Курс написан специально для частной школы

математики и программирования

«Matrix»

(Матрица)

Урок № 25

По дисциплине:

«Основы python»

На тему:

«Работа с диалоговыми ботами. Создание клавиатуры в сообщениях.»

Автор:

Лаврентьев С.М.

Под редакцией:

Иванов А.А.

Владимир 2023

Оглавление

Введение

В этом уроке мы научимся обрабатывать несколько сообщений сразу и будем писать диалоговых ботов. Также разберем все виды кнопок в Telegram. Будет интересно, погнали!

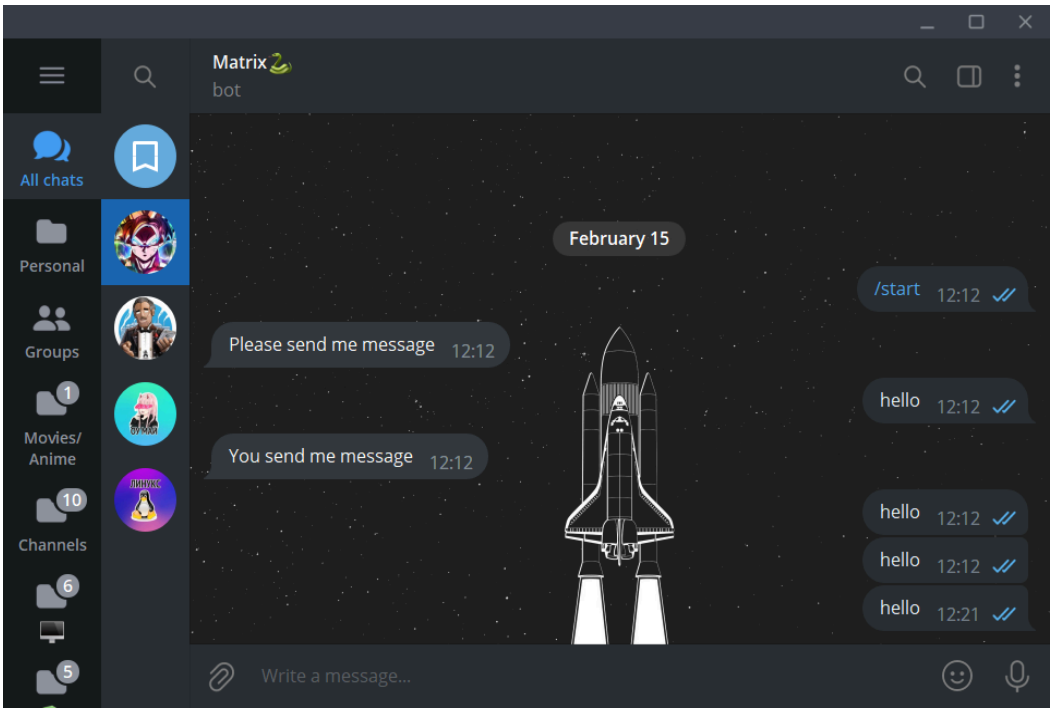
Последовательный ввод данных

Наверняка вы замечали, что ранее у нас не получалось делать последовательный ввод данных, как мы это делали при работе с консолью. Как раз таки с этим нам поможет метод **bot.register_next_step_handler(msg, function)**. Он принимает два обязательных аргумента. Первый - это **message**, а второй - это **function**. Работает таким образом: он ждёт сообщение пользователя и потом вызывает указанную функцию **function** с аргументом **message**. Пример ниже:

код:

```
import telebot

token = '5763354076:AAGTAKucMJbvJHPUfyhxm0XchZOYMYoJOds'
bot = telebot.TeleBot(token=token, parse_mode='Markdown')


@bot.message_handler(commands=['start'])
def welcome(message):
    mesg = bot.send_message(message.chat.id, 'Please send me message')
    bot.register_next_step_handler(mesg, test)


def test(message):
    bot.send_message(message.chat.id, 'You send me message')

bot.infinity_polling()
```

Результат:



Пример работы метода bot.register_next_step_handler()

Сейчас мы сделаем бота, который будет получать от пользователя два числа, а потом выводить результат их сложения:

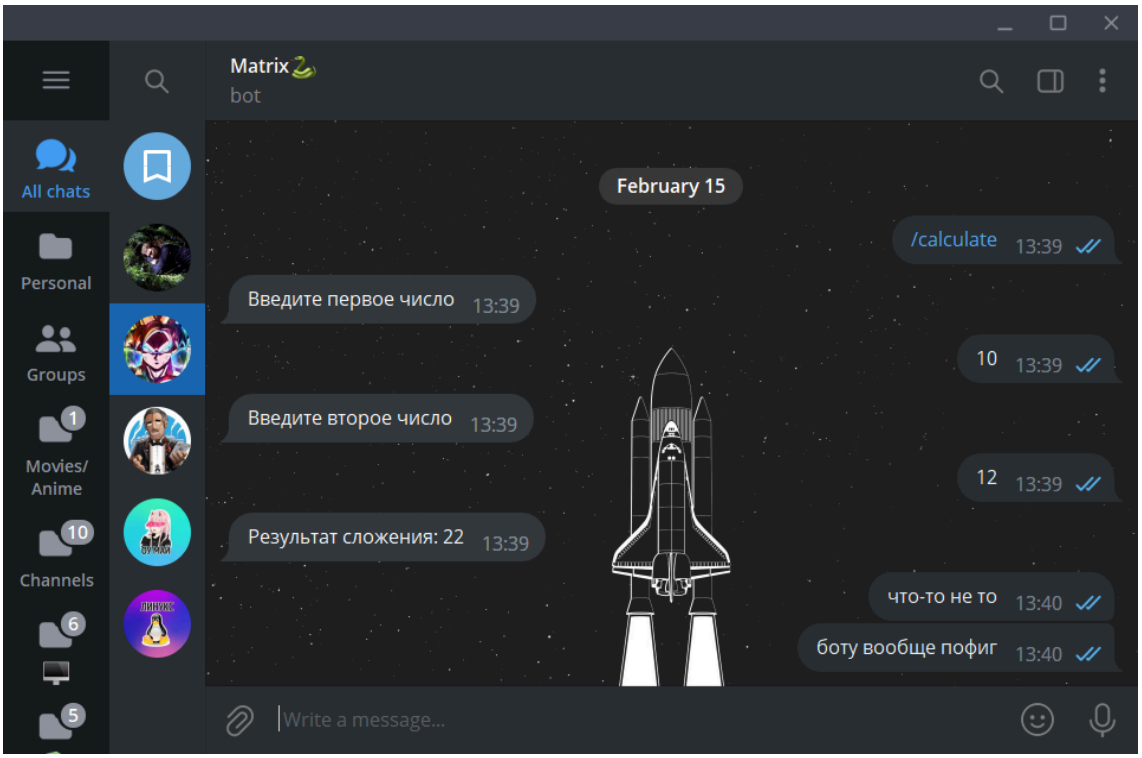
Код:

```
from pprint import pprint

import telebot

token = '5763354076:AAGTAKucMJbvJHPUfyhxm0XchZOYMYoJOds'
bot = telebot.TeleBot(token=token, parse_mode='Markdown')

history_msgs = {}


@bot.message_handler(commands=['calculate'])
def calculate(msg):
    msg = bot.send_message(msg.chat.id, 'Введите первое число')
    bot.register_next_step_handler(msg, step_1)


def step_1(msg):
    if not (msg.chat.id in history_msgs):
        history_msgs.update({msg.chat.id: []})
    history_msgs[msg.chat.id].append(msg)
    msg = bot.send_message(msg.chat.id, 'Введите второе число')
    bot.register_next_step_handler(msg, step_2)


def step_2(msg):
    history_msgs[msg.chat.id].append(msg)
    num_1 = int(history_msgs[msg.chat.id][-1].text)
    num_2 = int(history_msgs[msg.chat.id][-2].text)
    bot.send_message(chat_id=msg.chat.id, text=f'Результат сложения:
{num_1 + num_2}')
    pprint(history_msgs)
    history_msgs[msg.chat.id].clear()
    pprint(history_msgs)


bot.infinity_polling()
```

