

# 대화상자 실습

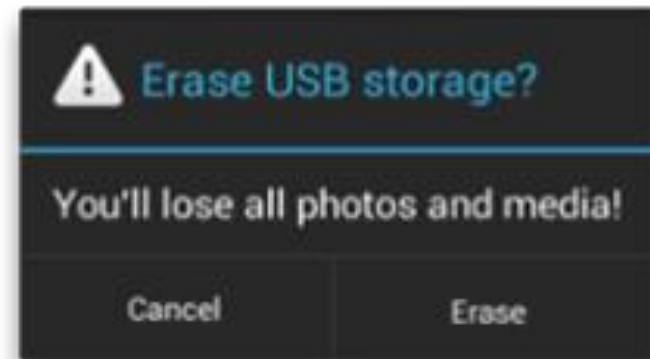
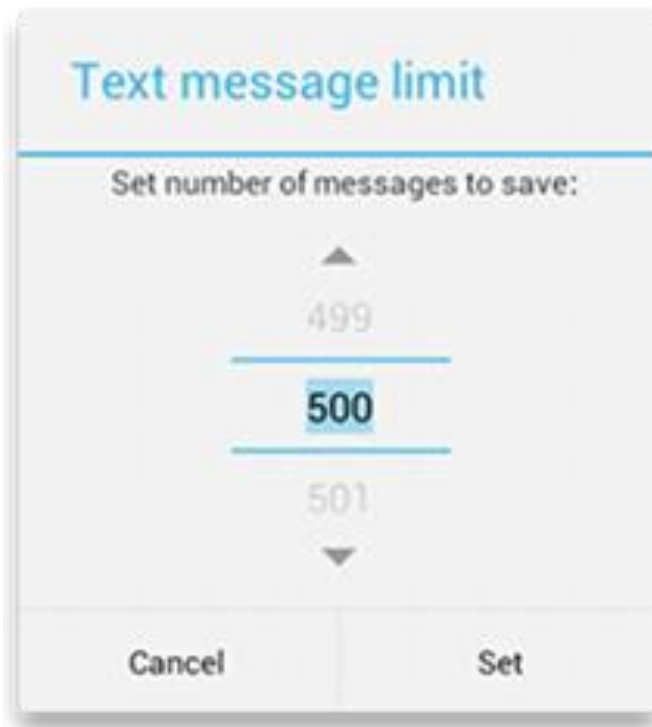
---

배 희호 교수  
경북대학교  
스마트IT과



# Dialog 개요

- Dialog는 사용자에게 중요한 사항을 Message로 출력하고 사용자로부터 입력을 받아들이는 아주 보편적인 User Interface
- Window의 모달(Modal) 방식
- 사용자와 대화를 하는 상자





# Dialog 개요

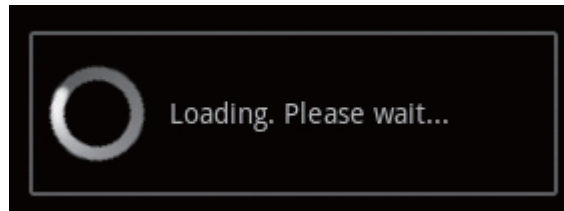
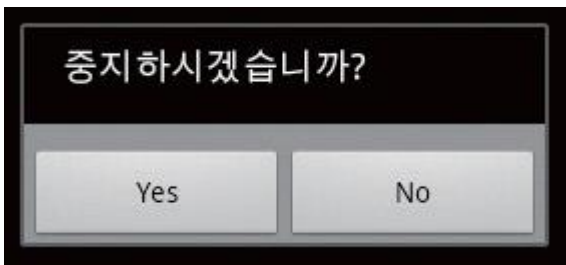


- Activity의 화면에 정면으로 표시되는 작은 Window(창)
- Dialog가 표시되면 Dialog의 아래에 위치하게 되는 Activity는 Focus를 잃고 Dialog가 사용자와 상호작용을 함
- Dialog는 사용자가 하던 일을 중단 시키고, Alert을 표시하거나 Progress Bar의 진행 상태를 표시하거나, 로그인 처리와 같은 작업에서 Application에 직접 관계된 짧은 작업등을 수행하기 위해 사용 됨



