Welcome to Android!

Android Basics

- **Activities**
  - A class responsible for managing user interactions
  - Activities respond to user interactions

- **Layout**
  - Defines a set of UI objects and how they are positioned on the screen
  - Created by writing definitions in XML
  - XML definitions are used to create objects on the user's screen
  - Layouts are stored in their own .xml files.

Create Our First App: The Quizzler

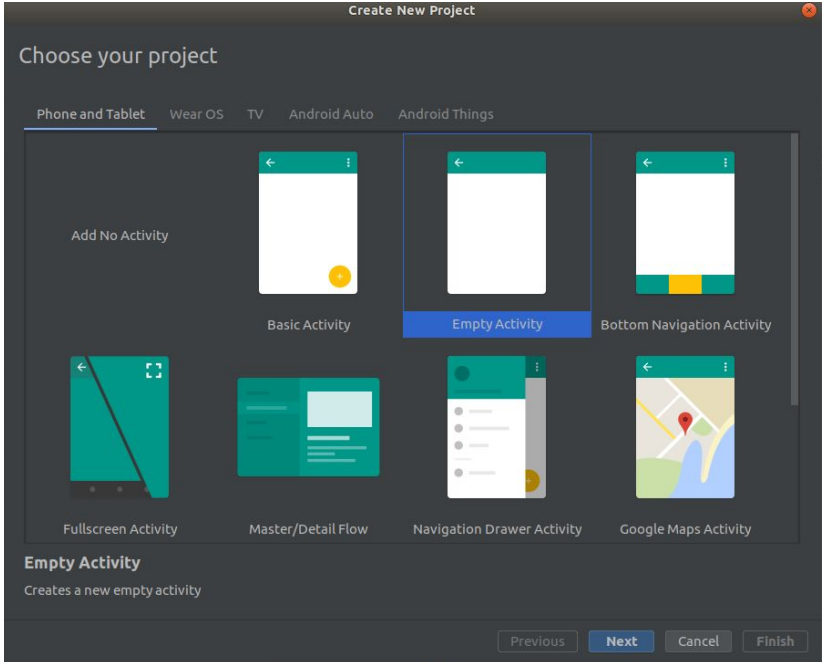
Let's create a simple app that will have one activity and one layout.

The XML layout will control the view of the app and buttons will represent an activity.