Результат(консоль):

```
PS D:\Programming\Python\PyCharm\TEST> python main.py
{963081479: [<telebot.types.Message object at 0x00000170F2D8A730>,
             <telebot.types.Message object at 0x00000170F2D8A370>]}
{963081479: []}
```

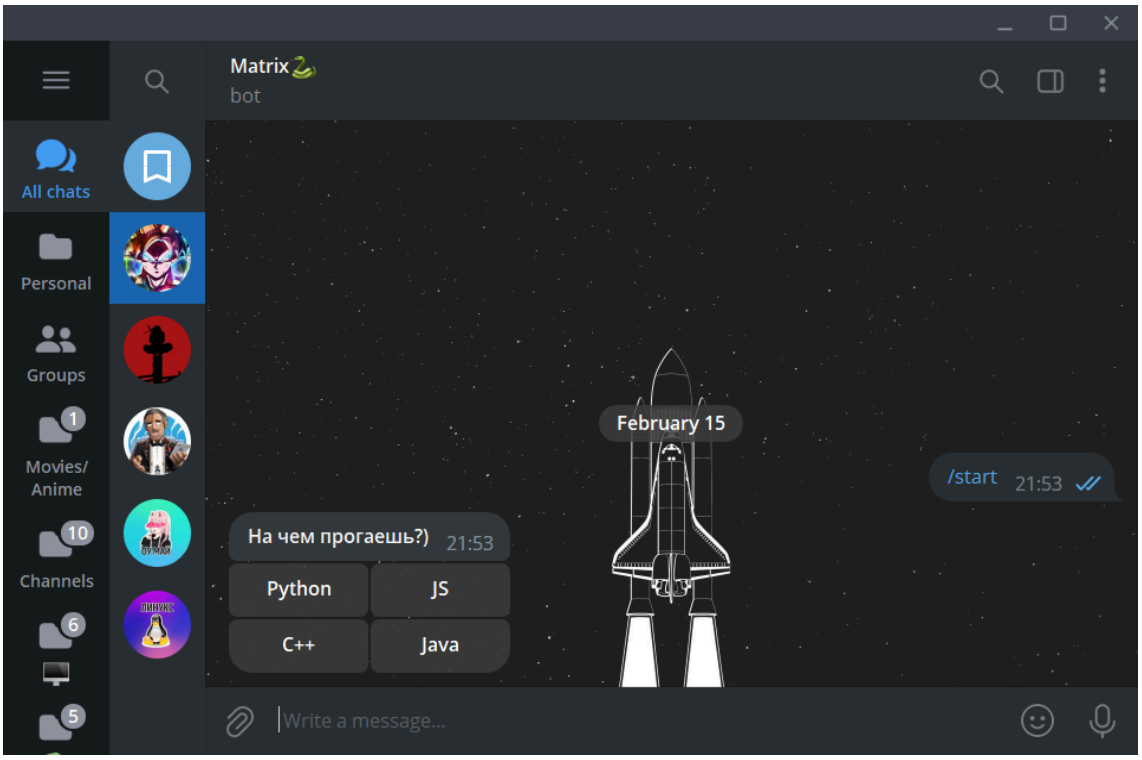
Результат(Telegram):



