

Открытие и чтение файла, запись в файл

Николай Свиридов



Проверка связи



Отправьте, пожалуйста, смайлик в чат

Если у вас все отлично со связью :-)

Если у вас есть какие-то проблемы со связью :-(



Если у вас нет звука:

- убедитесь, что на вашем устройстве и на колонках включен звук
- обновите страницу вебинара (или закройте страницу и заново присоединитесь к вебинару)
- откройте вебинар в другом браузере
- перезагрузите компьютер (ноутбук) и заново попытайтесь зайти



Поставьте “+”, если меня видно и слышно

Николай Свиридов

О спикере:

- Backend-разработчик
- IT-блогер



План занятия

1. Работа с файлами в Python
2. Основные ошибки при работе с файлами

Работа с файлами в Python

Files are everywhere

- хранение данных программы
- файлы конфигурации
- текстовые файлы
- картинки



Что такое файл?

Файл — именованная область данных на диске.

track.mp3

diplom.doc

100010100101010101010101010101100101010101010101010101010101101010101

Что такое файл?

Файл — именованная область данных на диске.

track.mp3

diplom.doc

10001010010101010101010101010101100101010101010101010101010101101010101

File object в Python — объект, позволяющий работать с данными, как с файлом.

[Глоссарий](#)

Модель работы с любыми ресурсами

Шаг	Команда
открыть	<code>f = open(имя файла, режим)</code>

[Ссылка на пример «Открытие и чтение файла»](#)

Модель работы с любыми ресурсами

Шаг	Команда
открыть	<code>f = open(имя файла, режим)</code>
основная логика: прочитать данные, записать данные	<code>f.write(...), f.read()</code>

[Ссылка на пример «Открытие и чтение файла»](#)

Модель работы с любыми ресурсами

Шаг	Команда
открыть	<code>f = open(имя файла, режим)</code>
основная логика: прочитать данные, записать данные	<code>f.write(...), f.read()</code>
закрыть	<code>f.close()</code>

[Ссылка на пример «Открытие и чтение файла»](#)

Упрощаем работу с ресурсами

С помощью конструкции **with** (менеджер контекста) Python сам проследит за закрытием файла.

Менеджеры контекста позволяют выделять и освобождать ресурсы строго по необходимости.

```
with open('test.txt') as f:
```

[Ссылка на пример «Упрощение работы с ресурсами»](#)

Методы объекта File для чтения

Методы	Значение
read	чтение всего файла целиком
readline	построчное чтение
readlines	чтение всех строк в список

Методы объекта File для чтения

Методы	Значение
read	чтение всего файла целиком
readline	построчное чтение
readlines	чтение всех строк в список

[Документация](#)

[Дополнительный материал по работе с файлами в Python](#)

```
with open('data.txt', 'r') as f:
    for line in f:
        print(f)
```

[Ссылка на пример «Методы чтения файла»](#)

Режимы работы с файлом

Режимы доступа

'r' — чтение (по-умолчанию)

'w' — запись

'a' — запись в конец файла

Режимы работы с файлом

Режимы доступа

'r' — чтение (по-умолчанию)

'w' — запись

'a' — запись в конец файла

Режимы чтения/записи

'b' — двоичный режим (работаем с байтами)

't' — текстовый режим (работаем со строками, по-умолчанию)

[Ссылка на пример «Запись в файл и режимы работы»](#)

Режимы работы с файлом

Режимы доступа

'r' — чтение (по-умолчанию)

'w' — запись

'a' — запись в конец файла

Формат записи:

```
open(Имя файла, режим)
```

Пример:

```
open('test.txt', 'w')
```

Режимы чтения/записи

'b' — двоичный режим (работаем с байтами)

't' — текстовый режим (работаем со строками, по-умолчанию)

[Ссылка на пример «Запись в файл и режимы работы»](#)

Путь к файлу в иерархии

1. Абсолютный (от корневого каталога)
2. Относительный (от текущего каталога)

Путь к файлу в иерархии

1. Абсолютный (от корневого каталога)
 2. Относительный (от текущего каталога)
- **Нужен временный файл или конкретный путь не задан?**

Путь к файлу в иерархии

1. Абсолютный (от корневого каталога)
 2. Относительный (от текущего каталога)
- **Нужен временный файл или конкретный путь не задан?**
Относительные пути

Путь к файлу в иерархии

1. Абсолютный (от корневого каталога)
 2. Относительный (от текущего каталога)
- **Нужен временный файл или конкретный путь не задан?**
Относительные пути
 - **Работаете с конкретной директорией?**

Путь к файлу в иерархии

1. Абсолютный (от корневого каталога)
2. Относительный (от текущего каталога)
 - **Нужен временный файл или конкретный путь не задан?**
Относительные пути
 - **Работаете с конкретной директорией?**
Абсолютные пути

Путь к файлу в иерархии

1. Абсолютный (от корневого каталога)
 2. Относительный (от текущего каталога)
- **Нужен временный файл или конкретный путь не задан?**
Относительные пути
 - **Работаете с конкретной директорией?**
Абсолютные пути
 - **Не забывайте про ОС**
Windows - C:\Users\romanova\
Linux - /home/users/romanova...

