06장. UI 화면 구성요소

01. 뷰, 위젯, 레이아웃
02. TextView
03. EditText와 Spinner
04. 버튼, 체크상자, 라디오 버튼
05. 날짜와 시간선택
06. 정보표시
07. 메뉴
08. 사용자 이벤트 처리
09. 스타일 다루기
10. 테마 다루기

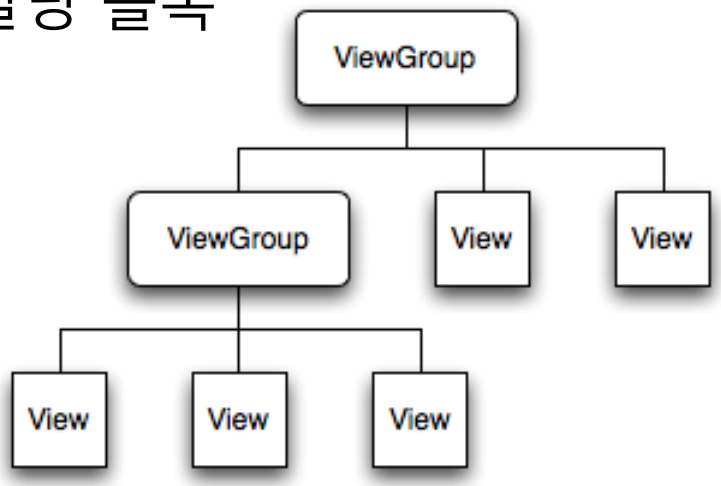
컴퓨터공학과 변영철 교수

구글의 음모(?)

http://www.youtube.com/watch?v=R7yfV6RzE30&feature=player_embedded

뷰, 위젯, 레이아웃

- 액티비티(Activity)
  - 하나의 (가상) 화면
  - 실제 한 화면이 구현되기 위해서는
    - 액티비티 - 윈도우 – 뷰 계층구조
- 뷰(View)
  - 화면에 사각형 영역을 점유하고 있는 UI 컴포넌트를 위한 빌딩 블록



뷰, 위젯, 레이아웃

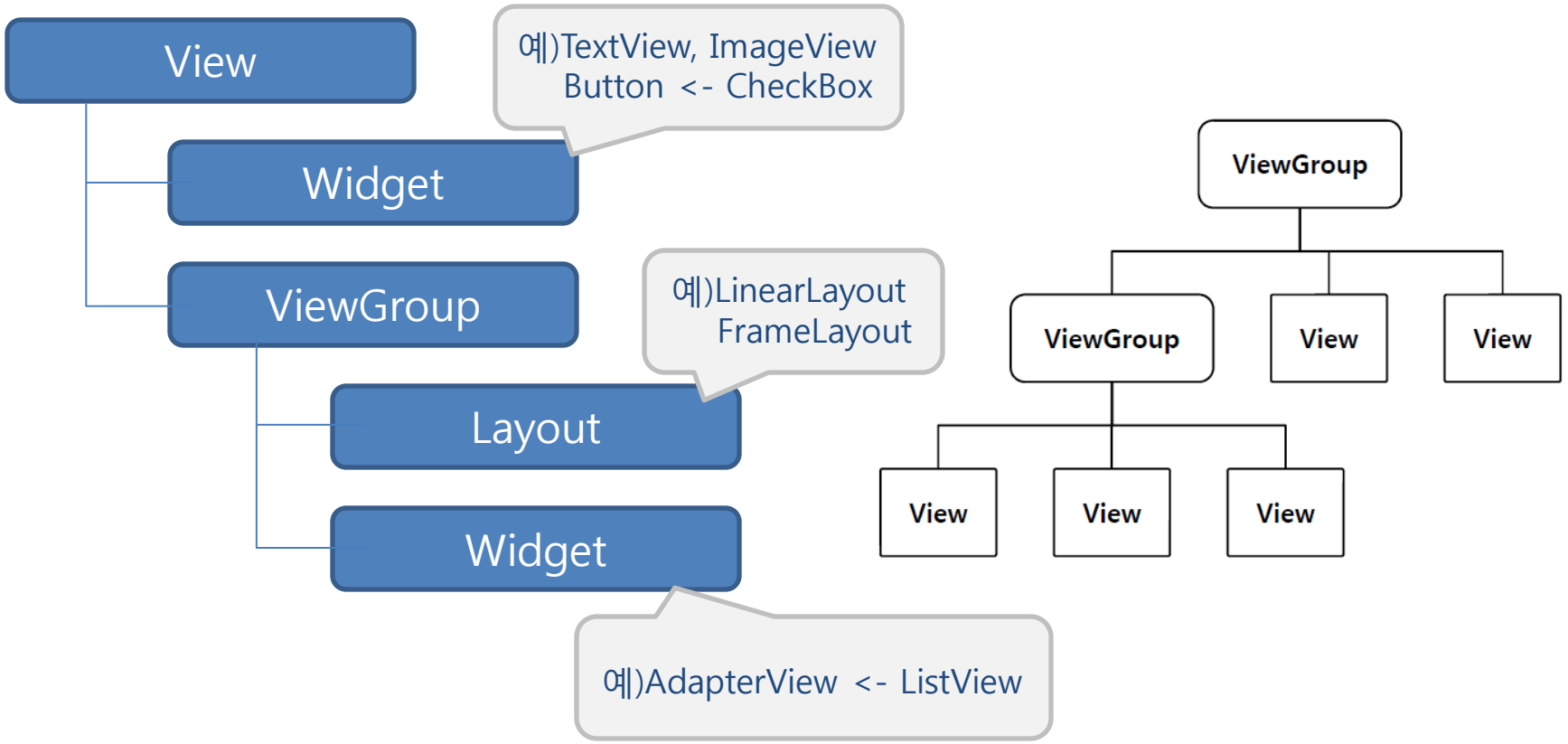
- 위젯(Widget 클래스)
  - 사용자와 상호작용하는 인터페이스 역할을 하는 뷰
  - 버튼(Button), 체크상자(CheckBox) 등
  - 따라서 위젯은 View 클래스의 자식 클래스

java.lang.Object
↳ android.view.View
↳ android.widget.TextView
↳ android.widget.Button

java.lang.Object
↳ android.view.View
↳ android.widget.TextView
↳ android.widget.Button
↳ android.widget.CompoundButton
↳ android.widget.CheckBox

- 뷰그룹(ViewGroup)
  - 여러 위젯(뷰)를 포함할 수 있도록 확장(extends)한 것
  - 따라서 뷰그룹도 뷰
  - 뷰그룹에서 상속받아 레이아웃 클래스를 선언함
  - 따라서 레이아웃 클래스도 뷰

뷰, 위젯, 레이아웃


View
예)TextView, ImageView
Button <- CheckBox
Widget
ViewGroup
예)LinearLayout
FrameLayout
Layout
Widget
예)AdapterView <- ListView
ViewGroup
ViewGroup
View
View
View
View
View

