

*Петербургский государственный
университет путей сообщения Императора
Александра I*

Введение в Python

Ведет: аспирант 2 года Волков Егор Алексеевич

gole00201@gmail.com

ауд. 11 - 304

Санкт-Петербург 2024

Строки и методы работы с НИМИ

Основные вопросы лекции:

- Что такое строки.
- Базовые операции со строками.
- Методы строк в Python.
- Форматирование строк.
- Примеры использования.

Что такое строки?

Строка (**string**) — это неизменяемая последовательность символов. В Python строки заключаются в одинарные или двойные кавычки:

```
s1 = 'Привет'  
s2 = "Hello"
```

- Многострочные строки можно создавать с помощью тройных кавычек:

```
s3 = """Это  
многострочная  
строка"""
```

Базовые операции со строками

- Конкатенация (объединение строк)

```
s1 = "Hello"  
s2 = "World"  
s3 = s1 + " " + s2  
print(s3)  
# "Hello World"
```

Дублирование строки

```
s = "Ha"  
print(s * 3)  
# "HaHaHa"
```

Доступ к символам по индексу

```
s = "Python"  
print(s[0])    # 'P'  
print(s[-1])   # 'n'
```

Методы строк в Python

`len()`: длина строки

```
s = "Hello"  
print(len(s))  
# 5
```

`lower()` и `upper()`: изменение регистра

```
s = "PyThOn"  
print(s.lower()) # "python"  
print(s.upper()) # "PYTHON"
```


`strip()`: удаление лишних пробелов

```
s = "    Hello    "  
print(s.strip())  
# "Hello"
```

`split()`: разделение строки на список

```
s = "apple,banana,orange"  
fruits = s.split(',')  
print(fruits)  
# ['apple', 'banana', 'orange']
```

`join()`: объединение элементов списка в строку

```
fruits = ['apple', 'banana', 'orange']  
s = ", ".join(fruits)  
print(s)  
# "apple, banana, orange"
```

`replace()`: замена подстроки

```
s = "I love Java"
s_new = s.replace("Java", "Python")
print(s_new)
# "I love Python"
```

Форматирование строк

Форматирование с помощью f-строк

```
name = "Егор"  
age = 25  
s = f"Меня зовут {name}, и мне {age} лет."  
print(s)  
# "Меня зовут Егор, и мне 25 лет."
```

Метод `format()`

```
s = "Привет, {}. Добро пожаловать в {}.".format("Егор", "Python")  
print(s)  
# "Привет, Егор. Добро пожаловать в Python."
```

Работа с файлами и парсинг строк

Основные вопросы лекции:

- Открытие и закрытие файлов.
- Чтение данных из файлов.
- Парсинг строк.
- Запись данных в файлы.
- Работа с различными форматами данных (CSV).

Открытие и закрытие файлов

В Python для работы с файлами используется
встроенная функция `open()`.

Пример открытия файла:

```
file = open('example.txt', 'r') # 'r' означает режим чтения (read)
```


Важно закрывать файл после работы:

```
file.close()
```

- Лучший способ работы с файлами — использовать конструкцию `with`, которая автоматически закрывает файл:

```
with open('example.txt', 'r') as file:  
    content = file.read()  
# Файл будет закрыт автоматически
```

Чтение данных из файлов

Методы чтения:

- `read()`: читает весь файл целиком.
- `readline()`: читает одну строку.
- `readlines()`: читает все строки и возвращает список.

Примеры:

```
# Чтение всего файла
with open('example.txt', 'r') as file:
    content = file.read()

# Чтение построчно
with open('example.txt', 'r') as file:
    line = file.readline()
    while line:
        print(line)
        line = file.readline()
```

Парсинг строк

- Часто при чтении файла нужно обработать его содержимое. Например, разобрать строки на отдельные части.

Разбиение строк на части:

```
line = "2024-10-10, John Doe, 500"  
parts = line.split(', ')  
print(parts)  
# ['2024-10-10', 'John Doe', '500']
```

Удаление лишних символов:

```
line = "  Some data with spaces  \n"  
cleaned = line.strip()  
print(cleaned)  
# "Some data with spaces"
```

