



## Calorie Counter - MyFitnessPal

MyFitnessPal, Inc. Health & Fitness

★★★★★ 1,604,137

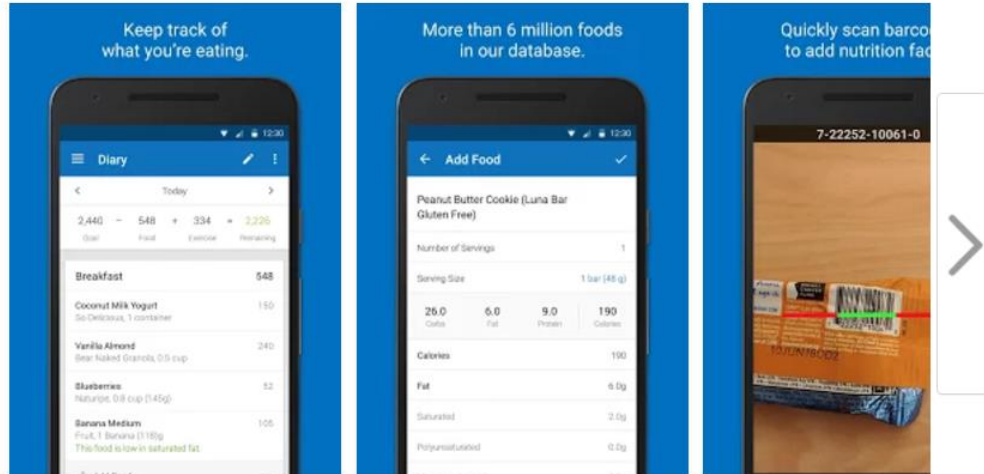
PEGI 3

Contains ads · Offers in-app purchases

This app is compatible with all of your devices.

Add to Wishlist

Install



## Calorie Counter - EasyFit free

Mario Hanna Health & Fitness

★★★★★ 21,758

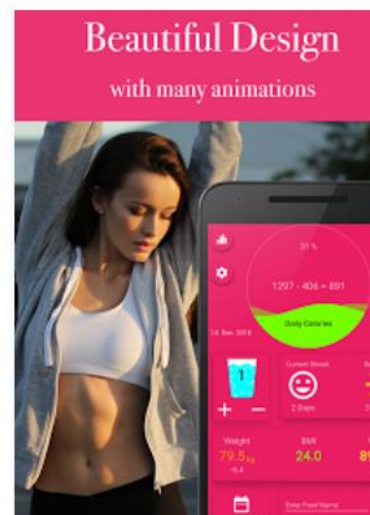
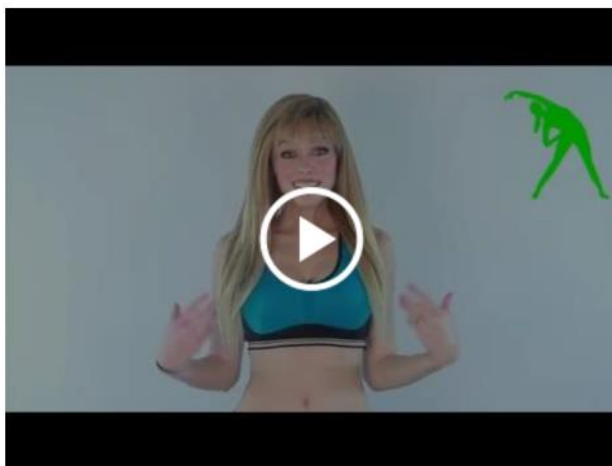
PEGI 3

Contains ads

This app is compatible with all of your devices.

Add to Wishlist

Install





## Calories Burned During Exercise, Activities, Sports and Work

Exercises, sports and activities are listed below, showing calories burned per hour (energy expended) for a 130, 155, 180 or 205 pound person. The amount of calories expended is influenced by many factors, including body weight, intensity of activity, conditioning level and metabolism.

In addition to the List of Calories Burned During Exercise, Sports, Activities below, see the activity specific pages for Cycling, Running, Walking, Swimming, Aerobics, Dancing and a variety of Work, Activities and Hobbies.