뷰, 위젯, 레이아웃

LinearLayout

RelativeLayout

TableLayout

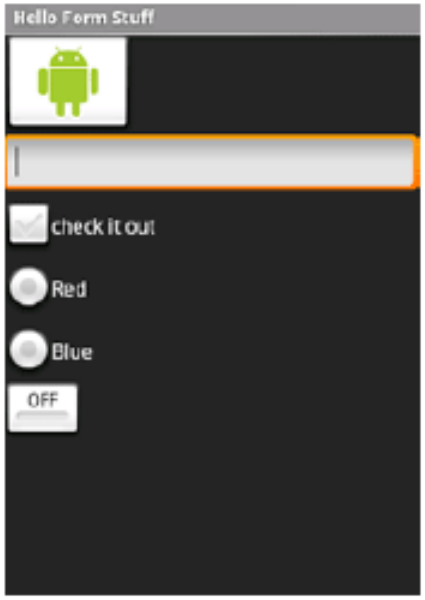
AbsoluteLayout : 위젯의 구체적 위치 지정

뷰, 위젯, 레이아웃

DatePicker

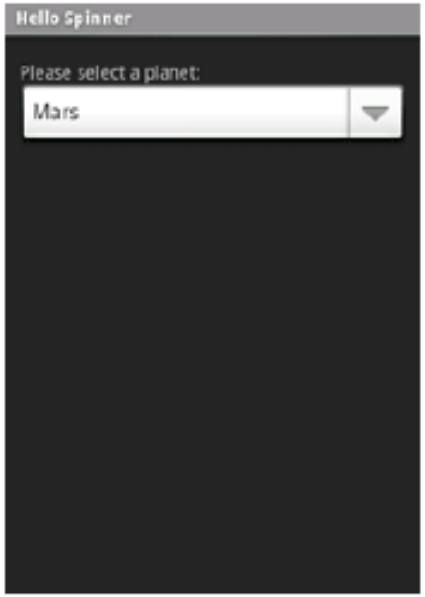
TimePicker

Form Stuff



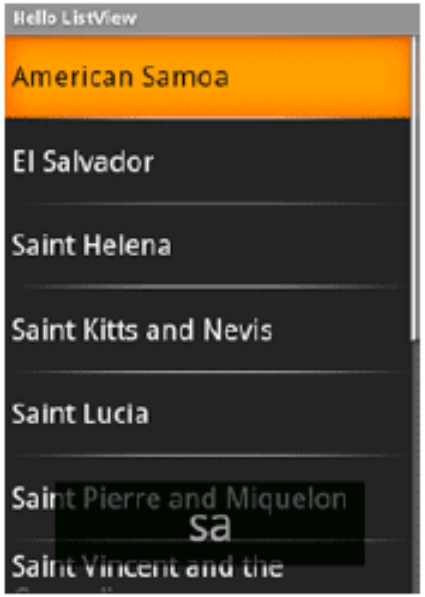
뷰, 위젯, 레이아웃

Spinner



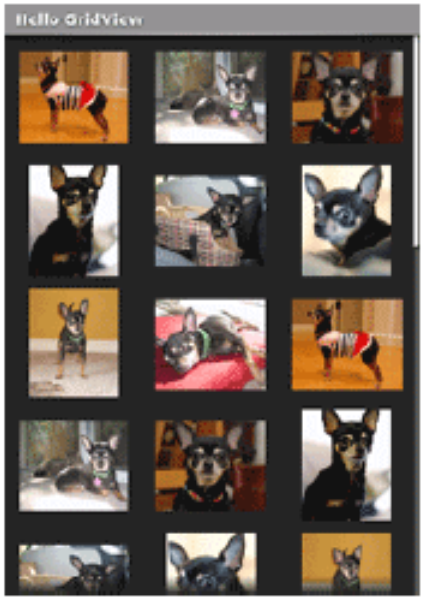
AutoComplete

ListView

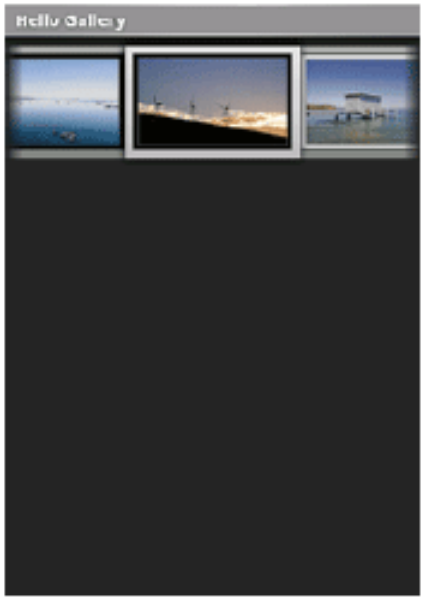


뷰, 위젯, 레이아웃

GridView



Gallery



TabWidget

뷰, 위젯, 레이아웃

MapView

WebView

TextView 위젯


7:12 AM
UI Elements
This is a test. My blog is at http://androidbook.blogspot.com
Email me at test@gmail.com
Call me at (800) 555-1212
Find me at my favorite coffee joint 1456 White Mountain Hwy, North Conway, NH 03860

This is a test. My blog is at http://androidbook.blogspot.com
Email me at test@gmail.com
Call me at (800) 555-1212
Find me at my favorite coffee joint 1456 White Mountain Hwy, North Conway, NH 03860

This is a test. My blog is at http://androidbook.blogspot.com
Email me at test@gmail.com
Call me at (800) 555-1212
Find me at my favorite coffee joint 1456 White Mountain Hwy, North Conway, NH 03860

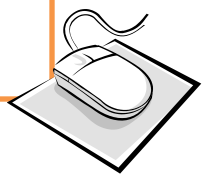
TextView 위젯

- 텍스트 지정 방식
  - 직접 지정
  - 간접 지정
  - 코드에서 액세스 : setText / getText
- 속성
  - lines : 표시할 줄 수 결정
  - ems : m과 같이(n이 아닌) 넓은 문자의 인쇄문자 너비 단위
  - ellipsize : 문자열이 길 경우 …로 표시

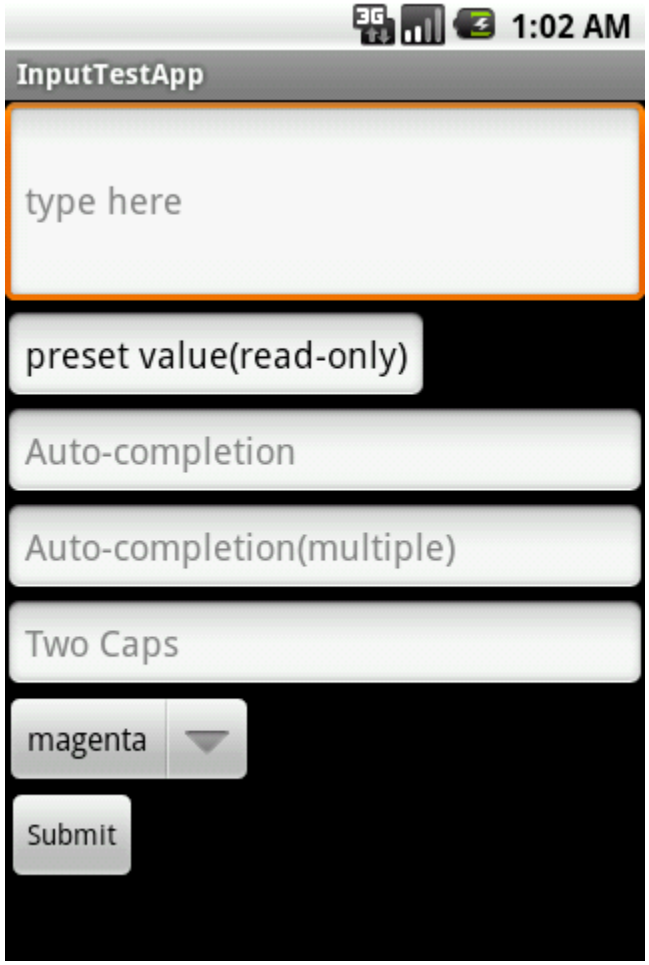
```
<TextView
android:id="@+id/TextView01"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Some sample text here" />
<TextView
android:id="@+id/TextView01"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/sample_text" />
```

```
<TextView
android:id="@+id/TextView03"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:lines="2"
android:ems="12"
android:ellipsize="marquee"
android:text="Android will be at the 2010 Game Developers
Conference in San Francisco from March 9th to March 11th." />
```

Views
(TextDisplay)



EditText 위젯


1:02 AM
InputTestApp
type here
preset value(read-only)
Auto-completion
Auto-completion(multiple)
Two Caps
magenta
Submit

EditText 위젯

- 한글 입력기 설치방법
  1. HangulKeyboard.apk 파일을 다운
  2. 안드로이드 SDK의 tools 경로 아래에 복사한다.
  3. 이클립스에서 에뮬레이터 구동
  4. 도스 창에서 아래 명령 실행

     adb install HangulKeyboard.apk

     -> /data/local 폴더에 설치됨. 언인스톨하려면

     adb uninstall com.socialnmobile.hangulkeyboard 실행
  5. 에뮬레이터에서 MENU 버튼 선택 > Settings > Language & keyboard 선택
  6. Android 키보드와 한글 키보드만 선택. 나머지는 선택 해제
  7. 홈 버튼을 클릭하여 메인으로 이동 후 구글 검색창을 클릭

