

# Вадим Крупинин

---

Разработчик на Python

Email: [vadimkrupinin@gmail.com](mailto:vadimkrupinin@gmail.com)

Telegram: @KVadim\_K

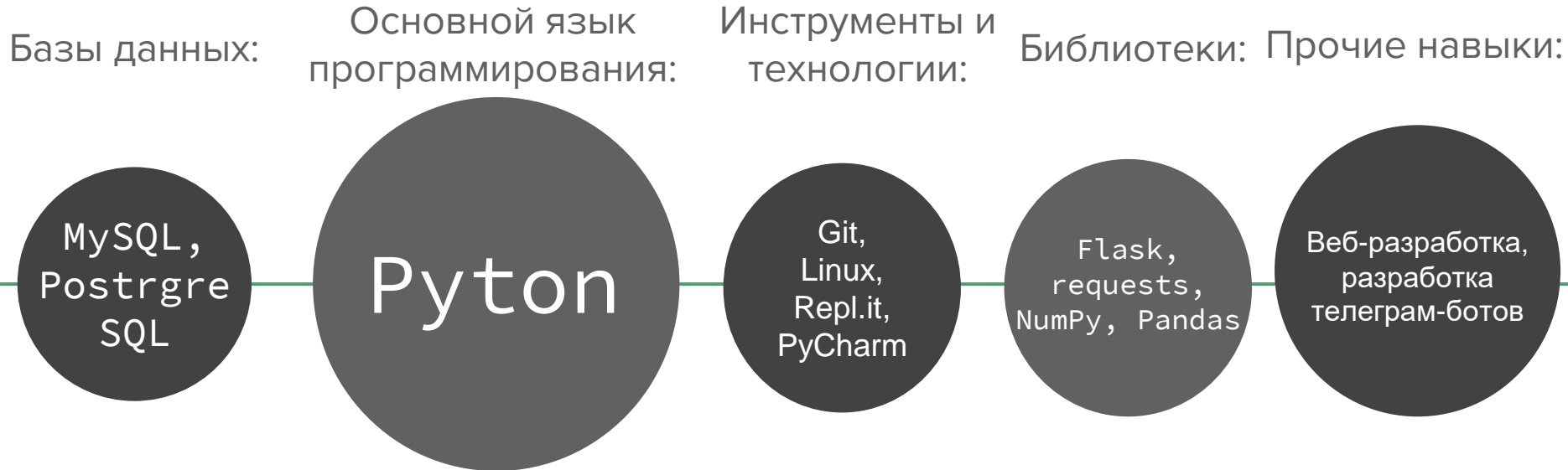
Github:

Linkedin:

# О себе

- Окончил ДГТУ
- Окончил курсы по программированию:
  - на Python от [Зерокодер](#)
  - [Интенсив по программированию на PYTHON с помощью chatGPT](#)
  - [Интенсива по разработке мобильных приложений](#)
- Имею большой интерес к машинному обучению и разработке веб-приложений.
- Люблю решать сложные и интересные задачи, постоянно развиваюсь в новых технологиях особенно работой с искусственным интеллектом.
- Я вырос в Ростове-на-Дону, увлекаюсь боевыми единоборствами и кроссфитом.
- Ищу интересные предложения!

# Навыки и знания



Проекты:

---

# H<sub>2</sub>O Reminder Bot

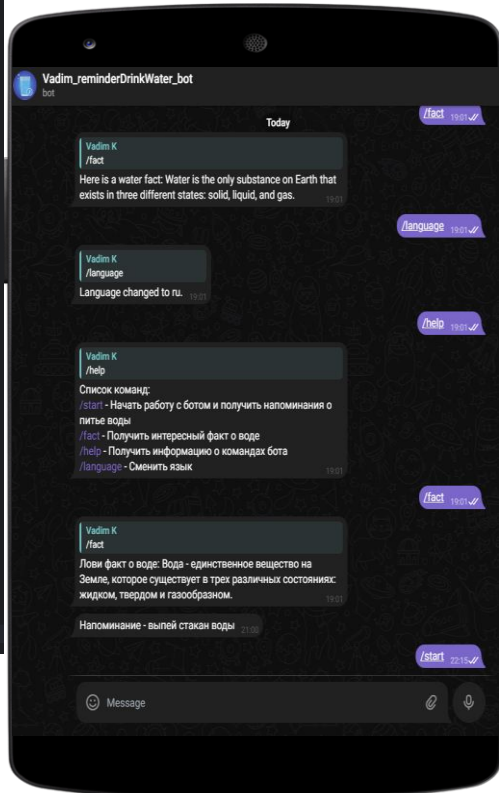
## Задачи проекта:

**Разработать Telegram-бота**, который будет напоминать пользователю о необходимости выпить стакан воды каждые 3 часа.

## Описание проекта:

**H<sub>2</sub>O Reminder Bot** — это ваш персональный ассистент по поддержанию водно-солевого баланса. В настоящее время легко забыть о простых, но важных аспектах заботы о себе. Но мой Telegram-бот решает эту проблему с помощью регулярных напоминаний о необходимости пить воду.

```
38 # Обработчик команды start
39 @bot.message_handler(commands=['start'])
40 def start_message(message):
41     user_language[message.chat.id] = 'ru' # Задаем стандартный язык
42     bot.reply_to(message, texts[user_language[message.chat.id]]['welcome'])
43     reminder_thread = threading.Thread(target=send_reminders, args=(message.chat.id,))
44     reminder_thread.start()
45
46 # Обработчик команды help
47 @bot.message_handler(commands=['help'])
48 def help_message(message):
49     bot.reply_to(message, texts[user_language[message.chat.id]]['help'])
50
51 # Обработчик команды fact
52 @bot.message_handler(commands=['fact'])
53 def fact_message(message):
54     language = user_language[message.chat.id]
55     random_fact = random.choice(texts[language]['facts'])
56     bot.reply_to(message, text=f'{texts[language]['fact']} {random_fact}')
57
58 # Обработчик команды language
59 @bot.message_handler(commands=['language'])
60 def change_language(message):
61     if user_language[message.chat.id] == 'ru':
62         user_language[message.chat.id] = 'en'
63     else:
64         user_language[message.chat.id] = 'ru'
65     bot.reply_to(message, text=f'Language changed to {user_language[message.chat.id]}')
66
67 # Функция отправки напоминаний
68 1 usage
69 def send_reminders(chat_id):
70     reminders = ["09:00", "12:00", "15:00", "18:00", "21:00"]
71     while True:
72         now = datetime.datetime.now().strftime('%H:%M')
73         if now in reminders:
74             bot.send_message(chat_id, texts[user_language[chat_id]]['reminder'])
75             time.sleep(60)
76             time.sleep(1)
77
78 bot.polling(none_stop=True)
```



# Task tracker

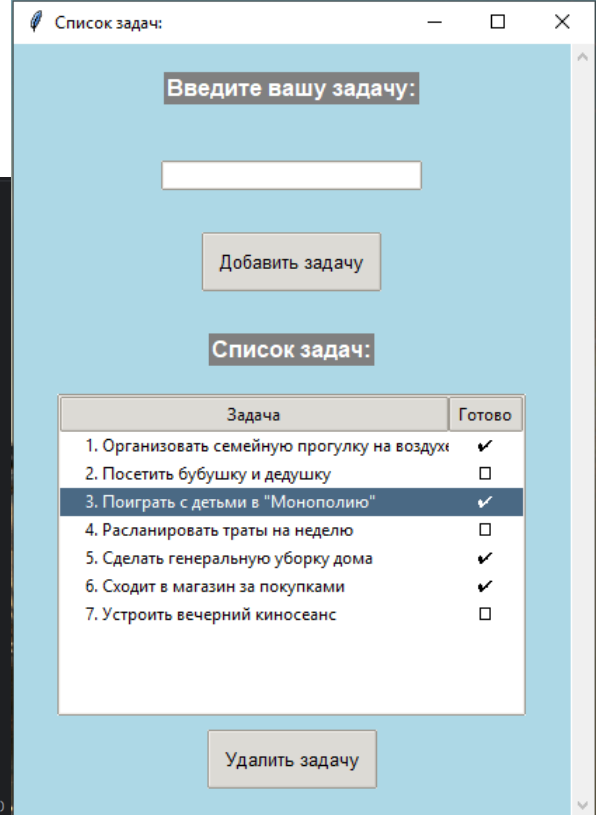
## Задачи проекта:

*Разработать приложения для управления задачами*, которое удобно использовать для организации и упорядочивания своих планов на ближайшее время.

## Возможности программы:

- добавлять задачи;
- помечать задачи выполненными;
- удалять задачи.

```
1 import tkinter as tk
2 from tkinter import ttk
3
4 usage
5 def add_task():
6     task = task_entry.get()
7     if task:
8         task_id = task_tree.insert( parent="", tk.END, text=task, values=("", "☐",))
9         task_entry.delete( first=0, tk.END)
10
11 usage
12 def check_task(event):
13     item = task_tree.selection()[0]
14     checked = task_tree.item(item, option="values")[0]
15     if checked == "☐":
16         task_tree.item(item, values=("/✓",))
17     else:
18         task_tree.item(item, values=("", "☐",))
19
20 usage
21 def delete_task():
22     selected_item = task_tree.selection()
23     for item in selected_item:
24         task_tree.delete(item)
25
26 root = tk.Tk()
27 root.title("Список задач:")
28 root.geometry("400x550+800+300")
29 root.configure(bg="lightblue")
30
31 style = ttk.Style()
32 style.theme_use("clam")
33 style.configure(style="TLabel", background="gray", foreground="white", font=("Arial", 12))
34 style.configure(style="TButton", padding=10, font=("Arial", 10))
35
36 scrollbar = tk.Scrollbar(root)
37 scrollbar.pack(side=tk.RIGHT, fill=tk.Y, expand=False) # здесь мы ставим полосу прокрутки expand = False по умолчанию
38
39 text1 = tk.Label(root, text="Введите вашу задачу:", style="TLabel", font="none 12 bold")
40 text1.pack(pady=20, padx=5)
41
42 task_entry = tk.Entry(root, width=30)
```



Я всегда открыт к новым проектам и  
возможностям! Связаться со мной можно  
следующим образом:

Telegram:



---

Разработчик на Python

Email: [vadimkrupinin@gmail.com](mailto:vadimkrupinin@gmail.com)

Телефон: +7(918)594-05-65

Telegram: @KVadim\_K

Github: