**Использование командных элементов управления**

**Кнопки**

Действие выполняется сразу после нажатия на кнопку. Часто особым образом выделяется кнопка по умолчанию, соответствующая наиболее часто используемому действию.

Кнопка - удобный и привлекательный с визуальной точки зрения элемент управления. Весь ее облик подсказывает, что на нее можно нажать, и это характеризует ее ожидаемое назначение. Рекомендуется изменять внешний вид нажатой кнопки, так как это облегчает понимание работы программы пользователем.

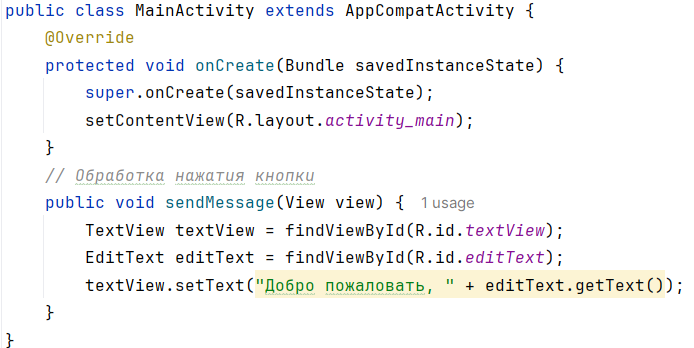
Для создания кнопок используется элемент **Button**

Ключевым атрибутом **Button** является **onClick** который задает обработчик нажатия кнопки

Для создания кнопки напишем следующий код в **activity\_main.xml**

****

При помощью атрибута **android:onClick** можно задать метод в коде Java, который будет обрабатывать нажатия кнопки. Так, в вышеприведенном примере это метод sendMessage. Теперь перейдем к коду **MainActivity** и пропишем в нем такой метод:



При создании метода обработки нажатия следует учитывать следующие моменты:

* Метод должен объявляться с модификатором public
* Должен возвращать значение void
* В качестве параметра принимать объект View. Этот объект View и представляет собой нажатую кнопку

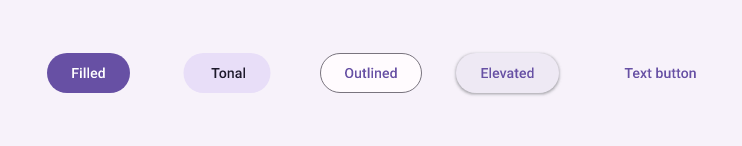
В данном случае после нажатия на кнопку в **TextView** выводится текст из **EditText**

Так же мы можем объявить метод **android:onClick** в коде Java с помощью слушателя **setOnClickListener**

****

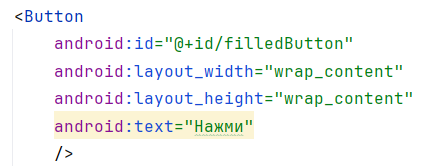
**Типы кнопок**

Основным инструментом для взаимодействия разработчика с пользователем является интерфейс приложения. Используемые элементы и их расположение напрямую определяют удобство пользователя. Например, кнопку выхода обычно изображают в виде кнопки с крестиком. Для унификации интерфейса можно использовать готовые решения встроенные в библиотеки интерфейса в Android. Приведенные ниже кнопки не отличаются по функционалу, только по визуальному оформлению, для их установки используется атрибут style работу которого подробнее вы рассмотрите в дальнейшем.



**Filled (Кнопка с заливкой)**

Это базовый вариант кнопки, для него не нужно устанавливать атрибут style. Сплошной фон с контрастным текстом делает кнопку более выделенной. Она используется для завершающих или разблокирующих действий.





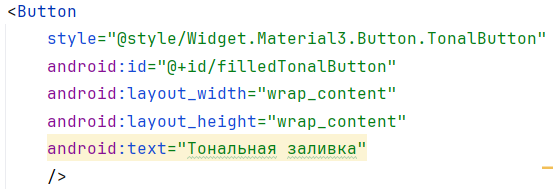
Кнопка - сложный компонент который может содержать другие компоненты. Добавим иконку к кнопке





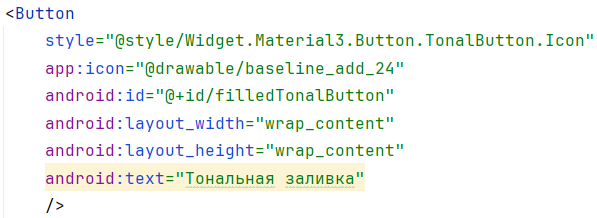
**Filled tonal (Кнопка с тональной заливкой)**

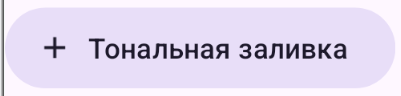
Кнопки с тональной заливкой имеют более светлый цвет фона и более темный цвет надписи, что делает их менее заметными визуально, чем обычная кнопка с заливкой. Они по-прежнему используются для заключительных или разблокирующих действий, но могут быть лучше, когда эти действия не требуют такого большого внимания.





