ANDROID DEVELOPMENT

UI CONTROLS

Tan Cher Wah (isstcw@nus.edu.sg)

Agenda

- Views (widgets, panels)
- ViewGroups (layout managers)
- Menu / Context Menu
- Messages
- Dialogs
- List Activity and Data Adapter

Views

- Sample views
  - Button
  - Text field
  - Checkbox
  - Radio button within RadioGroup
  - Toggle button & Switch
  - Spinner
  - Pickers

Event-Handling (1)

- Event handling via anonymous class

```
public class P1Activity extends AppCompatActivity {
    Button button1;
    CheckBox checkbox1;
    private View.OnClickListener myClickListener = new View.OnClickListener() {
        public void onClick(View v) {
            checkbox1.toggle();
        }
    };

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        button1 = (Button) findViewById(R.id.button1);
        checkbox1 = (CheckBox) findViewById(R.id.checkBox1);
        button1.setOnClickListener(myClickListener);
    }
}
```

Event-Handling (2)

- Event handling via implementing interface

```
public class P1Activity extends AppCompatActivity implements View.OnClickListener {
    Button button1;
    CheckBox checkbox1;
    public void onClick(View v) {
        checkbox1.toggle();
    }

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        button1 = (Button) findViewById(R.id.button1);
        checkbox1 = (CheckBox) findViewById(R.id.checkBox1);
        button1.setOnClickListener(this);
    }
}
```

Option Menu

- res/menu/menu_main.xml

```xml
<menu xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:app="http://schemas.android.com/apk/res-auto"
   xmlns:tools="http://schemas.android.com/tools"
   tools:context="com.cherwah.datapersistence.MainActivity">
   <item
      android:id="@+id/action_settings"
      android:orderInCategory="100"
      android:title="@string/action_settings"
      app:showAsAction="never" />
   <item
      android:id="@+id/say_hello"
      android:orderInCategory="101"
      android:title="@string/say_hello"
      app:showAsAction="never" />
</menu>
```

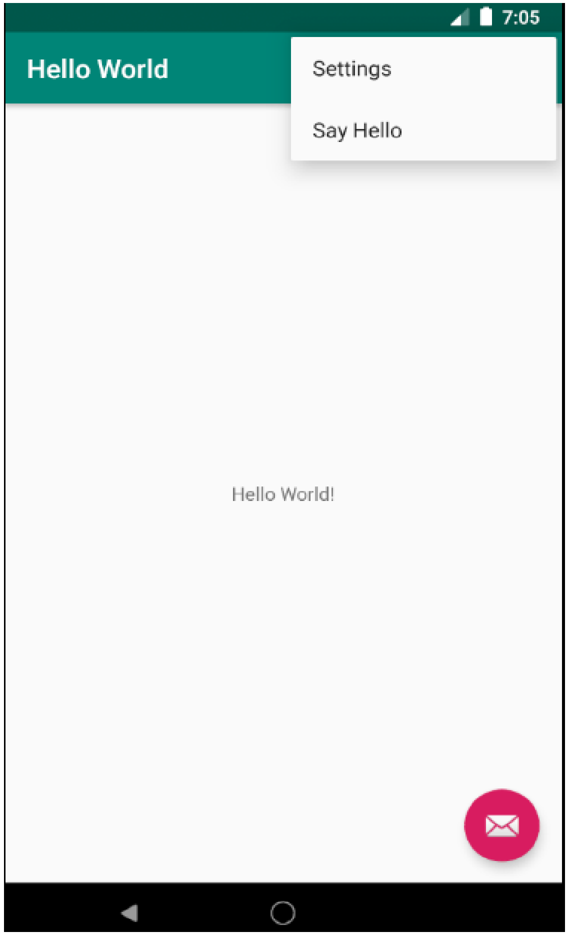
- res/values/strings.xml

```xml
<resources>
   <string name="app_name">Hello World</string>
   <string name="action_settings">Settings</string>
   <string name="say_hello">Say Hello</string>
   <string name="hello_msg">Hello, how are you?</string>
</resources>
```

Option Menu

- Show the menu

```
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.menu_main, menu);
    return true;
}
```



