# Содержание

[**Содержание**](#_v17csuetnfvp) **1**

[**Preparation**](#_dmhmm7585xau) **2**

[**Assignment #1**](#_nfz0uxn3y1mt) **3**

[Step 1](#_h2wkuixwhben) 4

[Step 2](#_g5jio9lgicq) 4

[Step 3](#_s04owduhpz1y) 5

[Step 4](#_wkmt1xank3rv) 6

[Step 6](#_73a29abdds3v) 7

[Step 7](#_yueop6v0f7wg) 8

[Step 8](#_3o0yaw8lkz3e) 9

[Step 9](#_x9l0r6k8cupm) 10

[Step 10](#_nu3kiny9p4dq) 11

[**Assignment #2**](#_nff9z1ldy8q6) **12**

[Step 1](#_c5wbjtl4tv2u) 13

[Step 2](#_vzgpwi855wcw) 13

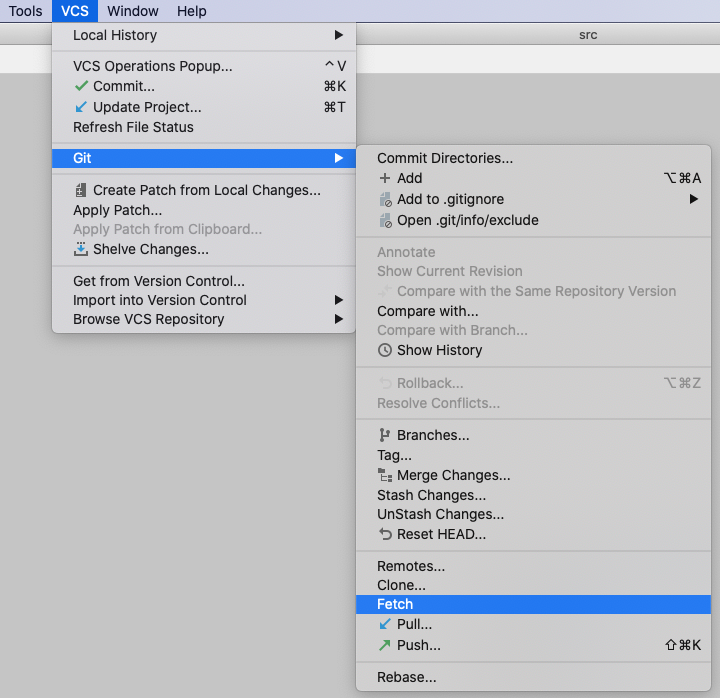
[Step 3](#_kz57gw2uftn) 14

[Step 4](#_6ngvbv2i1oy3) 15

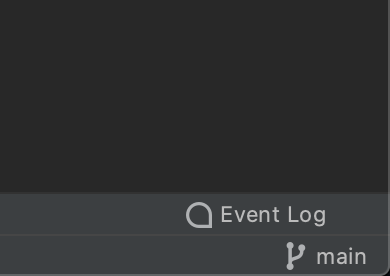
# 

# Preparation

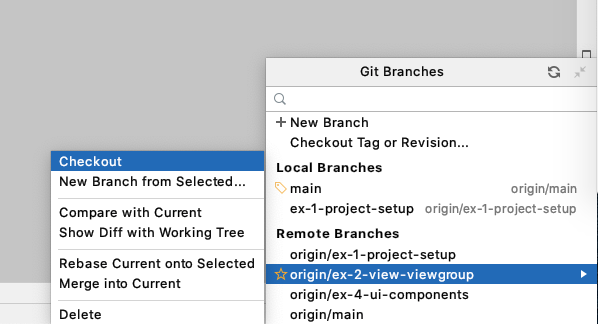
* Открываем проект с заданиями, которые мы скачивали с github на предыдущем [занятии](https://docs.google.com/document/d/e/2PACX-1vQgEDWRY0fG1tUAcvXYNHo_6fXBap3njwLV2R853QObA0qSQ5c-b0J4ATGJkXKAapzEgdpqFKXB7xgp/pub). Данный проект будет у вас называться "**fundamentals-2020-assignments**". Если не скачали, то можно это сделать [тут](https://github.com/Android-Academy-Global/fundamentals-2020-assignments)
* Если проект был ранее скачан, то необходимо обновить git. Для этого выбираем в studio VCS > Git > Fetch:



* Меняем ветку с **main** (главная) на ветку второго задания. В правом нижнем углу кликаем на **main**:



* Выбираем ветку "origin/ex-2-view-viewgroup" и кликаем опцию **Checkout**:



# 

# Assignment #1

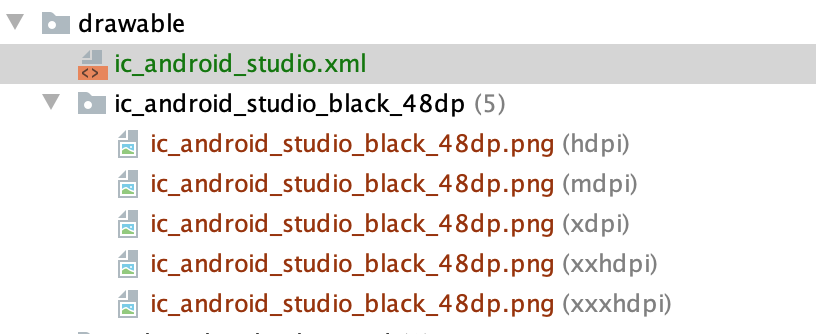
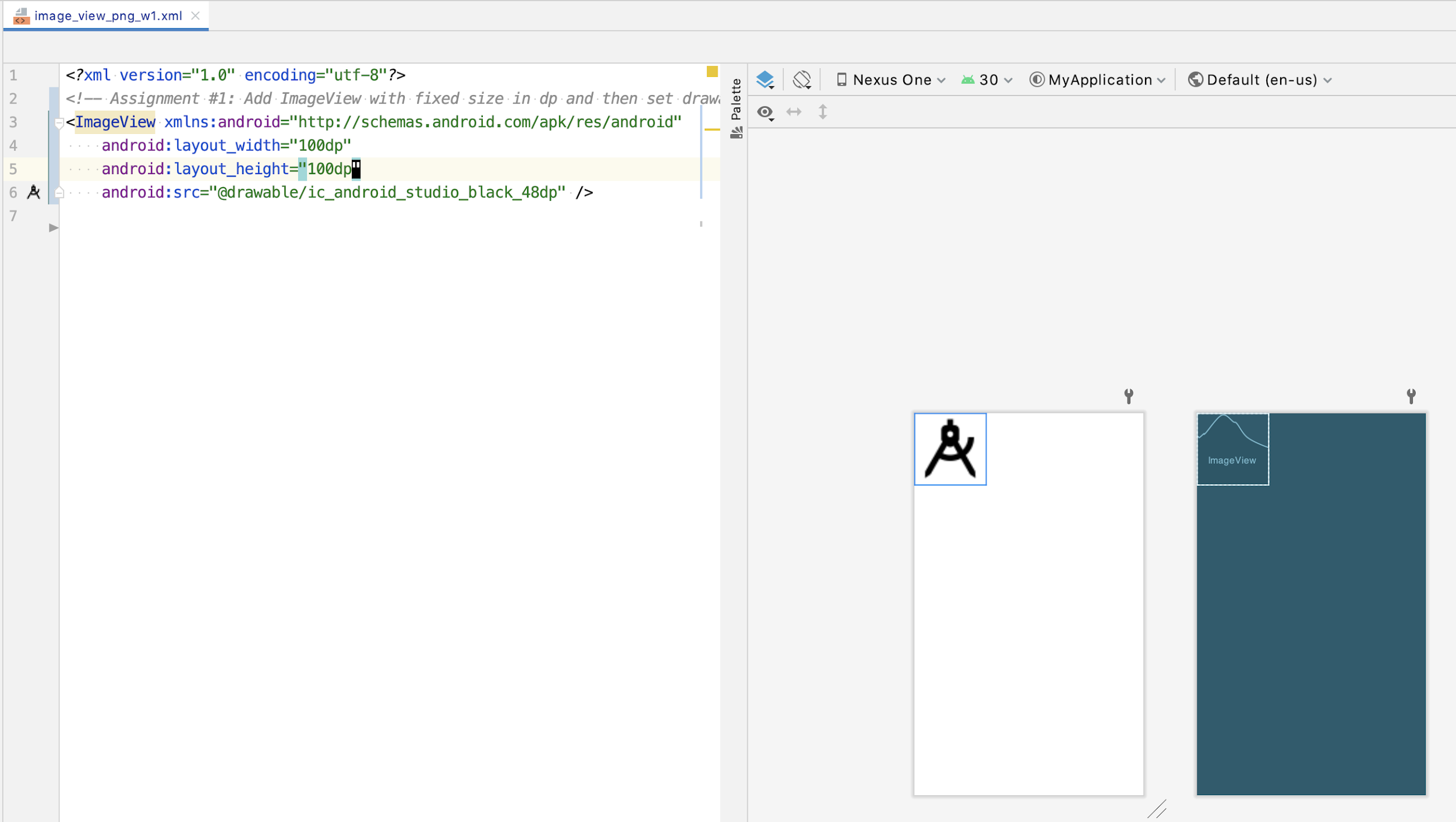
**Задача:**

