

#TELEGRAMBOT

Created by:

Chingiz Bapanov
Bektur Toktogulov
Marseille Nazarmatov

«Создание виртуального окружения»

В терминале написать

1) `sudo apt-get install python3-pip` скачивание и установка pip

2) `virtualenv venv` – скачать окружение

3) `ls` - открыть список папок для определения местонахождения папки с файлами бота

4) `mkdir` "название папки" - Создать папку для бота (если папка есть-то пропустить этот раздел)

5) `cd` "название папки" войти в папку с ботом

6) `touch bot.py config.py` - создать файлы в формате пайтон для бота и его конфигурации (если эти файлы есть-то пропустить)

7) `source venv/bin/Scripts/activate` активация виртуального окружения

8) `pip install pytelegrambotapi` скачивание и установка нужной библиотеки для бота

9) `Python3 bot.py` ввести команду чтобы завести бота после его написания (bot.py можно назвать как угодно, главное установить в конце названия расширения .py)

`pip install beautifulsoup4`

`pip freeze > requirements.txt`

```
import telebot
import config
from telebot import types
```

```
bot = telebot.TeleBot(TOKEN)
```

```
@bot.message_handler(commands=['start'])
def welcome(message):
```

Создание кнопок

```
markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
button1 = types.KeyboardButton("Как дела?")
button2 = types.KeyboardButton("Что делаешь?")
```

```
markup.add(button1, button2)
bot.send_message(message.chat.id, 'Привет
{0.first_name}!).format(message.from_user), reply_markup=markup)
```

Диалог с ботом

```
@bot.message_handler(content_types=['text'])
def get_text(message):
    if message.chat.type == 'private':
        if message.text == 'Сообщение':
            markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
            button1 = types.KeyboardButton('Ответ 1')
            button2 = types.KeyboardButton('Ответ 2')
            back = types.KeyboardButton('Назад')
            markup.add(button1, button2, back)
            bot.send_message(message.chat.id, 'Выберите какое у вас настроение: ',
reply_markup=markup)
```

```
elif message.text == 'Назад':
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    button1 = types.KeyboardButton('Как дела?')
    button2 = types.KeyboardButton('Что делаешь?')
    markup.add(button1, button2)
    bot.send_message(message.chat.id, 'главное меню', reply_markup=markup)
```

Создание виртуального
окружения

Создание бота в Botfather

Импорт библиотек

Старт

bot.polling(non_stop=True)
смыкание бота

Сохранения файлов(фото,
видео, аудио и т.д.)

```
@bot.message_handler(commands=['start'])
def welcome(message):
    stickers = ['stiker.tgs', 'stiker1.tgs']
    stiker = choice(stickers)
```

```
with open(f'static/{stiker}', 'rb') as sti:
    bot.send_sticker(message.chat.id, sti)
```

<http://api.openweathermap.org/data/2.5/weather?q=%7Bcity%7D&appid=%7BAPI%7D&units=metric>

Ссылка (погода и т.д.)

<https://openweathermap.org/api>

```
mkdir weather
virtualenv venv
source venv/bin/activate
pip install pytelegrambotapi
```

```
API = "e8007ae6742631ec424ab8742301d9d3"
sity = data['sity']
cur_weather = data['main']['temp']
humidity = data['main']['humidity']
pressure = data['wind']['speed']
sunrise_timepesmp = datetime.fromtimestamp(data['sys']['sunrise'])
sunset_timepesmp = datetime.fromtimestamp(data['sys']['sunset'])
length_of_the_day = datetime.fromtimestamp(data['sys']['sunset']) -
datetime.datetime.fromtimestamp(
data['sys']['sunrise'])
```

```
code_to_smile = {
    'Clear': "Ясно \U00002600",
    'Clouds': "Облачно \U00002601",
    'Rain': "Дождь \U00002614",
    'Drizzle': "Дождь \U00002614",
    'Thunderstorm': "Гроза \U000026A1",
    'Snow': "Снег \U0001F328",
    'Mist': "Туман \U0001F32B"
}
```

«Создание виртуального окружения»