Open Android Studio and Start a New Project

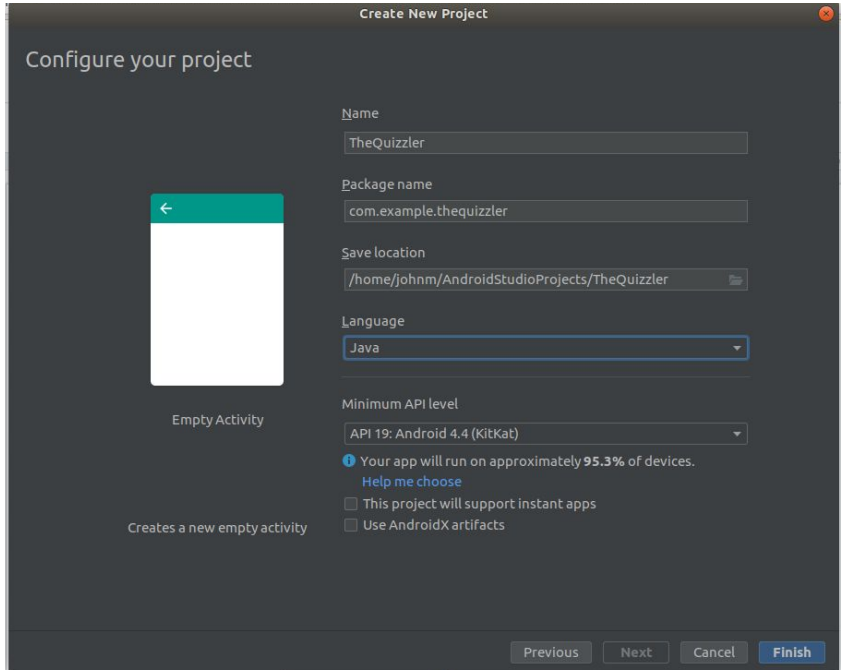
Maydm

Create an Empty Activity



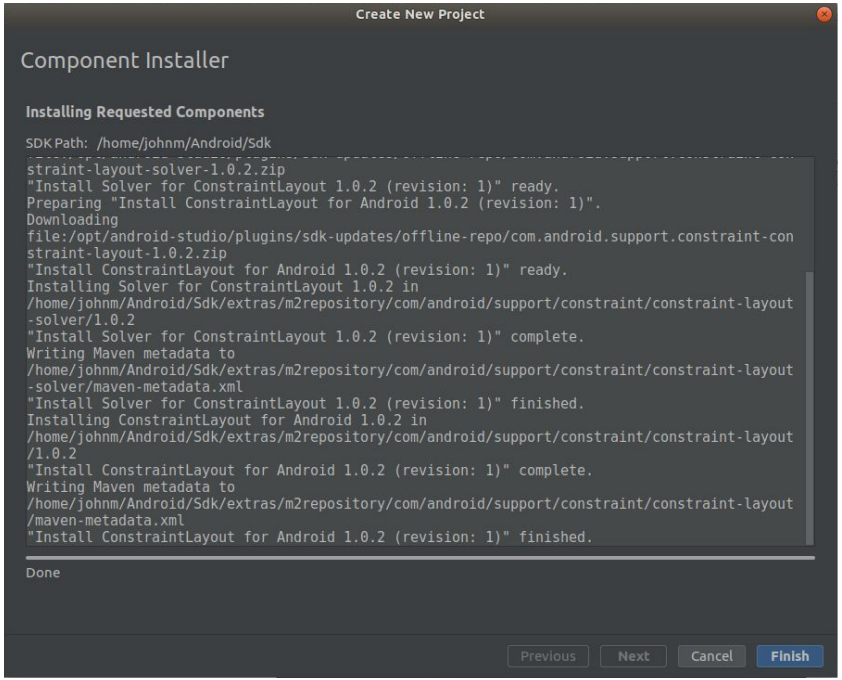
Maydm

Configure the Project Name



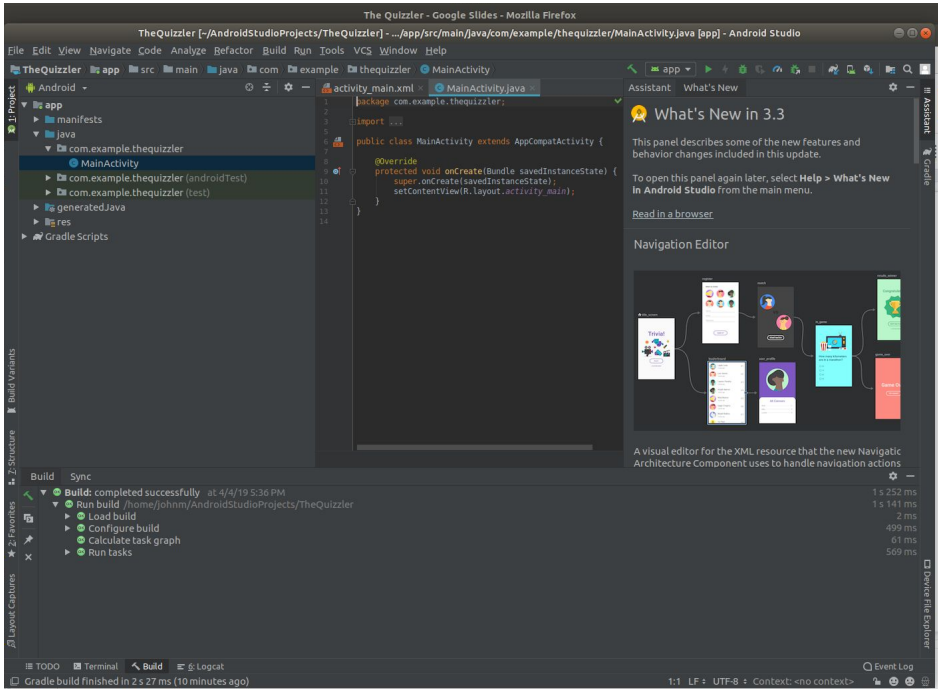
Maydm

Finish after Component Install Concludes



Maydm

Allow Project to Load and Configure

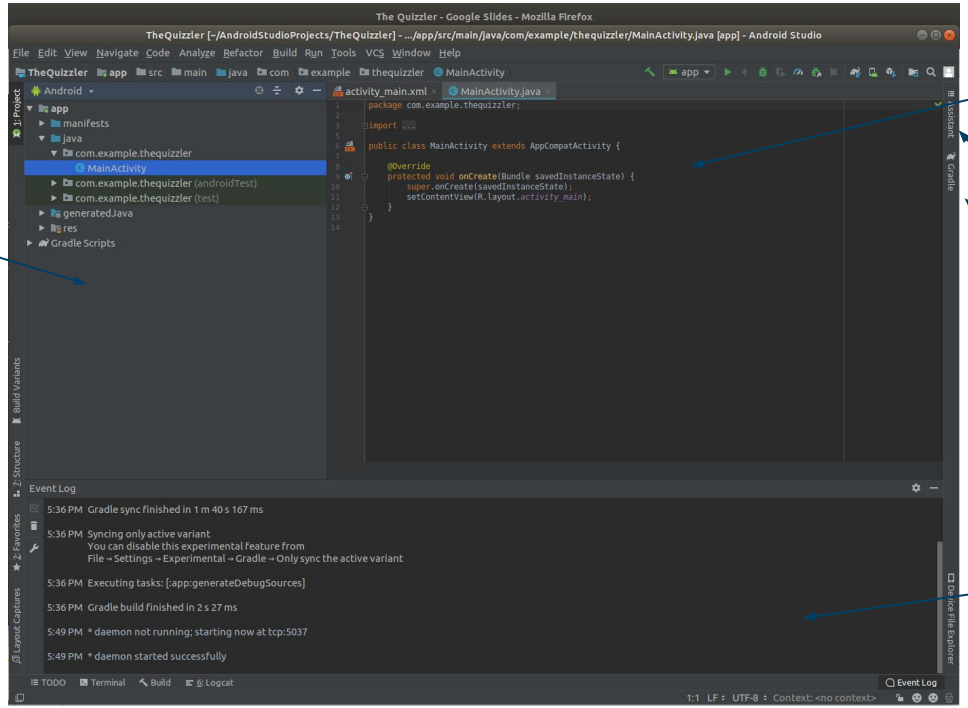


Maydm

Menus in Android Studio

Different panes are called tool windows

Project Tool Window: view and manage project files



Editor

Assistant menu

Gradle menu

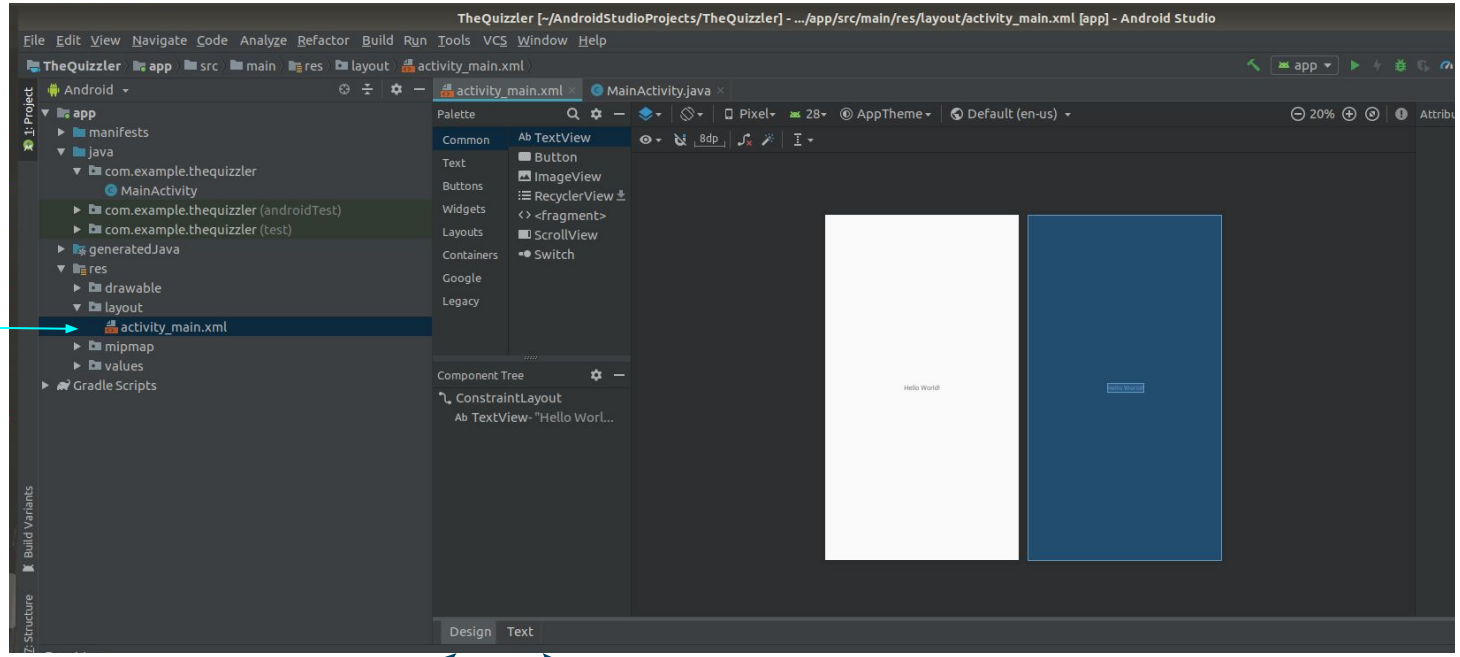
