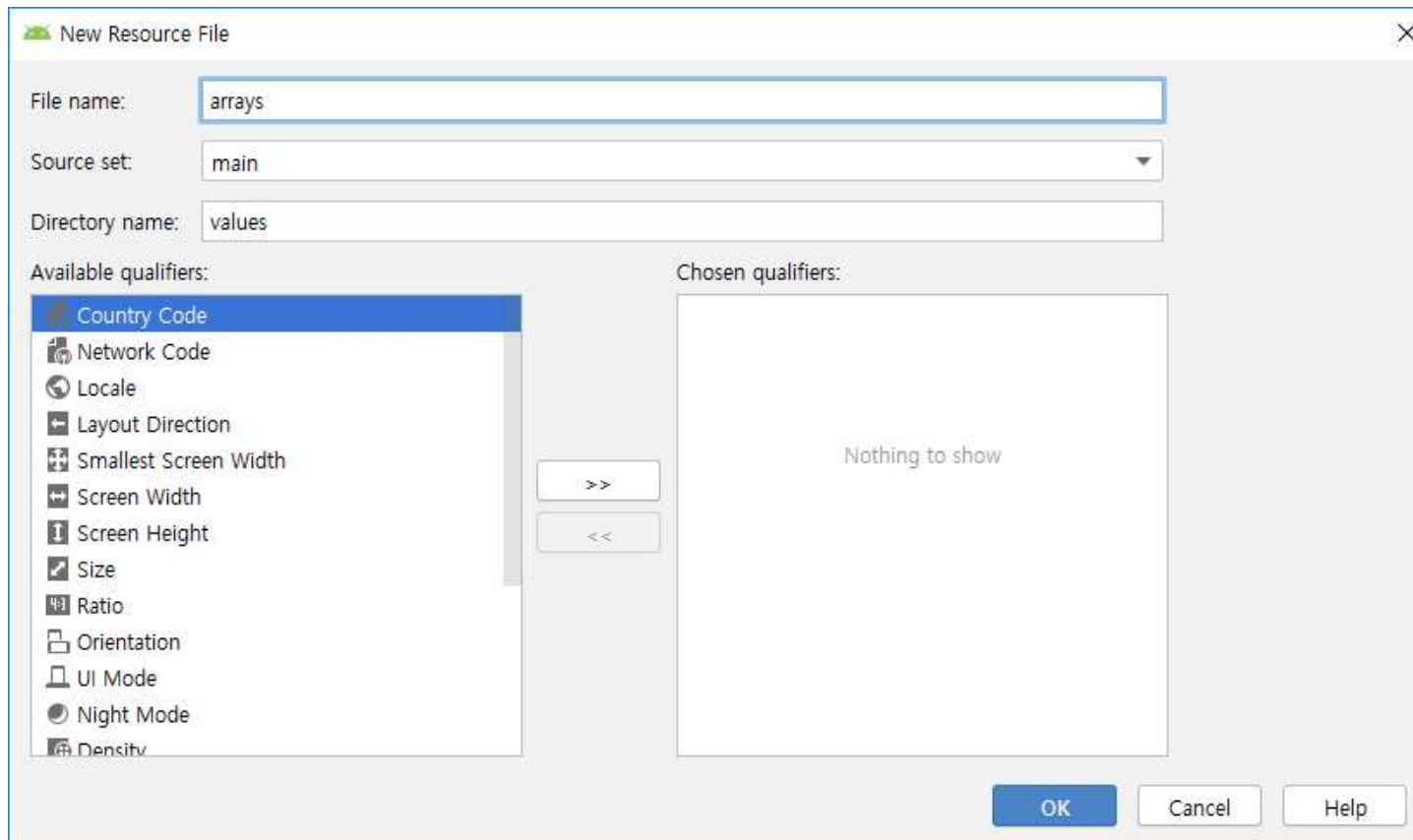
```
        <Button
            android:id="@+id/btn_time"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="time dialog" />

        <Button
            android:id="@+id/btn_custom"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:text="custom dialog" />

    </LinearLayout>
```

Step 5 _ arrays.xml 파일 생성





★ **Note:** A string array is a simple resource that is referenced using the value provided in the **name** attribute (not the name of the XML file). As such, you can combine string array resources with other simple resources in the one XML file, under one **<resources>** element.

