**Быстрое открытие терминала. Числа**

ПКМ на папку => Службы => Новый терминал по адресу папки

pwd – посмотреть адрес папки

ls – посмотреть файлы в текущей папке

python3 *имя*.py – запустить данный файл

**Простые типы данных. Числа**

a = 9

>>> import math

>>> math.sqrt(a)

3.0

>>> math.sin(a)

0.4121184852417566

>>> a = 3.14

>>> b = 3.64

>>> round(a)

3

>>> round(b)

4

>>> math.floor(a)

3

>>> math.floor(b)

3

>>> math.ceil(a)

4

>>> math.ceil(b)

4

>>>

Корень

Синус

округление

округление в меньшую сторону

округление в большую сторону

**Простые типы данных. Строки**

>>> a = 'hello'

>>> a.upper()

'HELLO'

>>> a = 'HELLO'

>>> a.lower()

'hello'

>>> a.capitalize()

'Hello'

>>> a.replace('E', 'A')

'HALLO'

>>> a = 'learn.python.com'

>>> a.split('.')

['learn', 'python', 'com']

>>>

>>>

>>>

>>>

>>> a = 24

>>> b = 'I am '

>>> massage = b + a

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

TypeError: can only concatenate str (not "int") to str

>>> massage = b + str(a)

>>> massage

'I am 24'

>>>

>>>

>>>

>>>

>>> a = 20

>>> b = 2001

>>> message = 'I am {}. I was born in {}'.format(a,b)

>>> message

'I am 20. I was born in 2001'

Все буквы заглавные

Все буквы прописные

Первая большая буква, а остальные прописные

Заменить одну букву на другую

Разделить

Преобразовать значение в строку

**Сложные типы данных. Списки**

>>> a = [1,6,8,5,2]

>>> a[0]

1

>>> a[10]

Traceback (most recent call last):

File "<stdin>", line 1, in <module>

IndexError: list index out of range

>>> a

[1, 6, 8, 5, 2]

>>> a[2:4]

[8, 5]

>>> a

[1, 6, 8, 5, 2]

>>> a.append(99)

>>> a

[1, 6, 8, 5, 2, 99]

>>> a = [1,2,3]

>>> b = [4,5,6]

>>> c = a+b

>>> c

[1, 2, 3, 4, 5, 6]

>>> c.reverse()

>>> c

[6, 5, 4, 3, 2, 1]

>>> c.sort()

>>> c

[1, 2, 3, 4, 5, 6]

>>>

Список, данные в []

0 – первое значение

Ошибка если нет такого числа в списке

Вывести элементы с 2-4

Добавить еще один элемент

Сложить элементы двух списков

Обратная очерёдность

Сортировка элементов

**Сложные типы данных. Словари.**

Добавить новые элементы

catalog\_item = {

“type”:”phone”,

“vendor”:”Apple”,

“model”:”Iphone 12”

“price”:37.5

}

>>> catalog\_item[‘audio\_jack’] = False *добавить значение*

>>> catalog\_item[‘price’] = 70 *изменить значение*

>>> print(catalog\_item)

{

“type”:”phone”,

“audio\_jack”:False,

“vendor”:”Apple”,

“model”:”Iphone 12”

“price”:70

}

*Возьмем данные по ключу*

>>> print(catalog\_item[‘price’])

70

*Сложить два значения*

>>> item\_name = catalog\_item[‘vendor’]+ ‘catalog\_item[‘model’]

>>> print(item\_name)

Apple Iphone 12

>>> print(catalog\_item[‘discount’])

KeyError:’discount’

*Такого ключа нет.*

Добавить ключ. Метод .get()

>>> catalog\_item.get(‘discount’,’Скидок нет!’)

‘Скидок нет!’

Есть ли ключ в словаре?

