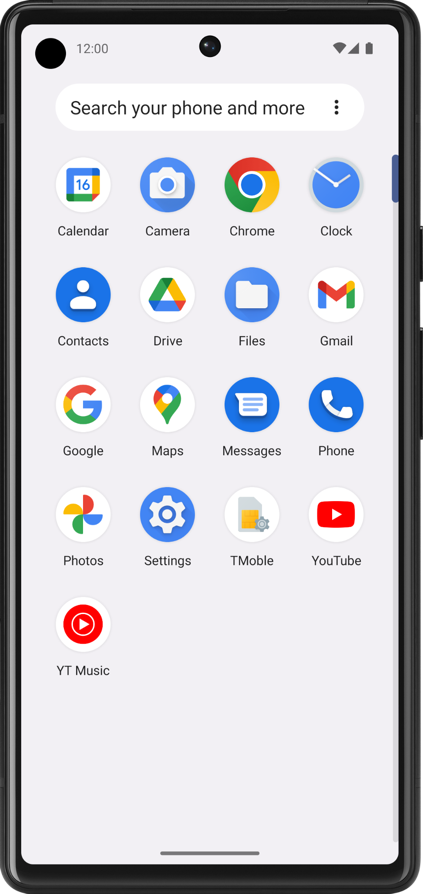
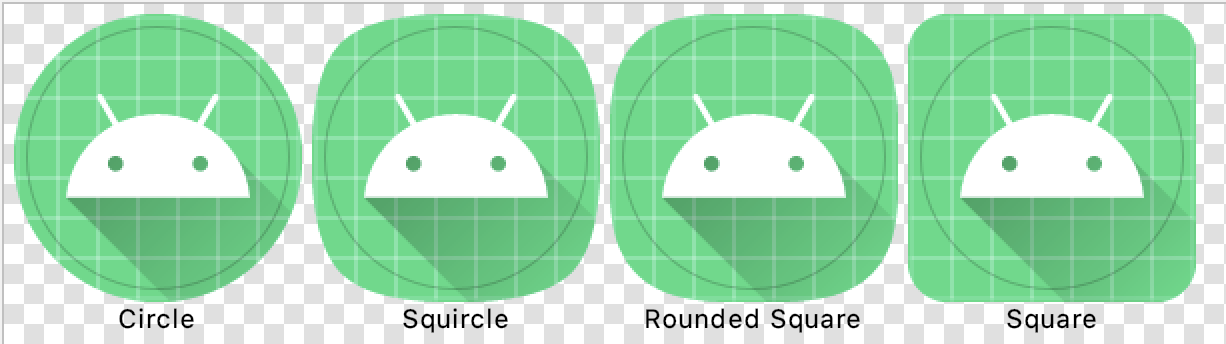
**19.2 Иконка приложения**

Иконка (значок приложения) – это важный способ выделить ваше приложение, добавив ему особый стиль и внешний вид. Значок приложения появляется в нескольких местах, включая главный экран, экран всех приложений и приложение «Настройки».



Если вы использовали разные устройства Android, возможно, вы заметили, что интерфейс запуска может выглядеть по-разному в зависимости от производителя устройства. Иногда производители устройств создают специальные средства запуска, которые являются визитной карточкой их бренда. Например, разные производители могут отображать иконки приложений в форме, отличной от круглой формы значка, показанной выше.

Например, они могут отображать все иконки приложений в форме квадрата, закругленного квадрата или прямоугольника (между квадратом и кругом).



Независимо от формы, которую выбирает производитель устройства, цель состоит в том, чтобы все значки приложений на одном устройстве имели единую форму для единообразного взаимодействия с пользователем.



Именно поэтому на платформе Android появилась поддержка **адаптивных иконок** (начиная с уровня API 26). Внедрив адаптивный значок для вашего приложения, ваше приложение сможет работать на большом количестве устройств, адаптируя значок запуска в зависимости от дисплея устройства.

Эта лаборатория кода предоставляет вам исходные файлы изображений для значка запуска приложения **Affirmations**. Вы будете использовать инструмент в Android Studio под названием **Image Asset Studio** для создания различных версий иконок запуска. После этого вы сможете применить полученные знания к значкам других приложений!



Загрузите стартовый код

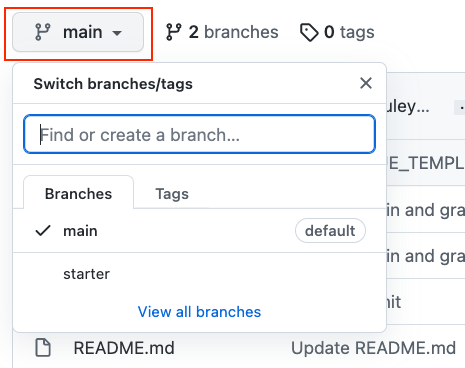
В Android Studio откройте basic-android-kotlin-compose-training-affirmationsпапку.

**URL стартового кода:**

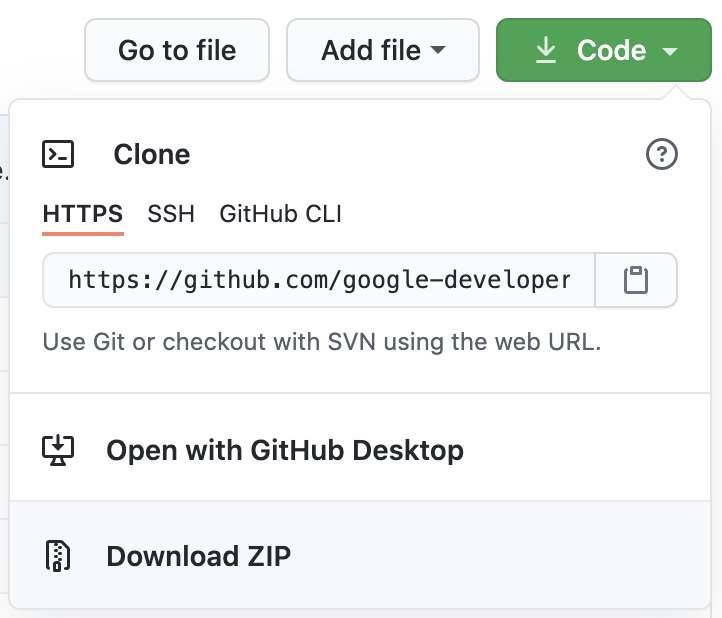
[https://github.com/google-developer-training/basic-android-kotlin-compose-training-affirmations](https://github.com/google-developer-training/basic-android-kotlin-compose-training-affirmations/tree/intermediate)

**Название филиала со стартовым кодом:** intermediate

1. Перейдите на предоставленную страницу репозитория GitHub для проекта.
2. Убедитесь, что имя ветки соответствует имени ветки, указанному в кодовой лаборатории. Например, на следующем снимке экрана имя ветки — **main** .



