**Создание телеграмм бота на Python**

**1. Регистрация**  
 Прежде чем начинать разработку, бота необходимо зарегистрировать и получить его уникальный id, являющийся одновременно и токеном. Для этого в Telegram существует специальный бот — [@BotFather](https://telegram.me/botfather).  
  
 Пишем ему /start и получаем список всех его команд.   
 Первая и главная — /newbot — отправляем ему и бот просит придумать имя нашему новому боту. Единственное ограничение на имя — оно должно оканчиваться на «bot». В случае успеха BotFather возвращает токен бота и ссылку для быстрого добавления бота в контакты, иначе придется поломать голову над именем.

Создайте файл config.py, сделайте там переменную token и поместите в неё токен.

**2.Установка**

Так как работать мы будет на языке программирования Python, требуется его установка.

Для работы с телеграммом требуется установить библиотеку pytelegrambotapi.

**3.Программирование**

Вначале требуется импортировать файл config для работы с токеном и telebot для работы с телеграммом.

Для привязки программы с ботом требуется написать следующую строку кода:

4 bot = telebot.TeleBot(config.TOKEN)

Теперь, следуя инструкциям телеграмма, требуется создать две команды: /start – для запуска бота и /help – для информации о других командах.

Для описания новой команды требуется написать строчку

6 @bot.message\_handler(commands=['start'])

Вместо ‘start’ можно писать и другой текст, это и будет текст вызываемой команды.

Теперь требуется описать дальнейший функционал команды, для этого мы создаём функцию, обязательным аргументом, которой должна быть переменная содержащая сообщение.

Для отправления сообщения человеку, нужно использовать метод send\_message, обязательные аргументы в котором id получателя и сообщение. Id человека мы можем получить командой message.from\_user.id, но в личном диалоге с ботом (не в групповом боте) легче использовать message.chat.id и получать id чата

9 bot.send\_message(message.chat.id, 'Hi!')

После создания определённого кол-ва функций требуются просто сообщения, на которые бот будет отвечать. Чтобы это осуществить, надо написать следующую строчку

11 @bot.message\_handler(content\_types=['text'])

После чего можно описывать разные функции. Но как их здесь есть два пути вызывать функцию обычным методом или использовать специальную возможность библиотеки telebot, при выборе второго варианта следует обратиться к методу register\_next\_step\_handler, который вызывается вот таким образом:

12 def first(message):

13 bot.register\_next\_step\_handler(message, second)

14 def second(message):

15 bot.send\_message(message.chat.id, 'Hi!')

И в последнюю очередь мне хотелось бы рассказать про клавиатуры. Они бывают двух видов: inline и reply.

Они обе описываются подобным образом, вначале мы пишем

keyboard\_1 = telebot.types.ReplyKeyboardMarkup()

Либо

keyboard\_2 = telebot.types.InlineKeyboardMarkup()

А далее создаем кнопки. Для этого нам потребуется два метода: Метод add принимает в себя любое количество кнопок, всегда начинает добавление с новой строки и переносит ряд при достижении значения установленной ширины. Метод row тоже принимает любое количество, но при этом не переносит кнопки на новый ряд, а добавляет всё полученное в одну строчку.

Чтобы отправить человеку клавиатуру, надо в отправке сообщения добавить аргумент reply\_markup значением которого будет являться наша клавиатура вот так: bot.send\_message(message.chat.id, 'Hi!', reply\_markup=keyboard\_1)

Больше информации вы можете найти здесь: <https://tlgrm.ru/docs/> .