[**Method 1**](https://gist.github.com/udacityandroid/a50468fc7cc66389b884#file-method-1)

|  |  |
| --- | --- |
|  | /\*\* |
|  | \* Get the email account name. |
|  | \* |
|  | \* @return the name of the account. |
|  | \*/ |
|  | private String getAccountName() { |
|  | return "android@gmail.com"; |
|  | return "droid@gmail.com"; |
|  | } |

[**Raw**](https://gist.github.com/udacityandroid/a50468fc7cc66389b884/raw/3ba625abe90d15be14d09683c343e8b0426fe70e/Method%25202)

[**Method 2**](https://gist.github.com/udacityandroid/a50468fc7cc66389b884#file-method-2)

|  |  |
| --- | --- |
|  | /\*\* |
|  | \* Add $4 of tip onto the current bill. |
|  | \* |
|  | \* @return the total price of the bill (including tip). |
|  | \*/ |
|  | private addTip(int bill) { |
|  | return bill + 4; |
|  | } |

[**Raw**](https://gist.github.com/udacityandroid/a50468fc7cc66389b884/raw/3ba625abe90d15be14d09683c343e8b0426fe70e/Method%25203)

[**Method 3**](https://gist.github.com/udacityandroid/a50468fc7cc66389b884#file-method-3)

|  |  |
| --- | --- |
|  | /\*\* |
|  | \* Sets up the app for the current city. |
|  | \*/ |
|  | private nothing setup() { |
|  | cityName = "London"; |
|  | } |

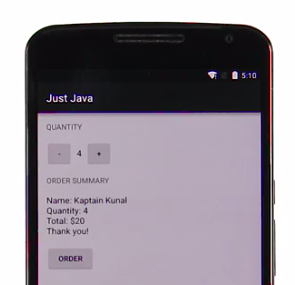
[**Raw**](https://gist.github.com/udacityandroid/a50468fc7cc66389b884/raw/3ba625abe90d15be14d09683c343e8b0426fe70e/Method%25204)

[**Method 4**](https://gist.github.com/udacityandroid/a50468fc7cc66389b884#file-method-4)

|  |  |
| --- | --- |
|  | /\*\* |
|  | \* Get the number of students in a class. |
|  | \* |
|  | \* @return the number of students. |
|  | \*/ |
|  | private int getStudentClassSize() { |
|  | return "20"; |
|  | } |

V10\_**Quiz: The DisplayMessage Method**

The end goal should look something like this:



In the activity\_main.xml layout:

1. Change **PRICE** label to **ORDER SUMMARY**
2. Change the TextView for price value to have an ID of @+id/order\_summary\_text\_view.

In MainActivity.java, in the displayMessage(String message) method:

1. Change variable name from priceTextView to orderSummaryTextView.
2. Change R.id.price\_text\_view to R.id.order\_summary\_text\_view.

In MainActivity.java:

1. Remember to delete the displayPrice() method because we don’t need that anymore.

V14\_**Video: What Are Java Objects?**

Here are some helpful links:

* [**More information about Java classes**](https://docs.oracle.com/javase/tutorial/java/concepts/class.html)
* [**Simplified TextView code**](https://gist.github.com/udacityandroid/47592c621d32450d7dbc)
* [**Simplified ImageView code**](https://gist.github.com/udacityandroid/bd550cc8fd37190d85a6)
* In the simplified TextView code, I mentioned you can read more about Context. Here’s a [**link**](http://stackoverflow.com/questions/3572463/what-is-context-in-android) that explains Context.
* [**Actual TextView code**](https://android.googlesource.com/platform/frameworks/base/+/refs/heads/master/core/java/android/widget/TextView.java?utm_source=udacity&utm_medium=course&utm_campaign=android_basics)
* [**Actual ImageView code**](https://android.googlesource.com/platform/frameworks/base/+/refs/heads/master/core/java/android/widget/ImageView.java?utm_source=udacity&utm_medium=course&utm_campaign=android_basics)

To have a link to the source code (for each Android class) embedded within Android documentation pages, check out[**this Chrome extension**](https://plus.google.com/+AndroidDevelopers/posts/1EyxkFnPjmD).

