레이아웃에

android:onClick="addClick"

를 추가하면 버튼이 눌렸을 때 해당 함수를 수행하도록 한다.

EditText = 텍스트 창

Button = 버튼

Button button = (Button)findViewById(R.id.buttonid)로 레이아웃 상의 개체를 아이디로 가져올 수 있다.

텍스트 창으로부터 숫자 가져오기

Int n = Integer.parseInt(textid.getText().toString());

텍스트 창에 텍스트 설정

Textid.setText(내용)

알림창 띄우기

AlertDialog.Builder 객체

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

객체 생성 후 알림창 제목 지정

Builder.setTitle(“ ”);

제목 지정 + 내용

Builder.setTitle(“ ”).setText(“ ”);

알림창에 리스트 추가

builder.setItems(items, new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int i) {  
 Toast.*makeText*(getApplicationContext(),items[i], Toast.*LENGTH\_SHORT*).show();  
 }  
});

첫번째 인자로 리스트, 두번째 인자로 리스트 항목 클릭시 처리이다.

Int i부분에 자동으로 몇번째 항목을 선택했는지가 들어온다.

해당 리스트 내용을 Toast메시지로 출력한다.

Toast 메시지는 간단한 알림 메시지라고 볼 수 있다.

첫 인자로 context를 필요로 한다

Context는 어떤 Activity, application인지를 구별하는 정보로 getapplicationcontext() 는 application에 대한 context를 가져온다고 보면 된다.

Items[i]로 해당 리스트의 항목을 선택하고, Toast.LENGTH\_SHORT로 표시할 길이를 설정한다.

.show()로 출력한다

AlertDialog alertDialog = builder.create()로 알림창 구성을 마친 후 알림창 객체를 생성하고

AlertDialog.show(); 로 출력을 표시한다.

알림창에 버튼 추가

Builder에

.setNegativeButton("취소", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialog, int which) {  
 dialog.cancel();  
 }  
})

를 추가함으로서 알림창에 yes/no 버튼을 만들 수 있다.

Dialog.cancel()은 알림창을 끄는 것이고.

Finish()는 어플을 종료한다.