

Домашнее задание 5



Домашнее задание: «Чат-бот с памятью на JSON»

Цель: Научиться сохранять и загружать данные чат-бота между запусками с помощью формата JSON, используя встроенный модуль `json` в Python.



Общая задача

Вы уже создали простого чат-бота (в предыдущих уроках). Сейчас он **забывает всё** после закрытия программы. Ваша задача — **добавить ему долговременную память**, чтобы:

- При первом запуске он спрашивал имя и возраст.
 - При последующих запусках — **вспоминал** эти данные.
 - Умел **запоминать факты** по команде `запомни ...`.
 - Сохранял всё это в файл `memory.json`.
 - При следующем запуске **загружал** данные из этого файла.
-



Пошаговое ТЗ (делайте строго по шагам)

◆ Шаг 1. Подготовка проекта

1. Создайте новую папку, например: `chatbot_json`
2. Внутри создайте файл `bot.py`
3. Скопируйте в него **базовую структуру чат-бота** из предыдущего урока (или используйте шаблон ниже).

💡 Шаблон начала бота (минимум):

```
def main():
    print("Привет! Я чат-бот.")
    name = input("Как тебя зовут? ")
    print(f"Приятно познакомиться, {name}!")
    while True:
        msg = input("Ты: ")
        if "пока" in msg:
            print("До встречи!")
            break
        else:
            print("Не понял. Скажи 'пока', чтобы выйти.")

if __name__ == "__main__":
    main()
```

◆ Шаг 2. Импортируйте модуль `json`

В самом верху файла `bot.py` добавьте:

```
import json
```

✅ Это встроенная библиотека — ничего устанавливать не нужно.

◆ Шаг 3. Реализуйте функцию `save_memory(data, filename="memory.json")`

Добавьте эту функцию **вне** `main()`:

```
def save_memory(data, filename="memory.json"):
    with open(filename, "w", encoding="utf-8") as f:
        json.dump(data, f, ensure_ascii=False, indent=4)
    print("✅ Бот сохранил всё в блокнот.")
```

🔍 Пояснение параметров:

- `ensure_ascii=False` — чтобы кириллица не превращалась в `\\u0410...`
- `indent=4` — чтобы файл был красивым и читаемым

◆ Шаг 4. Реализуйте функцию

`load_memory(filename="memory.json")`

Добавьте эту функцию рядом с предыдущей:

```
def load_memory(filename="memory.json"):
    try:
        with open(filename, "r", encoding="utf-8") as f:
            return json.load(f)
    except FileNotFoundError:
        print("Бот: У меня ещё нет записей о тебе. Начнём с чистого листа!")
        return {}
    except json.JSONDecodeError:
        print("Бот: Мой блокнот испортился! Начнём заново.")
        return {}
```

⚠ Обязательно обрабатывайте обе ошибки! Иначе программа упадёт при первом запуске или при повреждении файла.

◆ Шаг 5. Интегрируйте память в `main()`

1. В начале `main()` **загрузите память**:

```
user_info = load_memory()
```

2. Если имя ещё не сохранено — спросите его и сохраните:

```
if "name" not in user_info:
    user_info["name"] = input("Бот: Как тебя зовут? ").strip()
    print(f"Привет, {user_info['name']}!")
```

```
else:
    print(f"С возвращением, {user_info['name']}! Я тебя помню.")
```

3. Аналогично для возраста (опционально, но рекомендуется).

◆ Шаг 6. Добавьте команду «запомни»

Реализуйте обработку фразы `запомни ...`:

```
msg = input("Ты: ").strip()
if msg.startswith("запомни "):
    fact = msg[8:] # убираем "запомни "
    if "facts" not in user_info:
        user_info["facts"] = []
    user_info["facts"].append(fact)
    print(f"Запомнил: {fact}")
    continue
```

💡 Проверяйте, что fact не пустой.

◆ Шаг 7. Сохраняйте данные при выходе

Перед тем как завершить программу (после `break`), **обязательно сохраните**:

```
if "пока" in msg.lower():
    save_memory(user_info)
    print(f"До встречи, {user_info.get('name', 'друг')}!")
    break
```

❗ Без этого — данные не сохранятся!

◆ Шаг 8. Проверка работоспособности

Сделайте **три теста**:

1. **Первый запуск:**

- Бот спрашивает имя → вы вводите «Анна»
- Говорите: `запомни люблю кофе`
- Пишете `пока` → бот прощается и сохраняет

2. Проверка файла:

- Откройте `memory.json` в любом текстовом редакторе
- Убедитесь, что там есть:

```
{
  "name": "Анна",
  "facts": ["люблю кофе"]
}
```

3. Второй запуск:

- Бот должен сказать: `С возвращением, Анна!`
- Скажите: `что ты знаешь?` → бот отвечает: `Я знаю: люблю кофе`

🔧 Если что-то не работает — проверьте:

- Вызов `save_memory()` перед `break`
- Наличие `encoding="utf-8"` и `ensure_ascii=False`
- Правильность отступов и скобок в JSON

★ Дополнительное задание (по желанию)

Реализуйте **автосохранение каждые 5 сообщений**:

```
message_count = 0
while True:
    msg = input("Ты: ")
    message_count += 1
    # ... обработка ...
    if message_count % 5 == 0:
```

```
save_memory(user_info)
print("(Автосохранение)")
```



Что вы должны понимать после выполнения

- Как работает модуль `json` в Python.
- Разница между `dump / load` (файлы) и `dumps / loads` (строки).
- Зачем нужен `with open(...) as f`.
- Почему важно обрабатывать `FileNotFoundError` и `JSONDecodeError`.
- Как избежать проблем с кириллицей (`ensure_ascii=False`).
- Как сделать программу **устойчивой к ошибкам**.



Сдача работы

1. Отправьте файл `bot.py`
2. Приложите скриншот содержимого `memory.json` после двух запусков
3. Напишите 2–3 предложения: что было сложно, что получилось легко

✓ Совет: Не копируйте готовый код целиком из урока — набирайте сами. Это закрепит знания!

Удачи! Ваш бот теперь **помнит**. А это уже почти искусственный интеллект 🤖