Step 6 _ arrays.xml 작성

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="dialog_array">
        <item>기본 알람 소리</item>
        <item>Argon</item>
        <item>Awaken</item>
        <item>Bounce</item>
        <item>Carbon</item>
    </string-array>
</resources>
```

Step 7 _ Lab03_2Activity.java 작성

```
public class Lab03_2Activity extends AppCompatActivity implements View.OnClickListener {
```

```
    Button alertBtn;
```

```
    Button listBtn;
```

```
    Button dateBtn;
```

```
    Button timeBtn;
```

```
    Button customDialogBtn;
```

```
    // 이벤트를 처리를 위해 dialog 객체를 멤버변수로 선언
```

```
    AlertDialog customDialog; // android.app.AlertDialog
```

```
    AlertDialog listDialog; // android.app.AlertDialog
```

```
    AlertDialog alertDialog; // android.app.AlertDialog
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_lab03_2);

    // View 객체 획득
    alertBtn = findViewById(R.id.btn_alert);
    listBtn = findViewById(R.id.btn_list);
    dateBtn = findViewById(R.id.btn_date);
    timeBtn = findViewById(R.id.btn_time);
    customDialogBtn = findViewById(R.id.btn_custom);

    // 버튼 이벤트 등록
    alertBtn.setOnClickListener(this);
    listBtn.setOnClickListener(this);
    dateBtn.setOnClickListener(this);
    timeBtn.setOnClickListener(this);
    customDialogBtn.setOnClickListener(this);
}
```

// 매개변수의 문자열을 Toast로 띄우는 개발자 함수

```
private void showToast(String message){  
    Toast toast = Toast.makeText(this, message, Toast.LENGTH_SHORT);  
    toast.show();  
}
```

makeText

Added in API level 1

```
public static Toast makeText (Context context,  
                             CharSequence text,  
                             int duration)
```



Make a standard toast that just contains a text view.

Parameters	
context	Context: The context to use. Usually your Application or Activity object.
text	CharSequence: The text to show. Can be formatted text.
duration	int: How long to display the message. Either LENGTH_SHORT or LENGTH_LONG Value is LENGTH_SHORT , or LENGTH_LONG

// Dialog Button 이벤트 처리

```
DialogInterface.OnClickListener dialogListener = new DialogInterface.OnClickListener() {  
    @Override  
    public void onClick(DialogInterface dialog, int which) {  
        if (dialog == customDialog && which == DialogInterface.BUTTON_POSITIVE) {  
            showToast("custom dialog 확인 click....");  
        } else if (dialog == listDialog) {  
            // 목록 dialog의 항목이 선택되었을 때 항목 문자열 획득  
            String[] data = getResources().getStringArray(R.array.dialog_array);  
            showToast(data[which] + " 선택하셨습니다.");  
        } else if (dialog == alertDialog && which == DialogInterface.BUTTON_POSITIVE) {  
            showToast("alert dialog ok click...");  
        }  
    }  
};
```

@Override

```
public void onClick(View view) {
```

```
    if (view == alertBtn) {
```

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

```
        builder.setIcon(android.R.drawable.ic_dialog_alert);
```

```
        builder.setTitle("알림");
```

```
        builder.setMessage("정말 종료 하시겠습니까?");
```

```
        builder.setPositiveButton("OK", dialogListener);
```

```
        builder.setNegativeButton("NO", null);
```

```
        alertDialog = builder.create();
```

```
        alertDialog.show();
```

```
    }
```

`AlertDialog.Builder(Context context)`

Creates a builder for an alert dialog that uses the default alert dialog theme.