В терминале написать

- 1) `sudo apt-get install python3-pip` скачивание и установка
- 2) `virtualenv venv`
- 3) `ls` - открыть список папок для определения местонахождения папки с файлами бота
- 4) `mkdir "название папки"` - Создать папку для бота (*если папка есть-то пропустить этот раздел*)
- 5) `cd "название папки"` войти в папку с ботами
- 6) `touch bot.py config.py` - создать файлы в формате питон для бота и его конфигурации (*если эти файлы есть то пропустить*)
- 7) `source venv/bin(Scpirt)/activate` активация виртуального окружения
- 8) `pip install pytelegrambotapi` скачивание и установка нужной библиотеки для бота
- 9) `Python3 bot.py` ввести команду чтобы завести бот после его написания (`bot.py` можно назвать как угодно, главное установить в конце названия расширение `.py`)

1 В приложении телеграмм найти канал @BotFather

2 написать сообщение /newbot

BotFather is the one bot to create new bot accounts to existing bots. Information

@BotFather
Имя пользователя

otFather

Games

- /mygames - edit your games [beta]
- /newgame - create a new game
- /listgames - get a list of your games
- /editgame - edit a game
- /deletegame - delete an existing game

4:22

18 апреля

/newbot 6:27 ✓

Alright, a new bot. How are we going to call it? Please choose a name for your bot. 6:27

Good. Now let's choose a username for your bot. It must end in 'bot'. Like this, for example: TetrisBot or tetris_bot. 6:29

Janybay 6:29 ✓

JanybayBot 6:30 ✓

Done! Congratulations on your new bot. You will find it at t.me/JanybayBot. You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

Use this token to access the HTTP API:
53 [redacted] tji8

Keep your token secure and store it safely, it can be used by anyone to control your bot.

For a description of the Bot API, see this page:
<https://core.telegram.org/bots/api> 6:30

Меню Написать сообщение...

Bot

I can help you create and manage Telegram bots. If you're new to the Bot API, please [see the manual](#).

You can control me by sending these commands:

- /newbot - create a new bot
- /mybots - edit your bots [beta]

Edit Bots

- /setname - change a bot's name
- /setdescription - change bot description
- /setabouttext - change bot about info
- /setuserpic - change bot profile photo
- /setcommands - change the list of commands
- /deletebot - delete a bot

Bot Settings

- /token - generate authorization token
- /revoke - revoke bot access token
- /setinline - toggle inline mode
- /setinlinegeo - toggle inline location requests
- /setinlinefeedback - change inline feedback settings
- /setjoininggroups - can your bot be added to groups?
- /setprivacy - toggle privacy mode in groups

Games

- /mygames - edit your games [beta]
- /newgame - create a new game
- /listgames - get a list of your games
- /editgame - edit a game
- /deletegame - delete an existing game