If you don’t have Chrome, there are other ways to browse the [**Android source tree online**](http://grepcode.com/project/repository.grepcode.com/java/ext/com.google.android/android/).

Here's the TextView [**for example**](http://grepcode.com/file/repository.grepcode.com/java/ext/com.google.android/android/5.1.0_r1/android/widget/TextView.java#TextView).

Or if you’re hardcore, you can [**download it**](https://source.android.com/source/downloading.html?utm_source=udacity&utm_medium=course&utm_campaign=android_basics) onto your computer.

**Video: What Is A Java Class?**

Here are some helpful links:

* [**More information about Java classes**](https://docs.oracle.com/javase/tutorial/java/concepts/class.html)
* [**Simplified TextView code**](https://gist.github.com/udacityandroid/47592c621d32450d7dbc)
* [**Simplified ImageView code**](https://gist.github.com/udacityandroid/bd550cc8fd37190d85a6)
* In the simplified TextView code, I mentioned you can read more about Context. Here’s a [**link**](http://stackoverflow.com/questions/3572463/what-is-context-in-android) that explains Context.
* [**Actual TextView code**](https://android.googlesource.com/platform/frameworks/base/+/refs/heads/master/core/java/android/widget/TextView.java?utm_source=udacity&utm_medium=course&utm_campaign=android_basics)
* [**Actual ImageView code**](https://android.googlesource.com/platform/frameworks/base/+/refs/heads/master/core/java/android/widget/ImageView.java?utm_source=udacity&utm_medium=course&utm_campaign=android_basics)

To have a link to the source code (for each Android class) embedded within Android documentation pages, check out[**this Chrome extension**](https://plus.google.com/+AndroidDevelopers/posts/1EyxkFnPjmD).

