**Создание персонализированных элементов**

Drawable - это общая концепция графики, которую можно выводить на экран. Drawable используются для определения форм, цветов, границ, градиентов и т.д. которые затем могут быть применены к элементам.

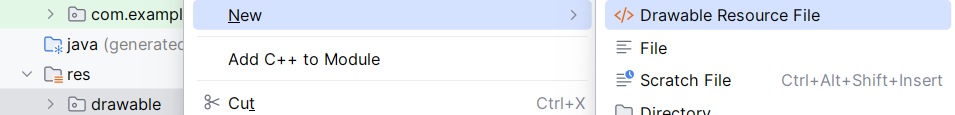
Drawable, как правило, определяются в XML, а затем могут быть применены к элементу через XML или Java.

**Рисование фигур**

[Shape Drawable](https://translated.turbopages.org/proxy_u/en-ru.ru.0495c2d7-66f0db34-a97a08d1-74722d776562/developer.android.com/guide/topics/resources/drawable-resource.html#Shape) - это XML-файл, который определяет геометрическую фигуру, включая цвета и градиенты. Они используются для создания сложных фигур, которые затем можно прикрепить в качестве фона экрану или элементу на экране. Например, вы можете использовать фигуру, чтобы изменить форму, границу и добавить градиент.

Фигура - это просто набор свойств, которые объединены для описания фона. Фигура может быть описана с помощью таких свойств, как corners для округления, gradient для фона, padding для отсупа, solid для цветов фона и stroke для границы.

Создайте новый файл drawable, для того что бы он содержал необходимые вам настройки. Чтобы создать файл drawable, нажмите: app -> res -> drawable (правый клик) -> New -> Drawable resource file и назовите его как хотите



**Настройка drawable**

Чтобы настроить drawable, понадобятся несколько элементов и атрибутов:

<shape> - Рисуемая форма, корневой элемент

Атрибуты:

xmlns:android - обязательный атрибут, определяющий пространство имен XML, значние должно иметь вид "http://schemas.android.com/apk/res/android"

shape - он определяет форму используемого виджета, принимает следующие значения:

* rectangle - прямоугольник, заполняющий содержащей его элемент. Это форма по умолчанию;
* oval - овальная форма, соответствующая размерам содержащего элемента;
* line - горизонтальная линия, охватывающая ширину содержащего элемента. Эта фигура требует, чтобы элемент <stroke> определял ширину линии;
* ring - форма кольца, для него свойственный отдельные атрибуты которые применяются для определения отдельных частей кольца.

<corners> - этот элемент создает закругленные углы фигуры. Применяется только в том случае, если фигура представляет собой прямоугольник.

Атрибуты:

android:radius - определяет радиус для всех углов в виде значения размера или ресурса размера . Это переопределяется для каждого угла следующими атрибутами.

android:topLeftRadius, android:topRightRadius, android:bottomLeftRadius, android:bottomRightRadius - определяют радиус конкретного угла

Изначально каждый угол должен иметь радиус угла больше 1, иначе углы не будут закруглены. Если вы хотите, чтобы определенные углы не были закруглены, обходным путем является использование android:radius чтобы установить радиус угла по умолчанию больше 1, а затем переопределить каждый угол, который не должен быть закруглен, указав 0 («0dp»).

<gradient> - задает цвет градиента для фигуры.

Атрибуты:

android:angle - целое число . Угол градиента в градусах. 0 слева направо, 90 снизу вверх.

android:centerX - относительное положение по оси X центра градиента (0–1,0).

android:centerY - относительное положение по оси Y центра градиента (0–1,0).

android:startColor - цвет располагающийся в начале градиента в виде шестнадцатеричного значения или ресурса цвета.

android:centerColor - цвет, который находится между начальным и конечным цветами в виде шестнадцатеричного значения или ресурса цвета.

android:endColor - цвет располагающийся в конце градиента в виде шестнадцатеричного значения или ресурса цвета.

android:gradientRadius - радиус градиента. Применяется только тогда, когда android:type="radial" .

android:type - тип применяемого шаблона градиента:

* linear - линейный градиент. Это значение по умолчанию;
* radial - радиальный градиент. Начальный цвет — это центральный цвет;
* sweep - широкая линия градиента.

