



Курс 3.0 «Чатбот со Стоуном»

Лекция 1. **ВВЕДЕНИЕ**

Курс лекций от Stone

2023

Давайте познакомимся!



Автор, лектор и еще много кто этого курса Стоун Семнадцатый, он же Кирилл Панфилов

- финалист Comedy Баттл
- неоднократный победитель шоу НЕ СПАТЬ
- отец двоих детей
- панк и рокнрольщик :)

А также семинарист курсов «Введение в Python» и «Погружение в Python»

Так что у Стоуна можно не только ботам поучиться – загляните на его Ютуб и ТГ каналы

https://www.youtube.com/@dirty_python

https://t.me/dirty_python

<https://t.me/+knXQRpqb05o0ODUy>

https://t.me/STONE_Py



Чем сегодня займемся?

- Разберемся, кто такие боты и с чем их едят.
- Заглянем немного под капот, чтобы разобраться с работой чатбота.
- Попросим у BotFather нового бота.
- Создадим первого простенького бота
- Поговорим о курсе

ЧТО такое бот и с чем его едят?

Робот, или **бот**, а также **интернет-бот** и тому подобное — специальная **программа**, выполняющая автоматически и/или по заданному расписанию какие-либо действия через интерфейсы, предназначенные для людей.

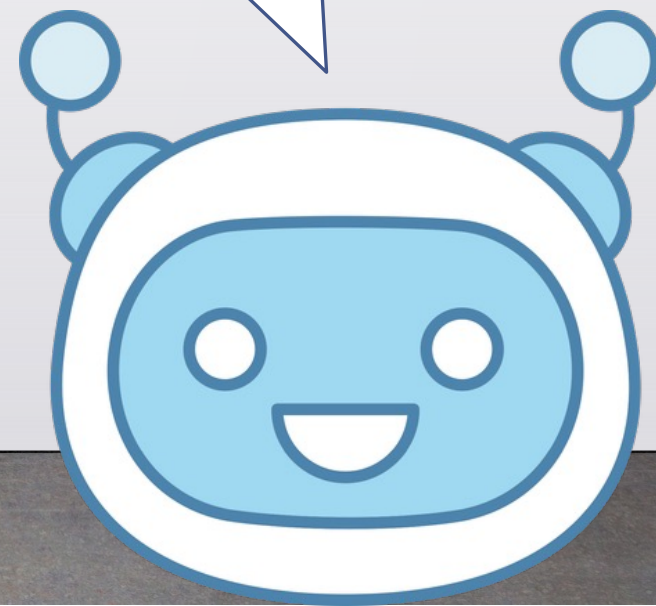
Ботов можно встретить:

Сайты.

Приложения.

И, конечно, мессенджеры!

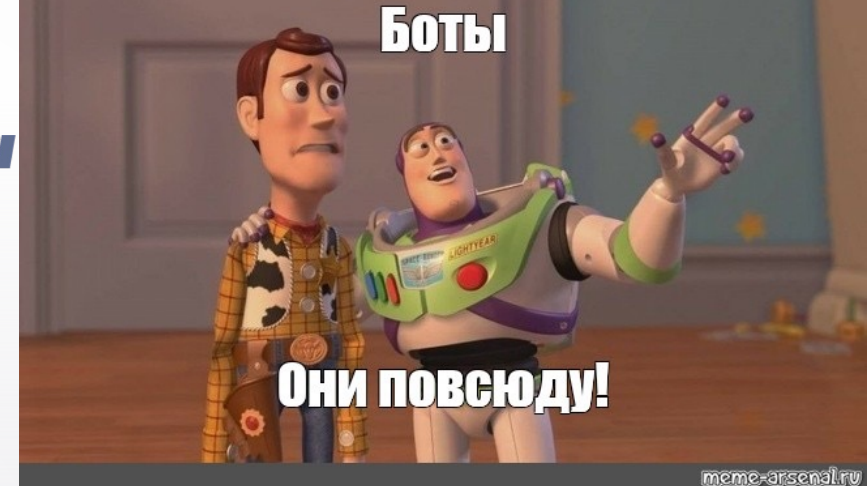
Я полезный!