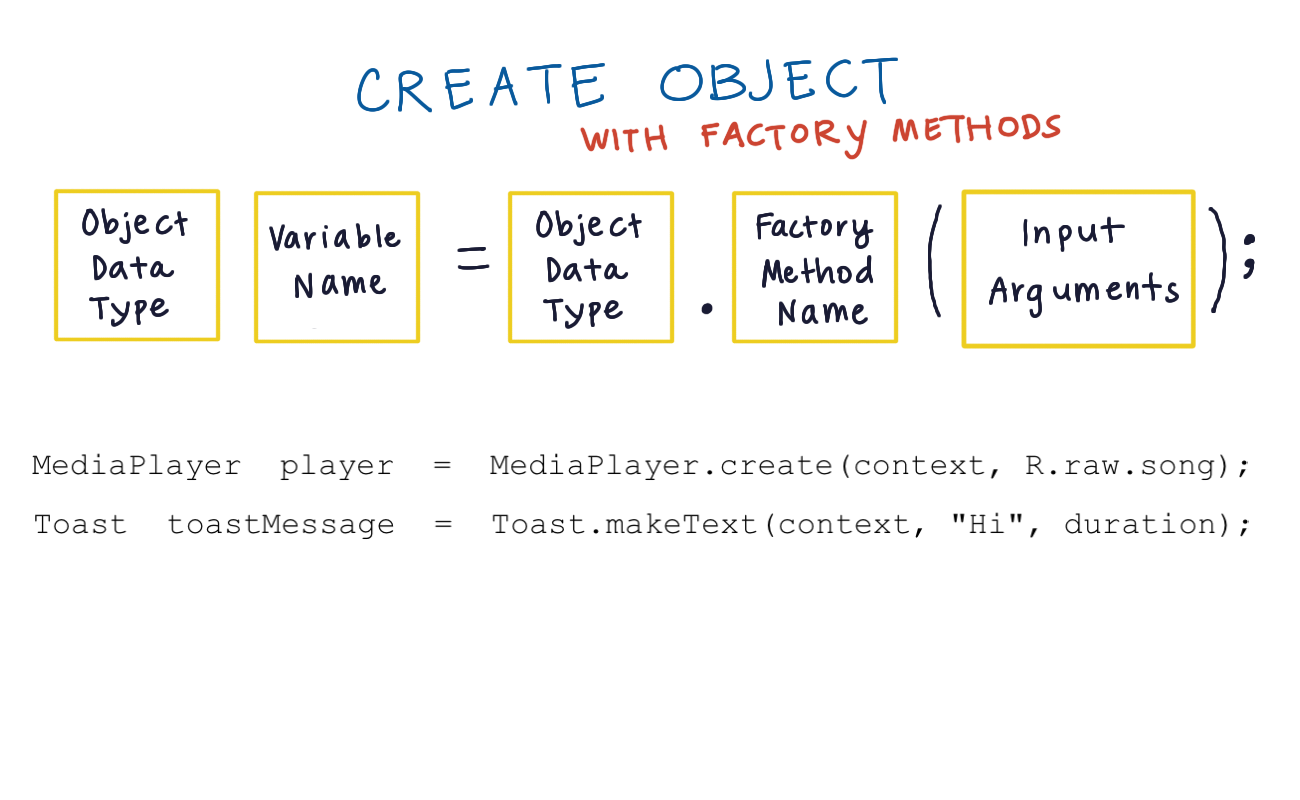
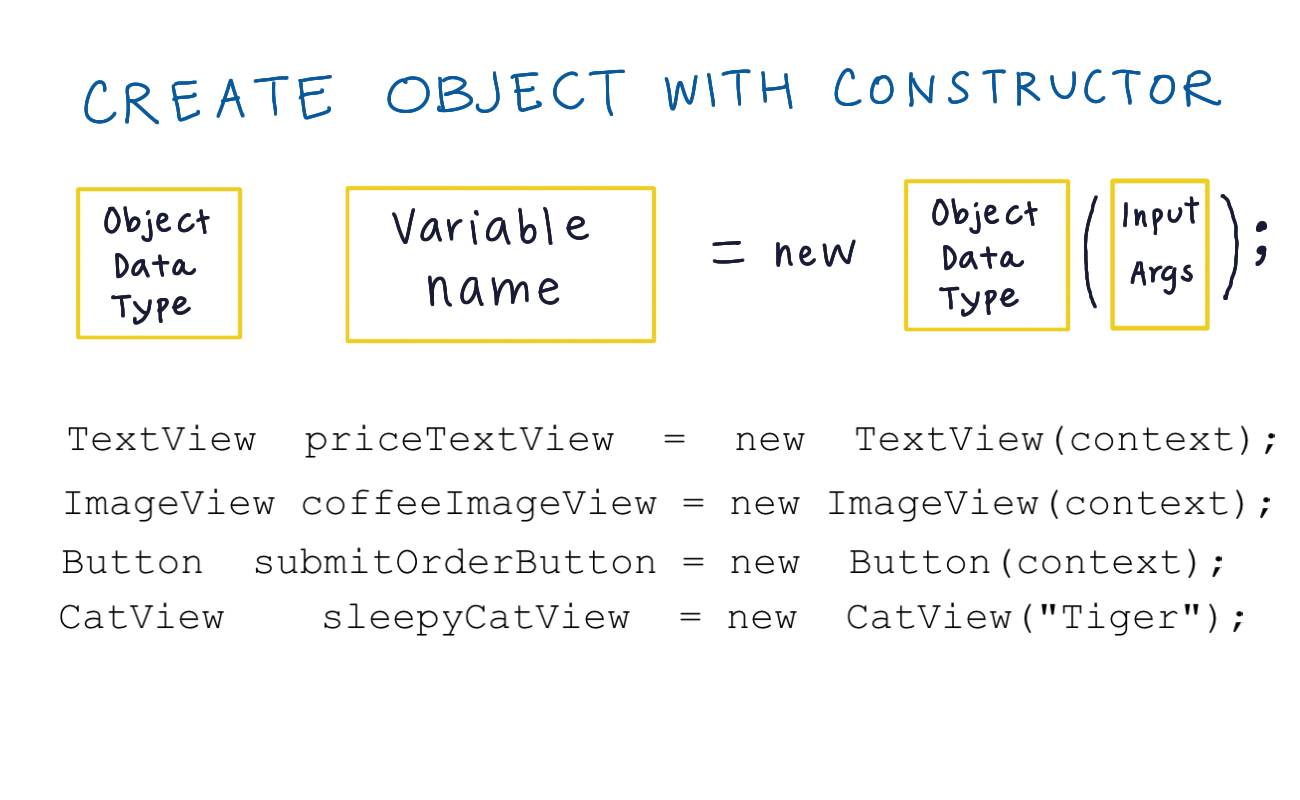
If you don’t have Chrome, there are other ways to browse the [**Android source tree online**](http://grepcode.com/project/repository.grepcode.com/java/ext/com.google.android/android/).