Option Menu

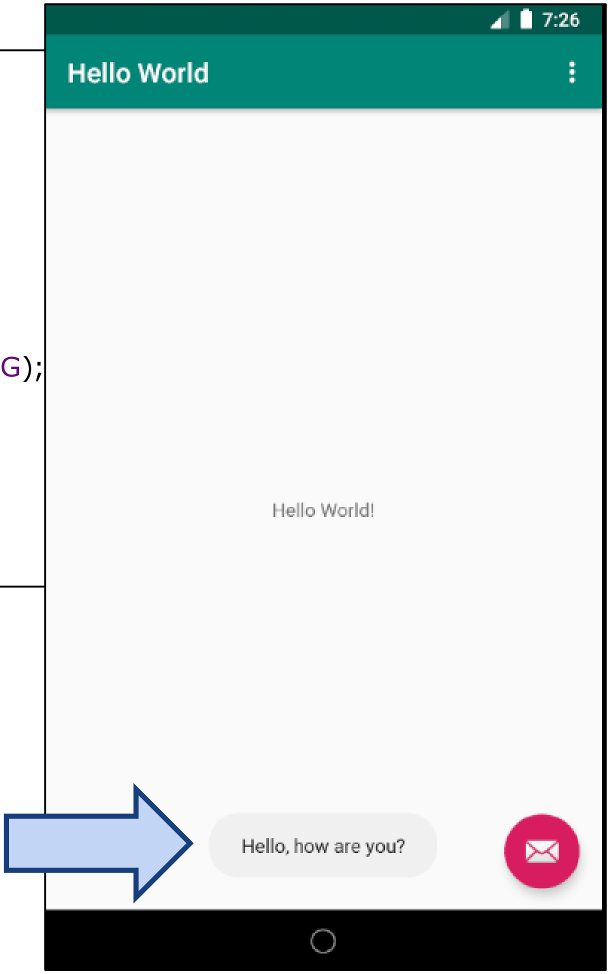
- Event Handling

```
@Override
public boolean onOptionsItemSelected(MenuItem item) {

    switch (item.getItemId()) {
        case R.id.action_settings:
            return true;

        case R.id.say_hello:
            String helloMsg = getString(R.string.hello_msg);
            Toast t = Toast.makeText(this, helloMsg, Toast.LENGTH_LONG);
            t.show();
            return true;
    }

    return super.onOptionsItemSelected(item);
}
```



Hello, how are you?

Toast Messages

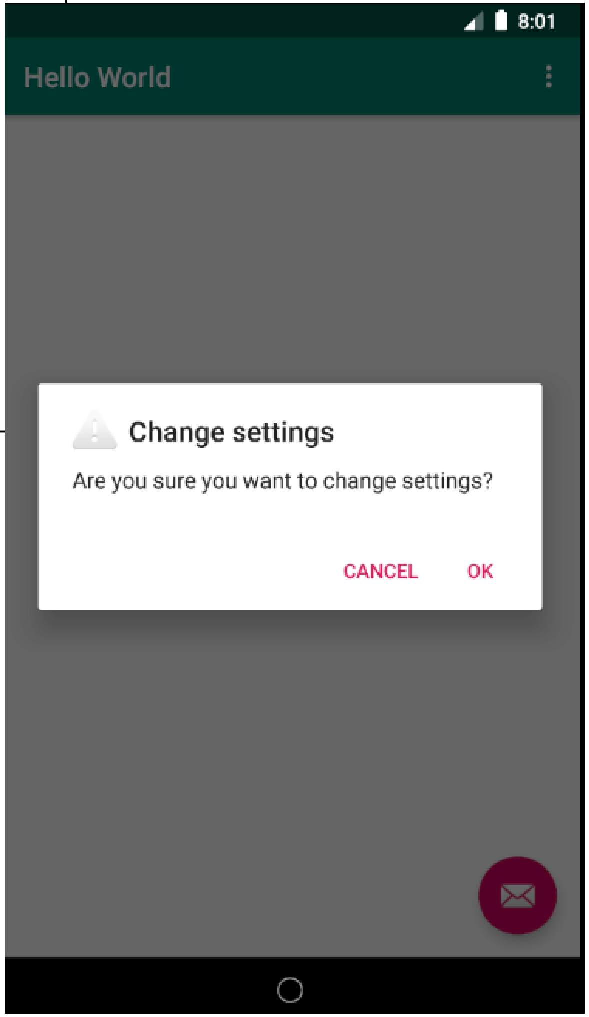
- Toast is a view containing a quick little message
- Will dismiss itself after time lapse - without user intervention (**LENGTH_SHORT** / **LENGTH_LONG**)

