

## Создание чата с помощью языка программирования Python

*Эрдман Александр Алексеевич*

*Приамурский государственный университет имени Шолом-Алейхема*

*Студент*

### Аннотация

В статье рассмотрен процесс создания программы чата в браузере. Программа написана на языке программирования Python с использованием библиотеки PyWebIO, которая позволяет генерировать web-интерфейс и работать с вводом и выводом в браузере. Результатом исследования будет являться браузерный чат и описание его работы.

**Ключевые слова:** чат, python, pywebio

## Creating an online chat using the Python programming language

*Erdman Alexander Alekseevich*

*Sholom-Aleichem Priamursky State University*

*Student*

### Abstract

The article describes the process of creating a chat program in the browser. The program is written in the Python programming language using the PyWebIO library, which allows you to generate a web interface and work with input and output in the browser. The result of the study will be a browser-based chat and a description of its work.

**Keywords:** chat, python, pywebio

## 1 Введение

### 1.1 Актуальность

В современном мире всё сильнее обостряется проблема общения между людьми. Ни для кого не секрет, что общение друг с другом – один из главных факторов, благодаря которому можно назвать человека человеком. Общение крайне важно для социализации человека в обществе, потому что человек социальное существо, которому обязательно нужно взаимодействовать с обществом: общаться с друзьями и родственниками, уметь договариваться с конкретными личностями, начиная от работодателя с вопросом о своей карьере и заканчивая воспитательным договором с маленьким ребёнком. Если же не общаться и не развивать свой навык общения, то возможны негативные последствия, которые могут быть как неловким моментом, когда человек не смог правильно выразить свою мысль и его никто не понял, так и серьёзные психологические проблемы. В 21 веке существует множество способов общения: телефонный разговор, электронные письма, множество

социальных сетей и мессенджеров, голосовые чаты как игр, так и отдельных специализированных программ (Discord, Zoom). Но самым простым и наиболее популярным способом общения является чат. Данный способ присутствует практически во всех программах и приложениях, главной целью которых является обеспечение людей тем или иным способом общения. Также данный способ распространен в использовании как между рабочими для общения друг с другом, так и между детьми для обычных дружеских бесед. Исходя из такой популярности чатов следует вывод, что чаты очень востребованы, а значит востребована разработка и поддержка данных программ, которая будет описана в данной статье.

## 1.2 Обзор исследований

Е.И. Еськин и Т.С. Катермина разработали чат-бота для коммерческих и информационных компаний [1]. Дарья Зубко рассмотрела языки программирования интернет-чатов [2]. С.С. Зайцева произвела обзор некоторых возможностей чат-ботов [3]. Е.Г. Луговская проанализировала экономию и избыточность речевых усилий в чатах [4]. Е.Д. Соболева, М.А. Постникова и другие рассмотрели технологии, позволяющие создавать чат-ботов [5].

## 1.3 Цель исследования

Целью исследования является создание чата в браузере при помощи языка программирования Python и библиотеки PyWebIO.

## 2 Материалы и методы

Для создания программы используется язык программирования Python и библиотека PyWebIO. В качестве IDE используется Pycharm.

## 3 Результаты и обсуждения

Для начала написания программы необходимо установить библиотеку PyWebIO. Для её установки нужно прописать следующую команду в терминале pycharm «pip install pywebio». После установки библиотеки нужно импортировать из неё необходимые модули, а также подключить модуль asyncio для функции, которая будет в асинхронном режиме обновлять список сообщений в чате (рис. 1).

```
1 import asyncio
2
3 from pywebio import start_server
4 from pywebio.input import *
5 from pywebio.output import *
6 from pywebio.session import defer_call, info as session_info, run_async, run_js
```

Рис. 1. Импорт модулей библиотеки PyWebIO и модуля asyncio

После импорта всех необходимых модулей нужно объявить 2 переменные и одну константу (рис. 2).

```
8 chat_msgs = []
9 online_users = set()
10
11 MAX_MESSAGES_COUNT = 100
```

