НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Факультет ФПИиКТ

Дисциплина: Информатика

Лабораторная работа № 4

Выполнил студент

Агнистова Алина Юрьевна

Группа № P3125

Преподаватель

Болдырева Елена Александровна

г. Санкт-Петербург

2022

Содержание

Задание

1. Обязательное задание

Написать программу на языке Python 3.x, которая бы осуществляла парсинг и конвертацию исходного файла в новый.

2. Доп. задание No1 (+10 баллов)

a) Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов.
b) Переписать исходный код, применив найденные библиотеки. Регулярные выражения также нельзя использовать.
c) Сравнить полученные результаты и объяснить их сходство/различие.

3. Доп. задание No2 (+10 баллов)

a) Переписать исходный код, добавив в него использование регулярных выражений.

b) Сравнить полученные результаты и объяснить их сходство/различие.

4. Доп. задание No3 (+10 баллов)

a) Используя свою исходную программу из обязательного задания, программу из дополнительного задания No1 и программу из дополнительного задания No2, сравнить десятикратное время выполнения парсинга + конвертации в цикле.

b) Проанализировать полученные результаты и объяснить их сходство/различие.

5. Доп. задание No4 (+5 баллов)

a) Переписать исходную, чтобы она осуществляла парсинг и конвертацию исходного файла в любой другой формат (кроме JSON, YAML, XML, HTML): PROTOBUF, TSV, CSV, WML и т.п.

b) Проанализировать полученные результаты, объяснить осоебнности использованного формата.

Отчет

Реализация основного задания

base.py

```python
from time import time #dop3
start = time() #dop3 (точка отсчёта времени)
def yaml_to_list(filename):
    with open(filename, "r") as file:
        strings = file.read().split('\n')
        dayname = ''
        count = -1 #counter of the number of lessons
        main = {} #nice sequence to do json
        for i in strings:
            strr = i.lstrip().rstrip()
            if strr.count(':') == 1:
                key, expression = strr.split(':')
                if '-' not in key and len(expression) == 0:
                    main[key] = []
                    dayname = key
                elif '-' in key:
                    temp = key.lstrip('- ')
                    main[dayname].append({temp: {}})
                    count += 1
                    lessonatm = temp # we save lesson what we have at the moment
                else:
                    expression = expression.lstrip().rstrip()
                    if "'" in expression:
                        expression = expression.lstrip("'").lstrip("'")
                    main[dayname][count][lessonatm][key] = expression
            elif strr.count(':') == 3:
                key, timepart1, timepart2, timepart3 = strr.split(":")
                expression = timepart1 + ":" + timepart2 + ":" + timepart3
                expression = expression.lstrip().rstrip()
                if "'" in expression:
                    expression = expression.lstrip("'").rstrip("'")
                main[dayname][count][lessonatm][key] = expression

        return [main, dayname]
```

```
###################################
def list_to_json(main, dayname, output_filename):
    spaces = 2
    with open(output_filename, "w") as file1:
        numb = 0
        file1.write("{\n")
        file1.write(spaces*"" + '"' + dayname + '": [\n')
        spaces += 2
        file1.write(spaces*"" + "{\n")
        spaces += 2
        for i in main[dayname]:
            for key1 in i.keys():
                file1.write(spaces*"" + '"' + key1 + '":{\n')
                temp_main = main[dayname][numb][key1]
                numb += 1
                spaces += 2
                for key2 in temp_main.keys():
                    file1.write(spaces*'' + '"' + key2 + '": ')
                    if key2 != "teacher":
                        file1.write('"' + temp_main[key2] + '", \n')
                    else:
                        file1.write('"' + temp_main[key2] + '"\n')
                spaces -= 2
                file1.write(spaces*'' + "}\n")
                if key1 != "ОПД(практика 2)":
                    file1.write(spaces * ' ' + "},\n")
                    file1.write(spaces * ' ' + "{\n")
                else:
                    file1.write(spaces * ' ' + "}\n")
                    spaces -= 2
                    file1.write(spaces * ' ' + "]\n")
                    spaces -= 2
                    file1.write(spaces * ' ' + "}\n")

                spaces += 2
```

```
middle_stage = yaml_to_list("yaml")
last_stage = list_to_json(*middle_stage, "json")
file = open("json").read()
print(file)
print("time base_task:", 10*(time() - start)) #dop3 (время работы программы, время в данный момент - время старта)
```

```
{
"Пятница": [
{
"ОПД(практика 1)":{
"time": "10:00-11:30",
"weeks": "2-16'",
"location": "Кронверкский пр., д.49, лит.А'",
"classroom": "2305/1'",
"teacher": "Белозубов Александр Владимирович'"
"type": "Очно-дистанционный'",
}
      },
      {
"ОПД(практика 2)":{
"time": "11:40-13:10",
"weeks": "2-16'",
"location": "Кронверкский пр., д.49, лит.А'",
"classroom": "2305/1'",
"teacher": "Белозубов Александр Владимирович'"
"type": "Очно-дистанционный'",
}
        }
      ]
    }

time base_task: 0.0067067146301269 53

Process finished with exit code 0
```

Реализация 1 доп. задания:

dop1.py

```python
#with using libs
import yaml
import json
from time import time #dop3
sec = time() #dop3 (точка отсчёта времени)
def yaml_to_json(input_file, output_file):
    with open(input_file, 'r', encoding="UTF-8") as yamlfile, open(output_file, "w", encoding="UTF-8") as jsonfile:
        yaml_dict = yaml.safe_load(yamlfile) #reading file and show it like a dictionary
        json.dump(yaml_dict, jsonfile, ensure_ascii=False, sort_keys=True, indent=2)#write json file and do a sort dictionaries, indenting, экранирование

yaml_to_json("yaml", "json1")
file = open("json1").read()
print(file)
print("time dop1_task:", 10*(time() - sec)) #dop3 (время работы программы, время в данный момент - время старта)
```

```
{
  "Пятница": [
    {
      "classroom": "2305/1",
      "location": "Кронверкский пр., д.49, лит.А",
      "teacher": "Белозубов Александр Владимирович",
      "time": "10:00-11:30",
      "type": "Очно-дистанционный",
      "weeks": "2-16",
      "ОПД(практика 1)": null
    },
    {
      "classroom": "2305/1",
      "location": "Кронверкский пр., д.49, лит.А",
      "teacher": "Белозубов Александр Владимирович",
      "time": "11:40-13:10",
      "type": "Очно-дистанционный",
      "weeks": "2-16",
      "ОПД(практика 2)": null
    }
  ]
}
time dop1_task: 0.034432411193847656

Process finished with exit code 0
```

Реализация 2 доп. задания:

dop2.py

```
import re
from time import time #dop3
sec = time() #dop3 (точка отсчёта времени)
def from_yaml_to_json_with_regulars(filename, output_filename):
    with open(filename, 'r', encoding="UTF-8") as yamlf, open(output_filename, "w", encoding="UTF-8") as jsonf:
        flag_last = False
        strr = yamlf.readlines()
        jsonf.write("{\n")
        for line in strr:
            temp = re.findall(r"^\w+:", line)
            if len(temp) > 0:
                result = re.sub(r"^\w+:", " " * 2 + '"' + temp[0].rstrip(":") + '": [', line)
                jsonf.write(result)
            temp = re.findall(r"\s+-", line)
            if len(temp) > 0:
                if "практика 2" in line:
                    flag_last = True
                jsonf.write("    {\n")
                temp = line.lstrip("  - ").rstrip(":\n")
                jsonf.write(6 * " " + '"' + temp + '": {\n')
            temp = re.findall(r"^\s+.+: .+", line)
            if len(temp) > 0:
                temp_2 = temp[0].lstrip().split(":")
                temp_2[1] = temp_2[1].lstrip(" '").rstrip("'")
                if len(temp_2) == 3:
                    jsonf.write(' ' * 8 + '"' + temp_2[0] + '": "' + temp_2[1] + ":" + temp_2[2].rstrip("'") + '",\n')
                elif "type" not in temp_2[0]:
                    jsonf.write(' ' * 8 + '"' + temp_2[0] + '": "' + temp_2[1] + '",\n')
                else:
                    jsonf.write(' ' * 8 + '"' + temp_2[0] + '": "' + temp_2[1] + '"\n')
                    if not flag_last:
                        jsonf.write(6 * ' ' + '}\n' + "    },\n")
                    else:
                        jsonf.write(6 * ' ' + '}\n' + "    }\n" + "  ]\n" + "}")
from_yaml_to_json_with_regulars("yaml", "json2")
file = open("json2").read()
print(file)
print("time dop2_task:", 10*(time() - sec)) #dop3 (время работы программы, время в данный момент - время старта)
```

```
{
  "Пятница": [
    {
      "ОПД(практика 1)": {
        "time": "10",
        "weeks": "2-16",
        "location": "Кронверкский пр., д.49, лит.А",
        "classroom": "2305/1",
        "teacher": "Белозубов Александр Владимирович",
        "type": "Очно-дистанционный"
      }
    },
    {
      "ОПД(практика 2)": {
        "time": "11",
        "weeks": "2-16",
        "location": "Кронверкский пр., д.49, лит.А",
        "classroom": "2305/1",
        "teacher": "Белозубов Александр Владимирович",
        "type": "Очно-дистанционный"
      }
    }
  ]
}
time dop2_task: 0.0046539306640625

Process finished with exit code 0
```

Все конвертированные файлы идентичны, за исключением 1 доп.задания и основного, где порядок характеристик другой из-за сортировки. Сравнение времени: основное решение работает быстрее всего, среднее время имеет решение с библиотеками (оно медленнее, чем основное, так как необходимо время на загрузку библиотек), самым медленным оказалось решение с помощью регулярных выражений из-за постоянного вызова библиотеки регулярных выражений.

Вывод

В ходе выполнения я ознакомилась с тем, что представляет из себя парсинг данных и языки разметки, попробовала решить задачу парсинга разными способами, улучшила навык работы с регулярными выражениями.

Список литературы

1. Регулярные выражения в Python от простого к сложному. // Habr URL: https://habr.com/ru/post/349860/ (дата обращения: 15.11.2022).
2. YAML vs JSON. // Habr URL: https://habr.com/ru/company/rambler_and_co/blog/525498/ (дата обращения: 15.11.2022).