Event Log

Maydm

Find the XML Layout File

Navigate to app/res/layout/activity_main.xml to see a graphic preview of the app

Project directories and files



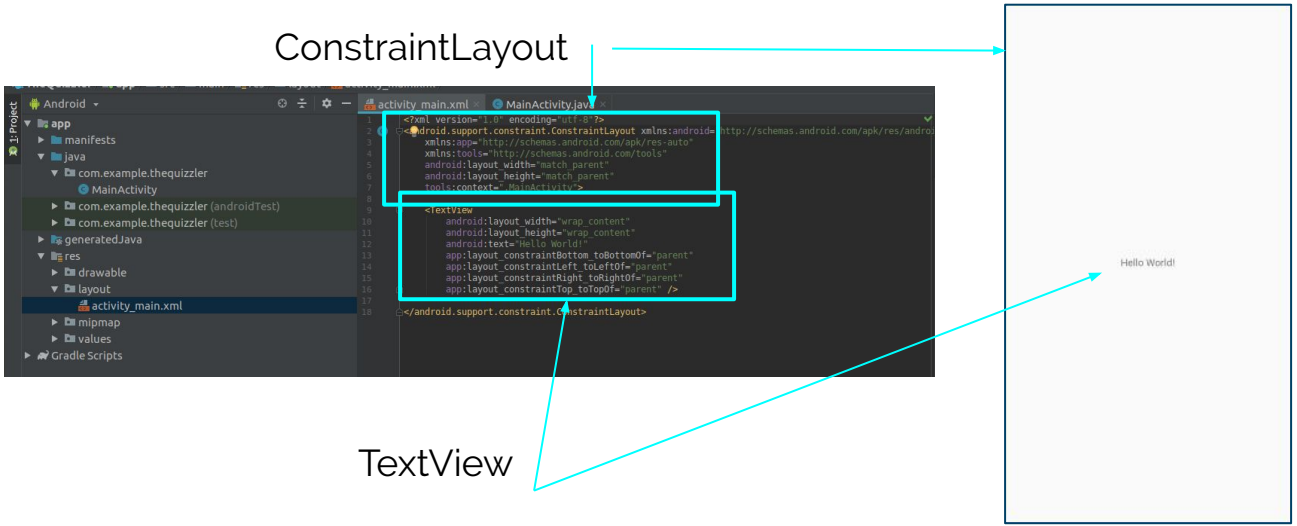
Design view | Text view of XML

Maydm

Widgets

- Building blocks used to compose (create) the UI (user interface)
- Widgets can show text or graphics
- Can be interactive
- Widgets can control and arrange other widgets
- Examples of Widgets:
    - Buttons
    - Text input controls
    - Checkboxes
- Every widget is an instance of the View class

The ConstraintLayout and TextView



ConstraintLayout and TextView are defined by the default activity layout.
Meaning, they come bundled with a new basic app.
Both ConstraintLayout and TextView are widgets.

Maydm

Widgets in The Quizzler

- **We will use five (5) widgets in the MainActivity interface**
  - A vertical **LinearLayout**
  - A **TextView**
  - A horizontal **LinearLayout**
  - Two **Buttons**

Write Your First XML in activity_main.xml

Add a TextView and two Button widgets after the ConstraintLayout XML and before the closing ConstraintLayout tag.

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
 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"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:orientation="vertical"
   tools:context=".MainActivity">
```

<--Place this code here -->

```
</android.support.constraint.ConstraintLayout>
```

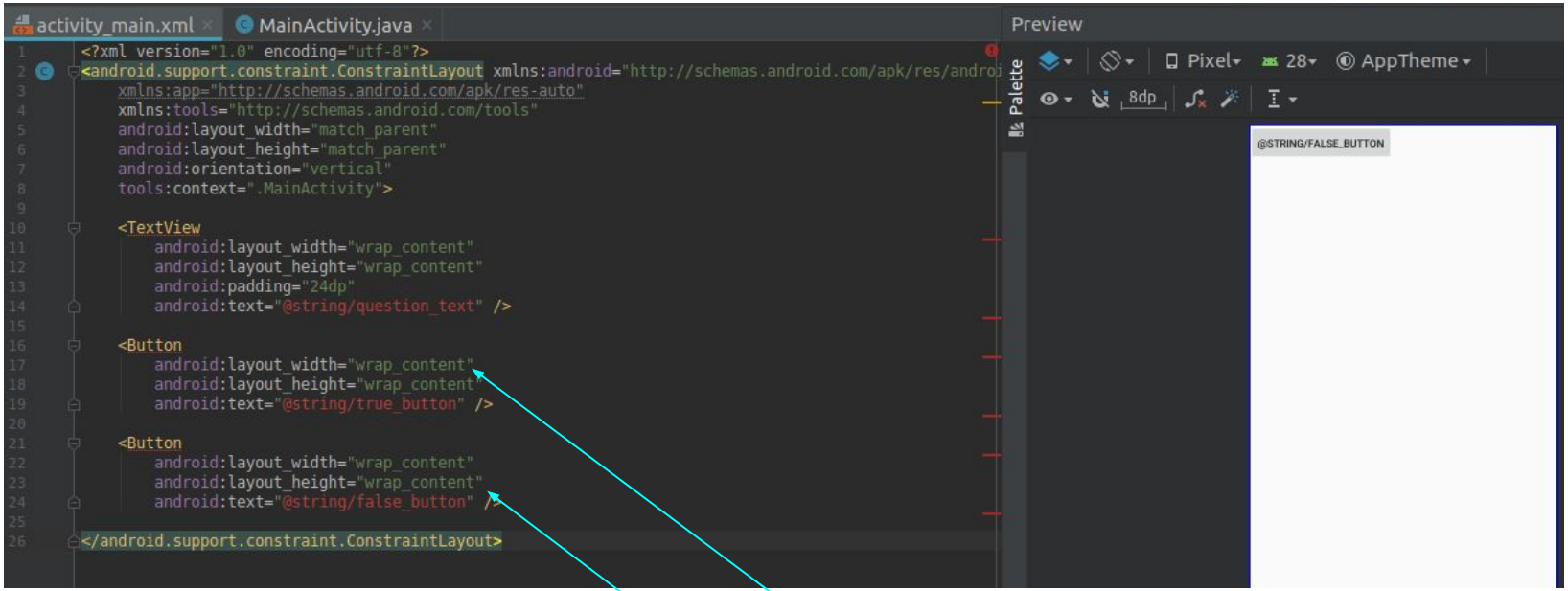
```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:padding="24dp"
    android:text="@string/question_text" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/true_button" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/false_button" />
```

Maydm

Review the XML in activity_main.xml



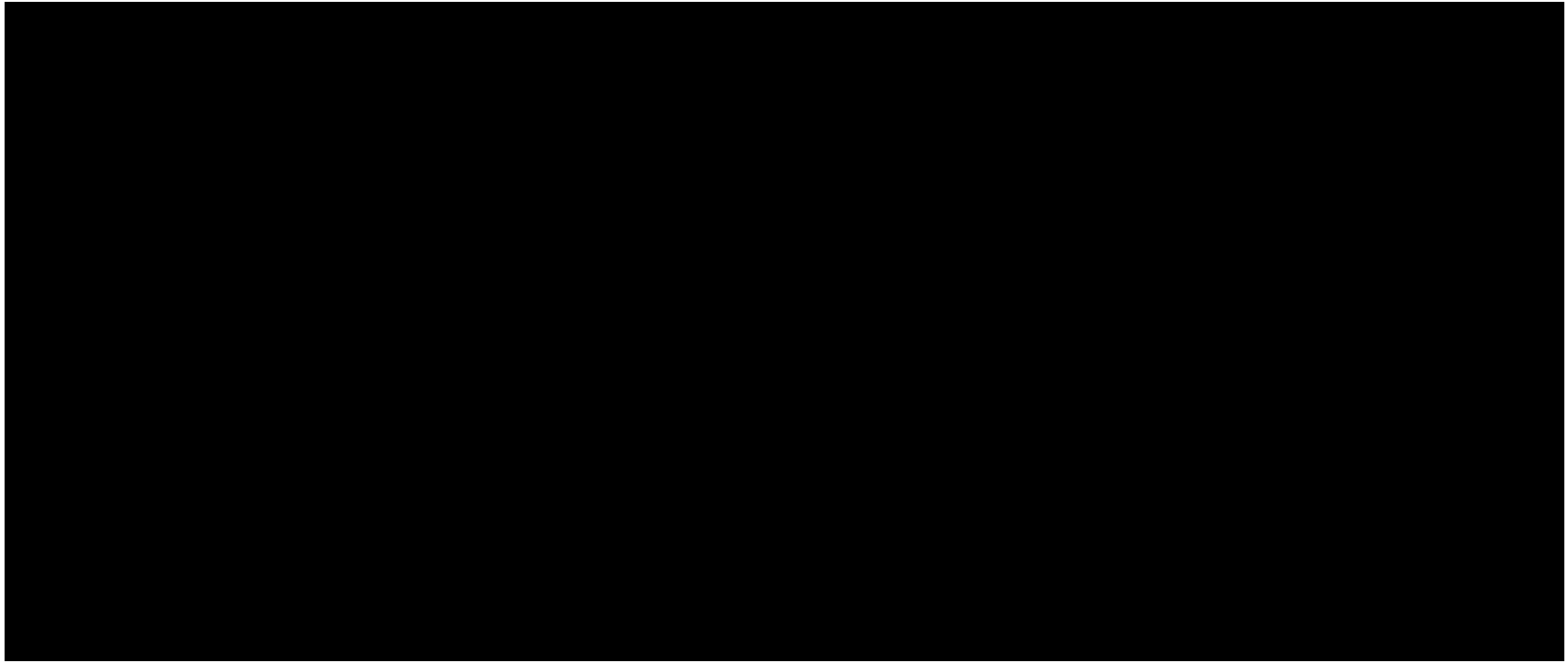
XML

Widget Attributes

ConstraintLayout

Maydm

Constrain the widgets in the ConstraintLayout



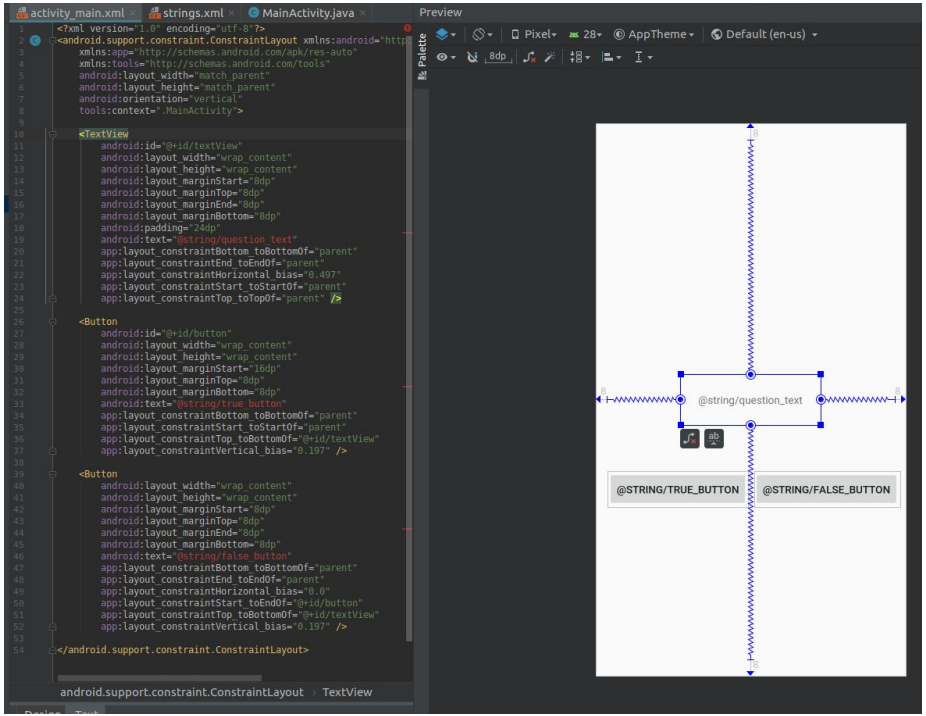
Maydm

Constrain the widgets in the ConstraintLayout