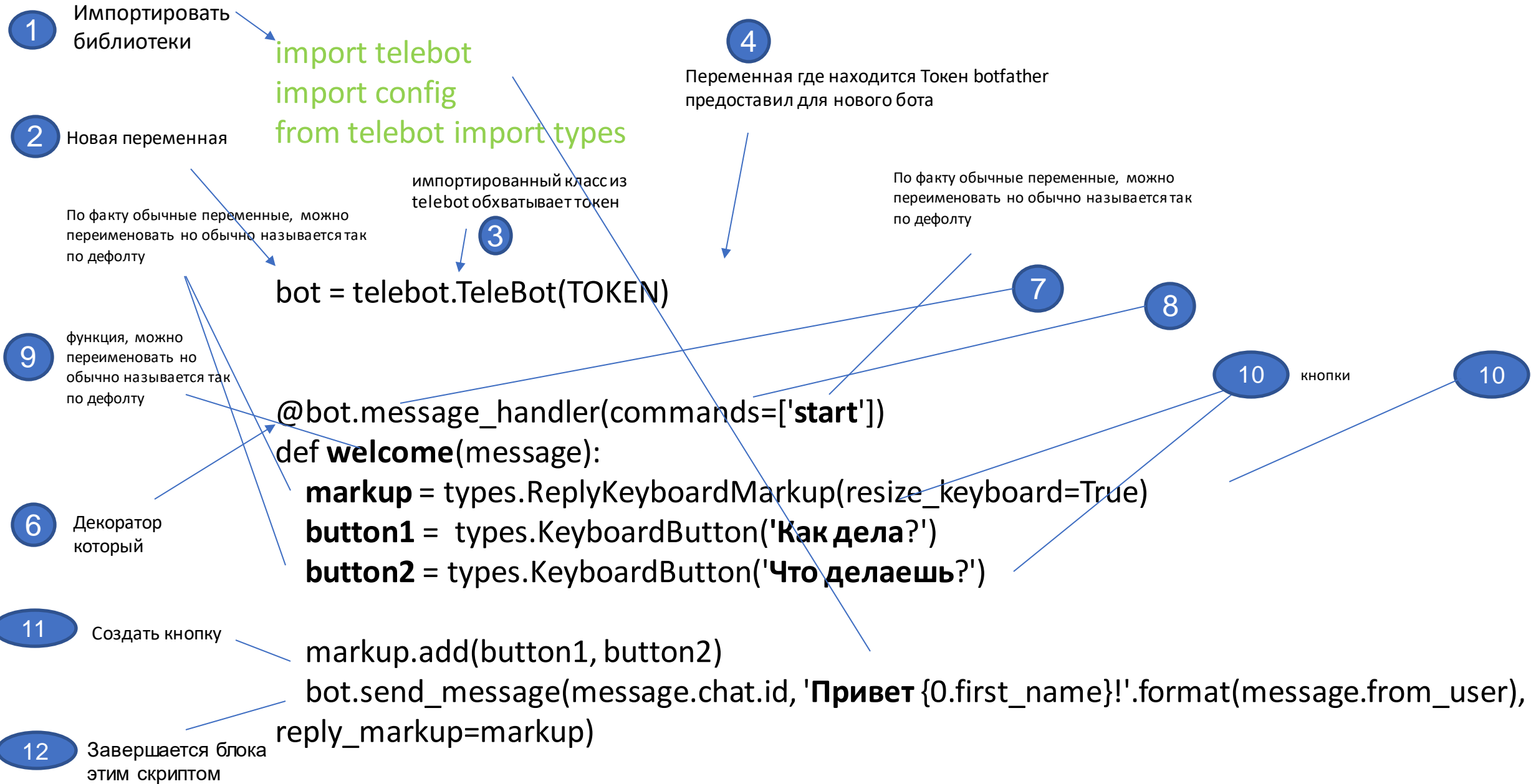
4:22

BotFather rule t new b mana, Инфо @Botf Имя г Уведс 9 ссы Пож: Ост

3 Дать имя боту написав в сообщении

4 Выбрать и поставить в конце имени вариант слова Bot

5 Скопировать Token и сохранить его в файле config.py таким образом
TOKEN = 'вставить токен'




Ответ на сообщение

```
@bot.message_handler(content_types=['text'])
def get_text(message):
    if message.chat.type == 'private':
        if message.text == 'Как дела?':
            markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
            button1 = types.KeyboardButton('Хорошо')
            button2 = types.KeyboardButton('Все плохо')
            back = types.KeyboardButton('Назад')
            markup.add(button1, button2, back)
            bot.send_message(message.chat.id, 'Выберите какое у вас настроение: ', reply_markup=markup)

        elif message.text == 'Назад':
            markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
            button1 = types.KeyboardButton('Как дела?')
            button2 = types.KeyboardButton('Что делаешь?')
            markup.add(button1, button2)
            bot.send_message(message.chat.id, 'главное меню', reply_markup=markup)
```

Что это?



Можно создать сколько
угодно кнопок

```

elif message.text == 'Что делаешь?':
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    button1 = types.KeyboardButton('Учу Python')
    button2 = types.KeyboardButton('Смотрю новости')
    back = types.KeyboardButton('Назад')
    markup.add(button1, button2, back)
    bot.send_message(message.chat.id, 'Выберите чем вы заняты: ', reply_markup=markup)

```

Переменная —
кнопки —
кнопки —
Регистрир. кнопки —

```

elif message.text == 'Хорошо':

```

```

    bot.send_message(message.chat.id, 'Ну круто че')

```

```

elif message.text == 'Все плохо':

```

```

    markup = types.InlineKeyboardMarkup(row_width=2)