```
public AlertDialog.Builder setPositiveButton (CharSequence text,  
                                             DialogInterface.OnClickListener listener)
```

```
public AlertDialog.Builder setNegativeButton (CharSequence text,  
                                             DialogInterface.OnClickListener listener)
```

```
else if (view == listBtn) {  
    AlertDialog.Builder builder = new AlertDialog.Builder(this);  
    builder.setTitle("알람 벨소리");  
    builder.setSingleChoiceItems(R.array.dialog_array, 0, dialogListener);  
  
    builder.setPositiveButton("확인", null);  
    builder.setNegativeButton("취소", null);  
    listDialog = builder.create();  
    listDialog.show();  
}
```

항목 구성을 위한 배열 리소스

항목 선택 이벤트 핸들러

초기 선택 항목의 인덱스 (If -1 no items are checked)

위의 함수를 사용하면 항목 옆에 체크박스가 나옴


```
else if (view == dateBtn) {  
    // 현재 날짜로 dialog를 띄우기 위해 날짜를 구함  
    Calendar c = Calendar.getInstance(); // java.util.Calendar  
    int year = c.get(Calendar.YEAR);  
    int month = c.get(Calendar.MONTH);  
    int day = c.get(Calendar.DAY_OF_MONTH);  
  
    DatePickerDialog dateDialog = new DatePickerDialog(this, new DatePickerDialog.OnDateSetListener() {  
        @Override  
        public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {  
            showToast(year + "-" + (month + 1) + "-" + dayOfMonth);  
        }  
    }, year, month, day);  
    dateDialog.show();  
}
```

```

else if (view == timeBtn) {
    // 현재 시간으로 Dialog를 띄우기 위해 시간을 구함
    Calendar c = Calendar.getInstance(); // java.util.Calendar
    int hour = c.get(Calendar.HOUR_OF_DAY); ← HOUR_OF_DAY is used for the 24-hour clock
    int minute = c.get(Calendar.MINUTE);

    TimePickerDialog timeDialog = new TimePickerDialog(this, new TimePickerDialog.OnTimeSetListener() {