>>> ‘model’ in catalog\_item

True

>>> ‘discount’ not in catalog\_item

True

>>> ‘discount’ in catalog\_item

False

Удалить ключ в каталоге Python

>>>del catalog\_item [‘price’]

>>> print(catalog\_item)

*Удаление ключа которого нет*

>>>del catalog\_item [‘discount’]

KeyError:’discount’

**Переменные.**

>>> a=2

>>> print(a+3)

5

>>> b=6

>>> a=b

>>> a+b

12

>>> a=2

>>> a=a+1

>>> print (a)

3

*Правила присвоения названия переменным:*

1. переменные всегда на английском;
2. название переменной отражает точный смысл того, что мы хотим с ней сделать.

**Функции.**

**Функция** - кусочек кода, которому мы даем имя и можем его вызывать из различных ест программы.

Функции помогают структурировать код, разбив его на отдельные действия, которые легко читать и легко отлаживать.

Написав функцию один раз мы можем ее использовать в разных частях кода.

Синтаксис.

def get\_vat(payment):

return 0

*def* - определить(define)

*get\_vat* –имя функции, что она делает

*payment* – параметр который принимает функция

*return* *0* – тело функции, функция, которая возвращает 0

*File text:*

def get\_vat(payment):

try:

vat = payment / 100 \* 20

return vat

result = get\_vat(40)

print(result)Изображение выглядит как текст, снимок экрана, монитор

Автоматически созданное описание

*Python result:*

8.0

*File text:*

def get\_vat(payment):

try:

vat = payment / 100 \* 20

return vat

except TypeError:

return"Can't count. Check data!"

result = get\_vat("olol")

print(result)

Изображение выглядит как текст, снимок экрана, монитор

Автоматически созданное описание

*Python result:*

Can't count. Check data!

*File text:*

def get\_vat(payment, persent=20):

try:

payment = float(payment)

vat = payment / 100 \* persent

vat = round (vat, 2)

return "Sum VAT: {}".format(vat)

except TypeError:

return"Can't count. Check data!"

result = get\_vat(400, 15)

print(result)

*Изображение выглядит как текст, снимок экрана, монитор, экран

Автоматически созданное описание*

*Python result:*

60

**Bot в Телеграмм**

Открываем BotFather

*Команды BotFather:*

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](https://core.telegram.org/bots/inline)

/setinlinegeo - toggle inline [location requests](https://core.telegram.org/bots/inline#location-based-results)

/setinlinefeedback - change [inline feedback](https://core.telegram.org/bots/inline#collecting-feedback) settings

/setjoingroups - can your bot be added to groups?

/setprivacy - toggle [privacy mode](https://core.telegram.org/bots#privacy-mode) in groups

**Games**

/mygames - edit your [games](https://core.telegram.org/bots/games) **[beta]**

/newgame - create a new [game](https://core.telegram.org/bots/games)

/listgames - get a list of your games

/editgame - edit a game

/deletegame - delete an existing game

*Пишем команду в окне BotFather:*

/newbot

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

Learn Python Simplebot

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

Learn\_python\_simplebot1\_bot

Done! Congratulations on your new bot. You will find it at t.me/Learn\_python\_simplebot1\_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:

5197935504:AAFrj0YRvrxfE7JlUGnjoicbTogige9iclk

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>

**Выводим фразу « Hello » через Telegram в Терминале**

*Открываем термина и вводим команду, чтобы установить библиотеку Telegram:*

pip3 install python-telegram-bot

*(должна установиться библиотека телеграмм)*

*Далее пишем код в Терминале (файл bot.py):*

from telegram.ext import Updater, CommandHandler

def start\_bot(bot, updater):

print("Hello")

def main():

updtr = Updater('5197935504:AAFrj0YRvrxfE7JlUGnjoicbTogige9iclk')

updtr.dispatcher.add\_handler(CommandHandler("start", start\_bot))

updtr.start\_polling()

updtr.idle()

if \_\_name\_\_ == "\_\_main\_\_":

main()

Изображение выглядит как текст

Автоматически созданное описание

Далее открываем файл bot.py в Терминале

*Прописываем в терминале:*

python3 bot.py

*Заходим в наш телеграмм по ссылке выше*

t.me/Learn\_python\_simplebot1\_bot

*и пишем там*

/start

Изображение выглядит как текст

Автоматически созданное описаниеВ окне терминала появилась фраза “Hello”