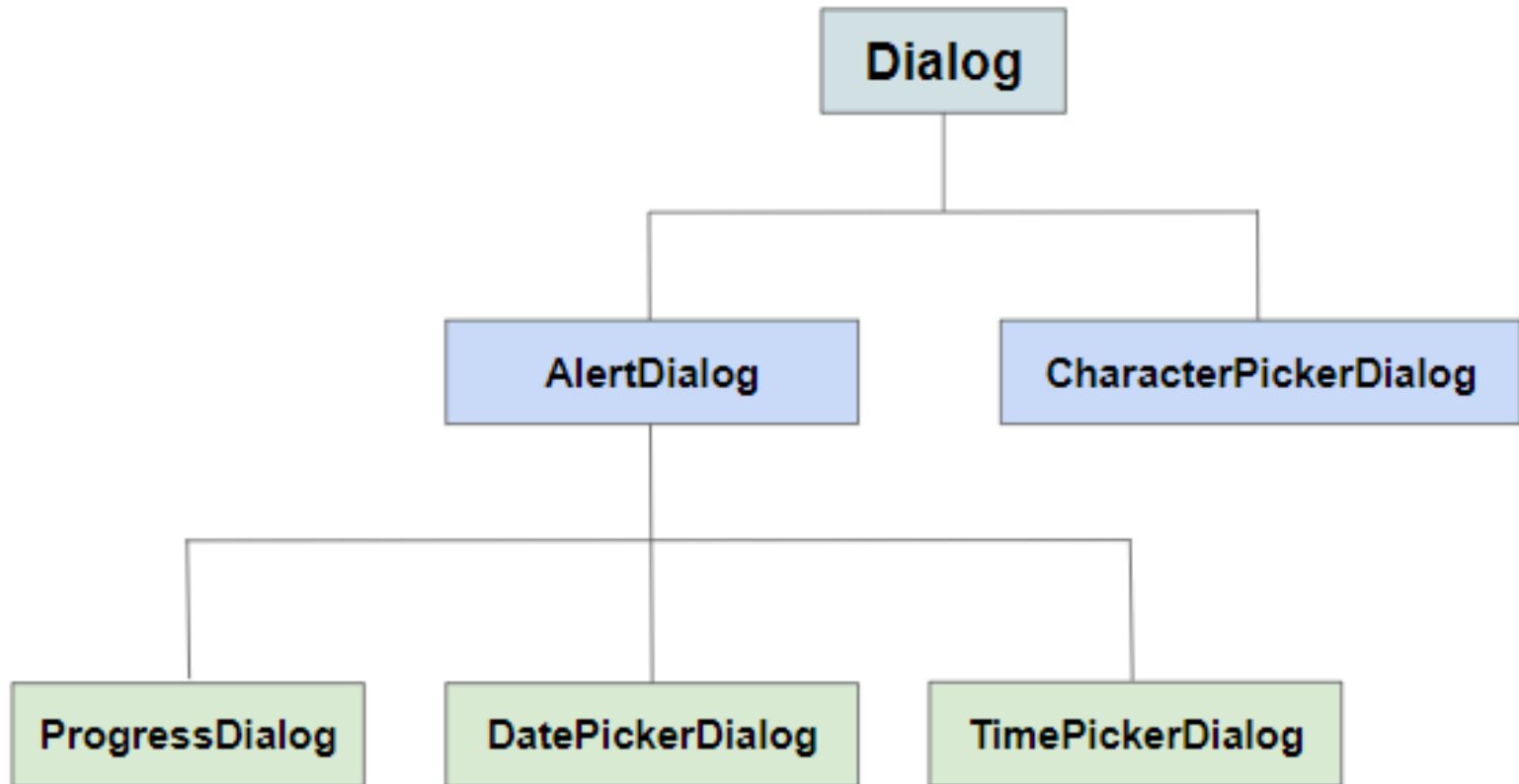
# Dialog 개요

- Dialog의 종류
  - AlertDialog
    - 0~3개 Button을 가질 수 있는 기본적인 Dialog
  - ProgressDialog
    - 진행을 나타내는 Wheel이나 Bar를 표시하는 Dialog
  - DatePickerDialog
    - 사용자가 날짜를 선택할 수 있도록 하는 Dialog
  - TimePickerDialog
    - 사용자가 시간을 선택할 수 있도록 하는 Dialog





# Dialog 개요





# Dialog 개요



- 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
    - 이럴 때는 setOwnerActivity(Activity) 메소드를 사용해서 Dialog를 Activity에 붙일 수 있음



# Dialog 개요

## ■ Dialog 생성 및 표시

- Dialog를 표시할 때는 showDialog(int) 메소드를 호출
  - 화면에 표시되기를 원하는 Dialog의 식별 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 리턴



# Dialog 개요



- 대화상자 생성 및 표시
  - 대화상자를 표시하기 전에 필요한 경우 `onPrepareDialog(int, Dialog)` 콜백 메소드를 호출할 수 있음
    - 이 메소드는 대화상자가 열릴 때 마다 속성의 변경이 필요한 경우에 정의해서 사용할 수 있음
    - `onCreateDialog()` 메소드는 처음 대화상자가 표시될 때 한번만 호출되는 것에 비해 `onPrepareDialog()` 메소드는 대화상자가 열릴 때 마다 호출됨





# Dialog 개요

```
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 개요



```
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]모달과 모달리스의 차이

- 윈도우의 Form(형식)과 Dialog(대화상자)를 모달 혹은 모달리스로 표시할 수 있음
  - 모달(Modal)
    - 간단한 개념으로 모달 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 메시지를 표시하는 Dialog는 거의 모달로 나타냄
    - 예) Visual studio의 정보 대화상자나, api의 messagebox



# [참조]모달과 모달리스의 차이

- 윈도우의 Form(형식)과 Dialog(대화상자)를 모달 혹은 모달리스로 표시할 수 있음
  - 모달리스(Modeless)
    - 모달과는 반대의 개념으로 창이 열리더라도 기존에 있던 창을 작업 할 수 있음
    - 모달리스는 사용자가 순서에 상관없이 접근이 가능하기 때문에 사용자의 동작에 관계 없이 응용 프로그램의 상태를 일관성 있게 유지해야 함
    - 프로그래밍 하기가 어려움
    - 예) 도구창은 종종 모달리스 형식으로 표시되며, Visual studio의 편집 메뉴에서 사용 할 수 있는 대화상자들
    - 모달리스 형식은 자주 사용하는 명령이나 정보를 표시하는데 사용



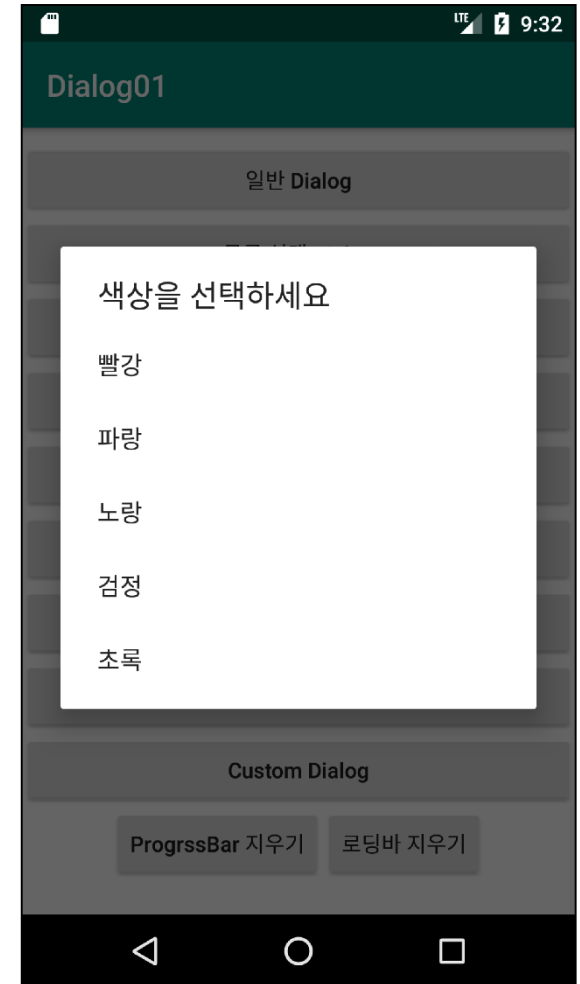
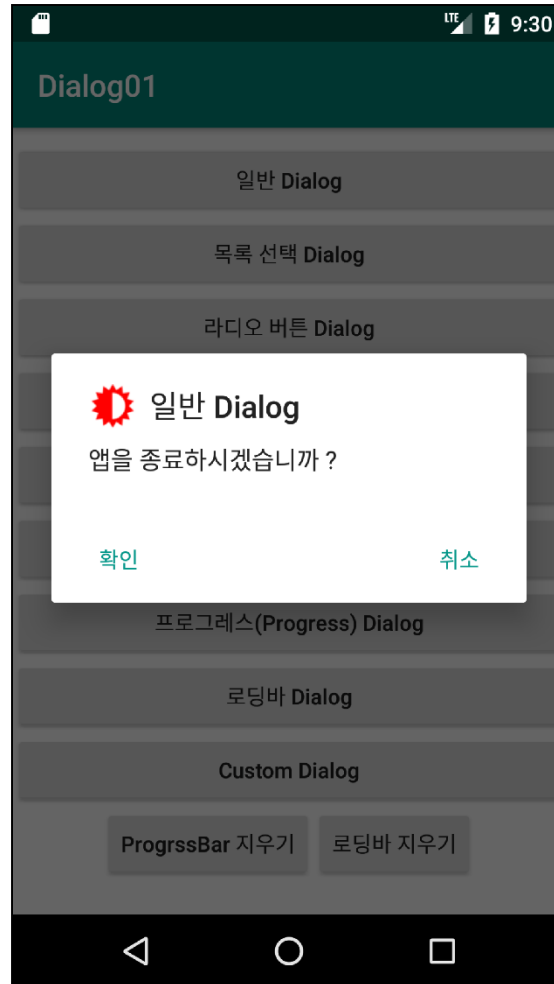
# [참조]모달과 모달리스의 차이