Follow the example in the video.
As we add constraints new attributes are added to the widgets.
Android has developed a GUI (Graphic User Interface) with the ConstraintLayout so that apps are easier for developers to design.

We can edit the XML attributes if needed, but this is not required.

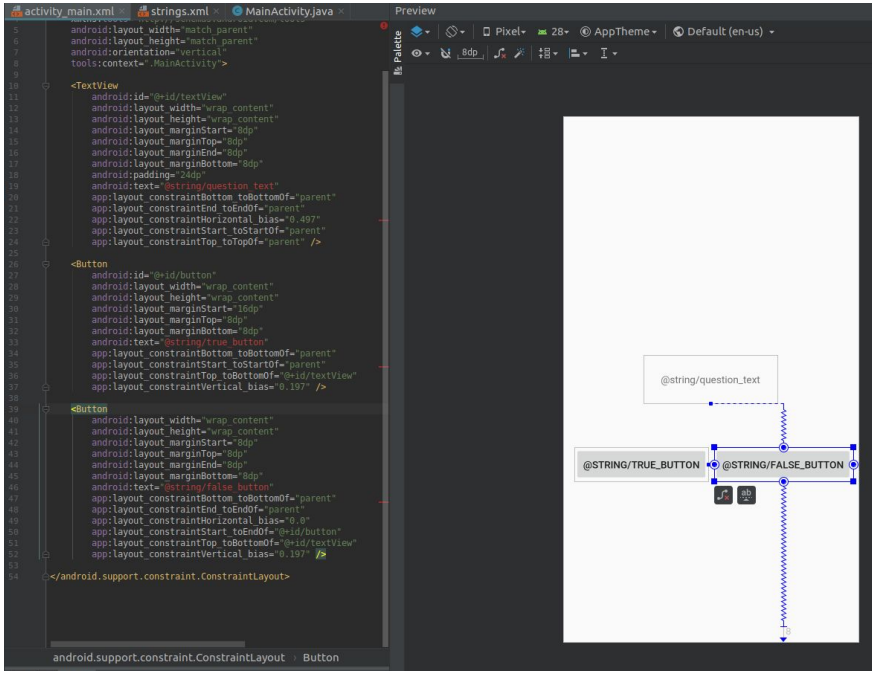
Using the GUI can save lots of time in App development!



Maydm

Constrain the widgets in the ConstraintLayout

Add constraints on all four sides of each widget to create defined widgets that will appear consistently on Android devices despite their size.



Maydm

Widget Attributes

- `android:layout_width` and `android:layout_height`
  - Required for almost every widget
  - Usually set to `match_parent` or `wrap_content`
- `match_parent` will create a view that is as big as its parent