И таки для чего он нужен?

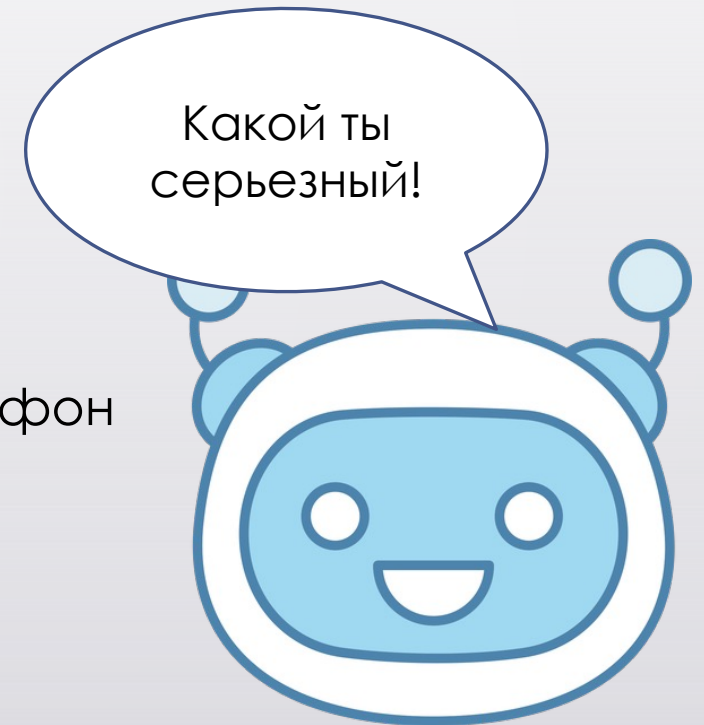
- Автоматизация некоторых процессов;
- Чатбот – разновидность бота в мессенджере, где он тоже может заменить человека, отвечая, например, на стандартные вопросы;
- Доступ к базам данных через тоже мессенджеры также очень удобно реализовать через бота
- Для малого и среднего бизнеса бот может заменить секретаря или диспетчера – покажет пользователю ассортимент и наличие товара или услуги, сформирует заказ, соберет данные, организует доставку и многое другое.
- Да и внутри компании бот пригодится как система оповещения, контроля и автоматизации некоторых процессов



Пфф... Мобильное приложение круче!

- А так ли?
- Никто не спорит, иметь свое приложение очень круто. Вот только не всегда надо. Иногда достаточно чатбота. Ведь он:

- дешевле
- проще в написании и поддержании
- не требует регистрации и скачивания на маркетах и специальной установки на телефон
- вполне способен справиться с простыми задачами



Ну и как же работает этот ваш бот?

- А сейчас все покажу! Разбираться будем на примере Telegram- бота.
- Как и у всякой программы, рассчитанной на непосредственное общение с пользователем у нее есть Frontend и Backend.

