**Spinner: dropdown menu of selectable strings**

<resources>

<string-array name=”nam”>

<item> hello </item>

<item> hey </item>

…

</ string-array>

</resources >

**Drag & drop then edit attributes in XML:**

-android:clickable=”bool”

-android:entries=”@array/nam” Must be in strings.xml:

**Configure events in java:**

Spinner spin = (Spinner) findViewByID(R.id.spinnerID);

spin.**setOnItemSelectedListener**( new AdapterView.**OnItemSelectedListener**(){

public void **onItemSelected**(AdapterView<?> sp, View v, int I, long id){

//do stuff using I as index of selected

}

public void **onNothingSelected**(AdapterView<?> sp){

//usually empty

}

}

);

**ListView: ordered collection of selectable choices**

**ArrayList<String> words = new ArrayList<String>();  
words.add...  
  
ArrayAdapter<String> adapter =**

**new ArrayAdapter<String>(this, android.R.layout.*simple\_list\_item\_1*, words);  
  
ListView list = (ListView) findViewById(R.id.*mylist*);  
list.setAdapter(adapter);  
  
list.setOnItemClickListener( *//Can also be ItemLongClick and ItemSelected*   
 new AdapterView.OnItemClickListener() {  
 public void onItemClick(AdapterView<?> parent, View view, int position, long id) {  
 *//Do stuff*  
 }**

**}**

**);**

**Intent: switch to other activity and carry data with you**

**Intent Intnt = new Intent(this, targetActivity.class);  
Intnt.putIntegerArrayListExtra("hello",list);  
Intnt.putStringArrayListExtra("list",slist);**

**Intnt.putExtra("name",int/string/etc…);  
startActivity(Intnt);**

**On the other side:**

**Intent Intnt = getIntent();  
ArrayList<String> plist = Intnt.getStringArrayListExtra("list");**

**Object thing = Intnt.getExtra("name");**

**Activity State: set of functions that run depending on what the activity is doing, can be over-written**

**onCreate(){} onPause(){} onStart(){} onRestart(){}**

**onDestroy(){} onResume(){} onStop(){}**

**Shared Preferences: object that stores permanent settings for app**

**SharedPreferences pref = getPreferences(Context.*MODE\_PRIVATE*);**

**//Write: //Read:  
SharedPreferences.Editor edt = pref.edit(); int i = pref.getInt("key",0);  
edt.putInt("key",88);  
edt.putString("alsokey","hello"); Default Value**

**edt.commit();**

**Toggle Button:**

**ToggleButton TG = findViewById(R.id.*toggleButton*);  
TG.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
  *//use flags to check if toggled on/off*   
 }  
});**

**Switch:**

**Switch SW = findViewById(R.id.*switch*);  
SW.setOnCheckedChangeListener(new CompoundButton.OnCheckedChangeListener() {  
 @Override  
 public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {  
 if(isChecked){  
 //…  
 }  
 else{  
 //…  
 }  
 }  
});**

**Toolbars:**

**Note:** turn off default toolbar from manifest: android:theme="@style/Theme.AppCompat.Light.NoActionBar"

**import android.support.v7.widget.Toolbar;**

**//INSIDE ONCREATE:**

**final MainActivity thisthing = this; //to reference current activity if needed (Toast or Intent)**

**Toolbar myToolbar = findViewById(R.id.*toolbar*); //Toolbar widget**

**setSupportActionBar(myToolbar);  
 myToolbar.setOnMenuItemClickListener(new Toolbar.OnMenuItemClickListener() {  
 @Override  
 public boolean onMenuItemClick(MenuItem menuItem) {  
 if(menuItem.getItemId() == R.id.*login*){   
 *//do stuff*  
 }  
 else{**

***NOTE:***

To get resource values use:  
**getResources.getString**(R.string.name);

**getResources.getArray**(R.array.name);

***//do other stuff*  
 }  
 return false;  
 }  
 });**

**@Override  
public boolean onCreateOptionsMenu(Menu menu){  
 MenuInflater inflater = getMenuInflater();  
 inflater.inflate(R.menu.*menu*,menu); *//Create a folder ‘menu’ in ‘res’ with Menu Resource***