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



# Dialog 예제 1

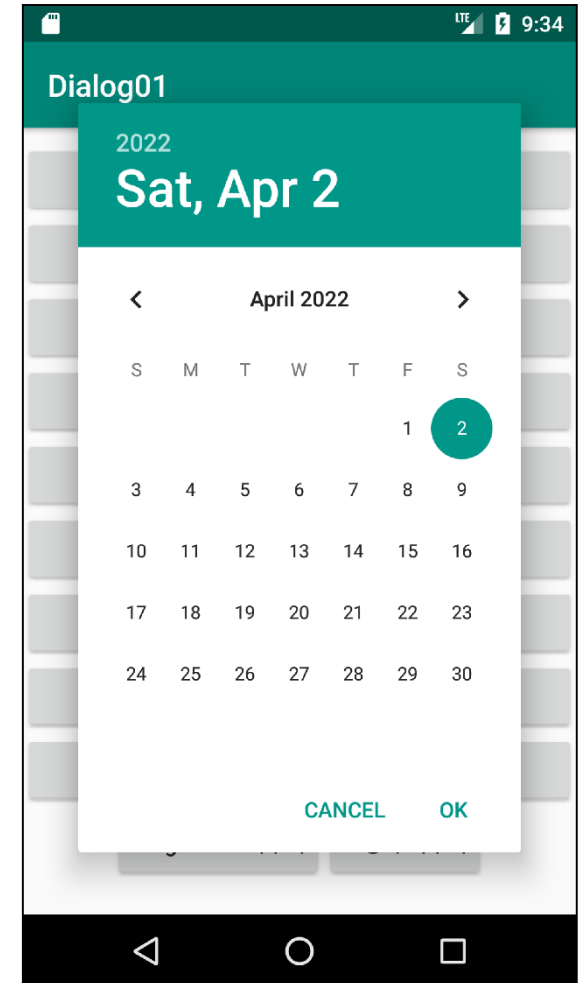
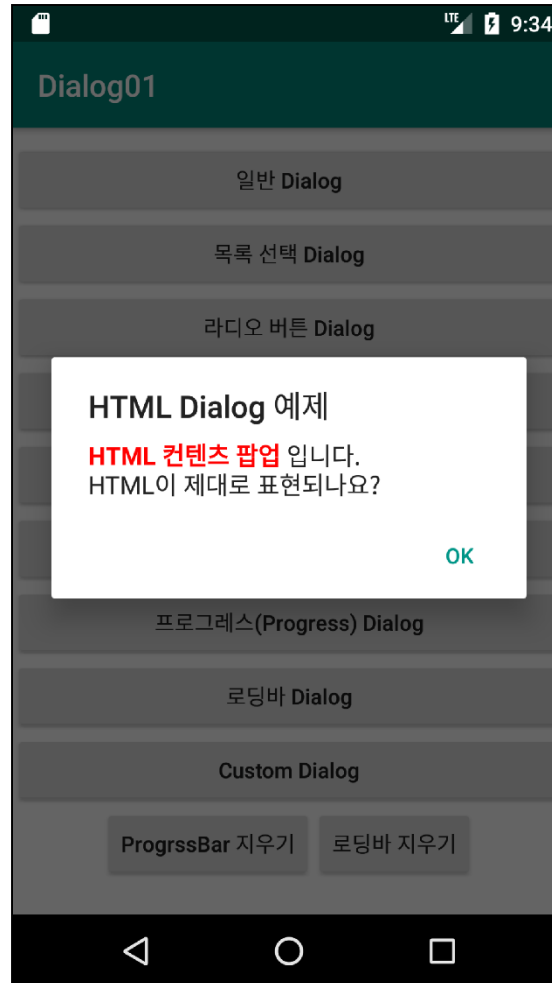
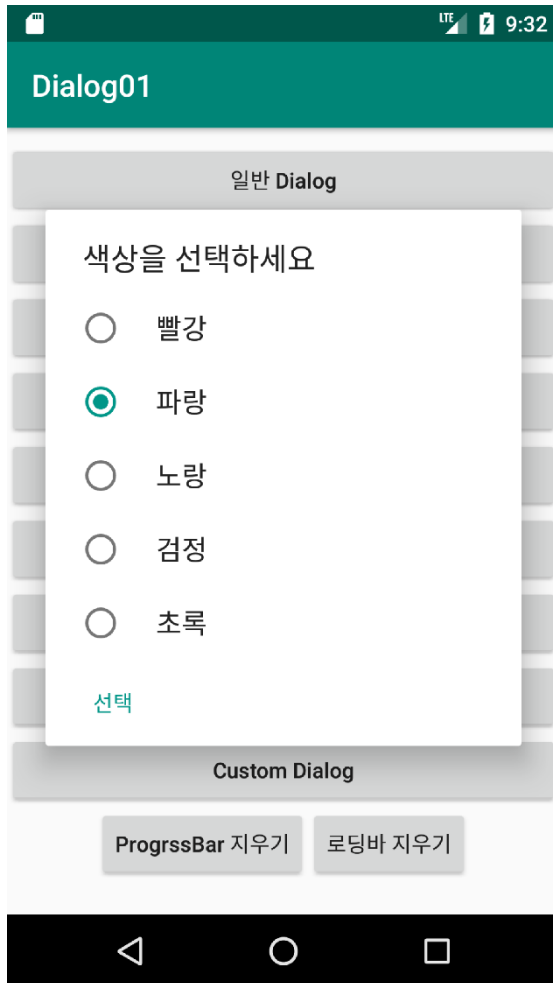
■ 다음과 같은 Dialog를 만들어보자