Here's the TextView [**for example**](http://grepcode.com/file/repository.grepcode.com/java/ext/com.google.android/android/5.1.0_r1/android/widget/TextView.java#TextView).

Or if you’re hardcore, you can [**download it**](https://source.android.com/source/downloading.html?utm_source=udacity&utm_medium=course&utm_campaign=android_basics) onto your computer.

16\_**Quiz: Create An Object**

Need a little help? Recall these structural examples:



* [**More information about Java classes/objects**](https://docs.oracle.com/javase/tutorial/java/concepts/class.html)
* [**Simplified TextView code**](https://gist.github.com/udacityandroid/47592c621d32450d7dbc)
* [**Simplified ImageView code**](https://gist.github.com/udacityandroid/bd550cc8fd37190d85a6)
* In the simplified TextView code, I mentioned you can read more about Context. Here’s a [**link to an explanation**](http://stackoverflow.com/questions/3572463/what-is-context-in-android).
* [**Actual TextView code**](https://android.googlesource.com/platform/frameworks/base/+/refs/heads/master/core/java/android/widget/TextView.java?utm_source=udacity&utm_medium=course&utm_campaign=android_basics)
* [**Actual ImageView code**](https://android.googlesource.com/platform/frameworks/base/+/refs/heads/master/core/java/android/widget/ImageView.java?utm_source=udacity&utm_medium=course&utm_campaign=android_basics)
* [**Learn about Android's Toast objects**](http://developer.android.com/guide/topics/ui/notifiers/toasts.html?utm_source=udacity&utm_medium=course&utm_campaign=android_basics)

To have a link to the source code (for each Android class) embedded within Android documentation pages, check out this [**Chrome extension**](https://plus.google.com/+AndroidDevelopers/posts/1EyxkFnPjmD).

If you don’t have Chrome, there are [**other ways**](http://grepcode.com/project/repository.grepcode.com/java/ext/com.google.android/android/) to browser the Android source tree on-line. For example here's a[**TextView.**](http://grepcode.com/file/repository.grepcode.com/java/ext/com.google.android/android/5.1.0_r1/android/widget/TextView.java#TextView)

Or if you’re hardcore, you can [**download it**](https://source.android.com/source/downloading.html?utm_source=udacity&utm_medium=course&utm_campaign=android_basics) onto your computer.

## 22\_Set Data on Views

### A Review Of What We Know

So we’ve now seen a few examples of how we can use the setText method to modify a view. Here’s one example we just looked at:

TextView orderSummaryTextView = (TextView) findViewById(R.id.order\_summary\_text\_view);

orderSummaryTextView.setText(message);

You can break this down into two steps:

**Step 1 : Get the view object using the view ID**

The first line of code is getting the TextView and storing it in a variable named orderSummaryTextView. To actually get the view, we use the findViewById method which is a method in the [**Activity class**](http://developer.android.com/reference/android/app/Activity.html#findViewById(int)). The argument required is the view ID, which we supply by typing R.id.IDOFVIEW. In this case, the ID of the view is order\_summary\_text\_view, as set in the XML. Note that you need to **cast** the object, which is what the (TextView) is doing. It is saying that the value findViewById is specifically a TextView and not just a generic View.

**Step 2 : Call a method ON the view object**

Since we are calling a method **on** an object, we use the dot syntax. orderSummaryTextView.setText(message); is equivalent to saying, take the **orderSummaryTextView** object which has the capability to **setText** (as seen in the[**documentation**](http://developer.android.com/reference/android/widget/TextView.html#setText(java.lang.CharSequence))), and setText to whatever string it is passed (in this case the string variable **message** is passed).

### Your Turn

You’re going to practice these steps—getting the view object, storing it in a variable, and then manipulating that view object.

