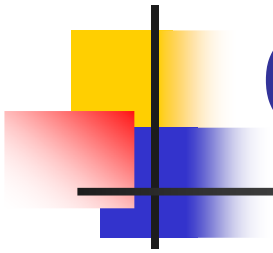
Trường Đại học Bách Khoa Hà Nội
Hanoi University of Science and Technology

# Chapter 4. Graphical User Interfaces

Hanoi University of Science and Technology

Nguyen Hong Quang
19/8/2012



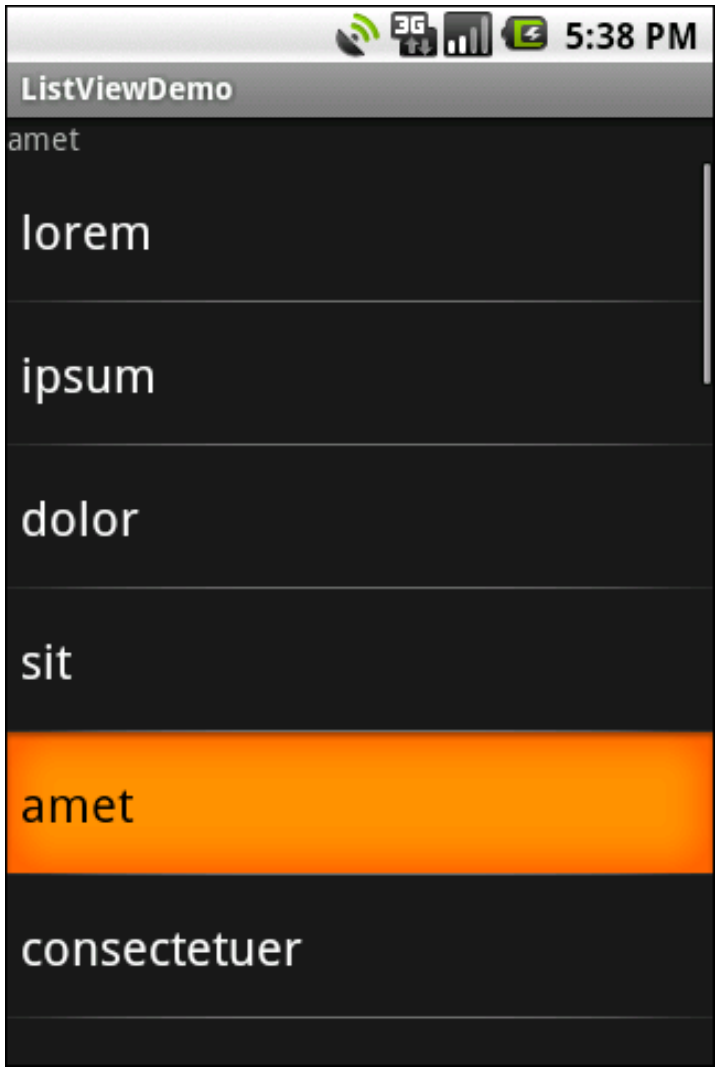
# Content

- 4.3. Advanced Adapter
  - ListView Widget/ ArrayList
  - Spin Control
  - GridView
  - Fields
  - Galleries

# *ListView Widget*

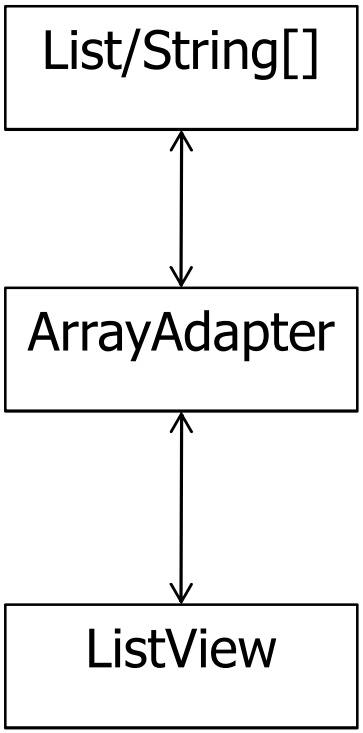
# 4.3. Advanced Adapter ListView Widget

- Select a choice from a set of items
- This item is shown in a TextView in the top of screen



# 4.3. Advanced Adapter
# Model of ListView

Lorem
Ipsum
Dolor
...



