Developing Android Application Tutorial Cheat sheet

Understanding the View:

TextView, ImageView, Button

Understanding the XML Syntax:

TextView layout can be done at the following website:

<http://labs.udacity.com/android-visualizer/#/android/sandbox>

TextView

Setting Wrap Content

android:layout\_width= “wrap\_content”

android:layout\_height=”wrap\_content”

android:text

android:background=

android:textAppearance="?android:textAppearanceLarge"

Learn Material Design to understand the best practice for font appearance, color, and so on.

The website is given as below:

<https://material.io/guidelines/material-design/introduction.html#introduction-principles>

Join Google Plus for Android Developer to discuss your doubt. It is one of good community where developer will help.

Color code

https://material.io/guidelines/style/color.html#color-color-tool

Documentation

<https://developer.android.com/guide/index.html>

<https://developer.android.com/index.html>

Snackbar and toast

Focus on design of application, which play an important in success of application

If you want more than one views, you have to create view group.

Type of View Group

1. Linear Layout

How it is positioned with respect in either vertical or horizonal.

1. Relative Layout

How it is positioned with respect to their parent or their sibling.

Width and Height

1. Fixed dp values
2. Wrap\_content
3. Match\_parent

How to make equal height or equal width?

There is another parather layout\_weight.

Equal weight means equal width or height depending on the orientation.

Relative Layout

android:layout\_alignParentRight="true"

android:layout\_alignParentTop="true"

android:layout\_alignParentBottom="true"

android:layout\_alignParentLeft="true"

android:layout\_centerVertical=”true”

Margin vs Padding

Vocabulary for Android

<https://developers.google.com/android/for-all/vocab-words/?utm_source=udacity&utm_medium=course&utm_campaign=android_basics>

OEM Driver used for USB connection

<https://developer.android.com/studio/run/oem-usb.html#Drivers>

Windows 10

To install the Android USB driver on Windows 10 for the first time, do the following:

1. Connect your Android device to your computer's USB port.
2. From Windows Explorer, open **Computer Management**.
3. In the **Computer Management** left pane, select **Device Manager**.
4. In the **Device Manager** right pane, locate and expand **Portable Devices**.
5. In **Portable Devices**, right-click the name of the device you connected, and then select **Update Driver Software**.
6. In the **Hardware Update wizard**, select **Browse my computer for driver software** and click **Next**.
7. Click **Browse** and then locate the USB driver folder. For example, the Google USB Driver is located in ***android\_sdk***\extras\google\usb\_driver\.
8. Click **Next** to install the driver

All image should be in drawable folder

Android:scr = “@drawable/androidparty”

Course available in Udacity

[**Android Basics: User Interface (this course)**](https://www.udacity.com/course/android-basics-make-your-first-app--ud834)

* [**Android Basics: User Input**](https://www.udacity.com/course/android-basics-button-clicks--ud836)
* [**Android Basics: Multiple App Screens**](https://www.udacity.com/course/android-basics-multi-screen-apps--ud839)
* [**Android Basics: Networking**](https://www.udacity.com/course/android-basics-networking--ud843)
* [**Android Basics: Data Storage**](https://www.udacity.com/course/android-basics-data-storage--ud845)

## How to Learn More on Your Own

As you continue learning Android, an important skill to have is the ability to learn on your own. You might find resources out there that feel too advanced for where you are on your journey, but we want you to become accustomed to how developers speak and share their ideas. You don’t have to understand every word, but you can skim for important ideas. Or you can google search for terms that you aren’t familiar with.

## Read your first Android blogpost article

Start by reading [**this post on the Android Developers blog**](http://android-developers.blogspot.com/). It's written by Google Design Advocate, Roman Nurik, who was the lead designer on the [**Google I/O 2014 app**](http://android-developers.blogspot.com/2014/08/material-design-in-2014-google-io-app.html). Google I/O is an annual conference that Google holds for developers.