Пример «Ошибки при работе с файлами»

Методы для работы с путями

- `import os`
- `os.getcwd()` — получить абсолютный путь текущего каталога
- `os.path.join(path, *paths)` — построение платформоспецифичного пути
- `os.path.dirname(path)` — абсолютный путь до каталога
- `os.path.basename(path)` — имя файла

Интерфейсы операционной системы

Манипуляции с путями

Кодировки

- Для компьютера текст (как и любые другие данные) — это набор байтов.
- Мы (люди) хотим работать со строками, байты нам читать сложно.

[Пример «Кодировки»](#)

[Обзор Unicode в официальной документации](#)

Кодировки

- Для компьютера текст (как и любые другие данные) — это набор байтов.
- Мы (люди) хотим работать со строками, байты нам читать сложно.
- **Кодировка** — это способ сопоставления байтов символам.
- По умолчанию в Python при работе с файлами используется **utf-8**.
- Для изменения кодировки в функции `open` есть параметр **encoding**.

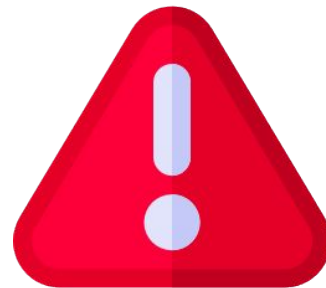
Пример «Кодировки»

Обзор Unicode в официальной документации

Основные ошибки при работе с файлами

Что такое исключение?

Исключение — событие, когда в программном коде допускается ошибка.
Может возбуждаться и перехватываться самим программным кодом.



Обработка исключений. Try/except

Пример кода:

```
try:  
    100 / 0  
except ZeroDivisionError:  
    print("На ноль делить нельзя")
```

Домашнее задание

Задача

Необходимо написать программу для кулинарной книги. Список рецептов должен храниться в отдельном файле в следующем формате:

```
Название блюда
Количество ингредиентов в блюде
Название ингредиента | Количество | Единица измерения
Название ингредиента | Количество | Единица измерения
...
```

- В одном файле может быть произвольное количество блюд.
- Читать список рецептов нужно из этого файла.
- Соблюдайте кодстайл, разбивайте новую логику на функции и не используйте глобальных переменных.

Задание 1

Создайте словарь на основе данных из файла.

```
cook_book = {
    'Омлет': [
        {'ingredient_name': 'Яйцо', 'quantity': 2, 'measure': 'шт.'},
        {'ingredient_name': 'Молоко', 'quantity': 100, 'measure': 'мл'},
        {'ingredient_name': 'Помидор', 'quantity': 2, 'measure': 'шт'}
    ],
    'Утка по-пекински': [
        {'ingredient_name': 'Утка', 'quantity': 1, 'measure': 'шт'},
        {'ingredient_name': 'Вода', 'quantity': 2, 'measure': 'л'},
        {'ingredient_name': 'Мед', 'quantity': 3, 'measure': 'ст.л'},
        {'ingredient_name': 'Соевый соус', 'quantity': 60, 'measure': 'мл'}
    ],
    'Запеченный картофель': [
        {'ingredient_name': 'Картофель', 'quantity': 1, 'measure': 'кг'},
        {'ingredient_name': 'Чеснок', 'quantity': 3, 'measure': 'зубч'},
        {'ingredient_name': 'Сыр гауда', 'quantity': 100, 'measure': 'г'},
    ]
}
```


Задание 2

Нужно написать функцию, которая на вход принимает список блюд из `cook_book` и количество персон для кого мы будем готовить.

```
get_shop_list_by_dishes(dishes, person_count)
```

На выходе мы должны получить словарь с названием ингредиентов и его количества для блюда.

```
{
  'Картофель': {'measure': 'кг', 'quantity': 2},
  'Молоко': {'measure': 'мл', 'quantity': 200},
  'Помидор': {'measure': 'шт', 'quantity': 4},
  'Сыр гауда': {'measure': 'г', 'quantity': 200},
  'Яйцо': {'measure': 'шт', 'quantity': 4},
  'Чеснок': {'measure': 'зубч', 'quantity': 6}
}
```

Задание 3

В папке лежит некоторое количество файлов. Считайте, что их количество и имена вам заранее известны.

Необходимо объединить их в один по следующим правилам:

- Содержимое исходных файлов в результирующем файле должно быть отсортировано по количеству строк в них (то есть первым нужно записать файл с наименьшим количеством строк, а последним - с наибольшим)
- Содержимое файла должно предваряться служебной информацией на 2-х строках: имя файла и количество строк в нем



Ваши вопросы?