

ФАЙЛЫ

РАБОТА С ФАЙЛАМИ

ФОРМАТ CSV

РАБОТА С CSV-ФАЙЛАМИ

Гаврилов Денис Андреевич, преподаватель кафедры СИ ФИТ НГУ

Открытие файла.

Чтение из файла.

Запись в файл.

Дополнительные методы
работы с файлами.

Закрытие файла.

РАБОТА С ФАЙЛАМИ

ОТКРЫТИЕ ФАЙЛА

Python предоставляет возможность работать не только с данными, вводимыми вручную, но и с записанными в форме цифрового файла:

```
f = open(<имя файла>, [<Параметры>])
```

```
f = open("fruit.txt", "w")
```

Открытие файла под названием «fruit.txt»

```
f.write("яблоки\n")
```

с целью записи данных в него

```
f.write("груши\n")
```

```
f.write("персики\n")
```

запись данных в файл

```
f.write("сливы\n")
```

```
f.close()
```

Заккрытие файла

ОТКРЫТИЕ ФАЙЛА

Важно: имя файла указывается вместе с *полным* или *относительным* адресом к этому файлу.

Пример имени файла с полным адресом:

```
"C:\\Users\\Денис\\Desktop\\fruit.txt"
```

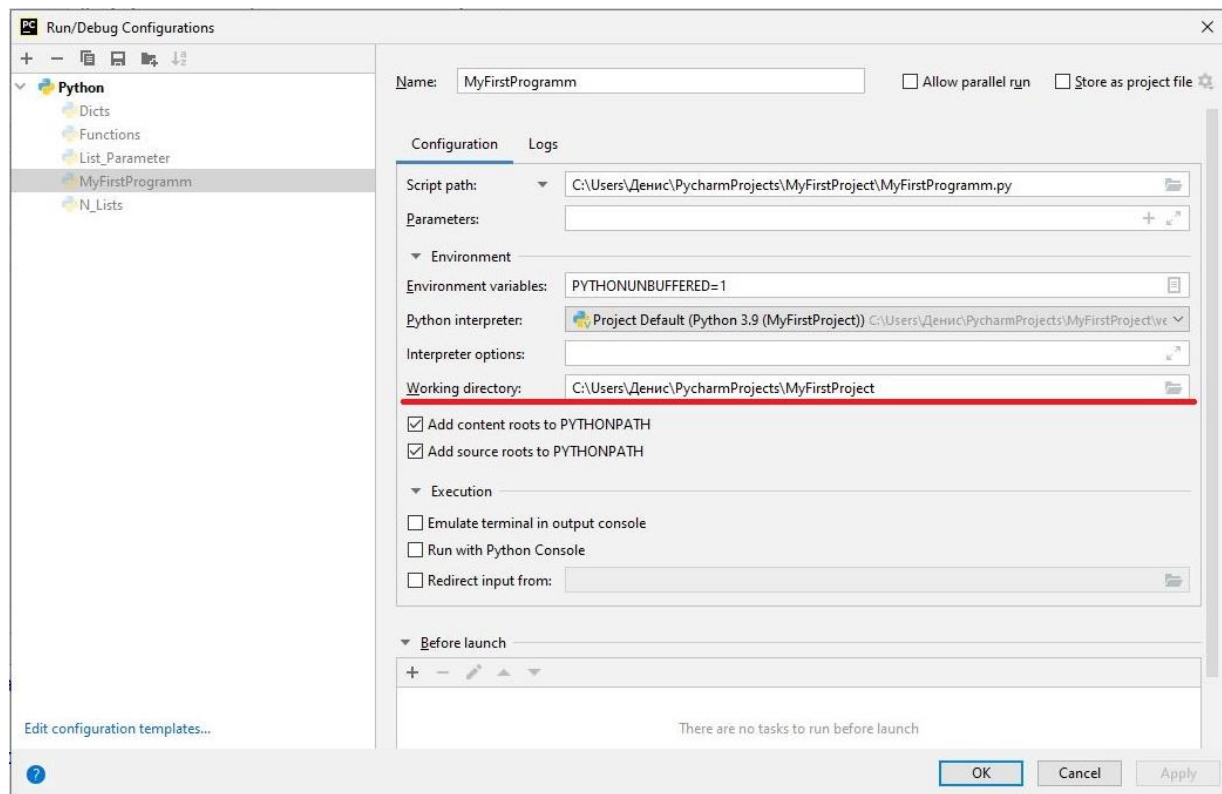
Префиксом для относительного адреса служит, по умолчанию, адрес рабочей директории (папки) текущего проекта. То есть, создав файл при помощи кода:

```
f = open("TestFiles\\fruit.txt", "w")
```