**Выводим фразу в самом Telegram**

*Набираем в Терминале текст:*

from telegram.ext import Updater, CommandHandler, CallbackContext

def start\_bot(update: Updater, context: CallbackContext):

mytext = """Hello user

I have only /start command! =)"""

update.message.reply\_text(mytext)

def main():

updtr = Updater('5197935504:AAFrj0YRvrxfE7JlUGnjoicbTogige9iclk')

updtr.dispatcher.add\_handler(CommandHandler("start", start\_bot))

updtr.start\_polling()

updtr.idle()

if \_\_name\_\_ == "\_\_main\_\_":

main()

Изображение выглядит как текст

Автоматически созданное описание

Далее открываем файл bot.py в Терминале

*Прописываем в терминале:*

python3 bot.py

*Заходим в наш телеграмм по ссылке выше*

t.me/Learn\_python\_simplebot1\_bot

*и пишем там*

/start

*И получаем следующий текст в окне Telegram:*

Hello user

I have only /start command! =)

Изображение выглядит как текст

Автоматически созданное описание

**Выводим фразу в самом Telegram через Словарь**

*Набираем в Терминале текст, через файл bot.py:*

from telegram.ext import Updater, CommandHandler, CallbackContext

def start\_bot(update: Updater, context: CallbackContext):

print(update)

mytext = """Hello user

I have only /start command! =)"""

update.message.reply\_text(mytext)

def main():

updtr = Updater('5197935504:AAFrj0YRvrxfE7JlUGnjoicbTogige9iclk')

updtr.dispatcher.add\_handler(CommandHandler("start", start\_bot))

updtr.start\_polling()

updtr.idle()

if \_\_name\_\_ == "\_\_main\_\_":

main()

Изображение выглядит как текст

Автоматически созданное описание

*Набираем в Терминале текст:*

python3 bot.py

*Заходим в приложение Teleram и пишем там:*

/start

*И получаем следующий текст (словарь) в Терминала:*

{'message': {'photo': [], 'new\_chat\_photo': [], 'channel\_chat\_created': False, 'supergroup\_chat\_created': False, 'entities': [{'type': 'bot\_command', 'offset': 0, 'length': 6}], 'text': '/start', 'message\_id': 14, 'new\_chat\_members': [], 'caption\_entities': [], 'group\_chat\_created': False, 'date': 1649624518, 'chat': {'id': 58325403, 'last\_name': 'Chayka', 'first\_name': 'Maria', 'type': 'private'}, 'delete\_chat\_photo': False, 'from': {'first\_name': 'Maria', 'id': 58325403, 'language\_code': 'ru', 'last\_name': 'Chayka', 'is\_bot': False}}, 'update\_id': 911657443}

Делаем наше сообщение в Telegram более конкретным, меняем файл bot.py, сохраняем его и запускаем:

from telegram.ext import Updater, CommandHandler, CallbackContext

def start\_bot(update: Updater, context: CallbackContext):

print(update)

mytext = """Hello {}

I have only /start command! =)""".format(update.message.chat.first\_name)

update.message.reply\_text(mytext)

def main():

updtr = Updater('5197935504:AAFrj0YRvrxfE7JlUGnjoicbTogige9iclk')