***File ‘menu’*  
 return super.onCreateOptionsMenu(menu);  
}**

**Files:**

**try {**

**bf2 = new BufferedWriter(**

**new FileWriter(**

**new File(**

**getFilesDir() + "name.txt")));  
  
 bf2.append("…");  
 bf2.newLine();**

**bf2.close();  
}**

**catch(FileNotFoundException e){  
 e.printStackTrace();  
}  
catch(IOException e1){  
 e1.printStackTrace();  
}**

**try {  
 bf = new BufferedReader(**

**new FileReader(**

**new File(**

**getFilesDir() + "name.txt")));  
  
 String s;  
 while ((s = bf.readLine()) != null){**

**//split s based on how we saved to the file**

**}  
 bf.close();  
}  
catch(IOException e1){  
 e1.printStackTrace();**  
**}**

**Notifications:**

**public void addNotif(View view) {**

**Intent intent = new Intent(Intent.ACTION\_VIEW, Uri.parse("http://www.google.com"));**

**PendingIntent pending = PendingIntent.getActivity(this, 0, intent, 0);**

***//Notification.Action action =***

***// new Notification.Action.Builder(R.drawable.ICON, "This is an action", pending).build();***

**Notification.Builder builder = new Notification.Builder(this)**

**.setContentTitle("title")**

**.setContentText("text")**

**.setAutoCancel(true)**

**.setSmallIcon(R.drawable.logomini)**

**.setContentIntent(pending);**

***//.addAction(action);***

**notification = builder.build();**

**NotificationManager manager = (NotificationManager)getSystemService(Context.NOTIFICATION\_SERVICE);**

**manager.notify(1221, notification);**

**}**

**public void removeNotif(View view) {**

**if(notification == null){**

**return;**

**}**

**NotificationManager manager = (NotificationManager)getSystemService(Context.NOTIFICATION\_SERVICE);**

**manager.cancel(1221);**

**}**

**Fragments:**

**public class FragmentOne extends Fragment {**

**@Override**

**public View onCreateView(LayoutInflater inflater,ViewGroup viewGroup, Bundle savedInstanceState) {**

**View view = inflater.inflate(R.layout.fragment\_one, viewGroup, false);**

**.**

**.**

**.**

***//use v.findViewByID(…) to change objects inside the fragment***

**.**

**return view;**

**}**

**}**

**public class MainActivity extends FragmentActivity{**

**@Override**

**public void onCreate(Bundle savedInstanceState) {**

**.**

**.**

**.**

**Fragment fragment = new FragmentOne();**

**FragmentManager manager = getSupportFragmentManager();**

**FragmentTransaction transaction = manager.beginTransaction();**

**transaction.replace(R.id.output, fragment);**

**transaction.commit();**

***//^use this when switching fragments***

**}**

**SQLite: We use a DBHandler to access the database**

**public class DBHandler extends SQLiteOpenHelper {**

**public DBHandler(Context context){**

**super(context,"UserList",null,1);**

**}**

**@Override**

**public void onCreate(SQLiteDatabase db) {**

**String createDB = "CREATE TABLE tableName (" +**

**"id INTEGER PRIMARY KEY AUTOINCREMENT," +**

**"someString TEXT," +**

**"someeInt INTEGER )";**

**db.execSQL(createDB);**

**}**

**@Override**

**public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {**

**db.execSQL("DROP TABLE IF EXISTS tableName");**

**onCreate(db);**

**}**

**//Getting**

**public List<ITEM> getList(){**

**SQLiteDatabase db = this.getReadableDatabase();**

**List<ITEM> thelist = new ArrayList<>();**

**String selectQuerry = "SELECT \* FROM tableName";**

**Cursor cursor = db.rawQuery(selectQuerry,null);**

**if(cursor.moveToFirst()){**

**do{**

**ITEM item = new ITEM(**

**cursor.getInt(0),**

**cursor.getString(1),**

**cursor.getString(2)**

**);**

**thelist.add(item);**

**}while(cursor.moveToNext());**

**}**

**cursor.close();**

**db.close();**

**return thelist;**

**}**

**public void add(String someString, int someInt){**

**SQLiteDatabase db = this.getWritableDatabase();**

**ContentValues values = new ContentValues();**

**values.put("someString", someString);**

**values.put("someInt", someInt);**

**db.insert("tableName",null,values);**

**}**

**public