EditText 위젯

- 사용자로부터 텍스트를 입력 받는데 사용
- 대부분의 기능들이 TextView에 이미 구현되어 있음
  - 단지 TextView에서는 그것들이 활성화되지 않았을 뿐
- 속성
  - hint
  - lines : 설정하지 않으면 자동으로 커지며, 설정하면 고정된 크기에서 내용이 많을 경우 스크롤되면서 입력
- 위젯에서 오랫동안 누르면 텍스트를 편집할 수 있는 문맥 메뉴가 나타남

```
<EditText
android:id="@+id/EditText01"
android:layout_height="wrap_content“
android:layout_width="fill_parent"
android:hint="type here"
android:lines="4“/>
```

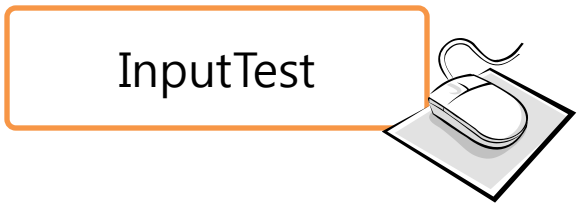
EditText 위젯

- 자동완성 기능
  - AutoCompleteTextView 위젯 : 개발자가 미리 정해둔 단어들에 부합하면 그 단어들이 제시
    - completionThreshold 속성값이 1이면 한 글자만 입력되어도 자동 완성 목록을 제시하라는 뜻
    - 위젯에 추천 단어를 가지고 있는 아답터 객체만 설정하면 됨 : text.setAdapter(adapter);
  - MultiAutoCompleteTextView 위젯 : 한 입력 상자 안에서 여러 개의 항목들에 대해 자동 완성 기능 제공
    - 한 문자열을 여러 개의 항목으로 구분하기 위한 Tokenizer 객체필요 -> Tokenizer는 쉼표로 항목들을 구분
    - MultiAutoCompleteTextView.CommaTokenizer
    - 다중 단어 자동 완성 기능은 블로그에서 여러 개의 태그들을 입력할 때 유용
    - AutoCompleteTextView와 기본 사용법은 동일하며, Tokenizer만 추가로 설정 mtext.setTokenizer(new MultiAutoCompleteTextView.CommaTokenizer());

EditText 위젯

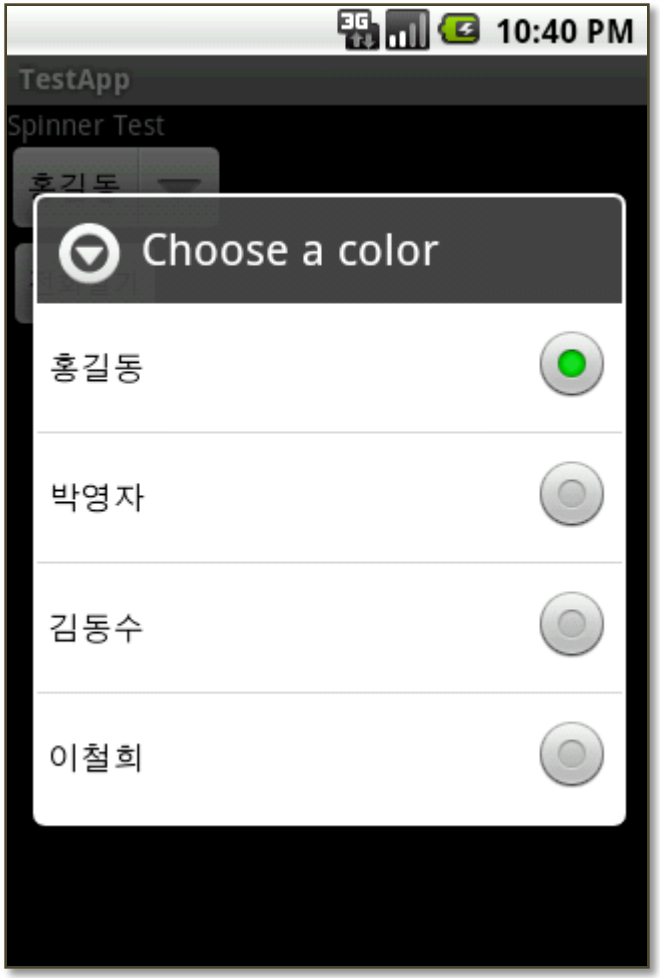
- 입력 제한
  - 2개의 대문자만 입력
  - 내장 필터 이용

```
final EditText text_filtered = (EditText) findViewById(R.id.input_filtered);

text_filtered.setFilters(new InputFilter[] {
      new InputFilter.AllCaps(),
      new InputFilter.LengthFilter(2)
});
```