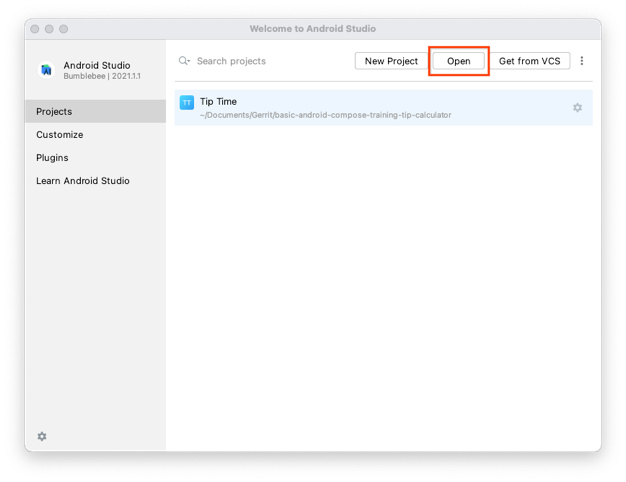
1. На странице проекта GitHub нажмите кнопку **«Код»** , после чего появится всплывающее окно.



1. Во всплывающем окне нажмите кнопку **«Загрузить ZIP»** , чтобы сохранить проект на свой компьютер. Подождите завершения загрузки.
2. Найдите файл на своем компьютере (вероятно, в папке **«Загрузки»** ).
3. Дважды щелкните ZIP-файл, чтобы распаковать его. При этом создается новая папка, содержащая файлы проекта.

Откройте проект в Android Studio.

1. Запустите Android-студию.
2. В окне **«Добро пожаловать в Android Studio»** нажмите « **Открыть»** .



1. В браузере файлов перейдите туда, где находится распакованная папка проекта (вероятно, в папке **«Загрузки»** ).
2. Дважды щелкните папку этого проекта.
3. Подождите, пока Android Studio откроет проект.
4. Нажмите кнопку **«Run»** , чтобы создать и запустить приложение. Убедитесь, что он строится так, как ожидалось.

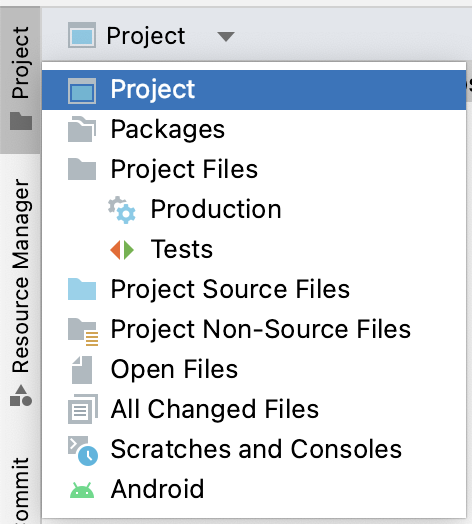
2. Значки запуска

Цель состоит в том, чтобы ваш значок запуска выглядел четким и ясным, независимо от модели устройства или плотности экрана. Плотность экрана показывает, сколько пикселей на дюйм или точек на дюйм (dpi) находится на экране. Для устройства со средней плотностью (mdpi) на экране имеется 160 точек на дюйм, а для устройства с экстра-экстра-экстра-высокой плотностью (xxxhdpi) — 640 точек на дюйм.

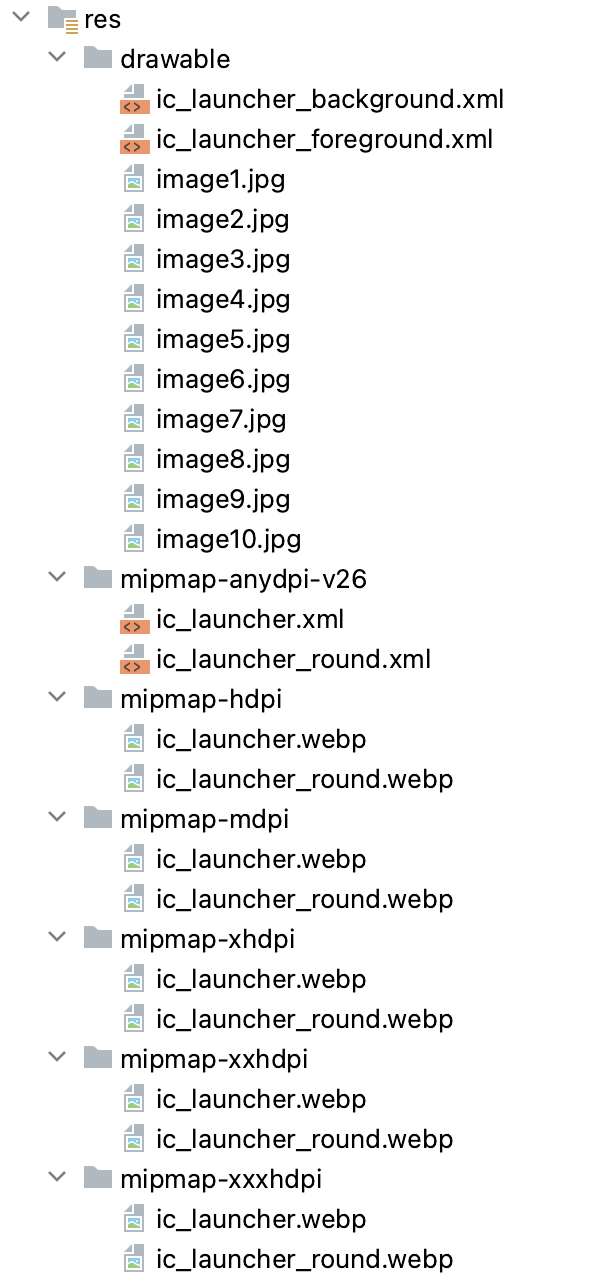
Чтобы учитывать устройства с разной плотностью экрана, вам необходимо предоставить разные версии значка вашего приложения.