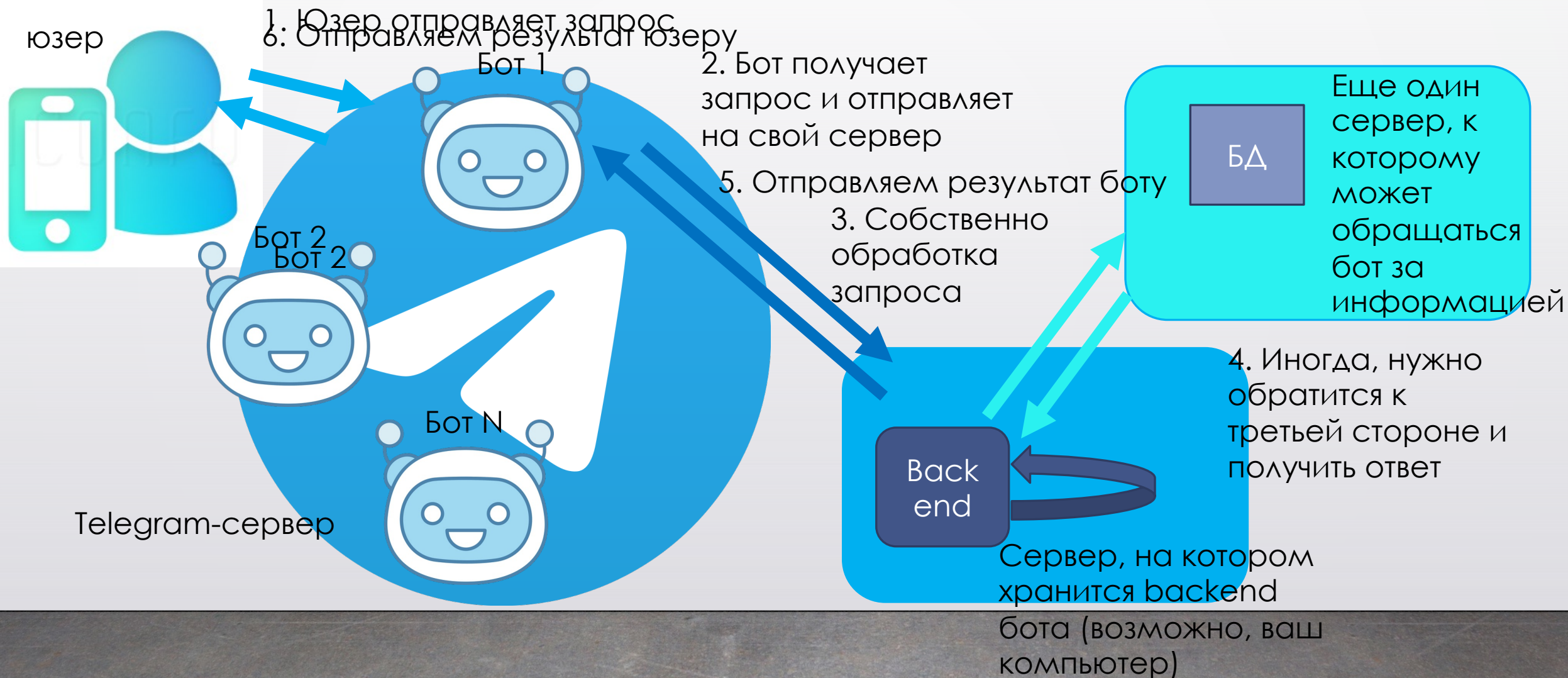


Ну и как же работает этот ваш бот?

- За Frontend отвечает сам Telegram, вам нужно только попросить у BotFather окрестить нового бота
- А вот Backend – это уже собственно программа, которую мы будем писать на Python при помощи библиотеки Aiogram, хотя конечно вариантов множество.



И где же боты? Давайте разбираться!

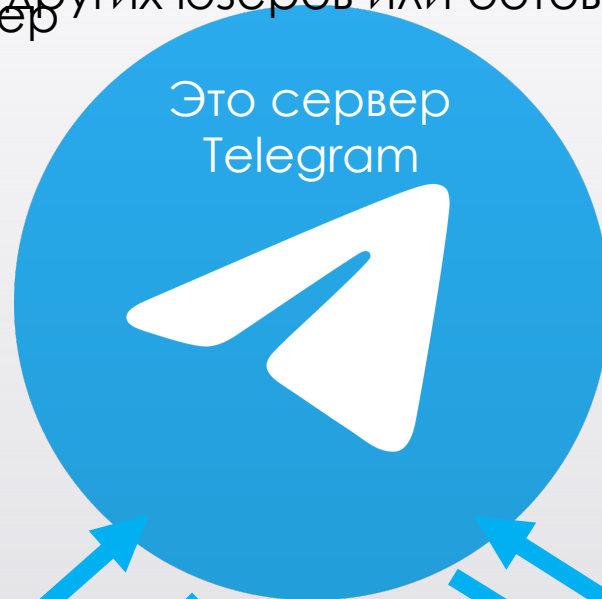


И что же происходит за спиной Botfather?

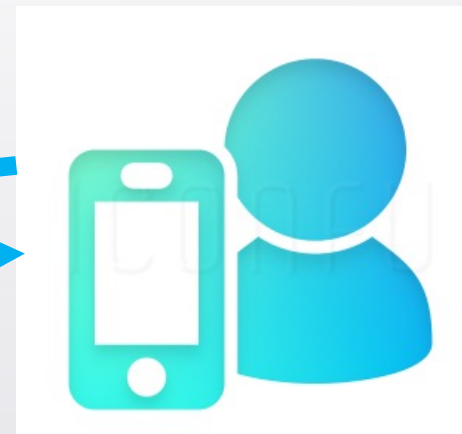
А это юзер, у которого есть устройство с Telegram



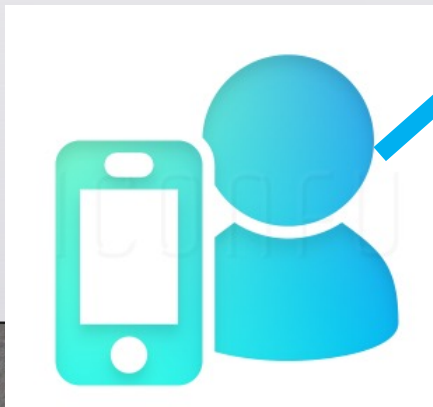
А уж потом сервер рассылает юзерам все их запросы и сообщения от других юзеров или ботов сначала на сервер



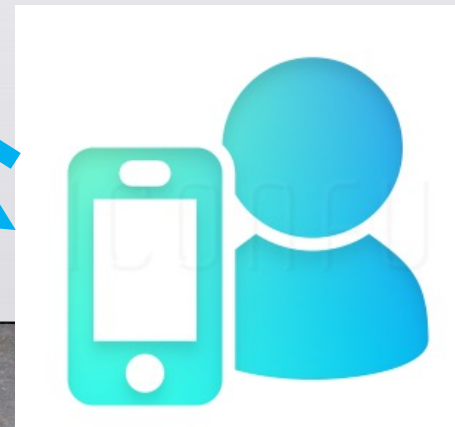
ОЧЕНЬ МНОГО ЮЗЕРОВ...



