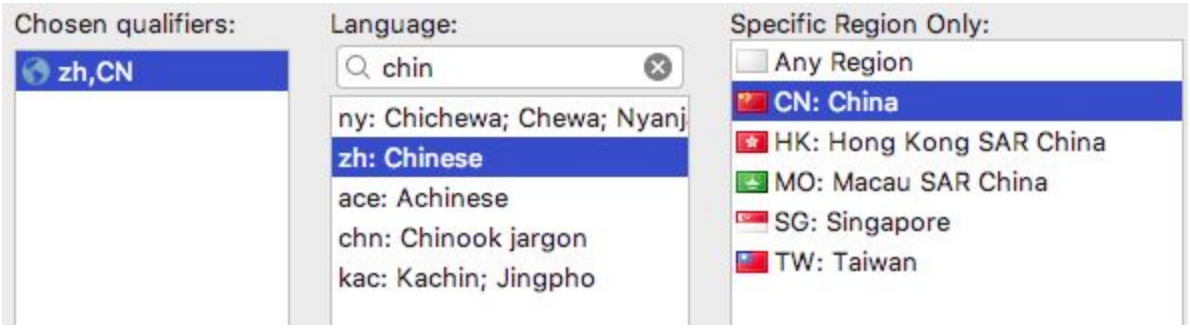
Hello Button

Step #1, so a demo of the solution!!!
Create a new project called **Hello Button** as an **Empty Activity**
Note: Don't have them mess with Git at all but share with them the organization where you keep your work. For me add the root as needed.
MainActivity is fine. Leave the checkboxes as the defaults.
Do the whole "David Fisher can make apps" and run the emulator (and a device) **Move icon**
Let's go! → Copy / paste the given strings.xml or **JUST type them**

res/values/strings.xml

```
<resources>
<string name="app_name">Hello Button</string>
<string name="message_start">Count = 0</string>
<string name="message_format">Count = %d</string>
<string name="button_decrement">-1</string>
<string name="button_reset">Reset</string>
<string name="button_increment">+1</string>
</resources>
```

Next make a new strings.xml for zh CN (right click res folder)



Copy in the strings.xml contents and edit it some for "Chinese"

res/values-zh-rCN/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<string name="app_name">Hello China</string>
<string name="message_start">x = 0</string>
<string name="message_format">x = %d</string>
<string name="button_decrement">-1</string>
<string name="button_reset">Reset!!!</string>
<string name="button_increment">+1</string>
</resources>
```

Add some color resources (just type these too):

colors.xml

```
<color name="background">#ff00aa00</color>
<color name="text">#dfd</color>
```

activity_main.xml (highlights so things you might miss)

```
<RelativeLayout
… stuff that doesn't change ...
  android:background="@color/background"
  tools:context=".MainActivity">

  <LinearLayout
     android:layout_width="match_parent"
     android:layout_height="wrap_content"
     android:layout_margin="8dp"
     android:orientation="horizontal">
     <Button
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/button_decrement"/>
     <Button
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/button_reset"/>
     <Button
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="@string/button_increment"/>
  </LinearLayout>

  <TextView
     android:id="@+id/message_text_view"
     android:layout_width="wrap_content"
     android:layout_height="wrap_content"
     android:layout_centerHorizontal="true"
     android:layout_centerVertical="true"
     android:text="@string/message_start"
     android:textColor="@color/text"
     android:textSize="32sp"/>
</RelativeLayout>
```

Run it in both English and Chinese!

Capture the text view in code. **First turn on auto import checkbox preferences!!!!**

MainActivity.java

```
public class MainActivity extends AppCompatActivity {
 private int mCount = 0;
 private TextView mTextView;

 @Override
 protected void onCreate(Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.activity_main);
   mTextView = findViewById(R.id.message_text_view);
 }
}
```

Do a quick test with

```
mTextView.setText("This is just a test");
```

Then remove it

```
// mTextView.setText("This is just a test");
```

Next add a Log message.

```
Log.d("HB", "********** Console logs are VERY useful!");
```

show that you can filter by your log message TAG

Next cause a crash on purpose

```
mTextView = findViewById(R.id.message_text_view);
//mTextView.setText("This is just a test");
//Log.d("HB", "********** Console logs are VERY useful!");
mTextView = findViewById(0);
mTextView.setText("This will cause a crash on purpose");
```

Take your time talking about the logs. Make sure the TAG filter is off. Then fix your mess. :)

```
   mTextView = findViewById(R.id.message_text_view);
//    mTextView.setText("This is just a test");
//    Log.d("HB", "********** Console logs are VERY useful!");
//    mTextView = findViewById(0);
//    mTextView.setText("This will cause a crash on purpose");
```

Next add onClick attributes for the buttons

main_activity.xml

```
<Button...   android:onClick="pressedDecrement"/>
<Button...   android:onClick="pressedReset"/>
<Button...   android:onClick="pressedIncrement"/>
```

MainActivity.java

```
public void pressedIncrement(View view) {
 mCount++;
 updateView();
}

public void pressedReset(View view) {
 mCount = 0;
 updateView();
}

public void pressedDecrement(View view) {
 mCount--;
 updateView();
}

private void updateView() {
 mTextView.setText(getString(R.string.message_format, mCount));
}
```

Challenge for the students…
Make the text invisible whenever mCount > 10 and visible below 10. Given hint...
mTextView.setVisibility(View.INVISIBLE)

```
private void updateView() {
 mTextView.setText(getString(R.string.message_format, mCount));

 // Easter egg
 if (mCount > 10) {
   mTextView.setVisibility(View.INVISIBLE);
 } else {
   mTextView.setVisibility(View.VISIBLE);
 }
}
```