Изучите файлы значков запуска

1. Чтобы увидеть, как выглядят значки панели запуска в проекте, откройте проект в Android Studio.
2. В **окне «Project»** переключитесь на представление **«Project»** . Это покажет вам файловую структуру вашего проекта.



1. Перейдите в каталог ресурсов ( **app > src > main > res** ) и разверните некоторые mipmap папки. В этих mipmap папках вы будете хранить ресурсы значков запуска вашего приложения Android.



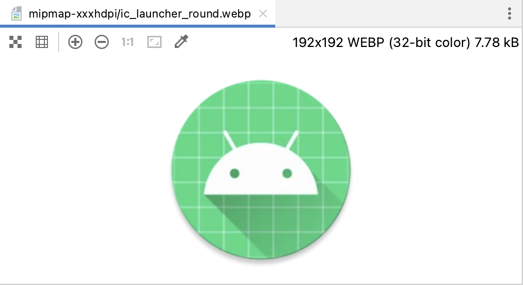
Папки с возможностью рисования содержат векторы для значка запуска в файлах XML. Вектор, в случае рисуемого значка, представляет собой серию инструкций, которые рисуют изображение при его компиляции. mdpi, hdpi, xhdpiи т. д. — это квалификаторы плотности, которые можно добавить к имени каталога ресурсов, например, mipmap,чтобы указать, что они являются ресурсами для устройств с определенной плотностью экрана. Ниже приведен список [квалификаторов плотности](https://developer.android.com/training/multiscreen/screendensities#TaskProvideAltBmp) на Android:

* mdpi- ресурсы для экранов средней плотности (~160 dpi)
* hdpi- ресурсы для экранов высокой плотности (~240 dpi)
* xhdpi- ресурсы для экранов сверхвысокой плотности (~320 точек на дюйм)
* xxhdpi- ресурсы для экранов сверхвысокой плотности (~480 точек на дюйм)
* xxxhdpi- ресурсы для экранов экстра-экстра-высокой плотности (~640 dpi)
* nodpi- ресурсы, не предназначенные для масштабирования, независимо от плотности пикселей экрана
* anydpi- ресурсы, которые масштабируются до любой плотности

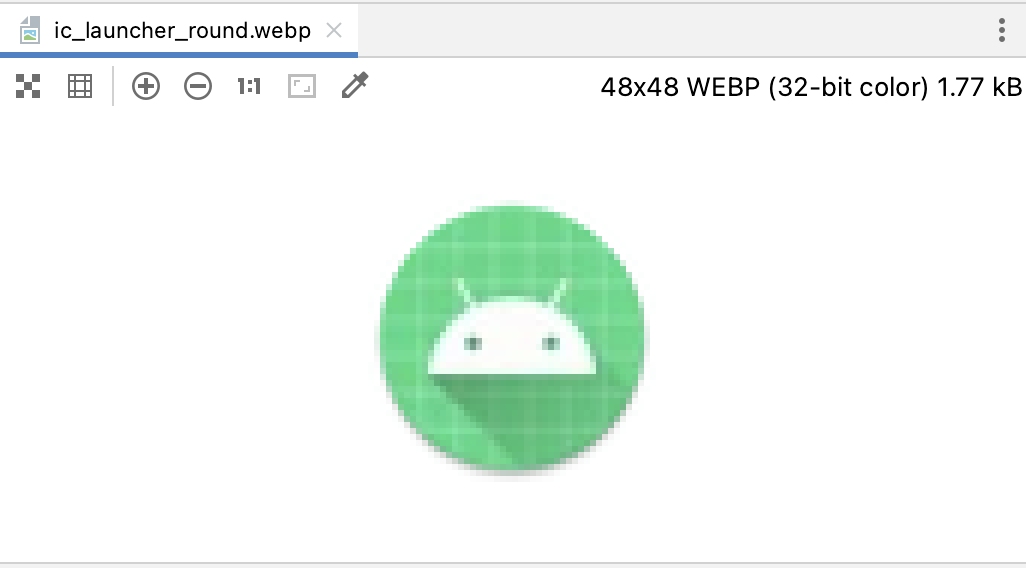
**Примечание.** Вы можете задаться вопросом, почему ресурсы значков запуска расположены в mipmap каталогах отдельно от других ресурсов приложения, расположенных в drawable каталогах. Это связано с тем, что некоторые программы запуска могут отображать значок вашего приложения в большем размере, чем предусмотрено в сегменте плотности устройства по умолчанию. Например, на hdpi устройстве определенная программа запуска устройства может использовать xhdpi вместо этого версию значка приложения. В этих каталогах хранятся значки, соответствующие устройствам, которым требуются значки с плотностью выше или ниже плотности по умолчанию.

1. Если вы нажмете на файлы изображений, вы увидите предварительный просмотр. Файлы ic\_launcher.webp содержат квадратную версию значка, а ic\_launcher\_round.webp файлы — круглую версию значка. Оба доступны в каждом каталоге MIP-карт.

Например, вот так выглядит **res > mipmap-xxxhdpi > ic\_launcher\_round.webp** . Обратите внимание, что размер актива указан в правом верхнем углу. Размер этого изображения составляет 192 x 192 пикселей.



Вот как выглядит **res > mipmap-mdpi > ic\_launcher\_round.webp** . Его размер всего 48х48 пикселей.



Как видите, эти файлы растровых изображений состоят из фиксированной сетки пикселей. Они были созданы для определенного разрешения экрана. Следовательно, качество может ухудшиться при изменении их размера.

