



# Dialog 실습

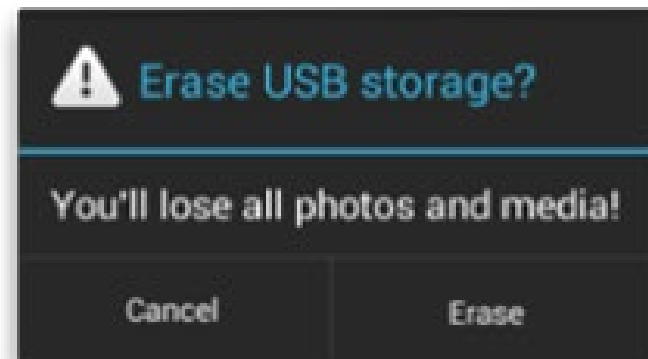
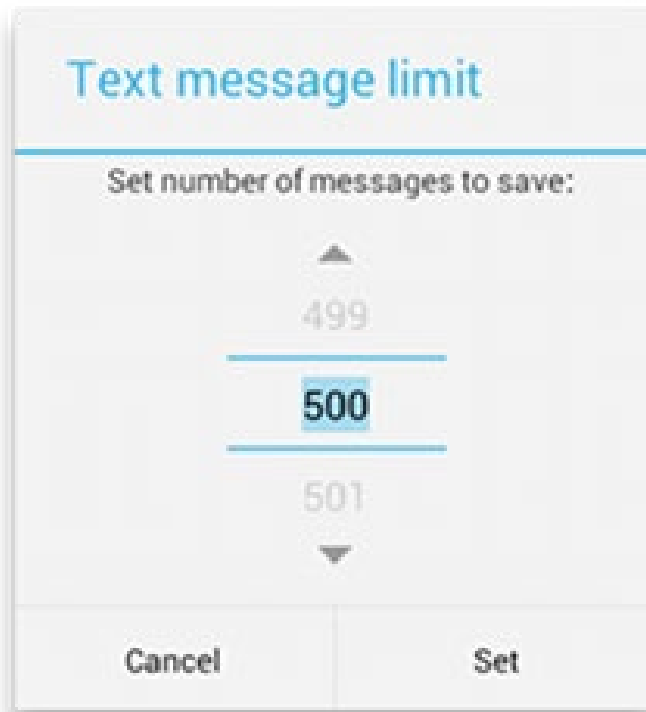
---

배 희호 교수  
경북대학교  
소프트웨어융합과



# Dialog

- 사용자와 상호 작용을 위해 화면에 표시되는 작은 Window
- 사용자에게 중요한 사항을 Message로 출력하고, 사용자로부터 입력을 받아들이는 아주 보편적인 User Interface
- Window의 모달(Modal) 방식





# Dialog

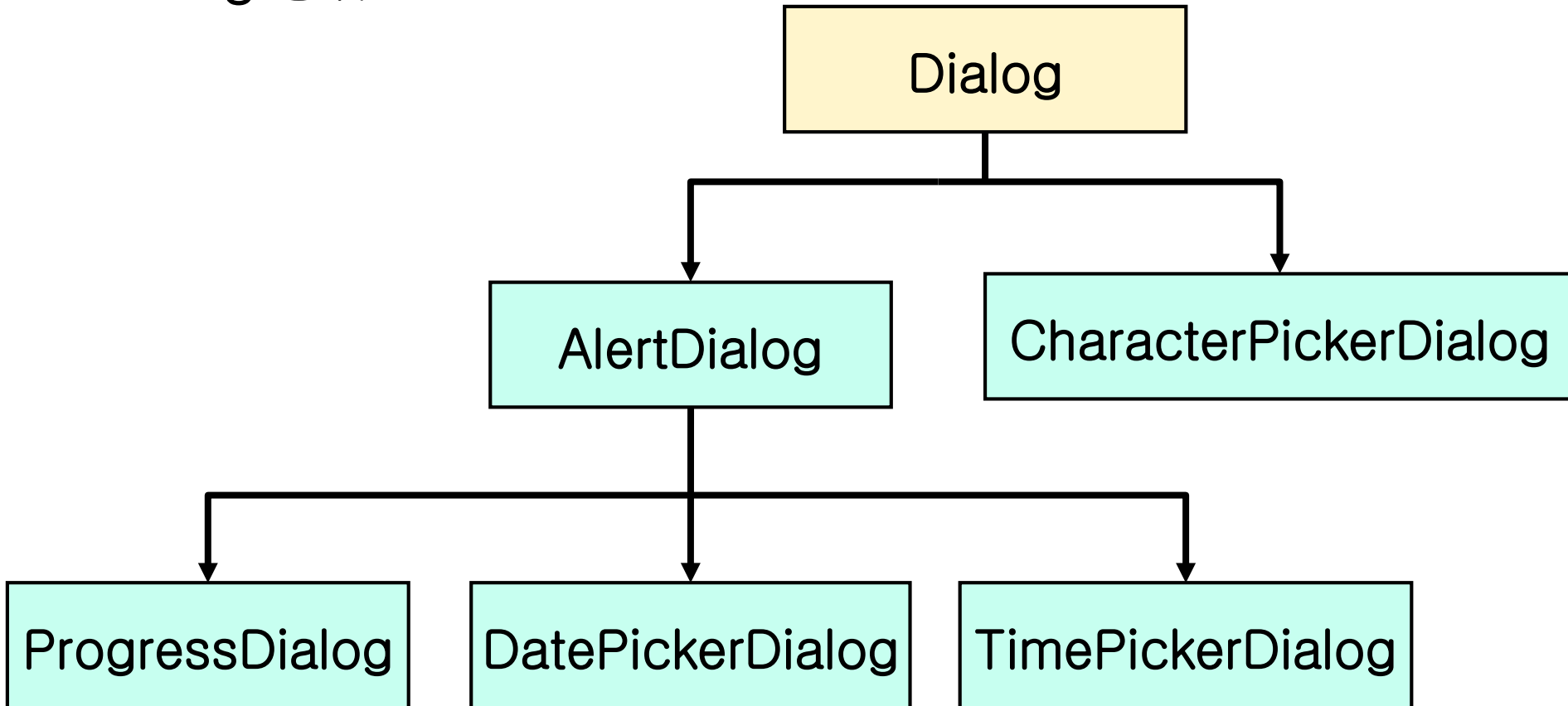


- Dialog는 사용자와 상호 작용 할 수 있는 기능으로 Activity 화면을 다 채우지 않고 일부 화면만 가리는 작은 Window
- Dialog가 표시되면 Dialog의 아래에 위치하게 되는 Activity는 Focus를 잃고, 해당 Dialog에서 사용자가 예/아니오 같은 선택을 하거나 추가적인 정보 입력을하기를 기다림
- 사용자가 응답하기 전까지는 화면에서 사라지지 않고 대기하며 Programming Code도 잠시 대기 상태가 됨
- Dialog는 사용자가 하던 일을 중단 시키고, Alert을 표시하거나 Progress Bar의 진행 상태를 표시하거나, Login 처리와 같은 작업에서 Application에 직접 관계된 짧은 작업등을 수행하기 위해 사용 됨
- 보통 Notification이나 Application에 연관된 간단한 작업을 처리하기 위해 사용



# Dialog

## ■ Dialog 종류





# Dialog



- Dialog 종류

- AlertDialog

- 경고 Message를 표시하거나 사용자에게 선택을 요구할 때 사용
    - Button, List, Custom Layout 등을 포함할 수 있음
    - 0~3개 Button을 가질 수 있는 기본적인 Dialog

- ProgressDialog (deprecated)

- 작업의 진행을 나타내는 Wheel이나 Bar를 표시하는 Dialog
    - 일반적으로 Loading 중이라는 Message나 진행률을 표시할 때 사용
    - 대신 ProgressBar와 AlertDialog를 함께 사용함



# Dialog

## ■ Dialog 종류

### ■ DatePickerDialog

- 사용자가 날짜를 선택하는 Interface를 제공하는 Dialog

- Calendar 형식으로 날짜를 선택할 수 있음

### ■ TimePickerDialog

- 사용자가 시간을 선택하는 Interface를 제공하는 Dialog

- 시계 형식으로 시간(시와 분)을 선택할 수 있음





# Dialog



- Dialog 종류

- CustomDialog

- CustomDialog는 사용자가 완전히 Customized된 Layout을 사용하는 Dialog

- BottomSheetDialog

- BottomSheetDialog는 화면의 하단에서 올라오는 Dialog

- 사용자 Interface를 하단에서 표시하여 화면의 나머지 부분을 여전히 볼 수 있게 해줌



# Dialog



- Dialog 생성 및 표시
  - Activity의 부분으로 생성되고 표시됨
  - 일반적으로 Dialog는 Activity의 onCreateDialog(int) 콜백 메소드안에 생성
    - 이 메소드를 사용하면 Android System은 자동적으로 Activity내에 있는 각 Dialog를 처리
    - 이 메소드는 API Level 8 즉, Android 2.2에서 추가된 메소드
  - 만일 onCreateDialog() 메소드의 외부에서 Dialog를 만들게 되면 Activity에 Dialog가 부착되지 않음
    - 이때는 setOwnerActivity(Activity) 메소드를 사용해서 Dialog를 Activity에 붙일 수 있음





# Dialog



- Dialog 생성 및 표시

- Dialog를 표시할 때는 showDialog(int) 메소드를 호출
  - 화면에 표시되기를 원하는 Dialog의 id 값을 넘겨줌
- Dialog가 처음으로 요청되면 Android는 Activity로부터 onCreateDialog(int) 메소드를 호출
  - 이 메소드는 showDialog(int) 메소드를 사용했을 때 넘겨받는 것과 같은 ID를 받음
- Dialog가 생성된 후에는 메소드의 마지막에 생성된 Dialog의 객체를 반환



# Dialog



- Dialog 생성 및 표시

- Dialog를 표시하기 전에 필요한 경우

- onPrepareDialog(int, Dialog) 콜백 메소드를 호출할 수 있음

- 이 메소드는 Dialog가 열릴 때 마다 속성의 변경이 필요한 경우에 정의해서 사용할 수 있음

- onCreateDialog() 메소드는 처음 Dialog가 표시될 때 한번만 호출되는 것에 비해 onPrepareDialog() 메소드는 Dialog가 열릴 때 마다 호출됨



# Dialog



```
public class DialogTestActivity extends Activity {  
    ...  
    protected Dialog onCreateDialog(int id) {    // 대화상자 생성  
        switch (id) {  
            case DIALOG_PAUSED_ID:  
                return new AlertDialog.Builder(AlertDialogTest.this).create();  
            ...  
        }  
        return null;  
    }  
    ...  
    showDialog(DIALOG_PAUSED_ID);  
    ...  
    dismissDialog(DIALOG_PAUSED_ID);    // 대화상자 제거  
    ...  
}
```



# Dialog



```
static final int PROGRESS_DIALOG = 0;

protected Dialog onCreateDialog(int id) {    //대화상자 생성
    switch(id) {
        case PROGRESS_DIALOG:
            //해당 대화상자 생성
        }
    }

    //대화상자의 내용이 변경되는 경우 자동 실행
    protected void onPrepareDialog(int id, Dialog dialog) {
        switch(id) {
            case PROGRESS_DIALOG:
                //해당 대화상자의 내용의 변경 시 처리할 내용 기술
            }
        }
    }
```



# [참조] Modal과 Modeless의 차이

- Window의 Form(형식)과 Dialog를 Modal 혹은 Modeless로 표시할 수 있음
  - Modal(모달)
    - 간단한 개념으로 Modal 창이 열렸을 때는 기존에 있던 창을 사용하지 못하는 방식
    - 즉 제어권을 독점하게 되어 그 창이 종료하기 전까지는 기존 창을 작업 할 수 없는 방식
    - 중요한 Message를 표시하는 Dialog는 거의 Modal로 나타냄
    - 예) Visual Studio의 정보 대화상자나, api의 messagebox



# [참조] Modal과 Modeless의 차이

- Window의 Form(형식)과 Dialog를 Modal 혹은 Modeless로 표시할 수 있음
  - Modeless(모달리스)
    - Modal과는 반대의 개념으로 창이 열리더라도 기존에 있던 창을 작업 할 수 있음
    - Modeless는 사용자가 순서에 상관없이 접근이 가능 하기 때문에 사용자의 동작에 관계 없이 Application의 상태를 일관성 있게 유지해야 함
    - Programming 하기가 어려움
    - 예) 도구창은 종종 Modeless 형식으로 표시되며, Visual Studio의 편집 메뉴에서 사용 할 수 있는 대화 상자들
    - Modeless 형식은 자주 사용하는 명령이나 정보를 표시 하는데 사용



# [참조] Modal과 Modeless의 차이

	Modal	Modeless
생성	DoModal	Create
종료	EndDialog	DestroyWindow
객체 선언 시	스택, 힙 가능	동적 할당만 가능하므로 힙(Heap)만 가능