Файл будет открыт (или создан) в директории *C:\\Users\\Denis\\PycharmProjects\\ProjectWithFiles\\TestFiles*

ОТКРЫТИЕ ФАЙЛА

Настроить рабочую директорию (папку) проекта в PyCharm можно по пути: *Run / Edit Configurations...*



ЗАПИСЬ В ФАЙЛ

Записать данные в файл можно при помощи метода `write()`:

```
f = open("fruit.txt", "w")
```

```
f.write("яблоки\n")
```

```
f.write("груши\n")
```

```
f.write("персики\n")
```

```
f.write("сливы\n")
```

```
f.close()
```

Открытие файла «fruit.txt»

с целью записи данных в него

запись данных в файл

Заккрытие файла

Результат (содержимое «fruit.txt»):

яблоки

груши

персики

сливы

ЗАПИСЬ В ФАЙЛ

Эквивалентного результата можно достичь при использовании функции *print()*:

```
f = open("fruit.txt", "w")  
  
print("яблоки", file=f)  
print("груши", file=f)  
print("персики", file=f)  
print("сливы", file=f)  
  
f.close()
```

Открытие файла «fruit.txt»

с целью записи данных в него

запись данных в файл

Заккрытие файла

Результат (содержимое «fruit.txt»):

```
яблоки  
груши  
персики  
сливы
```

ЧТЕНИЕ ИЗ ФАЙЛА

Для чтения из файла, его следует открыть в подходящем режиме ("r"), после чего воспользоваться методом *read()*:

```
f = open("fruit.txt", "r")
```

```
lines = f.read()  
print(lines)
```

```
f.close()
```

Открытие файла «fruit.txt»
с целью чтения данных из него

Чтение файла целиком
и вывод его на печать

Закрытие файла

Файл «fruit.txt»:

```
яблоки  
груши  
персики  
сливы
```


ЧТЕНИЕ ИЗ ФАЙЛА

При помощи метода *read()* можно прочитать конкретное количество символов, задав его в качестве параметра;

Пустое значение эквивалентно прочтению всех ещё не прочитанных (пройденных) данных файла:

```
f = open("Hello.txt", "r")
```

```
# Открытие файла под названием «fruit.txt»  
# с целью чтения данных из него
```

```
short_text = f.read(3)  
print(short_text)
```

```
# Вывод: Hel
```

```
long_text = f.read()  
print(long_text)
```

```
# Вывод: lo, World!
```

```
f.close()
```

```
# Заккрытие файла
```

Файл «Hello.txt»:

```
Hello, World!
```

ЧТЕНИЕ ИЗ ФАЙЛА ПОСТРОЧНО

Чтение файла построчно с использованием цикла:

```
f = open("fruit.txt", "r")  
  
for line in f:  
    print(line)  
  
f.close()
```

Чтение файла построчно с использованием метода *readlines()*:

```
f = open("fruit.txt", "r")  
  
lines = f.readlines()  
print(lines)  
f.close()
```

Файл «fruit.txt»:

```
яблоки  
груши  
персики  
сливы
```

ДОПОЛНИТЕЛЬНЫЕ МЕТОДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ

Различные режимы работы с файлом:

Режим	Функционал
'r'	открытие на чтение (является значением по умолчанию).
'w'	открытие на запись; если файл с таким именем уже существовал, его содержимое удаляется; если файла с таким именем не существует, создается новый.
'x'	открытие на запись при условии, что файла с таким именем изначально не существует; иначе исключение.
'a'	открытие на до-запись; информация добавляется в конец файла.
'b'	открытие в двоичном режиме.
't'	открытие в текстовом режиме (является значением по умолчанию).
'+'	открытие на чтение и запись

ДОПОЛНИТЕЛЬНЫЕ МЕТОДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ

Пример с открытием файла на до-запись:

```
f = open("fruit.txt", "a")

for i in range(3):
    name=input("Введите новое название фрукта:")
    f.write(name+"\n")

f.close()

f = open("fruit.txt", "r")
lines=f.read()
print(lines)

f.close()
```

Открытие файла под названием «fruit.txt»
с целью до-записи данных в него

Ввод 3-х дополнительных названий фруктов
Например: апельсины | киви | гранаты

Закрытие файла

Открытие файла под названием «fruit.txt»
с целью чтения данных из него

Закрытие файла

ДОПОЛНИТЕЛЬНЫЕ МЕТОДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ

Пример с открытием файла на до-запись:

```
f = open("fruit.txt", "a")

for i in range(3):
    name=input("Введите новое название фрукта:")
    f.write(name+"\n")

f.close()

f = open("fruit.txt", "r")
lines=f.read()
print(lines)

f.close()
```

Файл «fruit.txt» до:

яблоки
груши
персики
сливы

Файл «fruit.txt» после:

яблоки
груши
персики
сливы
апельсины
киви
гранаты

ДОПОЛНИТЕЛЬНЫЕ МЕТОДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ

Метод *tell()* возвращает текущую позицию «курсора» в файле. Например, если вы в сумме считали восемь символов, то «курсор» будет установлен в позицию 8:

```
f = open("Hello.txt", "r")
```

```
short_text = f.read(3)  
print(short_text, f.tell())
```

```
long_text = f.read(5)  
print(long_text, f.tell())
```

```
f.close()
```

Открытие файла под названием «fruit.txt»
с целью чтения данных из него

Чтение 3-х символов
Вывод: 'Hel' 3

Чтение 5-ти символов
Вывод: 'lo, W' 8

Заккрытие файла

Файл «Hello.txt»:

Hello, World!

ДОПОЛНИТЕЛЬНЫЕ МЕТОДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ

Метод *seek(n)*, в свою очередь, позволяет явно выставить позицию «курсора» в файле на n-ный СИМВОЛ.

Важно: нумерация позиций «курсора» начинается с 0 (нуля):

```
f = open("Hello.txt", "r")
```

```
f.seek(7)
```

```
short_text = f.read()
```

```
print(short_text)
```

```
f.close()
```

Открытие файла под названием «fruit.txt»

с целью чтения данных из него

Выставление курсора в позицию 7

Вывод: 'World!'

Заккрытие файла

Файл «Hello.txt»:

```
Hello, World!
```

ЗАКРЫТИЕ ФАЙЛА

Важно: после завершения работы с файлом его необходимо закрыть.

Помимо «ручного» закрытия с помощью метода *close()*, можно воспользоваться конструкцией *with-as*;

Данная конструкция гарантирует, что после выполнения всех операторов внутри её блока, файл будет закрыт автоматически (без необходимости прописывать метод *close()*).

```
with open("Hello.txt", "w") as g:  
    g.write("Hello, ")  
    g.write("World!")
```

Блок записи в файл

Закрытие файла происходит автоматически

```
with open("Hello.txt", "r") as f:  
    text = f.read()  
    print(text)
```

Блок чтения из файла

Вывод: 'Hello, World!'

Закрытие файла происходит автоматически

Формат файлов CSV.

Примеры.

Чтение CSV-файла.

