

알림 (Notification)

알림(Notification)

- 알림을 서비스하는 매니저 객체 얻기

```
NotificationManager manager =  
    (NotificationManager)getSystemService(Context.NOTIFICATION_SERVICE);
```

- 알림을 생성하기 위한 NotificationCompat.Builder 객체를 생성하여 얻기

```
NotificationCompat.Builder builder =  
    new NotificationCompat.Builder(getApplicationContext());
```

알림(Notification)

- Builder에 제목, 내용, 이미지를 설정

```
builder.setLargeIcon(BitmapFactory.decodeResource(getResources(),  
                                android.R.drawable.ic_input_add));  
builder.setSmallIcon(android.R.drawable.ic_input_delete);  
builder.setTitle("내용타이틀");  
builder.setContentText("내용입니다");
```

- NotificationManager에 Notification을 등록하여 알림 창에 알림을 나타낸다.

```
Builder 객체의 build() 메서드를 호출하여 Notification 객체를 얻는다.  
manager.notify(1, builder.build());
```

알림(Notification)

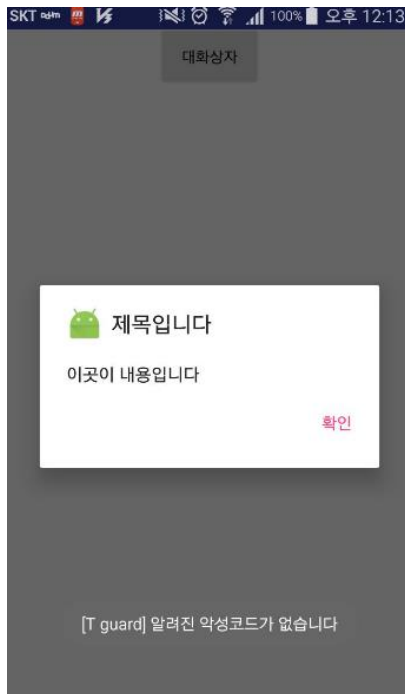
➤ 알림을 클릭 시 특정 액티비티 호출하기

```
// context, requestCode, Intent, Flag
PendingIntent pendingIntent = PendingIntent.getActivity(
    getApplicationContext(),
    0,
    new Intent(getApplicationContext(), AlarmActivity.class),
    PendingIntent.FLAG_UPDATE_CURRENT
);
builder.setContentIntent( pendingIntent);
builder.setAutoCancel( true ); //알림창에서 알림 선택 시 알림 아이콘삭제
// 소리 + 진동 발생
builder.setDefaults(
    Notification.DEFAULT_SOUND | Notification.DEFAULT_VIBRATE);
```

대화상자(AlertDialog)

대화상자(AlertDialog)

- 사용자에게 전달 사항을 알리고 질문을 통해 사용자의 선택을 받아들이는 가장 기본적인 통신 수단
- 배경에 액티비티를 그대로 유지한 채 중앙에 팝업으로 열리므로 메시지를 보여주거나 간단한 질문용으로 적합
- **Toast**도 사용자에게 보고하는 간편한 방법을 제공하나, 단순히 사실을 알리기만 할 뿐 질문을 할 수 없어 범용성이 떨어짐



대화상자(AlertDialog)

- **Dialog** 클래스를 좀 더 쓰기 편하게 고수준으로 래핑
- 문자열 메시지 뿐만 아니라 타이틀 바나 아이콘으로 출력 가능하며 버튼을 통해 사용자의 입력을 받아들일 수 있음
- 생성자가 **protected**로 숨겨져 있어 직접적으로 생성할 수 없으며, 내부 클래스인 **Builder**를 통해 생성해야 함

AlertDialog.Builder (Context context)

- 대화상자 객체 얻기

AlertDialog.Builder setMessage (CharSequence message)

- 내용 입력

AlertDialog.Builder setTitle (CharSequence title)

- 제목 설정

AlertDialog.Builder setIcon (int iconId)