*Добавить в проект png иконку для отображения в разных разрешениях.*

*Добавить в проект иконку в векторе.*

**Результат:**

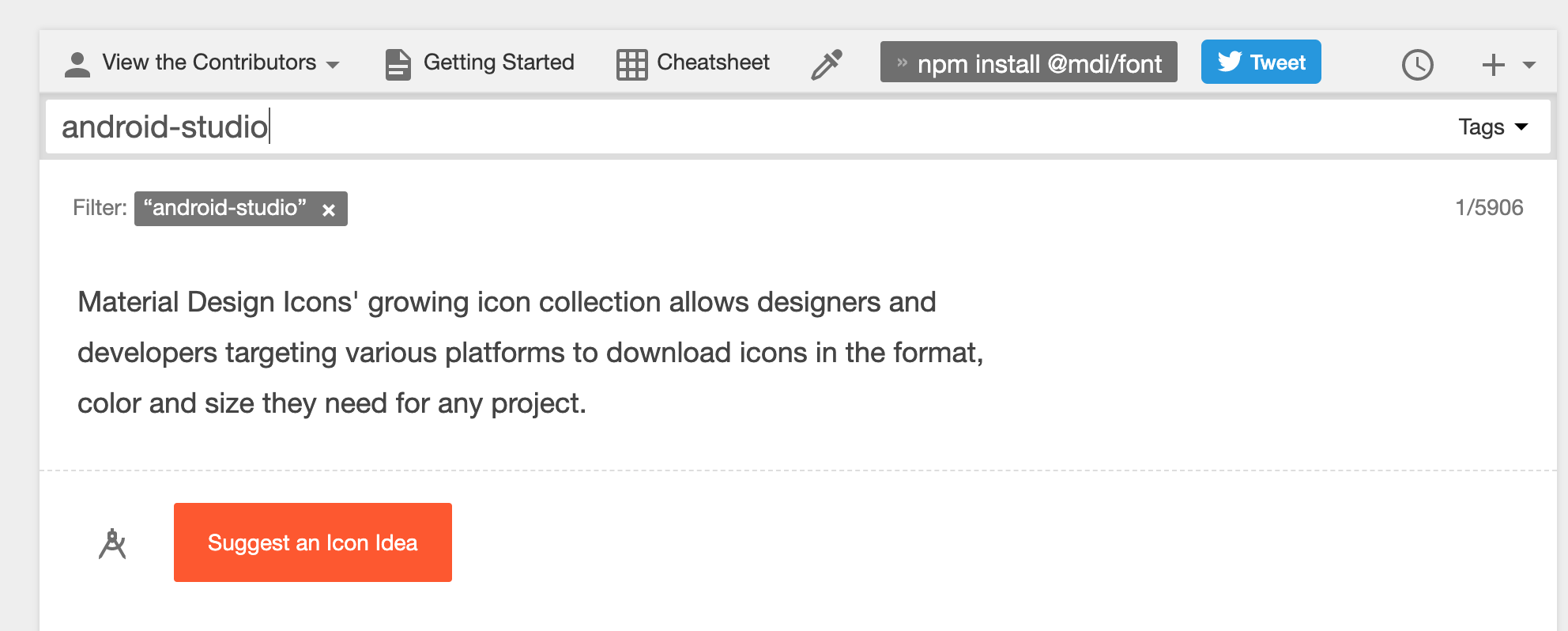
В конце задания у вас появятся картинки в разных drawable папках. И все это отображается в ImageView.

## 

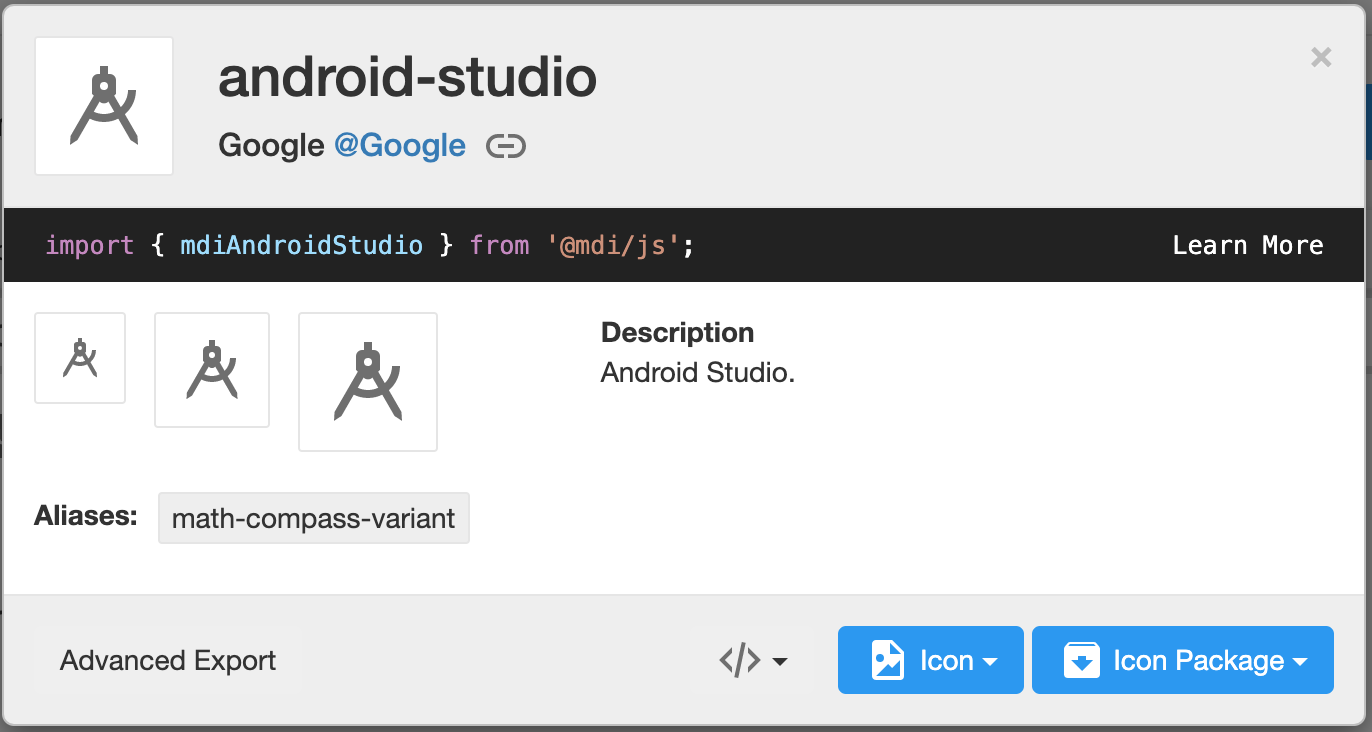
## Step 1

* Зайти на <https://materialdesignicons.com> и найти нужную иконку.  
  Мы для примера используем иконку с названием “android-studio”, однако вы для пробы можете взять любую иконку по нраву с этого сайта.

