Факультет ПИиКТ

Информатика

Лабораторная работа №3

Исследование языков разметки документов

Вариант 2

Преподаватель:

Ильина Аглая Геннадьевна

Выполнил: Бейгер Герман Юрьевич

Р3114

Санкт-Петербург

2020

Задание:

Вариант 2: XML -> YAML

1. Определить номер варианта как остаток деления на 35 порядкового номера в списке группы в ISU. В случае, если в данный день недели нет занятий, то увеличить номер варианта на восемь.
2. Изучить форму Бэкуса-Наура.
3. Изучить особенности языков разметки/форматов JSON, YAML, XML.
4. Понять устройство страницы с расписанием для своей группы:

http://www.ifmo.ru/ru/schedule/0/P3200/schedule.htm

1. Исходя из структуры расписания конкретного дня, сформировать файл с расписанием в формате, указанном в задании в качестве исходного.
2. Написать программу на языке Python 3.x, которая бы осуществляла парсинг и конвертацию исходного файла в новый.
3. Нельзя использовать готовые библиотеки, кроме re (регулярные выражения в Python) и библиотеки для загрузки XML-файлов.
4. Необязательное задание для получения оценки «4» и «5» (позволяет набрать от 75 до 89 процентов от максимального числа баллов БаРС за данную лабораторную).
   1. a)  Найти готовые библиотеки, осуществляющие аналогичный парсинг и конвертацию файлов.
   2. b)  Сравнить полученные результаты и объяснить их сходство/различие.
5. Необязательное задание для получения оценки «5» (позволяет набрать от 90 до 100 процентов от максимального числа баллов БаРС за данную лабораторную).
   1. a)  Используя свою программу и найденные готовые библиотеки, сравнить десятикратное время выполнения парсинга + конвертации в цикле.
   2. b)  Проанализировать полученные результаты и объяснить их сходство/различие.

10.Проверить, что все пункты задания выполнены и выполнены верно. 11.Написать отчёт о проделанной работе.  
12.Подготовиться к устным вопросам на защите.

Код программы

# task 1

import re

import xml.etree.ElementTree as ET

# data loading

from bs4 import BeautifulSoup

from bs4.element import Tag

import urllib3

import sys

# task 2

import xmlplain

#task 3

import time

def xml\_to\_yaml(string):

def process\_tag(root):

text = f"name: \"{root.tag}\"\n"

for name in root.attrib:

text += f"{name}: \"{root.attrib[name]}\"\n"

if type(root.text) == str:

inner\_text = root.text.replace('\n', '\n ')

text += f"text:\n {inner\_text}\n"

if len(root) > 0:

text += f"content: \n"

for child in root:

text += " - "+process\_tag(child).replace('\n', "\n ").rstrip()+"\n"

return text

root = ET.fromstring(string)

return process\_tag(root)

if \_\_name\_\_ == "\_\_main\_\_":

# проверяем передали ли параметр в скрипт

assert len(sys.argv) == 2

# извлекаем данные о расписании

http = urllib3.PoolManager()

r = http.request('GET', 'https://itmo.ru/ru/schedule/0/P3114/schedule.htm')

htm = r.data.decode()

psd = BeautifulSoup(htm, 'html.parser')

monday = str(psd.find\_all('div', {"class": "rasp\_tabl\_day"})[0])

with open('xml\_input.xml', 'w') as xml\_file:

xml\_file.write(monday)

if sys.argv[1] == "1":

# парсим и сохраняем готовый yaml

with open('xml\_input.xml') as xml\_file, open('yml\_output.yaml', 'w') as yml\_file:

yml\_file.write(xml\_to\_yaml('\n'.join(list(xml\_file))))

elif sys.argv[1] == "2":

mond\_obj = xmlplain.xml\_to\_obj(monday, strip\_space=True, fold\_dict=True)

with open('output\_task2.yaml', 'w') as f:

xmlplain.obj\_to\_yaml(mond\_obj, f)

elif sys.argv[1] == "3":

time1 = time.time()

for i in range(10):

xml\_to\_yaml(monday)

my\_code\_time = time.time() - time1

time1 = time.time()

for i in range(10):

mond\_obj = xmlplain.xml\_to\_obj(monday, strip\_space=True, fold\_dict=True)

xmlplain.obj\_to\_yaml(mond\_obj)

not\_my\_code\_time = time.time() - time1

print(f"Время работы моего кода в среднем за 10 запусков: {my\_code\_time}")

print(f"Время работы библиотеки в среднем за 10 запусков: {not\_my\_code\_time}")

Задание 4

Для конвертации из XML в YAML я использовал библиотеку xmlplain. Поскольку прямое конвертирование из XML в YAML невозможно (даже из названий можно сделать этот вывод, XML – eXtended Markup Language, YAML – YAML Ain’t Markup Language), отличия зависят от реализации и того, какие цели преследуются. Для сохранения информации и мой вариант и вариант, предложенный в библиотеке подходят.

Вариант из библиотеки:

div:

'@class': rasp\_tabl\_day

table:

'@border': '0'

'@cellpadding': '0'

'@cellspacing': '0'

'@class': rasp\_tabl

'@id': 1day

tbody:

Мой вариант:

name: "div"

class: "rasp\_tabl\_day"

text:

content:

- name: "table"

border: "0"

cellpadding: "0"

cellspacing: "0"

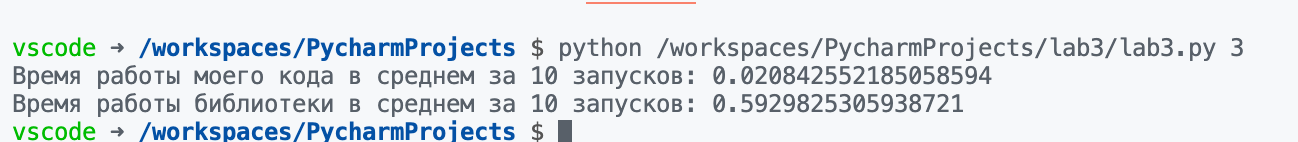
class: "rasp\_tabl"

id: "1day"

text:

Задание 5

Поскольку библиотека конвертирует в промежуточный формат, а также поддерживает дополнительные функции, которые усложняют код, мой код, предназначенный для простого случая и переводящий напрямую, работает быстрее.



Вывод: