사용자 인터페이스(II)

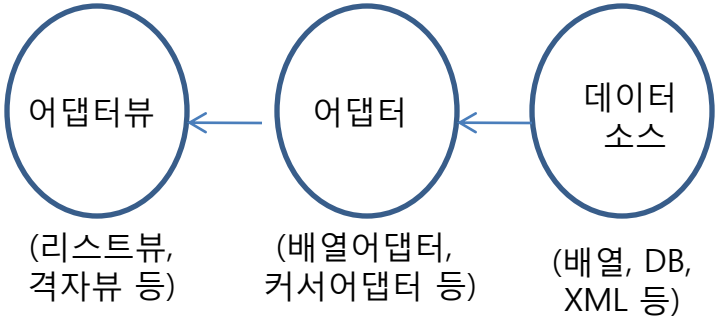
시작하면서

- **목차**
  - 리스트뷰: 어댑터, 어댑터뷰,
  - 대화창
  - 메뉴

리스트뷰-어댑터

Adapter

- 배열, XML 파일, 데이터베이스에서 다양한 종류의 값을 읽어오는 공통의 인터페이스
- 선택 위젯에게 데이터를 제공, 각 데이터 항목들을 위젯 내부에 표시하는 위젯 클래스로의 변환 수행
- 사용자 인터페이스인 뷰와 데이터 소스를 연결
  - Data Source: Array, XML, Database



Adapter

- **Adapter 종류**
  - ArrayAdapter: 배열
  - SimpleAdaper: XML
  - CursorAdapter: 데이터베이스, 컨텐트 제공자
- **ArrayAdapter**
  - 행 레이아웃 리소스 표준
    - simple_list_item_1(android.R.layout.simple_list_item_1)
      - 한 개의 텍스트 뷰로 구성
    - simple_list_item_2(android.R.layout.simple_list_item_2)
      - 두 개의 텍스트 뷰로 구성
  - 예) ArrayAdapter adapter1 = new ArrayAdapter(this, android.R.layout.simple_list_item_1, mStrings);
    - 컨텍스트에 액티비티 자신을 나타내는 this를 지정
    - 표현할 뷰의 리소스 ID로 simple_list_item_1를 지정
    - 데이터 소스로 배열 mStrings 지정

AdapterView와 리스트뷰

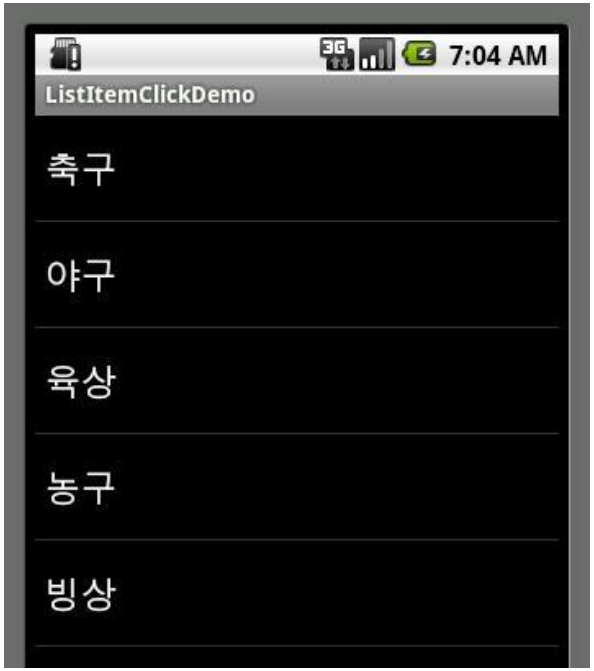
- 어댑터는 배열같은 외부 데이터와 어댑터 뷰 사이에 매개 역할
- 어댑터뷰는 어댑터가 넘겨주는 데이터를 화면상의 위젯에 표시
  - 레이아웃에 데이터를 채우는 역할
  - 사용자에 의한 항목 선택 이벤트 처리 역할
- 사용자는 컨테이너에 담긴 뷰 객체를 표준적인 방식에 따라 탐색
- 뷰 그룹 컨테이너: [표 6-2]
  - ListView, GridView, Gallery 등 AdapterView 파생 컨테이너
    - 같은 종류의 뷰를 반복 표시
  - ViewFlipper, ImageSwitcher, TextSwitcher 등 전환식 컨테이너
  - 뷰 객체들을 여러 탭으로 구조화하는 TabHost
  - 대화창