<padding> - отступ, применяемый к содержащему контенту фигуры. Изменяет положение содержимого, а не самой формы.

Атрибуты:

android:left, android:top, android:right, android:bottom - отступ с определенной стороны.

<size> - размер фигуры.

Атрибуты:

android:height, android:width - высота и ширина фигуры.

По умолчанию фигура масштабируется до размера контейнера, пропорционального определенным здесь размерам. Когда вы используете фигуру в ImageView , вы можете ограничить масштабирование, установив android:scaleType со значением "center" .

<solid> - Сплошной цвет для заполнения фигуры.

Атрибуты:

android:color - цвет, применяемый к фигуре, в виде шестнадцатеричного значения или ресурса цвета .

<stroke> - линия обводки фигуры.

Атрибуты:

android:width - толщина линии.

android:color - цвет линии.

android:dashGap - расстояние между штрихами линий. Действует только в том случае, если установлен android:dashWidth .

android:dashWidth - размер каждой пунктирной линии. Действует только в том случае, если установлен android:dashGap .

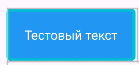
**Однотонные фигуры**

Пример рисования скругленного прямоугольника с границей:



Текстовое поле меняет форму с помощью атрибута background:

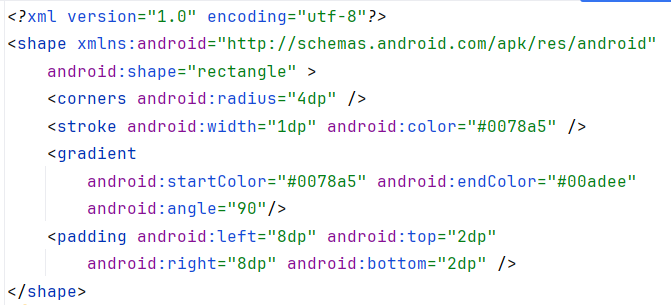




**Градиентные фигуры**

Фигуры также поддерживают градиенты фона, а также вспомогательные свойства, такие как startColor, centerColor, endColor, angle. С помощью атрибута type можно выбрать различные градиенты, такие как радиальный, линейный или развертка.

Пример простой формы линейного градиента:



Создайте стиль который будет обеспечивать применение созданного Drawable на вашу кнопку, убедитесь что кнопка не наследует Material Design, так как у данной библиотеки множество своих собственных настроек и свойств

#### 

Кнопка на которую применен данный стиль выглядит следующим образом:

#### 

Пример радиального градиента:



Применим этот Drawable на текстовое поле:

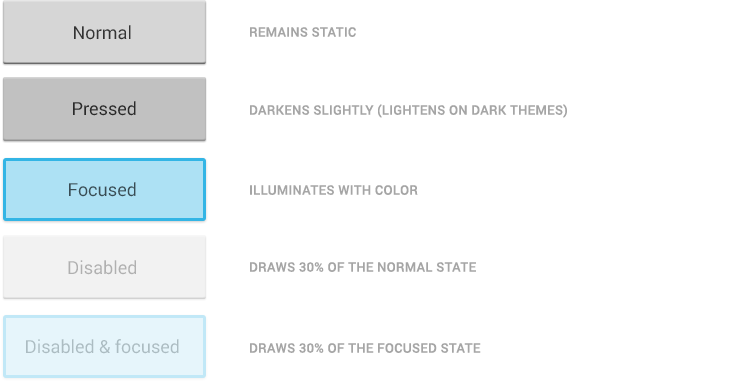


Используя однотонные фигуры и градиенты, мы можем настраивать внешний вид кнопок, макетов и других видов, не требуя использования каких-либо изображений.