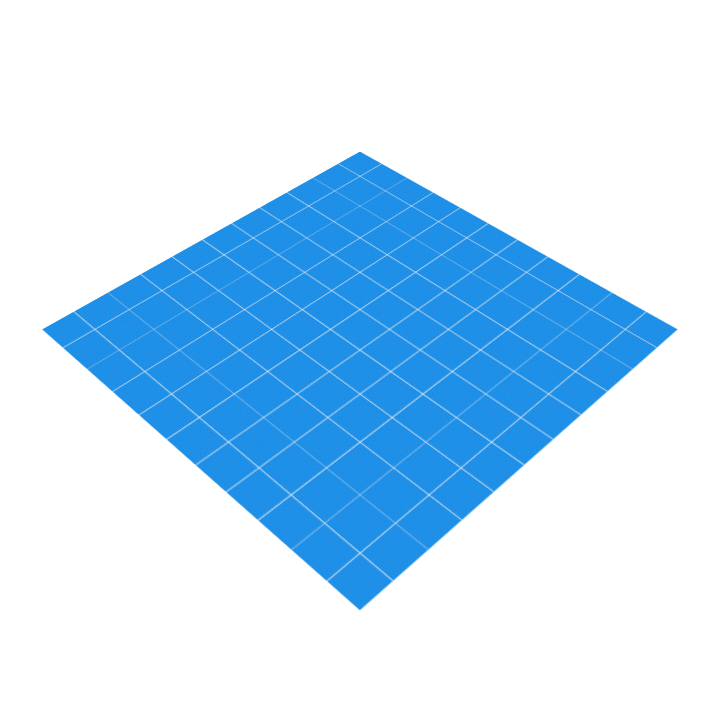
**Примечание** . Чтобы избежать размытия значка приложения, обязательно предоставьте разные растровые изображения значка для каждого сегмента плотности ( mdpi, hdpi, xhdpiи т. д.). Обратите внимание, что плотность экрана устройства не будет точно равна 160, 240, 320 и т. д. На основе плотности экрана устройства Android выбирает ресурс в ближайшем сегменте большей плотности, а затем уменьшает его.

Теперь, когда вы имеете представление о значках запуска, вы узнаете об адаптивных значках.

3. Адаптивные иконки

Слои переднего плана и фона

Начиная с [версии Android 8.0](https://developer.android.com/about/versions/oreo) (уровень API 26) появилась поддержка адаптивных значков, что обеспечивает большую гибкость и интересные визуальные эффекты. Для разработчиков это означает, что значок вашего приложения состоит из двух слоев: слоя переднего плана и фонового слоя.

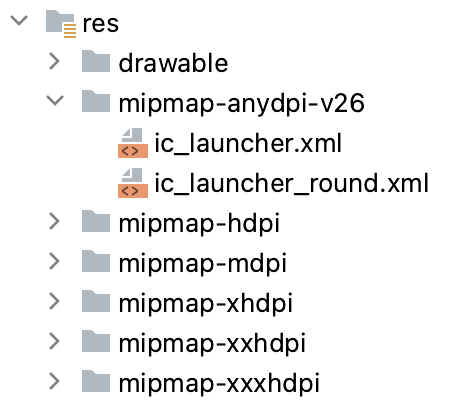


В приведенном выше примере белый значок Android находится на переднем слое, а сине-белая сетка — на фоновом слое. Слой переднего плана накладывается поверх фонового слоя. Затем поверх применяется маска, в данном случае круглая маска, для создания значка приложения круглой формы.

Изучите файлы адаптивных значков

Посмотрите файлы адаптивных значков по умолчанию, которые уже есть в коде вашего приложения **«Аффирмации» .**

1. В **окне проекта** Android Studio найдите и разверните каталог ресурсов **res > mipmap-anydpi-v26** .



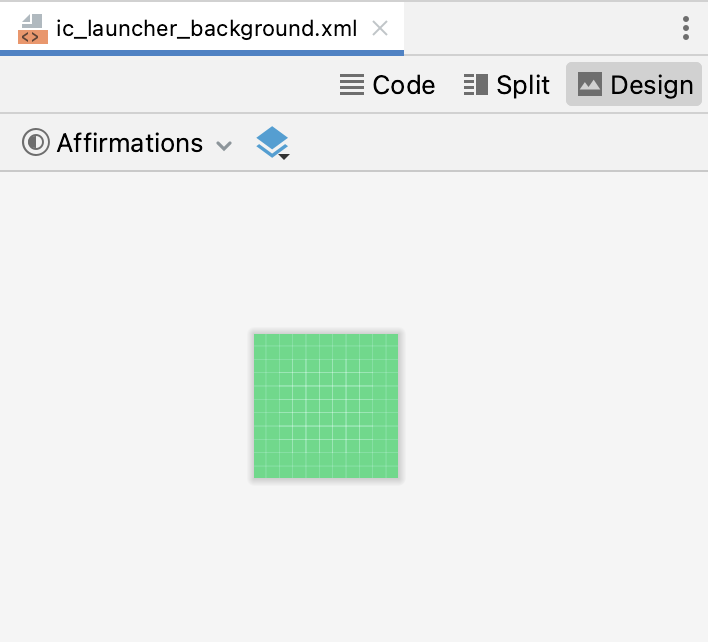
**Примечание** . Адаптивные значки были добавлены на уровне API 26 платформы, поэтому их следует объявлять в mipmapкаталоге ресурсов, в котором есть -v26квалификатор ресурса. Это означает, что ресурсы в этом каталоге будут применяться только на устройствах под управлением API 26 (Android 8.0) или выше. Файлы ресурсов в этом каталоге игнорируются на устройствах под управлением версии 25 или старше в пользу каталогов MIP-карт с сегментированием по плотности.

1. Откройте ic\_launcher.xmlфайл. Вы увидите это:

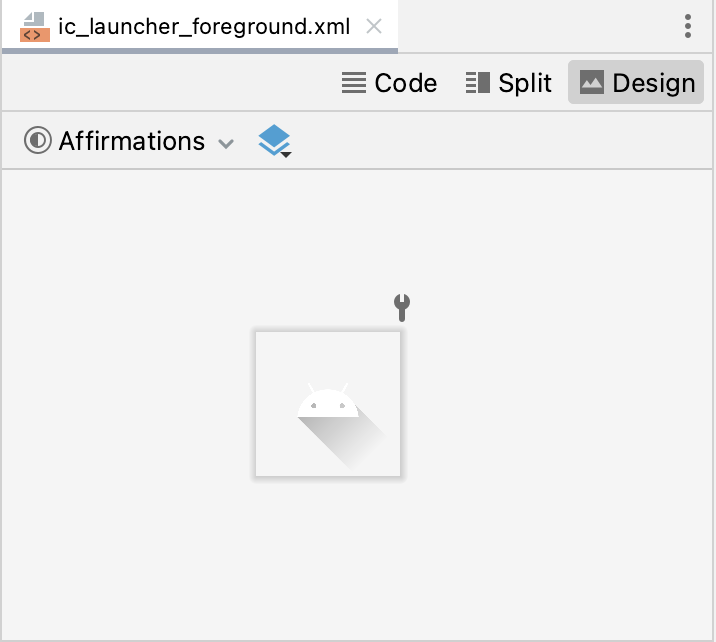
<?xml version="1.0" encoding="utf-8"?>  
<adaptive-icon xmlns:android="http://schemas.android.com/apk/res/android">  
    <background android:drawable="@drawable/ic\_launcher\_background"/>  
    <foreground android:drawable="@drawable/ic\_launcher\_foreground"/>  
    <monochrome android:drawable="@drawable/ic\_launcher\_foreground" />  
</adaptive-icon>