Dialogs

- Dialogs allow for a pop-up layout
- Unlike **Toast**, requires explicit dismissal
- Created via **new**
- Displayed via **show()**
- Removed via **dismiss()**

Alert Dialog

```
new AlertDialog.Builder(this)
    .setTitle("Change settings")
    .setMessage("Are you sure you want to change settings?")
    .setPositiveButton(android.R.string.yes,
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int which) {
                }
            })
    .setNegativeButton(android.R.string.no,
            new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int which) {
                }
            })
    .setIcon(android.R.drawable.ic_dialog_alert)
    .show();
```



Change settings

Are you sure you want to change settings?

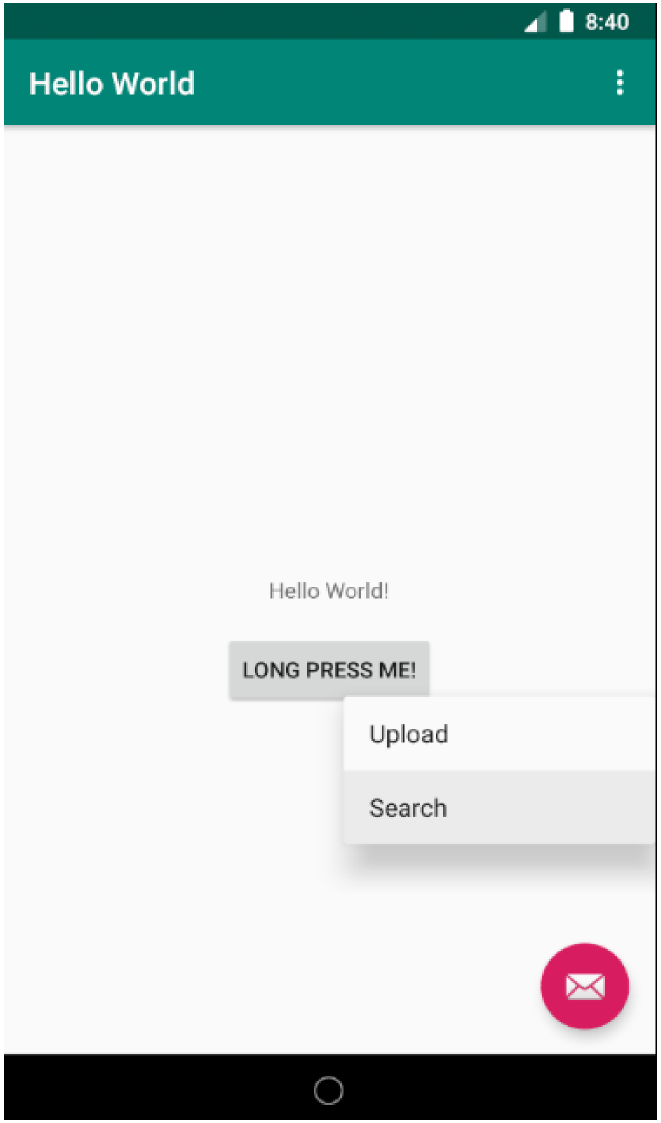
Context Menu

- Register context menu

```
registerForContextMenu(findViewById(R.id.long_press_btn));
```

- Show context menu

```
@Override
public void onCreateContextMenu(ContextMenu menu,
          View v, ContextMenu.ContextMenuInfo menuInfo)
{
   super.onCreateContextMenu(menu, v, menuInfo);
   MenuInflater inflater = getMenuInflater();
   inflater.inflate(R.menu.context_menu, menu);
}
```