# Dialog 예제 1

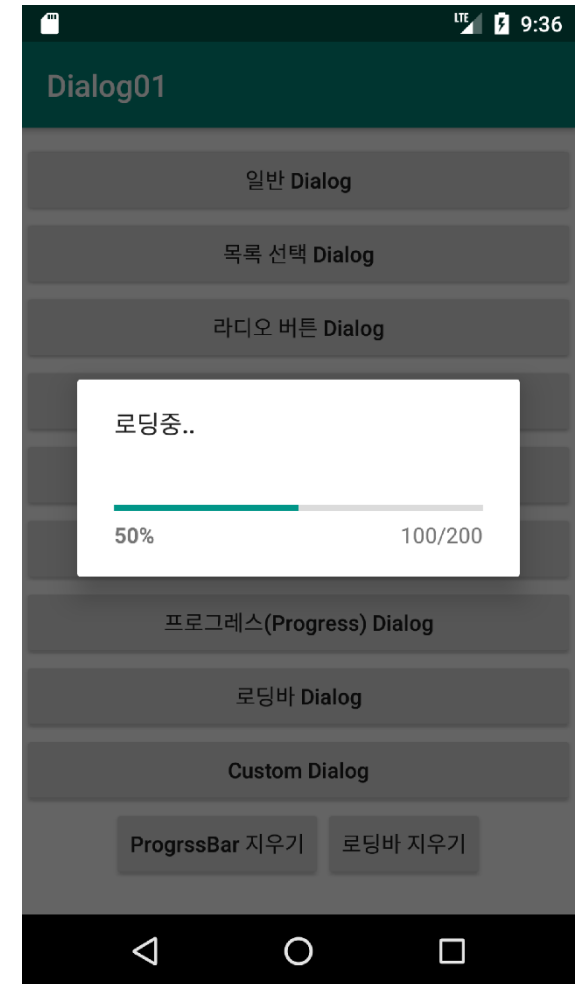
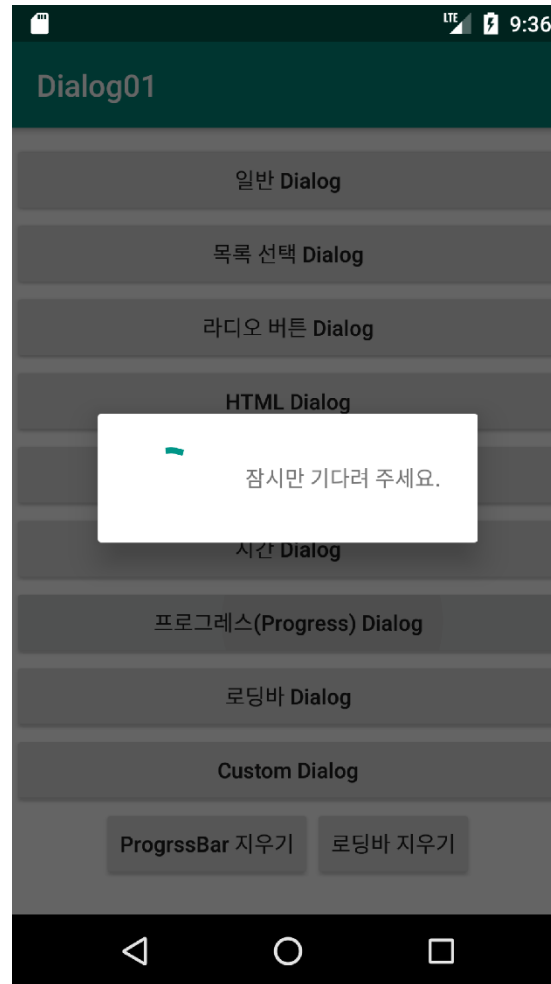
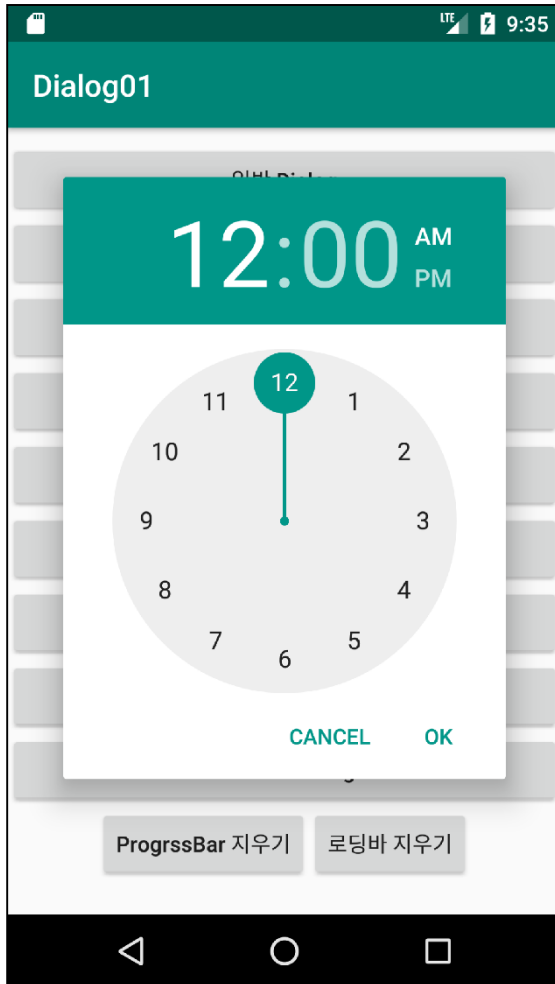
## ■ 실행 결과





# Dialog 예제 1

## ■ 실행 결과

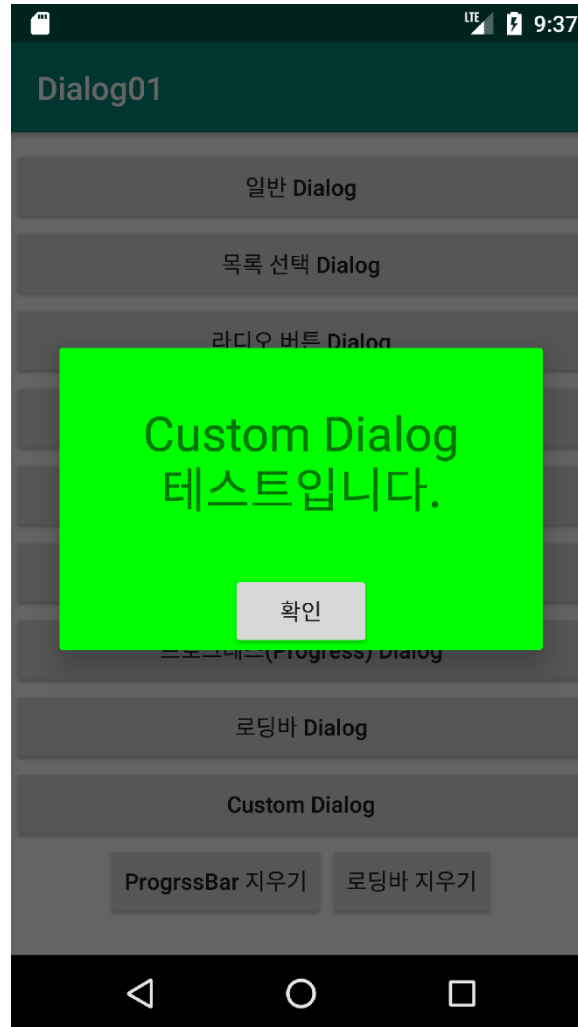






# Dialog 예제 1

## ■ 실행 결과





# 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"
    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 1

## ■ 사용자 인터페이스

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

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

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



# Dialog 예제 1

## ■ 사용자 인터페이스

```
<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="showDialog5"
    android:text="날짜 Dialog"
    android:textAllCaps="false" />

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="showDialog6"
    android:text="시간 Dialog"
    android:textAllCaps="false" />

<Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:onClick="showDialog7"
    android:text="프로그레스(Progress) Dialog"
    android:textAllCaps="false" />
```



# Dialog 예제 1

## ■ 사용자 인터페이스

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

```
<Button  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:onClick="showDialog9"  
    android:text="Custom Dialog"  
    android:textAllCaps="false" />
```



# Dialog 예제 1

## ■ 사용자 인터페이스

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center_horizontal"
    android:orientation="horizontal">
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="dismissDialog10"
        android:text="ProgrssBar 지우기"
        android:textAllCaps="false" />
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:onClick="dismissDialog11"
        android:text="로딩바 지우기" />
</LinearLayout>
</LinearLayout>
```



# Dialog 예제 1

## ■ 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 1

```
public void showDialog1 (View view) {  
    AlertDialog.Builder builder = new AlertDialog.Builder(this,  
        android.R.style.Theme_DeviceDefault_Light_Dialog_Alert);  
    builder.setMessage("앱을 종료하시겠습니까 ?")  
        .setIcon(R.drawable.star1).setTitle("일반 Dialog")  
        .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 1