        @Override
        public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
            showToast(hourOfDay + ":" + minute);
        }
    }, hour, minute, false);
    timeDialog.show();
}

```

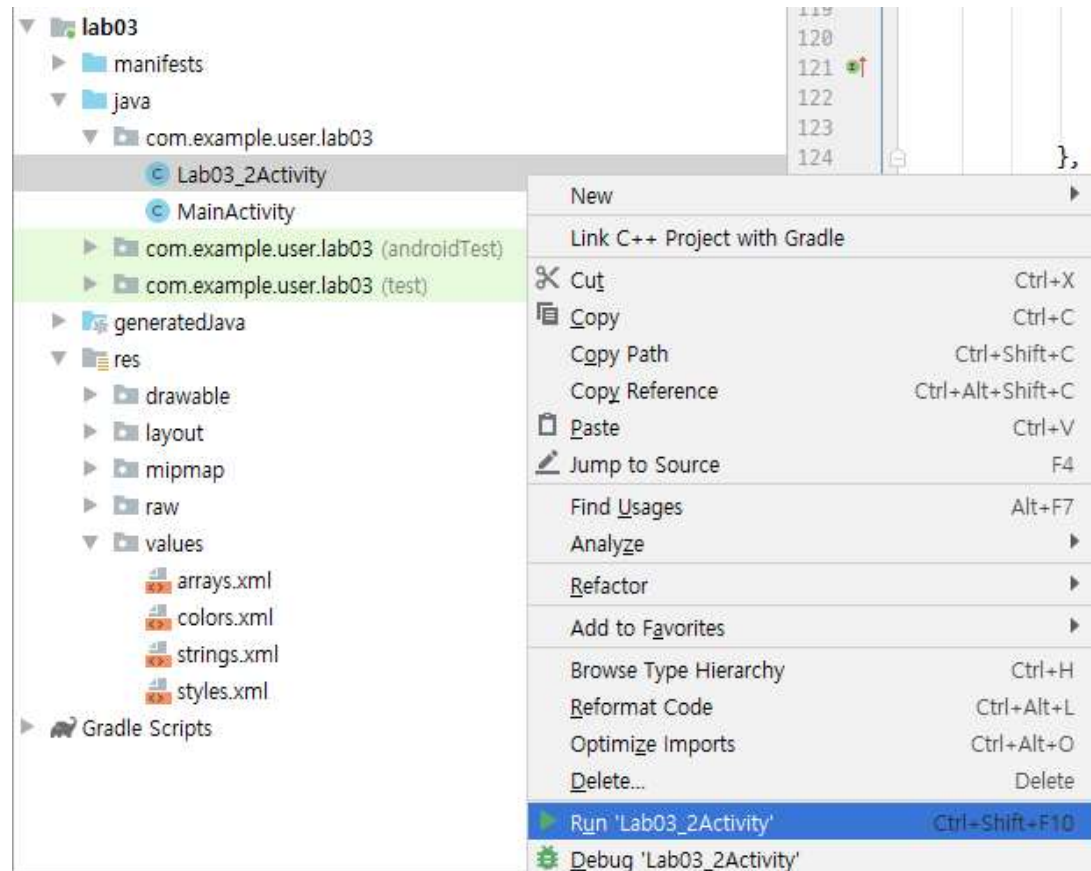
true: 24시간 체계
 false: 12시간 체계

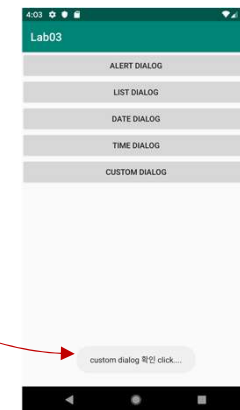
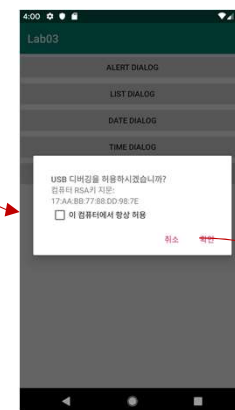
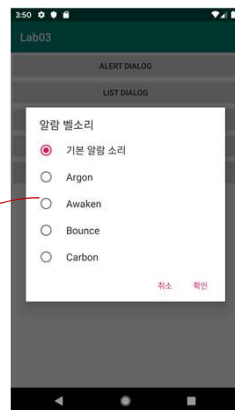
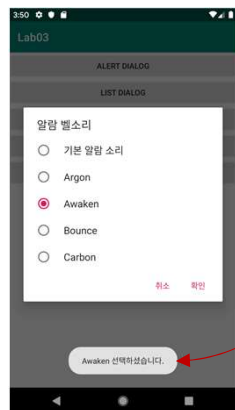
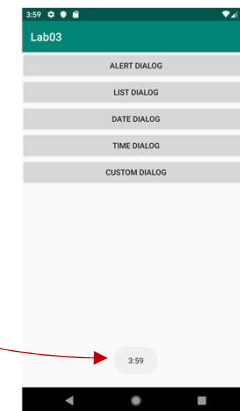
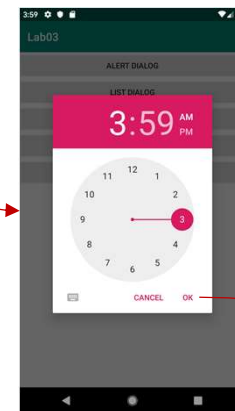
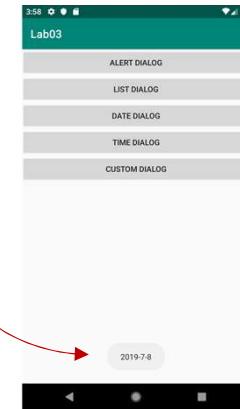
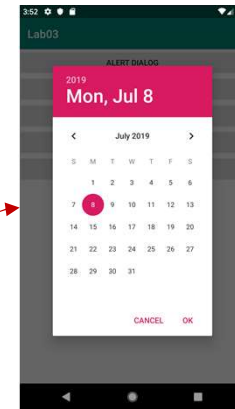
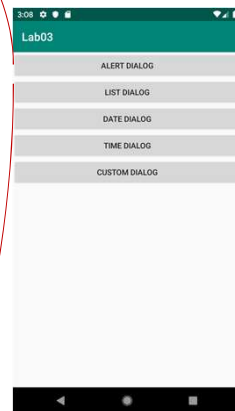
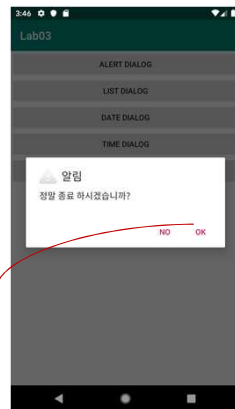
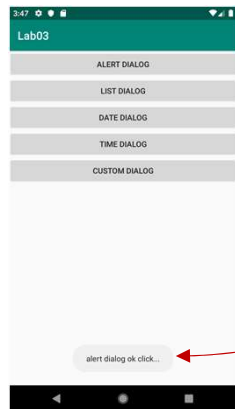
```
else if (view == customDialogBtn) {  
    AlertDialog.Builder builder = new AlertDialog.Builder(this);  
    // custom dialog를 위한 layout xml 초기화  
    LayoutInflater inflater = (LayoutInflater) getSystemService(LAYOUT_INFLATER_SERVICE);  
    View customDialogView = inflater.inflate(R.layout.dialog_layout, null);  
    builder.setView(customDialogView);  
    builder.setPositiveButton("확인", dialogListener);  
    builder.setNegativeButton("취소", null);  
  