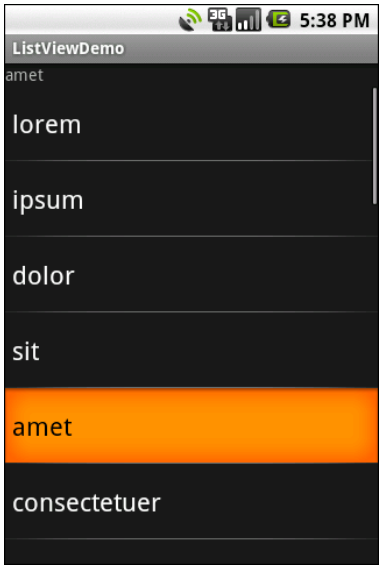

Data model

adapter = new ArrayAdapter(..., model);

Adapter

listView.setAdapter(adapter);

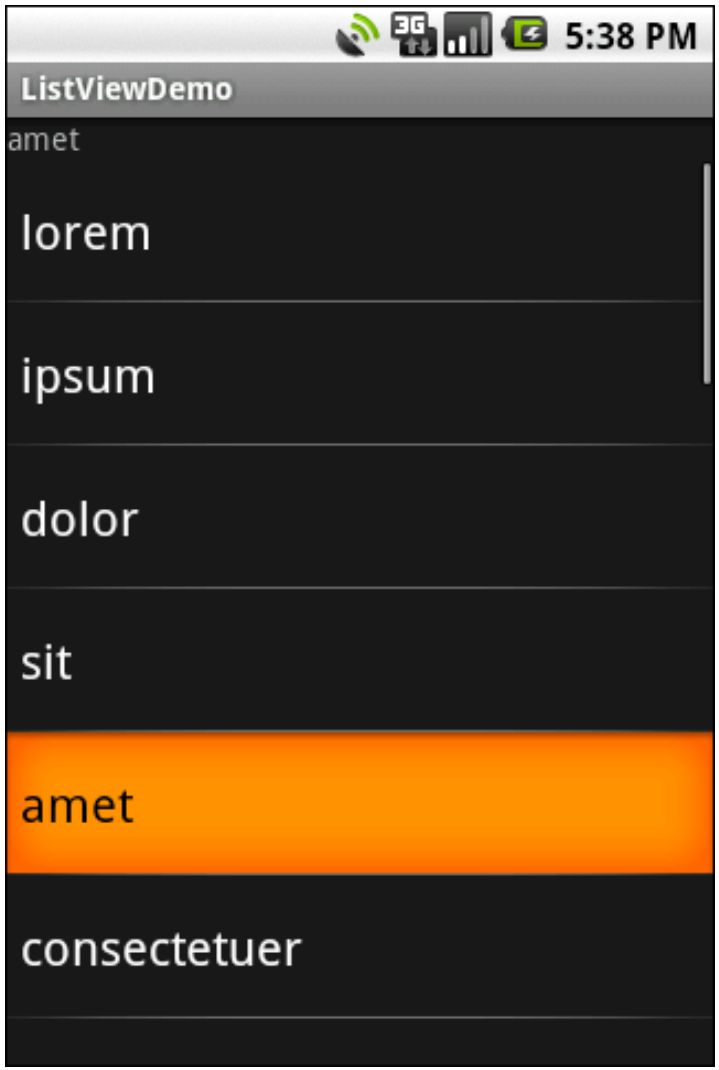
View




- Adding object:
  - adapter.add(r);

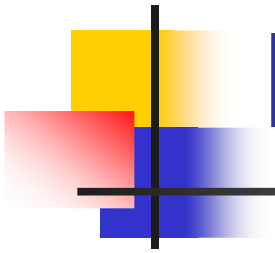
# 4.3. Advanced Adapter ListView Widget - Example

```
<LinearLayout
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >
<TextView
    android:id="@+id/selection"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"/>
<ListView
    android:id="@+id/list"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:drawSelectorOnTop="false"
/>
</LinearLayout>
```



# ListView Widget onCreate()

```
String[] dataModel ={"lorem", "ipsum", "dolor‟, "sit",
    "amet‟, "consectetuer‟};
ListView list=(ListView)findViewById(R.id.restaurants);

adapter=new ArrayAdapter<Restaurant>(this,
    android.R.layout.simple_list_item_1,
     dataModel);

list.setAdapter(adapter);

list.setOnItemClickListener(onListClick);
```



# ListView Widget

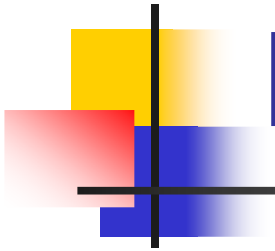
```
private AdapterView.OnItemClickListener onListClick = new
    AdapterView.OnItemClickListener() {
        public void onItemClick(AdapterView<?> parent,
                View view, int position, long id) {
            selection.setText(items[position]);
        }
    };
```

# *ListView Widget*

## Selection Modes

# ListView Widget Selection Modes

- ListView is set up simply to collect clicks on list entries.
- Sometimes, though, we want a list that tracks a user's selection, or possibly multiple selections.