# Dialog

## ■ Dialog 설정 순서

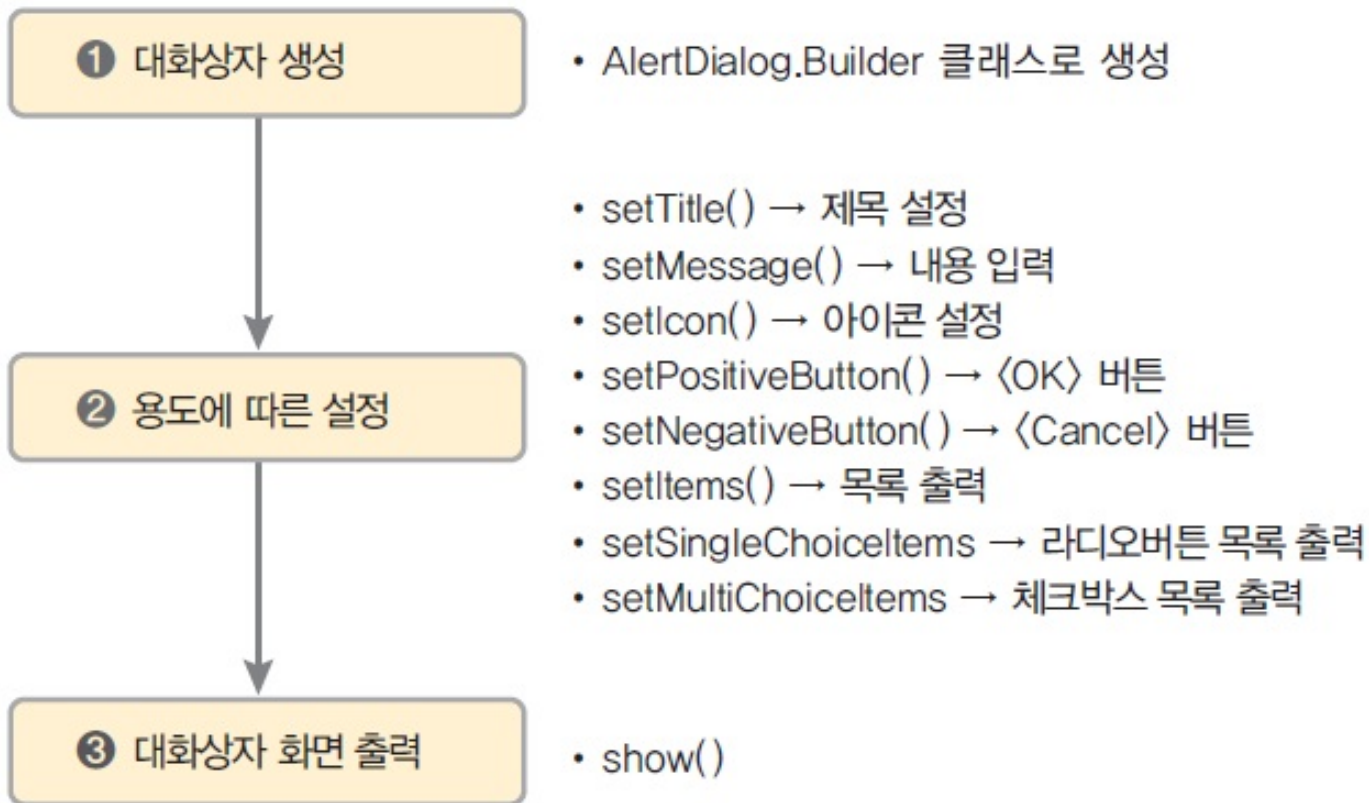


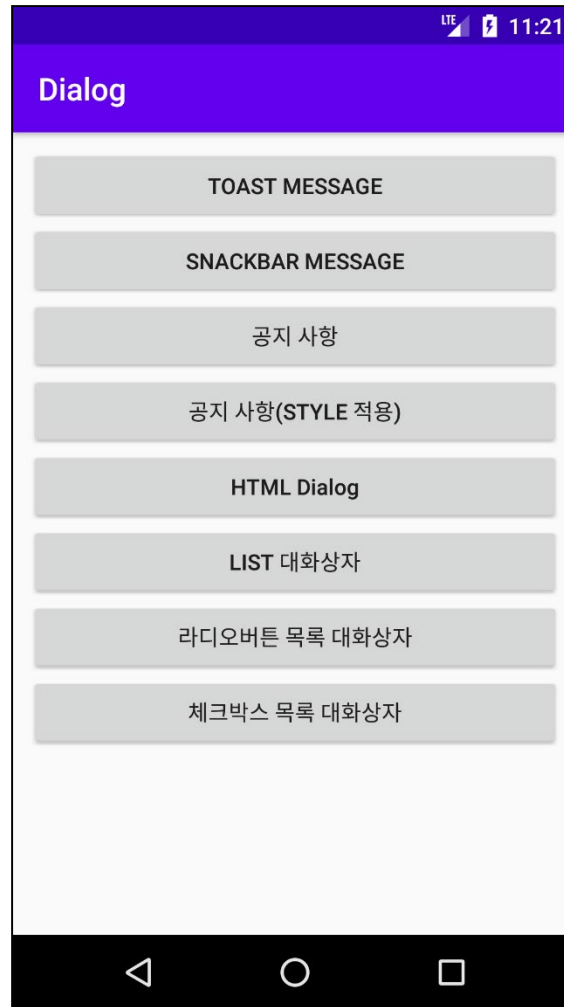
그림 7-10 대화상자 설정 순서





# Dialog 예제 1

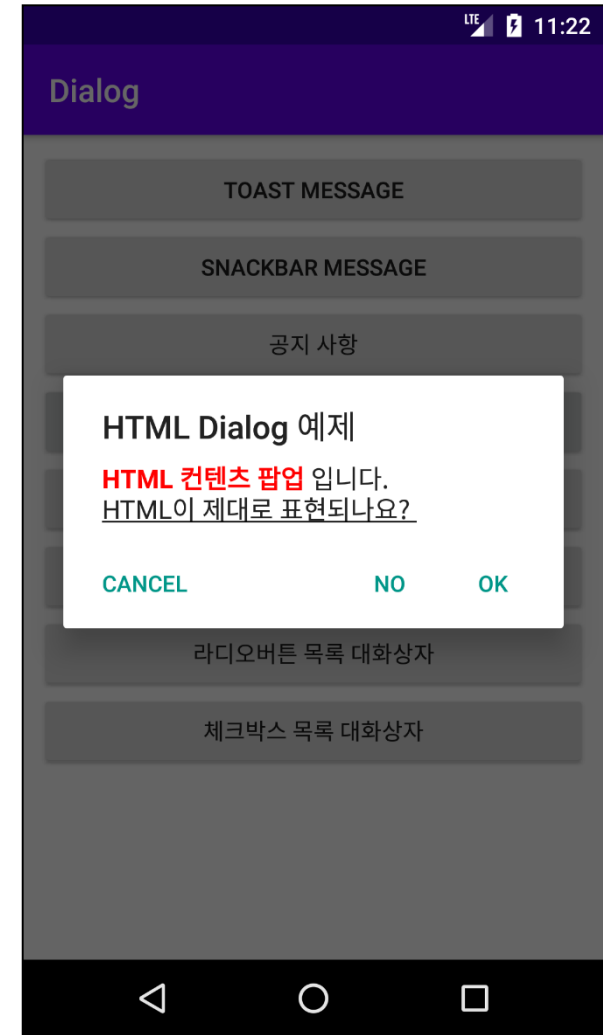
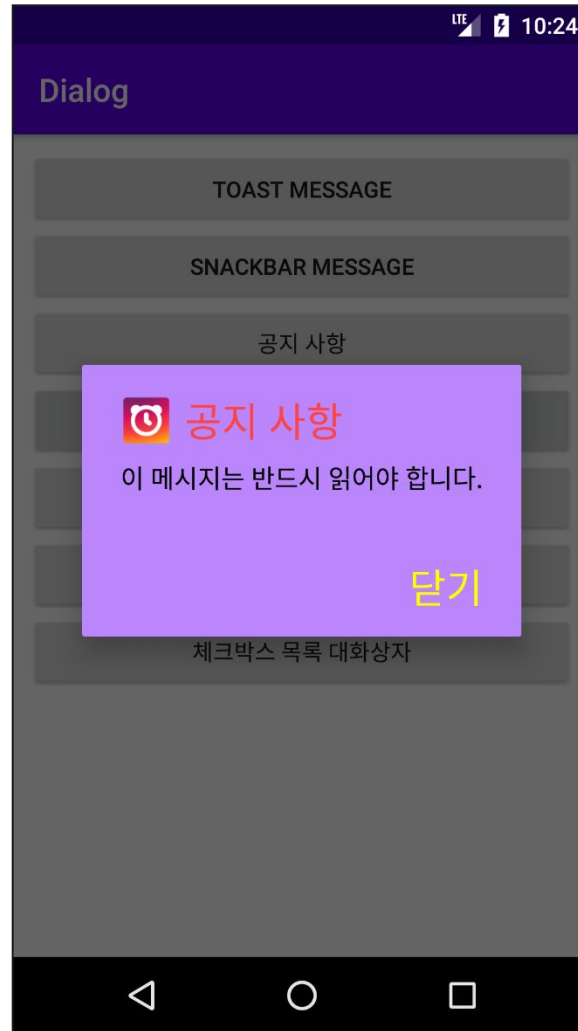
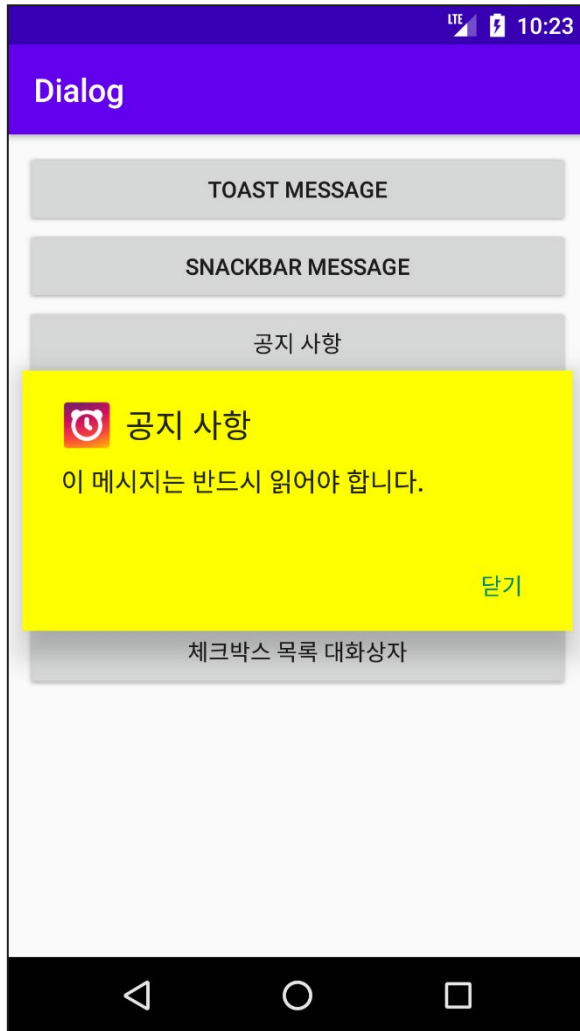
- 다음과 같은 각종 Dialog를 만들어 보자





# Dialog 예제 1

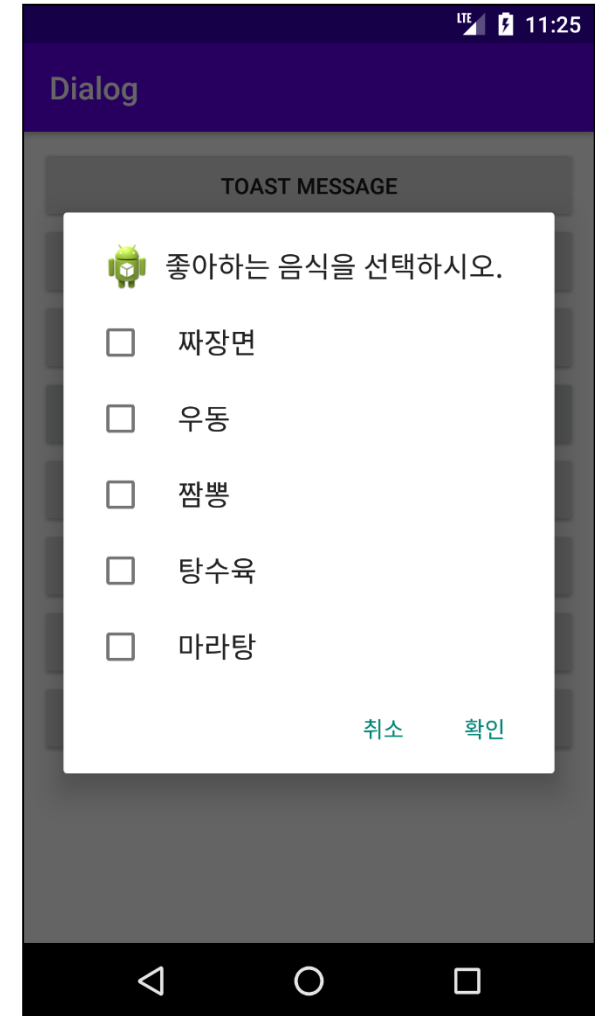
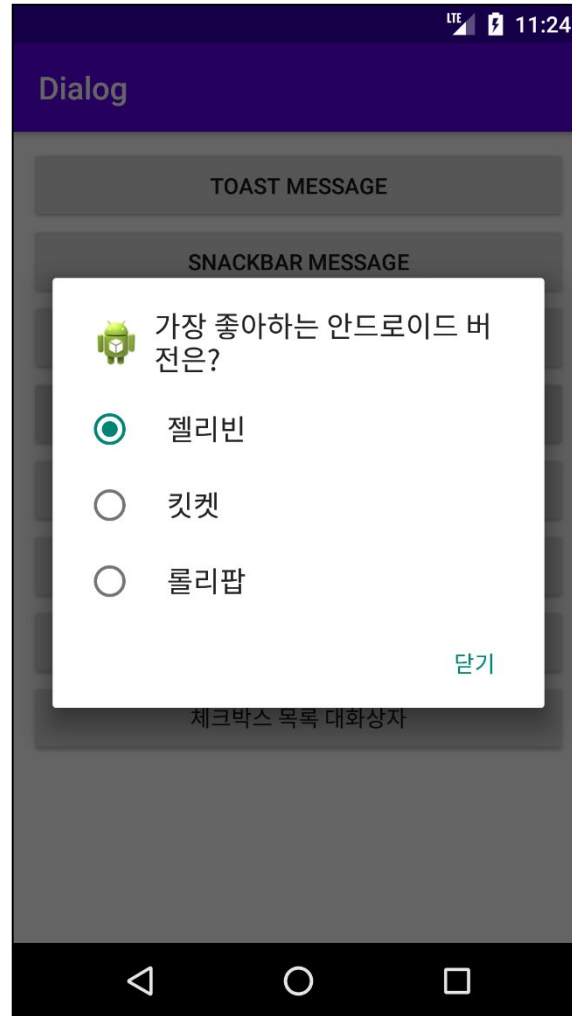
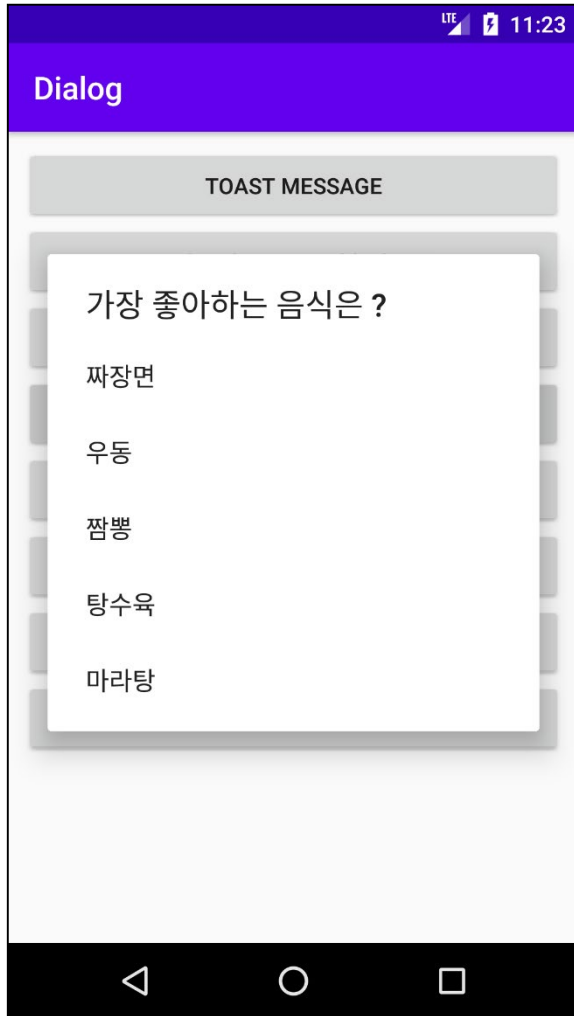
## ■ 실행 화면