Рис. 2. Объявление переменных и константы

Переменная «chat\_msgs = []» - это глобальный список сообщений в чате, переменная «online\_users = set()» - это множество, состоящее из онлайн пользователей в чате, константа «MAX\_MESSAGES\_COUNT = 100» отвечает за указанное количество сохраняемых сообщений в чате, в данном случае указанным значением является 100. Константа необходима для того, чтобы оперативная память пользователей не заполнялась до предельного объёма. После объявляется главная функция «main» и запуск сервера (рис. 3).

```
14 async def main():
15     global chat_msgs
16
17
18     if __name__ == "__main__":
19         start_server(main, debug=True, port=8080, cdn=False)
```

Рис. 3. Объявление функции и запуска сервера

Функция «main» вызывается сервером, когда пользователь открывает чат в браузере, поэтому в функции необходимо прописать вывод сообщения в чате, информирующего пользователя о чате, а также запросить у пользователя его никнейм, при этом прописать условия, если никнейм уже занят. После того, как пользователь введёт свои данные, то соответствующее уведомление появиться в чате (рис. 4).

```
14 async def main():
15     global chat_msgs
16
17     put_markdown("Добро пожаловать в чат!")
18
19     msg_box = output()
20     put_scrollable(msg_box, height=300, keep_bottom=True)
21
22     nickname = await input("Войти в чат", required=True, placeholder="Ваше имя",
23                            validate=lambda n: "Такое имя уже используется!" if n in online_users or n == '❄' else None)
24     online_users.add(nickname)
25
26     chat_msgs.append(❄, f'❄ {nickname} присоединился к чату!')
27     msg_box.append(put_markdown(f'❄ ❄ {nickname} присоединился к чату'))
```

Рис. 4. Ввод и вывод информации о чате и пользователе

После регистрации пользователя прописывается бесконечный цикл с постоянным запросом ввода от пользователя. Вместе с этим создаются две кнопки, отвечающие за отправку сообщения и за выход из чата. Если пользователь пишет и отправляет сообщение, то после отправки сообщения ему будет предложено снова ввести и отправить сообщение, а также выйти из чата (рис 5).

```

31 while True:
32     data = await input_group("📩 Новое сообщение", [
33         input(placeholder="Текст сообщения ...", name="msg"),
34         actions(name="cmd", buttons=["Отправить", {'label': "Выйти из чата", 'type': 'cancel'}])
35     ], validate=lambda m: ('msg', "Введите текст сообщения!") if m["cmd"] == "Отправить" and not m['msg'] else None)
36
37     if data is None:
38         break
39
40     msg_box.append(put_markdown(f"{nickname}: {data['msg']}"))
41     chat_msgs.append((nickname, data['msg']))

```

Рис. 5. Создание отправки сообщений и выхода из чата

Если пользователь выйдет из чата, то необходимо прописать уведомление о выходе в чате, удалить никнейм пользователя из общего списка имён пользователей. Также можно добавить кнопку «Перезайти», в случае если возникнут какие-либо проблемы у пользователя чата (рис. 6).

```

42 # Выход из чата
43 refresh_task.close()
44
45 online_users.remove(nickname)
46 toast("Вы вышли из чата!")
47 msg_box.append(put_markdown(f'👋 Пользователь `{nickname}` покинул чат!'))
48 chat_msgs.append(('👋', f'Пользователь `{nickname}` покинул чат!'))
49
50 put_buttons(['Перезайти'], onclick=lambda btn: run_js('window.location.reload()'))

```

Рис. 6. Создание уведомлений выхода из чата и возможности перезайти

Далее необходимо запрограммировать функцию «refresh\_msg». Но сперва нужно её запустить после того, как пользователь зарегистрировался в чате, то есть после того, как он ввёл своё имя. Данная функция отвечает за обновление списка сообщений в чате через указанное время. Функция представлена бесконечным циклом, в котором после каждой секунды идёт проверка новых сообщений от пользователей, в случае нахождения которых производится добавление сообщения в чат. Наравне с поиском нужно прописать проверку списка сообщений. Если их будет больше, чем было указано в константе «MAX\_MESSAGES\_COUNT = 100», то нужно очистить список, чтобы не переполнять оперативную память (рис. 7).

```

53 async def refresh_msg(nickname, msg_box):
54     global chat_msgs
55     last_idx = len(chat_msgs)
56
57     while True:
58         await asyncio.sleep(1)
59
60         for m in chat_msgs[last_idx:]:
61             if m[0] != nickname:
62                 msg_box.append(put_markdown(f"{m[0]}: {m[1]}"))
63
64
65         if len(chat_msgs) > MAX_MESSAGES_COUNT:
66             chat_msgs = chat_msgs[len(chat_msgs) // 2:]
67
68         last_idx = len(chat_msgs)