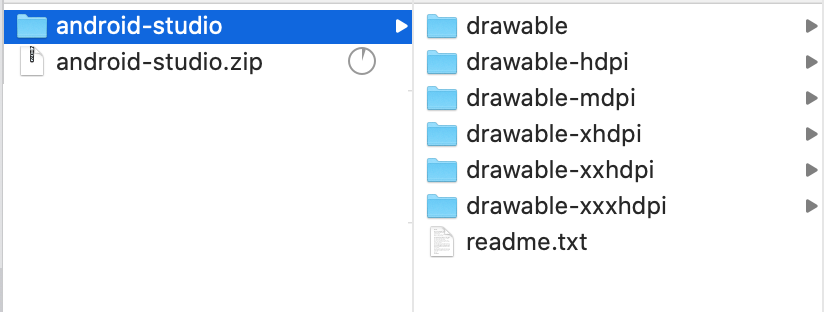
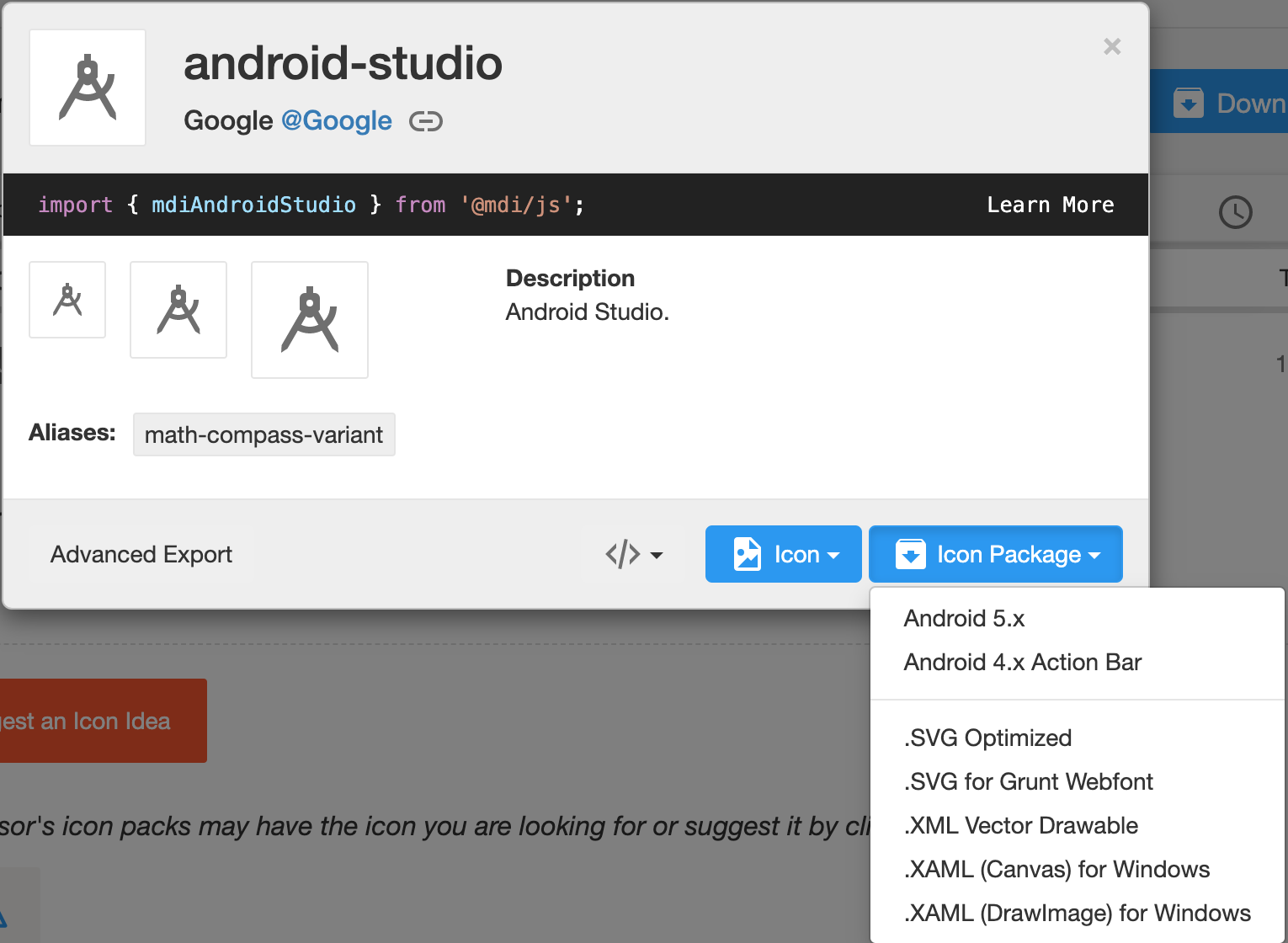


## Step 2

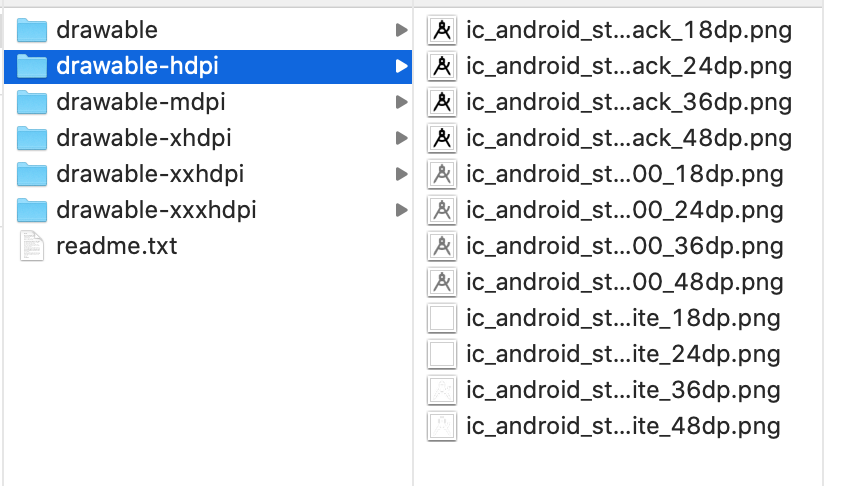
* При нажатии на найденную иконку покажется диалог с разными вариантами скачивания иконки.



Например, можно скачать целый package подготовленный для Android, где уже будут раскиданы файлы по папкам разных разрешений + в папке drawable будет лежать векторный xml.

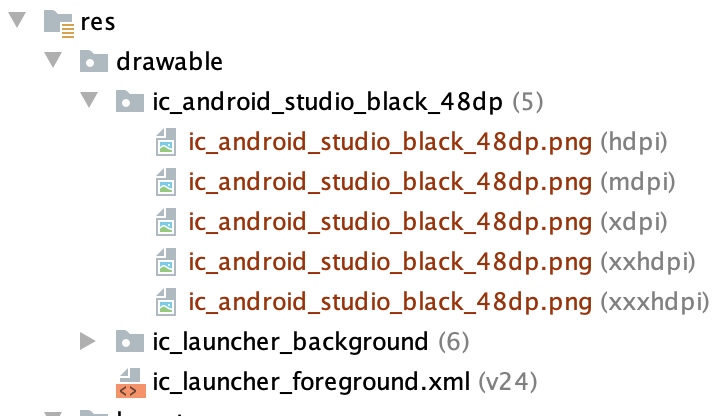


* Единственный минус такого варианта заключается в том, что скачиваются сразу все варианты размеров и цветов (черный, белый, серый) этой иконки и приходится либо удалять ненужные файлы, либо выбирать нужные и раскидывать их по папкам ручками. В каждой папке будет примерно такой набор