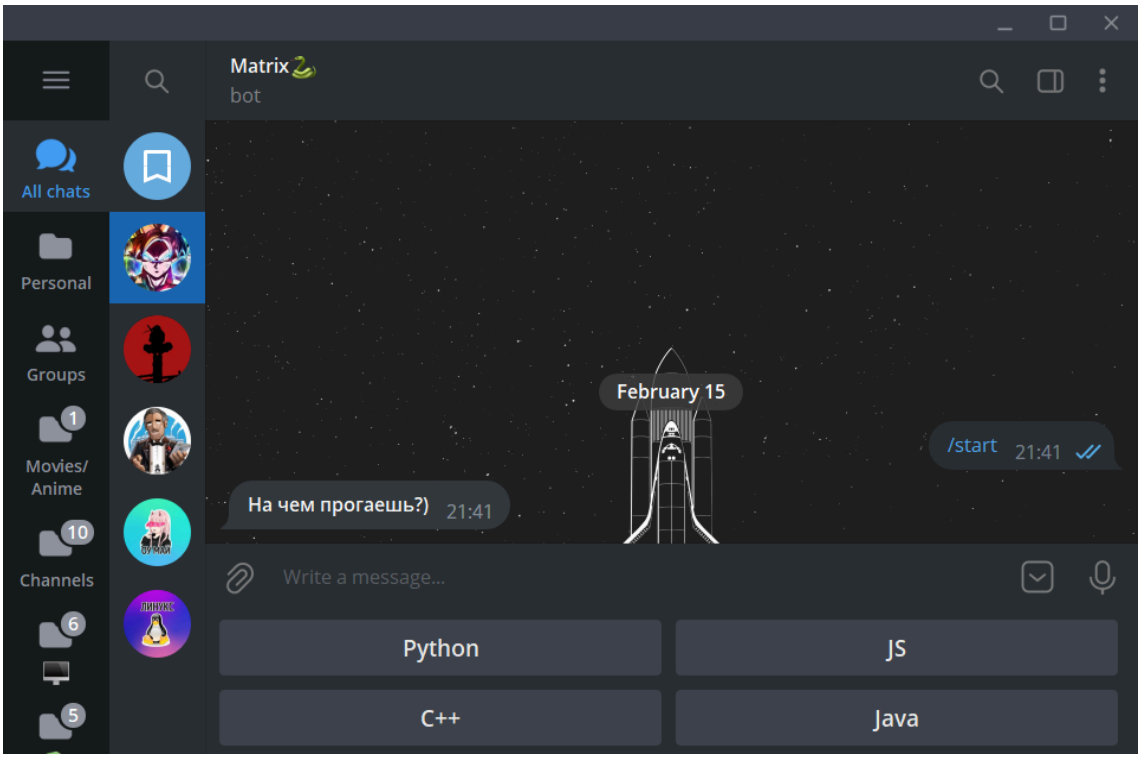
Виды кнопок и их отличие в телеграмм

Всего в Telegram есть два вида кнопок: inline и reply.

inline:



reply:



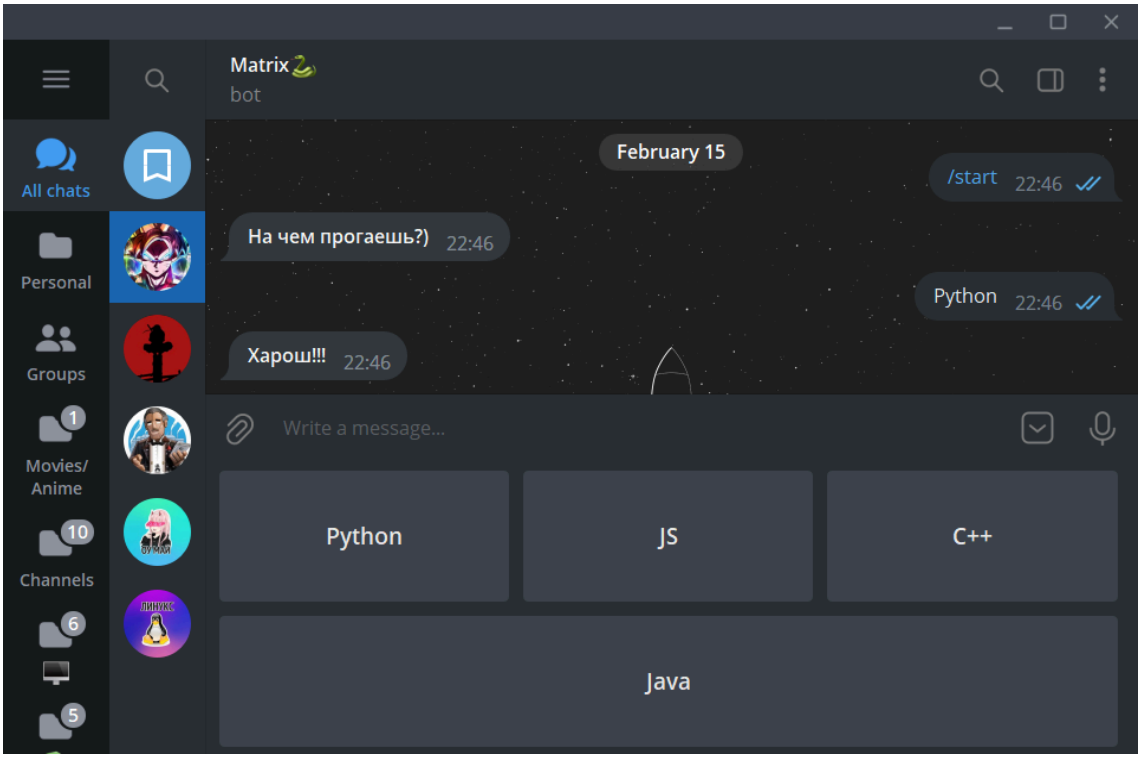
reply-кнопки

Сейчас мы рассмотрим создание reply-кнопок. Тут все совсем просто. Прежде всего нам нужно импортировать types из telebot. Сначала нам нужно создать клавиатуру для кнопок класса **ReplyKeyboardMarkup**. Потом создать кнопки класса **KeyboardButton(text)**, где **text** - это текст кнопки. И добавить их в клавиатуру при помощи метода **add()**. Затем нам нужно прикрепить эту клавиатуру к какому-либо сообщению, чтобы она появилась только после отправки этого сообщения. Рассмотрим на примере ниже:

Код:

```
import telebot
from telebot import types

token = '5763354076:AAGTAKucMJbvJHPUfyhxm0XchZOYMYoJOds'
bot = telebot.TeleBot(token=token, parse_mode='Markdown')


@bot.message_handler(commands=['start'])
def start(msg):
    markup = types.ReplyKeyboardMarkup()

    btn1 = types.KeyboardButton("Python")
    btn2 = types.KeyboardButton("JS")
    btn3 = types.KeyboardButton("C++")
    btn4 = types.KeyboardButton("Java")

    markup.add(btn1, btn2, btn3, btn4)

    bot.send_message(chat_id=msg.chat.id,
                     text='*На чем прогаешь?)*',
                     reply_markup=markup)


@bot.message_handler(content_types=['text'])
def python(msg):
    if msg.text == 'Python':
        bot.send_message(chat_id=msg.chat.id, text='*Харош!!!*')


bot.infinity_polling()
```

Результат:



Хочу обратить ваше внимание на то, что кнопки стали больше. На мой взгляд это не эстетично. Давайте рассмотрим все поля класса **`ReplyKeyboardMarkup`**.