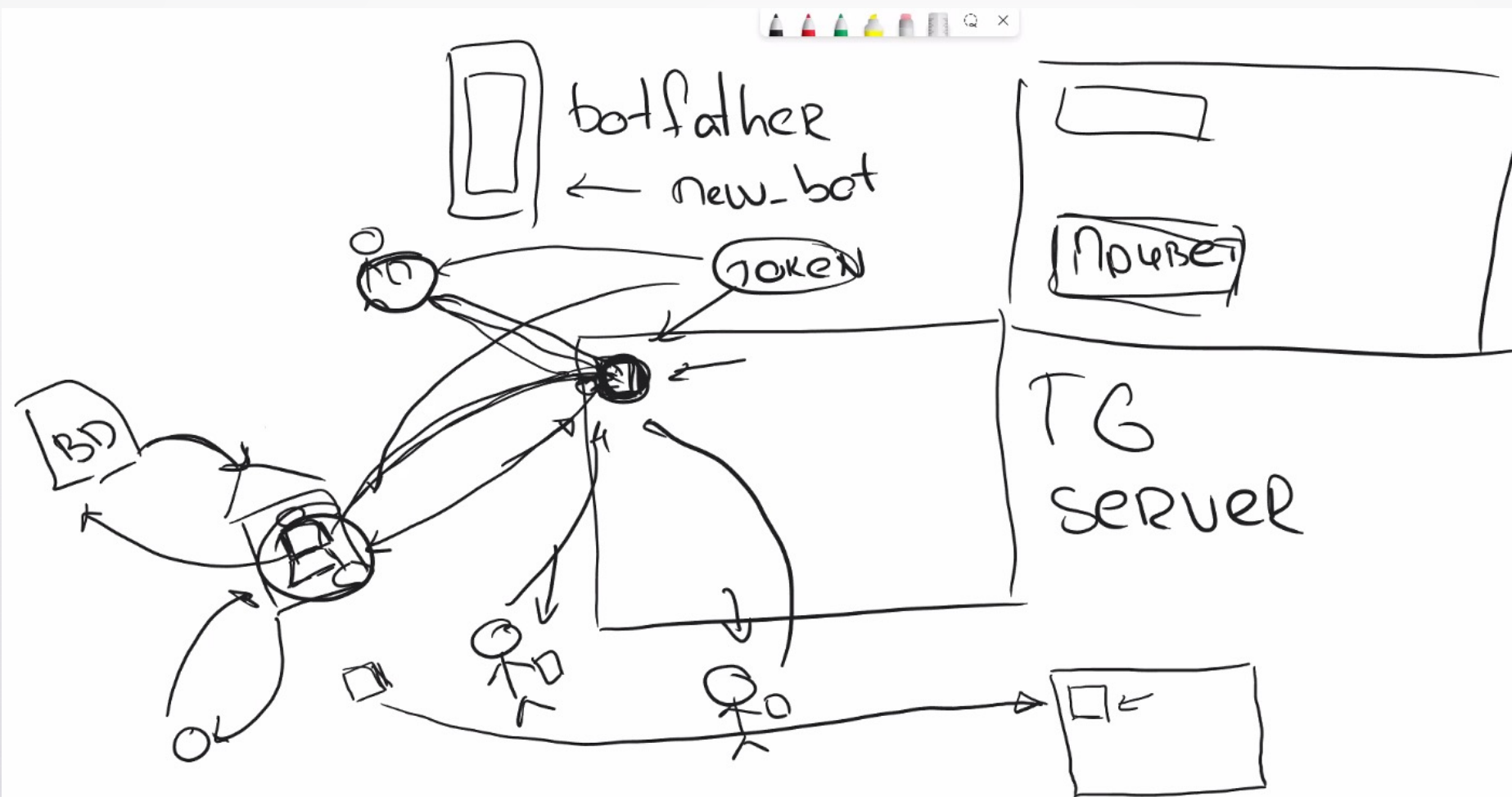
И еще один



И еще ...



А это божественный рисунок от Стоуна



Ага, понятно... Намудрили тут с простыми `input()` и `print()`

- Так да не так!
- Если с отправкой сообщения от бота юзеру все более менее похоже, разве что, есть свои команды, то с получением информации от пользователя все не так просто.
- Фактически, бот, как пионер, всегда готов и находится в режиме ожидания мессенджера для обработки. И это совсем не то же самое, что `input()`, который срабатывает, когда до него доходит код.
- А вот как обрабатывать приходящую от юзера информацию – уже другой разговор.



Так давайте же сделаем своего бота!

- Для этого пойдем к @BotFather (<https://t.me/BotFather>)
- И попросим его создать для нас нового бота /newbot



BotFather

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

- Теперь нам нужно бота окрестить, то есть создать ему имя, которое в общем-то может быть любым

Продолжаем делать бота



BotFather

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

- А вот теперь имя должно быть уникальным и заканчиваться на bot. Впрочем, если имя уже занято, BotFather любезно вам подскажет.
- После этого он пришлет длинное сообщение, в котором выдаст всю нужную информацию.



BotFather

Done! Congratulations on your new bot. You will find it at t.me/dddd_second_bot. 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:

5877338825:AAHzbV2zDxzGsBNTbGye5N9zu34Z6yV534

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>



Ссылка на новый бот

Токен – ключ к
вашему боту!
Никому его не
показывайте
(даже мне!)