# Dialog 예제 1

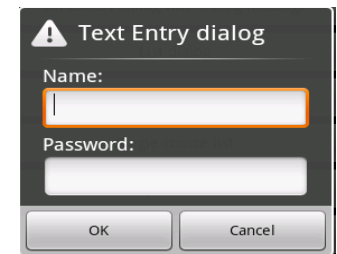
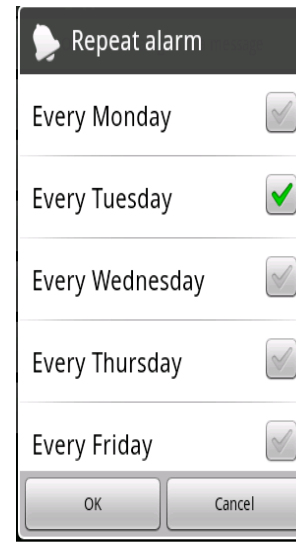
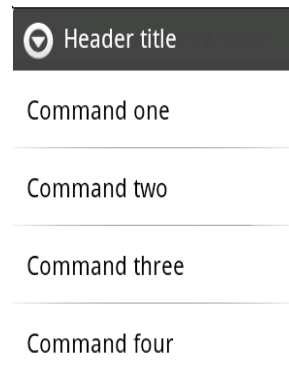
## ■ 실행 화면





# Dialog 예제 1

- 화면의 일부 영역을 할당 받아 사용자의 제어권을 확보하고, 정보의 알림과 사용자의 특정 행동을 유도할 수 있도록 Button을 제공해 주어 사용자의 선택과 정보 유도를 용이하게 해주는 역할을 수행하는 Widget
- Dialog의 내용에는 정보안내, RadioButton, CheckBox, List Box, Progress 등 다양한 Widget을 포함





# Dialog 예제 1

## ■ 제목과 설명 넣기

```
AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity3.this);  
builder.setTitle("공지 사항");  
builder.setMessage("이 메시지는 반드시 읽어야 합니다.")  
    .setIcon(R.drawable.alarm)  
    .setCancelable(false)  
    .setNegativeButton("닫기", null)  
    .show();
```

```
AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity3.this);  
builder.setTitle("공지 사항");  
builder.setMessage("이 메시지는 반드시 읽어야 합니다.")  
    .setIcon(R.drawable.alarm)  
    .setCancelable(false)  
    .setNegativeButton("닫기", null);  
Dialog dialog = builder.create();  
dialog.show();
```



# Dialog 예제 1



- 배경색 지정하기

- Dialog를 생성 후에

- ```
dialog.getWindow().setBackgroundDrawable(new  
ColorDrawable(Color.YELLOW));
```

를 이용하여 배경색 지정

- Activity 화면 어둡게 변하는 것 방지하는 방법

- Dialog를 생성 후 띄우면 Dialog를 제외한 나머지 배경은 어둡게 변함

- 방지하기 위해서는

- ```
dialog.getWindow().setDimAmount(0); 나  
dialog.getWindow().clearFlags(WindowManager.Layou  
tParams.FLAG_DIM_BEHIND);
```

를 이용



# Dialog 예제 1

## ■ Style 적용하기

- Android AlertDialog를 styles.xml로 다양하게 변화시킬 수 있음

```
<style name="MyDialogStyle" parent="@style/Theme.AppCompat.Light.Dialog">  
  <item name="android:textColor">@color/yellow</item>  
  <item name="android:textSize">25dp</item>  
  <item name="android:textColorPrimary">@color/black</item>  
  <item name="android:typeface">sans</item>  
  <item name="android:background">@color/purple_200</item>  
  <item name="android:windowTitleStyle">@style/DialogTitle</item>  
</style>
```

```
<style name="DialogTitle"  
  parent="RtlOverlay.DialogWindowTitle.AppCompat">  
  <item name="android:textColor">@android:color/holo_red_light</item>  
</style>
```



# Dialog 예제 1

- Html tag를 사용하는 방법
  - Html.fromHtml(String, int) 메소드를 사용하지만, API가 24이상 (Android 7.0 Nougat)에서 androidx HtmlCompat의 fromHtml(String, int) 메소드를 사용

```
HtmlCompat.fromHtml(  
    getString(R.string.warning_message),  
    HtmlCompat.FROM_HTML_MODE_LEGACY);
```





# Dialog 예제 1



- TextView Widget이 지원하는 HTML tag
  - <a href="...">
  - <b>
  - <big>
  - <blockquote>
  - <br>
  - <cite>
  - <dfn>
  - <div align="...">
  - <em>
  - <font size="..." color="..." face="...">
  - <h1>
  - <h2>
  - <h3>
  - <h4>



# Dialog 예제 1

## ■ TextView Widget이 지원하는 HTML tag

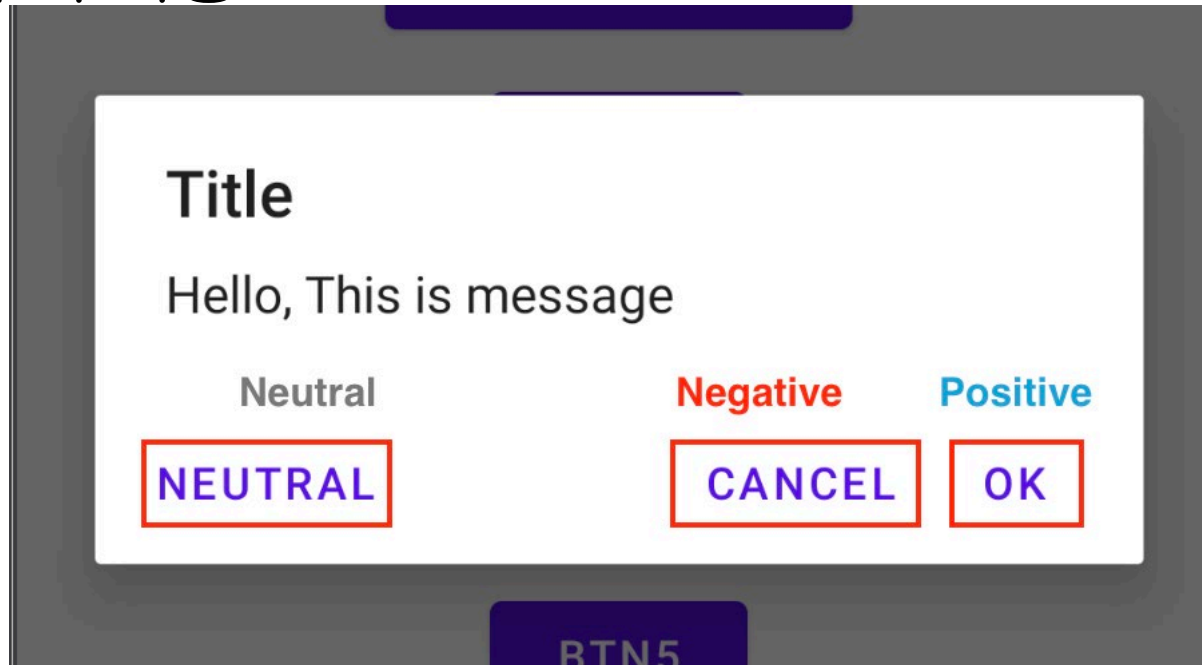
- <h5>
- <h6>
- <i>
- 
- <p>
- <small>
- <strike>
- <strong>
- <sub>
- <sup>
- <tt>
- <u>



# Dialog 예제 1

## ■ Button 추가하기

- 공간상의 제약으로 인해 Dialog에는 Button을 최대 3개까지만 배치할 수 있음
- 사용자로부터 응답은 Positive, Negative, Neutral 3가지를 추가 가능





# Dialog 예제 1

## ■ Button 추가하기

```
builder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {  
    public void onClick(DialogInterface dialog, int which) {  
        finish();  
    }  
});
```

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

```
builder.setNeutralButton("Cancel", new DialogInterface.OnClickListener() {  
    public void onClick(DialogInterface dialog, int which) {  
        dialog.cancel();  
    }  
});
```



# Dialog 예제 1

## ■ List(목록) 보여주기

- 더 많은 선택 사항 중 하나를 입력 받으려면 목록 대화상자를 사용해야 함

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

- 첫 번째 인수가 사용자에게 보여줄 목록이며 문자열 배열을 전달할 수도 있고 배열 Resource의 ID를 지정
- 두 번째 인수는 사용자가 목록의 항목 중 하나를 Click할 때의 동작을 지정하는 Click Listener이며 Click시 다음 메소드를 호출

```
void onClick(DialogInterface dialog, int which)
```



# Dialog 예제 1

## List(목록) 보여주기

```
AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity3.this);
builder.setTitle("가장 좋아하는 음식은 ?");
builder.setItems(R.array.foods, new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int pos) {
        String selected = getResources().getStringArray(R.array.foods)[pos];
        Toast.makeText(MainActivity3.this, selected,
            Toast.LENGTH_SHORT).show();
    }
});

AlertDialog dialog = builder.create();
dialog.getWindow().clearFlags(
    WindowManager.LayoutParams.FLAG_DIM_BEHIND);
dialog.show();
```

List 형태의 AlertDialog를 만들 때는 Message를 작성하면 안됨

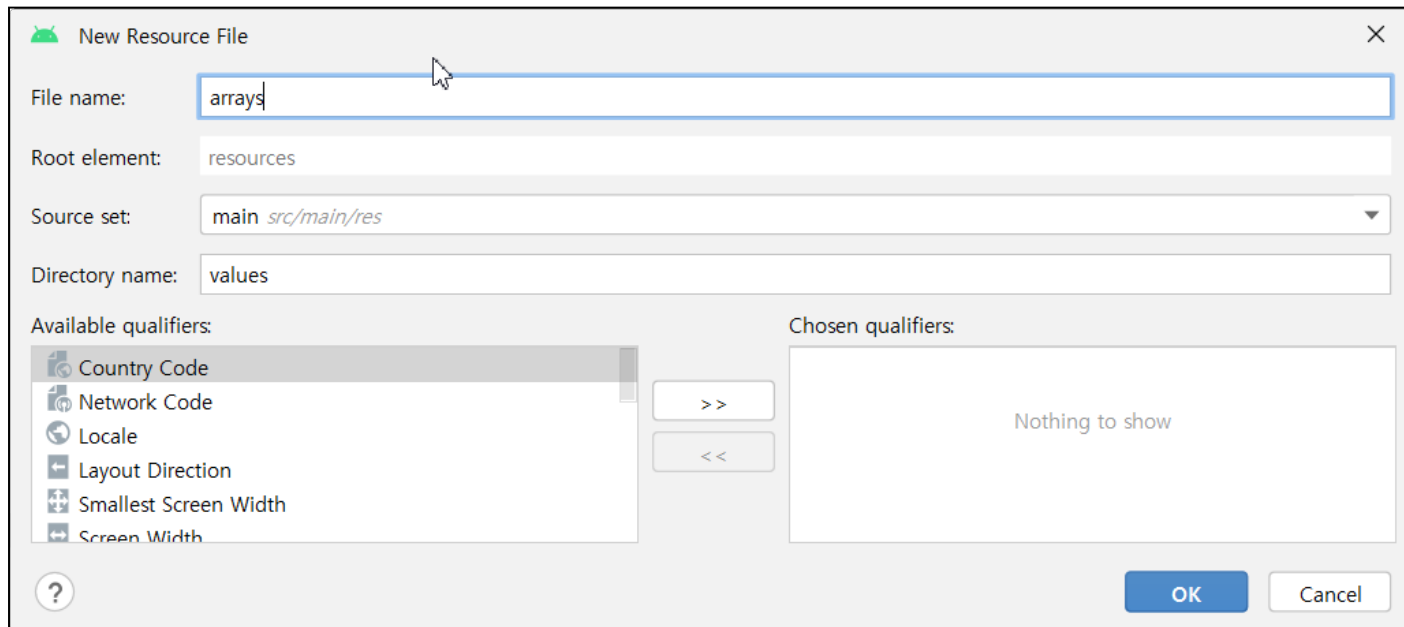


# Dialog 예제 1

- List(목록) Data 저장 방법

- Array resource

- Application내에 고정으로 사용해야 하는 Data들은 XML File을 만들고, 그 안에 <string-array>를 사용하면 Data들을 한곳에 모아서 간단히 처리할 수 있음





# Dialog 예제 1

- List(목록) Data 저장 방법

- String

- String은 하나의 클래스.
    - JAVA Program의 모든 문자열 literal은 이 클래스의 Instance로 구현
    - 중요한 점은 문자열 값은 작성된 후에 변경할 수 없음
    - String 객체에 보관하는 문자열은 Unicode로 변형되므로 HTML과 같은 Markup 문자를 입출력할 때 문제가 발생
    - 이와 같이 Markup 문자를 입력하여 사용할 수 없기 때문에 변경할 수 없는 문자열이라고 부름
    - 변경할 수 있는 문자열을 지원하는 StringBuilder와 StringBuffer가 있음