1. Обратите внимание, как <adaptive-icon>элемент используется для объявления слоев <background>и <foreground>значка приложения, предоставляя для каждого из них возможности рисования ресурсов.
2. Вернитесь в представление **«Project»** и найдите рисуемые объекты фона и переднего плана: **res > drawable > ic\_launcher\_background.xml** и **res > drawable > ic\_launcher\_foreground.xml** .
3. Переключитесь в режим **«Design»** , чтобы просмотреть каждый из них:

Фон:



Передний план:



1. Оба файла являются векторными. У них нет фиксированного размера в пикселях. Если вы переключитесь в представление **«Code»** , вы увидите XML-объявление вектора, который можно нарисовать с помощью <vector>элемента.

ic\_launcher\_foreground.xml

<!--  
    Copyright (C) 2023 The Android Open Source Project  
  
    Licensed under the Apache License, Version 2.0 (the "License");  
    you may not use this file except in compliance with the License.  
    You may obtain a copy of the License at  
  
         https://www.apache.org/licenses/LICENSE-2.0  
  
    Unless required by applicable law or agreed to in writing, software  
    distributed under the License is distributed on an "AS IS" BASIS,  
    WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
    See the License for the specific language governing permissions and  
    limitations under the License.  
-->  
  
<vector xmlns:android="http://schemas.android.com/apk/res/android"  
    xmlns:aapt="http://schemas.android.com/aapt"  
    android:width="108dp"  
    android:height="108dp"  
    android:viewportWidth="108"  
    android:viewportHeight="108">  
  <path android:pathData="M31,63.928c0,0 6.4,-11 12.1,-13.1c7.2,-2.6 26,-1.4 26,-1.4l38.1,38.1L107,108.928l-32,-1L31,63.928z">  
    <aapt:attr name="android:fillColor">  
      <gradient  
          android:endX="85.84757"  
          android:endY="92.4963"  
          android:startX="42.9492"  
          android:startY="49.59793"  
          android:type="linear">  
        <item  
            android:color="#44000000"  
            android:offset="0.0" />  
        <item  
            android:color="#00000000"  
            android:offset="1.0" />  
      </gradient>  
    </aapt:attr>  
  </path>  
  <path  
      android:fillColor="#FFFFFF"  
      android:fillType="nonZero"  
      android:pathData="M65.3,45.828l3.8,-6.6c0.2,-0.4 0.1,-0.9 -0.3,-1.1c-0.4,-0.2 -0.9,-0.1 -1.1,0.3l-3.9,6.7c-6.3,-2.8 -13.4,-2.8 -19.7,0l-3.9,-6.7c-0.2,-0.4 -0.7,-0.5 -1.1,-0.3C38.8,38.328 38.7,38.828 38.9,39.228l3.8,6.6C36.2,49.428 31.7,56.028 31,63.928h46C76.3,56.028 71.8,49.428 65.3,45.828zM43.4,57.328c-0.8,0 -1.5,-0.5 -1.8,-1.2c-0.3,-0.7 -0.1,-1.5 0.4,-2.1c0.5,-0.5 1.4,-0.7 2.1,-0.4c0.7,0.3 1.2,1 1.2,1.8C45.3,56.528 44.5,57.328 43.4,57.328L43.4,57.328zM64.6,57.328c-0.8,0 -1.5,-0.5 -1.8,-1.2s-0.1,-1.5 0.4,-2.1c0.5,-0.5 1.4,-0.7 2.1,-0.4c0.7,0.3 1.2,1 1.2,1.8C66.5,56.528 65.6,57.328 64.6,57.328L64.6,57.328z"  
      android:strokeWidth="1"  
      android:strokeColor="#00000000" />  
</vector>

Хотя векторное изображение и растровое изображение описывают графику, между ними есть важные различия.

Растровое изображение мало что понимает в изображении, которое оно содержит, за исключением информации о цвете в каждом пикселе. С другой стороны, векторная графика умеет рисовать формы, определяющие изображение. Эти инструкции состоят из набора точек, линий и кривых, а также информации о цвете. Преимущество в том, что векторную графику можно масштабировать под любой размер холста и любую плотность экрана без потери качества.

Vector [drawable](https://developer.android.com/develop/ui/views/graphics/vector-drawable-resources" \t "_blank) — это реализация векторной графики в Android, предназначенная для обеспечения гибкости на мобильных устройствах. Вы можете определить их в XML с помощью этих [возможных элементов](https://developer.android.com/reference/kotlin/android/graphics/drawable/VectorDrawable) . Вместо предоставления версий растрового ресурса для всех сегментов плотности вам нужно определить изображение только один раз. Таким образом, вы уменьшите размер вашего приложения и упростите его обслуживание.

**Примечание.** Существуют [компромиссы](https://medium.com/androiddevelopers/understanding-androids-vector-image-format-vectordrawable-ab09e41d5c68) между использованием векторного изображения и растрового изображения. Например, значки могут быть идеальными в качестве векторных изображений, поскольку они состоят из простых фигур, тогда как фотографию сложнее описать как серию фигур. В этом случае было бы более эффективно использовать растровый ресурс.

Теперь пришло время перейти к изменению значка приложения!

4. Загрузите новые ресурсы

Загрузите следующие два новых ресурса, которые позволят вам создать адаптивный значок для приложения **«Аффирмации»** . Вам не нужно беспокоиться о понимании каждой детали векторных файлов. Их содержимое автоматически генерируется для вас с помощью инструментов дизайна.

1. Download [ic\_launcher\_background.xml](https://raw.githubusercontent.com/google-developer-training/basic-android-kotlin-compose-training-affirmations/main/app/src/main/res/drawable/ic_launcher_background.xml)— вектор, который можно нарисовать для фонового слоя. Если ваш браузер отображает файл вместо того, чтобы загрузить его, выберите **File > Save Page As...,** чтобы сохранить его на свой компьютер.
2. Download [ic\_launcher\_foreground.xml](https://raw.githubusercontent.com/google-developer-training/basic-android-kotlin-compose-training-affirmations/main/app/src/main/res/drawable/ic_launcher_foreground.xml)— вектор, который можно нарисовать для слоя переднего плана.