Документация, с которой
неплохо бы ознакомиться

И что же теперь с этим делать?

- Можете отправляться прямиком в браузер и поиграть со своим ботом прямо сейчас
- Например, вы можете узнать о своем боте такой командой
https://api.telegram.org/botТУТ_ДОЛЖЕН_БЫТЬ_ВАШ_ТОКЕН/getMe
- https://api.telegram.org/botТУТ_ДОЛЖЕН_БЫТЬ_ВАШ_ТОКЕН/getUpdates - а так узнать об обновлениях.
- И даже отправить сообщение можно!
- https://api.telegram.org/botТУТ_ДОЛЖЕН_БЫТЬ_ВАШ_ТОКЕН/sendMessage?chat_id=ТУТ_ЦИФРЫ_ID_ЧАТА&text='А тут текст'

В браузере в ответ на все эти команды, вы будет получать json файл – формат обмена данными

Чтобы удобнее было работать, можно прогнать, через codeBeautify

The screenshot displays the Code Beautify website's JSON Viewer. The main area shows a large, minified JSON array of 15 messages. The interface includes a 'Tree Viewer' on the right, which is currently expanded to show the structure of the first message object. The tree view shows the following structure:

```
object (2)
  ok: true
  result (15)
    0 (2)
      update_id: 16350820
      message (6)
        message_id: 1
        from (7)
          id: 409205647
          is_bot: false
          first_name: Стоун
          last_name: Семидесятый
          username: STONE_Py
          language_code: ru
          is_premium: true
        chat (5)
          id: 409205647
          first_name: Стоун
          last_name: Семидесятый
          username: STONE_Py
          type: private
          date: 1679157832
          text: /start
          entities (1)
            0 (3)
              offset: 0
              length: 6
              type: bot_command
```

The interface also features a 'Beautify' button, a 'Minify' button, a 'Validate' button, and a 'Download' button. The 'JSON Full Form' option is selected. The bottom status bar indicates the file size is 5.66 KB.

Вот здесь можно увидеть все данные о сообщении: Id, имя и фамилию, текст сообщения и др.

Если наигрались с API, приступим наконец-то к коду!

- Для начала создадим проект, придумаем ему название. И если вы не хотите работать в PyCharm, тогда позаботьтесь об окружении.
- Установите Aiogram любым удобным для вас способом.
- Теперь нам нужно создать две сущности: собственно бота и диспатчера.
- Через бота мы привяжемся к нашему телеграм-боту и тут нам опять понадобится TOKEN
- Бота мы назовем bot
- А диспатчера dp
- Потом объясним, что это за звери



Итак, мы готовы кодить!

```
1 from aiogram import Bot, Dispatcher
2 from aiogram.utils import executor
3 from aiogram.types import Message
4
5 bot = Bot(token='6237643982:AAF13smExjrR6HSjF8UQS5M8uu_HoEHoNVg')
6 dp = Dispatcher(bot)
7
```

Из Aiogram импортируем Бота и Диспатчера
Ну и экзекUTOR нам пригодиться
А это чтобы проще работать с сообщениями
Создаем бота, которого называем Bot, прописываем токен (СВОЙ ТОКЕН!)
И диспатчера тоже называем так

- Если пока ничего не понятно – не страшно: Стоун еще раз объяснит, что это за сущности и почему так.

pythonProject6

venv library root

main.py

External Libraries

Scratches and Console

3

4

5

6

7

8

9

10

11

12

13

14

15

```
from aiogram.types import Message

bot = Bot(token='6237643982:AAF13smExjrR6HSjf8UQS5M8uu_HoEHoNVg')
dp = Dispatcher(bot)

@dp.message_handler(commands=['start'])
async def com_start(message: Message):
    await message.answer(f'{message.from_user.first_name}, тебе привет!')

executor.start_polling(dp, skip_updates=True)
```

Создаем хэндлер – это грубо говоря рука бота, которая будет выполнять определенную функцию в ответ на определенный «раздражитель»