Запись CSV-файла.

Дополнительные методы
работы с csv-файлами.

ФАЙЛЫ ФОРМАТА CSV

ФОРМАТ CSV

CSV (Comma-Separated Values — значения, разделённые запятыми) — текстовый формат файлов, предназначенный для представления табличных данных. Строка таблицы соответствует строке текста, которая содержит одно или несколько полей, разделенных запятыми:

```
"месяц год","min","max"  
"январь 2021",-34,-20  
"февраль 2021",-16,-8  
"март 2021",-12,10  
"апрель 2021",-8,15  
"май 2021",-3,22  
"июнь 2021",8,25  
"июль 2021",18,32  
"август 2021",15,24  
"сентябрь 2021",7,21
```

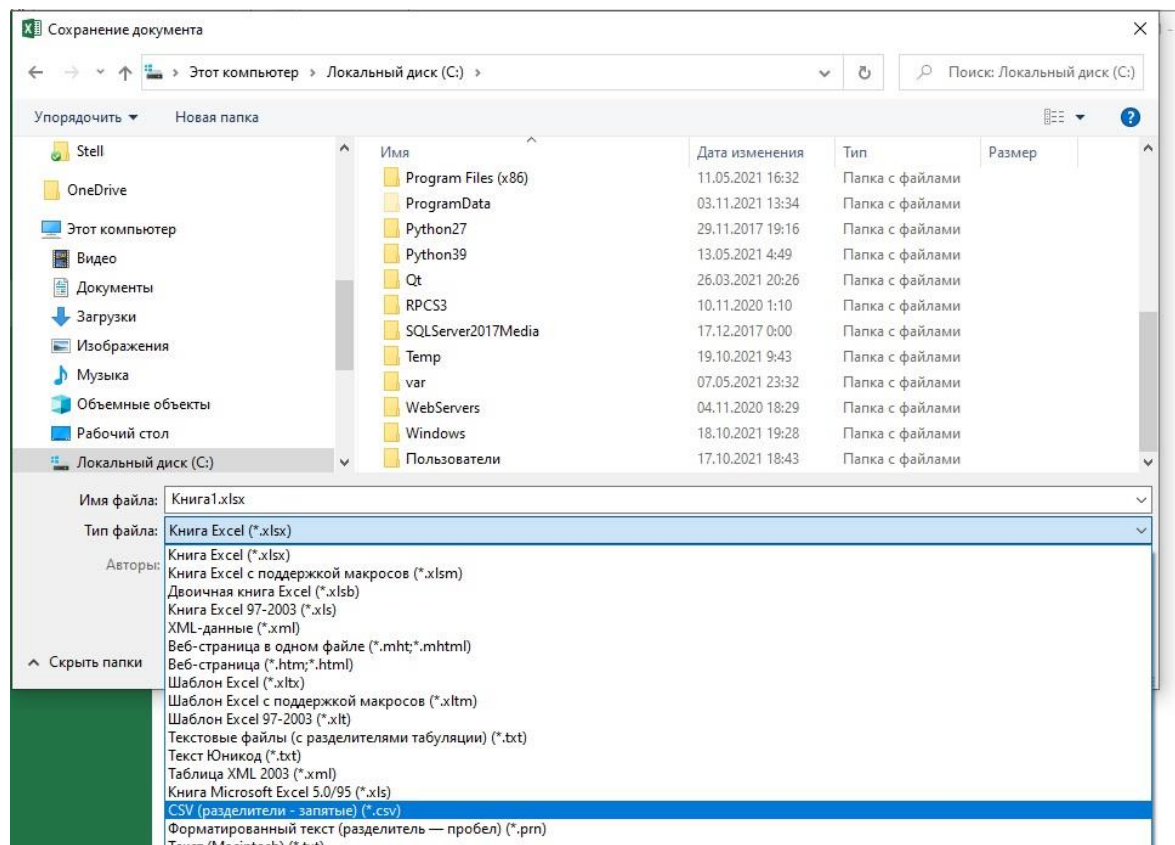
Файл содержит данные по максимальной и минимальной температуре воздуха