# Dialog 예제 1



- List(목록) Data 저장 방법

- CharSequence

- CharSequence는 클래스가 아니라 Interface
    - Interface명(character + sequence)에서 짐작되듯이 char 값을 읽을 수 있는 sequence
    - 이 Interface는 다양한 종류의 char Sequence에 대해 균일한 읽기 전용 접근 권한을 제공
    - CharSequence를 implements하여 구현된 대표적인 클래스로는 String, SpannableStringBuilder, StringBuilder, StringBuffer 등이 있음
    - CharSequence 객체에 보관하는 문자열은 같은 Unicode라도 Markup 문자를 사용할 수 있음
    - String 클래스와 반대로 변형과 가공을 할 수 있기 때문에 Style 문자 또는 Sequence 문자라고 함
    - XML에서 사용하는 Style



# Dialog 예제 1

## ■ RadioButton 목록 보여주기

- List의 item중에서 1개를 선택할 수 있도록 RadioButton을 Text 옆에 보여 줌
- 이렇게 생성한 List의 item들은 오직 1개만 선택할 수 있으며, 사용자가 item을 선택하더라도 Dialog는 안 닫힘
- RadioButton을 위해서는 setSingleChoiceItems()를 사용

```
setSingleChoiceItems(CharSequence[] items,  
                    int checkedItem,  
                    DialogInterface.OnClickListener listener)
```

- 첫 번째 인자는 item 배열
- 두 번째 인자는 RadioButton의 Check 위치
  - -1은 선택 안함
  - 0은 첫번째 item 선택
- 세 번째 인자는 선택 시 발생하는 Event Listener



# Dialog 예제 1

## ■ CheckBox 표시하기

- 여러 item중에서 item을 원하는 만큼 중복 선택할 수 있도록 CheckBox를 Text 옆에 보여 줌
- CheckBox를 위해서는 setMultiChoiceItems()를 사용

```
setMultiChoiceItems(CharSequence[] items,  
                    boolean[] checkedItems,  
                    DialogInterface.OnClickListener listener)
```

- 첫 번째 인자는 item 배열
- 두 번째 인자는 CheckBox의 Check 위치
  - boolean 배열 초기화 가능
- 세 번째 인자는 선택 시 발생하는 Event Listener



# Dialog 예제 1

## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:layout_margin="10dp"
    tools:context=".MainActivity">

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Toast Message"/>

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="SnackBar Message"/>
```



# Dialog 예제 1

## ■ 사용자 인터페이스

```
<Button  
    android:id="@+id/button3"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="공지 사항"/>
```

```
<Button  
    android:id="@+id/button4"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="공지 사항(Style 적용)"/>
```

```
<Button  
    android:id="@+id/button5"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="HTML Dialog"  
    android:textAllCaps="false" />
```



# Dialog 예제 1

## ■ 사용자 인터페이스

```
<Button
    android:id="@+id/button6"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="List 대화상자"/>

<Button
    android:id="@+id/button7"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="라디오버튼 목록 대화상자"/>

<Button
    android:id="@+id/button8"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="체크박스 목록 대화상자"/>
</LinearLayout>
```



# Dialog 예제 1

## ■ arrays.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string-array name="foods">
    <item>짜장면</item>
    <item>우동</item>
    <item>짬뽕</item>
    <item>탕수육</item>
    <item>마라탕</item>
  </string-array>
</resources>
```



# Dialog 예제 1

## ■ JAVA 프로그램

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main3);  
  
    Button button1 = findViewById(R.id.button1);  
    button1.setOnClickListener(new Button.OnClickListener() {  
        public void onClick(View v) {  
            Toast.makeText(getBaseContext(), "Toast Message 입니다",  
                           Toast.LENGTH_SHORT).show();  
        }  
    });  
    Button button2 = findViewById(R.id.button2);  
    button2.setOnClickListener(new Button.OnClickListener() {  
        public void onClick(View v) {  
            Snackbar.make(v, "SnackBar Message 입니다",  
                           Snackbar.LENGTH_SHORT).show();  
        }  
    });  
}
```





# Dialog 예제 1

## ■ JAVA 프로그램

```
Button button3 = findViewById(R.id.button3);
button3.setOnClickListener(new Button.OnClickListener() {
    public void onClick(View v) {
        AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity3.this);
        builder.setTitle("공지 사항");
        builder.setMessage("이 메시지는 반드시 읽어야 합니다.")
            .setIcon(R.drawable.alarm)
            .setCancelable(false)
            .setNegativeButton("닫기", null);
        Dialog dialog = builder.create();
        dialog.getWindow().setBackgroundDrawable(
            new ColorDrawable(Color.YELLOW));
        dialog.getWindow().setDimAmount(0);
        dialog.show();
    }
});
```



# Dialog 예제 1

## ■ JAVA 프로그램

```
Button button4 = findViewById(R.id.button4);
button4.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        new AlertDialog.Builder(MainActivity3.this, R.style.MyDialogStyle)
            .setTitle("공지 사항")
            .setMessage("이 메시지는 반드시 읽어야 합니다.")
            .setIcon(R.drawable.alarm)
            .setCancelable(false)
            .setNegativeButton("닫기", null)
            .show();
    }
});
```



# Dialog 예제 1



```
Button button5 = findViewById(R.id.button5);
button5.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity3.this,
            android.R.style.Theme_DeviceDefault_Light_Dialog_Alert);
        String strHtml = "<b><font color='#ff0000'>
            HTML 콘텐츠 팝업</font></b> 입니다." +
            "<br> <u>HTML이 제대로 표현되나요? </u></br> ";
        Spanned spanned = HtmlCompat.fromHtml(strHtml,
            HtmlCompat.FROM_HTML_MODE_LEGACY);
        builder.setTitle("HTML Dialog 예제")
            .setMessage(spanned)
            .setPositiveButton("OK", null)
            .setNeutralButton("Cancel", null)
            .setNegativeButton("No", null)
            .setCancelable(false)
            .show();
    }
});
```



# Dialog 예제 1

## ■ JAVA 프로그램

```
Button button6 = findViewById(R.id.button6);
button6.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity3.this);
        builder.setTitle("가장 좋아하는 음식은 ?");
        builder.setItems(R.array.foods, new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int pos) {
                String selected = getResources().getStringArray(R.array.foods)[pos];
                Toast.makeText(MainActivity3.this, selected,
                    Toast.LENGTH_SHORT).show();
            }
        });
        AlertDialog dialog = builder.create();
        dialog.getWindow().clearFlags(
            WindowManager.LayoutParams.FLAG_DIM_BEHIND);
        dialog.show();
    }
});
```



# Dialog 예제 1

## ■ JAVA 프로그램

```
Button button7 = findViewById(R.id.button7);
button7.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String[] version = new String[]{"젤리빈", "킷켓", "롤리팝"};
        AlertDialog.Builder builder = new AlertDialog.Builder(MainActivity3.this);
        builder.setTitle("가장 좋아하는 안드로이드 버전은?");
        builder.setIcon(R.drawable.ic_launcher);
        builder.setSingleChoiceItems(version, 0,
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int which) {
                    Toast.makeText(getApplicationContext(), "좋아하는 버전 : "
                        + version[which], Toast.LENGTH_LONG).show();
                }
            });
        builder.setPositiveButton("닫기", null);
        builder.show();
    }
});
```



# Dialog 예제 1

## ■ JAVA 프로그램

```
Button button8 = findViewById(R.id.button8);
button8.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        final boolean[] select = {false, false, false, false, false};
        new AlertDialog.Builder(MainActivity3.this)
            .setTitle("좋아하는 음식을 선택하십시오.")
            .setIcon(R.drawable.ic_launcher)
            .setMultiChoiceItems(R.array.foods, select,
                new DialogInterface.OnMultiChoiceClickListener() {
                    public void onClick(DialogInterface dialog, int which,
                        boolean isChecked) {
                        select[which] = isChecked;
                    }
                })
    }
})
```



# Dialog 예제 1

```
.setPositiveButton("확인", new DialogInterface.OnClickListener() {
    public void onClick(DialogInterface dialog, int whichButton) {
        String[] foods = getResources().getStringArray(R.array.foods);
        String result = "";
        for (int i = 0; i < select.length; i++) {
            if (select[i]) {
                result += foods[i] + " ";
            }
        }
        result = result.trim();
        result = result.replaceAll(" ", ", ");
        if (result.equals(""))
            result = "없음";
        Toast.makeText(getApplicationContext(), "선택한 음식 = " + result,
            Toast.LENGTH_LONG).show();
    }
});
.setNegativeButton("취소", null)
.show();
}
});
}
```



# Dialog 예제 1

## ■ styles.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="MyDialogStyle" parent="@style/Theme.AppCompat.Light.Dialog">
        <item name="android:textColor">@color/yellow</item>
        <item name="android:textSize">25dp</item>
        <item name="android:textColorPrimary">@color/black</item>
        <item name="android:typeface">sans</item>
        <item name="android:background">@color/purple_200</item>
        <item name="android:windowTitleStyle">@style/DialogTitle</item>
    </style>

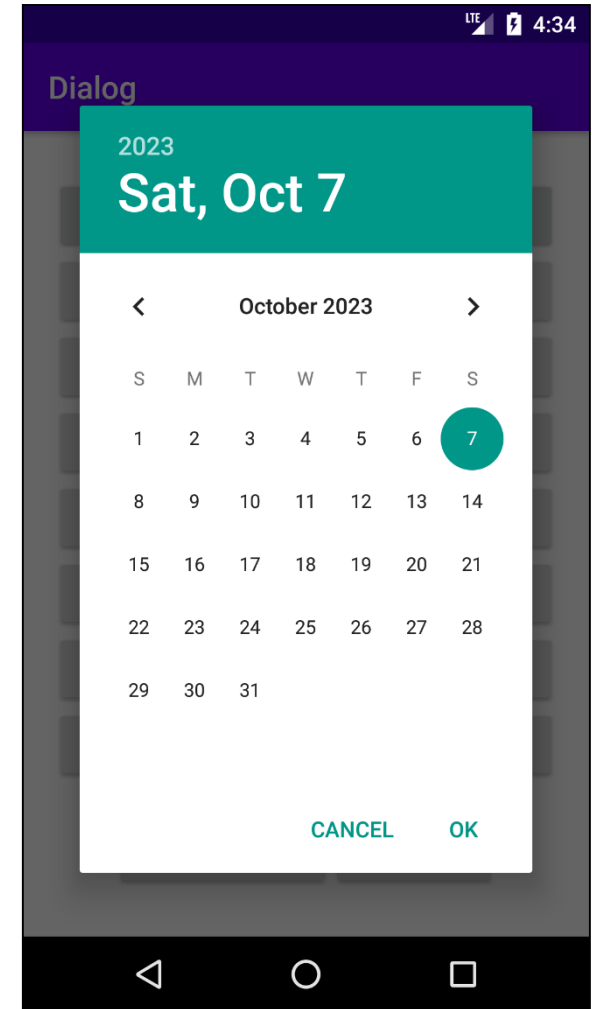
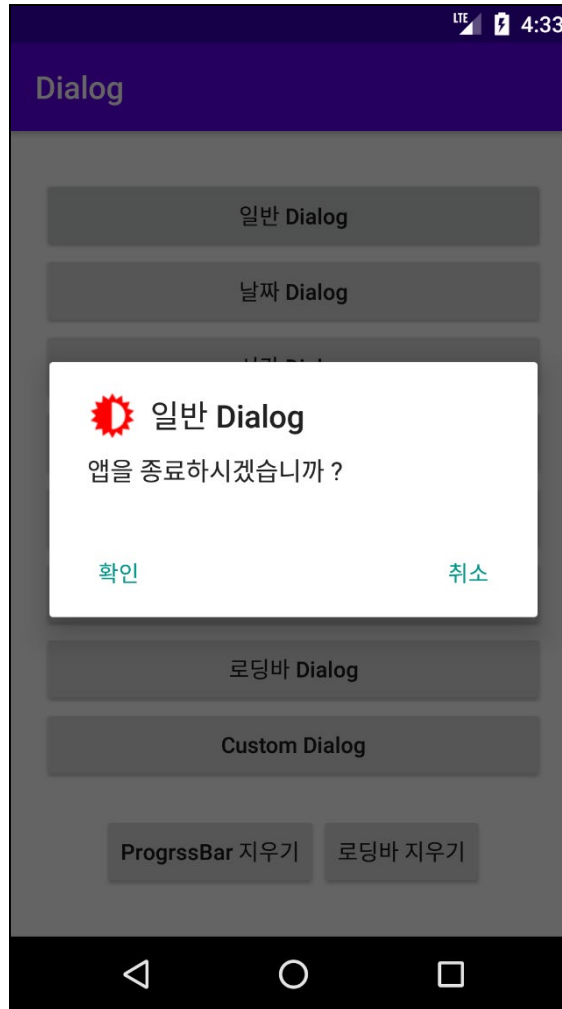
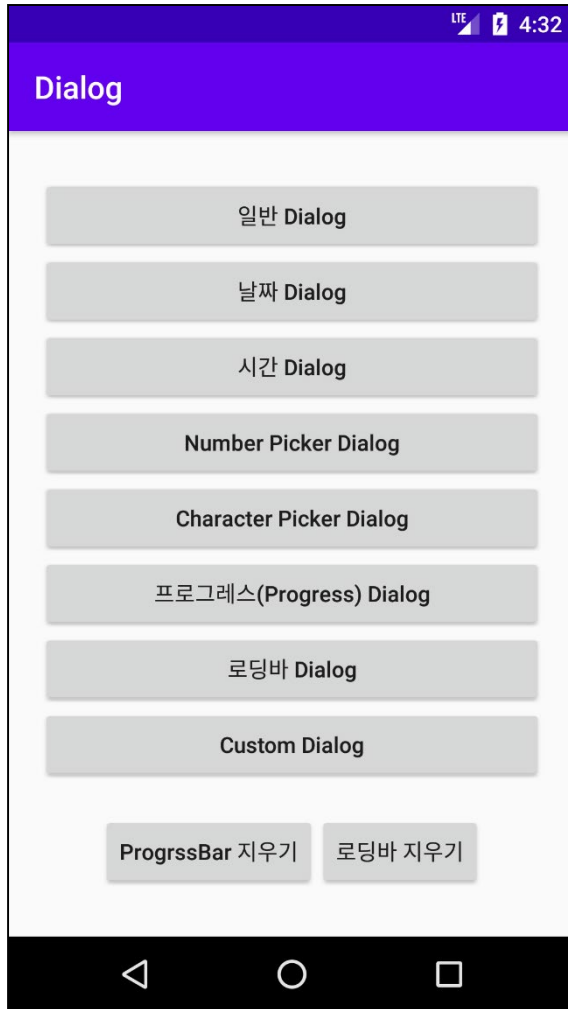
    <style name="DialogTitle"
        parent="RtlOverlay.DialogWindowTitle.AppCompat">
        <item name="android:textColor">@android:color/holo_red_light</item>
    </style>
</resources>
```