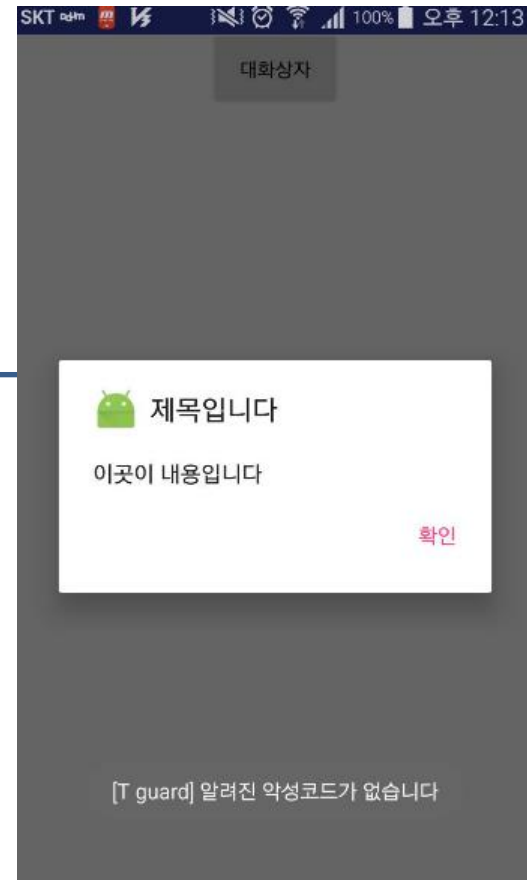
- 아이콘 설정

AlertDialog show ()

- 화면 출력

대화상자(AlertDialog)

```
AlertDialog.Builder dlg = new AlertDialog.Builder(MainActivity.this);  
dlg.setTitle("제목입니다");  
dlg.setMessage("이곳이 내용입니다");  
dlg.setIcon(R.drawable.ic_launcher);  
dlg.setPositiveButton("확인", null);  
dlg.show();
```



대화상자 버튼 추가하기

- 가급적이면 대화상자 자체에 닫기 버튼을 두는 것이 좋고 사용하기도 편리함
- 긍정(**Positive**), 부정(**Negative**), 중립(**Neutral**)로 이름이 붙어 있으며 모두 배치 시 왼쪽에서 순서대로 배치됨

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

text

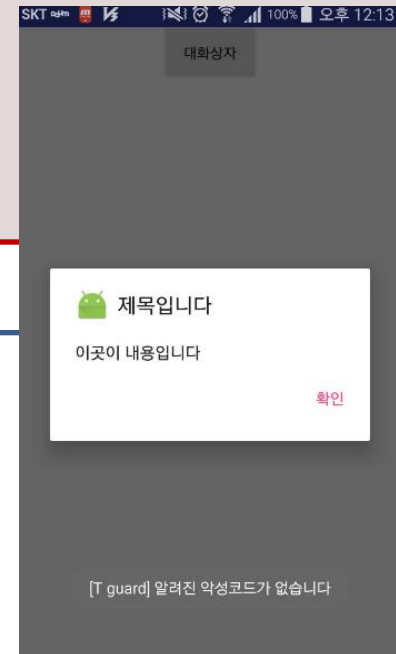
- 버튼에 표시할 텍스트

listener

- 버튼을 클릭할 때 호출될 콜백 함수

대화상자(AlertDialog)

```
AlertDialog.Builder dlg = new AlertDialog.Builder(MainActivity.this);
dlg.setPositiveButton("확인",
    new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog,
            int which) {
            // 버튼 클릭시 해야할 일 정의
        }
    }
);
dlg.show();
```



목록 아이템을 이용한 선택 및 입력

- 공간상의 제약으로 인해 대화상자는 버튼을 최대 3개까지만 배치 가능
- 더 많은 선택 사항 중 하나를 입력 받으려면 목록 대화상자를 사용
- 메시지 대신 선택 가능한 항목을 목록으로 보여주고 선택 받는 방식
- 목록 대화 상자 생성을 위해 빌더의 아래 메서드로 선택 가능한 목록을 전달

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

setItems (int itemId, DialogInterface.OnClickListener listener)

첫번째 매개변수

- 사용자에게 보여줄 목록, 문자열 배열 또는 배열 리소스의 ID 지정

listener

- 목록의 항목 중에서 하나를 클릭할 때의 처리를 지정

목록 선택

```
AlertDialog.Builder dlg = new AlertDialog.Builder(MainActivity.this);  
final String[] versionArray = new String[] { "젤리빈", "킷캣", "롤리팝" };  
dlg.setItems(  
    versionArray,  
    new DialogInterface.OnClickListener() {  
        public void onClick(DialogInterface dialog,  
                               int which) {  
            button1.setText(versionArray[which]);  
        }  
    }  
);
```

