

## Interaction with Android

The Android program can output data by working with the xml files. The main use of the java files is to get some data, make decisions and change the output based on the results of those decisions.

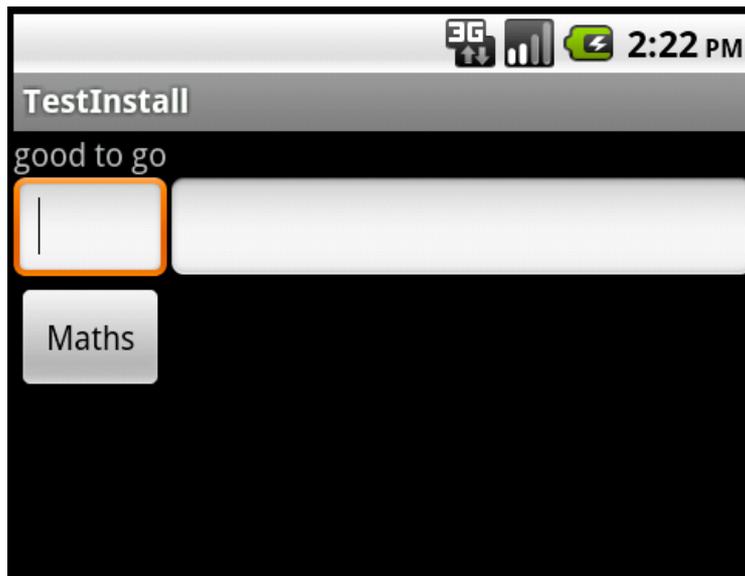
To keep the problem simple we will use

- 2 EditText widgets so 2 things can be compared
- A TextView for the output
- At least 1 Button to fire the code.

The following code shows the XML to set up a table layout of rows and columns. The basic objects have been set up in the graphical editor followed by some heavy tweaking of the xml. The xml tags need to be properly closed and nested or the layout will not work.

```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/tableLayout1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:stretchColumns="1" >

    <TableRow >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello"
        android:id="@+id/txtOut"/>
    </TableRow>
    <TableRow >
        <EditText android:id="@+id/tLeft">
            <requestFocus></requestFocus>
        </EditText>
        <EditText android:id="@+id/tRight" >
        </EditText>
    </TableRow>
    <TableRow>
        <Button android:text="Maths"
            android:id="@+id/bMaths"
            android:layout_height="wrap_content" />
    </TableRow>
</TableLayout>
```



The EditText Objects are added to the xml and instantiated in the java in the same way as the TextView.

The Button is set up in the same way but needs registering with an onClickListener to be able to act on a click. The following lines appear to throw an error in Android because they are missing some linked objects

```
bMaths = (Button) findViewById(R.id.bMaths);
bMaths.setOnClickListener(this);
```

The imports section needs expanding to include a view.

```
import android.view.View;
```

The line that declares that the java program is an activity needs expanding to state that it is also (inherits from) a View.OnClickListener.

```
public class inputOutput extends Activity implements
View.OnClickListener
```

This code will ask for an onClick method to be implemented.

```
public void onClick(View v) {
}
```

This method can then pick up the click event on the Button.

```
public void onClick(View v) {
    if (v == findViewById(R.id.bMaths))
    {
        tOutput.setText("OK so far");
    }
}
```

This code will output a test line to the TextField tOutput.

The data from the EditText objects can be assigned to String variables:

```
String left, right, result;  
left = eLeft.getText().toString();  
right = eRight.getText().toString();
```

This is not a lot of use if any maths is going to have to take place. Integer.parseInt can solve that problem.

```
iLeft = Integer.parseInt(left);
```

There are similar classes for Float and Double

```
float fRight = Float.valueOf(right);  
double dRight = Double.valueOf(right);
```

There is not a lot of call for doing maths on Strings (although the + operator will concatenate them) but Android has a pile of String manipulation methods to compare Strings or find Strings within other Strings.

The following are amongst the most useful:

char charAt (int index)	Returns the character at the specified offset in this string.
int compareTo (String string)	Returns 0 if two Strings exactly the same. This is a better plan than == which can cause problems
int compareToIgnoreCase (String string)	A useful variant of the above
String copyValueOf (char[] data, int start, int length)	Copy some or all of a char array into a String
boolean endsWith (String suffix)	Does a String end in a set suffix?
int indexOf (int c)	Find a char in a String, -1 is returned if it is not present
int length ()	How long is a String?
String replace (char oldChar, char newChar)	Copies this string replacing occurrences of the specified character with another character.
String substring (int start, int end)	Returns a string containing a subsequence of characters from this string
public char[] toCharArray ()	Copies the characters in this string to a character array.
String toLowerCase ()	Make the String lower case
String toUpperCase ()	Make the String upper case
String trim ()	Trim leading and trailing whitespace

For example, adding up is easy:

```
String left, right;
left = eLeft.getText().toString();
right = eRight.getText().toString();
String result = "words";
result = left + right;
tOutput.setText(result);
```

Some String method examples

```
int num = left.length();
tOutput.setText(String.valueOf(num));
```

One more

```
if (left.length() > 5)
    result = left.substring(1, 4);
else
    result = "too short";
tOutput.setText(result);
```

This is the way to go rather than using = with Strings

```
int num = left.compareTo(right);
if (num == 0)
    result = "left and right are the same";
else
    result = "left and right are different";
```

Another

```
if (input.indexOf("a") != -1)
    txtOut.setText("a is at position " +
        input.indexOf("a") + " in " + input);
else
    txtOut.setText("there is no 'a' in " + input);
```

This is all fine but we cannot be sure that the user has entered a word or a number. The easiest way is to set the input type of the EditText in xml. The settings can be browsed from the properties list of the visual designer.

```
<EditText android:id="@+id/tLeft"
    android:inputType="number">
```

Of course the java code can check what a char data type contains. We know that a String is a string of char data so can work from there.

```
String right;
String result="data";
right = eRight.getText().toString();
char[] letters = right.toCharArray();
for (int x=0;x<letters.length;x++)
{
    if(! Character.isDigit(letters[x]))
        result = "not a number";
}
```

The Character class has a number of similar methods including isLetter(), isLetterOrDigit() and isWhitespace(),