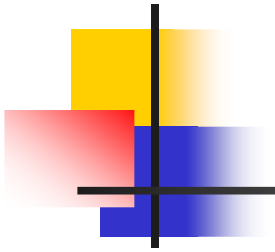


# Declaration in XML layout

```
<ListView
    android:id="@android:id/list"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:drawSelectorOnTop="false"
    android:choiceMode="multipleChoice"
/>
```

- setListAdapter(new ArrayAdapter<String>(this, android.R.layout.simple_list_item_multiple_choice, items));

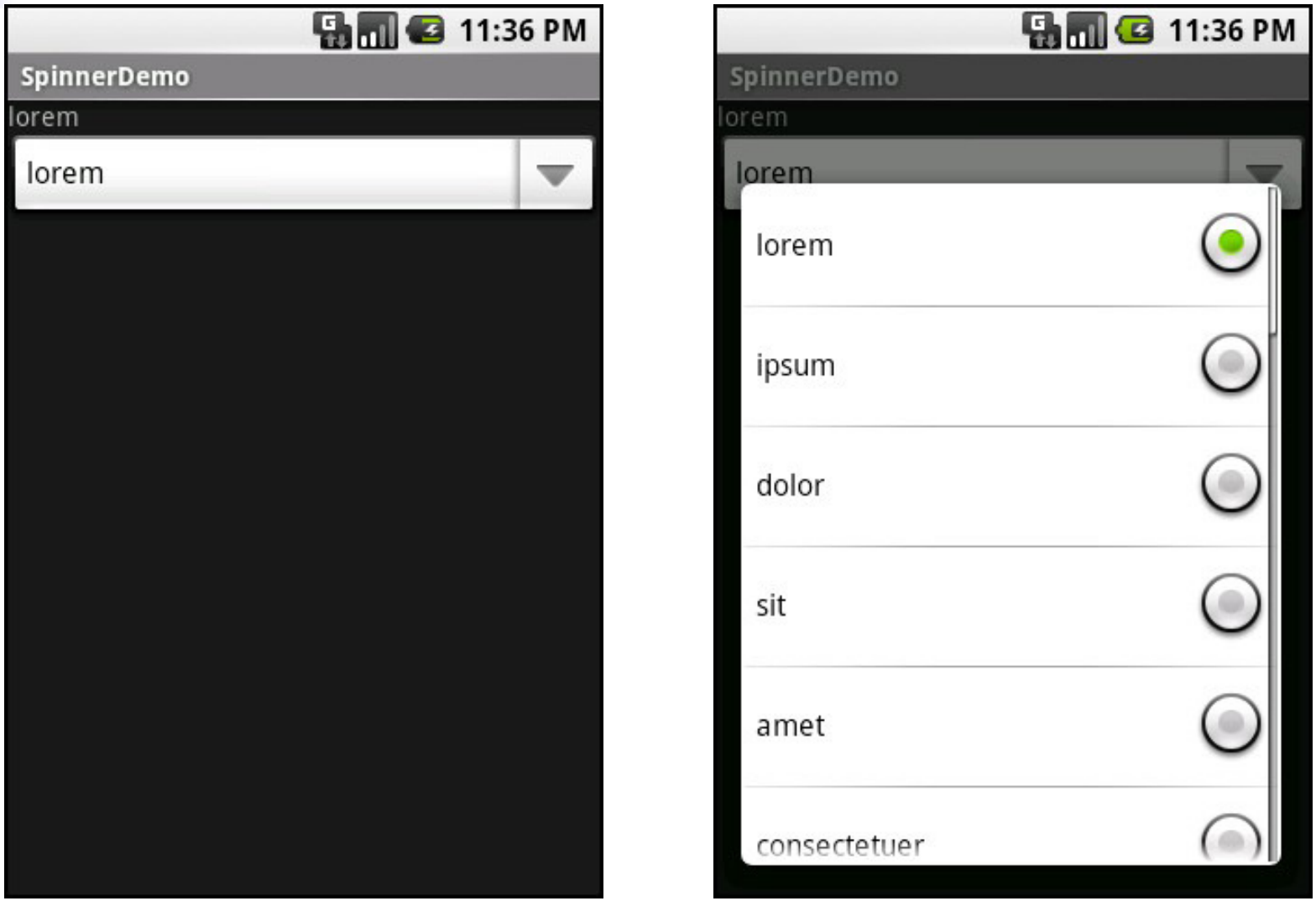
# ListView Widget Selection Modes

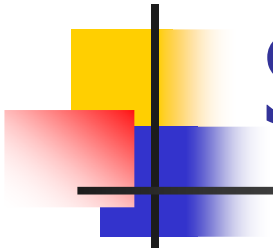


- Find out which items the user checked: getCheckedItemPositions();
- Check (or un-check) a specific entry: setItemChecked();

# Spin Control

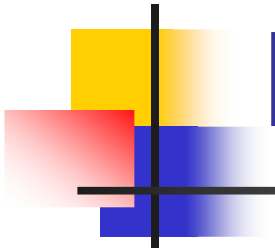
# Spin Control





# Spin Control

- Provide the adapter for data and child views via setAdapter()
- Hook in a listener object for selections via setOnItemSelectedListener().