## Follow official Android Development channels on social media

Aside from the blog, you can get the latest news about Android development via:

* [**Android Developers G+ page**](https://plus.google.com/+AndroidDevelopers/posts)
* [**Android Developers Twitter page**](https://twitter.com/AndroidDev)
* [**Google Developers YouTube channel**](https://www.youtube.com/user/GoogleDevelopers)

## Kirill's Favorite Resources

In addition to the official channels for Android development news, there’s a ton of content online, and a vibrant ecosystem of Android developers who are happy to share their knowledge through blog post articles, social media tips, and conference talks.

Here are some of Kirill’s favorite Android resources:

* [**Styling Android blog**](http://blog.stylingandroid.com/): A blog that shows off various technical aspects of building design elements of Android apps.
* [**Chris Banes' blog**](https://chris.banes.me/): A blog that gives you a deeper look into Android support libraries.
* [**Fragmented Podcast**](http://fragmentedpodcast.com/): A weekly podcast filled with Android development discussion.

You can [**find Kirill on G+**](https://plus.google.com/+KirillGrouchnikov), and check out his #pixelpushing series.

Please check out <https://in.udacity.com/courses/all/> for more course

## How to Learn More on Your Own

### Find an interesting Android article

Have you ever had a class where the instructions are to browse the internet for whatever interests you? Well, here it is. Search online to find another Android development article that you're interested in and read it (or it can be a video or podcast). Figure out what the key ideas of the piece are and share those ideas with others. You can start by checking out these sites.

* [**Android Weekly**](http://androidweekly.net/) is a weekly newsletter about the latest news in Android development.
* [**MaterialUp**](https://www.materialup.com/) curates the best of Material design inspiration and tools.

### Build up your network of Android developers on social media

Google Developer Experts are a network of professional developers in industry (external to Google). They are recognized as leaders in their communities - speaking at conferences, publishing tutorials, and mentoring developers. You can look at their profiles [**here**](https://developers.google.com/experts/all/technology/android?utm_source=udacity&utm_medium=course&utm_campaign=android_basics) and follow them on social media.

There are also multiple communities that you can be involved with. These are great places to ask those burning Android questions.

* [**Android Development G+ Community**](https://plus.google.com/communities/105153134372062985968)
* [**Android App Design Community**](https://plus.google.com/communities/117140012142044995433)

### Alice's Favorite Resources

Here are the top three resources that Alice finds most useful:

* Lately she's been using the [**material design**](http://www.google.com/design/spec/material-design/introduction.html?utm_source=udacity&utm_medium=course&utm_campaign=android_basics) spec a lot for user interface development.
* [**Stack Overflow**](http://stackoverflow.com/questions/tagged/android) is definitely used the most for day-to-day development. She usually does a search in Google and click on any Stack Overflow results first.
* Git is a great tool for working on software projects in teams and keeping track of different version of the code. Here's [**one**](http://gitref.org/index.html) reference guide to Git that she likes to use, but there are lots of other tutorials/references out there too.

You can follow [**Alice on G+**](http://google.com/+AliceYangColeman).

### The Final Just Java Code for Lesson Two

Speaking of Git, you can find the final code for [**Just Java**](https://github.com/udacity/Just-Java) on this [**Github**](https://github.com/) web-page. Github is a website that facilitates sharing code online. Code on Github is organized via the [**Git**](https://git-scm.com/) version control system. To learn more about Git and Github, consider taking our [**version control course**](https://www.udacity.com/course/how-to-use-git-and-github--ud775).

Three Step to note before view

1. Select the View

Type of view and their number

1. Position View

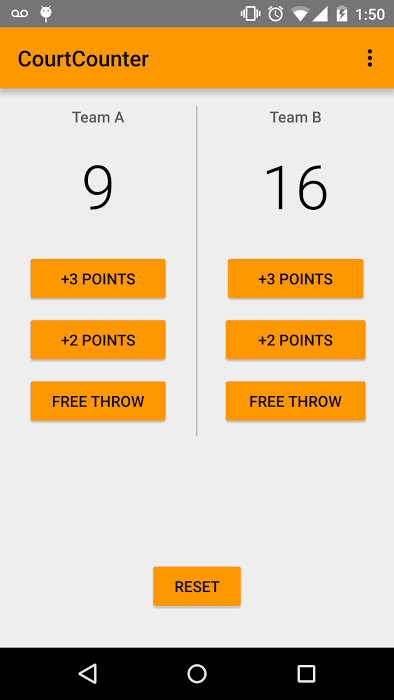
Gridlayout, linearlayout or so on

1. Style View

Spacing, padding, margin, capital letter

## Make your App Pretty

For this quiz you’ll be making your app look like this:

