

Погружение в Python

Урок 7

Файлы и файловая система





Содержание урока








План курса





Что будет на уроке сегодня

-  Разберёмся в особенностях работы с файлами и каталогами в Python
-  Изучим функцию `open` для работы с содержимым файла
-  Узнаем о возможностях стандартной библиотеки для работы с файлами и каталогами





Файлы





Функция open()

В Python для получения доступа файлу используют функцию open()

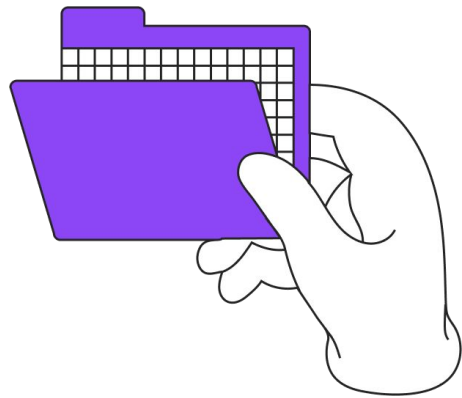
```
open(file, mode='r', buffering=-1,  
encoding=None,  
      errors=None, newline=None, closefd=True,  
opener=None)
```





Режимы работы с файлами

- ✓ 'r' — открыть для чтения (по умолчанию)
- ✓ 'w' — открыть для записи, предварительно очистив файл
- ✓ 'x' — открыть для эксклюзивного создания. Вернёт ошибку, если файл уже существует
- ✓ 'a' — открыть для записи в конец файла, если он существует
- ✓ 'b' — двоичный режим
- ✓ 't' — текстовый режим (по умолчанию)
- ✓ '+' — открыты для обновления (чтение и запись)





Метод `close()`

✓ `f.close()`

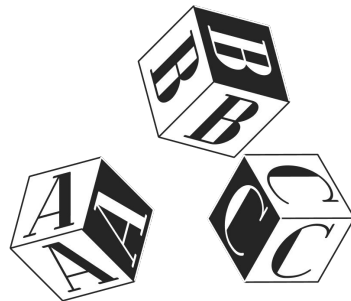
Если в коде отсутствует метод `close()`, то даже при успешном завершении программы не гарантируется сохранение всех данных в файле





Прочие необязательные параметры функции open

- ✓ `buffering` — определяет режим буферизации
- ✓ `errors` — используется только в текстовом режиме и определяет поведение в случае ошибок кодирования или декодирования
- ✓ `newline` — отвечает за преобразование окончания строки
- ✓ `closefd` — указывает оставлять ли файловый дескриптор открытым при закрытии файла
- ✓ `opener` — позволяет передать пользовательскую функцию для открытия файла

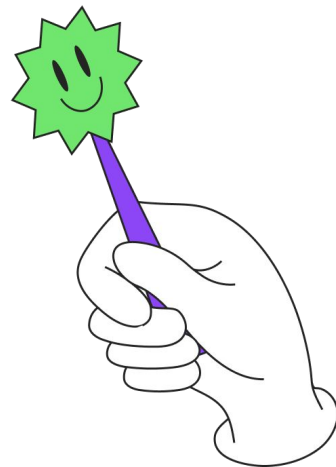




Менеджер контекста with open

```
with open('text_data.txt', 'r+', encoding='utf-8') as f:  
    print(list(f))
```

- ✓ with гарантирует закрытие файла и сохранение информации





Чтение файла



`list(f)`

Чтение в список



`res = f.read()`

Чтение методом read



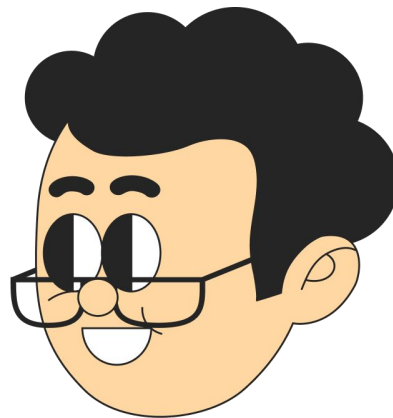
`res = f.readline()`

Чтение методом readline



`for line in f:`

Чтение циклом for





Запись и добавление в файл



res = f.write(text)

Запись методом write



f.writelines('\n'.join(text))

Запись методом writelines



print(text, file=f)

print в файл



w

создаём новый пустой файл для записи.
Если файл существует, открываем его для записи и удаляем данные, которые в нём хранились



x

создаём новый пустой файл для записи.
Если файл существует, вызываем ошибку



a

открываем существующий файл для записи в конец, добавления данных. Если файл не существует, создаём новый файл и записываем в него



Методы перемещения в файле



f.tell()

Метод `tell` возвращает текущую позицию в файле



seek(offset, whence=0)

`offset` — смещение относительно опорной точки,
`whence` — способ выбора опорной точки.

- ✓ `whence=0` — отсчёт от начала файла
- ✓ `whence=1` — отсчёт от текущей позиции в файле
- ✓ `whence=2` — отсчёт от конца файла



truncate(size=None)

Метод изменяет размер файла. Если не передать значение в параметр `size` будет удалена часть файла от текущей позиции до конца



Перед вами несколько строк кода. Что будет храниться в файле после завершения работы программы? И что будет выведено на печать?

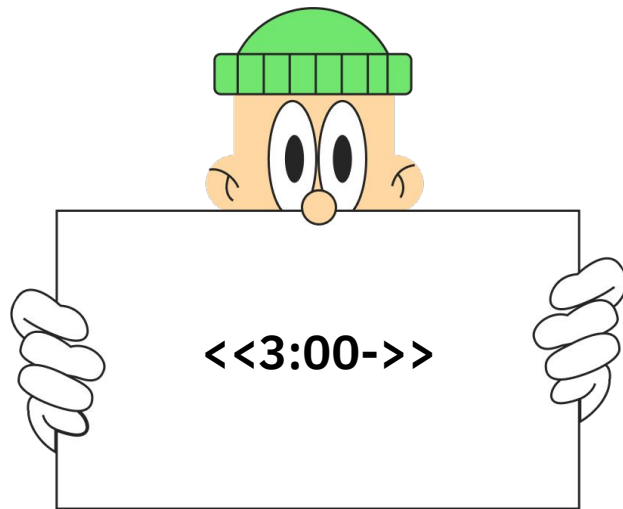
У вас 3 минуты.





Модули

```
start = 10
stop = 100
with open('data.bin', 'bw+') as f:
    for i in range(start, stop + 1):
        f.write(str(i).encode('utf-8'))
        if i % 3 == 0:
            f.seek(-2, 1)
    f.truncate(stop)
    f.seek(0)
    res = f.read(start)
    print(res.decode('utf-8'))
```





Файловая система

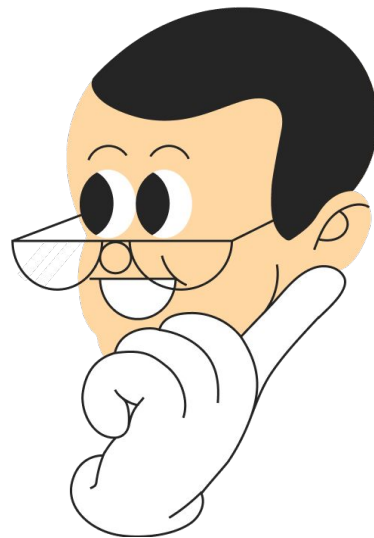




Работа с каталогами

Рассмотрим разные варианты решения задачи на примерах

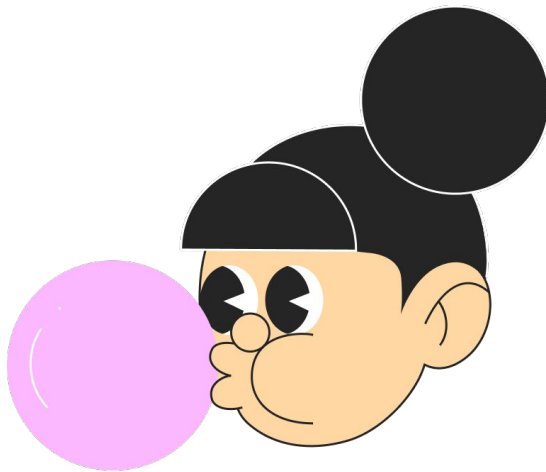
- ✓ Определить текущий каталог
- ✓ Создать новый каталог
- ✓ Удалить существующий каталог
- ✓ Формирование пути





Чтение данных о каталогах

- ✓ Получение списка каталогов и файлов
- ✓ Проверка на директорию, файл и ссылку
- ✓ Обход папок через `os.walk()`





Работа с файлами

Рассмотрим базовые операции по работе с файлами как с отдельными объектами:

- ✓ Переименование файлов
- ✓ Перемещение файлов
- ✓ Копирование файлов
- ✓ Удаление файлов





Перед вами несколько строчек кода.
Какие каталоги и файлы будут созданы
после его выполнения?

У вас 3 минуты.

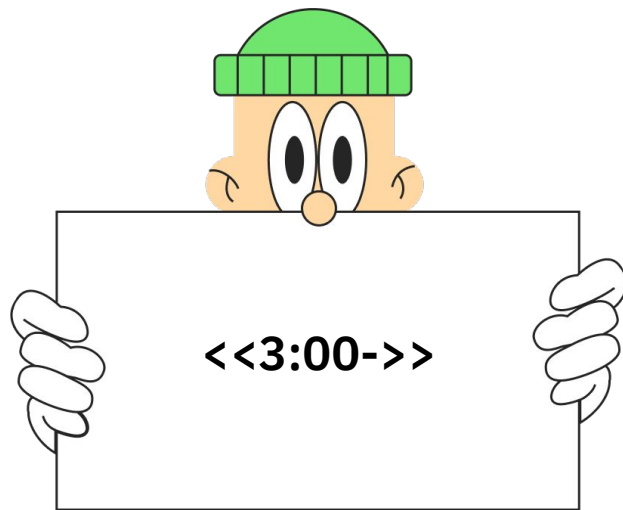




Модули

```
import os
import shutil
from pathlib import Path

for i in range(10):
    with open(f'file_{i}.txt', 'w',
encoding='utf-8') as f:
        f.write('Hello world!')
os.mkdir('new_dir')
for i in range(2, 10, 2):
    f = Path(f'file_{i}.txt')
    f.replace('new_dir' / f)
shutil.copytree('new_dir', Path.cwd() /
'dir_new')
```








Итоги занятия





На этой лекции мы

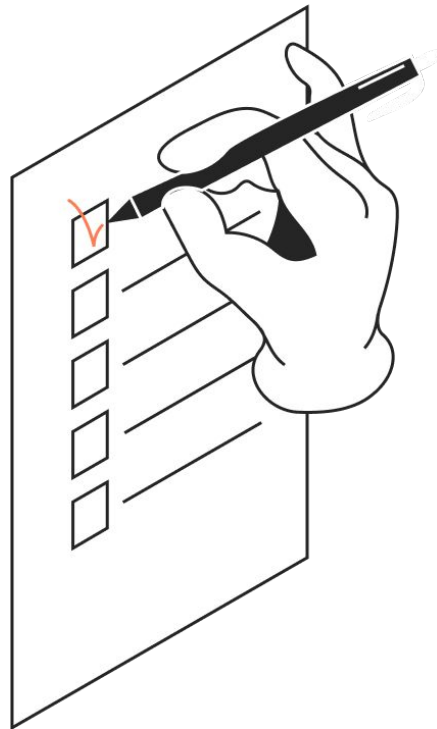
-  Разобрались в особенностях работы с файлами и каталогами в Python
-  Изучили функцию `open` для работы с содержимым файла
-  Узнали о возможностях стандартной библиотеки для работы с файлами и каталогами





Задание

Загляните в документацию к Python и изучите особенности и нюансы работы с файлами и каталогами в рассматриваемых на уроке модулях.





Спасибо за внимание