Spinner 위젯


10:39 PM
TestApp
Spinner Test
홍길동
전화걸기


10:40 PM
TestApp
Spinner Test
Choose a color
홍길동
박영자
김동수
이철희

Spinner 위젯

- 사용자가 직접 입력하게 하는 대신 미리 정해진 것들 중 하나를 선택
- 속성
  - entries
  - prompt
- 사용 방법

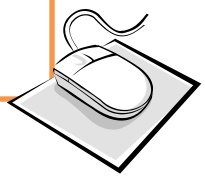
  [1]레이아웃 디자인 :<Spinner /> 앨리먼트 작성

  [2]res/values/strings.xml에 표시할 문자열 배열(colors) 작성

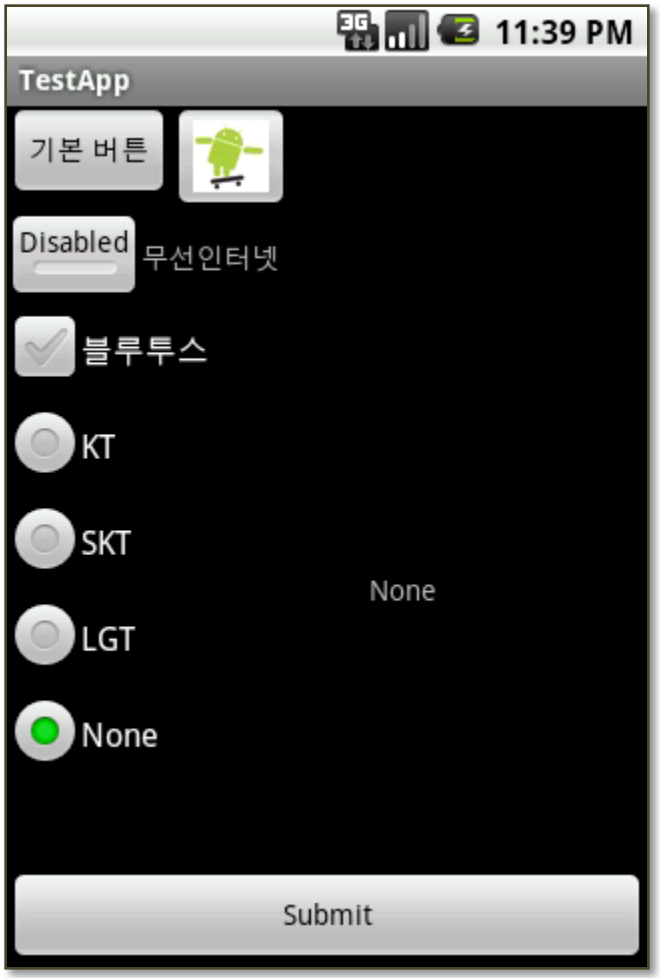
  [3]colors 배열을 <Spinner /> 앨리먼트에 연결

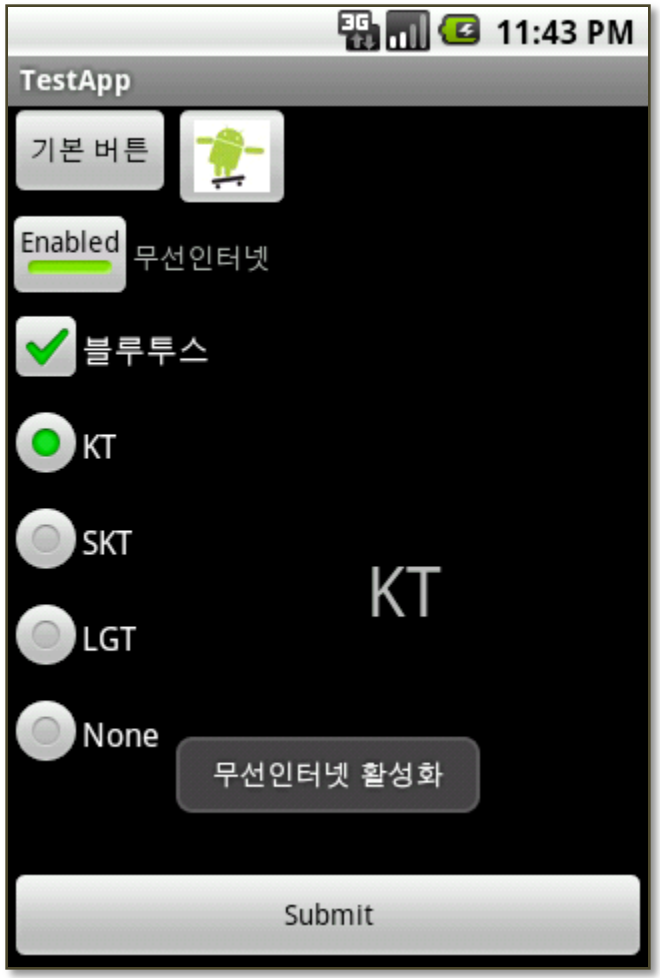
  [4]선택한 문자열 액세스

SpinnerTest



버튼, 체크상자, 라디오 버튼 위젯





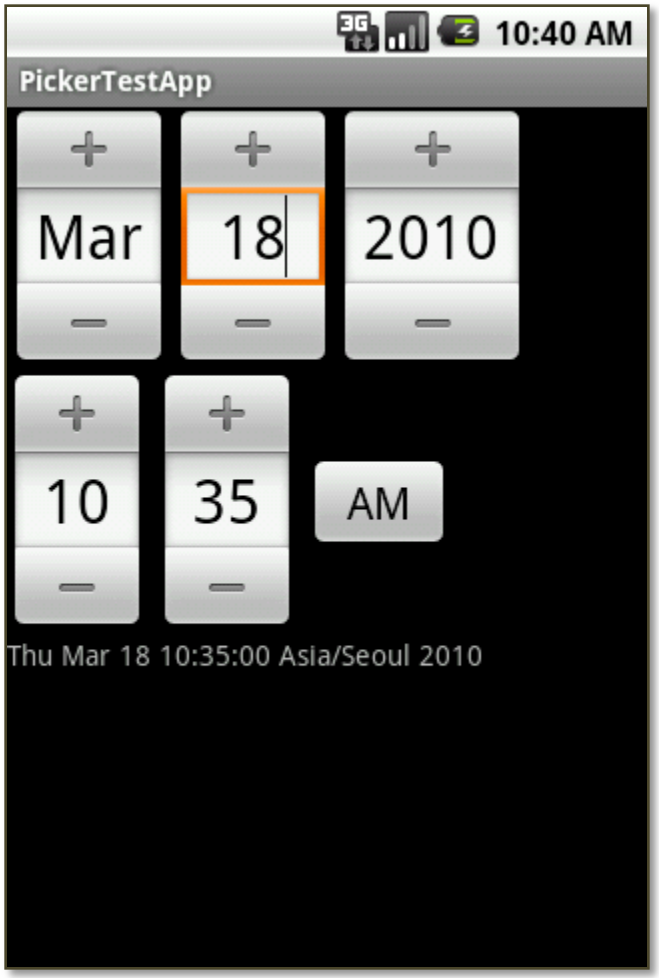
버튼, 체크상자, 라디오 버튼 위젯

- 버튼의 유형
  - Button, ImageButton
  - ToggleButton : 두 가지 상태 (texton, textoff 속성)
  - CheckBox : 두 가지 상태
  - RadioButton (RadioGroup과 함께 사용)
- 사용방법

  [1]레이아웃 디자인 (layout/button_test.xml)

  [2]이벤트 처리 : 각 버튼에 대하여 OnClickListener 작성

날짜와 시간 선택 위젯


10:40 AM
PickerTestApp
Mar
18
2010
10
35
AM
Thu Mar 18 10:35:00 Asia/Seoul 2010

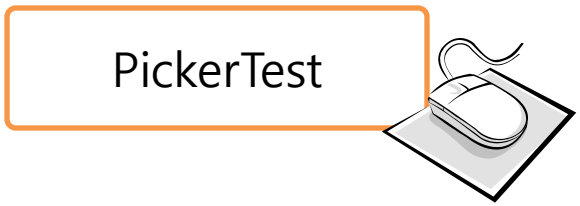
날짜와 시간 선택 위젯

- 날짜, 시간 선택 위젯
  - DatePicker, TimePicker
- 사용 방법

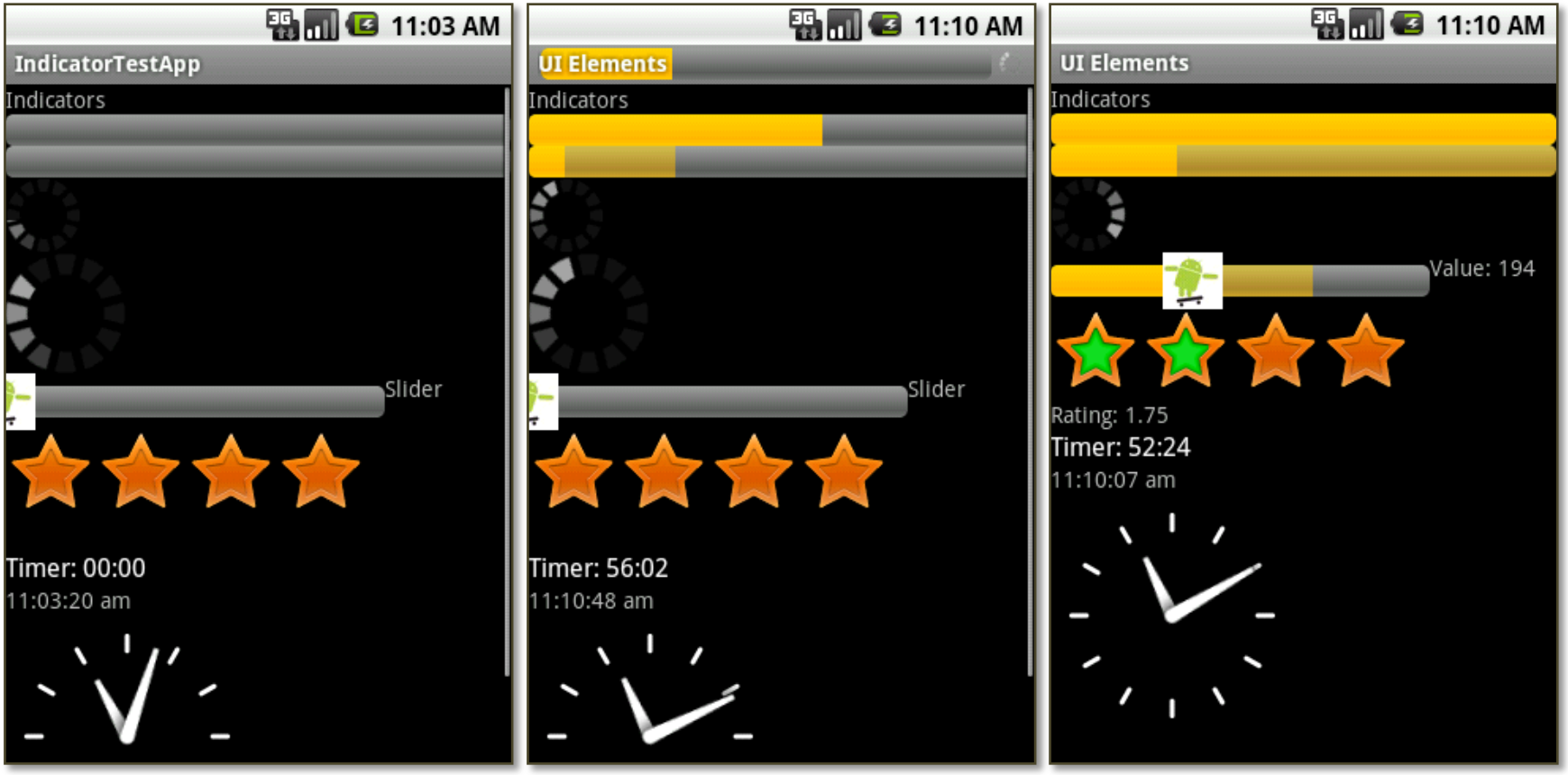
  [1]레이아웃 디자인 (layout/picker_test.xml)

