

Работа с файлами

Последовательность работы

- 1) Открытие файла с помощью метода `open()`
- 2) Чтение файла с помощью метода `read()` или запись в файл посредством метода `write()`
- 3) Заккрытие файла методом `close()`

Открытие и закрытие файла

Чтобы начать работу с файлом, его надо открыть с помощью функции `open()`, которая имеет следующее формальное определение:

```
open(путь до файла, режим)
```

Режимы работы с файлом (mode)

- r (Чтение, если файл не существует, генерируется ошибка)
- w (Перезапись, если файл не существует, он создается)
- a (Дозапись)
- b (Работа с бинарными файлами, rb, rw, ra)
- Так же существуют r+, w+, a+

Пример

открытие текстового файла на запись

```
myfile = open("hello.txt", "w")
```

вызов функции say_hello

```
myfile = open("image.png", "rb")
```

```
say_hello()
```

```
say_hello()
```

Конструкция with

Эта конструкция определяет для открытого файла переменную `myfile` и выполняет набор инструкций. После их выполнения файл автоматически закрывается.

```
with open(file, mode) as myfile:
```

```
    #инструкции
```

Конструкция with

Пример:

```
with open("hello.txt", "w") as myfile:  
    print("Работа с файлом myfile")
```

Запись в текстовый файл

Чтобы открыть текстовый файл на запись, необходимо применить режим **w** (перезапись), **a**(дозапись) или **r+/w+** (чтение и запись)

Затем для записи применяется метод **write(str)**, в который передается записываемая строка

Запись в текстовый файл

Приемр:

```
with open("hello.txt", "w") as file:  
    file.write("hello world")  
  
print("Файл записан")
```

Запись в текстовый файл

Для записи списка строк имеется метод `writelines()`. В качестве параметра он принимает список строк:

```
lines = ["Hello Word\n", "Hello Work\n", "Hello World\n"]
with open("hello2.txt", "a") as file:
    file.writelines(lines)

print("Список строк записан в файл")
```

Чтение файла

Для чтения файла он открывается с режимом r (Read), и затем мы можем считать его содержимое различными методами:

- `readline()`: считывает одну строку из файла
- `read()`: считывает все содержимое файла в одну строку
- `readlines()`: считывает все строки файла в список

Чтение из файла

Например, считаем выше записанный файл
построчно:

```
with open("hello.txt", "r") as file:  
    for line in file:  
        print(line)
```

Чтение из файла

Теперь явным образом вызовем метод `readline()` для чтения отдельных строк:

```
with open("hello.txt", "r") as file:
```

```
    str1 = file.readline()    # считываем первую строку
```

```
    print(str1, end="")
```

```
    str2 = file.readline()    # считываем вторую строку
```

```
    print(str2)
```

Чтение из файла

```
with open("hello.txt", "r") as file:
```

```
    line = file.readline()
```

```
    while line:
```

```
        print(line, end="")
```

```
        line = file.readline()
```

Чтение из файла

Если файл небольшой, то его можно разом считать с помощью метода `read()`:

```
with open("hello.txt", "r") as file:  
    content = file.read()  
    print(content)
```

Чтение из файла

И также применим метод `readlines()` для считывания всего файла в список строк:

```
with open("hello.txt", "r") as file:  
    contents = file.readlines()  
    str1 = contents[0]  
    str2 = contents[1]  
    print(str1, end="")  
    print(str2)
```


Чтение из файла

При чтении файла мы можем столкнуться с тем, что его кодировка не совпадает с ASCII. В этом случае мы явным образом можем указать кодировку с помощью параметра `encoding`:

```
filename = "hello.txt"  
with open(filename, encoding="utf8") as file:  
    text = file.read()
```

Задание

Создать приложение с интерактивным меню, предоставить пользователю возможность просматривать содержимое файла, добавлять данные файл, перезаписывать содержимое

Модуль OS и работа с файловой системой

Что это

Ряд возможностей по работе с каталогами и файлами предоставляет встроенный модуль `os`. Основные функции:

- `mkdir()`: создает новую папку
- `rmdir()`: удаляет папку
- `rename()`: переименовывает файл
- `remove()`: удаляет файл

Создание и удаление папки

Для создания папки применяется функция `mkdir()`, в которую передается путь к создаваемой папке:

```
import os
# путь относительно текущего скрипта
os.mkdir("hello")
# абсолютный путь
os.mkdir("c://somedir")
os.mkdir("c://somedir/hello")
```

Создание и удаление папки

Для удаления папки используется функция `rmdir()`, в которую передается путь к удаляемой папке:

```
import os
# путь относительно текущего скрипта
os.rmdir("hello")
# абсолютный путь
os.rmdir("c://somedir/hello")
```

Переименование файла

Для переименования вызывается функция `rename(source, target)`, первый параметр которой - путь к исходному файлу, а второй - новое имя файла.

```
import os  
os.rename("C://SomeDir/somefile.txt", "C://SomeDir/hello.txt")
```

Удаление файла

Для удаления вызывается функция `remove()`, в которую передается путь к файлу:

```
import os  
  
os.remove("C://SomeDir/hello.txt")
```


Существование файла

Уже до открытия файла проверить, существует ли он или нет с помощью метода `os.path.exists(path)`.

```
filename = input("Введите путь к файлу: ")  
if os.path.exists(filename):  
    print("Указанный файл существует")  
else:  
    print("Файл не существует")
```