- `wrap_content` will be as large as its contents require

Widget Attributes

- `android:padding="24dp"` attribute adds padding around the text in the TextView widget.
- `android:orientation` determines if a child will be vertical or horizontal
- `android:text` instructs the widget to display a string

String Resources

String Resources are strings that exist in an XML file of their own. This file is called a *strings file*.

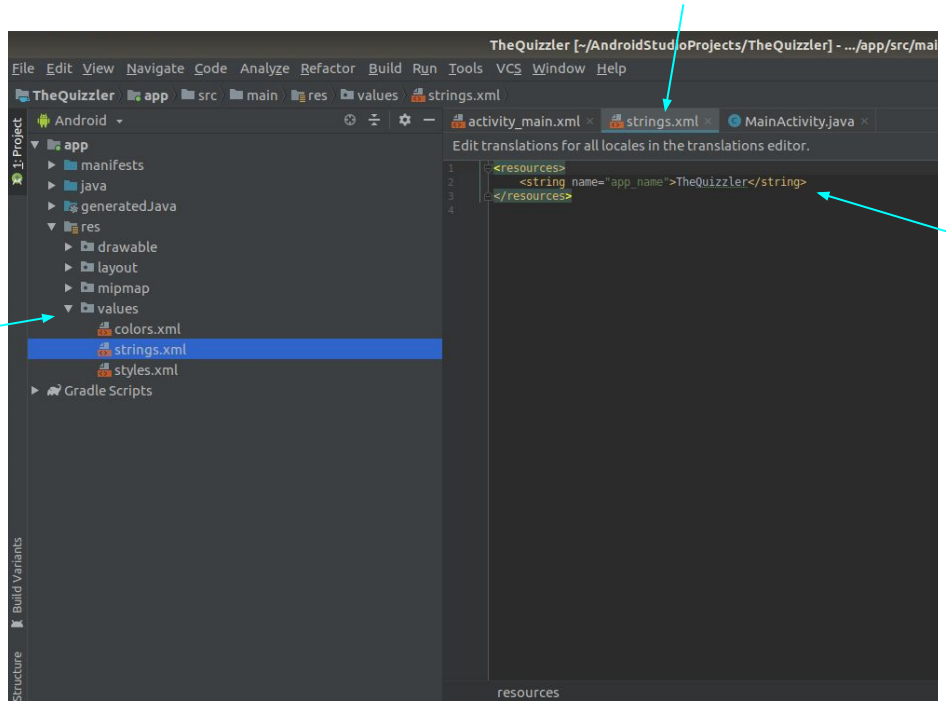
We can use the strings file to hardcode strings and reference them in our activities.

Find the default strings file, *strings.xml* in `res/values/strings.xml`

Maydm

The Default String Resources File

res/values/strings.xml



Do not delete the default string

Maydm

Edit strings.xml

Let's add three string resources to strings.xml. Add these after the "app_name" string resource, inside of the <resources> tag.