И прописываем, что она делает

Про экзекутор немного позже, но он нужен, чтобы бот начал выполнять те функции, что на него возложены

Нужен декоратор

А теперь создаем функцию и указываем в качестве аргумента message

Terminal

Local

Requirement already satisfied: idna>=2.0 in c:\program files\python311\lib\site-packages (from yarl<2.0,>=1.0->aiohttp<3.9.0,>=3.8.0->aiogram) (3.4)

PS D:\WorkShopPy\pythonProject6>

pythonProject6

main.py

11:5 CRLF UTF-8 4 spaces Python 3.11 (pythonProject6)

pythonProject6

Dark

////////////////////////////////////
А вот как наш бот теперь умеет)))

Today



Diana M
/start

✓✓ 19:08

приветик

✓✓ 19:08



Diana M
/start

✓✓ 19:27



Наш первый бот
Diana, тебе привет!

19:27

- Теперь в ответ на команду /start наш бот вежливо здоровается!

Давайте усложним задачу!

```
8
9 .message_handler(commands=['start'])
10 async def com_start(update: Message):
11     await update.answer(f'{update.from_user.first_name}, тебе тоже привет')
12
13 .message_handler(content_types='sticker')
14 async def sticker(update: Message):
15     await update.answer('Какой охуенный стикер!')
16
17 .message_handler()
18 async def sticker(update: Message):
19     await update.answer(f'{update.from_user.first_name}, смотри что поймал - {update.text}')
20
```