в течение каждого месяца 2021 года

В качестве разделителя столбцов выступают запятые

ФОРМАТ CSV

CSV (Comma-Separated Values — значения, разделённые запятыми)



РАБОТА С ФАЙЛАМИ CSV: ЧТЕНИЕ

```
import csv

filename="temperature.csv"
min_t = []
max_t = []

with open(filename, encoding="utf8") as f:
    reader = csv.reader(f)
    header_row = next(reader)
    print(header_row)

    for row in reader:
        print(row)
        z = int(row[1])
        z1 = int(row[2])
        min_t.append(z)
        max_t.append(z1)

print(min_t)
print(max_t)
```

В Python предусмотрен модуль для автоматизированного парсинга (разбиения на составные элементы) строк csv-файлов.

Рассмотрим пример его использования на файле temperature.csv:

```
"месяц год","min","max"
"январь 2021",-34,-20
"февраль 2021",-16,-8
"март 2021",-12,10
"апрель 2021",-8,15
"май 2021",-3,22
"июнь 2021",8,25
"июль 2021",18,32
"август 2021",15,24
"сентябрь 2021",7,21
```

РАБОТА С ФАЙЛАМИ CSV: ЧТЕНИЕ

```
import csv

filename="temperature.csv"
min_t = []
max_t = []

with open(filename, encoding="utf8") as f:
    reader = csv.reader(f)
    header_row = next(reader)
    print(header_row)

    for row in reader:
        print(row)
        z = int(row[1])
        z1 = int(row[2])
        min_t.append(z)
        max_t.append(z1)

print(min_t)
print(max_t)
```

- Подключаем дополнительный пакет «csv»;
- Задаём имя файла, с которым будем работать;
- Объявляем два пустых списка – для хранения минимальных и максимальных значений температуры;

РАБОТА С ФАЙЛАМИ CSV: ЧТЕНИЕ

```
import csv

filename="temperature.csv"
min_t = []
max_t = []

with open(filename, encoding="utf8") as f:
    reader = csv.reader(f)
    header_row = next(reader)
    print(header_row)

    for row in reader:
        print(row)
        z = int(row[1])
        z1 = int(row[2])
        min_t.append(z)
        max_t.append(z1)

print(min_t)
print(max_t)
```

- Воспользовавшись конструкцией *with-as*, открываем файл для чтения;
- Формируем объект-читатель, который будет распознавать строки csv-файла и преобразовывать их во вложенный список;
- При помощи функции *next()* получаем данные очередной строки из csv-файла (в текущем случае - первой);
- Выводим эти данные на печать:

['месяц год', 'min', 'max']

РАБОТА С ФАЙЛАМИ CSV: ЧТЕНИЕ

```
import csv

filename="temperature.csv"
min_t = []
max_t = []

with open(filename, encoding="utf8") as f:
    reader = csv.reader(f)
    header_row = next(reader)
    print(header_row)

    for row in reader:
        print(row)
        z = int(row[1])
        z1 = int(row[2])
        min_t.append(z)
        max_t.append(z1)

print(min_t)
print(max_t)
```

- В цикле, для каждой строки данных, получаемой из объекта-читателя *reader*, выполнить последовательность операторов;
- А именно:
 - получить значение минимальной температуры в текущем месяце, сохранив его в переменную *z*
 - получить значение максимальной температуры в текущем месяце, сохранив его в переменную *z1*
- Сохранить полученные значения в соответствующие списки; (обратите внимание, что название месяца *row[0]* игнорируется)

РАБОТА С ФАЙЛАМИ CSV: ЧТЕНИЕ

```
import csv

filename="temperature.csv"
min_t = []
max_t = []

with open(filename, encoding="utf8") as f:
    reader = csv.reader(f)
    header_row = next(reader)
    print(header_row)

    for row in reader:
        print(row)
        z = int(row[1])
        z1 = int(row[2])
        min_t.append(z)
        max_t.append(z1)

print(min_t)
print(max_t)
```