<b>Activity, Exercise or Sport (1 hour)</b>	<b>130 lb</b>	<b>155 lb</b>	<b>180 lb</b>	<b>205 lb</b>
Cycling, mountain bike, bmx	502	598	695	791
Cycling, <10 mph, leisure bicycling	236	281	327	372
Cycling, >20 mph, racing	944	1126	1308	1489
Cycling, 10-11.9 mph, light	354	422	490	558
Cycling, 12-13.9 mph, moderate	472	563	654	745
Cycling, 14-15.9 mph, vigorous	590	704	817	931
Cycling, 16-19 mph, very fast, racing	708	844	981	1117
Unicycling	295	352	409	465
Stationary cycling, very light	177	211	245	279
Stationary cycling, light	325	387	449	512
Stationary cycling, moderate	413	493	572	651
Stationary cycling, vigorous	620	739	858	977
Stationary cycling, very vigorous	738	880	1022	1163
Calisthenics, vigorous, pushups, situps...	472	563	654	745
Calisthenics, light	207	246	286	326
Circuit training, minimal rest	472	563	654	745
Weight lifting, body building, vigorous	354	422	490	558



# Create an Android Project

This lesson shows you how to create a new Android project with Android Studio

(<https://developer.android.com/studio/index.html>) and describes some of the files in the project.

You should also read

[Projects Overview](#)

1. In Android Studio, create a new project:

- If you don't have a project opened, in the **Welcome to Android Studio** window, click **Start a new Android Studio project**.
- If you have a project opened, select **File > New Project**.

2. In the **New Project** screen, enter the following values:

- **Application Name:** "My First App"
- **Company Domain:** "example.com"

You might want to change the project location, but leave the other options as they are.

3. Click **Next**.

4. In the **Target Android Devices** screen, keep the default values and click **Next**.

If you're curious about how these SDK versions affect your app, read [Supporting Different Platform Versions](https://developer.android.com/training/basics/supporting-devices/platforms.html) (<https://developer.android.com/training/basics/supporting-devices/platforms.html>).

5. In the **Add an Activity to Mobile** screen, select **Empty Activity** and click **Next**.

6. In the **Customize the Activity** screen, keep the default values and click **Finish**.

After some processing, Android Studio opens the IDE. Now take a moment to review the most important files.

First, be sure the **Project** window is open (select **View > Tool Windows > Project**) and the **Android** view is selected from the drop-down list at the top of that window. You can then see the following files:

**app > java > com.example.myfirstapp > MainActivity.java**

This is the main activity (the entry point for your app). When you build and run the app, the

system launches an instance of this Activity (<https://developer.android.com/reference/android/app/Activity.html>) and loads its layout.

#### **app > res > layout > activity\_main.xml**

This XML file defines the layout for the activity's UI. It contains a `TextView` (<https://developer.android.com/reference/android/widget/TextView.html>) element with the text "Hello world!".

#### **app > manifests > AndroidManifest.xml**

The manifest file (<https://developer.android.com/guide/topics/manifest/manifest-intro.html>) describes the fundamental characteristics of the app and defines each of its components.

#### **Gradle Scripts > build.gradle**

You'll see two files with this name: one for the project and one for the "app" module. Each module has its own `build.gradle` file, but this project currently has just one module. You'll mostly work with the module's `build.gradle` file to configure how the Gradle tools compile and build your app. For more information about this file, see [Configure Your Build](https://developer.android.com/studio/build/index.html) (<https://developer.android.com/studio/build/index.html>).

To run the app, continue to the next lesson (<https://developer.android.com/training/basics/firstapp/running-app.html>).



# Build a Simple User Interface

In this lesson, you'll use the Android Studio Layout Editor to create a layout that includes a text box and a button. In the next lesson, you'll make the app respond to the button tap by sending the content of the text box to another activity.

This lesson teaches you to

- Open the Layout Editor
- Add a text box
- Add a button
- Change the UI strings
- Make the text box size flexible



Figure 1. Screenshot of the final layout

The user interface for an Android app is built using a hierarchy of *layouts* (*ViewGroup* (<https://developer.android.com/reference/android/view/ViewGroup.html>) objects) and *widgets* (*View* (<https://developer.android.com/reference/android/view/View.html>) objects). Layouts are invisible containers that control how its child views are positioned on the screen. Widgets are UI components such as buttons and text boxes.