◉ 리스트뷰(ListView)

- 화면에 표시할 내용들이 리스트 형태로 되어 있는 경우 적합
- ListActivity 클래스:
  - 리스트뷰를 이용하는 액티비티를 생성
  - 데이터들을 리스트 형태로 쉽게 만듬
- 참조 클래스: android.widget.ListView

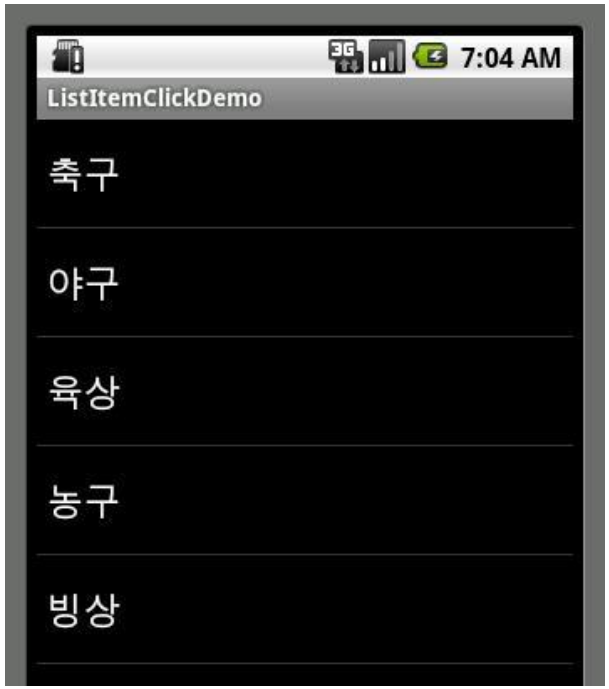
리스트뷰

- **<실습 6-1>**
  - ListItemClickDemo 프로젝트 생성
    - <코드 6-2>를 main.xml 복사
    - <코드 6-3>로 ListItemClickDemo.java 수정
    - [실행결과]



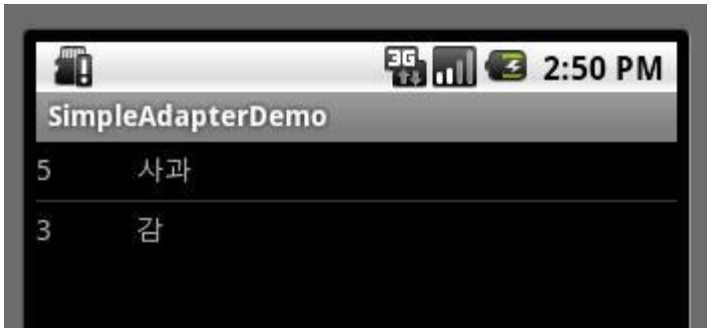
리스트뷰

- **<실습 6-2>**
  - ListViewClickDemo 프로젝트 생성
    - ListActivity 대신에 Activity에서 상속
    - <코드 6-2>를 main.xml 복사
    - <코드 6-4>로 ListViewClickDemo.java 수정
    - [실행결과]