➤ Наконец, выводим результат обработки файла на печать:

[-34, -16, -12, -8, -3, 8, 18, 15, 7]

[-20, -8, 10, 15, 22, 25, 32, 24, 21]

Оригинальный файл:

```
"месяц год","min","max"
"январь 2021",-34,-20
"февраль 2021",-16,-8
"март 2021",-12,10
"апрель 2021",-8,15
"май 2021",-3,22
"июнь 2021",8,25
"июль 2021",18,32
"август 2021",15,24
"сентябрь 2021",7,21
```


РАБОТА С ФАЙЛАМИ CSV: ЗАПИСЬ

```
import csv

filename = "temperature_copy.csv"

min_t = [-34, -16, -12, -8, -3, 8, 18, 15, 7]
max_t = [-20, -8, 10, 15, 22, 25, 32, 24, 21]

months = ["январь 2021", "февраль 2021",
"март 2021", "апрель 2021", "май 2021",
"июнь 2021", "июль 2021", "август 2021",
"сентябрь 2021"]

with open(filename, "w", encoding="utf8", newline=") as f:
    writer = csv.writer(f, delimiter=' ')
    writer.writerow(["месяц год", "min", "max"])

    for row in range(len(months)):
        writer.writerow([months[row], str(min_t[row]), str(max_t[row])])
```

- Подключаем дополнительный пакет «csv»;
- Задаём имя файла, с которым будем работать;
- Объявляем списки для хранения названий месяцев, минимальных и максимальных значений температуры;

РАБОТА С ФАЙЛАМИ CSV: ЗАПИСЬ

```
import csv

filename = "temperature_copy.csv"

min_t = [-34, -16, -12, -8, -3, 8, 18, 15, 7]
max_t = [-20, -8, 10, 15, 22, 25, 32, 24, 21]

months = ["январь 2021", "февраль 2021",
"март 2021", "апрель 2021", "май 2021",
"июнь 2021", "июль 2021", "август 2021",
"сентябрь 2021"]

with open(filename, "w", encoding="utf8", newline=") as f:
    writer = csv.writer(f, delimiter=' ')
    writer.writerow(["месяц год", "min", "max"])

    for row in range(len(months)):
        writer.writerow([months[row], str(min_t[row]), str(max_t[row])])
```

- Воспользовавшись конструкцией *with-as*, открываем файл для чтения;
- Формируем объект-писатель, который будет записывать строки csv-файла из заданного списка;
 - один список = одна строка в файле;
 - обратите внимание, что на этот раз мы задали в качестве разделителя *пробелы*, а не *запятые*;
- Записываем первую строку файла (заголовки);

РАБОТА С ФАЙЛАМИ CSV: ЗАПИСЬ

```
import csv

filename = "temperature_copy.csv"

min_t = [-34, -16, -12, -8, -3, 8, 18, 15, 7]
max_t = [-20, -8, 10, 15, 22, 25, 32, 24, 21]

months = ["январь 2021", "февраль 2021",
"март 2021", "апрель 2021", "май 2021",
"июнь 2021", "июль 2021", "август 2021",
"сентябрь 2021"]

with open(filename, "w", encoding="utf8", newline=") as f:
    writer = csv.writer(f, delimiter=' ')
    writer.writerow(["месяц год", "min", "max"])

    for row in range(len(months)):
        writer.writerow([months[row], str(min_t[row]), str(max_t[row])])
```

➤ В цикле записываем каждую из строк файла;

➤ Получаем файл на выходе:

temperature_copy.csv:

```
"месяц год" min max
"январь 2021" -34 -20
"февраль 2021" -16 -8
"март 2021" -12 10
"апрель 2021" -8 15
"май 2021" -3 22
"июнь 2021" 8 25
"июль 2021" 18 32
"август 2021" 15 24
"сентябрь 2021" 7 21
```