## ■ MainActivity.JAVA

```
public void showDialog2(View view) {  
    final CharSequence[] color = {"빨강", "파랑", "노랑", "검정", "초록"};  
    AlertDialog.Builder builder = new AlertDialog.Builder(this,  
        android.R.style.Theme_DeviceDefault_Light_Dialog_Alert);  
    builder.setTitle("색상을 선택하세요")  
        .setItems(color, new DialogInterface.OnClickListener() {  
            @Override  
            public void onClick(DialogInterface dialog, int which) {  
                Toast.makeText(getApplicationContext(), color[which] +  
                    "을 선택했습니다", Toast.LENGTH_LONG).show();  
            }  
        })  
        .setCancelable(false)  
        .show();  
}
```



# Dialog 예제 1

## ■ MainActivity.JAVA

```
int selectItem = -1;
public void showDialog3(View view) {
    final CharSequence[] color = {"빨강", "파랑", "노랑", "검정", "초록"};
    AlertDialog.Builder builder = new AlertDialog.Builder(this,
        android.R.style.Theme_DeviceDefault_Light_Dialog_Alert);
    builder.setTitle("색상을 선택하세요")
        .setSingleChoiceItems(color, -1,
            new DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    selectItem = which;
                }
            })
}
```



# Dialog 예제 1

## ■ MainActivity.JAVA

```
.setNeutralButton("선택", new DialogInterface.OnClickListener() {  
    public void onClick(DialogInterface dialog, int which) {  
        if (selectItem == -1) {  
            Toast.makeText(getApplicationContext(), "선택해 주세요",  
                Toast.LENGTH_LONG).show();  
        } else {  
            Toast.makeText(getApplicationContext(), color[selectItem],  
                Toast.LENGTH_LONG).show();  
        }  
    }  
})  
.setCancelable(false);  
AlertDialog dialog = builder.create();  
dialog.getWindow().clearFlags(  
    WindowManager.LayoutParams.FLAG_DIM_BEHIND);  
dialog.show();  
}
```



# Dialog 예제 1

## ■ MainActivity.JAVA

```
public void showDialog4(View view) {  
    AlertDialog.Builder builder = new AlertDialog.Builder(this,  
        android.R.style.Theme_DeviceDefault_Light_Dialog_Alert);  
    String strHtml =  
        "<b><font color='#ff0000'>HTML 콘텐츠 팝업</font></b> 입니다.  
        <br/>HTML이 제대로 표현되나요?";  
    Spanned oHtml;  
    if (android.os.Build.VERSION.SDK_INT <  
        android.os.Build.VERSION_CODES.M) {  
        oHtml = Html.fromHtml(strHtml);  
    } else {  
        oHtml = Html.fromHtml(strHtml, Html.FROM_HTML_MODE_LEGACY);  
    }  
    builder.setTitle("HTML Dialog 예제")  
        .setMessage(oHtml)  
        .setPositiveButton("ok", null)  
        .setCancelable(false)  
        .show();  
}
```



# Dialog 예제 1

## ■ MainActivity.JAVA

```
public void showDialog5(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 mDateSetListener =  
        new DatePickerDialog.OnDateSetListener() {  
        public void onDateSet(DatePicker view, int year, int monthOfYear,  
                                int dayOfMonth) {  
            String strDate = year + "년 ";  
            strDate += (monthOfYear + 1) + "월 ";  
            strDate += dayOfMonth + "일";  
            Toast.makeText(getApplicationContext(), strDate,  
                                Toast.LENGTH_SHORT).show();  
        }  
    };  
};
```



# Dialog 예제 1

## ■ MainActivity.JAVA

```
DatePickerDialog dialog = new DatePickerDialog(this,  
        android.R.style.Theme_DeviceDefault_Light_Dialog,  
        mDateSetListener, year, month, day);  
dialog.getWindow().clearFlags(  
        WindowManager.LayoutParams.FLAG_DIM_BEHIND);  
dialog.show();  
}
```



# Dialog 예제 1

## ■ MainActivity.JAVA

```
public void showDialog6(View view) {
    TimePickerDialog.OnTimeSetListener mTimeSetListener =
        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,
        mTimeSetListener, 0, 0, false);
    dialog.getWindow().clearFlags(
        WindowManager.LayoutParams.FLAG_DIM_BEHIND);
    dialog.show();
}
```



# Dialog 예제 1

## ■ MainActivity.JAVA

```
public void showDialog7(View view) {  
    progress = new ProgressDialog(this,  
        android.R.style.Theme_DeviceDefault_Light_Dialog);  
    progress.setProgressStyle(ProgressDialog.STYLE_SPINNER);  
    progress.setMessage("잠시만 기다려 주세요.");  
    progress.show();  
}  
  
public void showDialog8(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.show();  
    for (int i = 1; i <= 10; i++) {  
        progress1.setProgress(i * 10);  
    }  
}
```





# Dialog 예제 1

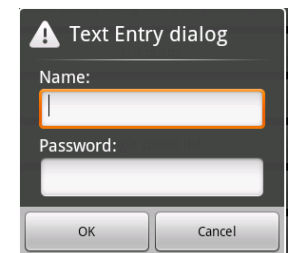
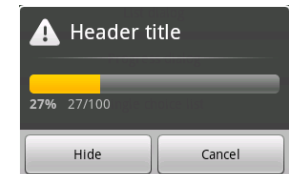
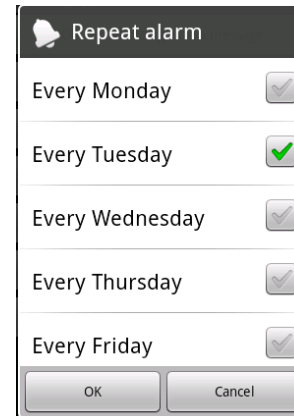
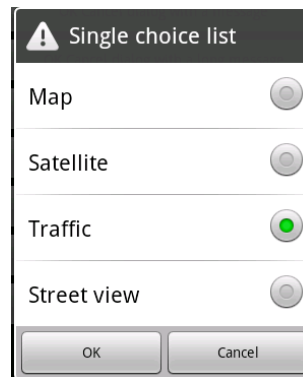
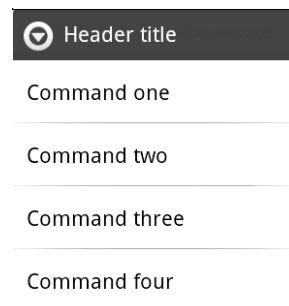
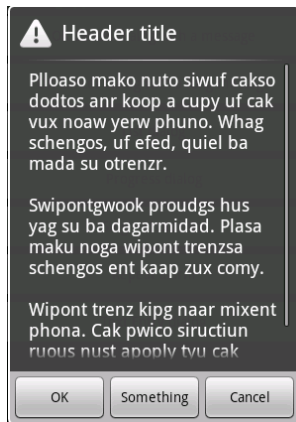
## ■ MainActivity.JAVA

```
public void showDialog9(View view) {  
    CustomDialog dialog = new CustomDialog(this);  
    dialog.setCancelable(false);  
    dialog.show();  
}  
  
public void dismissDialog10(View view) {  
    if (progress != null)  
        progress.dismiss();  
}  
  
public void dismissDialog11(View view) {  
    if (progress1 != null)  
        progress1.dismiss();  
}  
}
```