`resize_keyboard` (`bool`) – этот параметр отвечает за способность кнопок подгоняться по вертикали, чтобы они занимали меньше места. По умолчанию стоит `False`.

`one_time_keyboard` (`bool`) – после нажатия на кнопку клавиатура скрывается, но по-прежнему доступна пользователю после нажатия специальной кнопки в поле ввода. По умолчанию стоит `False`.

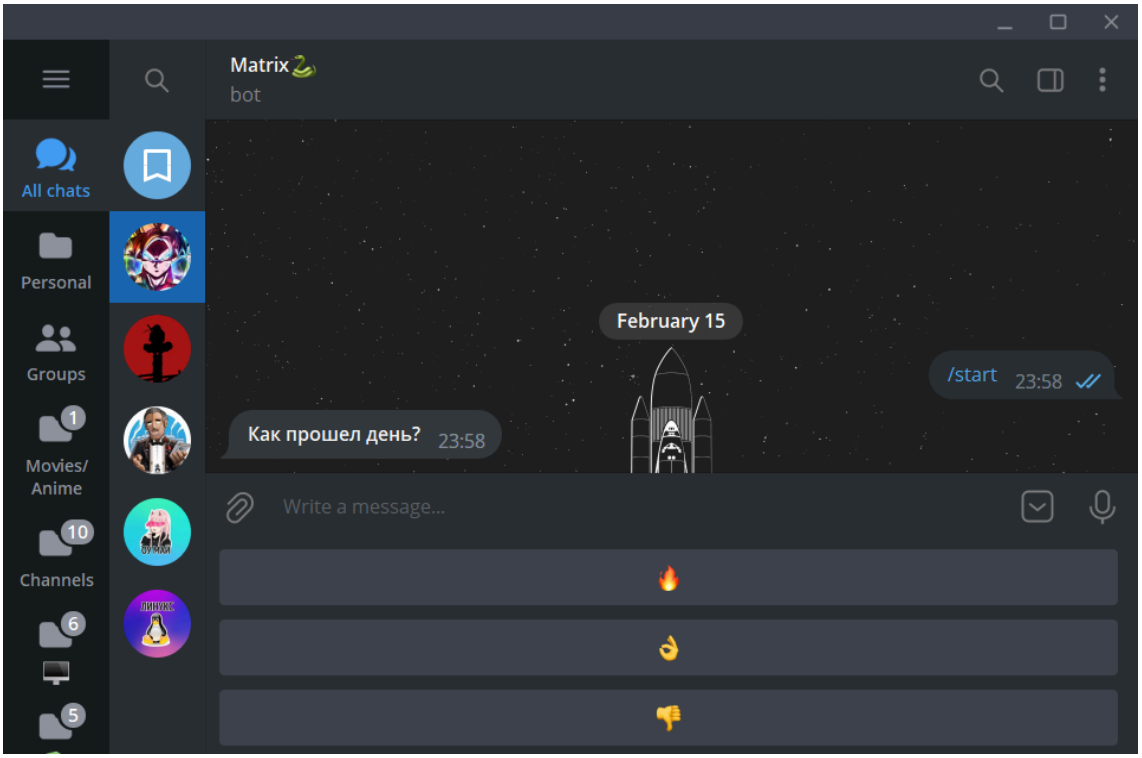
`row_width` (`int`) - этот параметр устанавливает, какое количество кнопок должно располагаться в одном ряду. По умолчанию стоит `3`.

Пример использования reply-кнопок

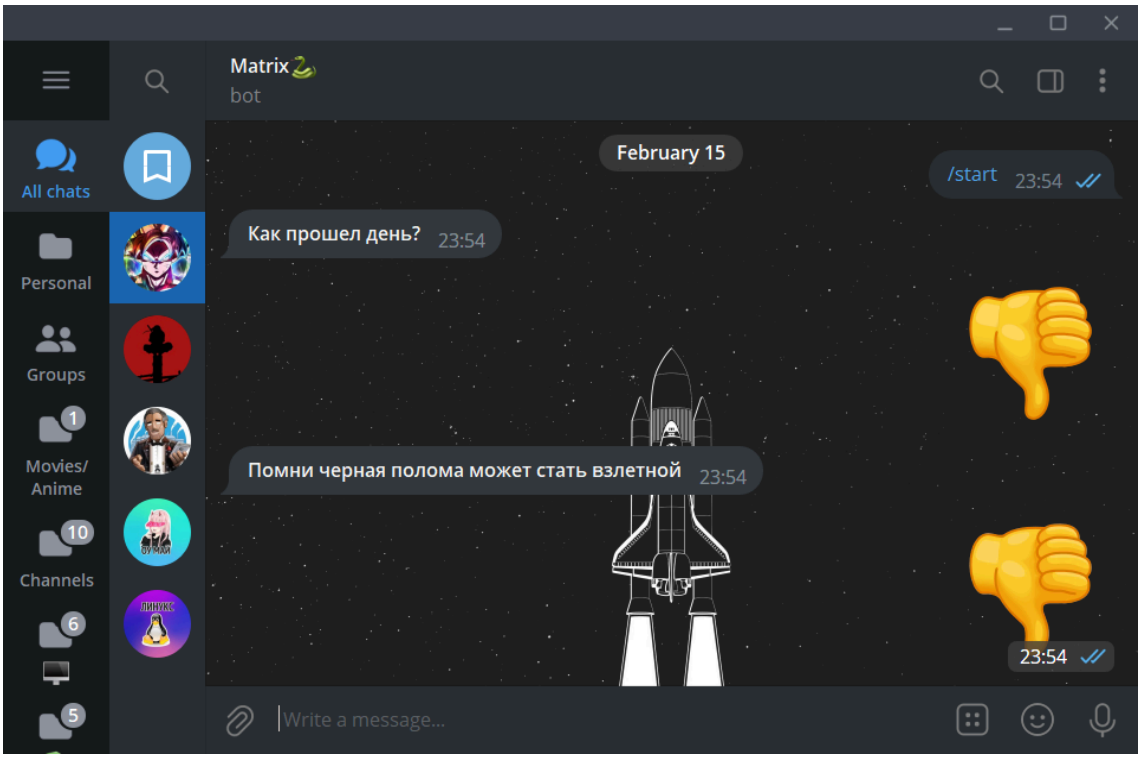
Код:

```
import telebot
from telebot import types

token = '5763354076:AAGTAKucMJbvJHPUfyhxm0XchZOYMYoJOds'
bot = telebot.TeleBot(token=token, parse_mode='Markdown')


@bot.message_handler(commands=['start'])
def start(msg):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True, row_width=1)
    btn1 = types.KeyboardButton("🔥")
    btn2 = types.KeyboardButton("👍")
    btn3 = types.KeyboardButton("👎")
    markup.add(btn1, btn2, btn3)
    bot.send_message(chat_id=msg.chat.id,
                     text='*Как прошел день?*',
                     reply_markup=markup)
    bot.register_next_step_handler(msg, answer)


def answer(msg):
    if msg.text == '🔥':
        bot.send_message(chat_id=msg.chat.id, text='*Ну и хорошо,
умывайся и в тепленькую постельку)*')
    elif msg.text == '👍':
        bot.send_message(chat_id=msg.chat.id, text='*Ни о чем не
беспокойся и иди дремать, завтра будет лучше)*')
    elif msg.text == '👎':
        bot.send_message(chat_id=msg.chat.id, text='*Помни, черная
полоса может стать взлетной*')


bot.infinity_polling()
```