**Step 1 : Create a New Project**

To start this exercise, create a new project (use the **Empty** Activity template on Android Studio 1.4 and newer). The application name should be **Cookies**.

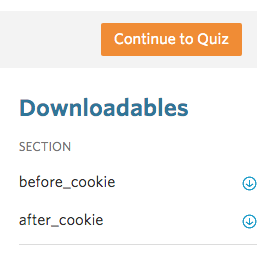
Create this new project the same way you created a project for the **[BirthdayCard](https://www.udacity.com/course/viewer" \l "!/c-ud837/l-4034888704/m-4353418561" \t "_blank)** app, [**Just Java**](https://www.udacity.com/course/viewer#!/c-ud837/l-4038208680/e-4296376368/m-4374678722), and [**Court Counter**](https://www.udacity.com/course/viewer#!/c-ud837/l-4036438656/e-4279218620/m-4319178998).

**Step 2 : Copy over files**

Copy over the Java and XML code into the correct files. The code is [**here**](https://gist.github.com/udacityandroid/8c4604af1d6b6afe12d6).

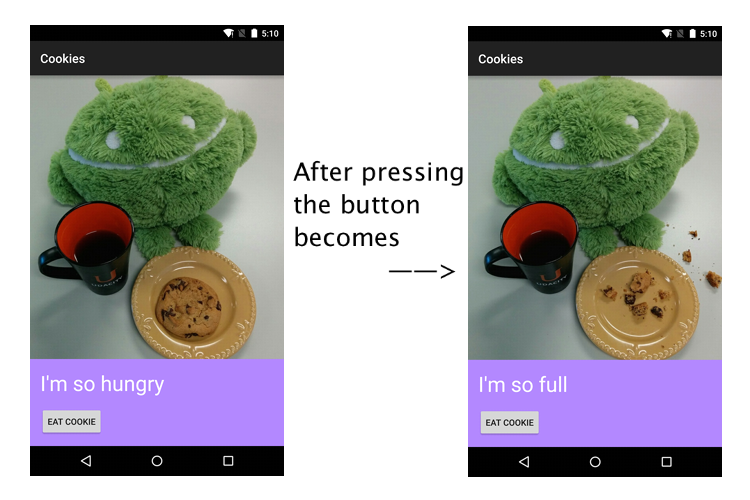
Then you'll want to copy the line below into your app's build.gradle file: compile "com.android.support:appcompat-v7:22.1.0"

Android studio may suggest a newer version of the appcompat library, and you should use that one.

Also place these two image files in the drawable folder, as you did in the **[BirthdayCard app](https://www.udacity.com/course/viewer" \l "!/c-ud837/l-4034888704/e-4134780917/m-4192598920" \t "_blank)**. They can be downloaded from the **Downloadables** section: 

**Step 3 : Hook up the button**

Hook up the “Eat Cookie" button so that, when it's clicked, the image and the text change as seen below.



You’ll need to modify the XML to handle a button being pressed (you’ve done this in Lesson 2 and the practice set). Then you’ll need to use the skills you just learned to manipulate the image and text with the Java code. Good luck!

SKIP TO QUIZ

**Quiz: Set Data On Views**

Have questions? Head to the [**forums**](https://discussions.udacity.com/c/standalone-courses/android-beginners) for discussion with the Udacity Community.

#### Supporting Materials

**[before\_cookie.jpg](https://www.udacity.com/api/nodes/4576240946/supplemental_media/before-cookiejpg/download" \t "_blank)**

**[after\_cookie.jpg](https://www.udacity.com/api/nodes/4576240946/supplemental_media/after-cookiejpg/download" \t "_blank)**

## 23\_Read Data from Views

### Getter and Setter Methods Review

You’ve been using methods such as setText and setImageResource. These are called setter methods because they are meant to modify or manipulate one value of a view (such as the text or image that it stores). Conventionally they start with the word "set".