```

Рис. 7. Создание функции обновления и очистки списка сообщений в чате

Программа чата готова. Полный код программы (рис. 8 и 9). Теперь после запуска программы в консоли появится ссылка, которую нужно

скопировать и вставить в браузер и перейти по ней. После перехода по ссылке будет представлен онлайн чат (рис. 10)

```

1 import asyncio
2
3 from pywebio import start_server
4 from pywebio.input import *
5 from pywebio.output import *
6 from pywebio.session import defer_call, info as session_info, run_async, run_js
7
8 chat_msgs = []
9 online_users = set()
10
11 MAX_MESSAGES_COUNT = 100
12
13
14 async def main():
15     global chat_msgs
16
17     put_markdown("Добро пожаловать в чат!")
18
19     msg_box = output()
20     put_scrollable(msg_box, height=300, keep_bottom=True)
21
22     nickname = await input("Войти в чат", required=True, placeholder="Ваше имя",
23                          validate=lambda n: "Такое имя уже используется!" if n in online_users or n == '❄' else None)
24     online_users.add(nickname)
25
26     chat_msgs.append('❄', f'"{nickname}" присоединился к чату!')
27     msg_box.append(put_markdown(f'❄ "{nickname}" присоединился к чату'))
28
29     refresh_task = run_async(refresh_msg(nickname, msg_box))
30
31     while True:
32         data = await input_group("❄ Новое сообщение", [
33             input(placeholder="Текст сообщения ...", name="msg"),
34             actions(name="cmd", buttons=["Отправить", {'label': "Выйти из чата", 'type': 'cancel'}])
35             ], validate=lambda m: ('msg', "Введите текст сообщения!") if m["cmd"] == "Отправить" and not m["msg"] else None)
36
37         if data is None:
38             break

```

Рис. 8. Код чата

```

40     msg_box.append(put_markdown(f'"{nickname}": {data["msg"]}'))
41     chat_msgs.append((nickname, data['msg']))
42     # Выход из чата
43     refresh_task.close()
44
45     online_users.remove(nickname)
46     toast("Вы вышли из чата!")
47     msg_box.append(put_markdown(f'❄ Пользователь "{nickname}" покинул чат!'))
48     chat_msgs.append('❄', f'Пользователь "{nickname}" покинул чат!'))
49
50     put_buttons(['Перезайти'], onClick=lambda btn: run_js('window.location.reload()'))
51
52
53 async def refresh_msg(nickname, msg_box):
54     global chat_msgs
55     last_idx = len(chat_msgs)
56
57     while True:
58         await asyncio.sleep(1)
59
60         for m in chat_msgs[last_idx:]:
61             if m[0] != nickname:
62                 msg_box.append(put_markdown(f'"{m[0]}": {m[1]}'))
63
64
65         if len(chat_msgs) > MAX_MESSAGES_COUNT:
66             chat_msgs = chat_msgs[len(chat_msgs) // 2:]
67
68         last_idx = len(chat_msgs)
69
70
71 if __name__ == "__main__":
72     start_server(main, debug=True, port=8080, cdn=False)

```

Рис. 9. Код чата (продолжение)

Добро пожаловать в чат!