Пример: Парсинг CSV- файлов

- CSV (Comma Separated Values) — это формат, где данные разделены запятыми. Как журнал вашей группы, что лежит в гите.

```
ФИО;1ДЗ; 2ДЗ; 3ДЗ; 4ДЗ; 5ДЗ; 6ДЗ; 7ДЗ; 8ДЗ; 9ДЗ; 10ДЗ; КР; Зачет  
Карпов Артемий Алексеевич;НЗ;-;*;  
Лукьянова Евгения Александровна;НЗ;-;*;  
Машкина Анастасия Денисовна;НЗ;-;*;  
Петриенко Максим Андреевич;НЗ;-;*;  
Тихоненко Ирина Сергеевна;3(с минусом);3;3;  
Числов Илья Николаевич;НЗ;3;3;  
...
```

```
with open('data.csv', 'r') as file:
    for line in file:
        values = line.strip().split(',') # .strip() - метод строки убирающий
        # лишние символы
        # .split() - метод строки разделяющий строку на под-строки и размещающий
        # результат в список
        print(values)
```

Использование библиотеки csv для работы с CSV- файлами:

```
import csv

with open('data.csv', 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```


Запись данных в файлы

- Для записи данных используется режим 'w' (write) или 'a' (append — добавление в конец файла).

```
with open('output.txt', 'w') as file:  
    file.write('Hello, World!')
```

Запись списка строк:

```
lines = ['First line', 'Second line', 'Third line']  
  
with open('output.txt', 'w') as file:  
    file.writelines(f"{line}\n" for line in lines)
```

Задание 1: Парсинг CSV

Дана строка, представляющая данные о студентах в формате CSV:

```
data = "Иванов Иван, 20, Математика; Петров Петр, 21, Физика; Сидоров Сидор, 22, Химия"
```

Напишите программу, которая:

Разделяет строку на отдельные записи о студентах.

Для каждой записи выводит имя студента, его возраст и факультет в формате:

```
Имя: Иванов Иван  
Возраст: 20  
Факультет: Математика
```

Задание 2: Извлечение email-адресов

Дана строка, содержащая текст с email-адресами:

```
text = "Контакты: ivanov@example.com, petrov@work.net, sid@mail.ru"
```

Напишите программу, которая:

- Извлечет все email-адреса из строки.
- Сохранит их в список и выведет результат.

Задание 3: Подсчет количества слов

Дана строка, содержащая предложение:

```
sentence = "Python is a powerful and easy-to-learn programming language."
```

Напишите программу, которая:

- Разделит предложение на отдельные слова.
- Подсчитает, сколько слов в предложении.

Задание 4: Удаление дубликатов символов

Дана строка:

```
s = "aaabbbcccaadddd"
```

Напишите программу, которая:

- Удалит все повторяющиеся подряд символы, оставив только один символ из каждого фрагмента.
- Выведет получившуюся строку.

Задание 5: Извлечение чисел

Дана строка, содержащая текст и числа:

```
text = "Сегодня 20 градусов, завтра будет 18 градусов, а вчера было 22 градуса."
```

Напишите программу, которая:

- Извлечет все числа из строки.
- Выведет их в виде списка.

Пример вывода:

```
[20, 18, 22]
```

Задание на курсовой проект

- Необходимо реализовать консольное приложение, которое осуществляет считывание текстового файла csv, состоящего из строк следующего формата:

YEAR;MONTH;DAY;HOUR;MINUTE;TEMPERATURE

- dddd - год 4 цифры
- mm - месяц 2 цифры
- dd - день 2 цифры
- hh - часы 2 цифры
- mm - минуты 2 цифры
- temperature - целое число от -99 до 99

- В архиве файле хранится статистика собранная датчиком температуры за 1 календарный год. Предполагается, что датчик собирал информацию не чаще чем 1 раз в минуту и сохранял в заданном формате каждое значение в текстовый файл с новой строки.
- В какой-то момент времени датчик мог не работать, тогда данные поэтому периоду могут отсутствовать.

Требования к обработке данных

Необходимо вывести статистику по каждому месяцу,
с учетом исходных данных:

- среднемесячная температура
- минимальная температура в текущем месяце
- максимальная температура в текущем месяце

Также необходимо вывести статистику за год:

- среднегодовая температура
- минимальная температура
- максимальная температура

- Приложение должно обрабатывать аргументы командной строки:

-h Описание функционала приложения. Список ключей;

-f <filename.csv> входной файл csv для обработки;

-m <номер месяца> если задан данный ключ, то выводится только статистика за указанный месяц;

если нет параметров, то выдается help.

Файлы подлежащие парсингу лежат в директории с этой лекцией

!Важно

- В одном из файлов ошибка форматирования csv файла. Ваша программа должна наглядно мне сказать в каком месте допущена ошибка.

Формат сдачи

- Сделать в вашем гите папку с проектом (пр. NAME_parser_csv)
- Положить туда код программы
- Положить туда файл (pdf, md) в котором содержатся скриншоты корректной работы программы