# Dialog 예제 2

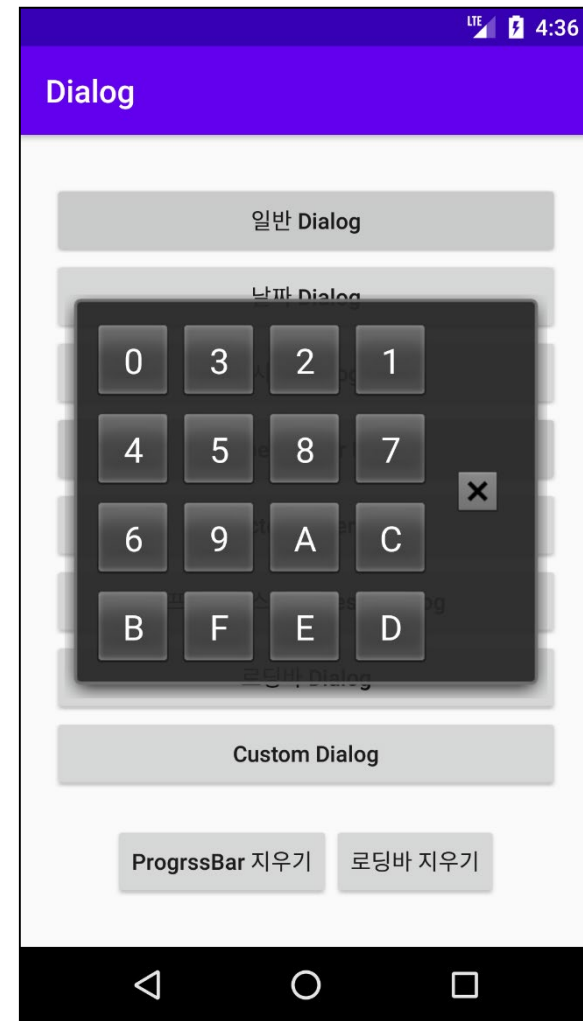
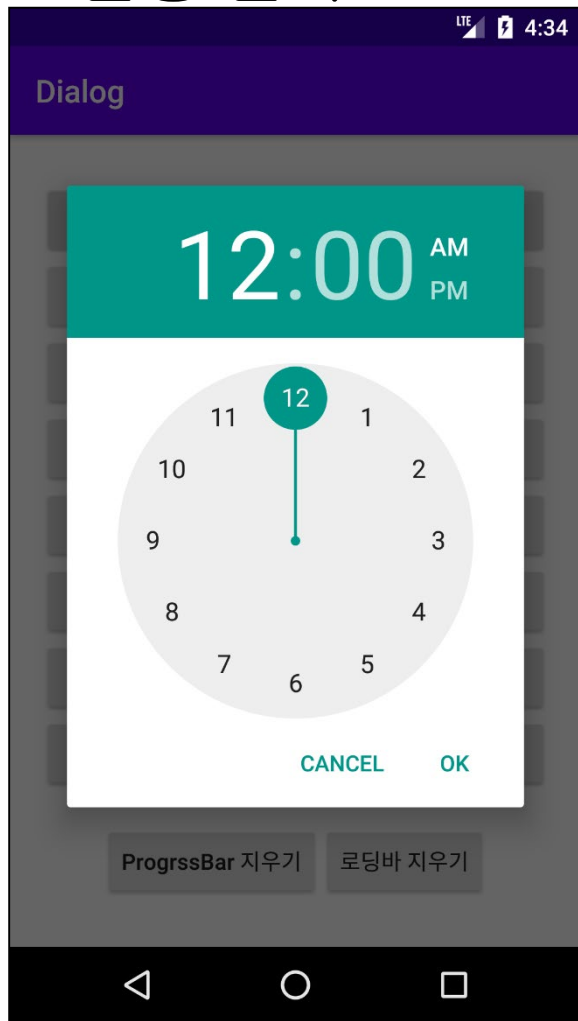
■ 다음과 같은 Dialog를 만들어보자





# Dialog 예제 2

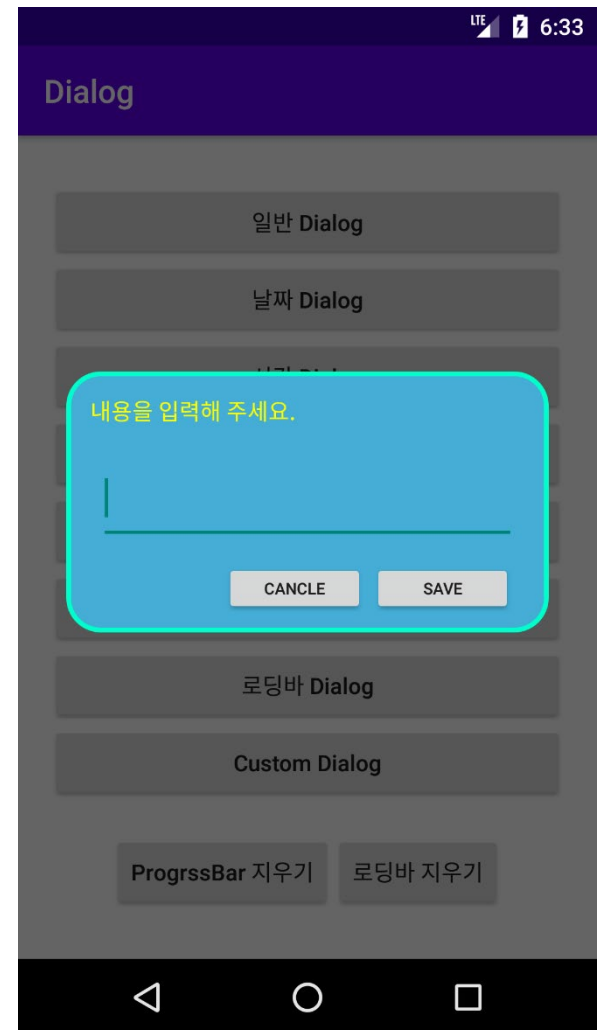
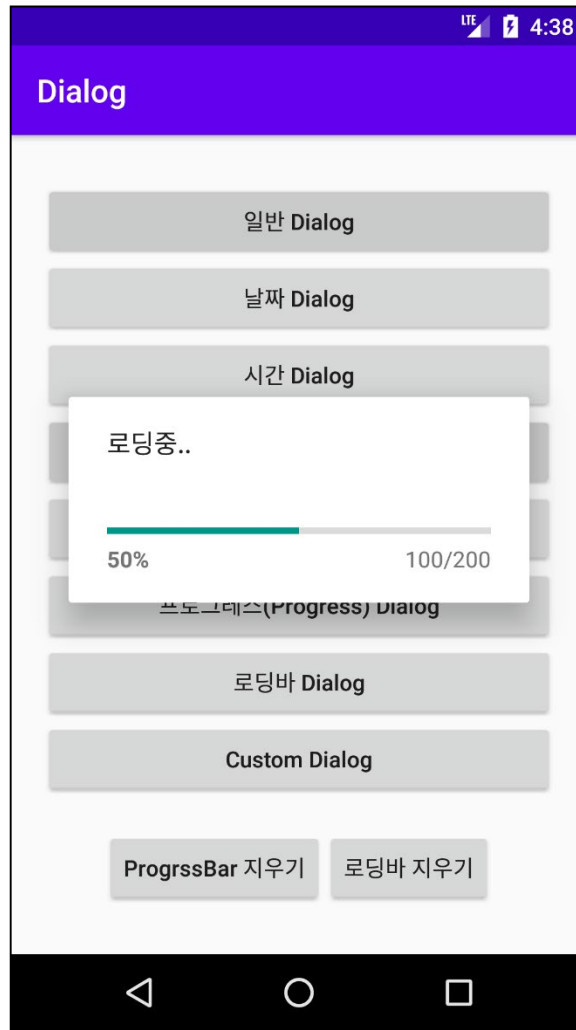
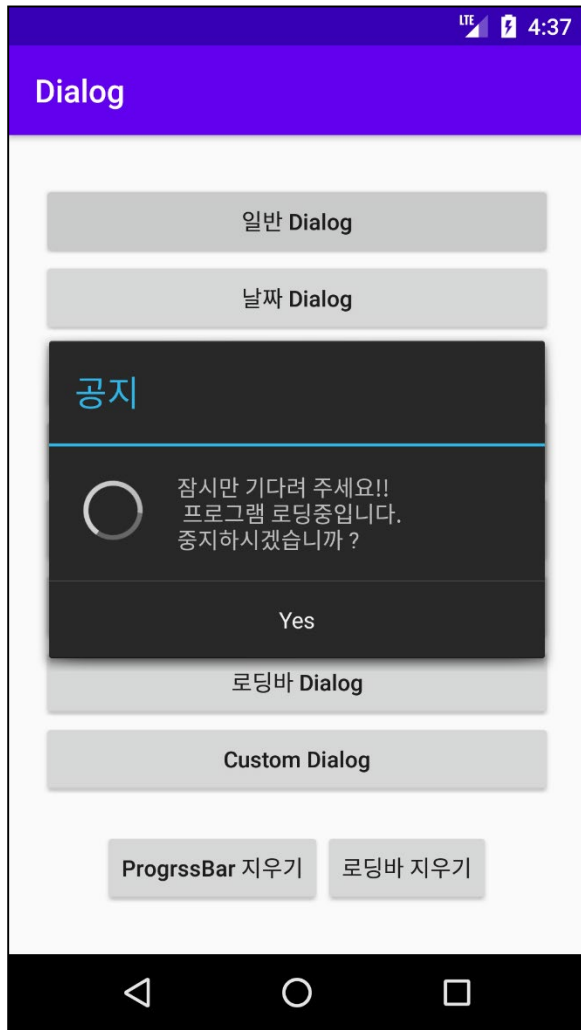
## ■ 실행 결과





# Dialog 예제 2

## ■ 실행 결과





# Dialog 예제 2

## ■ 사용자 인터페이스

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    android:padding="20dp"
    tools:context=".MainActivity">

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:onClick="showDialog1"
        android:text="일반 Dialog"
        android:textAllCaps="false" />
```



# Dialog 예제 2

## ■ 사용자 인터페이스

```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="showDialog2"
    android:text="날짜 Dialog"
    android:textAllCaps="false" />

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="showDialog3"
    android:text="시간 Dialog"
    android:textAllCaps="false" />

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="showDialog4"
    android:text="Number Picker Dialog"
    android:textAllCaps="false" />
```



# Dialog 예제 2

## ■ 사용자 인터페이스

<Button

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:onClick="showDialog5"  
    android:text="Character Picker Dialog"  
    android:textAllCaps="false" />
```

<Button

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:onClick="showDialog6"  
    android:text="프로그레스(Progress) Dialog"  
    android:textAllCaps="false" />
```

<Button

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:onClick="showDialog7"  
    android:text="로딩바 Dialog"  
    android:textAllCaps="false" />
```



# Dialog 예제 2

## ■ 사용자 인터페이스

<Button

```
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:onClick="showDialog8"  
    android:text="Custom Dialog"  
    android:textAllCaps="false" />
```



# Dialog 예제 2

## ■ 사용자 인터페이스

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:gravity="center_horizontal">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="dismissDialog1"
        android:text="ProgrssBar 지우기"
        android:textAllCaps="false" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="dismissDialog2"
        android:text="로딩바 지우기" />
</LinearLayout>
</LinearLayout>
```





# Dialog 예제 2

## ■ dialog\_birthday.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:background="@color/yellow"
    android:baselineAligned="false"
    android:orientation="vertical">

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:layout_marginBottom="10dp"
        android:gravity="center"
        android:minHeight="42dp"
        android:textSize="20dp"
        android:text="출생년도를 선택해주세요" />
```



# Dialog 예제 2

## ■ dialog\_birthday.xml

```
<NumberPicker
    android:id="@+id/birthdayPicker"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_gravity="center">

    <Button
        android:id="@+id/cancel"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:gravity="center"
        android:text="취소" />
```



# Dialog 예제 2

## ■ dialog\_birthday.xml

```
<Button
    android:id="@+id/ok"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:gravity="center"
    android:text="확인" />
</LinearLayout>
</LinearLayout>
```



# Dialog 예제 2

## ■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {  
    ProgressDialog progress, progress1;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
}
```



# Dialog 예제 2

## ■ MainActivity.JAVA

```
public void showDialog1(View view) {
    AlertDialog.Builder builder = new AlertDialog.Builder(this,
        android.R.style.Theme_DeviceDefault_Light_Dialog_Alert);
    builder.setMessage("앱을 종료하시겠습니까 ?");
    builder.setIcon(R.drawable.star1).setTitle("일반 Dialog");
    builder.setPositiveButton("취소", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            Toast.makeText(getBaseContext(), "취소를 눌렀습니다",
                Toast.LENGTH_LONG).show();
        }
    }).setNegativeButton("확인", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
            Toast.makeText(getBaseContext(), "확인하였습니다",
                Toast.LENGTH_SHORT).show();
        }
    }).setCancelable(false) // 백버튼으로 팝업창이 닫히지 않도록 한다.
    .show();
}
```



# Dialog 예제 2