# Declaration in XML layout

```
<LinearLayout
   android:orientation="vertical“ >
<TextView
    android:id="@+id/selection“ />
<Spinner
    android:id="@+id/spinner“
    android:drawSelectorOnTop="true” />
</LinearLayout>
```

# Java code

public class SpinnerDemo extends Activity implements AdapterView.OnItemSelectedListener

String[] dataModel ={"lorem", "ipsum", "dolor‟, "sit", "amet‟, "consectetuer‟};

- onCreate method:

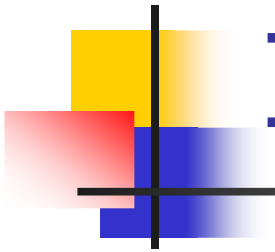
Spinner spin=(Spinner)**findViewById(R.id.spinner);**

spin.**setOnItemSelectedListener(this);**

ArrayAdapter<String> aa=new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item,
items);

aa.**setDropDownViewResource(**
android.R.layout.simple_spinner_dropdown_item);

spin.**setAdapter(aa);**



# Implementing Interface

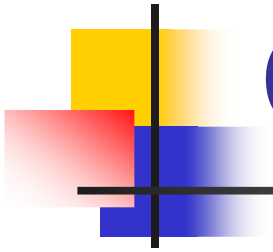
```
public void onItemSelected(AdapterView<?> parent,
    View v, int position, long id) {
    selection.setText(items[position]);
}
public void onNothingSelected(AdapterView<?> parent) {
    selection.setText("");
}
```

# GridView

# GridView

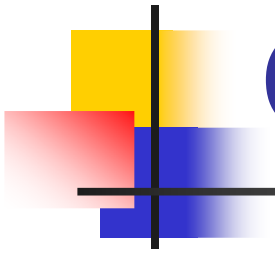

11:55 AM
GridDemo
consectetuer
lorem
ipsum
dolor
sit
amet
consectetuer
adipiscing
elit
morbi
vel
ligula
vitae
arcu
aliquet
mollis
etiam
vel
erat
placerat
ante
porttitor
sodales
pellentesque
augue


11:56 AM
GridDemo
sit
amet
consectetuer
adipiscing
elit
morbi
vel
ligula
vitae
arcu
aliquet
mollis
etiam
vel
erat
placerat
ante
porttitor
sodales
pellentesque
augue
purus



# GridView – number of rows

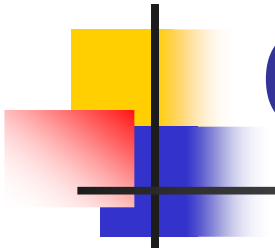
- Number of rows is dynamically determined based on the number of items the supplied adapter says are available for viewing



# GridView – number of columns

- android:numColumns
- Spells out how many columns there are
- If we supply a value of auto_fit, Android will compute the number of columns based on available space and the properties listed below.

# GridView
# Others properties



- android:verticalSpacing
- android:horizontalSpacing
- android:columnWidth
- android:stretchMode

# GridView Declaration in XML file

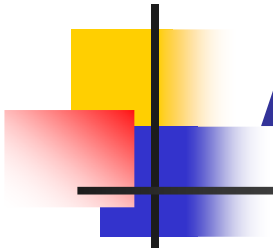
```
<LinearLayout>
<TextView/>
<GridView
    android:id="@+id/grid"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:verticalSpacing="40dip"
    android:horizontalSpacing="5dip"
    android:numColumns="auto_fit"
    android:columnWidth="100dip"
    android:stretchMode="columnWidth"
    android:gravity="center”/>
</LinearLayout>
```

# GridView
# The grid cells definition

- The grid cells are defined by a separate res/layout/cell.xml file, referenced in our ArrayAdapter as R.layout.cell:

```
<?xml version="1.0" encoding="utf-8"?>
<TextView
   xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textSize="14dip"
/>
```