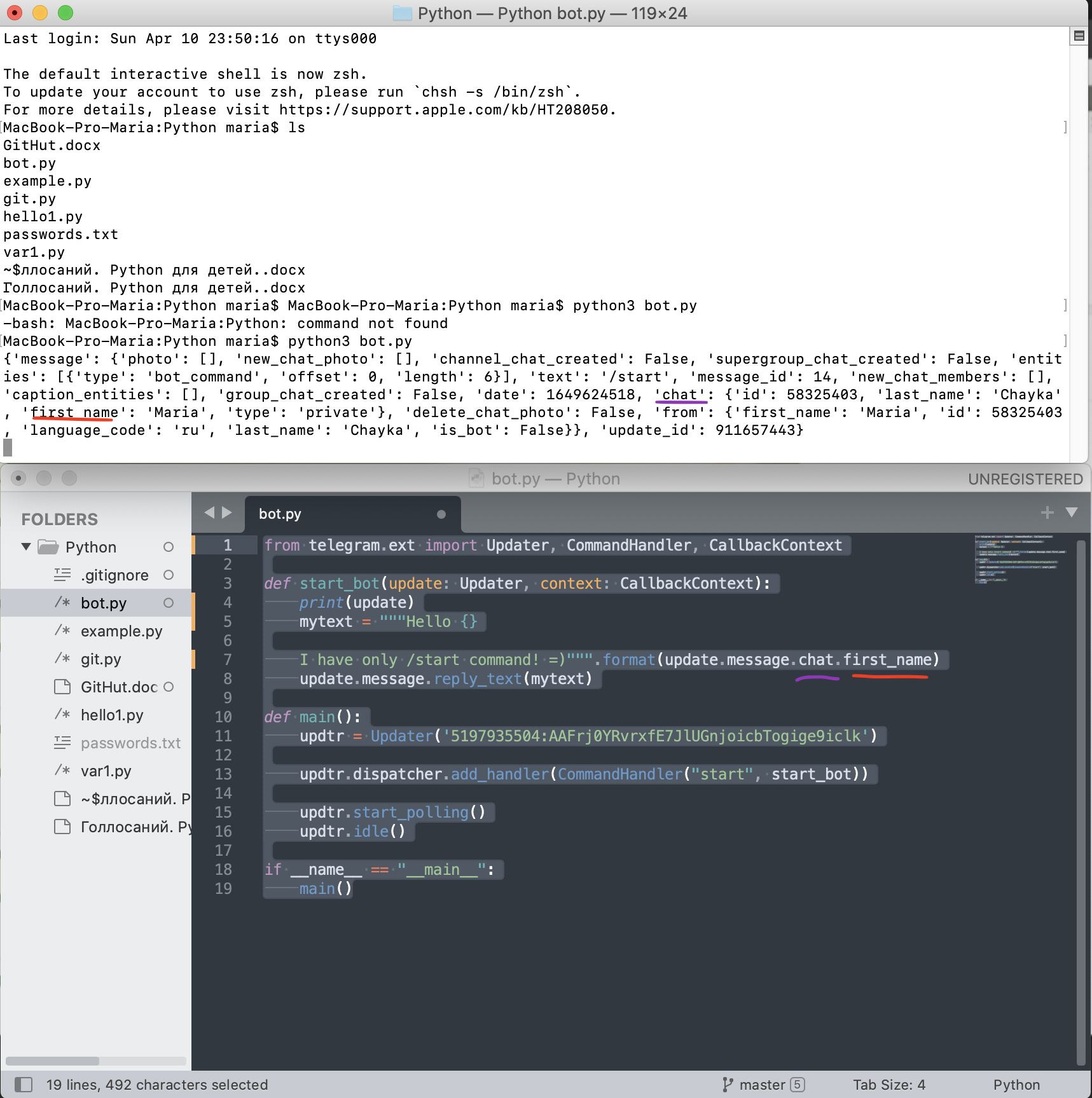
updtr.dispatcher.add\_handler(CommandHandler("start", start\_bot))

updtr.start\_polling()

updtr.idle()

if \_\_name\_\_ == "\_\_main\_\_":

main()



*Заходим в приложение Teleram и пишем там:*

/start

*Получаем следующий текст в окне Telegram:*

*Изображение выглядит как текст

Автоматически созданное описание*

**Добавляем нашему боту в Telegram файл логирования bot.log, чтобы отслеживать его состояние.**

*Правим наш файл bot.py:*

import logging

from telegram.ext import Updater, CommandHandler, CallbackContext

logging.basicConfig (format='%(asctime)s - %(levelname)s - %(message)s',

level=logging.INFO,

filename='bot.log'