Context Menu

- Event Handling

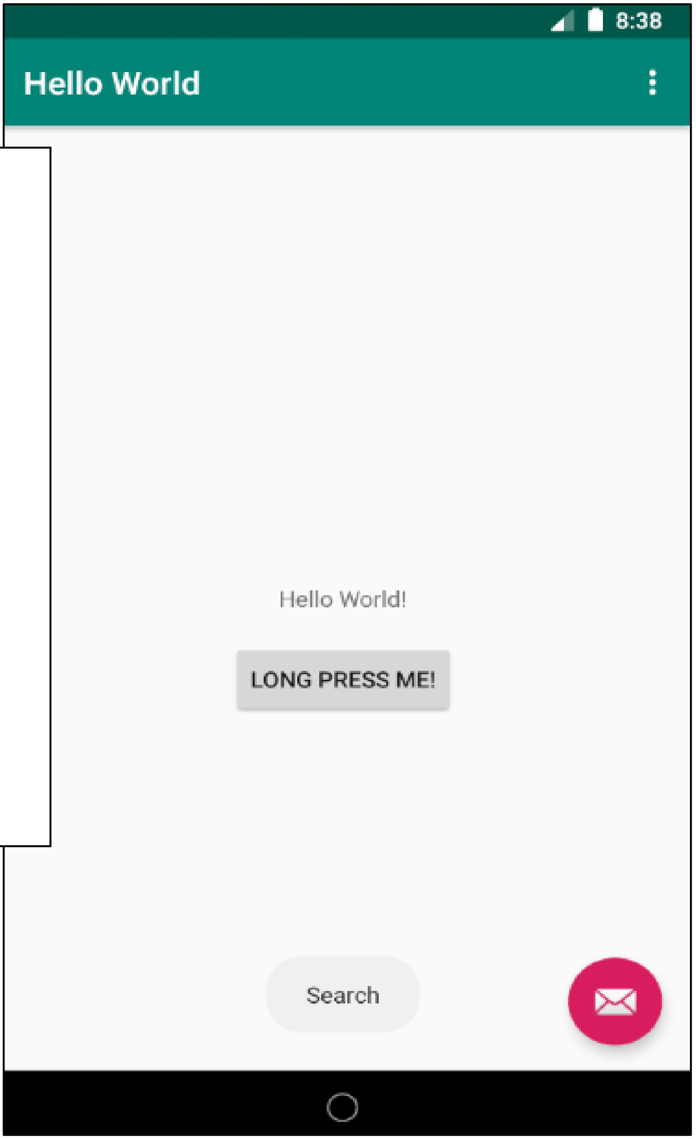
```
@Override
public boolean onContextItemSelected(MenuItem item) {
    String selected = null;

    switch (item.getItemId()) {
        case R.id.menu_item_upload:
            selected = getString(R.string.upload);
            break;

        case R.id.menu_item_search:
            selected = getString(R.string.search);
            break;
    }

    if (selected != null) {
        Toast.makeText(this, selected, Toast.LENGTH_SHORT).show();
        return true;
    }

    return super.onContextItemSelected(item);
}
```



ListView

- **ListView** can be used to display data sequence as a scrollable list
  - Scrolling through list and selecting particular row
  - Additional layout is used to specify how each row looks

- Simplified using **ListActivity** instead of **AppCompatActivity**

Passing list data via Intent Bundle

- Prepare a list of data and pass to our MyListActivity

```
int requestCode = 1;

String[] values = new String[] { "Apple", "Orange",
        "Pear", "Blueberry", "Banana", "Mango",
        "Strawberry", "Apricot", "Durian", "Lemon" };

Intent i = new Intent(this, MyListActivity.class);
i.putExtra("list", values);

startActivityForResult(i, requestCode);
```