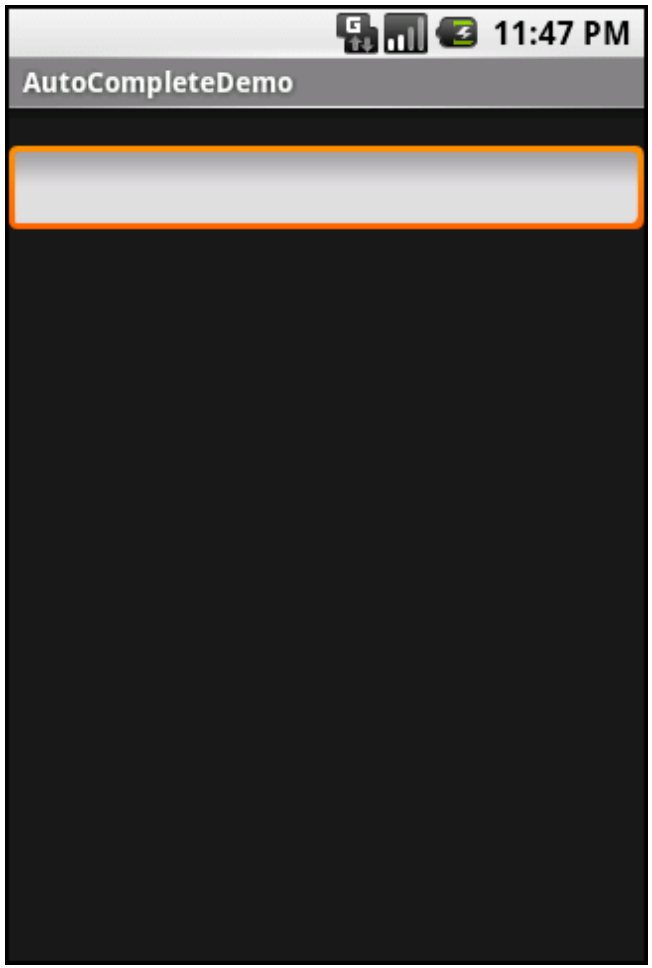
# AutoCompleteTextView

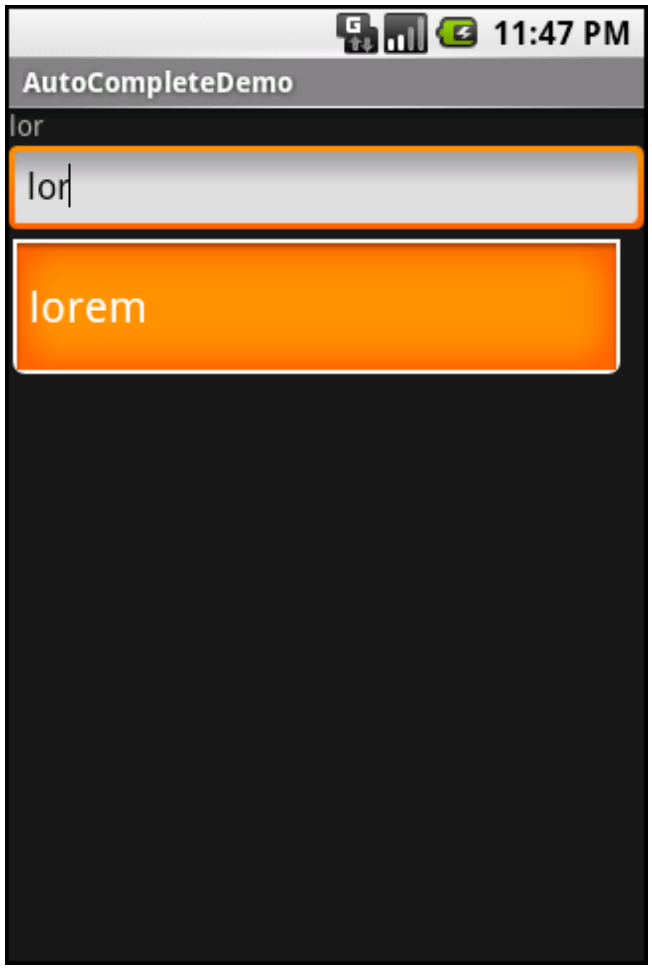


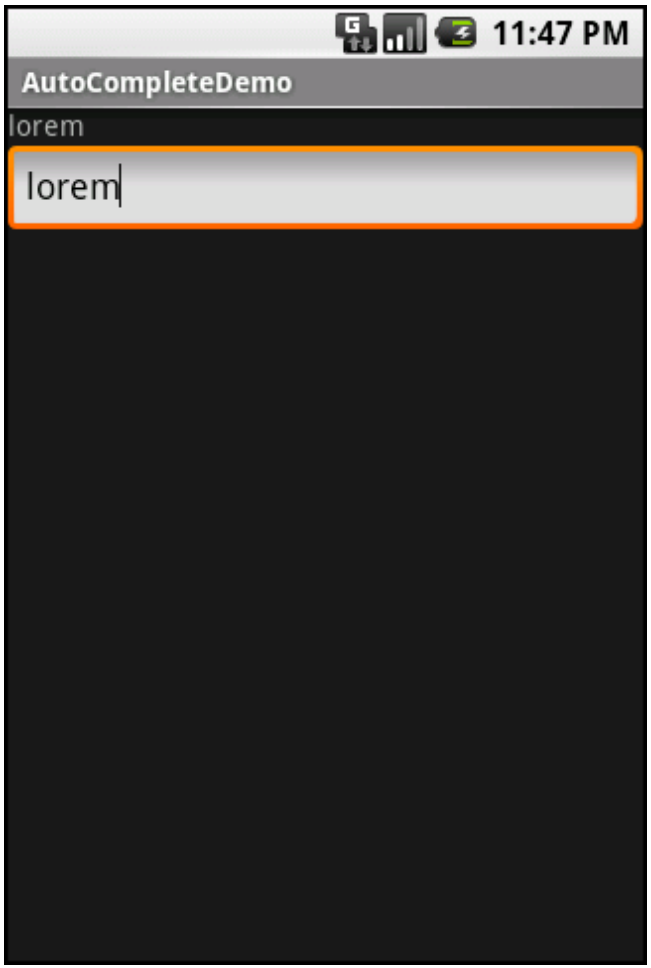
# AutoCompleteTextView

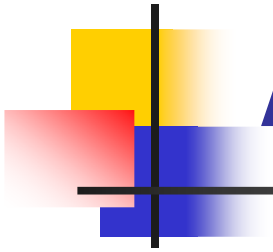
- The AutoCompleteTextView is sort of a hybrid between the EditText (field) and the Spinner.
- With auto-completion, as the user types, the text is treated as a prefix filter, comparing the entered text as a prefix against a list of candidates.
- Matches are shown in a selection list that folds down from the field. The user can either type out an entry (e.g., something not in the list) or choose an entry from the list to be the value of the field.

# AutoCompleteTextView



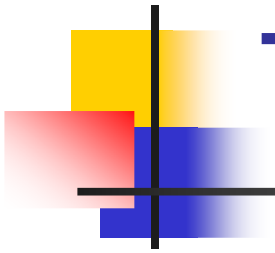





# AutoCompleteTextView

- android:completionThreshold
- indicate the minimum number of characters a user must enter before the list filtering begins

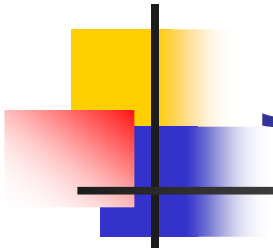


# TextWatcher

- to be notified when the text changes.
- These events will occur either because of manual typing or from a selection from the drop-down list.

# Declaration in XML file

```
<LinearLayout>
    <TextView
      android:id="@+id/selection“ />
    <AutoCompleteTextView
      android:id="@+id/edit“
      android:completionThreshold="3"/>
</LinearLayout>
```



# Java code

```
public class AutoCompleteDemo extends Activity implements
    TextWatcher {
    String[] items={"lorem", "ipsum", "dolor‟, "sit", "amet‟};
}
```

- OnCreate

```
edit.addTextChangedListener(this);
edit.setAdapter(new ArrayAdapter<String>(this,
    android.R.layout.simple_dropdown_item_1line,
    items));
```