```

```

    button1 = types.InlineKeyboardButton('Поесть мороженое', callback_data='ice_cream')

```

```

    button2 = types.InlineKeyboardButton('Пусть еще будет хуже', callback_data='sad')

```

```

    markup.add(button1, button2)

```

```

    bot.send_message(message.chat.id, 'Не растраивайся', reply_markup=markup)

```

В один ряд можно написать кнопок

Характеристики
(длина и
ширина)
кнопок ряды.
По стандарту 3

```
@bot.callback_query_handler(func=lambda call:True)
def callback_inline(call):
    if call.message:
        if call.data == 'ice_cream':
            markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
            button1 = types.KeyboardButton('Клубника с шоколадом')
            button2 = types.KeyboardButton('Шоколадное блуженство')
            markup.add(button1, button2)
            bot.send_message(call.message.chat.id, 'Выберите: ', reply_markup=markup)
        elif call.data == 'sad':
            markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
            button1 = types.KeyboardButton('Прыгнуть с крыши')
            button2 = types.KeyboardButton('Прыгнуть под поезд')
            markup.add(button1, button2)
            bot.send_message(call.message.chat.id, 'Выберите: ', reply_markup=markup)

bot.polling(non_stop=True)
```

Замыкание Бота




```
import telebot
from telebot import types
from config import TOKEN
```

```
bot = telebot.TeleBot(TOKEN)
```

```
@bot.message_handler(commands=['hello'])
```

```
def whats_up(message):
```

```
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
```

```
    button1 = types.KeyboardButton('Как жизнь молодая?')
```

```
    button2 = types.KeyboardButton('Чем занимаешься?')
```

```
    markup.add(button1, button2)
```

```
    bot.send_message(message.chat.id, 'Ну привет {0.first_name}!'.format(message.from_user), reply_markup=markup)
```

```
@bot.message_handler(content_types=['text'])
```

```
def get_text(message):
```

```
    if message.chat.type == 'private':
```

```
        if message.text == 'Как жизнь молодая?':
```

```
            markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
```

```
            button1 = types.KeyboardButton('Хорошо')
```

```
            button2 = types.KeyboardButton('Хорошо, но хочу еще лучше')
```

```
            button3 = types.KeyboardButton('Могло быть лучше')
```

```
            button4 = types.KeyboardButton('Подает')
```

```
            back = types.KeyboardButton('Вернуться')
```

```
            markup.add(button1, button2, button3, button4) # соединяет кнопки
```

```
            bot.send_message(message.chat.id, 'Проявите свое настроение нажав на кнопку', reply_markup=markup)
```

```
        elif message.text == 'Вернуться':
```

```
            markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
```

```
            button1 = types.KeyboardButton('Как жизнь молодая?')
```

```
            button2 = types.KeyboardButton('Чем занимаешься?')
```

```
            markup.add(button1, button2)
```

```
            elif message.text == 'Вернуться':
```

```
            bot.send_message(message.chat.id, 'Главное меню', reply_markup=markup)
```

```
        elif message.text == 'Чем занимаешься?':
```

```
            markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
```

```
            button1 = types.KeyboardButton('Учу Питон')
```

```
            button2 = types.KeyboardButton('Смотрю новости')
```

```
            button3 = types.KeyboardButton('Строю код')
```

```
            button4 = types.KeyboardButton('Тараю железо')
```

```
            back = types.KeyboardButton('Вернуться')
```

```
            markup.add(button1, button2, button3, button4, back)
```

```
            bot.send_message(message.chat.id, 'Чем занимаешься?', reply_markup=markup)
```

```
        elif message.text == 'Хорошо':
```

```
            bot.send_message(message.chat.id, 'Ну молодец!')
```

```
        elif message.text == 'Так себе':
```

```
            markup = types.ReplyKeyboardMarkup(row_width=2)
```

```
            button1 = types.KeyboardButton('Поешь мороженого', callback_data='ice_cream')
```

```
            button2 = types.KeyboardButton('Хочешь забыть о боли и вспомнить о дурой?', callback_data='sad')
```

```
            markup.add(button1, button2)
```

```
            bot.send_message(message.chat.id, 'Ладно, не расстраивайся', reply_markup=markup)
```

```
        elif message.text == "Учу Питон": # не было вопроса добавили скрипт вопрос и ответ
```

```
            bot.send_message(message.chat.id, "Учись хорошо!") # добавили ответ
```

```
        elif message.text == "Смотрю новости": # не было вопроса добавили скрипт вопрос и ответ
```

```
            bot.send_message(message.chat.id, "Читай книги!")
```

```
        elif message.text == "Строю код": # не было вопроса добавили скрипт вопрос и ответ
```

```
            bot.send_message(message.chat.id, "Строй внимательно!")
```

```
        elif message.text == "Тараю железо": # не было вопроса добавили скрипт вопрос и ответ
```

```
            bot.send_message(message.chat.id, "Тренируйся усердно!")
```

```
        elif message.text == 'Как жизнь молодая?': # не было вопроса добавили скрипт вопрос и ответ
```

```
            bot.send_message(message.chat.id, "Хорошо") # добавили ответ
```

```
        elif message.text == "Учу Питон": # не было вопроса добавили скрипт вопрос и ответ
```