```
public void showDialog2(View view) {  
    Calendar calendar = Calendar.getInstance();  
    int year = calendar.get(Calendar.YEAR);  
    int month = calendar.get(Calendar.MONTH);  
    int day = calendar.get(Calendar.DAY_OF_MONTH);  
    DatePickerDialog.OnDateSetListener dateSetListener =  
        new DatePickerDialog.OnDateSetListener() {  
        public void onDateSet(DatePicker view, int year, int monthOfYear,  
                                int dayOfMonth) {  
            String temp = year + "년 ";  
            temp += (monthOfYear + 1) + "월 ";  
            temp += dayOfMonth + "일";  
            Toast.makeText(getApplicationContext(), temp, Toast.LENGTH_SHORT).show();  
        }  
    };  
    DatePickerDialog dialog = new DatePickerDialog(this,  
        android.R.style.Theme_DeviceDefault_Light_Dialog,  
        dateSetListener, year, month, day);  
    dialog.show();  
}
```



# Dialog 예제 2

## ■ MainActivity.JAVA

```
public void showDialog3(View view) {  
    TimePickerDialog.OnTimeSetListener timeSetListener =  
        new TimePickerDialog.OnTimeSetListener() {  
        public void onTimeSet(TimePicker view, int hourOfDay, int minute) {  
            Toast.makeText(getApplicationContext(), hourOfDay + ":" + minute,  
                Toast.LENGTH_SHORT).show();  
        }  
    };  
    TimePickerDialog dialog = new TimePickerDialog(this,  
        android.R.style.Theme_DeviceDefault_Light_Dialog,  
        timeSetListener, 0, 0, false);  
    dialog.show();  
}
```



# Dialog 예제 2

## ■ MainActivity.JAVA

```
public void showDialog4(View view) {  
    Calendar calender = Calendar.getInstance();  
    int year = calender.get(Calendar.YEAR);  
  
    Dialog dialog = new Dialog(this);  
    dialog.requestWindowFeature(Window.FEATURE_NO_TITLE);  
    dialog setContentView(R.layout.dialog_birthday);  
    Button ok = dialog.findViewById(R.id.ok);  
    Button cancel = dialog.findViewById(R.id.cancel);  
    NumberPicker picker = dialog.findViewById(R.id.birthdayPicker);  
    picker.setMinValue(year - 80);  
    picker.setMaxValue(year - 15);  
    picker.setDescendantFocusability(  
        NumberPicker.FOCUS_BLOCK_DESCENDANTS);  
    picker.setWrapSelectorWheel(false);  
    picker.setValue(year - 20);  
}
```





# Dialog 예제 2



```
picker.setOnValueChangedListener(  
    new NumberPicker.OnValueChangeListener() {  
        @Override  
        public void onValueChange(NumberPicker picker, int oldVal, int newVal) {  
        }  
    });  
ok.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        Toast.makeText(MainActivity2.this, String.valueOf(picker.getValue()),  
            Toast.LENGTH_SHORT).show();  
        dialog.dismiss();  
    }  
});  
cancel.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View v) {  
        dialog.dismiss();  
    }  
});  
dialog.show();  
}
```



# Dialog 예제 2

## ■ MainActivity.JAVA

```
public void showDialog5(View view) {  
    String options = "0321458769ACBFED";  
    CharacterPickerDialog dialog =  
        new CharacterPickerDialog(this, new View(this),  
                                null, options, false) {  
        public void onClick(View v) {  
            Toast.makeText(getApplicationContext(), "selected character :" +  
                ((Button) v).getText().toString(), Toast.LENGTH_SHORT).show();  
            dismiss();  
        }  
    };  
    dialog.show();  
}
```



# Dialog 예제 2

## ■ MainActivity.JAVA

```
public void showDialog6(View view) {  
    progress = new ProgressDialog(this, AlertDialog.THEME_HOLO_DARK);  
    progress.setTitle("공지");  
    progress.setMessage("잠시만 기다려 주세요!!\n 프로그램 로딩중입니다.  
                \n 중지하시겠습니까 ?");  
  
    progress.setButton("Yes", new DialogInterface.OnClickListener() {  
        @Override  
        public void onClick(DialogInterface dialogInterface, int i) {  
            progress.dismiss();  
            Toast.makeText(getApplicationContext(), "중지하였습니다",  
                Toast.LENGTH_SHORT).show();  
        }  
    });  
    progress.getWindow().clearFlags(  
        WindowManager.LayoutParams.FLAG_DIM_BEHIND);  
    progress.show();  
}
```



# Dialog 예제 2

## ■ MainActivity.JAVA

```
public void showDialog7(View view) {  
    progress1 = new ProgressDialog(this,  
        android.R.style.Theme_DeviceDefault_Light_Dialog);  
    progress1.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);  
    progress1.setMessage("로딩중..");  
    progress1.setMax(200);  
    progress1.getWindow().setDimAmount(0);  
    progress1.show();  
    for (int i = 1; i <= 10; i++) {  
        progress1.setProgress(i * 10);  
    }  
}  
  
public void showDialog8(View view) {  
    CustomDialog dialog = new CustomDialog(this);  
    dialog.setCancelable(false);  
    dialog.show();  
}
```



# Dialog 예제 2

## ■ MainActivity.JAVA

```
public void dismissDialog1(View view) {  
    if (progress != null) progress.dismiss();  
}  
  
public void dismissDialog2(View view) {  
    if (progress1 != null) progress1.dismiss();  
}  
}
```



# Dialog 예제 2

## ■ dialog\_round.xml

```
<?xml version="1.0" encoding="utf-8"?>
<layer-list xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content">
    <item>
        <shape android:shape="rectangle">
            <corners android:radius="20dp" />
            <solid android:color="#45ADD6" />
            <stroke android:width="3dp" android:color="#00FFCC"/>
        </shape>
    </item>
</layer-list>
```



# Dialog 예제 2

## ■ custom\_dialog.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:layout_gravity="center_horizontal"
    android:background="@drawable/dialog_round"
    android:orientation="vertical"
    android:padding="10dp">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="0dp"
        android:layout_margin="5dp"
        android:layout_weight="0.6"
        android:text="내용을 입력해 주세요."
        android:textColor="#FF0" />
```



# Dialog 예제 2

## ■ custom\_dialog.xml

```
<EditText
    android:id="@+id/put_text"
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:layout_margin="10dp"
    android:layout_weight="1.7" />

<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="0dp"
    android:layout_gravity="right"
    android:layout_marginEnd="12dp"
    android:layout_weight="1">
```





# Dialog 예제 2

## ■ custom\_dialog.xml

```
<Button
    android:id="@+id/btnCancle"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:scaleType="center"
    android:text="cancel"
    android:textSize="10dp" />
```

```
<Button
    android:id="@+id/btnSave"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_marginStart="4dp"
    android:scaleType="center"
    android:text="save"
    android:textSize="10dp" />
```

```
</LinearLayout>
```

```
</LinearLayout>
```



# Dialog 예제 2



## ■ CustomDialog.JAVA

```
class CustomDialog extends Dialog {  
    private Context context;
```

```
    public CustomDialog(Context context) {  
        super(context);  
        this.context = context;  
    }
```

@Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);
```

```
    Objects.requireNonNull(getWindow()).setBackgroundDrawable(  
        new ColorDrawable(Color.TRANSPARENT));  
    WindowManager.LayoutParams window = new WindowManager.LayoutParams();  
    window.flags = WindowManager.LayoutParams.FLAG_DIM_BEHIND;  
    window.dimAmount = 0.5f;  
    getWindow().setAttributes(window);  
    setContentView(R.layout.custom_dialog);
```



# Dialog 예제 2

```
EditText editText = findViewById(R.id.put_text);

Button saveButton = findViewById(R.id.btnSave);
saveButton.setOnClickListener(new Button.OnClickListener() {
    @Override
    public void onClick(View view) {
        Toast.makeText(context, editText.getText().toString(),
                        Toast.LENGTH_SHORT).show();
        dismiss();
    }
});

Button cancelButton = findViewById(R.id.btnCancel);
cancelButton.setOnClickListener(new Button.OnClickListener() {
    @Override
    public void onClick(View view) {
        dismiss();
    }
});
}
```



# Dialog 예제 2

## ■ CustomDialog.JAVA

```
TextView textView = findViewById(R.id.textView);
textView.setText("Custom Dialog 테스트입니다.");
Button button = findViewById(R.id.btnOK);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        onClickBtn(v);
    }
});

public void onClickBtn(View view) {
    this.dismiss();
}
}
```



# Dialog 예제 2

## ■ 일반 Dialog

- 일반적으로 알림/경고 용도로 많이 사용하고 있는 Popup Window
- 확인, 취소 Button을 표시하고 선택에 따라 행동하는 상황에서 많이 사용하게 됨
- AlertDialog.Builder 를 생성할 때 생성자 두번째 인자에 Theme를 지정할 수 있음

```
AlertDialog.Builder oDialog = new AlertDialog.Builder(this,  
    android.R.style.Theme_DeviceDefault_Light_Dialog);
```



# Dialog 예제 2



## ■ DatePickerDialog

- 날짜의 설정 및 변경을 목적으로 제공하는 Dialog
- 날짜를 선택하는 DatePicker Popup Window

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

## ■ DatePickerDialog의 Theme 옵션 (Calendar vs Spinner)

- `AlertDialog.THEME_TRADITIONAL`
- `AlertDialog.THEME_HOLO_DARK` : Spinner 방식
- `AlertDialog.THEME_HOLO_LIGHT` : Spinner 방식
- `AlertDialog.THEME_DEVICE_DEFAULT_DARK` : Calendar 방식
- `AlertDialog.THEME_DEVICE_DEFAULT_LIGHT` : Calendar 방식



# Dialog 예제 2



- DatePickerDialog를 만들기 위해서는 먼저 OnDateSetListener를 구현하고 그 안에 onDateSet() 메소드를 Overriding해야 함
- onDateSet() 메소드는 DatePickerDialog가 활성화되고 사용자가 날짜 정보를 입력한 뒤 완료 Button을 Click하였을 때 실행되는 메소드
- OnDateSetListener는 DatePickerDialog 객체를 생성할 때 두 번째 인자로 넘겨주게 됨
- 그런 다음 최종적으로 DatePickerDialog의 show() 메소드를 통해 Dialog 창을 띄우게 됨



# Dialog 예제 2

## ■ TimePickerDialog

### ■ 시간의 설정 및 변경을 목적으로 제공하는 Dialog

- ✓ TimePickerDialog(Context context, TimePickerDialog.OnTimeSetListener listener, int hourofDay, int minute, boolean is24HourView)
- ✓ TimePickerDialog(Context context, int themeResId, DTimePickerDialog.OnTimeSetListener listener, int hourofDay, int minute, boolean is24HourView)

- 첫번째 context는 Activity의 this
- 두번째 listener는 시간을 설정하고 설정 Button을 눌렀을 때 콜백 될 OnTimeSetListener
- 세번째 hourofDay는 TimePickerDialog를 띄울 때 설정 되었을 시각이며 24시간 단위로 작성됨
- 네번째 minute는 분
- 마지막 매개변수는 false면 오전/오후를 선택할 수 있고 12시간 단위로 나타나고, true로 하면 오전/오후 선택없이 24시간 단위 표시





# Dialog 예제 2



## ■ TimePickerDialog

- TimePickerDialog를 띄우는 과정도 DatePickerDialog를 띄우는 과정과 비슷함
- Dialog 창의 완료 Button을 Click하였을 때 Event 처리를 위해 TimePickerDialog.OnTimeSetListener()를 구현해야 함
- 해당 Listener의 onTimeSet() 메소드를 Overriding하여 사용자가 시간을 입력하고 완료 Button을 Click하였을 때 처리해야 할 동작을 정의해야 함
- 그런 다음 TimePickerDialog 객체를 생성하고 show()를 통해 Dialog 창을 활성화



# Dialog 예제 2



## ■ ProgressDialog

- AlertDialog의 확장된 개념으로 시간 요소가 필요한 작업에 대해 현재 작업 상태를 보여주는 기능으로 사용
- Spinning Wheel과 Progress Bar 2종류의 Graphic 형태를 제공
- 진행 정도를 알 수 없는 경우 Spinning Wheel이 사용되고, 진행 정도를 파악할 수 있는 경우 Progress Bar 사용
  - Progress Bar 형태로 표시

```
setProgressStyle(ProgressDialog.STYLE_HORIZONTAL)
```

- ProgressDialog의 구현은 ProgressDialog.show() 메소드를 호출하는 것만으로 처리할 수 있음



# Dialog 예제 2



## ■ ProgressDialog