РАБОТА С ФАЙЛАМИ CSV: ЗАПИСЬ

```
import csv

filename = "temperature_copy.csv"

min_t = [-34, -16, -12, -8, -3, 8, 18, 15, 7]
max_t = [-20, -8, 10, 15, 22, 25, 32, 24, 21]

months = ["январь 2021", "февраль 2021",
"март 2021", "апрель 2021", "май 2021",
"июнь 2021", "июль 2021", "август 2021",
"сентябрь 2021"]

with open(filename, "w", encoding="utf8", newline=") as f:
    writer = csv.writer(f, delimiter=' ')
    writer.writerow(["месяц год", "min temp", "max temp"])

    for row in range(len(months)):
        writer.writerow([months[row], str(min_t[row]), str(max_t[row])])
```

➤ В цикле записываем каждую из строк файла;

➤ Получаем файл на выходе:

temperature_copy.csv:

```
"месяц год" "min temp" "max temp"
"январь 2021" -34 -20
"февраль 2021" -16 -8
"март 2021" -12 10
"апрель 2021" -8 15
"май 2021" -3 22
"июнь 2021" 8 25
"июль 2021" 18 32
"август 2021" 15 24
"сентябрь 2021" 7 21
```

ДОПОЛНИТЕЛЬНЫЕ МЕТОДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ CSV

При чтении строк csv-файла есть возможность сохранять их данные в виде словаря при помощи особого объекта-читателя *DictReader()*:

```
import csv

filename = "temperature.csv"

with open(filename, encoding="utf8") as f:
    reader = csv.DictReader(f)

    for row in reader:
        print(row)
```

Вывод:

```
{'месяц год': 'январь 2021', 'min': '-34', 'max': '20'}
{'месяц год': 'февраль 2021', 'min': '-16', 'max': '8'}
{'месяц год': 'март 2021', 'min': '-12', 'max': '10'}
{'месяц год': 'апрель 2021', 'min': '-8', 'max': '15'}
{'месяц год': 'май 2021', 'min': '-3', 'max': '22'}
{'месяц год': 'июнь 2021', 'min': '8', 'max': '25'}
{'месяц год': 'июль 2021', 'min': '18', 'max': '32'}
{'месяц год': 'август 2021', 'min': '15', 'max': '24'}
{'месяц год': 'сентябрь 2021', 'min': '7', 'max': '21'}
```

ДОПОЛНИТЕЛЬНЫЕ МЕТОДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ CSV

При чтении строк csv-файла есть возможность сохранять их данные в виде словаря при помощи особого объекта-читателя *DictReader()*:

```
import csv

filename = "temperature.csv"

with open(filename, encoding="utf8") as f:
    reader = csv.DictReader(f)

    for row in reader:
        print(row)
```

temperature.csv:

```
"месяц год" min max
"январь 2021" -34 -20
"февраль 2021" -16 -8
"март 2021" -12 10
"апрель 2021" -8 15
"май 2021" -3 22
"июнь 2021" 8 25
"июль 2021" 18 32
"август 2021" 15 24
"сентябрь 2021" 7 21
```

ДОПОЛНИТЕЛЬНЫЕ МЕТОДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ CSV

Значения ключей для словарей также можно задавать вручную:

```
import csv

filename = "temperature.csv"
keys_lst = ["время года", "наим.", "наиб."]

with open(filename, encoding="utf8") as f:
    reader = csv.DictReader(f, fieldnames=keys_lst)

    for row in reader:
        print(row)
```