# Dialog 예제 1

- 화면의 일부 영역을 할당 받아 사용자의 제어권을 확보하고, 정보의 알림과 사용자의 특정 행동을 유도할 수 있도록 버튼을 제공해 주어 사용자의 선택과 정보 유도를 용이하게 해주는 역할을 수행하는 Widget
- Dialog의 내용에는 정보안내, 라디오 버튼, 체크박스, 리스트 박스, 프로그래스 등 다양한 위젯을 포함
- Dialog 안에 3 종류의 버튼을 선택적으로 사용할 수 있도록 제공





# Dialog 활용 2

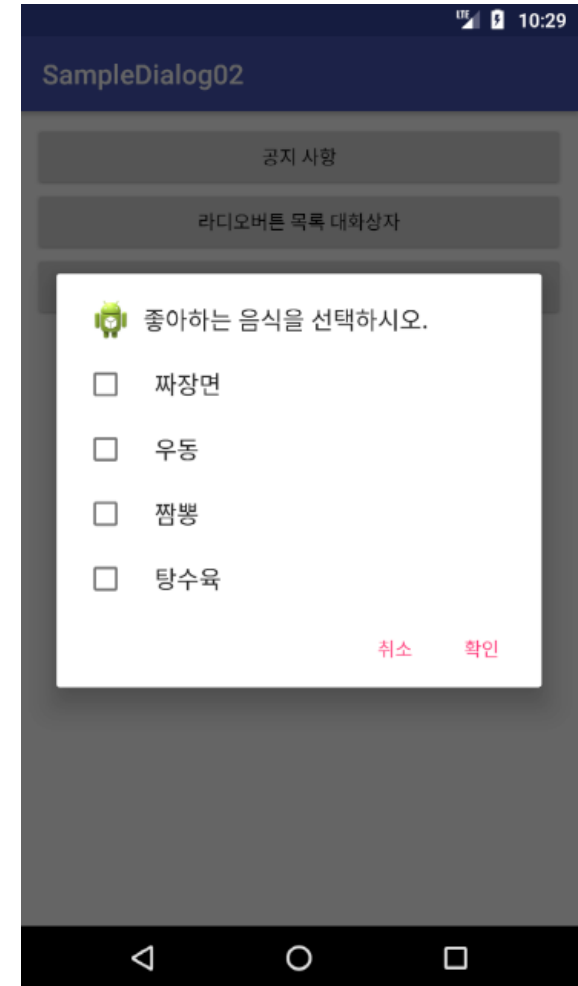
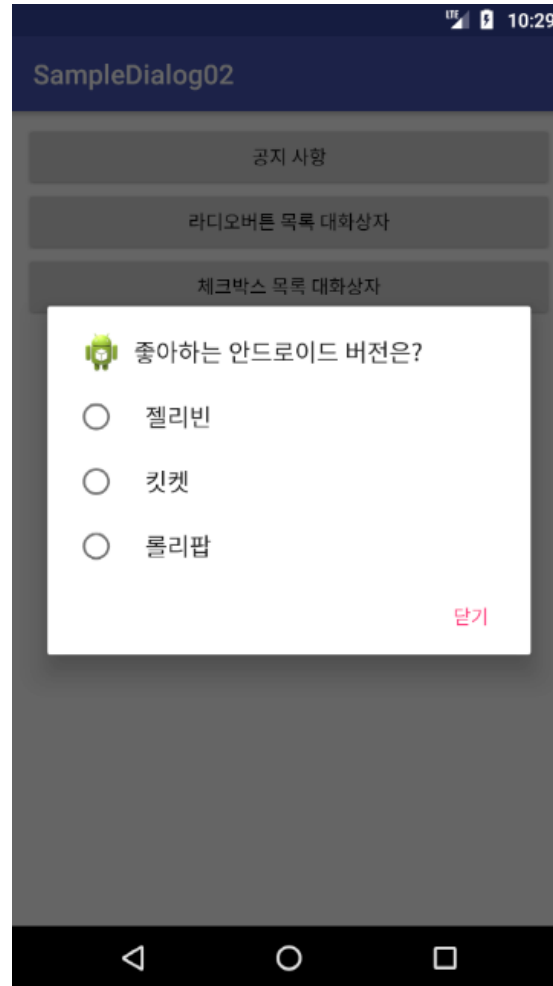
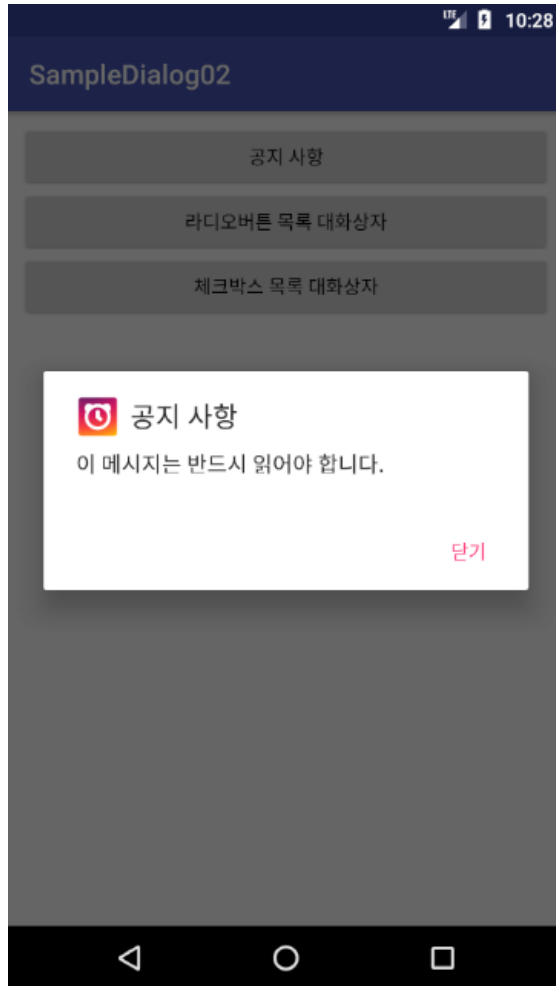
- 다음과 같은 각종 대화상자를 만들어 보자