Обратите внимание, что существуют определенные требования к этим ресурсам переднего и фонового слоев, например, оба должны иметь размер 108 x 108 точек на дюйм. Более подробную информацию можно просмотреть в [документации AdaptiveIconDrawable](https://developer.android.com/reference/kotlin/android/graphics/drawable/AdaptiveIconDrawable) , а также просмотреть руководство по дизайну [Android](https://m3.material.io/styles/icons/overview" \t "_blank) icons на сайте Material Design.

Поскольку края значка могут быть обрезаны (в зависимости от формы маски, указанной производителем устройства), важно поместить ключевую информацию о значке в « [safe](https://medium.com/google-design/designing-adaptive-icons-515af294c783) zone ». Безопасная зона — круг диаметром 66 dpi в центре переднего слоя. Содержимое за пределами безопасной зоны не должно быть существенным, например цвет фона, и ничего страшного, если оно будет обрезано.

[5. Измените значок приложения.](https://developer.android.com/codelabs/basic-android-kotlin-compose-training-change-app-icon?continue=https%3A%2F%2Fdeveloper.android.com%2Fcourses%2Fpathways%2Fandroid-basics-compose-unit-3-pathway-2%23codelab-https%3A%2F%2Fdeveloper.android.com%2Fcodelabs%2Fbasic-android-kotlin-compose-training-change-app-icon#4)

Вернитесь в Android Studio, чтобы использовать только что загруженные новые ресурсы.

1. Сначала удалите старые доступные для рисования ресурсы, содержащие значок Android и зеленый фон сетки. В  **представлении «Project »** щелкните файл правой кнопкой мыши и выберите **«Delete»** .

Удалить:

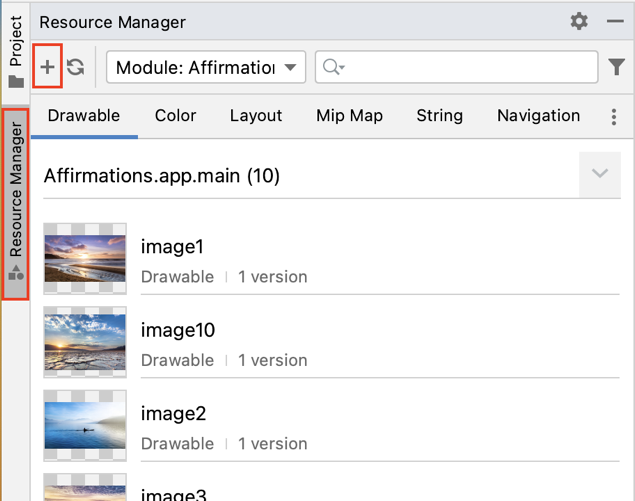
drawable/ic\_launcher\_background.xml  
drawable/ic\_launcher\_foreground.xml

Удалить:

mipmap-anydpi-v26/  
mipmap-hdpi/  
mipmap-mdpi/  
mipmap-xhdpi/  
mipmap-xxhdpi/  
mipmap-xxxhdpi/

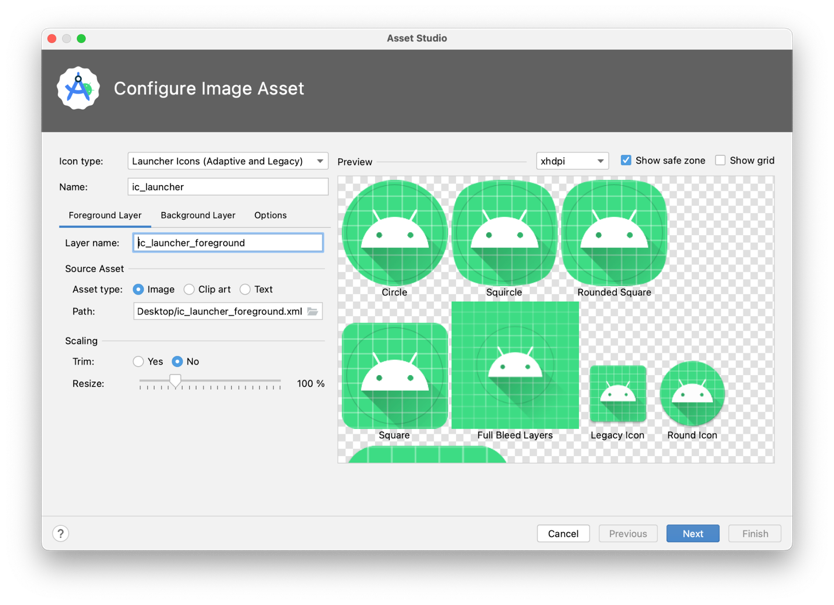
Вы можете снять флажок « **Safe delete (with usage search)** **»** и нажать **«ОК»** . **Safe delete (with usage search)** ищет в коде использование ресурса, который вы собираетесь удалить. В этом случае вы замените эти папки новыми с тем же именем, поэтому вам не нужно беспокоиться о **Safe delete**.

1. Создайте новый **Image Asset**. Вы можете либо щелкнуть правой кнопкой мыши каталог **res** и выбрать  **New > Image Asset**, либо щелкнуть вкладку **Resource Manager**, щелкнуть значок **«+»** , а затем выбрать **Image Asset** в раскрывающемся списке.

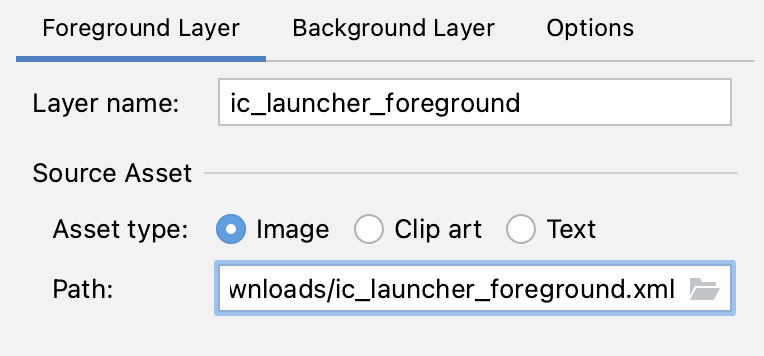


1. **Откроется инструмент Image Asset Studio** в Android Studio .
2. Оставьте настройки по умолчанию:

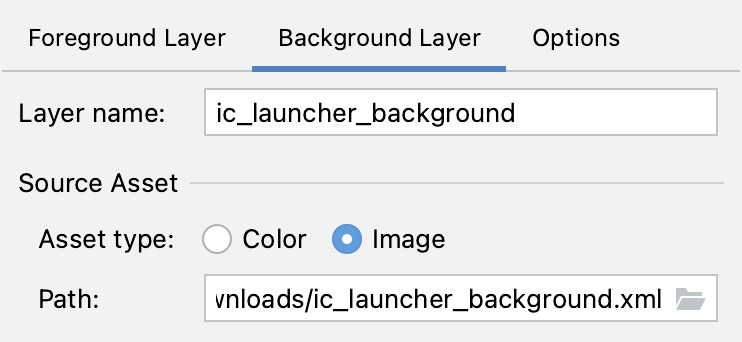
* **Icon Type:** значки запуска (адаптивные и устаревшие)
* **Name:** ic\_launcher



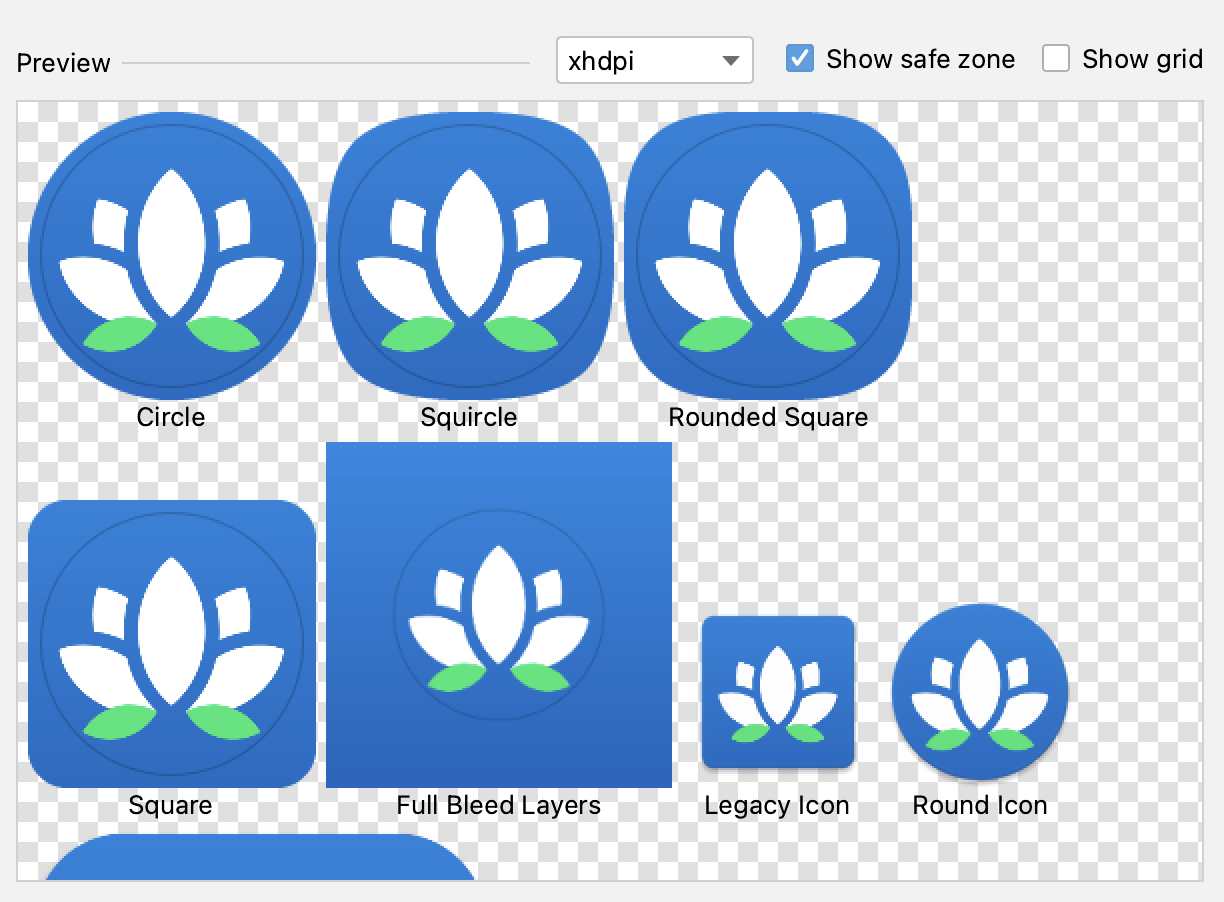
1. Выбрав вкладку **«Foreground Layer** **»** , перейдите в подраздел **«Source Asset»** . В поле **«Path»** щелкните значок папки.
2. Появится приглашение просмотреть ваш компьютер и выбрать файл. Найдите местоположение нового ic\_launcher\_foreground.xml файла, который вы только что скачали. Он может находиться в папке **«Загрузки»** на вашем компьютере. Найдя его, нажмите **«Open»** .
3. Path теперь обновляется с указанием местоположения нового рисуемого вектора переднего плана. Оставьте **Layer Name** как **ic\_launcher\_foreground** и **Asset Type** как **Image**.



1. Затем переключитесь на вкладку **«Background Layer»** интерфейса. Оставьте значения по умолчанию.
2. Щелкните значок папки в поле **«Path»** .
3. Найдите местоположение файла ic\_launcher\_background.xml,  который вы только что скачали. Нажмите **Open**.



1. Предварительный просмотр должен обновляться по мере выбора новых файлов ресурсов. Вот как это должно выглядеть с новыми слоями переднего плана и фона.

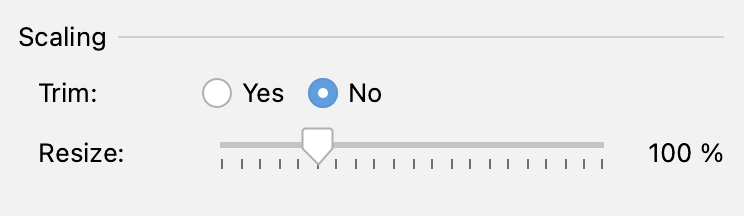


Представляя значок вашего приложения в двух слоях, производители устройств (сокращенно называемые производителями оригинального оборудования или OEM-производителями) могут создавать разные формы в зависимости от устройства Android, как показано в предварительном просмотре выше. OEM-производитель предоставляет маску, которая применяется ко всем значкам приложений на устройстве.