```
ProgressDialog.show(MyActivity.this, "",  
    "Loading. Please wait...", true, true);
```

- 첫번째 매개 변수 : 현재 액티비티
- 두번째 매개 변수 : 제목
- 세번째 매개 변수 : 메시지
- 네번째 매개 변수 : 진행 정도 확정/불확정 여부
- 다섯번째 매개 변수 : 취소 가능 여부



# Dialog 예제 2

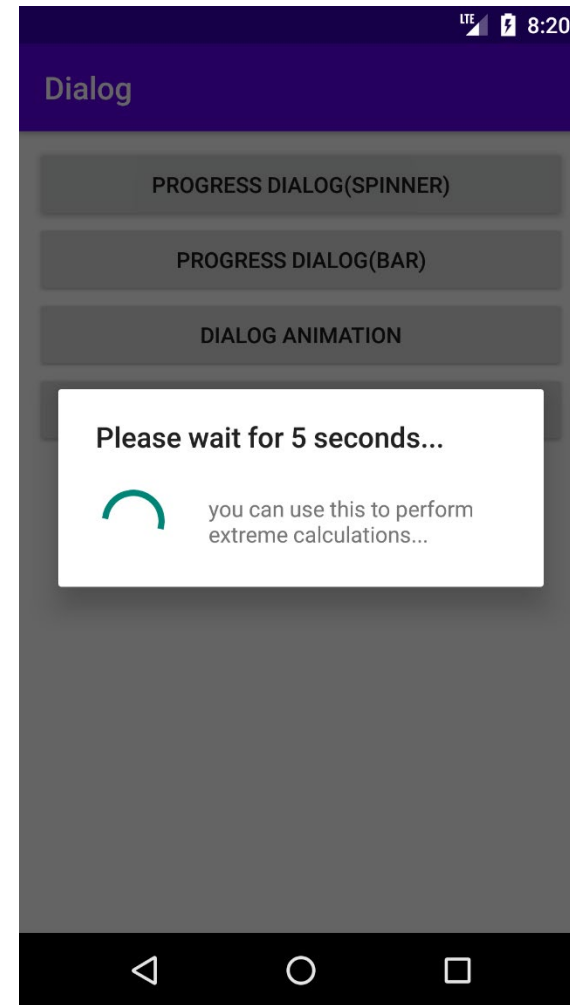
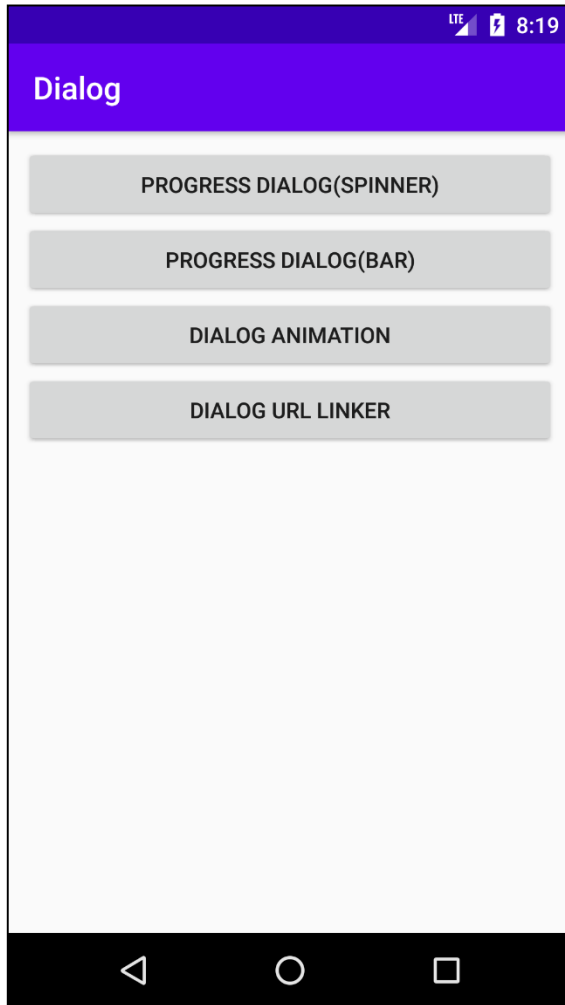


- Custom Dialog
  - 사용자가 임의로 Dialog를 Design하는 상자
  - ImageView와 TextView 등을 사용해 Dialog를 만들기가 가능
  
- Custom Dialog 만드는 방법
  - Design XML 작성하기 (dialog\_round.xml)
  - Custom Dialog 작성하기
  - 사용하기



# Dialog 예제 3

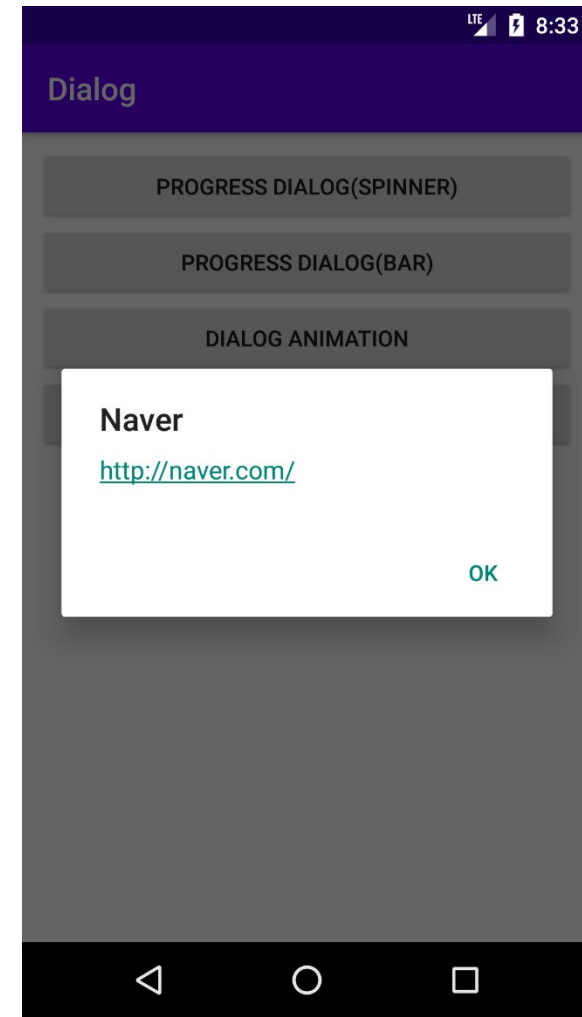
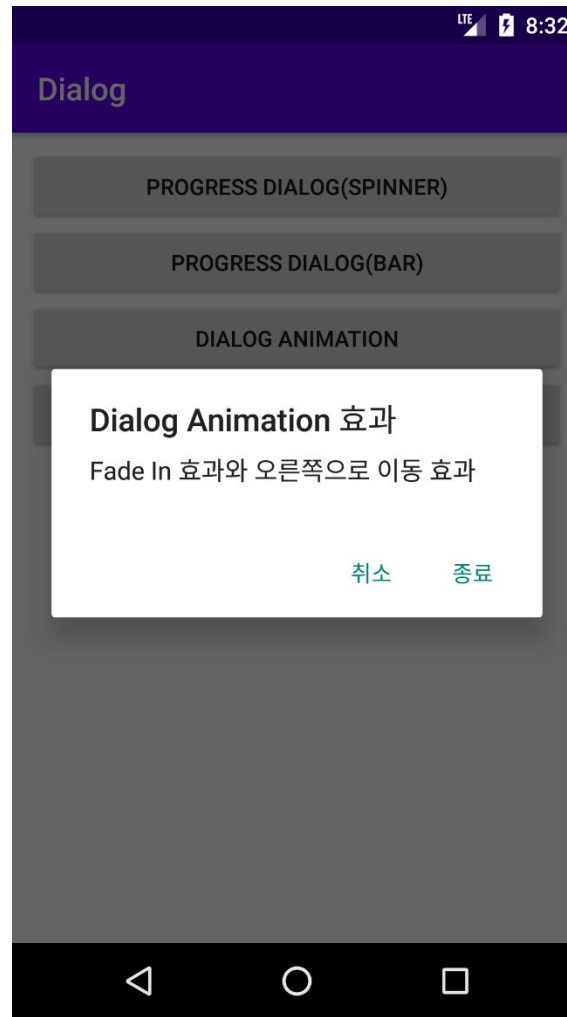
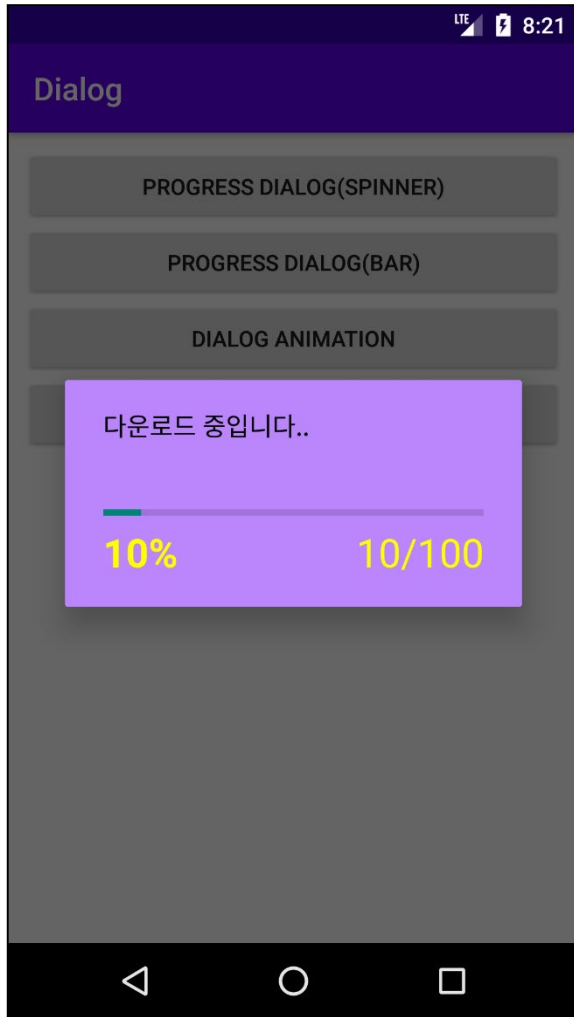
- 다음과 같은 Dialog를 만들어보자





# Dialog 예제 3

## ■ 실행 결과





# Dialog 예제 3



- ProgressDialog를 독립적으로 사용되면 의미가 없음
  - 시간이 걸리는 작업을 하기 때문에 Thread나 AsyncTask 클래스에 포함되어서 사용
- AsyncTask
  - Thread를 사용할 일이 있을 때 좀더 편리하게 이용할 수 있도록 Android에서 지원하고 있는 클래스
  - AsyncTask를 상속받아 만든 클래스는 3가지 메소드들에 ProgressDialog 기능을 구현
    - onPreExecute()
      - 작업시작, ProgressDialog 객체를 생성하고 시작
    - doInBackground()
      - 진행중, ProgressDialog의 진행 정도를 표현
    - doPostExecute()
      - 종료, ProgressDialog 종료 기능을 구현



# Dialog 예제 3



## ■ Animation 효과 적용 방법

### ■ styles.xml 내부에 Animation을 위한 XML 작성

```
<style name="PauseDialogAnimation">
    <item name="android:windowEnterAnimation">@anim/fadein</item>
    <item name="android:windowExitAnimation">
        @android:anim/slide_out_right</item>
</style>
```

- 첫 번째 windowEnterAnimation은 적용한 Widget이 시작 될 때 적용
- 세부 설정값은 @anim/fadein에 있음
- 두 번째 windowExitAnimation은 Widget이 사라질 때 설정
  - @android:anim/slide\_out\_right은 Android에서 제공하는 설정 값으로 사라질 때 화면 오른쪽으로 이동





# Dialog 예제 3

## ■ anim/fadein.xml

```
<?xml version="1.0" encoding="utf-8"?>
<alpha xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="1000"
    android:fromAlpha="0"
    android:interpolator="@android:anim/accelerate_interpolator"
    android:toAlpha="1" />
```

## ■ anim/fadeout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<alpha xmlns:android="http://schemas.android.com/apk/res/android"
    android:duration="1000"
    android:fromAlpha="1"
    android:interpolator="@android:anim/anticipate_interpolator"
    android:toAlpha="0"/>
```



# Dialog 예제 3



## ■ URL Link 거는 방법

- TextView뿐만 아니라 Dialog에도 다른 Web Page로 이동할 수 있도록 Link를 달 수가 있음
- 일반적으로 URL Link를 거는 방법과 동일하며 Link가 걸린 Text를 Dialog Message로 보내서 만드는 것임
- Link를 만들기 위해서 SpannableString 객체를 생성
  - 만들 때 생성자로 URL을 넣음
- Linkify.addLinks() 메소드를 이용해 SpannableString내에 있던 URL Text에 Link가 가능하도록 만듦



# Dialog 예제 3

## ■ 사용자 인터페이스

```
<?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"
    android:padding="10dp">

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Progress Dialog(Spinner)" />

    <Button
        android:id="@+id/button2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Progress Dialog(Bar)" />
```



# Dialog 예제 3

## ■ 사용자 인터페이스

```
<Button  
    android:id="@+id/button3"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Dialog Animation" />
```

```
<Button  
    android:id="@+id/button4"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:text="Dialog URL Linker" />
```

```
</LinearLayout>
```



# Dialog 예제 3

## ■ MainActivity.JAVA

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

@Override

```
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);
```

```
    Button button1 = findViewById(R.id.button1);  
    button1.setOnClickListener(this);  
    Button button2 = findViewById(R.id.button2);  
    button2.setOnClickListener(this);  
    Button button3 = findViewById(R.id.button3);  
    button3.setOnClickListener(this);  
    Button button4 = findViewById(R.id.button4);  
    button4.setOnClickListener(this);
```

```
}
```



# Dialog 예제 3

## ■ MainActivity.JAVA

@Override

```
public void onClick(View v) {  
    switch (v.getId()) {  
        case R.id.button1:  
            ProgressDialog dialog = ProgressDialog.show(MainActivity.this,  
                "Please wait for 5 seconds...",  
                "you can use this to perform extreme calculations...", true);  
            new Thread() {  
                public void run() {  
                    try {  
                        sleep(5000);  
                    } catch (Exception e) {  
                        }  
                    }  
                dialog.dismiss();  
            }.start();  
            break;
```



# Dialog 예제 3

## ■ MainActivity.JAVA

```
case R.id.button2:
```

```
    DownloadTask task = new DownloadTask(MainActivity.this);
```

```
    task.execute();
```

```
    break;
```

```
case R.id.button3:
```

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

```
    builder.setTitle("Dialog Animation 효과");
```

```
    builder.setOnKeyListener(new DialogInterface.OnKeyListener() {
```

```
        public boolean onKey(DialogInterface dialog,  
                               int keyCode, KeyEvent event) {
```

```
            if (keyCode == KeyEvent.KEYCODE_BACK) {
```

```
                finish();
```

```
                dialog.dismiss();
```

```
                return true;
```

```
            }
```

```
            return false;
```

```
        }
```

```
    });
```



# Dialog 예제 3

## ■ MainActivity.JAVA

```
builder.setMessage("Fade In 효과와 오른쪽으로 이동 효과")
    .setCancelable(false)
    .setPositiveButton("종료", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            finish();
        }
    })
    .setNegativeButton("취소", new DialogInterface.OnClickListener() {
        public void onClick(
            DialogInterface dialog, int id) {
            dialog.cancel();
        }
    });
AlertDialog alertDialog = builder.create();
alertDialog.getWindow().getAttributes().windowAnimations =
    R.style.PauseDialogAnimation;

alertDialog.show();
break;
```





# Dialog 예제 3

## ■ MainActivity.JAVA

case R.id.*button4*:

SpannableString site = new SpannableString("http://naver.com/");

Linkify.addLinks(site, Linkify.*ALL*);

AlertDialog builder1 = new AlertDialog.Builder(MainActivity.*this*)

.setTitle("Naver")

.setMessage(site)

.setPositiveButton("OK", null).create();

builder1.show();

((TextView) builder1.findViewById(android.R.id.*message*))

.setMovementMethod(LinkMovementMethod.*getInstance()*);

}

}

}



# Dialog 예제 3

## ■ DownloadTask.java

```
public class DownloadTask extends AsyncTask<Void, Void, Void> {  
    private Context context;  
    private ProgressDialog dialog;  
  
    public DownloadTask(Context context) {  
        this.context = context;  
        dialog = new ProgressDialog(context, R.style.MyDialogStyle);  
    }  
  
    @Override  
    protected void onPreExecute() {  
        dialog.setProgressStyle(ProgressDialog.STYLE_HORIZONTAL);  
        dialog.setMessage("다운로드 중입니다..");  
        dialog.show();  
    }  
}
```