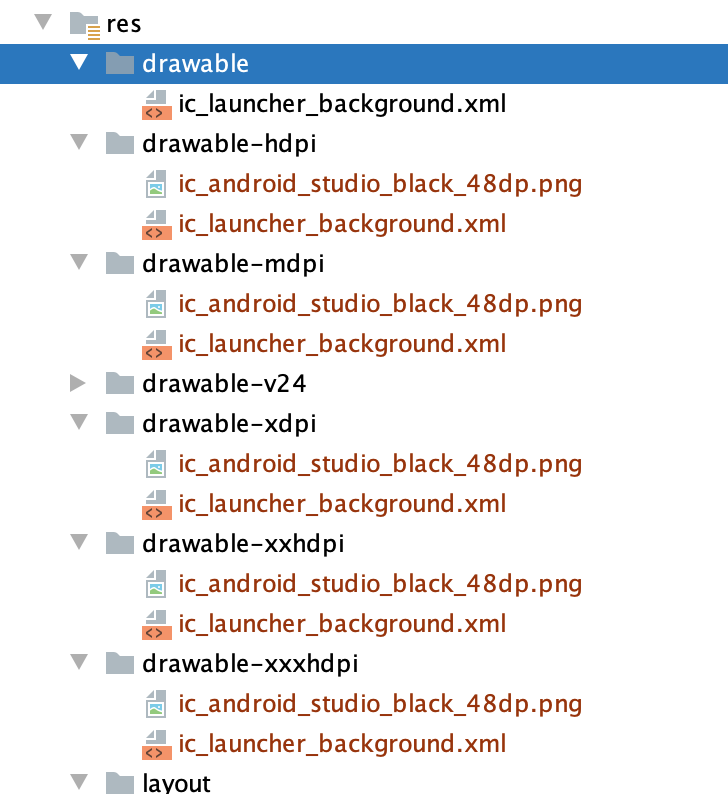


## Step 3

* Для проекта возьмем файл ic\_android\_studio\_black\_48dp.png
* Раскидайте файлы из скачанной папки в соответствующие папки своего проекта (hdpi, mdpi….) согласно тому как указано в названии файла-картинки
* В результате в проекте должно отобразиться что под разные разрешения у вас разные файлы:

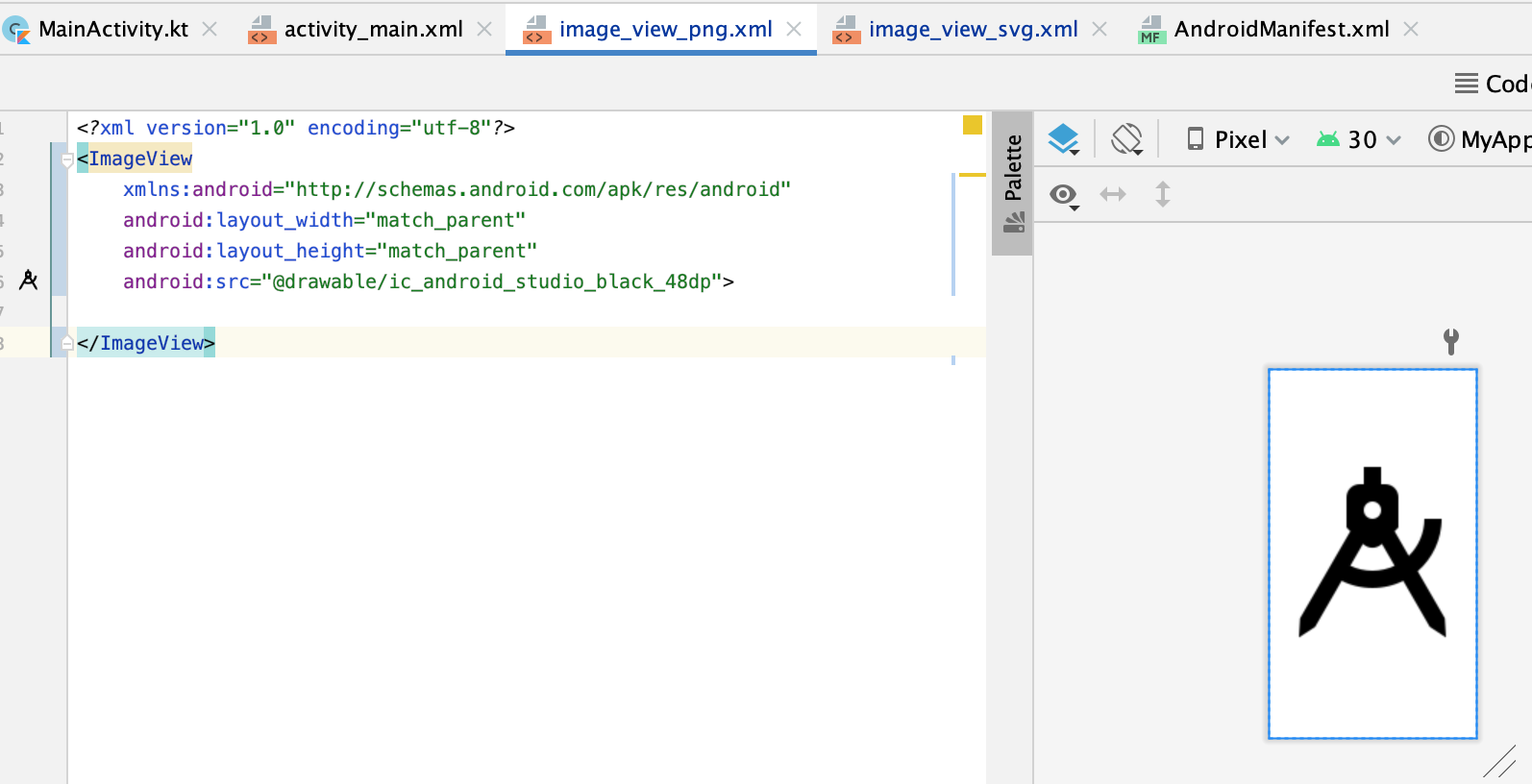


В структуре проекта же это будет выглядеть так

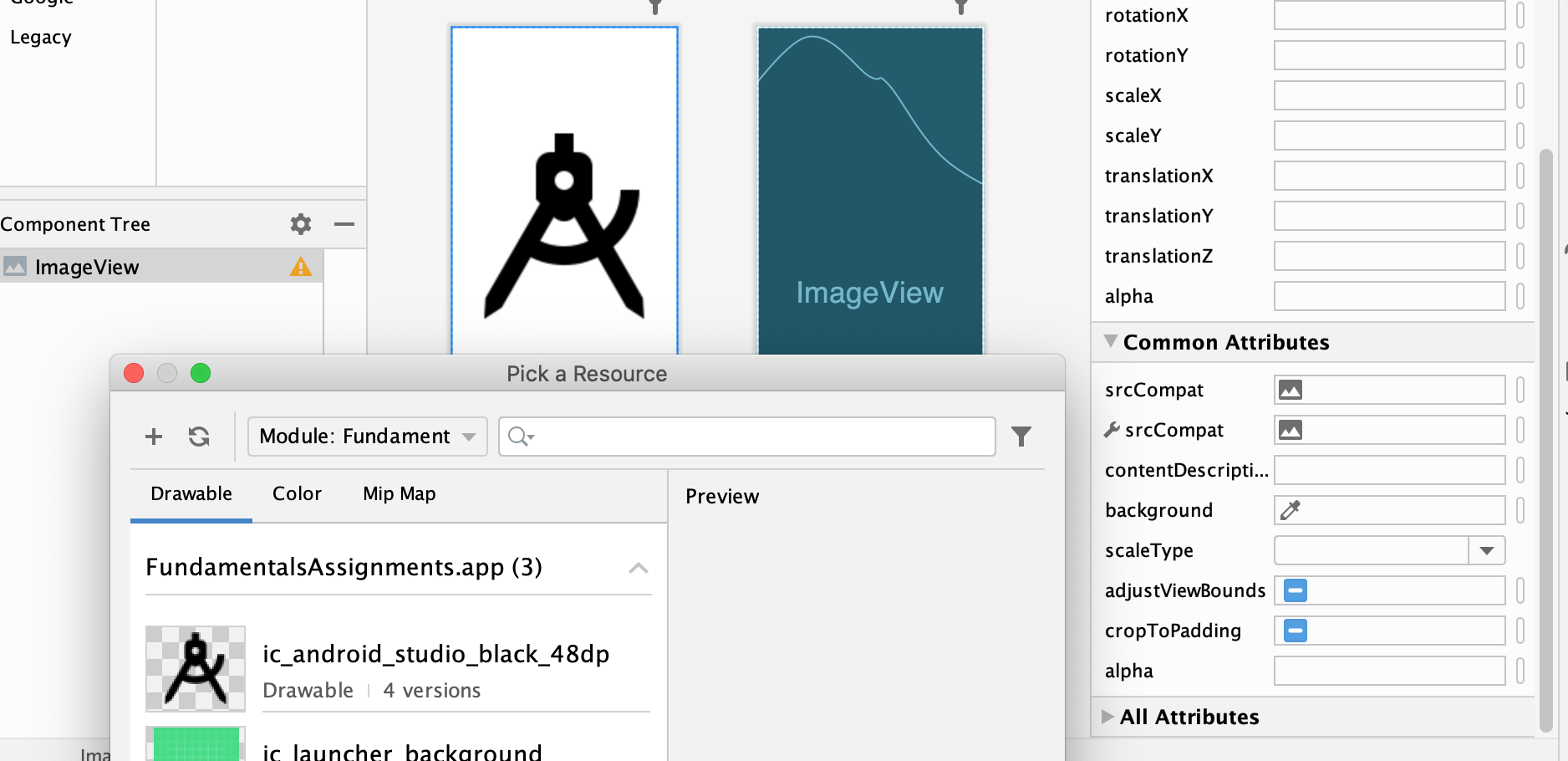


## Step 4

* Теперь можем закинуть эту иконку в ImageView
* Открываем файл image\_view\_png\_w1.xml в layout папке в ресурсах
* Добавляем ImageView и определяем ей размеры любые
* Для того, чтобы отобразить иконку в ImageView, можем либо в коде прописать "android:src" внутри тэга ImageView с указанием пути до иконки как мы это делали в примере в презентации:

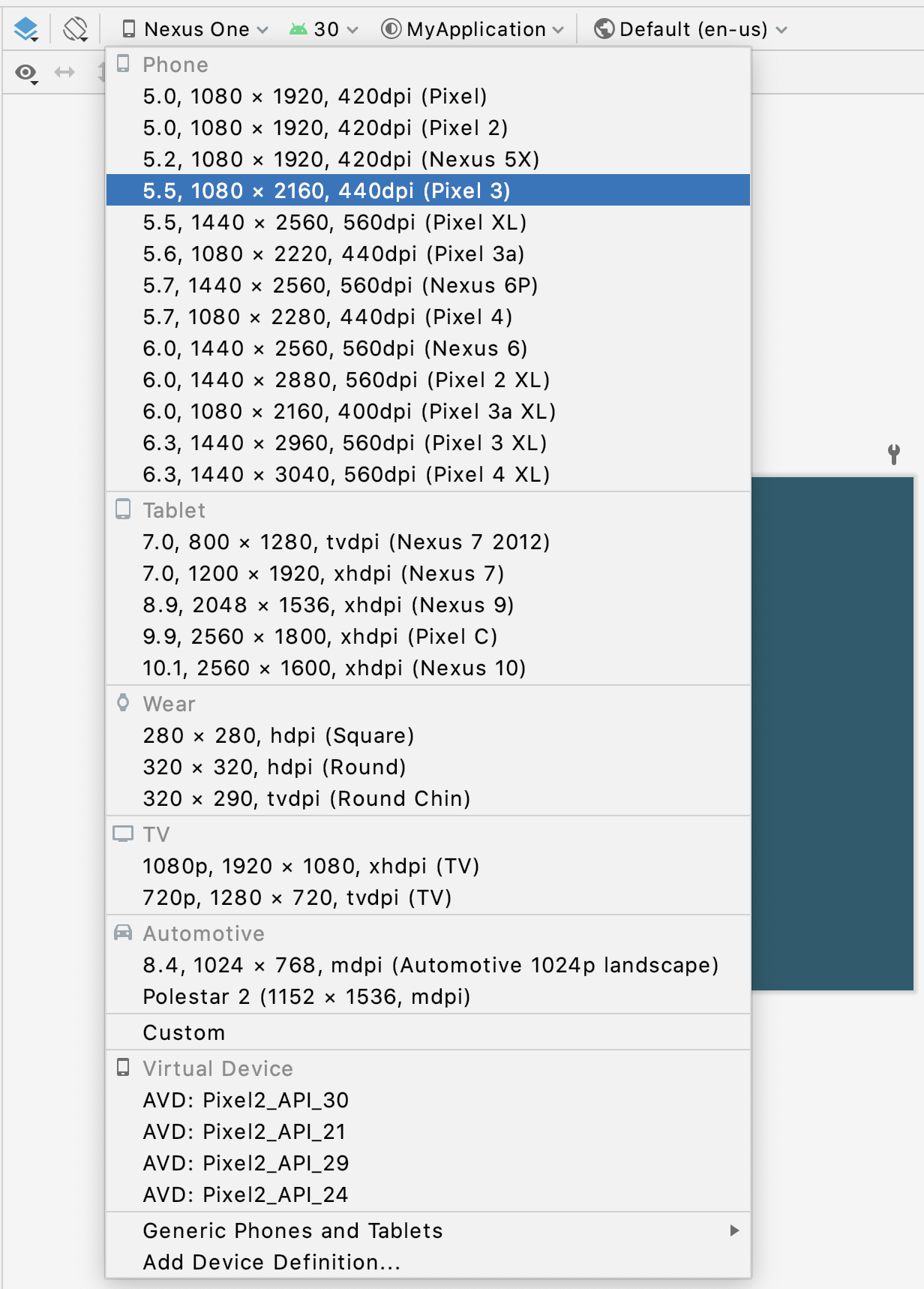


либо в Design представлении у LayoutEditor в панели "Attributes" найти srcCompat параметр у этого ImageView:



## Step 6

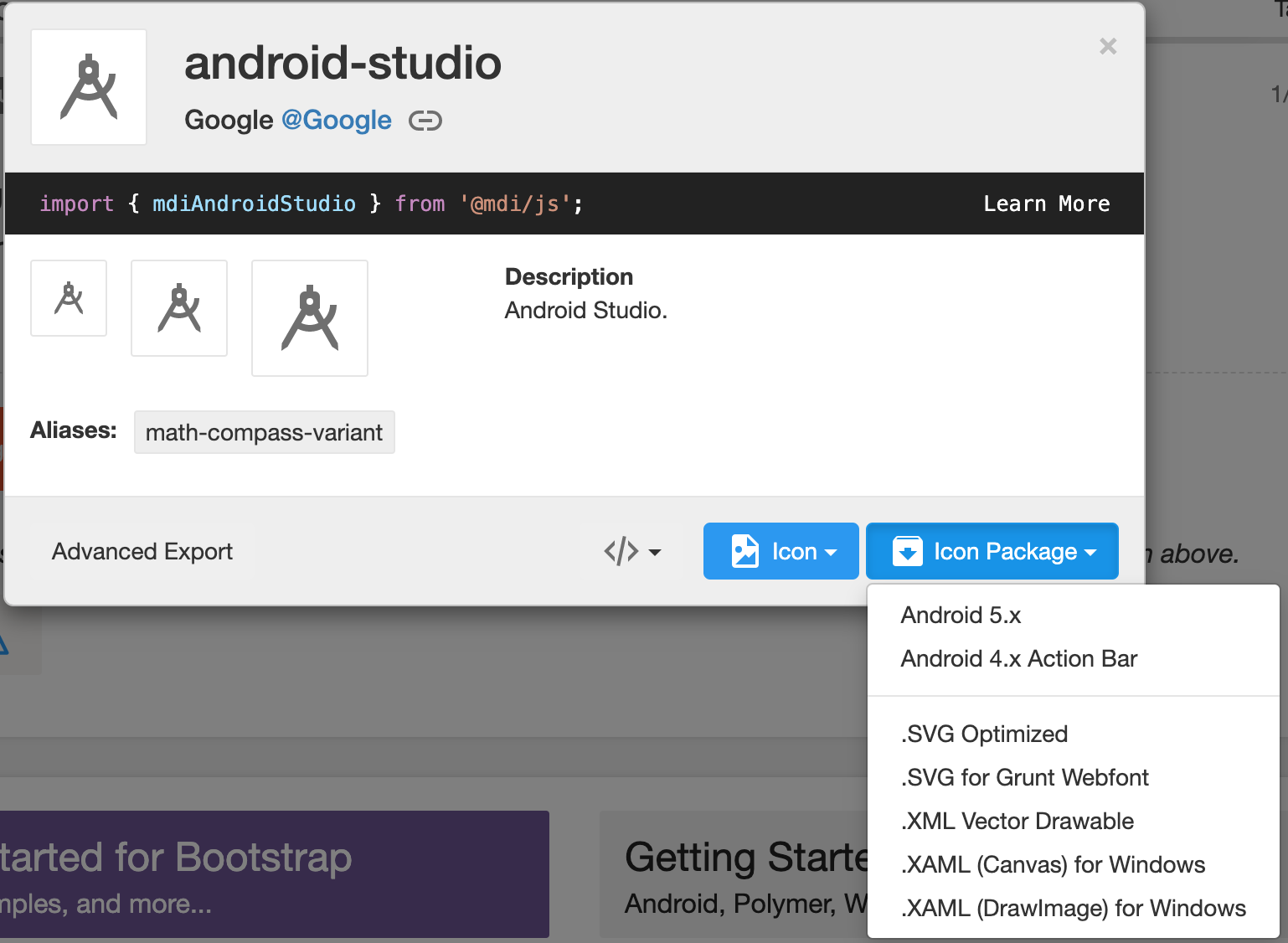
* Убедимся, что на разных устройствах все выглядит одинаково хорошо. Для этого посмотри как картинка выглядит на разных эмуляторах в Design представлении:



## Step 7

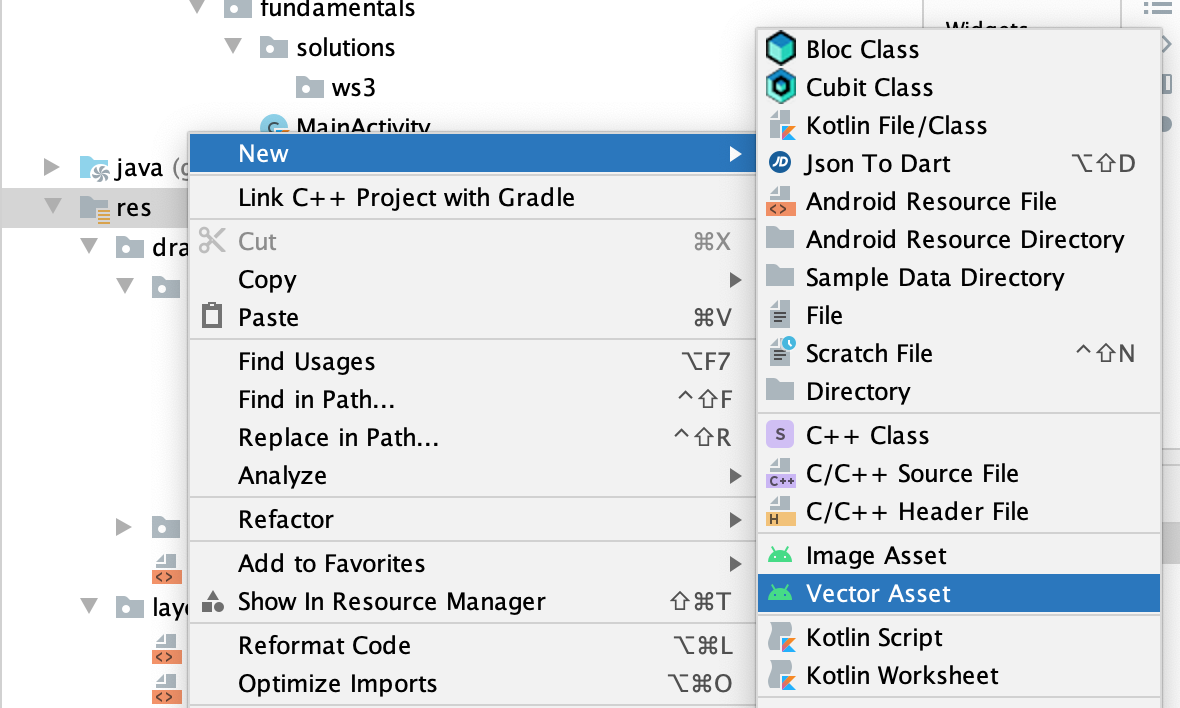
Теперь попробуем добавить векторную картинку. Если мы уже скачали package с полными данными, который был описан в шаге 2 выше, то у нас уже есть векторное представление в папке drawable. Но зачастую svg могут придти от дизайнера или еще откуда-то, поэтому проделаем лучше полностью операцию по превращению вектора в xml:

* Скачиваем XML с того же сайта <https://materialdesignicons.com> через SVG Optimized. Видим что можно сразу скачать .xml vector drawable, но так же слишком скучно, да?)