```
<resources>
   <string name="app_name">TheQuizzler</string>

   <string name="question_text">Madison is the capital of Wisconsin.</string>
   <string name="true_button">True</string>
   <string name="false_button">False</string>

</resources>
```

Remember that a "tag" refers to an XML element that controls the user interface.

On this slide, <resources> and <string> are both tags or elements, the terms are used interchangeably and mean the same thing.

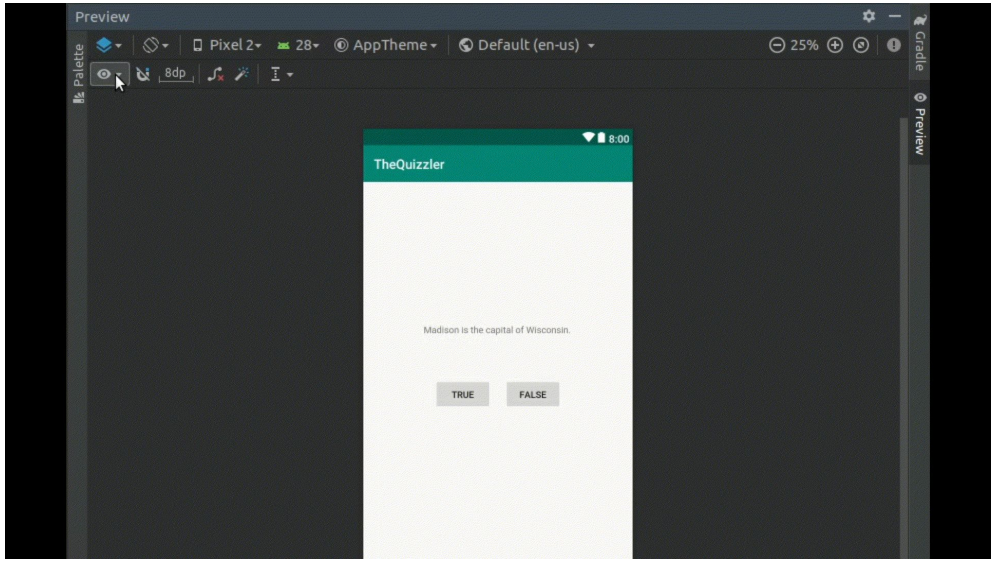
Activity_main.xml Updates

If you look at Activity_main.xml you'll notice that the buttons and text box have updated automatically. We have created reusable elements that could be used anytime we required within the app.

However, you may also notice that the buttons are misaligned now. We can drag and drop the elements (the buttons and text box) in the GUI to where we want them.

Layout Review

We can preview the The Quizzler app to this point in the preview screen by looking at activity_main.xml. Emulate what the app will look like on a phone by selecting the eye icon and then checking “Show Layout Decoration”



Maydm

Any Questions?

This concludes the first steps in creating The Quizzler

- Activities
- Layout
- XML
- Widgets
- ConstraintLayout
- TextView
- Buttons
- Elements
- Constraining Elements
- String Resources
- Layout Preview

Maydm