Вывод:

```
{'время года': 'месяц год', 'наим.': 'min', 'наиб.': 'max'}
{'время года': 'январь 2021', 'наим.': '-34', 'наиб.': '-20'}
{'время года': 'февраль 2021', 'наим.': '-16', 'наиб.': '-8'}
{'время года': 'март 2021', 'наим.': '-12', 'наиб.': '10'}
{'время года': 'апрель 2021', 'наим.': '-8', 'наиб.': '15'}
{'время года': 'май 2021', 'наим.': '-3', 'наиб.': '22'}
{'время года': 'июнь 2021', 'наим.': '8', 'наиб.': '25'}
{'время года': 'июль 2021', 'наим.': '18', 'наиб.': '32'}
{'время года': 'август 2021', 'наим.': '15', 'наиб.': '24'}
{'время года': 'сентябрь 2021', 'наим.': '7', 'наиб.': '21'}
```

ДОПОЛНИТЕЛЬНЫЕ МЕТОДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ CSV

Аналогичным образом работает запись при помощи *DictWriter()* :

```
import csv

filename = "temperature_copy.csv"
keys_lst = ["время года", "наим.", "наиб."]

with open(filename, "w", encoding="utf8") as f:
    writer = csv.DictWriter(f, fieldnames=keys_lst)

    writer.writeheader()
    writer.writerow({"время года": "январь 2021", "наим.": -34, "наиб.": -20})
    writer.writerow({"время года": "февраль 2021", "наим.": -16, "наиб.": -8})
    writer.writerow({"время года": "март 2021", "наим.": -12, "наиб.": 10})
    ...
```


ДОПОЛНИТЕЛЬНЫЕ МЕТОДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ CSV

При записи и чтении из csv-файла можно задавать различные *диалекты*:

```
import csv

filename = "temperature_excel.csv"

min_t = [-34, -16, -12, -8, -3, 8, 18, 15, 7]
max_t = [-20, -8, 10, 15, 22, 25, 32, 24, 21]

months = ["январь 2021", "февраль 2021",
"март 2021", "апрель 2021", "май 2021",
"июнь 2021", "июль 2021", "август 2021",
"сентябрь 2021"]

with open(filename, "w", encoding="utf8", newline=") as f:
    writer = csv.writer(f, dialect='excel')
    writer.writerow(["месяц год", "min temp", "max temp"])

    for row in range(len(months)):
        writer.writerow([months[row], str(min_t[row]), str(max_t[row])])
```

```
import csv

filename = "temperature_unix.csv"

min_t = [-34, -16, -12, -8, -3, 8, 18, 15, 7]
max_t = [-20, -8, 10, 15, 22, 25, 32, 24, 21]

months = ["январь 2021", "февраль 2021",
"март 2021", "апрель 2021", "май 2021",
"июнь 2021", "июль 2021", "август 2021",
"сентябрь 2021"]

with open(filename, "w", encoding="utf8", newline=") as f:
    writer = csv.writer(f, dialect='unix')
    writer.writerow(["месяц год", "min temp", "max temp"])

    for row in range(len(months)):
        writer.writerow([months[row], str(min_t[row]), str(max_t[row])])
```

ДОПОЛНИТЕЛЬНЫЕ МЕТОДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ CSV

Диалект – набор правил, определяющий нюансы записи/чтения данных csv-файлов:

temperature_excel.csv:

```
месяц год,min temp,max temp
январь 2021,-34,-20
февраль 2021,-16,-8
март 2021,-12,10
апрель 2021,-8,15
май 2021,-3,22
июнь 2021,8,25
июль 2021,18,32
август 2021,15,24
сентябрь 2021,7,21
```

temperature_unix.csv:

```
"месяц год","min temp","max temp"
"январь 2021","-34","-20"
"февраль 2021","-16","-8"
"март 2021","-12","10"
"апрель 2021","-8","15"
"май 2021","-3","22"
"июнь 2021","8","25"
"июль 2021","18","32"
"август 2021","15","24"
"сентябрь 2021","7","21"
```

БЛАГОДАРЮ ЗА ВНИМАНИЕ!

Гаврилов Денис Андреевич, преподаватель кафедры СИ ФИТ НГУ