Результат (1):



Результат (2):



Как мы видим, на первом скриншоте кнопки занимают мало места, и они располагаются по одной в каждой строке. А на втором скриншоте мы видим, что после нажатия на кнопку клавиатура скрылась, но по-прежнему доступна, то есть она не удалилась полностью. Ну и плюсом, используя метод **`register_next_step_handler`**, мы смогли сделать так, чтобы бот отвечал только на следующее сообщение после вопроса, а не на все.

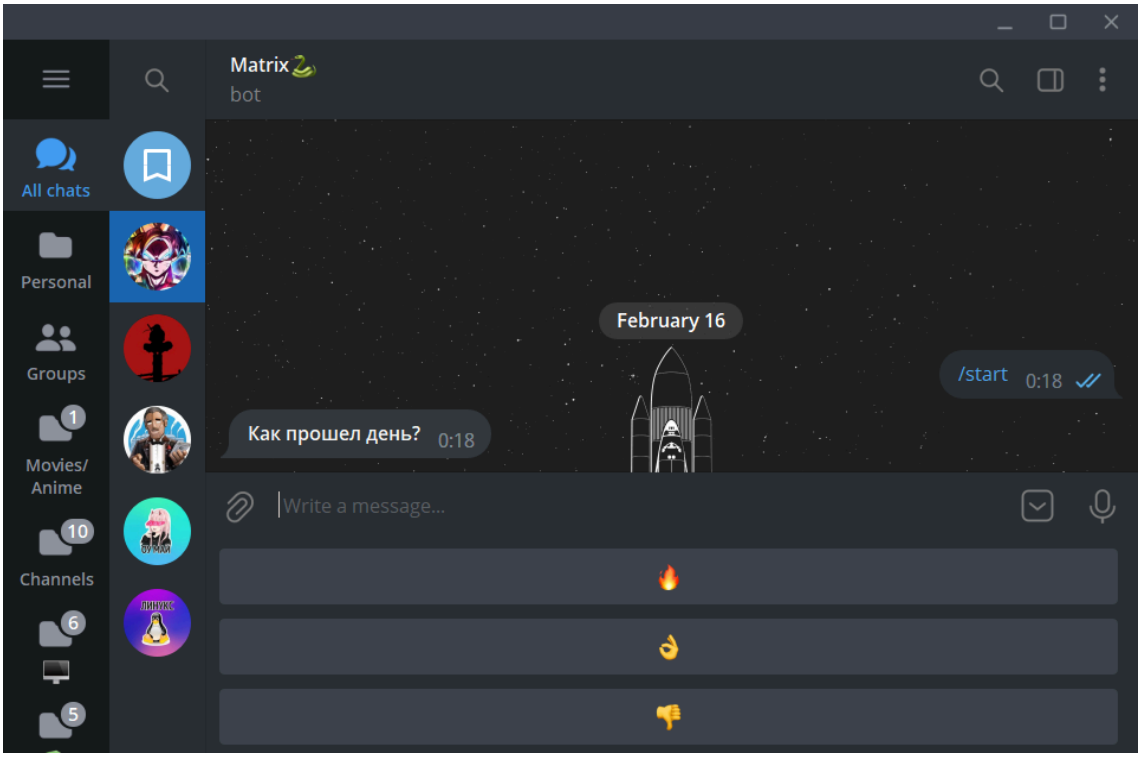
Полное удаление reply-клавиатуры

Для этого нам нужно прикрепить к сообщению (после которого мы хотим, чтобы наша reply-клавиатура пропала) клавиатуру класса **ReplyKeyboardRemove().** Посмотрев на пример ниже, вы сразу все поймете:

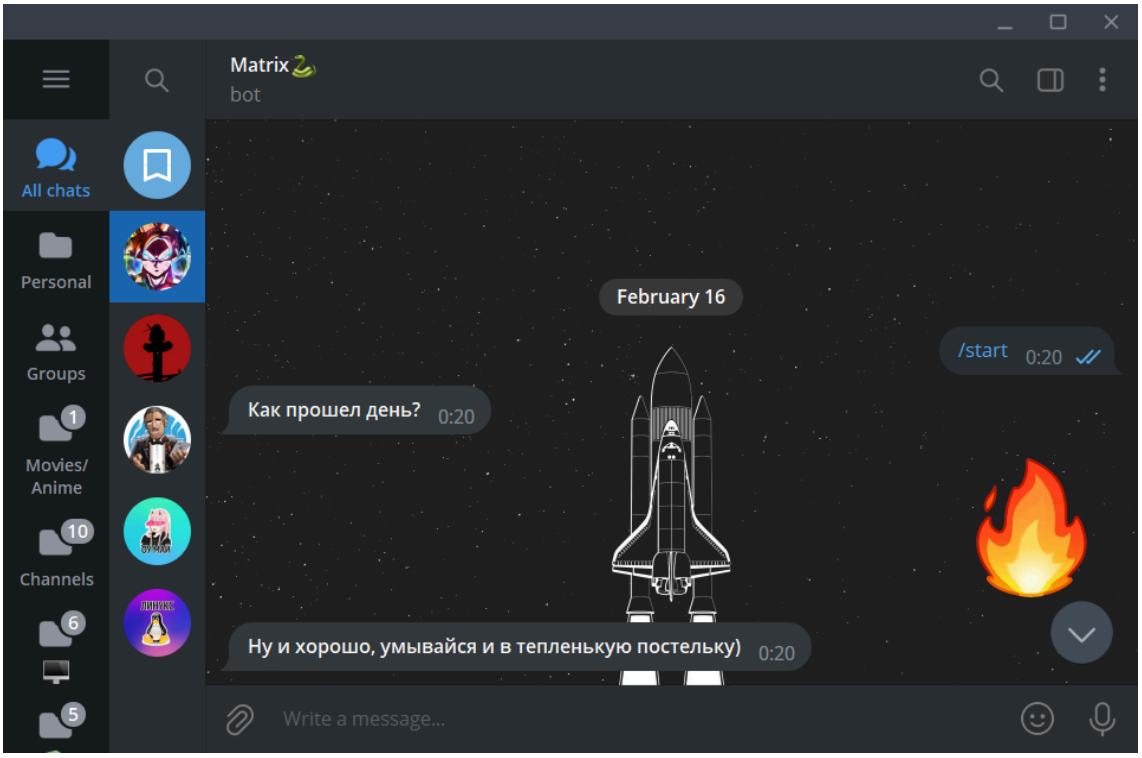
Код:

```
import telebot
from telebot import types

token = '5763354076:AAGTAKucMJbvJHPUfyhxm0XchZOYMYoJOds'
bot = telebot.TeleBot(token=token, parse_mode='Markdown')


@bot.message_handler(commands=['start'])
def start(msg):
    markup = types.ReplyKeyboardMarkup(resize_keyboard=True,
one_time_keyboard=True, row_width=1)
    btn1 = types.KeyboardButton("🔥")
    btn2 = types.KeyboardButton("👍")
    btn3 = types.KeyboardButton("👎")
    markup.add(btn1, btn2, btn3)
    bot.send_message(chat_id=msg.chat.id,
                     text='*Как прошел день?*',
                     reply_markup=markup)
    bot.register_next_step_handler(msg, answer)


def answer(msg):
    if msg.text == '🔥':
        bot.send_message(chat_id=msg.chat.id, text='*Ну и хорошо,
умывайся и в тепленькую постельку)*',
reply_markup=types.ReplyKeyboardRemove())
    elif msg.text == '👍':
        bot.send_message(chat_id=msg.chat.id, text='*Ни о чем не
беспокойся и иди дремать, завтра будет лучше)*',
reply_markup=types.ReplyKeyboardRemove())
    elif msg.text == '👎':
        bot.send_message(chat_id=msg.chat.id, text='*Помни, черная
полоса может стать взлетной*', reply_markup=types.ReplyKeyboardRemove())


bot.infinity_polling()
```

Результат (1):



Результат (2):



Как мы видим, теперь клавиатура пропала окончательно.

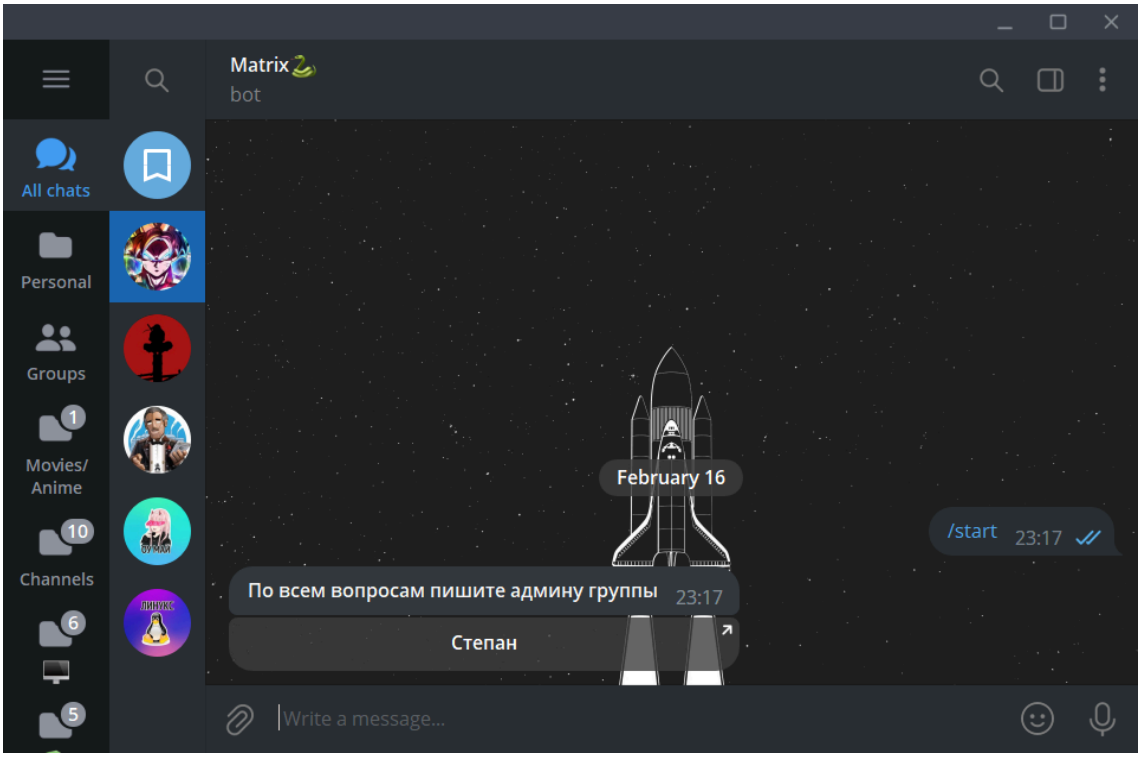
inline-кнопки со ссылкой

Тут все так же просто, как и с reply-кнопками. Сначала нам нужно создать клавиатуру для кнопок класса **InlineKeyboardMarkup**. Потом создать кнопки класса **InlineKeyboardButton(text, url)**, где **text** - это текст кнопки, а **url** – это ссылка, по которой пользователь будет переходить при нажатии. И добавить их в клавиатуру при помощи метода **add()**. Затем нам нужно прикрепить эту клавиатуру к какому-нибудь сообщению. Рассмотрим на примере ниже:

Код:

```
import telebot
from telebot import types

token = '5763354076:AAGTAKucMJbvJHPUfyhxm0XchZOYMYoJOds'
bot = telebot.TeleBot(token=token, parse_mode='Markdown')


@bot.message_handler(commands=['start'])
def start(msg):
    markup = types.InlineKeyboardMarkup()
    btn1 = types.InlineKeyboardButton(text='Степан',
url='https://t.me/NeuroNeron')
    markup.add(btn1)
    bot.send_message(chat_id=msg.chat.id,
                     text='*По всем вопросам пишите админу группы*',
                     reply_markup=markup)


bot.infinity_polling()
```

Результат:



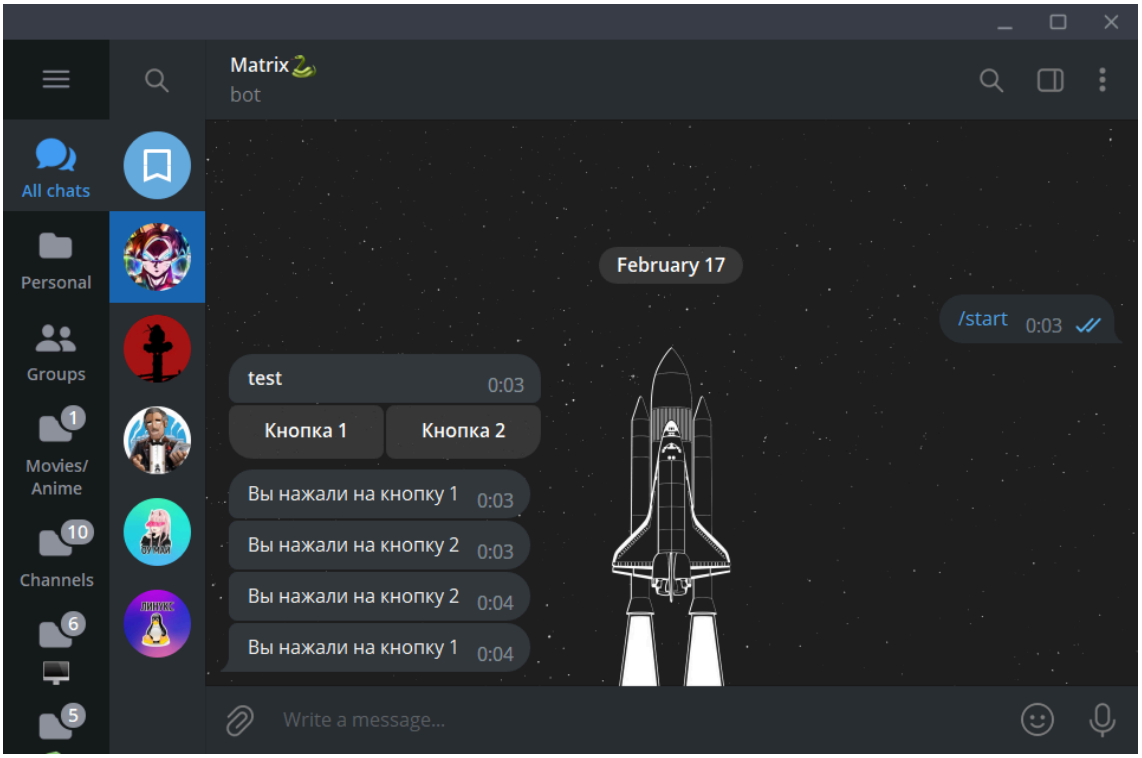
Иной метод работы с inline-кнопками

На одних ссылках далеко не уедешь. Мы можем сделать обработчик кнопок, который даст нам знать, когда пользователь нажал на кнопку, и на какую именно он нажал. Проанализируйте код, приведённый ниже, и попробуйте сделать какие-то выводы (анализ – ключ к высшему):

Код:

```
import telebot
from telebot import types

token = '5763354076:AAGTAKucMJbvJHPUfyhxm0XchZOYMYoJOds'
bot = telebot.TeleBot(token=token, parse_mode='Markdown')


@bot.message_handler(commands=['start'])
def start(msg):
    markup = types.InlineKeyboardMarkup()
    btn1 = types.InlineKeyboardButton(text='Кнопка 1',
callback_data='btn1_data')
    btn2 = types.InlineKeyboardButton(text='Кнопка 2',
callback_data='btn2_data')
    markup.add(btn1, btn2)
    bot.send_message(chat_id=msg.chat.id,
                     text='*test*',
                     reply_markup=markup)


@bot.callback_query_handler(func=lambda call: call.data == 'btn1_data')
def but1_pressed(call: types.CallbackQuery):
    bot.send_message(chat_id=call.message.chat.id, text='Вы нажали на
кнопку 1')


@bot.callback_query_handler(func=lambda call: call.data == 'btn2_data')
def but2_pressed(call: types.CallbackQuery):
    bot.send_message(chat_id=call.message.chat.id, text='Вы нажали на
кнопку 2')


bot.infinity_polling()
```

Результат:



Как мы видим, для того чтобы мы могли обрабатывать нажатия на кнопки, мы должны указать не **url**, а **callback_data** и в обработчике определять, какая кнопка была нажата именно по этим данным. Для обработчика используйте следующую конструкцию:

```
@bot.callback_query_handler(func=lambda call: call.data ==
‘callback_data‘)
def but1_pressed(call: types.CallbackQuery):
    ...
```

Информация хранящаяся в CallbackQuery

При каждом нажатии на кнопку Telegram отправляет нам обратно запрос, что была нажата такая-то кнопка, в таком-то чате и для того, чтобы узнать эту информацию мы должны воспользоваться полем **json** нашего запроса **call.** Это поле возвращает нам словарь (как вы понимаете и работать с ним надо, как со словарем) со всеми необходимыми данными. Давайте рассмотрим на примере ниже:

Код:

```
from pprint import pprint

import telebot
from telebot import types

token = '5763354076:AAGTAKucMJbvJHPUfyhxm0XchZOYMYoJOds'
bot = telebot.TeleBot(token=token, parse_mode='Markdown')


@bot.message_handler(commands=['start'])
def start(msg):
    markup = types.InlineKeyboardMarkup()
    btn1 = types.InlineKeyboardButton(text='Кнопка 1',
callback_data='btn1_data')
    markup.add(btn1)
    bot.send_message(chat_id=msg.chat.id,
                     text='*test*',
                     reply_markup=markup)


@bot.callback_query_handler(func=lambda call: call.data == 'btn1_data')
def but1_pressed(call: types.CallbackQuery):
    pprint(call.json)


bot.infinity_polling()
```