리스트뷰

- **<실습 6-3> 어댑터 활용(p.201)**
  - Data source: HashMap으로 이루어지는 ArrayList
  - ArrayList: 리스트 특성을 갖는 동적 배열
  - HashMap: <키, 값>의 쌍으로 구성
  - SimpleAdapter: 정적 데이터를 리소스 파일에서 정의한 뷰로 사상하는 어댑터
  - ListView: 한 줄에 텍스트뷰 2개가 출력되는 리스트
- SimpleAdapterDemo 프로젝트 생성
  - <코드 6-5>를 main.xml 복사
  - <코드 6-7>~<코드 6-8>로 SimpleAdaperDemo.java에 복사
  - [실행결과]



대화창

대화창

- 사용자에게 확실하게 알리거나 반응을 받을 필요가 있을 때 사용

경고 대화창

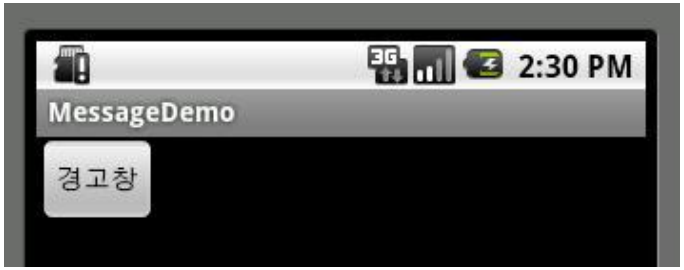
- 사용자의 반응이 있어야 다음 진행
- 윈도우의 모달 방식
- 최대 3개의 버튼 설정 가능
  - setPositiveButton(), setNeutralButton(), setNegativeButton()
- 대표적인 대화창: <표 6-2>

대화창

<실습 6-4> 대화창 생성

- 버튼 클릭시 경고 대화창 생성
- 경고 대화창내 2개 버튼 클릭시 경고 대화창 사라짐

- SimpleAdapterDemo 프로젝트 생성
  - <코드 6-9>를 main.xml 복사
  - <코드 6-10>~<코드 6-11>로 AlertDialogDemo.java에 복사
  - [실행결과]



날짜 선택창

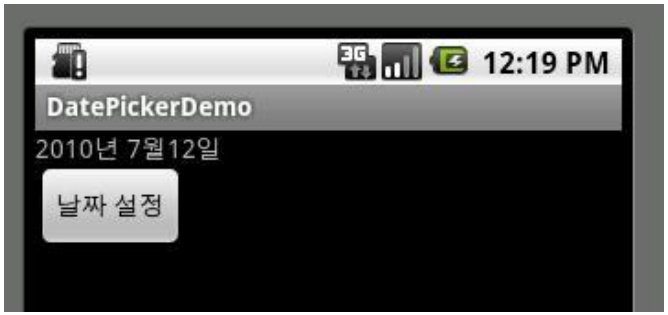
날짜 선택창

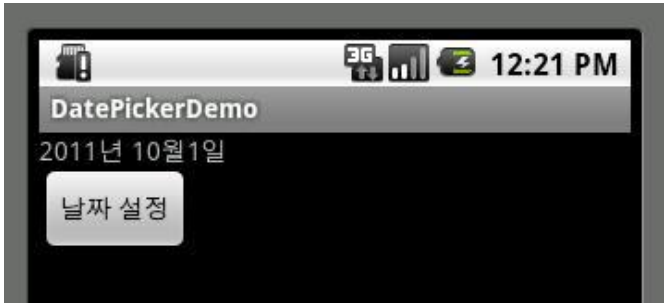
- 달력을 기준으로 연/월/일을 선택하는 DatePicker를 포함하는 간단한 대화창
- OnDateSetListener()를 설정
  - 사용자가 선택한 날짜 값을 받음
- 날짜 선택창에서 연, 월, 일 값을 선택

  → onDateSet() 메소드 호출
  - 이 메소드를 통해 사용자가 선택한 연,월, 일 값을 화면에 표시

날짜 선택창

- **<실습 6-5> 대화창 생성**
  - DatePickerDemo 프로젝트 생성
    - <코드 6-12>를 main.xml 복사
    - <코드 6-13>~<코드 6-18>을 DatePickerDemo.java에 복사
    - [실행결과]