# Dialog 예제 3



## ■ DownloadTask.java

@Override

```
protected Void doInBackground(Void... arg0) {  
    try {  
        for (int i = 0; i <= 10; i++) {  
            dialog.setProgress(i * 10);  
            Thread.sleep(1000);  
        }  
    } catch (InterruptedException e) {  
        Toast.makeText(context, e.getMessage(), Toast.LENGTH_SHORT).show();  
    }  
    return null;  
}
```

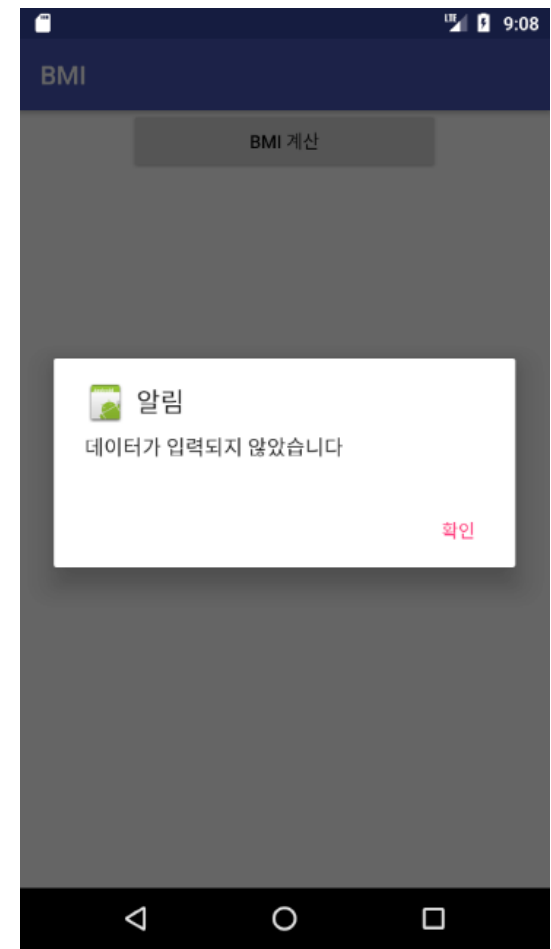
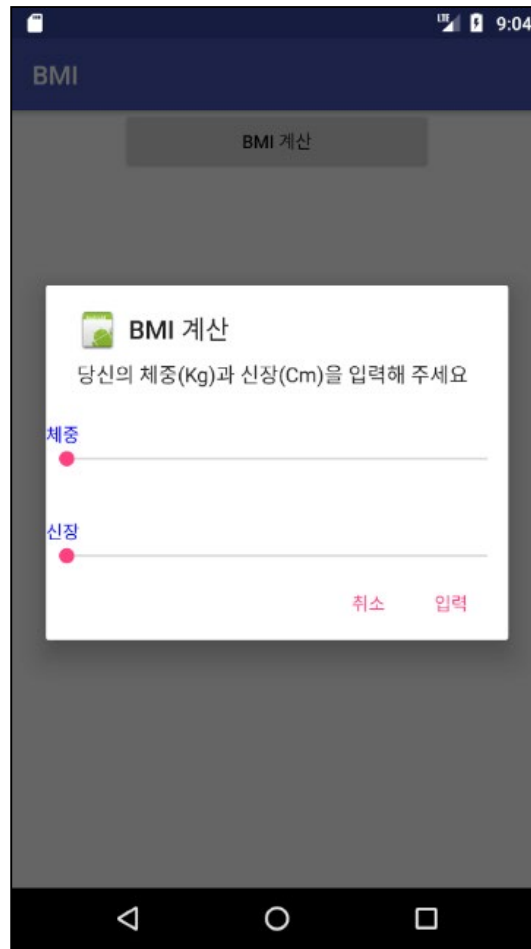
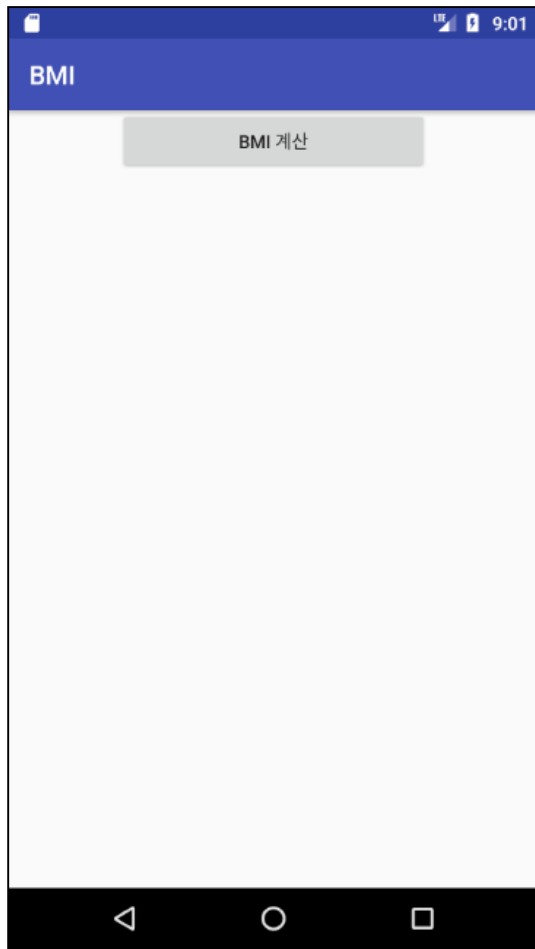
@Override

```
protected void onPostExecute(Void result) {  
    dialog.dismiss();  
}  
}
```



# Dialog 예제 4(BMI 계산)

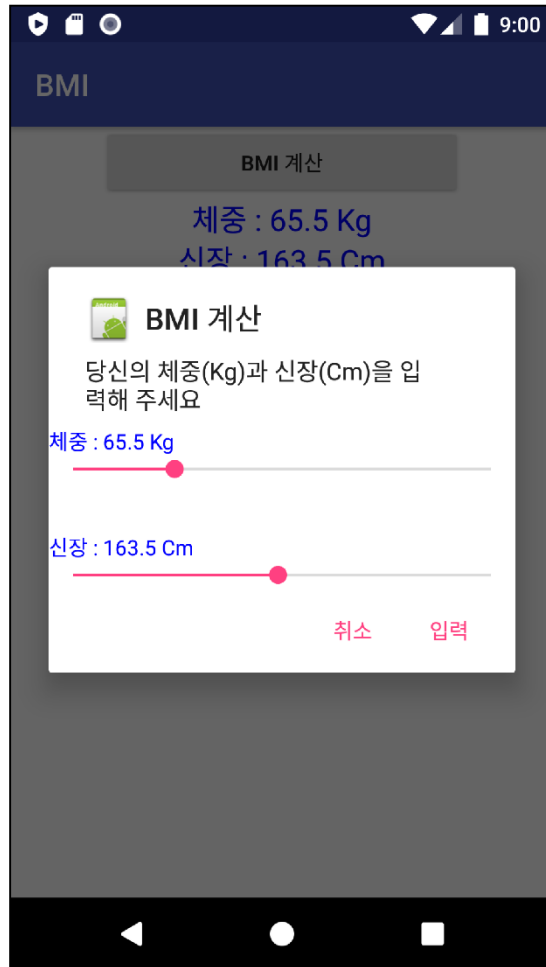
- BMI 계산을 위한 체중과 신장 입력을 대화상자를 통해서 받는 프로그램을 만들어보자





# Dialog 예제 4(BMI 계산)

## ■ 실행 화면





# Dialog 예제 4(BMI 계산)

■ BMI 지수 = 몸무게(kg) ÷ (키(m) × 키(m))





# Dialog 예제 4(BMI 계산)

## ■ 사용자 인터페이스

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

    <Button
        android:id="@+id/button1"
        android:layout_width="240dp"
        android:layout_height="wrap_content"
        android:text="BMI 계산" />
```



# Dialog 예제 4(BMI 계산)

## ■ 사용자 인터페이스

```
<TextView  
    android:id="@+id/textView1"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text=" "  
    android:textColor="#0000FF"  
    android:textSize="20dp" />
```

```
<TextView  
    android:id="@+id/textView2"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text=" "  
    android:textColor="#0000FF"  
    android:textSize="20dp" />
```





# Dialog 예제 4(BMI 계산)



## ■ 사용자 인터페이스

```
<TextView
    android:id="@+id/textView3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text=" "
    android:textColor="#0000FF"
    android:textSize="20dp" />
</LinearLayout>
```



# Dialog 예제 4(BMI 계산)

## ■ JAVA 프로그램

```
public class MainActivity extends AppCompatActivity {  
    TextView textView, textView1, textView2;  
    double weight = 0.0;  
    double height = 0.0;
```

### @Override

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    textView = findViewById(R.id.textView1);  
    textView1 = findViewById(R.id.textView2);  
    textView2 = findViewById(R.id.textView3);
```



# Dialog 예제 4(BMI 계산)

## ■ JAVA 프로그램

```
Button button = findViewById(R.id.button1);
button.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        ShowDialog("BMI 계산",
            "당신의 체중(Kg)과 신장(Cm)을 입력해 주세요");
    }
});
}
```



# Dialog 예제 4(BMI 계산)

## ■ JAVA 프로그램

```
private void alertDialog(String title, final String msg) {  
    AlertDialog.Builder alert = new AlertDialog.Builder(this);  
    alert.setTitle(title);  
    alert.setMessage(msg);  
    alert.setIcon(R.drawable.icon);  
    alert.setPositiveButton("확인", new DialogInterface.OnClickListener() {  
        @Override  
        public void onClick(DialogInterface dialogInterface, int i) {  
            Toast.makeText(getBaseContext(), msg,  
                           Toast.LENGTH_LONG).show();  
        }  
    });  
    alert.show();  
}
```



# Dialog 예제 4(BMI 계산)

## ■ JAVA 프로그램

```
private void ShowDialog(String title, String msg) {  
    LinearLayout linearLayout = new LinearLayout(this);  
    LinearLayout.LayoutParams layoutParams =  
        new LinearLayout.LayoutParams(  
            ViewGroup.LayoutParams.MATCH_PARENT,  
            ViewGroup.LayoutParams.MATCH_PARENT);  
    linearLayout.setLayoutParams(layoutParams);  
    linearLayout.setOrientation(LinearLayout.VERTICAL);  
    TextView textView3 = new TextView(this);  
    textView3.setText("체중");  
    textView3.setTextColor(Color.BLUE);  
    TextView textView4 = new TextView(this);  
    textView4.setPadding(0,100,0,0);  
    textView4.setText("신장");  
    textView4.setTextColor(Color.BLUE);  
    final SeekBar seekBar = new SeekBar(this);  
    SeekBar seekBar1 = new SeekBar(this);
```



# Dialog 예제 4(BMI 계산)



## ■ JAVA 프로그램

```
seekBar.setMax(250);
seekBar1.setMax(300);
linearLayout.addView(textView3);
linearLayout.addView(seekBar);
linearLayout.addView(textView4);
linearLayout.addView(seekBar1);
AlertDialog.Builder builder = new AlertDialog.Builder(this);
builder.setTitle(title);
builder.setMessage(msg);
builder.setIcon(R.drawable.icon);
builder.setView(linearLayout);
```



# Dialog 예제 4(BMI 계산)

## ■ JAVA 프로그램

```
builder.setPositiveButton("입력",
    new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog, int which) {
    if (weight == 0.0 || height == 0.0) {
        alertDialog("알림", "데이터가 입력되지 않았습니다");
        textView.setText("");
        textView1.setText("");
        textView2.setText("");
    } else {
        double test = height / 100.0;
        double bmi = weight / (test * test);

        textView2.setText(String.format(
            "BMI : %, .2fWn%s", bmi, getBMIResult(bmi)));
        weight = 0.0;
    }
}
});
```



# Dialog 예제 4(BMI 계산)

## ■ JAVA 프로그램

```
builder.setNegativeButton("취소",  
    new DialogInterface.OnClickListener() {  
        public void onClick(DialogInterface dialog, int which) {  
            textView.setText("");  
            textView1.setText("");  
            textView2.setText("");  
        }  
    });  
builder.show();
```





# Dialog 예제 4(BMI 계산)

## ■ JAVA 프로그램

```
seekBar.setOnSeekBarChangeListener(  
    new SeekBar.OnSeekBarChangeListener() {  
        @Override  
        public void onProgressChanged(SeekBar seekBar, int i, boolean b) {  
            weight = ((i * 5) / 10.0) + 35.0;  
            textView.setText(String.format("체중 : %.1f Kg", weight));  
            textView3.setText(String.format("체중 : %.1f Kg", weight));  
        }  
  
        @Override  
        public void onStartTrackingTouch(SeekBar seekBar) {  
        }  
  
        @Override  
        public void onStopTrackingTouch(SeekBar seekBar) {  
        }  
    });
```



# Dialog 예제 4(BMI 계산)

## ■ JAVA 프로그램

```
seekBar1.setOnSeekBarChangeListener(  
    new SeekBar.OnSeekBarChangeListener() {  
        @Override  
        public void onProgressChanged(SeekBar seekBar, int i, boolean b) {  
            height = ((i * 5)/10.0) + 90.0 ;  
            textView1.setText(String.format("신장 : %.1f Cm", height));  
            textView4.setText(String.format("신장 : %.1f Cm", height));  
        }  
  
        @Override  
        public void onStartTrackingTouch(SeekBar seekBar) {  
        }  
  
        @Override  
        public void onStopTrackingTouch(SeekBar seekBar) {  
        }  
    });  
}
```



# Dialog 예제 4(BMI 계산)

## ■ JAVA 프로그램

```
private String getBMIResult(double bmi) {  
    String bmiResult = "당신은 ";  
    if(bmi <= 18.5){  
        bmiResult += "저 체중 입니다!";  
    }else if(bmi > 18.5 && bmi < 23){  
        bmiResult += "정상 체중 입니다!";  
    }else if(bmi >= 23 && bmi < 25){  
        bmiResult += "과 체중 입니다!";  
    }else{  
        bmiResult += "비만 입니다!";  
    }  
    return bmiResult;  
}
```