)

def start\_bot(update: Updater, context: CallbackContext):

print(update)

mytext = """Hello {}

I have only /start command! =)""".format(update.message.chat.first\_name)

update.message.reply\_text(mytext)

def main():

updtr = Updater('5197935504:AAFrj0YRvrxfE7JlUGnjoicbTogige9iclk')

updtr.dispatcher.add\_handler(CommandHandler("start", start\_bot))

updtr.start\_polling()

updtr.idle()

if \_\_name\_\_ == "\_\_main\_\_":

logging.info('Bot started!')

main()

Изображение выглядит как текст

Автоматически созданное описание

*Набираем в Терминале текст:*

python3 bot.py

*У нас появляется новый файл в Sublime text bot.log:*

Изображение выглядит как текст

Автоматически созданное описание

*Заходим в приложение Teleram и запускаем его пишем там:*

/start

*Получаем следующий текст в окне Telegram:*

*Изображение выглядит как текст

Автоматически созданное описание*

*Закрываем наш Бот Телеграмм, набираем в Терминале текст:*

^C

(Не получается с этой командой!)

*Теперь в нашем новом файле bot.log появился следующий текст:*

Изображение выглядит как текст

Автоматически созданное описание

**Добавляем нашему боту в Telegram возможность ответа на наши вопросы.**

Часть 1. Делаем чтобы выводилась фраза Hello на любой набранный текст.

*Правим наш файл bot.py:*

import logging

from telegram.ext import Updater, CommandHandler, CallbackContext, MessageHandler, Filters

logging.basicConfig(format='%(asctime)s - %(levelname)s - %(message)s',

level=logging.INFO,

filename='bot.log'

)

def start\_bot(update: Updater, context: CallbackContext):

print(update)

mytext = """Hello {}

I have only /start command! =)""".format(update.message.chat.first\_name)

update.message.reply\_text(mytext)

def chat(update: Updater, context: CallbackContext):

text = "Hello"

update.message.reply\_text(text)

def main():

updtr = Updater('5197935504:AAFrj0YRvrxfE7JlUGnjoicbTogige9iclk')

updtr.dispatcher.add\_handler(CommandHandler("start", start\_bot))

updtr.dispatcher.add\_handler(MessageHandler(Filters.text, chat))

updtr.start\_polling()

updtr.idle()

if \_\_name\_\_ == "\_\_main\_\_":

logging.info('Bot started!')

main()

Изображение выглядит как текст

Автоматически созданное описание

*Набираем в Терминале текст:*

python3 bot.py

*Заходим в приложение Teleram и запускаем его пишем там:*

фывфывфыв

*Получаем текст Hello в окне Telegram:*

Изображение выглядит как текст

Автоматически созданное описание

Часть 2. Делаем чтобы выводилась фраза, которую мы написали в Телеграмм и bot.log файле.

*Правим наш файл bot.py:*

import logging

from telegram.ext import Updater, CommandHandler, CallbackContext, MessageHandler, Filters

logging.basicConfig(format='%(asctime)s - %(levelname)s - %(message)s',

level=logging.INFO,

filename='bot.log'

)

def start\_bot(update: Updater, context: CallbackContext):

print(update)

mytext = """Hello {}

I have only /start command! =)""".format(update.message.chat.first\_name)

update.message.reply\_text(mytext)

def chat(update: Updater, context: CallbackContext):

text = update.message.text

logging.info(text)

update.message.reply\_text(text)

def main():

updtr = Updater('5197935504:AAFrj0YRvrxfE7JlUGnjoicbTogige9iclk')

updtr.dispatcher.add\_handler(CommandHandler("start", start\_bot))

updtr.dispatcher.add\_handler(MessageHandler(Filters.text, chat))

updtr.start\_polling()

updtr.idle()

if \_\_name\_\_ == "\_\_main\_\_":

logging.info('Bot started!')

main()

Изображение выглядит как текст

Автоматически созданное описание

*Набираем в Терминале текст:*

python3 bot.py

*Заходим в приложение Teleram и запускаем его пишем там:*

fsds

*Получаем текст Hello в окне Telegram:*

Изображение выглядит как текст

Автоматически созданное описание

*Теперь в нашем новом файле bot.log появился такой же текст fsds:*

Изображение выглядит как текст

Автоматически созданное описание