**Список состояний**

[StateListDrawable](https://translated.turbopages.org/proxy_u/en-ru.ru.0495c2d7-66f0db34-a97a08d1-74722d776562/developer.android.com/guide/topics/resources/drawable-resource.html#StateList) - это рисованный объект, определенный в XML, который использует несколько разных изображений для представления одной и той же графики, в зависимости от состояния объекта. Например, виджет кнопки может находиться в одном из нескольких различных состояниях (нажато, сфокусировано или ни в одном из них), и, используя возможность рисования списка состояний, вы можете предоставить различное фоновое изображение для каждого состояния.

Список состояний поддерживает различные состояния представления, такие как android:state\_pressed, android:state\_focused, android:state\_enabled, android:state\_selected и многие другие. На рисунке ниже показаны все основные состояния, которые могут быть представлены:



Например, XML-файл списка состояний для кнопки может выглядеть следующим образом.

Добавьте 3 drawable файла со свойствами для изменения кнопки.

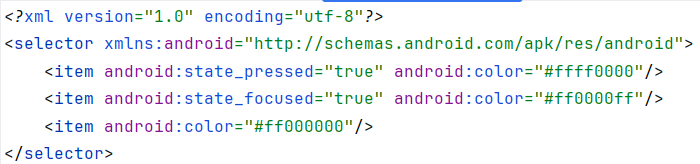
Создайте XML-файл списка состояний для кнопки:



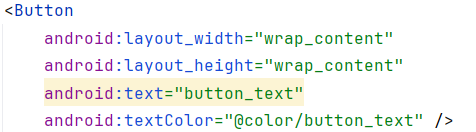
Установите данный drawable кнопке с помощью атрибута background.

Теперь, когда кнопка нажата или выбрана, drawable, используемый для представления, изменится соответствующим образом.

Существуют также селекторы состояния цвета, которые позволяют выбирать цвет на основе состояния элемента, например, создайте в папке res подпапку color, в папке color создайте файл с именем button\_text.xml:



Свойства данного drawable будут применяться к любому полю, которое принимает значение цвета, например, к textColor атрибуту кнопки:



Использование списков состояний позволяет легко определять динамические элементы, которые реагируют на нажатие, выделение, включенные или другие связанные состояния.

**Создание списка слоев**

LayerDrawable - это drawable объект, который управляет массивом других drawable объектов. Каждый drawable в списке рисуется в порядке следования списка — последний drawable в списке рисуется сверху остальных. Каждый drawable представлен <item> элементом внутри одного <layer-list> элемента.

LayerList можно использовать для рисования множества других drawable (фигур, изображений и т.д.), и позиционирования их относительно друг друга. Слои по умолчанию располагаются друг на друге, затем слои могут быть сдвинуты с помощью свойств left, right, top и bottom.

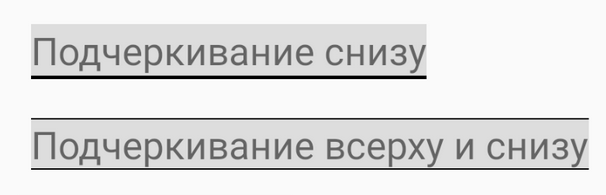
Например, два drawable с измененным фоном и подчеркиваниями:

Drawable с нижним подчеркиванием:



Drawable с нижним и верхним подчеркиванием:

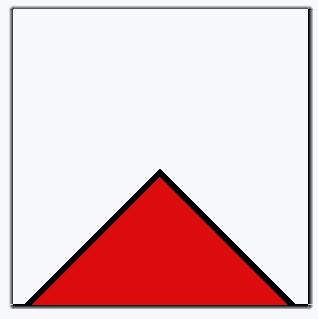




LayerList можно использовать для рисования фигур:

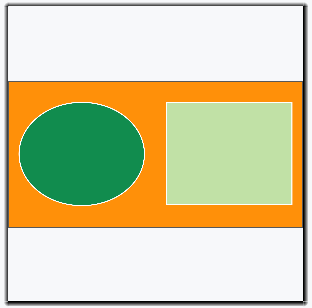
Треугольник:





Несколько фигур относительно друг друга:





**Задание для самостоятельной работы:**

1. Разработайте приложение с интерфейсом соответствующим следующему макету. На макете должны быть расположены только три кнопки и контейнер для них

### 

### 