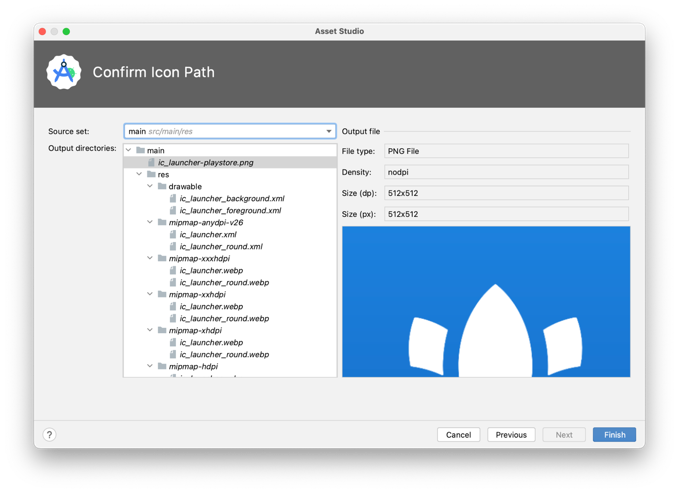
Когда круглая маска применяется к обоим слоям значка вашего приложения, в результате получается круглый значок с изображением Android и синим сеточным фоном (левое изображение выше). В качестве альтернативы можно применить закругленную квадратную маску для создания значка приложения справа вверху.

Наличие как переднего, так и фонового слоя позволяет создавать интересные визуальные эффекты, поскольку эти два слоя могут перемещаться независимо друг от друга и масштабироваться. Несколько забавных примеров того, как могут выглядеть визуальные эффекты, можно найти [в блоге «Проектирование адаптивных иконок»](https://medium.com/google-design/designing-adaptive-icons-515af294c783) в разделе «Соображения дизайна». Поскольку вы не знаете, какое устройство будет у вашего пользователя или какую маску OEM-производитель применит к вашему значку, вам необходимо настроить адаптивный значок, чтобы важная информация не обрезалась.

1. Если важный контент обрезан или кажется слишком маленьким, вы можете использовать ползунок **«Resize» в разделе «Scaling»** каждого слоя, чтобы убедиться, что все отображается в безопасной зоне. Чтобы ничего не обрезалось, измените размер изображений переднего плана и фона до 99 %, перетащив ползунок **«Resize»** на вкладках **«Foreground Layer»** и **«Background Layer»** .



1. Нажмите **Next**.
2. Этот шаг заключается в **Confirm Icon Path**. Вы можете щелкнуть отдельные файлы, чтобы просмотреть их.



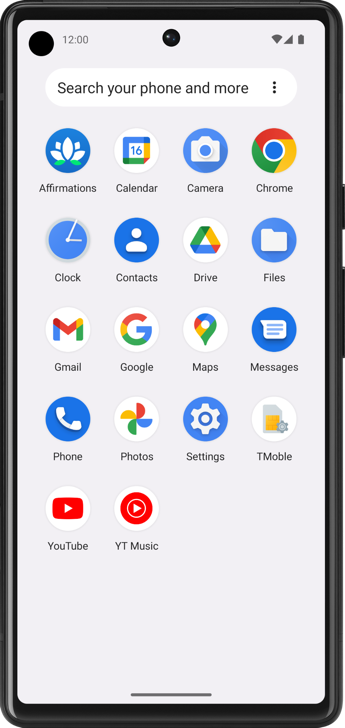
1. Нажмите **«Finish»** .
2. Убедитесь, что все сгенерированные ресурсы правильно выглядят в mipmap папках. Примеры:

Отличная работа! Теперь вы сделаете еще одно изменение.

Проверьте свое приложение

1. Проверьте, появился ли значок вашего нового приложения. Запустите приложение на своем устройстве (эмуляторе или физическом устройстве).
2. Нажмите кнопку **«Home»** на вашем устройстве.
3. Проведите пальцем вверх, чтобы отобразить список **всех приложений** .
4. Найдите приложение, которое вы только что обновили. Вы должны увидеть значок нового приложения.



**Примечание.** В зависимости от модели вашего устройства вы можете увидеть значок запуска другой формы. Тем не менее, слой переднего плана должен отображаться поверх фонового слоя с примененной к нему маской какого-либо типа.

Хорошая работа! Новый значок приложения выглядит великолепно.

Адаптивные и устаревшие значки запуска

Теперь, когда ваш адаптивный значок работает хорошо, вы можете задаться вопросом, почему вы не можете избавиться от всех растровых изображений значков приложений. Эти файлы вам по-прежнему нужны, чтобы значок вашего приложения отображался в более старых версиях Android в высоком качестве, что называется обратной совместимостью.

Для устройств под управлением **Android 8.0 или более поздней версии** (версия API 26 и выше) можно использовать **адаптивные значки** (комбинация рисуемого вектора переднего плана, рисуемого вектора фона с нанесенной поверх него OEM-маской). Это соответствующие файлы в вашем проекте:

res/drawable/ic\_launcher\_background.xml  
res/drawable/ic\_launcher\_foreground.xml  
res/mipmap-anydpi-v26/ic\_launcher.xml  
res/mipmap-anydpi-v26/ic\_launcher\_round.xml

На устройствах под управлением **Android 8.0** (но выше минимально необходимого уровня API вашего приложения) используются **устаревшие значки запуска**mipmap (растровые изображения в папках с разной плотностью). Это соответствующие файлы в вашем проекте:

res/mipmap-mdpi/ic\_launcher.webp  
res/mipmap-mdpi/ic\_launcher\_round.webp  
res/mipmap-hdpi/ic\_launcher.webp  
res/mipmap-hdpi/ic\_launcher\_round.webp  
res/mipmap-xhdpi/ic\_launcher.png  
res/mipmap-xhdpi/ic\_launcher\_round.webp  
res/mipmap-xxhdpi/ic\_launcher.webp  
res/mipmap-xxhdpi/ic\_launcher\_round.webp  
res/mipmap-xxxhdpi/ic\_launcher.webp  
res/mipmap-xxxhdpi/ic\_launcher\_round.webp

По сути, Android возвращается к растровым изображениям на старых устройствах без поддержки адаптивных значков.

Поздравляем, вы выполнили все действия по смене значка приложения!