Результат(консоль):

```
{'chat_instance': '-3739627214141287415',
 'data': 'btn1_data',
 'from': {'first_name': 'Stepan',
          'id': 1234567890,
          'is_bot': False,
          'language_code': 'ru',
          'last_name': 'Lavrentiev',
          'username': 'Potato1407'},
 'id': '4136403457269145319',
 'message': {'chat': {'id': -1001758289917,
                      'title': 'Test Matrix group',
                      'type': 'supergroup'},
             'date': 1676629732,
             'entities': [{'length': 4, 'offset': 0, 'type': 'bold'}],
             'from': {'first_name': 'Matrix🐍',
                      'id': 5763354076,
                      'is_bot': True,
                      'username': 'matrix_the_best_bot'},
             'message_id': 229,
             'reply_markup': {'inline_keyboard': [[{'callback_data':
'btn1_data',
                                                    'text': 'Кнопка
1'}]]},
             'text': 'test'}}
```

Результат(Telegram):

Информации здесь просто океан и пару морей. Эти данные нам потом очень сильно помогут, поэтому наматывайте ус.

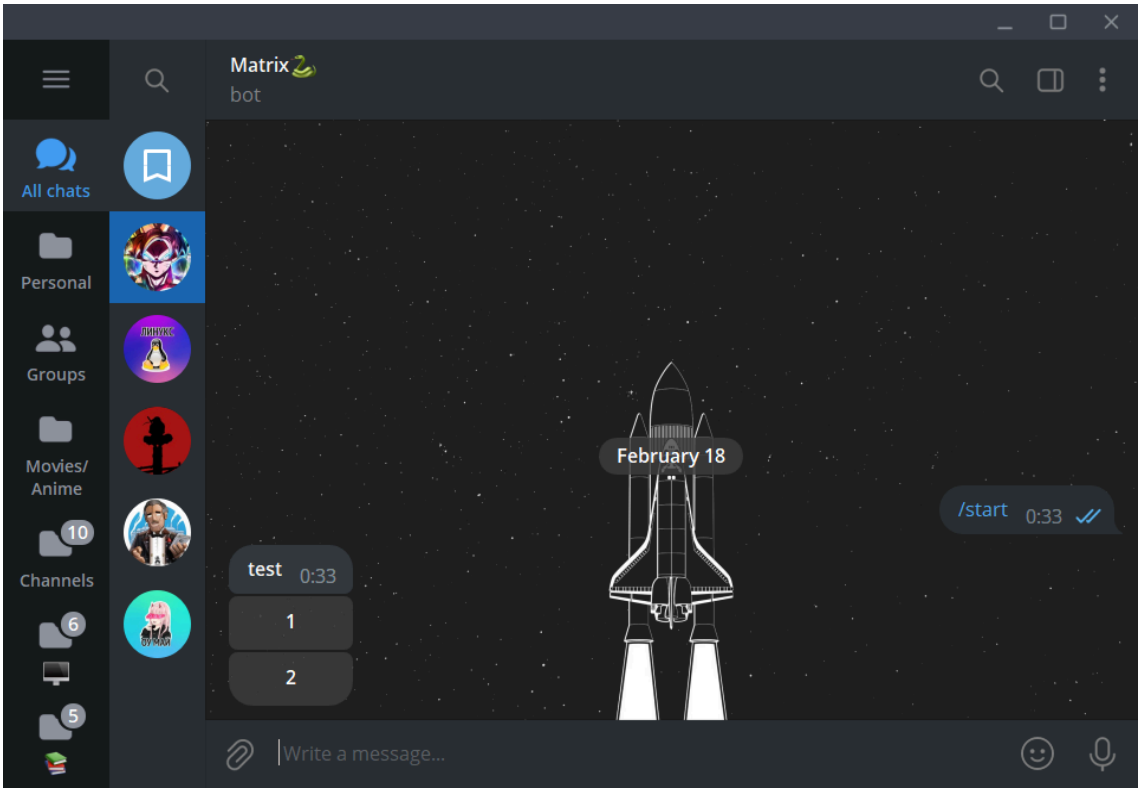
Изменение текста на кнопке. Метод bot.edit_message_text()

Для того чтобы бот мог изменять сообщения написанные самим ботом, нам надо использовать метод **bot.edit_message_text(text, chat_id, message_id, reply_markup)**, где **text** - это текст сообщения, **chat_id** - id чата, **message_id** - id сообщения, **reply_markup** - клавиатура, которую мы хотим прикрепить сообщению. Пример ниже:

Код:

```
import telebot
from telebot import types

token = '5763354076:AAGTAKucMJbvJHPUfyhxm0XchZOYMYoJOds'
bot = telebot.TeleBot(token=token, parse_mode='Markdown')


@bot.message_handler(commands=['start'])
def start(msg):
    markup = types.InlineKeyboardMarkup(row_width=1)
    btn1 = types.InlineKeyboardButton(text='1', callback_data='btn1_data')
    btn2 = types.InlineKeyboardButton(text='2', callback_data='btn2_data')
    markup.add(btn1, btn2)
    bot.send_message(chat_id=msg.chat.id,
                     text='*test*',
                     reply_markup=markup)


@bot.callback_query_handler(func=lambda call: call.data == 'btn1_data')
def but1_pressed(call: types.CallbackQuery):
    markup = types.InlineKeyboardMarkup(row_width=1)
    for i in call.json['message']['reply_markup']['inline_keyboard']:
        for j in i:
            if j['callback_data'] == 'btn1_data':

markup.add(types.InlineKeyboardButton(text=f'{int(j["text"]) + 1} ',
callback_data=j['callback_data']))
            else:
                markup.add(types.InlineKeyboardButton(text=j['text'],
callback_data=j['callback_data']))
    bot.edit_message_text(text=call.json['message']['text'],
                          chat_id=call.json['message']['chat']['id'],
                          message_id=call.json['message']['message_id'],
                          reply_markup=markup)


bot.infinity_polling()
```

Результат (до):



Результат (после шести нажатий на кнопку “1”):