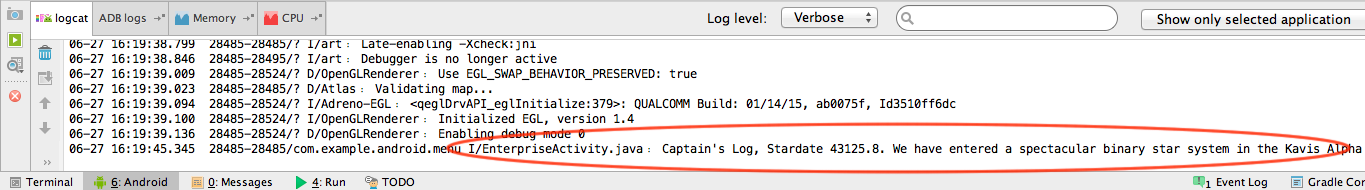
There’s also a category of methods called getter methods, whose sole purpose is to "get" one value of a view, such as getting the current text of a view. Conventionally they start with the word "get". We’ll be using some getter methods in this next exercise.

### Logs

Another skill you will need for this exercise is the ability write to the Android Logs. More information can be found[**here**](http://developer.android.com/tools/debugging/debugging-log.html), but essentially you write a Java statement like this in your code:

Log.i("EnterpriseActivity.java", "Captain's Log, Stardate 43125.8. We have entered a spectacular binary star system in the Kavis Alpha sector on a most critical mission of astrophysical research.");

Then you run the app and look at the Android Log section of your screen. You’ll see some output that looks like this:

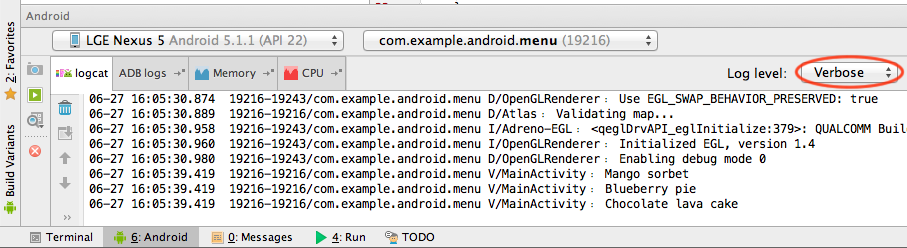


Note the first argument is the **name of the class** that the logging statement comes from. The second is the **text** you want to display.

We’ve used Log.i() here which stands for an "information" level log. You have these other options as well:

* [**e(String, String)**](http://developer.android.com/reference/android/util/Log.html#e(java.lang.String,%20java.lang.String)) (error)
* [**w(String, String)**](http://developer.android.com/reference/android/util/Log.html#w(java.lang.String,%20java.lang.String)) (warning)
* [**i(String, String)**](http://developer.android.com/reference/android/util/Log.html#i(java.lang.String,%20java.lang.String)) (information)
* [**d(String, String)**](http://developer.android.com/reference/android/util/Log.html#d(java.lang.String,%20java.lang.String)) (debug)
* [**v(String, String)**](http://developer.android.com/reference/android/util/Log.html#v(java.lang.String,%20java.lang.String)) (verbose)

They correlate to different log levels, which you can set when you’re running your app here:



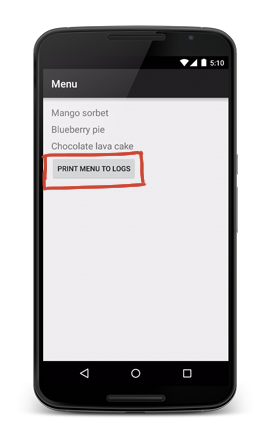
When you set a log level, it will show anything from that log level **and above**, so the verbose log level displays the most information, while the error log level shows only the most critical logs.

### Your Turn

This next quiz also requires creating a new app. Create a new app called **Menu** and copy over this [**code**](https://gist.github.com/udacityandroid/042a390d4414f32a5558).

Note: If using Android Studio 1.4 or newer, be sure to configure your project using the **Empty** Activity template.

It should look like this when you load it up:



When you press the button **Print menu to logs** the XML code has already been set up to trigger a method calledprintToLogs. Add code to this method to read data from each of the menu item TextViews and to print the value of each menu item to the logs.

If you're feeling lost, don't fret—you can look at the quiz image on the next page for an example.

SKIP TO QUIZ

**Quiz: Read Data From Views**

Use the starter code from [**here**](https://gist.github.com/udacityandroid/042a390d4414f32a5558).