  [2]날짜와 시간을 변경할 경우 실행되는 리스너 등록
    - DatePicker 리스너(OnDateChangedListener) 등록 -> 리스너 객체를 생성한 후 init 메소드로 등록
    - TimePicker 리스너(OnTimeChangedListener) 등록 -> setOnTimeChangedListener 메소드로 등록

  [참고]애뮬레이터 시간 설정->한국



정보 표시



디자인만 한 모습

정보 표시

- 다소 시간이 걸리는 일을 수행해야 하는 경우
  - 얼마나 오래 걸릴 지 알 때
  - 얼마나 오래 걸릴지 알 수 없는 경우 : indeterminate 진행 표시기 (둥근 모양)

ProgressBar 위젯

- 기본 ProgressBar
  - 중간 크기의 둥근 indeterminate 표시기

```
<ProgressBar
android:id="@+id/progress_bar4"
android:layout_width="wrap_content"
android:layout_height="wrap_content" />
```

  - 큰 크기의 표시기 -> 안드로이드 내장 스타일

```
style="?android:attr/progressBarStyleLarge"
```

- 수평 진행 표시기

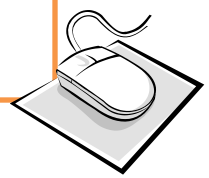
```
<ProgressBar
android:id="@+id/progress_bar"
style="?android:attr/progressBarStyleHorizontal"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:max="100" />
```

ProgressBar 위젯

- 액티비티 타이틀 바를 진행 표시기로 설정
  - 화면 공간을 절약하고자 할 때
  - 일반적인 바와 오른쪽 작은 원
  - 최대값 : 10,000 (100이 아님)

```
requestWindowFeature(Window.FEATURE_INDETERMINATE_PROGRESS);
setContentView(R.layout.indicators);
requestWindowFeature(Window.FEATURE_PROGRESS);
```

IndicatorTest



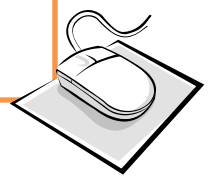
SeekBar 위젯

- 사용자가 진행 위치를 직접 지정
  - 동영상 재생에서 사용자가 재생 위치 지정할 때
  - 엄지(thumb) -> 다른 drawable 이미지(jpg등) 표시가능

```
<SeekBar
android:id="@+id/seekbar1"
android:layout_height="wrap_content"
android:layout_width="240px"
android:max="512"
android:thumb="@drawable/droid" />
```

- 위치를 변경하였을 경우 실행되는 리스너 등록
  - OnSeekBarChangeListener

IndicatorTest



RatingBar 위젯

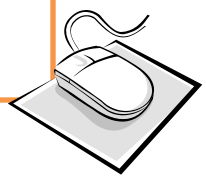
- 영화평에서 흔히 볼 수 있는 "별점" 모양의 위젯

```
<RatingBar
android:id="@+id/ratebar1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:numStars="4"
android:stepSize="0.25" />
```

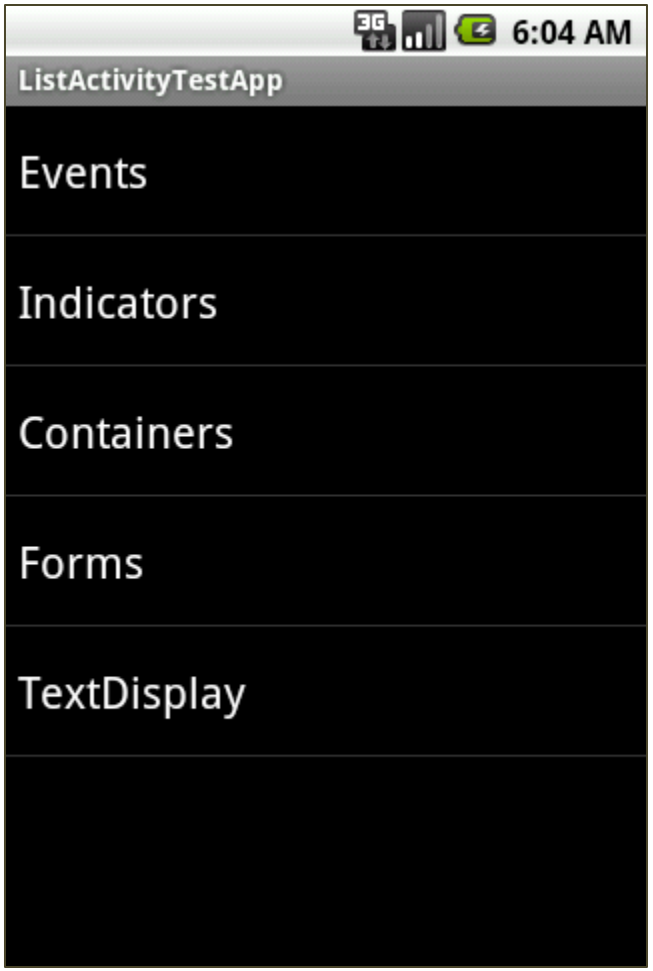
- 값을 변경하였을 경우 실행되는 리스너 등록
  - OnRatingBarChangeListener

```
RatingBar rate = (RatingBar) findViewById(R.id.ratebar1);

rate.setOnRatingBarChangeListener(new RatingBar.OnRatingBarChangeListener() {
    public void onRatingChanged(RatingBar ratingBar, float rating,
        boolean fromTouch) {
            ((TextView)findViewById(R.id.rating_text)).setText("Rating: "+ rating);
        }});
```