ListActivity

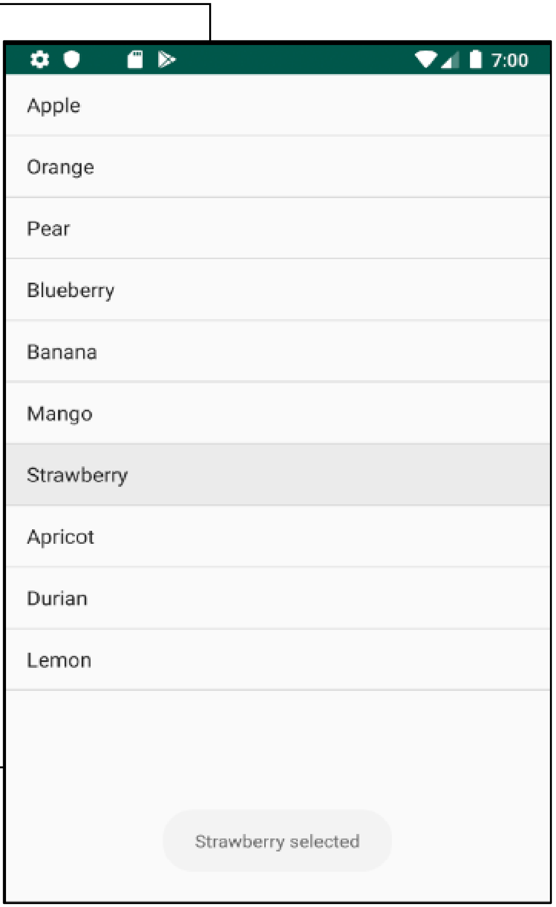
- Default list and row (simple_list_item_1) layouts are predefined if you use ListActivity

```
public class MyListActivity extends ListActivity
{
   @Override
   public void onCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);

      Bundle extras = getIntent().getExtras();
      String []values = extras.getStringArray("list");

      ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_1, values);
      setListAdapter(adapter);
   }


   @Override
   protected void onListItemClick(ListView l, View v, int position, long id)
   {
      String item = (String) getListAdapter().getItem(position);
      Toast.makeText(this, item + " selected", Toast.LENGTH_LONG).show();
   }
}
```



7:00

Apple
Orange
Pear
Blueberry
Banana
Mango
Strawberry
Apricot
Durian
Lemon

Strawberry selected

ListActivity

- Predefined layouts for ListActivity

default list view from ListActivity

```
<ListView android:id="@android:id/list"
   android:layout_width="match_parent"
   android:layout_height="match_parent" >
</ListView>
```

simple_list_item_1.xml

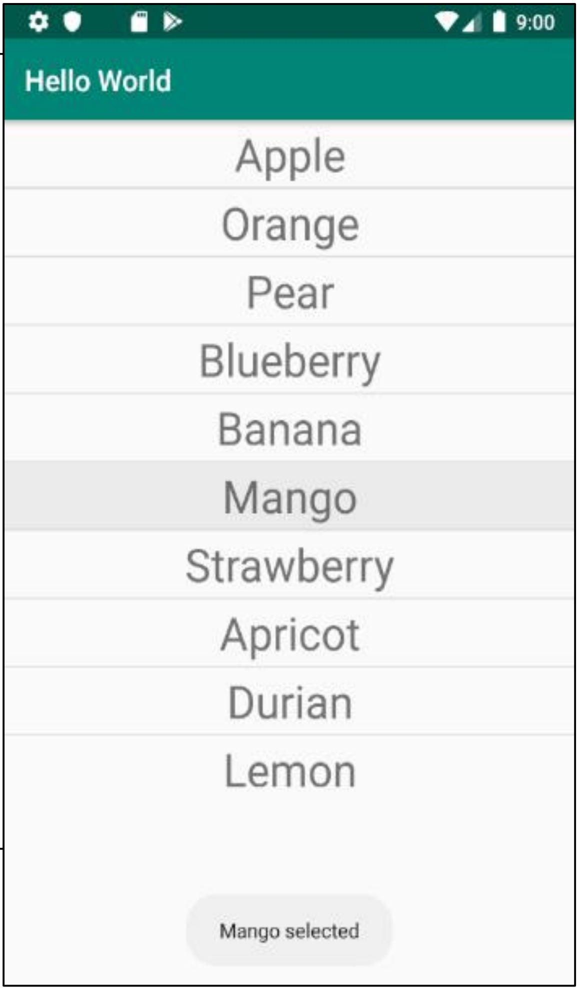
```
<TextView
   android:id="@android:id/text1"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content" >
</TextView>
```

Using AppCompatActivity (in place of ListActivity)

```
public class MyListActivity extends AppCompatActivity
        implements AdapterView.OnItemClickListener {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.my_list_activity);
        Bundle extras = getIntent().getExtras();
        String[] values = extras.getStringArray("list");
        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
                R.layout.row, R.id.textView, values);
        ListView list = (ListView) findViewById(R.id.listView1);
        list.setAdapter(adapter);
        list.setOnItemClickListener(this);
    }

    @Override
    public void onItemClick(AdapterView<?> av, View v, int pos, long id)
    {
        String item = (String) av.getItemAtPosition(pos);
        Toast.makeText(this, item + " selected",
                Toast.LENGTH_LONG).show();
    }
}
```