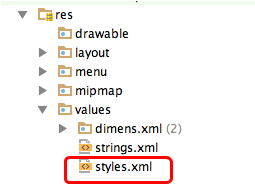


We’ll do this in five steps:

1. Copy over new contents for the styles.xml file.
2. Make a grey line between the two LinearLayouts.
3. Adjust the LinearLayouts sizes.
4. Update the text size, color and font.
5. Add the correct padding and margin.

### Step 1 : Copy Over New styles.xml File

The styles.xml file is a type of resource file that defines the format and look for a layout. You can set things like button colors and the color of the action bar.



The XML code to copy is below.

<resources>

<!-- Base application theme. -->

<style name="AppTheme" parent="Theme.AppCompat.Light">

<!-- Primary theme color of the app (sets background color of app bar) -->

<item name="colorPrimary">#FF9800</item>

<!-- Background color of buttons in the app -->

<item name="colorButtonNormal">#FF9800</item>

</style>

</resources>

colorPrimary will change the color of the Action Bar. colorButtonNormal will change the color of the Buttons.

**Note: colorButtonNormal will only work on phones running API 22 and above. So if you have an older phone, the buttons will not be colored orange.**

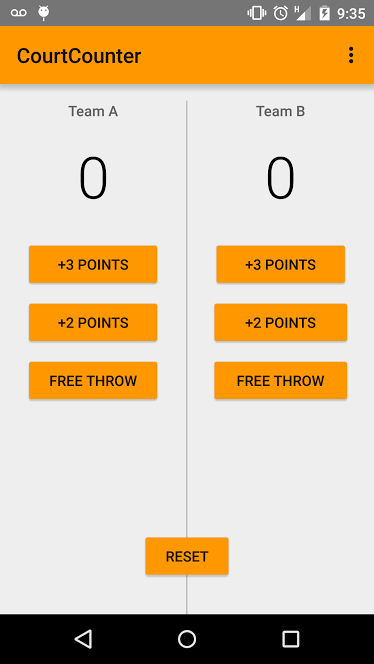
### Step 2 : Make the Grey Line

You can use the View tag to make a view box and you can then color in the box using the background attribute. The color of the view should be @android:color/darker\_gray. By making the box 1dp wide, it will look like a line.

Figuring out where exactly to position the box is up to you.

### Step 3 : Adjust the LinearLayouts

When you first add the view, you might see that the layout looks like this:



You’ll need to adjust the sizes of various views so that the grey line extends only to the end of the buttons (don’t worry about the top of the line, you’ll fix that in step 5). **DO NOT** use fixed widths, instead use match\_parent and wrap\_content only.

### Step 4 : Update the Text Size, Color and Font

Update the the TextViews to the following specifications.

**Team Name TextViews**:

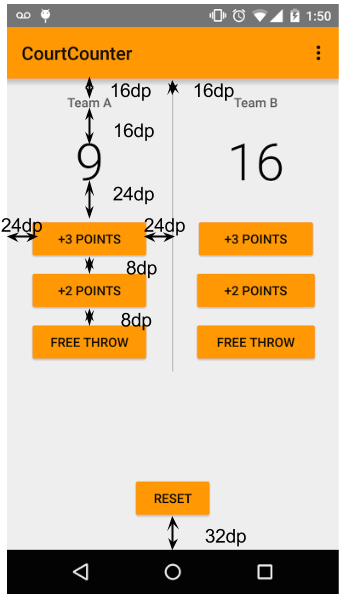
* **Size** 14sp
* **Color** #616161
* **Font** sans-serif-medium