# Dialog 활용 2

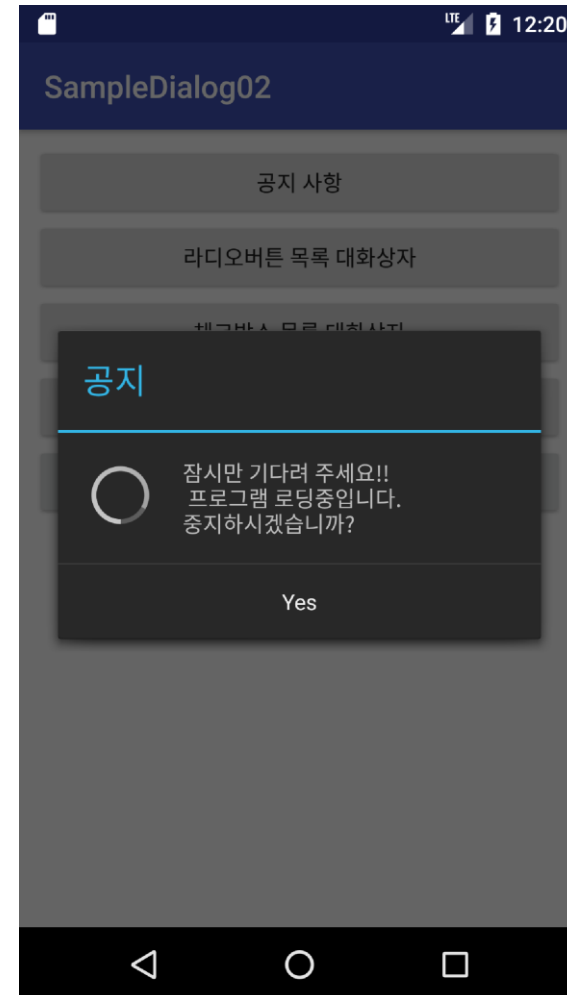
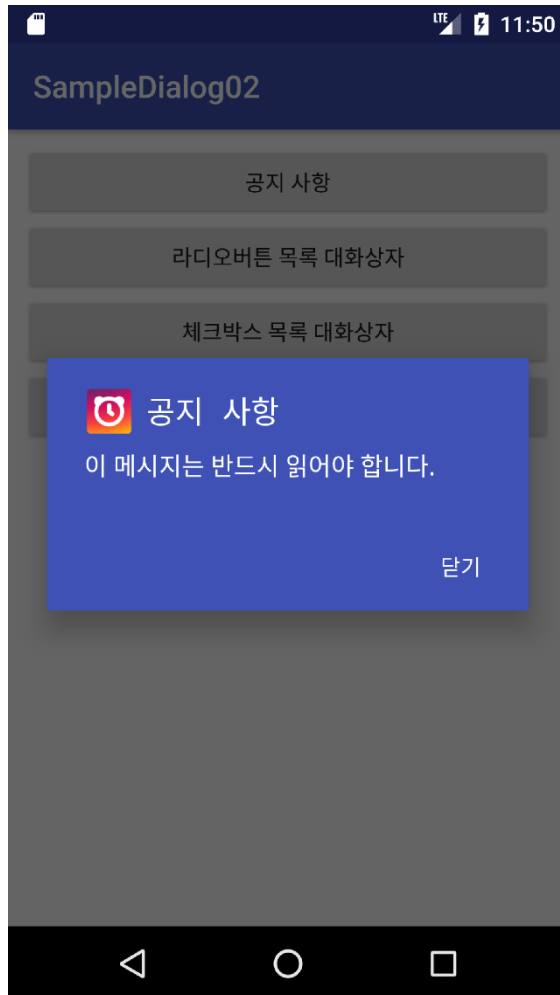
## ■ 실행 화면





# Dialog 활용 2

## ■ 실행 화면





# Dialog 활용 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:layout_margin="10dp"
    tools:context="com.example.bae.report2.MainActivity">

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



# Dialog 활용 2

## ■ 사용자 인터페이스

```
<Button
    android:id="@+id/button1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="라디오버튼 목록 대화상자"/>

<Button
    android:id="@+id/button2"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="체크박스 목록 대화상자"/>

</LinearLayout>
```



# Dialog 활용 2

## ■ 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>
  </string-array>
</resources>
```





# Dialog 활용 2

## ■ Java 프로그램

```
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 button1 = findViewById(R.id.button1);  
        Button button2 = findViewById(R.id.button2);
```



# Dialog 활용 2

## ■ Java 프로그램

```
button.setOnClickListener(new Button.OnClickListener() {  
    public void onClick(View v) {  
        new AlertDialog.Builder(MainActivity.this)  
            .setTitle("공지 사항")  
            .setMessage("이 메시지는 반드시 읽어야 합니다.")  
            .setIcon(R.drawable.alarm)  
            .setCancelable(false)  
            .setNegativeButton("닫기", null)  
            .show();  
    }  
});
```



# Dialog 활용 2

## ■ Java 프로그램

```
button1.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        final String[] versionArray = new String[]{"젤리빈", "킷켓", "롤리팝"};  
        AlertDialog.Builder dlg = new AlertDialog.Builder(MainActivity.this);  
        dlg.setTitle("좋아하는 안드로이드 버전은?");  
        dlg.setIcon(R.drawable.ic_launcher);  
        dlg.setSingleChoiceItems(versionArray, -1,  
            new DialogInterface.OnClickListener() {  
                public void onClick(DialogInterface dialog, int which) {  
                    Toast.makeText(MainActivity.this, "좋아하는 버전 : "  
                        + versionArray[which], Toast.LENGTH_LONG).show();  
                }  
            });  
        dlg.setPositiveButton("닫기", null);  
        dlg.show();  
    }  
});
```



# Dialog 활용 2

## ■ Java 프로그램

```
button2.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        final boolean[] mSelect = {false, false, false, false};  
  
        new AlertDialog.Builder(MainActivity.this)  
            .setTitle("좋아하는 음식을 선택하십시오.")  
            .setIcon(R.drawable.ic_launcher)  
            .setMultiChoiceItems(R.array.foods, mSelect,  
                new DialogInterface.OnMultiChoiceClickListener() {  
                    public void onClick(DialogInterface dialog,  
                        int which, boolean isChecked) {  
                        mSelect[which] = isChecked;  
                    }  
                })  
    })
```



# Dialog 활용 2

## ■ Java 프로그램

```
.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 < mSelect.length; i++) {  
                if (mSelect[i]) {  
                    result += foods[i] + " ";  
                }  
            }  
        }  
    })
```



# Dialog 활용 2

## ■ Java 프로그램

```
        result = result.trim();
        result = result.replaceAll(" ", " ");
        if (result.equals(""))
            result = "없음";
        Toast.makeText(MainActivity.this,
"선택한 음식 = " + result, Toast.LENGTH_LONG).show();
    }
})
.setNegativeButton("취소", null)
.show();
}
});
```



# Dialog 활용 2

## ■ Java 프로그램

```
Button button3 = findViewById(R.id.button3);
button3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        new AlertDialog.Builder(MainActivity.this, R.style.MyAlertDialogStyle)
            .setTitle("공지 사항")
            .setMessage("이 메시지는 반드시 읽어야 합니다.")
            .setIcon(R.drawable.alarm)
            .setCancelable(false)
            .setNegativeButton("닫기", null)
            .show();
    }
});
```



# Dialog 활용 2

## ■ Java 프로그램

```
Button button4 = findViewById(R.id.button4);
button4.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        final ProgressDialog dialog = new ProgressDialog(MainActivity.this,
            AlertDialog.THEME_HOLO_DARK);

        dialog.setTitle("공지");
        dialog.setMessage("잠시만 기다려 주세요!!\n 프로그램 로딩중입니다.\n 중지하시겠습니까?");

        dialog.setCancelable(false);
```





# Dialog 활용 2

## ■ Java 프로그램

```
dialog.setButton("Yes", new DialogInterface.OnClickListener() {  
    @Override  
    public void onClick(DialogInterface dialogInterface, int i) {  
        dialog.dismiss();  
        Toast.makeText(getApplicationContext(), "중지하였습니다",  
            Toast.LENGTH_SHORT).show();  
    }  
});  
dialog.show();  
}  
});  
}  
}
```



# Dialog 활용 2

## ■ style