IndicatorTest



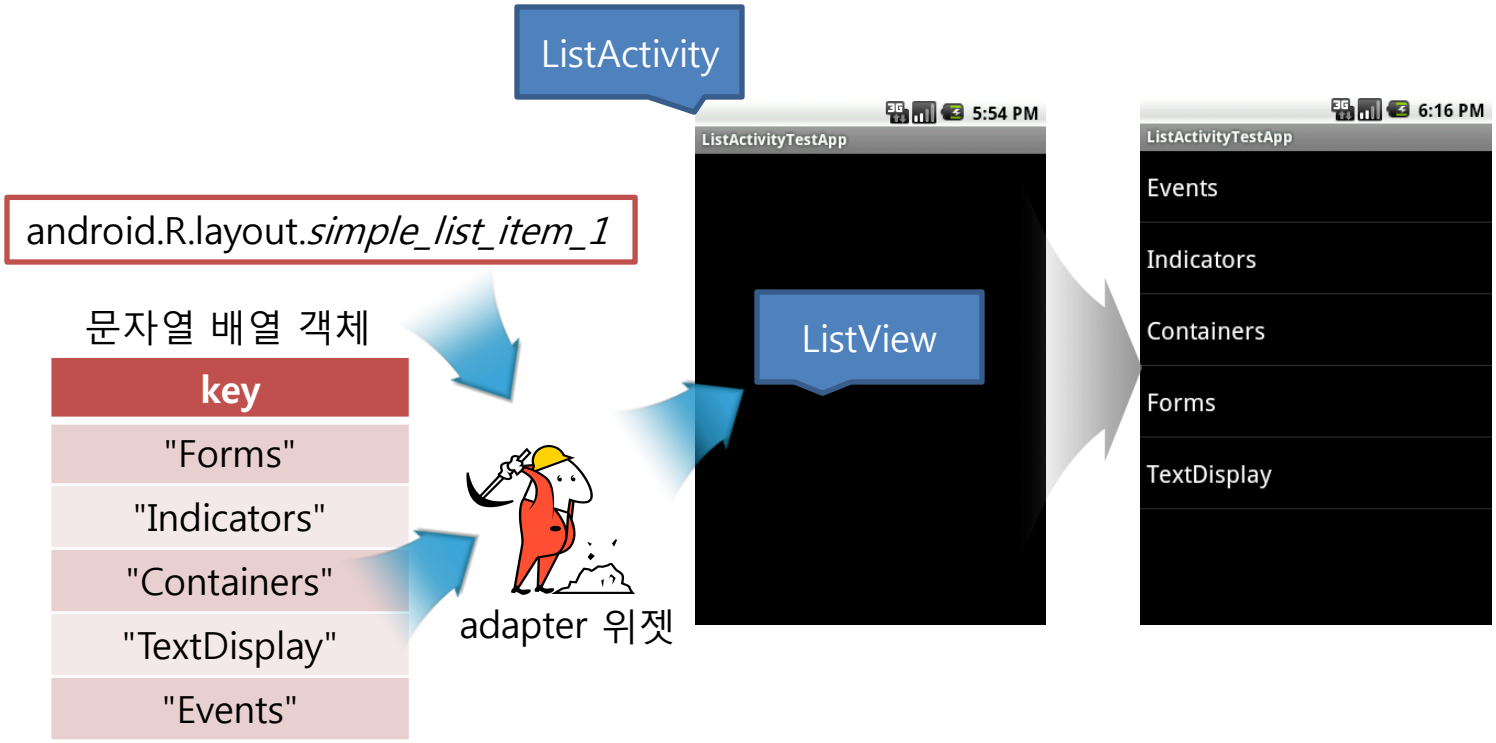
리스트 메뉴


6:04 AM
ListActivityTestApp
Events
Indicators
Containers
Forms
TextDisplay

리스트 메뉴

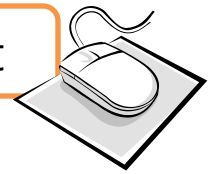
- ListActivity 액티비티
  - 아이템들을 수직으로 표시하는 ListView를 내부에 가지며, 따라서 레이아웃 파일에 ListView 디자인할 필요 없음
  - 따라서 ListActivity는 아이템 리스트를 표시할 수 있고, 아이템을 선택할 경우 이벤트를 발생할 수 있는 액티비티
  - setListAdapter 메소드를 이용하여 표시할 아이템들을 갖는 ListAdapter를 ListActivity에 설정
  - ListAdapter는 Adapter 인터페이스를 상속받는 인터페이스
  - ListAdapter를 구현하는 아답터 클래스들 : ArrayAdapter<T>, SimpleAdapter, CursorAdapter, SimpleCursorAdapter 등이 있음

리스트 메뉴

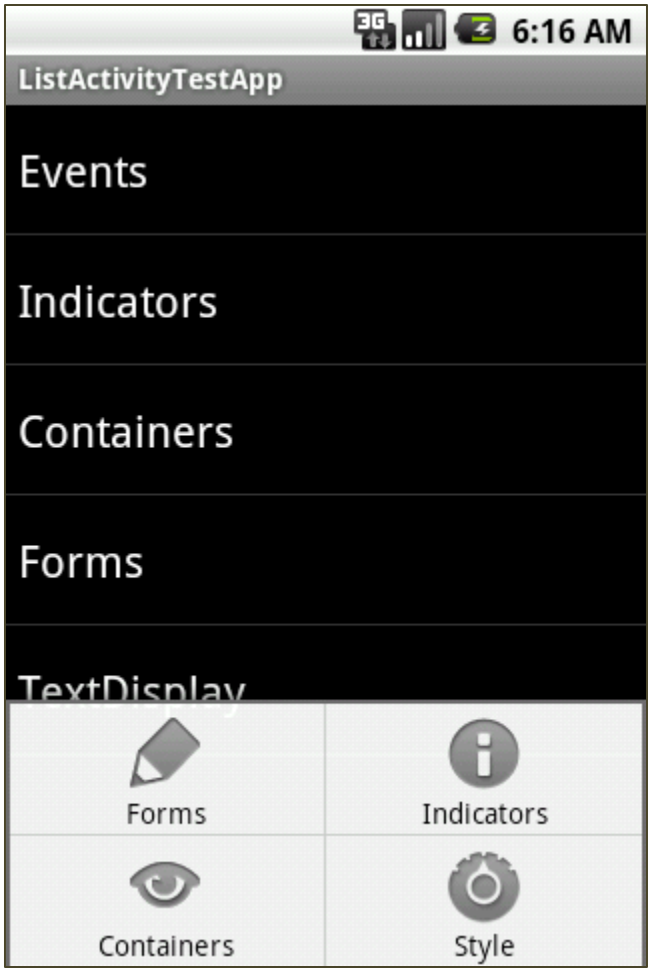



```
ArrayAdapter<String> adapter =
      new ArrayAdapter<String>(this, android.R.layout.simple_list_item_1, keys);

setListAdapter(adapter);
```

ListActivityTest



옵션 메뉴



android.R.drawable.*ic_menu_edit*

android.R.drawable.*ic_menu_info_details*

android.R.drawable.*ic_menu_view*

android.R.drawable.*ic_menu_preferences*

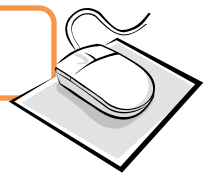
옵션 메뉴

- 응용 실행 도중 메뉴 버튼(“Menu”)을 눌르면 나타나는 응용 고유의 메뉴
- 응용에 대한 추가적인 제어나 기타 설정 수단으로 쓰임
- 옵션 메뉴 생성
  - Activity 클래스의 onCreateOptionsMenu 메소드 재정의
  - onCreateOptionsMenu 파라미터 menu에 원하는 만큼의 옵션 메뉴(메뉴 이름, 아이콘, 인텐트)를 추가
- 메뉴 선택 시 실행
  - 각 메뉴 항목에 이미 인텐트를 지정하였으므로 자동으로 실행됨
  - 추가적으로 수행하고자 하는 코드는 onOptionsItemSelected 메소드 재정의

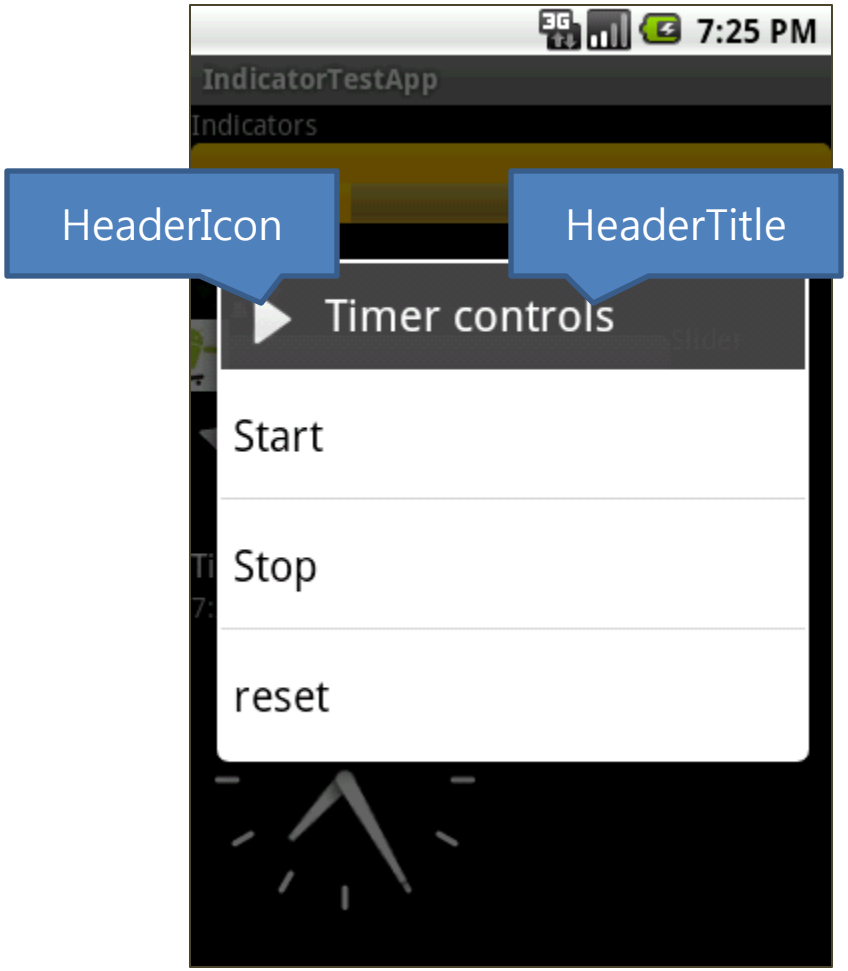
옵션 메뉴

```
@Override
public boolean onCreateOptionsMenu(android.view.Menu menu) {
    super.onCreateOptionsMenu(menu);

    menu.add("Forms")
        .setIcon(android.R.drawable.ic_menu_edit)
        .setIntent(new Intent(this, AActivity.class));
    menu.add("Indicators")
        .setIntent(new Intent(this, BActivity.class))
        .setIcon(android.R.drawable.ic_menu_info_details);
    menu.add("Containers")
        .setIcon(android.R.drawable.ic_menu_view)
        .setIntent(new Intent(this, CActivity.class.class));

    SubMenu sub_menu =
    menu.addSubMenu("Style").setIcon(android.R.drawable.ic_menu_preferences);

    sub_menu.add(1, 1, 1, "Light").setChecked(isLight);
    sub_menu.add(1, 2, 2, "Dark").setChecked(!isLight);
    sub_menu.setGroupCheckable(1, true, true);

    return true;
}

private boolean isLight = true;
```