Hello World

Apple

Orange

Pear

Blueberry

Banana

Mango

Strawberry

Apricot

Durian

Lemon

Mango selected

ListActivity

- Customized list view layouts

my_list_activity.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent">

   <ListView
      android:id="@+id/listView1"
      android:layout_width="match_parent"
      android:layout_height="wrap_content">
   </ListView>

</LinearLayout>
```

row.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
>

   <TextView
      android:id="@+id/textView"
      android:gravity="center"
      android:height="120px"
      android:textSize="14pt"
      android:layout_width="match_parent"
      android:layout_height="match_parent" />

</LinearLayout>
```

Customized row layout

- Row layout may contain more than 1 widgets

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical">

   <TextView
      android:id="@+id/textView1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content" />

   <TextView
      android:id="@+id/textView2"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content" />

</LinearLayout>
```

row2.xml

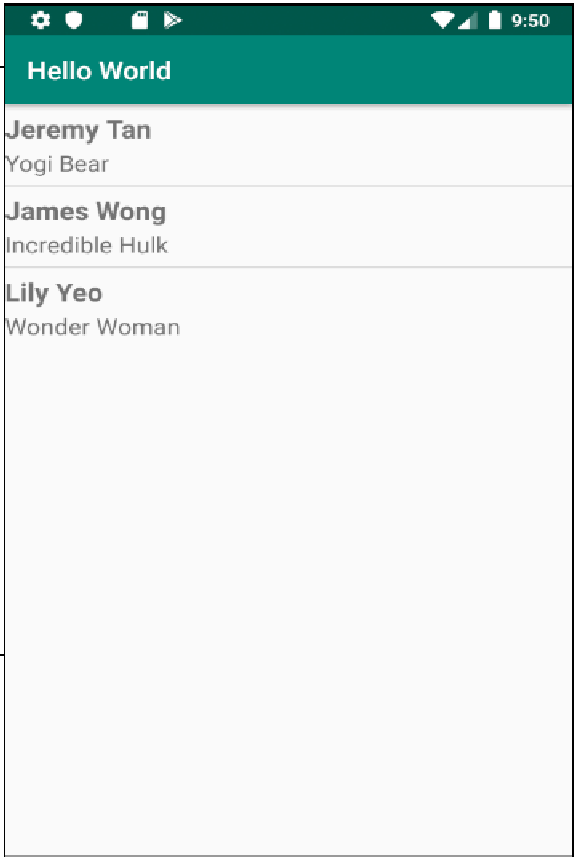
SimpleAdapter

- SimpleAdapter allows binding **fields** to **text views**

```
@Override
public void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.my_list_activity);

    List<Person> list = new ArrayList<Person>();
    list.add(new Person("Jeremy Tan", "Yogi Bear"));
    list.add(new Person("James Wong", "Incredible Hulk"));
    list.add(new Person("Lily Yeo", "Wonder Woman"));

    ListView listView = findViewById(R.id.listView1);
    listView.setAdapter(new SimpleAdapter(this, list, R.layout.row2,
            new String[] {"name", "nickname"},
            new int[] { R.id.textView1, R.id.textView2}));
}
```



Hello World

Jeremy Tan
Yogi Bear

James Wong
Incredible Hulk

Lily Yeo
Wonder Woman

Enhancing Adapter (via inheritance)

- Inherits from existing adapter and overriding **getView()** to return a row for display

```
public class MyAdapter extends ArrayAdapter<String> {
    private List<String> items;
    int resource;
    public MyAdapter(Context context, int resource, List<String> items) {
        super(context, resource, items);
        this.resource = resource;
        this.items = items;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        LayoutInflater inflater = (LayoutInflater) getContext()
                .getSystemService(Activity.LAYOUT_INFLATER_SERVICE);
        View v = inflater.inflate(resource, null);
        String eid = items.get(position);
        if (eid != null) {
            TextView e = v.findViewById(R.id.textView);
            e.setText(eid);
        }
        return v;
    }
}
```

Active Recall

Toast
ListView
Event Handler
onClick
interface
View.OnClickListener
SimpleAdapter
Context Menu
Data Adapter