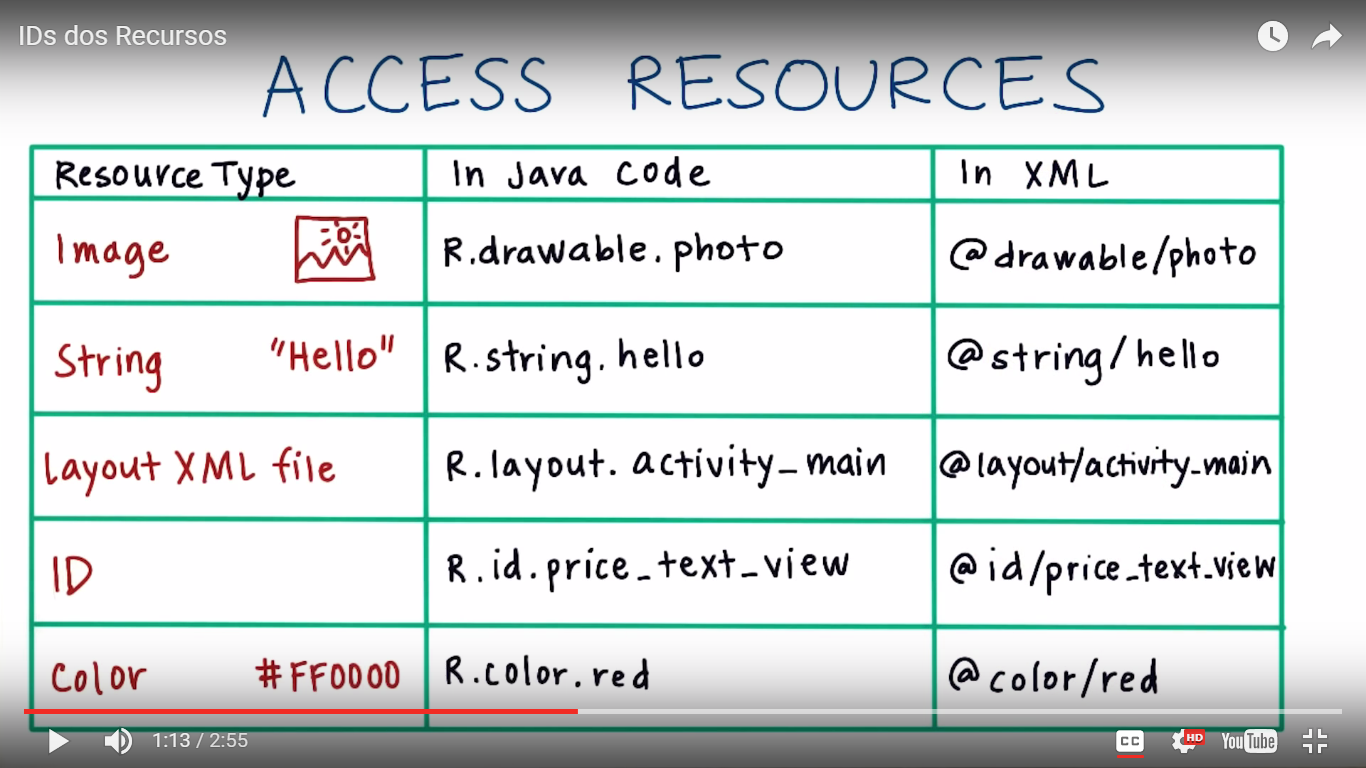
**Score TextViews**

* **Size** 56sp
* **Color** #000000
* **Font** sans-serif-light

### Step 5 : Padding and Margin

Update the layout to have the correct spacing, as shown in the diagram:





To print log message,

Log.v()

Most common use

Intent

Action.View

<https://developer.android.com/reference/android/content/Intent.html>

<https://developer.android.com/guide/components/intents-common.html>

String can be given in string.xml

To fetch value from string.xml, getString(R.string.(string variable))

<https://developer.android.com/guide/topics/resources/string-resource.html>

Watch video

<https://youtu.be/9K-okZ98-CI>