ListActivityTest



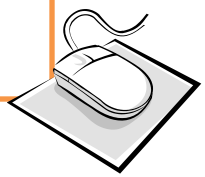
컨텍스트 메뉴


7:25 PM
IndicatorTestApp
Indicators
HeaderIcon
HeaderTitle
Timer controls
Start
Stop
reset

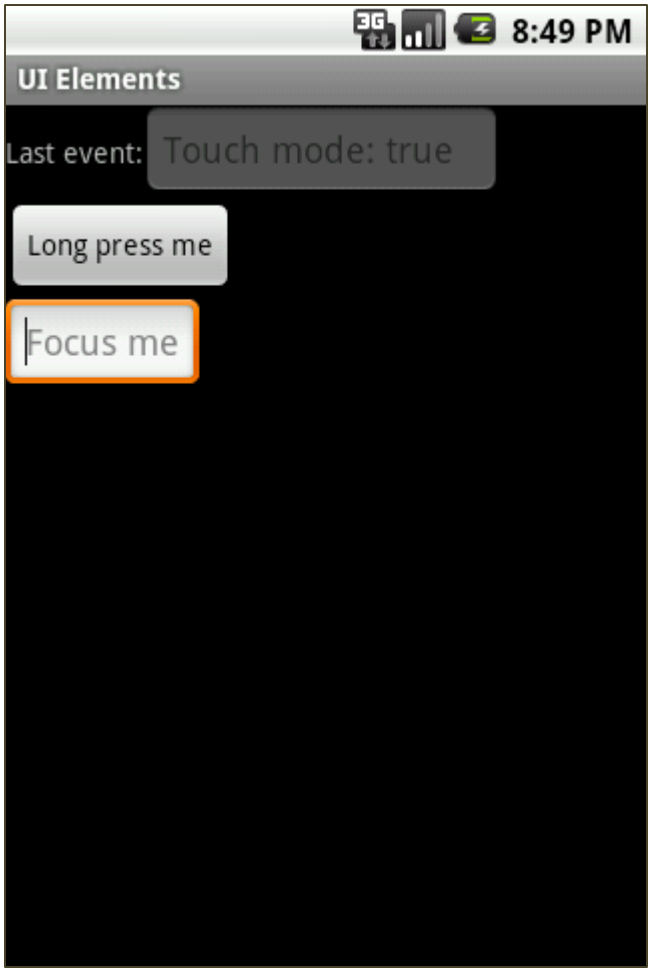
컨텍스트 메뉴

- 화면 구성 요소(View)를 오래 누르면 나타나는 메뉴
- 현재 선택된 항목에 대한 추가적인 동작들을 사용자에게 제시하는 용도로 쓰임
- 컨텍스트 메뉴 생성
  - xml 메뉴 작성 (res/menu/context_menu.xml)
  - onCreateContextMenu 메소드 재정의 : 이 함수는 옵션 메뉴에서와는 달리 매번 메뉴가 표시될 때마다 호출됨
  - 메뉴 xml을 컨텍스트 메뉴로 인플레이션(inflation)하고 HeaderIcon과 HeaderTitle 설정
- 메뉴 선택 시 실행
  - onContextItemSelected가 자동으로 호출되며, 선택된 메뉴 항목이 item 파라미터로 전달됨
  - 메뉴 항목 ID를 확인하여 적절히 수행

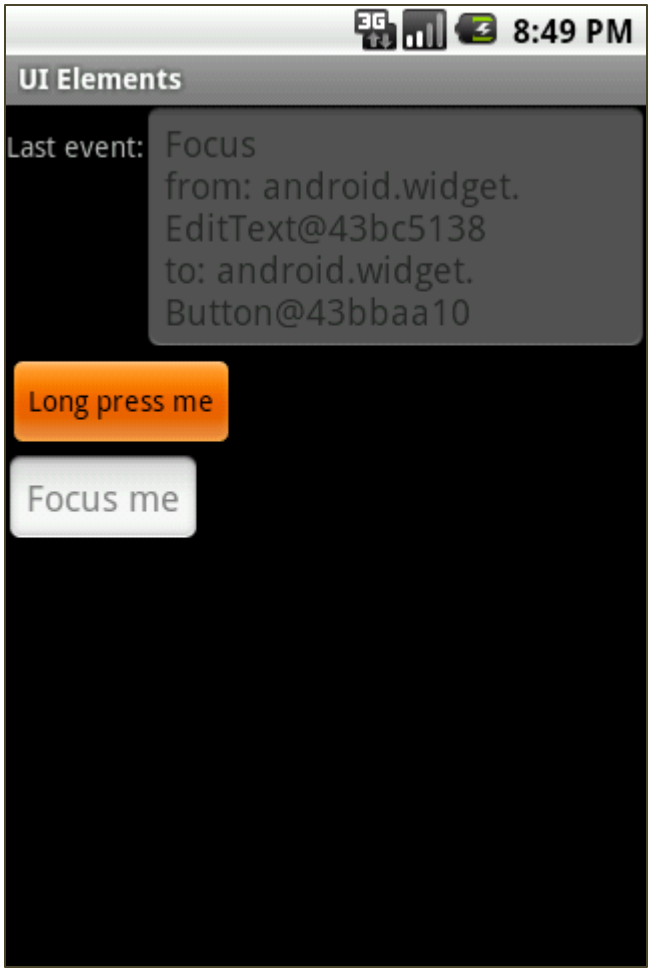
IndicatorTest



터치모드 변경 감지



터치할 때

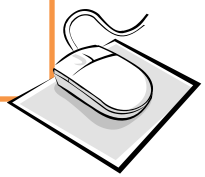


마우스 휠을 돌릴 때

터치모드 변경 감지

- 화면 혹은 위젯을 터치하는 순간 터치모드
  - EditText 같은 위젯들은 초점을 받지만 버튼은 초점을 받을 새도 없이 바로 실행
- 마우스 휠을 돌리거나 키보드를 누르는 순간 비터치 모드
  - 모든 위젯들이 초점을 받음
  - Enter나 선택 키로 항목을 작동
- 특정 위젯이 초점을 가졌느냐 아니냐에 따라 응용프로그램의 행동이 달라야 한다면 터치모드 여부를 알아야 함
  - 최상위 뷰의 뷰 트리 옵저버(ViewTreeObserver) 객체에 OnTouchModeChangeListerner를 설정함
  - 리스너 안에서 isInTouchMode 파라미터로 알 수 있음
  - 이처럼 ViewTreeObserver 객체는 화면 전체뿐만 아니라 모든 자식 위젯에 대한 이벤트를 감지함