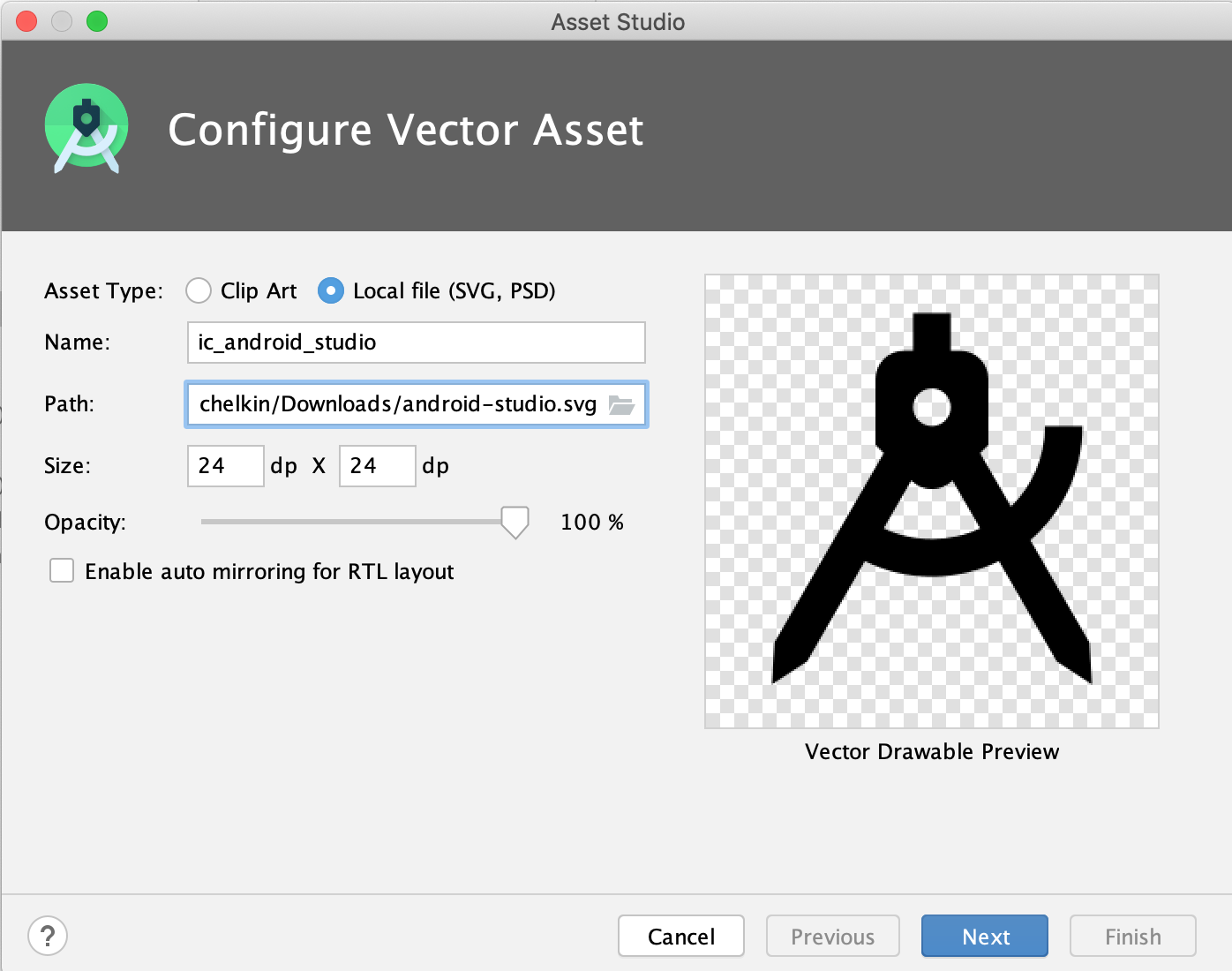
## Step 8

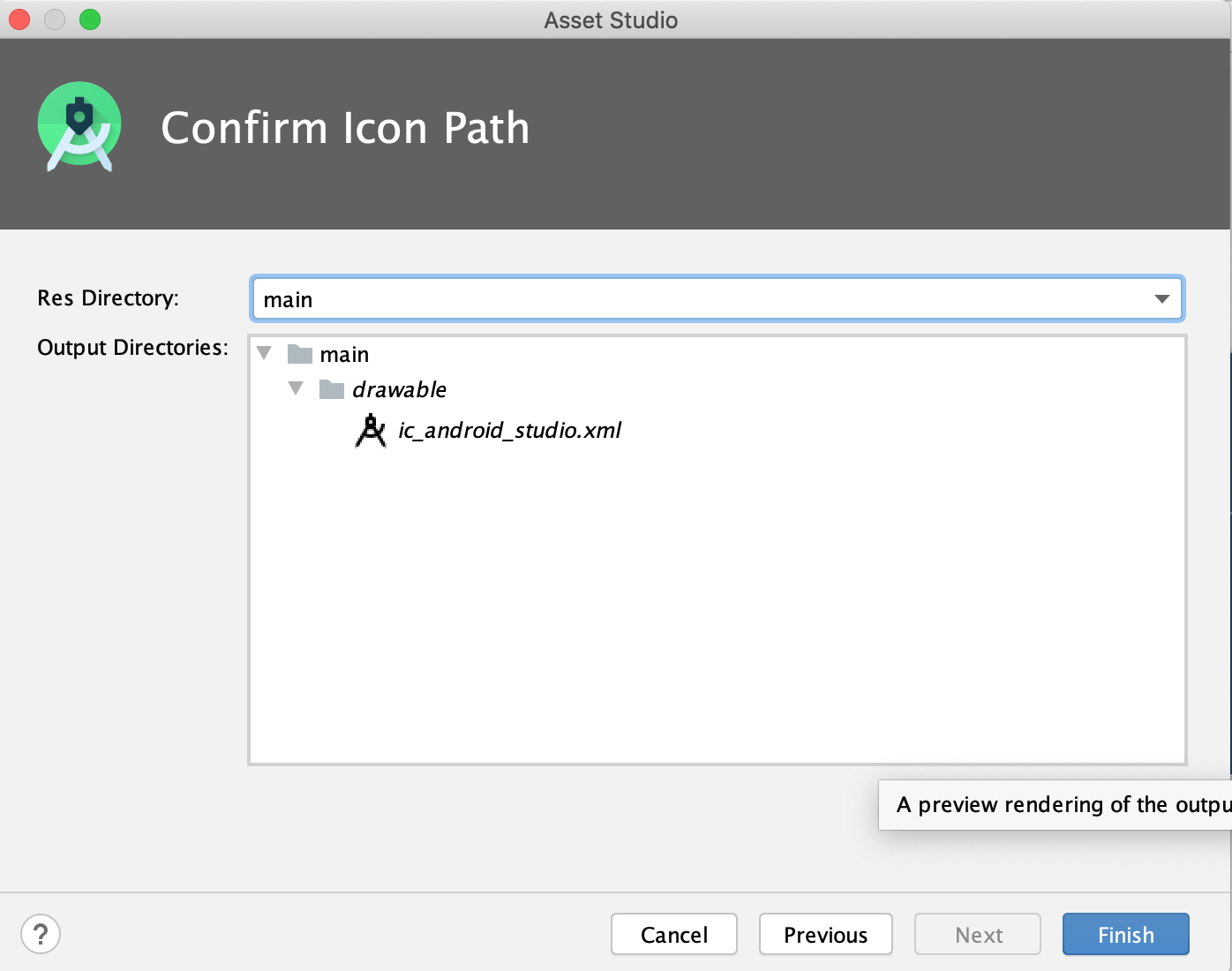
* Нажимаем правой кнопкой мыши на папку res в Android Studio, выбираем "New->Vector Asset"

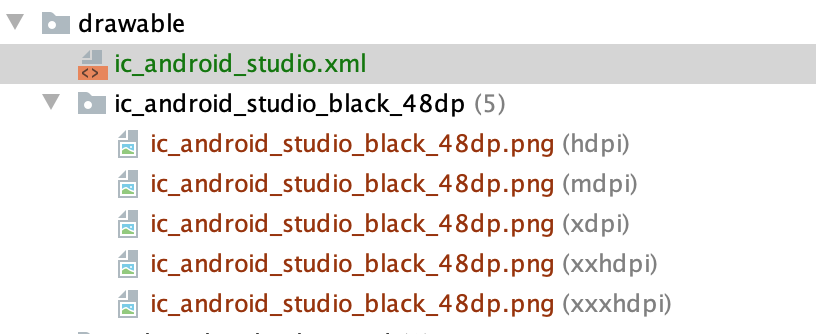


## Step 9

* Появится такой диалог, в нем надо выбрать Local file и далее указать путь к файлу векторной графики и выходной оптимальный размер картинки и где именно в res она будет храниться. Profit!