메뉴

옵션 메뉴

- **메뉴 키 클릭시 화면 하단에 나타나는 메뉴**
  - 아이콘 메뉴: 메뉴 키를 눌렀을 때 화면 하단에 처음 나타나는 아이콘 메뉴 항목
  - 확장 메뉴(extended menu): 메뉴 항목이 6개 초과시 나머지 항목들을 표시하기 위해 사용
  - 부메뉴(submenu): 메뉴 항목이 또다른 메뉴 항목들로 구성
- **옵션 메뉴 생성**
  - onCreateOptionsMenu(Menu menu)
  - onOptionsItemSelected(MenuItem item)

옵션 메뉴

<실습 6-6> 옵션 메뉴 생성

- 메뉴 항목 7개 생성
  - 단축키 생성: setAlpabeticShortcut('a')
  - 아이콘 이미지 추가: setIcon(R.drawable.icon)
- OptionMenuDemo 프로젝트 생성
  - <코드 6-19>를 main.xml 복사
  - <코드 6-20>로 OptionMenuDemo.java에 복사
  - [실행결과]

컨텍스트 메뉴

- 화면의 특정 뷰를 길게 누를 때(long click) 나타나는 메뉴
- PC에서 오른쪽 마우스 클릭시 나타나는 메뉴에 대응

컨텍스트 메뉴 생성

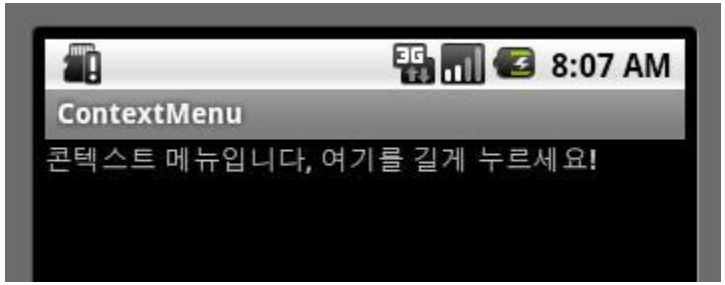
- onCreateContextMenu(ContextMenu menu, View view, ContextMenuInfo menuinfo)
- onContextItemSelected(MenuItem item)

특정 뷰와 컨텍스트 메뉴를 연결

- setOnCreateContextMenuListener(this)

컨텍스트 메뉴

- **<실습 6-7> 컨텍스트 메뉴 생성**
  - 메뉴 항목 7개 생성
  - ContextMenuDemo 프로젝트 생성
    - <코드 6-19>를 main.xml 복사
    - <코드 6-21>로 ContextMenuDemo.java에 복사
    - [실행결과]



메뉴 인플레이션

- 메뉴 구조를 프로그램이 아닌 외부 리소스 파일에서 정의
- 필요할 때 인플레이션시켜 메뉴 객체 생성
- 리소스 파일로 메뉴 구조를 정의: menutest.xml(<코드 6-22>)
  - <menu> …
    - <item>
      <menu> …
      </menu>
    - </item> …
  - </menu>
- MenuInflater 클래스: 메뉴를 정의하는 리소스 파일을 이용하여 메뉴 객체를 생성
  - OnCreateOptionsMenu(Menu menu)
    - MenuInflater inflater = getMenuInflater();
    - inflater.inflate(R.menu.menutest, menu);

메뉴 인플레이션

- **<실습 6-8> 메뉴 인플레이션**
  - 메뉴 항목 7개 생성
  - 항목 3에 뷰메뉴 추가
- MenuInflateDemo 프로젝트 생성
  - <코드 6-19>를 main.xml 복사
  - <코드 6-22>를 res/menu 폴더의 menu.xml에 복사
  - <코드 6-23>을 MenuInflateDemo.java에 복사
  - [실행결과]