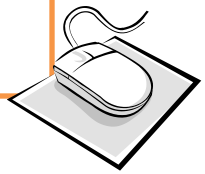
EventTest



화면 전체 이벤트 감지

- 화면 터치 이외에 다음 이벤트도 감지함
  - PreDraw : 뷰가 그려지기 전에 발생하는 이벤트
    - 처리시 ViewTreeObserver.OnPreDrawListener 인터페이스 구현 및 addOnPreDrawListener 메소드로 등록
  - GlobalLayout : 뷰의 레이아웃, 가시성 변경이 발생
    - ViewTreeObserver.OnGlobalLayoutListener 인터페이스 구현 / addOnGlobalLayoutListener 메소드로 등록
  - GlobalFocusChange : 뷰의 입력 초점이 변할 때 발생
    - ViewTreeObserver.OnGlobalFocusChangeListener 인터페이스 구현 / addOnGlobalFocusChangeListener 메소드로 등록

EventTest



긴 클릭 감지

- 긴 클릭 이벤트에 대한 반응으로 문맥 메뉴가 흔히 쓰이나 문맥 메뉴 표시 이외의 처리도 가능
- 하지만 문맥 메뉴가 등록되어 있는 상태에서 긴 클릭 이벤트에 대한 또 다른 처리는 사용자 혼란을 야기할 수 있으므로(문맥 메뉴가 뜨길 기대했는데 다른 처리가 수행됨) 이벤트 처리의 일관성 필요
- 긴 클릭 처리의 예
  - 가령, 버튼 위젯을 오래 누르면 버튼에 LongClick 이벤트가 발생함
  - 이를 처리하고자 한다면 OnLongClickListener 리스너 객체를 만들어서 버튼에 등록함

포커스 변경

- 초점 변경 처리의 예
  - 가령, 특정 TextView 위젯의 포커스가 바뀌면 FocusChange 이벤트가 발생함
  - 이를 처리하고자 한다면 View.OnFocusChangeListener 리스너 객체를 만들어서 해당 위젯에 등록 (setOnFocusChangeListener 메소드)

제스처 인식

- (마우스 버튼을, 화면을) 누른 상태에서 움직여서 떼는 것을 Scroll 제스처라 함
- (마우스 버튼을, 화면을) 누른 상태에서 빨리 움직여서 떼는 것을 Fling 제스처라 함
- 제스처 처리 방법
  - 제스처는 화면을 터치하는 것이므로 제스처 발생 시 액티비티의 onTouchEvent 가상함수가 실행됨
  - onTouchEvent 파라미터로 MotionEvent가 전달됨
  - MotionEvent를 제스처 인식기(GestureDetector)로 전송
  - 제스처 인식기는 이를 해석하여 Fling 발생 시 자신의 onFling을, Scroll은 onScroll 메소드 호출

스타일 다루기

- 스타일(style)이란 개별 위젯에 적용되는 속성 집합
- 하나의 스타일을 정의해 두면 그것을 여러 개의 위젯에 적용할 수 있음
- 스타일에는 위젯을 그리는 데 쓰이는 글꼴의 종류나 텍스트 색상 등이 포함
- 스타일을 정의하는 자원 파일은 일반적으로 res/values/styles.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <style name="padded_small">
        <item name="android:padding">2px</item>
        <item name="android:textSize">8px</item>
    </style>
    <style name="padded_large">
        <item name="android:padding">4px</item>
        <item name="android:textSize">16px</item>
    </style>
</resources>
```

```
<TextView
        style="@style/padded_small"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Small Padded" />
```

테마 다루기

- 테마(theme)는 개별 위젯이 아니라 한 액티비티(화면)의 모든 위젯에 적용되는 속성 집합
- 색상 구성(color scheme)이나 기타 공통적인 위젯 특성 설정 집합을 정의
- 하나의 테마를 일련의 화면에 적용하면 사용자 인터페이스의 모습에 일관성을 유지하기 편함
- 안드로이드 테마는 본질적으로 하나의 스타일이며, 단지 화면 전체에 적용된다는 점이 다름

테마 다루기

- 테마의 예(스타일도 테마)

```
<style name="right">
<item name="android:gravity">right</item>
</style>
```

- 코드 안에서 테마 적용 : 화면(액티비티) 전체에 테마 적용
  - 화면의 모든 위젯의 gravity 특성이 right가 되어 위젯들이 모두 오른쪽으로 정렬됨

```
setTheme(R.style.right);
```

  - 오른쪽으로 정렬하고 또한 번쩍이도록 여러 개의 테마를 적용

```
setTheme(R.style.right);
setTheme(R.style.green_glow);
```

- 코드 밖에서 테마 적용 : 매니패스트 xml 파일에서 적용

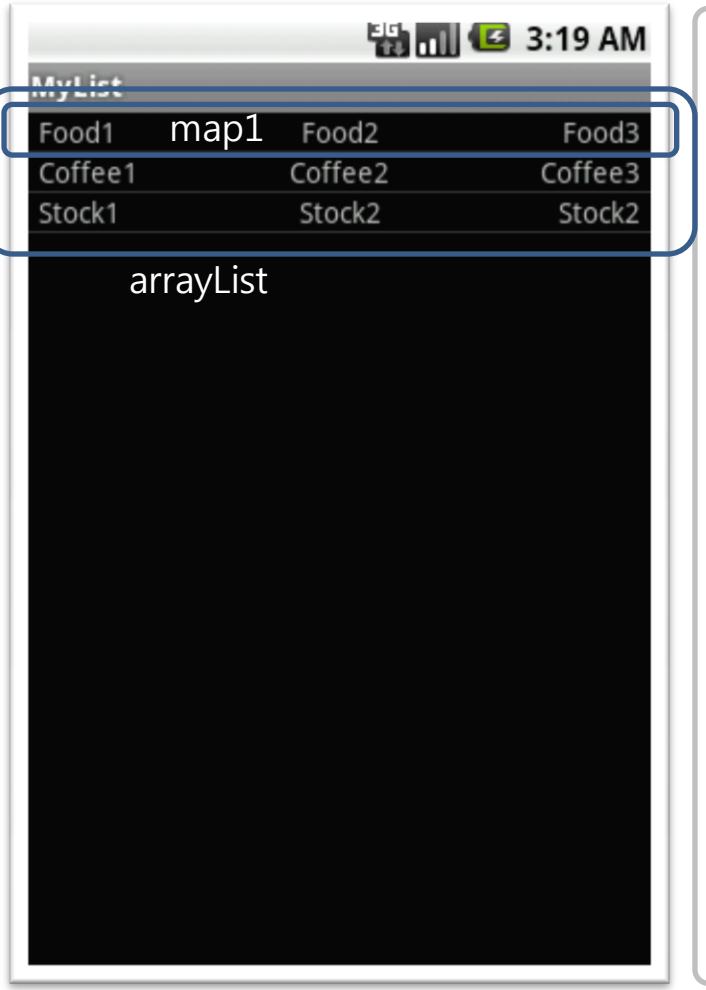
```
<activity android:name=".myactivityname"
android:label="@string/app_name"
android:theme="@style/myAppIsStyling">
```

Live Wallpapers

http://www.youtube.com/watch?v=Hz1YBcYw_qE

http://www.youtube.com/watch?v=G-O_-zF2ECQ

ListView 위젯



01. ListView 위젯은 (아답터가 데이터를 채워주는) 아답터 뷰 위젯

02. 아답터 객체 adapter를 생성한 후 아답터 뷰 listview에게 setAdapter하라고만 하면 됨

```
listview.setAdapter(adapter);
```

03.이제 아답터 객체를 만듦. 이를 위해 다음 정보가 필요

*하나의 레코드 데이터(하나 혹은 그 이상의 필드로 구성된)를 갖는 HashMap 객체
*여러 HashMap 객체를 담고 있는 ArrayList 객체(arrayList)
*실제로 데이터를 TextView를 가지고 있는 레이아웃 xml 파일(R.layout.item)
*각 필드 데이터를 찾는데 사용할 키
*찾은 필드 데이터를 넣을 TextView ID

```
ListAdapter adapter = new SimpleAdapter(
   this, arrayList, R.layout.item,
   new String[] {COL1, COL2, COL3},
   new int[]{R.id.text_1, R.id.text_2, R.id.text_3});
```

ListViewTest