Lecture 0 – Android Cheat Sheet

# Creating Views

## Buttons

<**Button  
 android:id="@+id/toastButton"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
  
 app:layout\_constraintTop\_toTopOf="parent"  
 app:layout\_constraintBottom\_toBottomOf="parent"  
  
 android:text="@string/toast\_text"  
 android:background="@color/toast\_color"  
 android:layout\_margin="10dp"  
  
 android:textColor="@color/toast\_text\_color"** />

This will create a centred button (constraint to top of parent and bottom of parent). Experiment with margins, layout and colours!

You can add a listener that refers to the method that you want to run onClick in XML too. Then you don’t have to instantiate your button in code – DON’T do this if you need to dynamically generate your buttons e.g. in a fragment:

**android:onClick="doSomething"**

In your main method:

**public void** doSomething(View view)  
{  
 *//Do something on click!*}

## EditText

EditText are used as text inputs for the user. We can call methods to get the text from the edittext on events such as button clicks (for example, a submit button).

<**EditText  
 android:id="@+id/enterText"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintBottom\_toTopOf="@+id/toastButton"  
 app:layout\_constraintTop\_toTopOf="parent"  
 android:paddingLeft="10dp"** />

The example above is constrained to the toastButton. You should constrain it whatever you like!

## Getting Text From an EditText

You first need to find load your EditText element as a member variable of your class (e.g. in your main class). Use the element ID defined from the element XML to inflate the view in the OnCreate function.

**private** EditText **mMyEditText**;

@Override  
**protected void** onCreate(Bundle savedInstanceState)  
{  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 **mMyEditText** = findViewById(R.id.***enterText***);  
 **myTextOutput** = findViewById(R.id.***textOutput***);  
}

To get the text from the editText as a string, use

String input = **mMyEditText**.getText().toString();