Добавим иконку





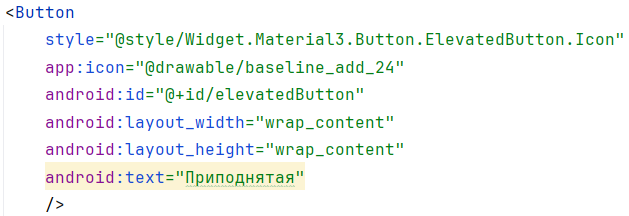
**Elevated (Приподнятые кнопки)**

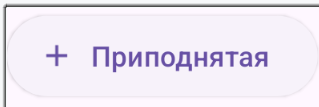
Приподнятые кнопки это обведенные тенью кнопки. Благодаря тени они хорошо визуально отделяются от фона. Они предназначены для основных действий в приложении, таких как "отправить" и "сохранить". Эффект тени подчеркивает важность кнопки.





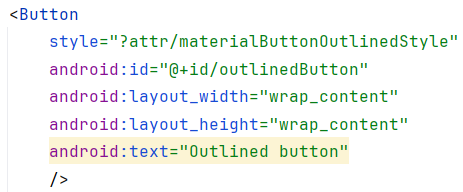
Добавим иконку

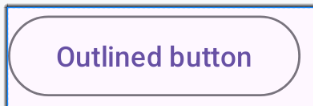




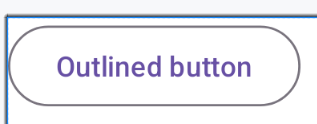
**Outlined (Выделенные кнопки)**

Выделенные кнопки предназначены для действий, которые требуют внимания, но не являются основными, например “Просмотреть все” или “Добавить в корзину”. Эта кнопка также используется для того, чтобы дать кому-то возможность изменить свое решение в процессе, например “Отмена” или “Назад”.

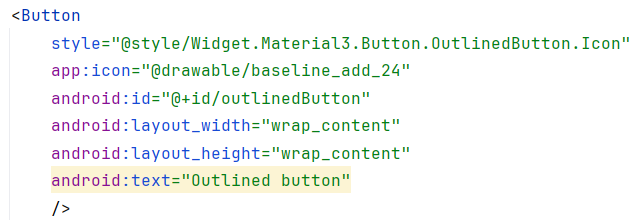


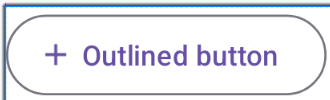


Изменим фон контейнера на белый



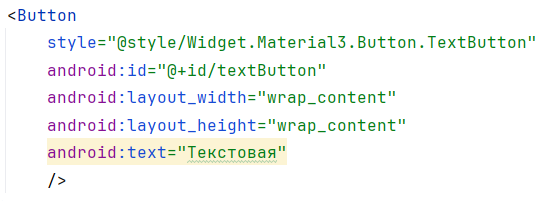
Добавим иконку

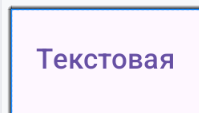




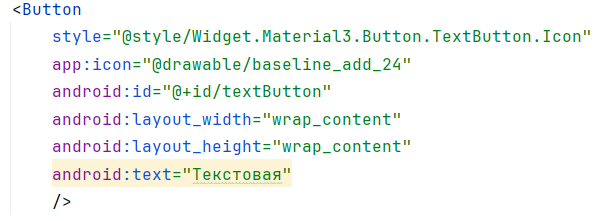
**Text Button (Текстовые кнопки)**

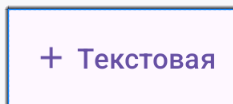
Текстовые кнопки менее заметны визуально, поэтому их следует использовать для действий с небольшим акцентом, например, при представлении нескольких вариантов. Они визуально похожи на обычные текстовые поля, что может вызвать путанницу у пользователя.





Добавим иконку



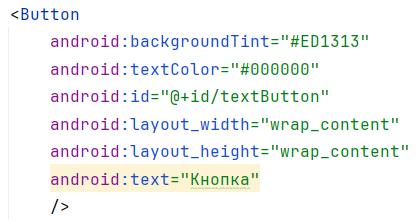


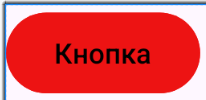
**Визуальное отображение кнопок**

Рассмотрим некоторые свойства для изменения визуального отображения кнопок

Цвет фона задается с помощью атрибута backgroundTint

Здесь в качестве фонового цвета применяется красный, а в качестве цвета содержимого - черный:

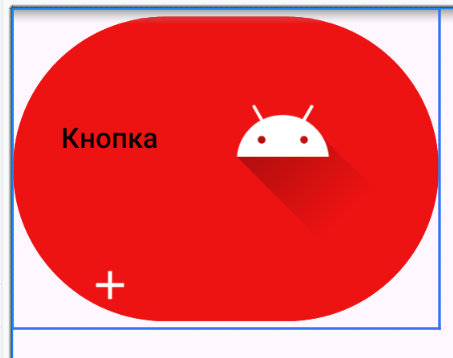




Вы можете устанавливать изображения не только с помощью атрибута icon а еще с помощью семейства атрибутов drawable

Здесь используются атрибуты drawableEnd и drawableBottom





**Задание для самостоятельной работы:**

1. Создайте текстовое поле и кнопку по заданному макету. По нажатию на кнопку в текстовом поле выводится случайное число от 1 до 6