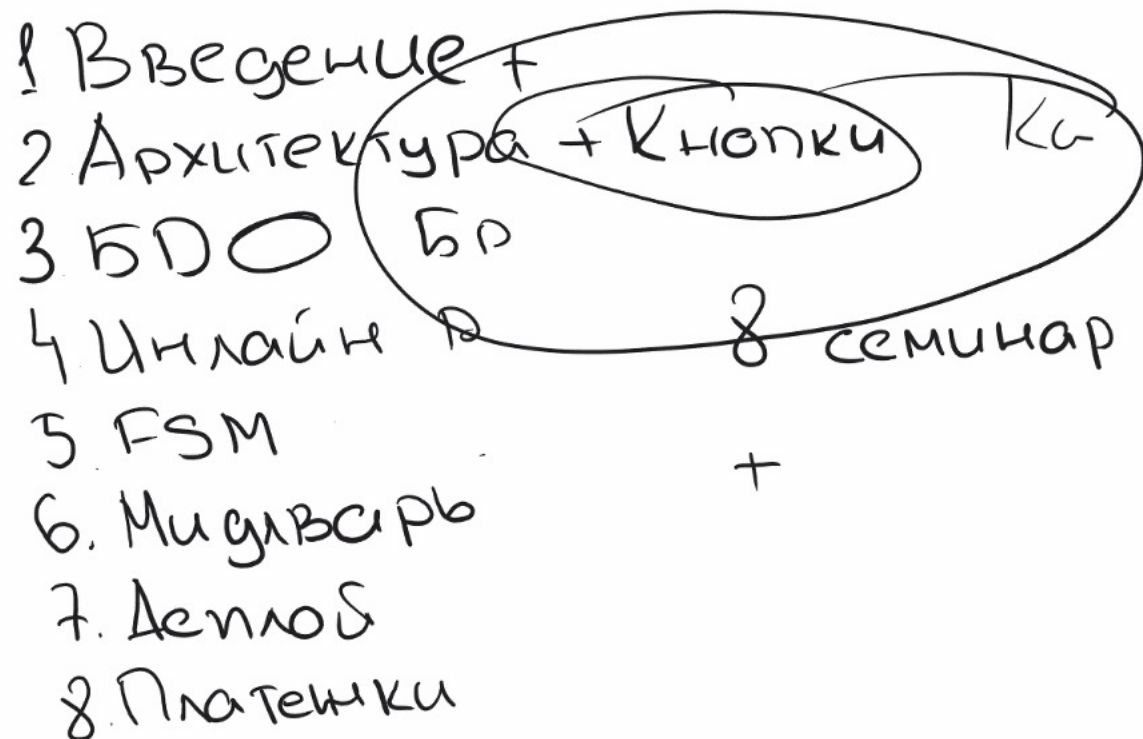
Эта функция отлавливает стикеры

А эта реагирует на любое текстовое сообщение и персылает его же, как эхо

```
Run ▶ ⚙ ⓘ
D:\WorkShopPy\my_venv\Scripts\python.exe D:\WorkShopPy\pythonProject4\main.py
Updates were skipped successfully.
Goodbye!
```

////////////////////////////////////

Не совсем понятно, как это работает?
Не беда – в процессе разберем все!



- 8 лекций
- 8 семинаров
- Записи
- Конспекты
- Поддержка
- Все для вас!

А теперь пару слов о курсе

От Стоуна

- + 8 полноценных лекций он-лайн, запись лекций, конспект и код (и офигенные рисунки!)
- + 8 настоящих семинаров, где можно отточить все полученные навыки и позадавать вопросы
- + поддержка вне лекций и семинаров: вы всегда можете задать вопрос и получить на него ответ
- + невероятная харизма!

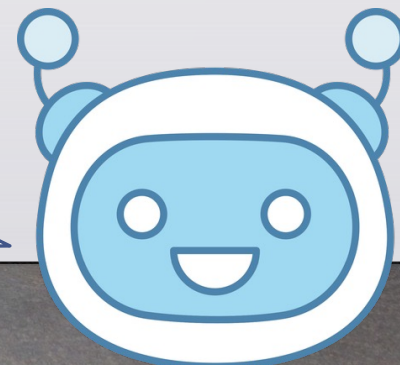
От команды:

- + конспекты
- + ответы на организационные вопросы и не только – по мере сил будем отвечать и по ботам
- + доброжелательность и настрой

От вас:

- + желание и мотивация
- + 5000 руб

Мы вас ждем!



А ты записался

на курс от Стоуна?





Ну и еще немного ответов на вопросы

Что по итогу будем уметь?

Создавать полноценных ботов – с клавиатурой, обращением к базам данных и непрерывной работе на сервере. Тут все ограничивается вашей фантазией. НО важнее еще то, что это отличный способ погрузиться в Python на реальном проекте, который не стыдно и в резюме показать. Создание своего бота поможет вам работать с библиотеками, понять, что же такое эта ваша архитектура и почему это важно, мы немного заглянем в классы, пощупаем деплой и в целом получим навык практического программирования

Но я вообще-то не разработчик, а аналитик, тестировщик... (нужное подчеркнуть), зачем мне этот курс?

В принципе азы программирования никому не повредят, а кроме того при помощи ботов вы можете решать свои задачи – автоматизация тестирования, сбор и первичный анализ данных.

-



Сколько будет занятий?

Планируется 8 лекций и 8 семинаров, но при необходимости можно провести дополнительный семинар

А расписание?

В начале недели Кирилл будет скидывать расписание на неделю – в среднем по 3 занятия, например 2 лекции и 1 семинар. По времени будем учитывать ваши пожелания, но всем не угодить

Как будет проходить занятие?

Zoom в помощь))) Будем собираться на конференции, смотреть, как рисует и кодит Кирилл, получать ответы прямо в процессе. На семинарах будем писать бота, разбирать код практически построчно и отвечать на практические вопросы



А запись будет?

Да, да и еще раз да!

Что на счет домашних заданий?

Они будут – но исключительно добровольные. Мы здесь не для оценок собрались. Но тем не менее, задания лучше выполнять – это поможет лучше усвоить курс, набить руку и выявить проблемные места

Что на счет оплаты?

Пишем Стоуну в личку. И имеем в виду, что это последний набор по такой цене.

Еще вопросы?

Не стесняемся и задаем их в личку Стоуну или Диане