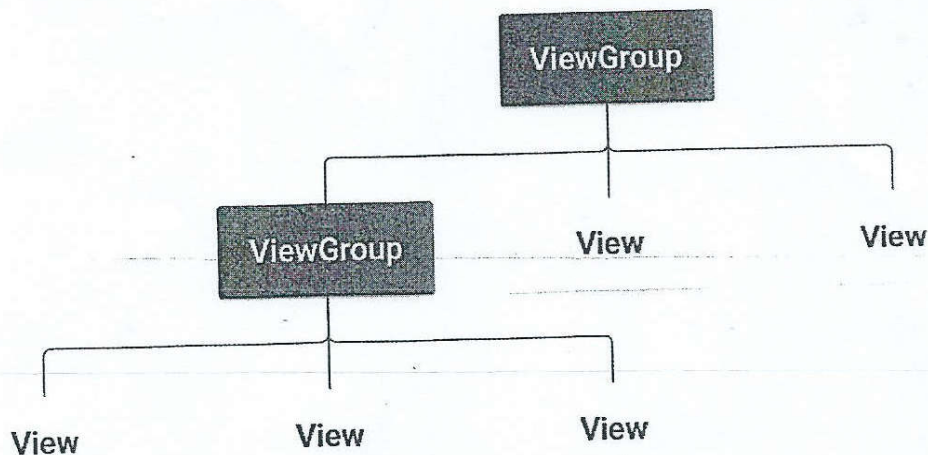


Figure 2. Illustration of how *ViewGroup* objects form branches in the layout and contain *View*

# Run Your App

In the previous lesson (<https://developer.android.com/training/basics/firstapp/creating-project.html>), you created an Android project that displays "Hello World." You can now run the app on a real device or an emulator.

This lesson teaches you to

Run on a real device

Run on an emulator

## Run on a real device

Set up your device as follows:

1. Connect your device to your development machine with a USB cable. If you're developing on Windows, you might need to install the appropriate USB driver for your device. For help installing drivers, see the OEM USB Drivers (<https://developer.android.com/studio/run/oem-usb.html>) document.
2. Enable **USB debugging** on your device by going to **Settings > Developer options**.

**Note:** On Android 4.2 and newer, **Developer options** is hidden by default. To make it available, go to **Settings > About phone** and tap **Build number** seven times. Return to the previous screen to find **Developer options**.

Run the app from Android Studio as follows:

1. In Android Studio, click the **app** module in the **Project** window and then select **Run > Run** (or click **Run ▶** in the toolbar).
2. In the **Select Deployment Target** window, select your device, and click **OK**.

Android Studio installs the app on your connected device and starts it.


That's "hello world" running on your device! To start developing, continue to the next lesson (<https://developer.android.com/training/basics/firstapp/building-ui.html>).

## Run on an emulator



Before you run your app on an emulator, you need to create an Android Virtual Device (<https://developer.android.com/tools/devices/index.html>) (AVD) definition. An AVD definition specifies the characteristics of an Android phone, tablet, Android Wear, or Android TV device that you want to simulate in the Android Emulator.

Create an AVD Definition as follows:

1. Launch the Android Virtual Device Manager by selecting **Tools > Android > AVD Manager**, or by clicking the AVD Manager icon  in the toolbar.
2. In the **Your Virtual Devices** screen, click **Create Virtual Device**.
3. In the **Select Hardware** screen, select a phone device, such as Pixel, and then click **Next**.
4. In the **System Image** screen, click **Download** for one of the recommended system images. Agree to the terms to complete the download.
5. After the download is complete, select the system image from the list and click **Next**.
6. On the next screen, leave all the configuration settings as they are and click **Finish**.
7. Back in the **Your Virtual Devices** screen, select the device you just created and click **Launch this AVD in the emulator ▶**.

While the emulator starts up, close the Android Virtual Device Manager window and return to your project so you can run the app:

1. Once the emulator is booted up, click the **app** module in the **Project** window and then select **Run > Run** (or click **Run ▶** in the toolbar).
2. In the **Select Deployment Target** window, select the emulator and click **OK**.

Android Studio installs the app on the emulator and starts it.

That's "hello world" running on the emulator! To start developing, continue to the next lesson

(<https://developer.android.com/training/basics/firstapp/building-ui.html>).