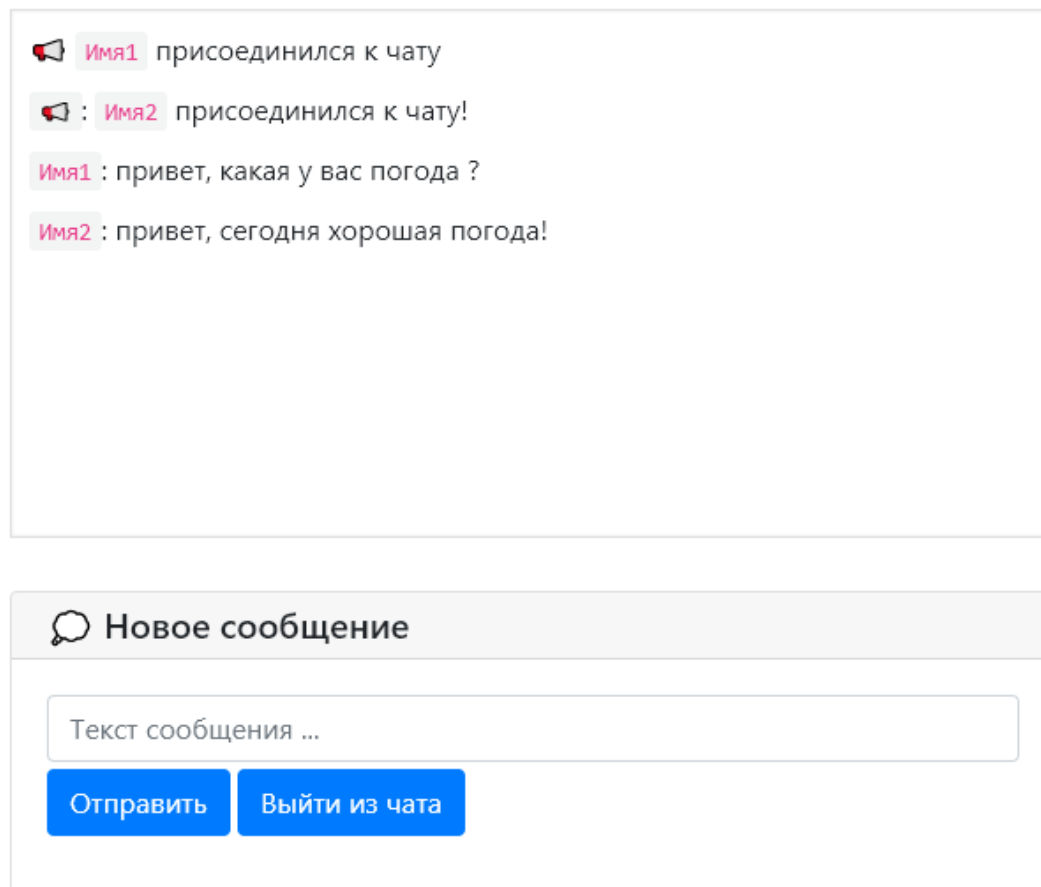


Рис. 10. Работа чата

Таким образом с помощью языка Python и библиотеки PyWebIO была разработана программа чата в браузере, а также был описан принцип работы данной программы.

### Библиографический список

1. Еськин Е.И., Катермина Т.С. Разработка чат-бота, его возможности и организация полноценного чата с оператором // Научный электронный журнал Меридиан. 2020. № 15 (49). С. 21-23.
2. Зубко Д. Язык интернет-чата (на примере чатов: chathome, volchat, p2p.cherovets, live chat) // В книге: IX Школьные Харитоновские чтения. Межрегиональная олимпиада школьников "Будущие исследователи - будущее науки". 2009. С. 150-152.
3. Зайцева С.С. О некоторых приложениях чат-ботов // В сборнике: СОВРЕМЕННЫЕ ТЕНДЕНЦИИ РАЗВИТИЯ ФУНДАМЕНТАЛЬНЫХ И ПРИКЛАДНЫХ НАУК. Материалы Всероссийской с международным участием научно-практической конференции. Под ред. С.А. Коньшаковой. 2018. С. 52-54.
4. Луговская Е.Г. Пунктуационно-графическое оформление речи vs эффективность чат-коммуникации // В сборнике: Язык и речь в

- Интернете: личность, общество, коммуникация, культура. сборник статей IV Международной научно-практической конференции: в 2 томах. Российский университет дружбы народов. 2020. С. 120-126.
5. Соболева Е.Д., Постникова М.А., Маркарян Р.Р., Гапанюк Ю.Е. Обзор технологий создания чат-ботов // Естественные и технические науки. 2019. № 5 (131). С. 207-210.