void edit(int ID, String someString, int someInt){**

**SQLiteDatabase db = this.getWritableDatabase();**

**ContentValues values = new ContentValues();**

**values.put("someString", someString);**

**values.put("someInt", someInt);**

**db.update("tableName",values,"id = " + ID,null);**

**}**

**}**

**Broadcast Receiver: Listens to system-wide events & intents**

**public class MyReceiver extends BroadcastReceiver {**

**public MyReceiver() {**

**}**

**@Override**

**public void onReceive(Context context, Intent intent) { *//intent is the received broadcast***

**Toast.makeText(context, "Action: " + intent.getAction(), Toast.LENGTH\_LONG).show();**

**}**

**}**

**Note: we can define the receiver in the AndroidManifest or dynamically**

**Mainfest Method:**

**<application>**

**…**

**<receiver android:name="myReceiver" android:enabled="true" android:exported="true">**

**<intent-filter>**

**<action android:name="android.net.conn.CONNECTIVITY\_CHANGE"/>**

**<action android:name="android.net.wifi.WIFI\_STATE\_CHANGED"/>**

**</intent-filter>**

**</receiver>**

**</application>**

**Dynamic Method:**

**public class MainActivity extends AppCompatActivity {**

**@Override**

**protected void onCreate(Bundle savedInstanceState) {**

**…**

**IntentFilter filter = new IntentFilter("android.net.conn.CONNECTIVITY\_CHANGE");**

**myBroadcastReceiver myReceiver = new myBroadcastReceiver();**

**registerReceiver(myReceiver, filter);**

**}**

**}**

**Some filters include:**

**-** **android.intent.action.AIRPLANE\_MODE - android.net.wifi.WIFI\_STATE\_CHANGED**

**- android.intent.action.BATTERY\_LOW - android.net.conn.CONNECTIVITY\_CHANGE**

**- android.intent.action.BOOT\_COMPLETED - android.intent.action.DATE\_CHANGED**

**Example:**

**public class myBroadcastReceiver extends BroadcastReceiver {**

**@Override**

**public void onReceive(Context context, Intent intent) {**

**if (intent.getAction().equals("android.net.conn.CONNECTIVITY\_CHANGE")) {**

**ConnectivityManager cm = (ConnectivityManager) context.getSystemService(Context.CONNECTIVITY\_SERVICE);**

**NetworkInfo activeNetwork = cm.getActiveNetworkInfo();**

**boolean isConnected = activeNetwork != null && activeNetwork.isConnected();**

**if (isConnected) {**

**boolean isWiFi = activeNetwork.getType() == ConnectivityManager.TYPE\_WIFI;**

**try {**

**if (isWiFi) {**

**WifiManager mainWifi = (WifiManager) context.getSystemService(Context.WIFI\_SERVICE);**

**WifiInfo currentWifi = mainWifi.getConnectionInfo();**

**String name = currentWifi.getSSID();**

**String m = currentWifi.getMacAddress();**

**Toast.makeText(context, "connected to" + name, Toast.LENGTH\_LONG).show();**

**} else if (activeNetwork.getType() == ConnectivityManager.TYPE\_MOBILE) {**

**{**

**Toast.makeText(context, "Connected to 3G", Toast.LENGTH\_LONG).show();**

**}**

**}**

**} catch (Exception e) {…}**

**} else {*//No connection* }**

**}**

**}**

**}**

**Async Tasks: Perform heavy tasks in the background**

**First we need a class to extend AsyncTask:**

**private class AsyncTaskRunner extends AsyncTask<String, String, String>{**

**String resp;**

**@Override**

**protected String doInBackground(String... params) {**

**publishProgress("Doing stuff.."); *// Calls onProgressUpdate()***

**try {**

***//put final status in resp***

**}**

**catch (InterruptedException e) { e.printStackTrace(); resp = e.getMessage(); }**

**return resp;**

**}**

**@Override**

**protected void onPostExecute(String result) {**

***//after doInBackground, receives res from it***

**}**

**@Override**

**protected void onPreExecute() {**

***//before doInBackground***

**}**

**@Override**

**protected void onProgressUpdate(String... text) {**

***//handle updates from doInBackground***

**}**

**}**

**To begin the task (which can only be called once) we should create an instance of our task and execute it:**

**AsyncTaskRunner runner = new AsyncTaskRunner();**

**runner.execute(…);**