```
<style name="MyAlertDialogStyle"
        parent="@style/Theme.AppCompat.Dialog.Alert">
    <item name="android:textColor">@color/white</item>
    <item name="android:typeface">monospace</item>
    <item name="android:background">@color/colorPrimary</item>
</style>
```



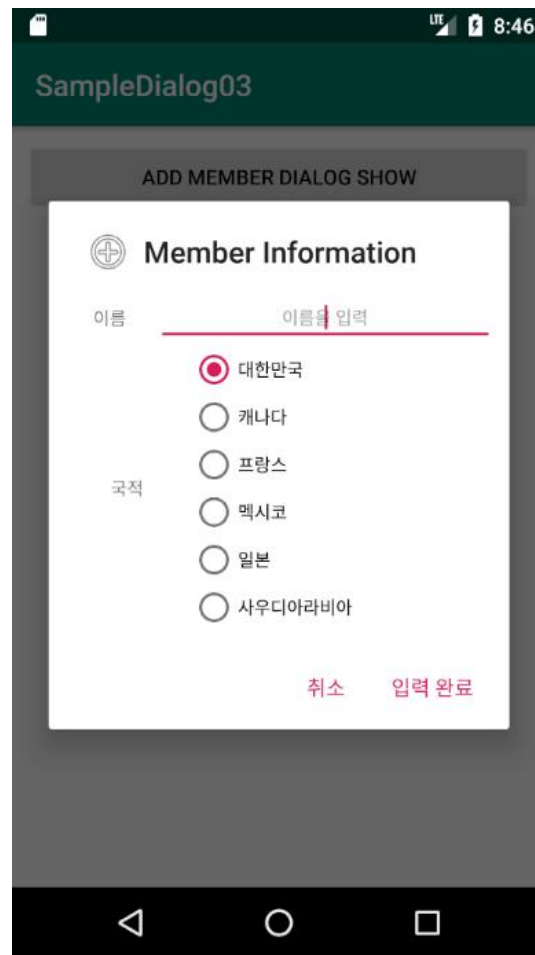
# Dialog 활용 3(회원가입)

- 회원 명단 추가를 위해 이름과 국적을 입력할 수 있는 Dialog를 만들어 보여주는 프로그램을 만들어보자
  - '이름'은 EditText를 이용하여 입력 받고, '국적'은 RadioButton을 통해 선택하도록 하여라
  - 버튼을 누르면 멤버 정보를 입력할 수 있는 Dialog를 보여주고, 입력을 완료하면 메인 화면의 TextView에 추가된 멤버들의 정보를 보여준다



# Dialog 활용 3(회원가입)

## ■ 실행화면





# Dialog 활용 3(회원가입)

## ■ 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="10dp"
    tools:context=".MainActivity">

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:onClick="OnClick"
        android:text="Add Member Dialog show" />
```



# Dialog 활용 3(회원가입)

## ■ 사용자 인터페이스

```
<TextView  
    android:id="@+id/text"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:padding="5dp"  
    android:textSize="18dp"  
    android:textStyle="bold" />
```

```
</LinearLayout>
```



# Dialog 활용 3(회원가입)



## ■ dialog\_addmember.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="250dp"
    android:layout_height="wrap_content"
    android:orientation="vertical"
    android:padding="10dp">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:weightSum="5">
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="match_parent"
            android:layout_weight="1"
            android:text="이름"
            android:textSize="12dp"
            android:gravity="center"/>
    </LinearLayout>
</LinearLayout>
```



# Dialog 활용 3(회원가입)



## ■ dialog\_addmember.xml

```
<EditText
    android:id="@+id/dialog_edit"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="4"
    android:hint="이름을 입력"
    android:textSize="12dp"
    android:gravity="center"/>
```

```
</LinearLayout>
```

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:weightSum="6">
```





# Dialog 활용 3(회원가입)

## ■ dialog\_addmember.xml

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="2"
    android:text="국적"
    android:textSize="12dp"
    android:gravity="center"/>
```

```
<RadioGroup
    android:id="@+id/dialog_rg"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_weight="4"
    android:orientation="vertical">
```



# Dialog 활용 3(회원가입)



## ■ dialog\_addmember.xml

```
<RadioButton
    android:id="@+id/dialog_rb_korea"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="대한민국"
    android:textSize="12dp"
    android:checked="true"/>
<RadioButton
    android:id="@+id/dialog_rb_canada"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="캐나다"
    android:textSize="12dp"/>
<RadioButton
    android:id="@+id/dialog_rb_france"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="프랑스"
    android:textSize="12dp"/>
```



# Dialog 활용 3(회원가입)

## ■ dialog\_addmember.xml

```
<RadioButton  
    android:id="@+id/dialog_rb_mexico"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="멕시코"  
    android:textSize="12dp"/>
```

```
<RadioButton  
    android:id="@+id/dialog_rb_poland"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="일본"  
    android:textSize="12dp"/>
```



# Dialog 활용 3(회원가입)

## ■ dialog\_addmember.xml

```
<RadioButton
    android:id="@+id/dialog_rb_saudi"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="사우디아라비아"
    android:textSize="12dp"/>
</RadioGroup>
</LinearLayout>
</LinearLayout>
```



# Dialog 활용 3(회원가입)

## ■ MainActivity.JAVA

```
public class MainActivity extends AppCompatActivity {  
    TextView textView1;  
    String str = "";  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        textView1 = findViewById(R.id.text);  
    }  
}
```



# Dialog 활용 3(회원가입)

## ■ MainActivity.JAVA

```
public void onClick(View view) {  
    LayoutInflater inflater = getLayoutInflater();  
    final View dialogView = inflater.inflate(R.layout.dialog_addmember, null);  
    AlertDialog.Builder buider = new AlertDialog.Builder(this);  
    buider.setTitle("Member Information");  
    buider.setIcon(android.R.drawable.ic_menu_add);  
    buider.setView(dialogView);  
    buider.setPositiveButton("입력 완료", new DialogInterface.OnClickListener() {  
        String name = "";  
        int checkedId = 0;  
        String nation = "";
```



# Dialog 활용 3(회원가입)

## ■ MainActivity.JAVA

```
@Override
public void onClick(DialogInterface dialog, int which) {
    EditText edit_name = dialogView.findViewById(R.id.dialog_edit);
    RadioGroup rg = dialogView.findViewById(R.id.dialog_rg);
    name = edit_name.getText().toString();
    checkedId = rg.getCheckedRadioButtonId();
    if (name.equals("")) {
        Toast.makeText(MainActivity.this, "새로운 멤버를 입력해주세요",
            Toast.LENGTH_SHORT).show();
    } else {
        RadioButton rb = rg.findViewById(checkedId);
        nation = rb.getText().toString();
        str += String.format("%-12.12s %20.20s\n", name, nation);
        textView1.setText(str);
        Toast.makeText(MainActivity.this, "새로운 멤버를 추가했습니다",
            Toast.LENGTH_SHORT).show();
    }
}
});
```



# Dialog 활용 3(회원가입)

## ■ MainActivity.JAVA

```
buidner.setNegativeButton("취소", new DialogInterface.OnClickListener() {  
    @Override  
    public void onClick(DialogInterface dialog, int which) {  
        Toast.makeText(MainActivity.this, "멤버 추가를 취소합니다",  
            Toast.LENGTH_SHORT).show();  
    }  
});  
  
AlertDialog dialog = buider.create();  
dialog.setCanceledOnTouchOutside(false);  
dialog.setCancelable(false);  
dialog.show();  
}  
}
```