- /setname — Изменить имя робота.
- /setdescription — Изменить описание робота, представляющее собой короткий текст с описанием бота. Пользователи увидят его в самом начале, под заголовком «Что умеет этот робот?».
- /setabouttext — Изменить информацию о боте, ещё более короткий текст, отображающийся в профиле бота. Ещё, если кто-то поделится вашим ботом, то вместе со ссылкой на него отправится этот текст.
- /setuserpic — Изменить аватарку бота. Картинки — всегда хорошо.
- /setcommands — Изменить список команд бота. Каждая команда состоит из собственно командного слова, начинающегося с символа кривой черты («/») и короткого описания. Пользователи увидят список команд при вводе символа «/».
- /setjoingroups — Определяет, можно ли добавлять вашего бота в группы.
- /setprivacy — Определяет, все ли сообщения видит ваш бот в группах. В выключенном состоянии роботу будут отправляться все сообщения.
- /deletebot — Удалить бота и его имя пользователя.

ЕЩЕ ОДИН БОТ

```
import telebot
from random import choice
from telebot import types
from datetime import datetime
```

```
TOKEN = ""
bot = telebot.TeleBot(TOKEN)
```

```
@bot.message_handler(commands=['start'])
def welcome(message):
    stickers = ['stiker.tgs', 'stiker1.tgs']
    stiker = choice(stickers)
```

```
    with open(f'static/{stiker}', 'rb') as sti:
        bot.send_sticker(message.chat.id, sti)
```

```
    text = ""Добро пожаловать {0.first_name}
Я Бот {1.first_name} созданный помогать людям"".format(
        message.from_user, bot.get_me())
```

```
    bot.send_message(message.chat.id, text)
```

```
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True)
    item1 = types.KeyboardButton('Как дела')
    item2 = types.KeyboardButton('Что делаешь')
    item3 = types.KeyboardButton('отправь стикер')
    item4 = types.KeyboardButton('отправь время')
```

```
    markup.add(item1, item2, item3, item4)
```

```
    bot.send_message(
        message.chat.id,
        "Выберите Кнопку",
        reply_markup=markup)
```

```
@bot.message_handler(content_types=['text'])
def text(message):
    if message.chat.type == 'private':
        if message.text == 'Как дела':
            bot.send_message(
                message.chat.id,
                'Отлично а у вас как?'
            )
        elif message.text == 'Что делаешь':
            bot.send_message(
                message.chat.id,
                'сичу думаю когда же вы меня улучшите'
            )
        elif message.text == 'отправь стикер':
            stickers = ['stiker.tgs', 'stiker1.tgs']
            stiker = choice(stickers)

            with open(f'static/{stiker}', 'rb') as sti:
                bot.send_sticker(message.chat.id, sti)
        elif message.text == 'отправь время':
            bot.send_message(
                message.chat.id,
                datetime.today().strftime('%H:%M:%S')
            )
        else:
            bot.send_message(
                message.chat.id,
                'Я вас не понимаю!!!!!!!!!!!!!!'
            )
```

Документация

<https://docs-python.ru/packages/biblioteka-python-telegram-bot-python/>