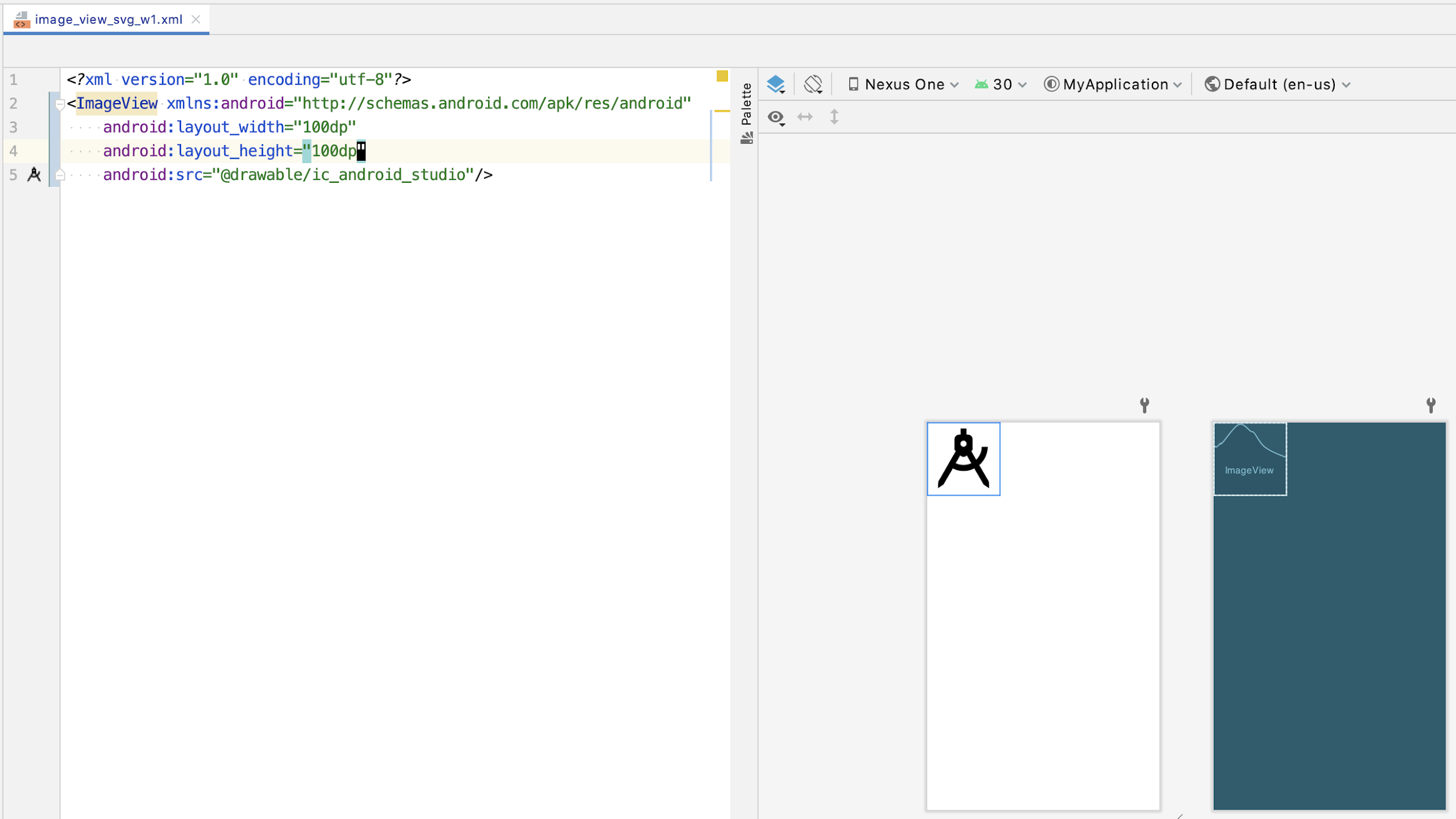




В результате этот ресурс появится в корне drawable  
  


## Step 10

* Теперь откроем файл image\_view\_svg\_w1.xml в layout и отобразим в нем наш drawable. C точки зрения xml ничего не изменилось - работаем также как и в первом случае



# 

# Assignment #2

**Задача:**

*Познакомиться с основными типа вью и параметрами вью*

**Результат:**

*В конце каждого из шага у вас будет представление для каждого из виджетов.*

**Описание:**

*В папке layout есть файлы "edit\_text\_w1.xml", "button\_w1.xml", "text\_view\_w1.xml". К каждому нужно добавить параметры, чтобы получился результат как на скриншоте в шагах.*

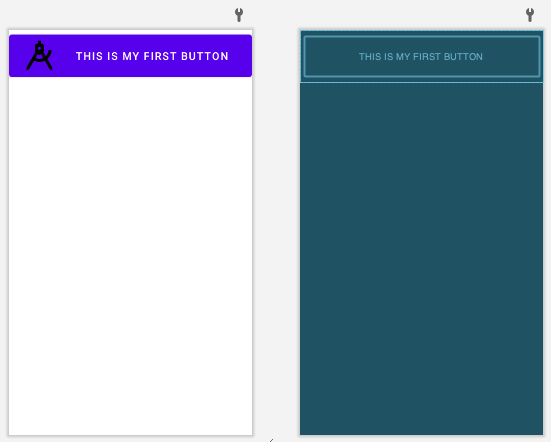
*Можно выполнять как через LayoutEditor, так и через xml код с помощью тэгов. А лучше попробовать и так и так, чтобы быстрее привыкнуть к атрибутам вьюх*

## 

## Step 1

Попробуем выставить различные размеры вью. Необходимо сделать так как будет показано на картинке.

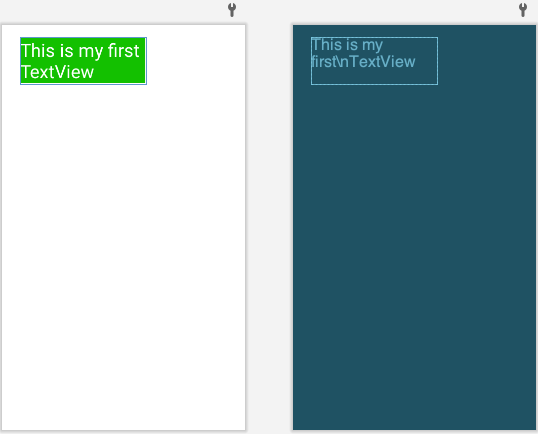
* Открываем файл "button\_w1.xml", который лежит в layout папке.
* Добавляем виджет Button
* С помощью layout\_width и layout\_height растягиваем виджет Button по ширине и сделать так, чтобы высота была выровнена по размеру внутреннего контента чтобы получилось как на скриншоте.
* Добавляем текст через атрибут "android:text"
* Слева от контента кнопки добавить иконку "ic\_android\_studio" из 1го задания. Для этого есть атрибут drawableLeft. Таким же образом можно добавить картинку и с других сторон кнопки. Такой же атрибут есть и у EditText



## Step 2

Пробуем выставлять "margin" и основные атрибуты TextView

* Открываем "text\_view\_w1.xml", который лежит в layout папке
* Добавляем TextView виджет, где высота и ширина - по контенту
* Выставляем цвет background в "#12C700"
* Выставляем цвет текста через атрибут "android:textColor" в "FFFFFF"
* Выставляем размер шрифта через атрибут "android:textSize" в 24sp
* Выставляем отступ слева в 24dp, а отступ сверху в 16dp и добавляем текст как показано все на картинке:

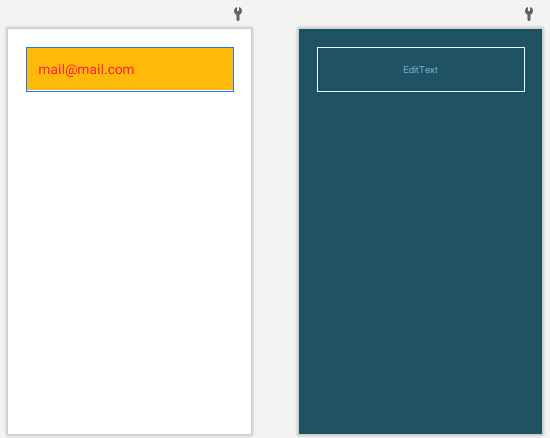


**Дополнительно:**  
*Можно попробовать выставить разную высоту и ширину, а также разные отступы.*

## Step 3

Пробуем выставить разный "padding" и основные атрибуты EditText

* Открываем "edit\_text\_w1.xml", который лежит в layout папке
* Добавляем EditTextView виджет, где ширина по экрану, а высота по контенту
* Выставляем цвет background в "#FFC107"
* Выставляем отступ от всех краев в 24dp
* Выставляем отступ внутреннего контента EditText - 16dp
* Добавляем тэг "android:hint" и выставляем ему значение "mail@mail.com". Этот тэг отвечает за первичное заполнение, когда EditText пустой и является подсказкой что можно записать в этот EditText (например, почту при регистрации)
* Выставляем цвет hint в "#FF2344"
* Подтверждаем, что пользователь будет вводить в поле именно email. Для этого добавляем android:inputType="textEmailAddress"



## Step 4

Посмотрите на остальные атрибуты этих виджетов в LayoutEditor и на что они влияют. Например, параметр backgroundTint у кнопки, enabled у кнопки и editText, maxLines у textView, fontFamily и другие кастомизаторы цвета.

Развлекайтесь ;)