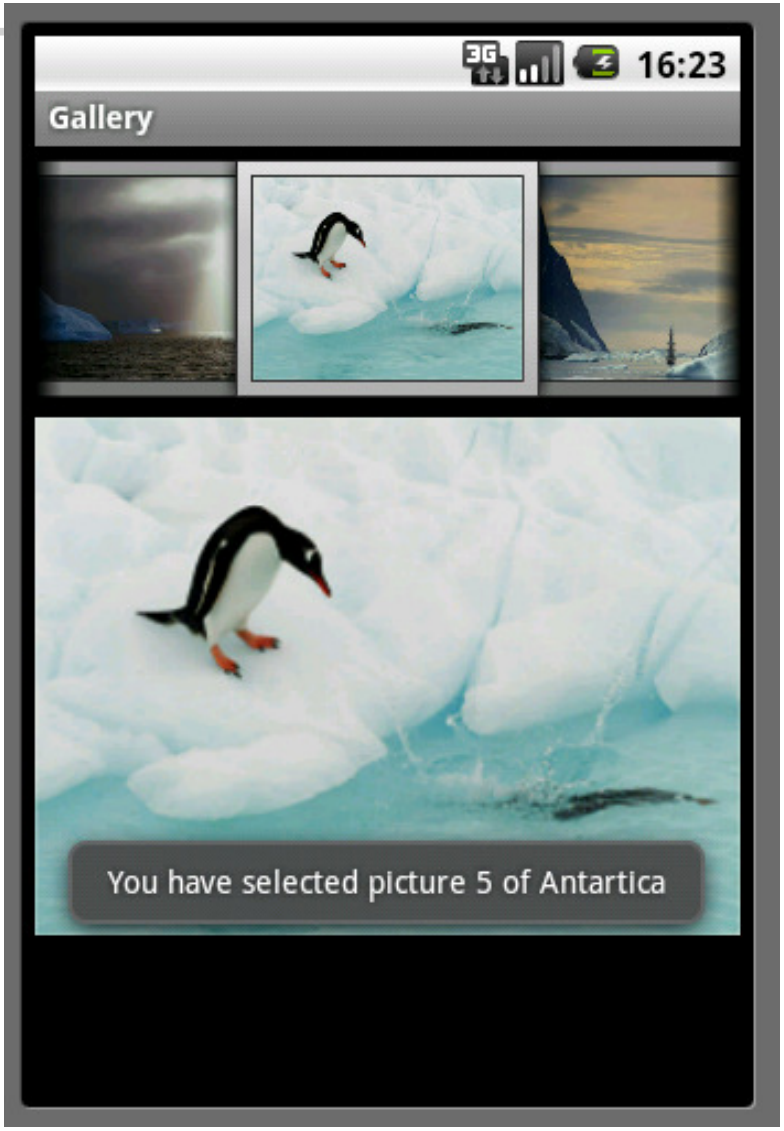
# Implementing TextWacher interface

```
public void onTextChanged(CharSequence s, int start, int before,
    int count) {
        selection.setText(edit.getText());
}
public void beforeTextChanged(CharSequence s, int start,
    int count, int after) {
        // needed for interface, but not used
}
public void afterTextChanged(Editable s) {
        // needed for interface, but not used
}
```

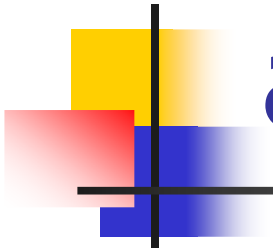
# *Gallery View*

# Gallery View

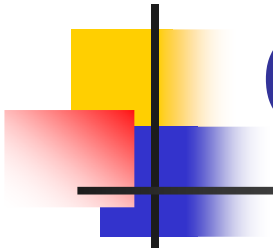
- A horizontally-laid-out listbox.
- One choice follows the next across the horizontal plane, with the currently-selected item highlighted.



# Compare Gallery with ListView and Spinner

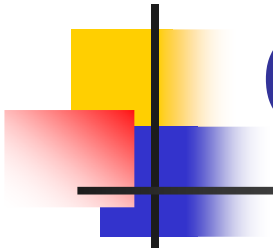


- Compared to the ListView, the Gallery takes up less screen space while still showing multiple choices at one time (assuming they are short enough).
- Compared to the Spinner, the Gallery always shows more than one choice at a time.



# GalleryView properties

- android:spacing controls the number of pixels between entries in the list
- android:spinnerSelector controls what is used to indicate a selection – this can either be a reference to a Drawable (see the resources chapter) or an RGB value in #AARRGGBB or similar notation