    customDialog = builder.create();  
    customDialog.show();  
}
```

← 커스텀 다이얼로그는 개발자가 직접 구성한 뷰를 setView() 함수를 통해 다이얼로그 본문으로 지정하여 구현

```
}  
}
```

Step 8 _ Lab03_2Activity.java 실행





AlertDialog

java.lang.Object
↳ android.app.Dialog
↳ android.app.AlertDialog

- AlertDialog가 닫히는 세 가지 경우
 - dialog 하단의 버튼이 눌림
 - 스마트폰의 뒤로 가기 버튼이 눌림
 - dialog 창 밖의 화면을 터치했을 때
- AlertDialog.Builder
 - public AlertDialog.Builder setCancelable (boolean cancelable)
 - 이 함수의 값을 false로 지정하면
 - 뒤로 가기 버튼 뿐 아니라 창밖을 터치해도 닫히지 않음
- Dialog
 - public void setCanceledOnTouchOutside (boolean cancel)
 - 이 함수의 값을 false로 지정하면
 - 창밖을 터치했을 때 닫히지 않음

DatePickerDialog

```
java.lang.Object
└─ android.app.Dialog
    └─ android.app.AlertDialog
        └─ android.app.DatePickerDialog
```

DatePickerDialog

Added in API level 1

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



Creates a new date picker dialog for the specified date using the parent context's default date picker dialog theme.

Parameters	
context	Context: the parent context This value must never be <code>null</code> .
listener	DatePickerDialog.OnDateSetListener: the listener to call when the user sets the date This value may be <code>null</code> .
year	int: the initially selected year
month	int: the initially selected month (0-11 for compatibility with <code>Calendar#MONTH</code>)
dayOfMonth	int: the initially selected day of month (1-31, depending on month)

TimePickerDialog

java.lang.Object
↳ android.app.Dialog
↳ android.app.AlertDialog
↳ android.app.TimePickerDialog

TimePickerDialog

Added in API level 1



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

Creates a new time picker dialog.

Parameters	
context	Context: the parent context
listener	TimePickerDialog.OnTimeSetListener: the listener to call when the time is set
hourOfDay	int: the initial hour
minute	int: the initial minute
is24HourView	boolean: whether this is a 24 hour view or AM/PM