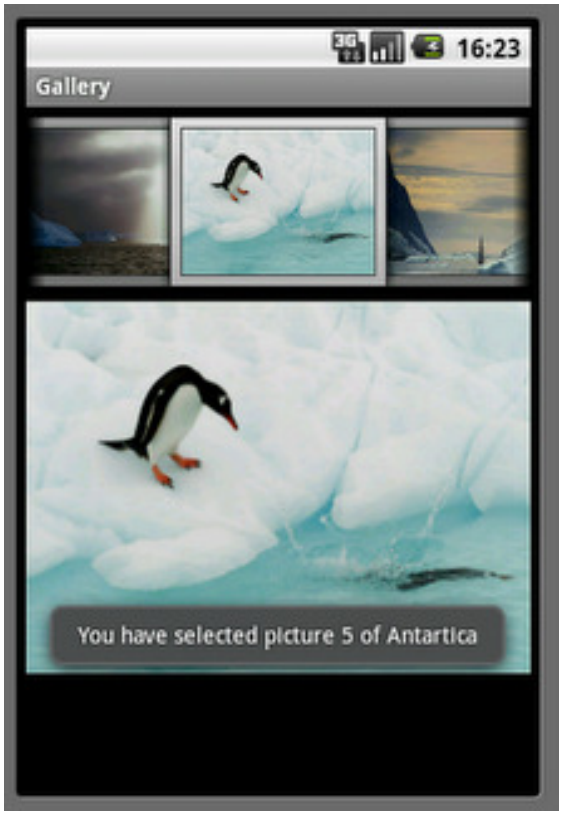


# Gallery properties

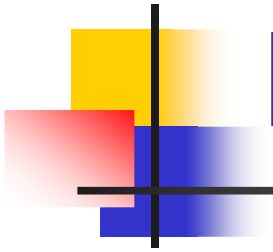
- android:drawSelectorOnTop indicates if the selection bar (or Drawable) should be drawn before (false) or after (true) drawing the selected child
- If we choose true, be sure that your selector has sufficient transparency to show the child through the selector, otherwise users will not be able to read the selection

# Gallery View Example – XML layout

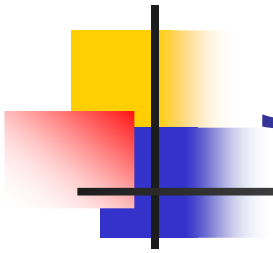
```
<Gallery
        android:id="@+id/Gallery01"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content">
</Gallery>


<ImageView
    android:id="@+id/ImageView01"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">
</ImageView>
```



# Gallery View Example –

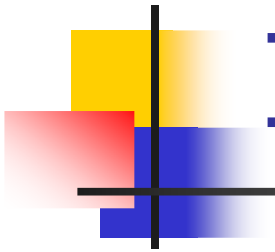


- created small sized images of antartica and stored them in the res/drawable-ldpi folder starting form antartica1.png to antartica10.png.



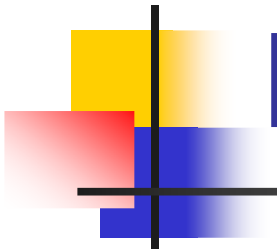
# Java code

```
Integer[] pics = {
    R.drawable.antartica1,
    R.drawable.antartica2,
    R.drawable.antartica3,
    R.drawable.antartica4,
    R.drawable.antartica5,
    R.drawable.antartica6,
    R.drawable.antartica7,
    R.drawable.antartica8,
    R.drawable.antartica9,
    R.drawable.antartica10
    };
```



# Insert adapter for Gallery

```
Gallery ga = (Gallery)findViewById(R.id.Gallery01);

ga.setAdapter(new ImageAdapter(this));
```

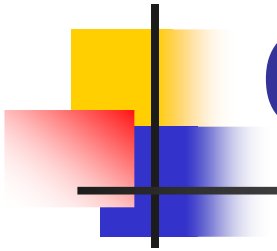


# BaseAdapter

```
public class ImageAdapter extends BaseAdapter {
    private Context ctx;
    int imageBackground;
    public ImageAdapter(Context c) {
        ctx = c;
        TypedArray ta = obtainStyledAttributes(R.styleable.Gallery1);
        imageBackground =
    ta.getResourceId(R.styleable.Gallery1_android_galleryItemBackground,
    1);
        ta.recycle();
    }
    …
    }
```

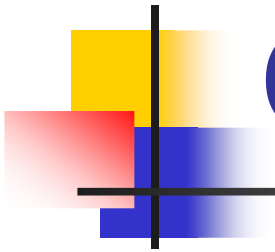
# attributes.xml in the res/values folder

```
<resources>
   <declare-styleable name="Gallery1“>
   <attr name="android:galleryItemBackground"/>
   </declare-styleable>
</resources>
```



# Over ridden methods

```
@Override
    public int getCount() {
        return pics.length;
    }
    @Override
    public Object getItem(int arg0) {
        return arg0;
    }
    @Override
    public long getItemId(int arg0) {
        return arg0;
    }
```



# Over ridden methods

```
@Override
    public View getView(int arg0, View arg1, ViewGroup arg2)
    {
        ImageView iv = new ImageView(ctx);
        iv.setImageResource(pics[arg0]);
        iv.setScaleType(ImageView.ScaleType.FIT_XY);
        iv.setLayoutParams(new
          Gallery.LayoutParams(150,120));
        iv.setBackgroundResource(imageBackground);
        return iv;
    }
```

# Captured the onClick event on a Gallery item

```
ga.setOnItemClickListener(new OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> arg0, View arg1,
    int arg2, long arg3) {
        Toast.makeText(getBaseContext(), "You have selected
          picture " + (arg2+1) + " of Antartica",
            Toast.LENGTH_SHORT).show();
        imageView.setImageResource(pics[arg2]);
    }
    });
```

# End of Lecture

Q&A