

Лабораторная работа 10. Парсинг JSON.

Для выполнения потребуется файл `interface_config.json` и файл `interfaces.json`.

Задание 1. Преобразование json данных в объект python.

Создайте файл с именем `lab10_1.py`. Создайте следующий скрипт:

1. Импортируйте модуль `json`, функцию `pprint` из модуля `pprint` (тема по импорту будет позже):

```
import json
from pprint import pprint
```

2. С помощью конструкции `with open` откройте в режиме чтения и прочитайте файл `interface_config.json`, сохранив результат в переменную `json_text`.
 3. Выведите на экран тип переменной `json_text`.
 4. Выведите на экран содержимое переменной `json_text`.
- Результат будет выглядеть так:

```
Тип переменной json_text - <class 'str'>
Содержимое переменной json_text
{
    "ietf-interfaces:interface": {
        "name": "GigabitEthernet2",
        "description": "Wide Area Network",
        "enabled": true,
        "ietf-ip:ipv4": {
            "address": [
                {
                    "ip": "172.16.0.2",
                    "netmask": "255.255.255.0"
                }
            ]
        }
    }
}
```

5. С помощью функции loads() модуля json преобразуйте json формат в формат данных Python и сохраните результат в переменной json_data.
6. Выведите на экран тип переменной json_data.
7. С помощью pprint() получите «красивый» вывод содержимого переменной json_data. Результат будет такой:

```
Тип переменной json_data - <class 'dict'>
Содержимое переменной json_data
{'ietf-interfaces:interface': {'description': 'Wide Area Network',
                                'enabled': True,
                                'ietf-ip:ipv4': {'address': [{'ip': '172.16.0.2',
                                                                'netmask': '255.255.255.0'}]},
                                'name': 'GigabitEthernet2'}}
```

Задание 2. Работа с вложенными данными.

Добавьте код в файл lab10_2.py, где содержатся результаты задания 1 этой лабораторной работы:

1. Импортируйте модуль json, функцию pprint из модуля pprint (как в предыдущем задании этой работы)
2. Откройте в режиме чтения файл interface_config.json с помощью with и внутри конструкции with выполните парсинг json в python в переменную json_data.
3. С помощью pprint() выведите на экран значение ключа ***'ietf-interfaces:interface'***

Будет получен следующий результат:

```
{'description': 'Wide Area Network',
 'enabled': True,
 'ietf-ip:ipv4': {'address': [{'ip': '172.16.0.2',
                               'netmask': '255.255.255.0'}]},
 'name': 'GigabitEthernet2'}
```

4. Извлеките и с помощью pprint() выведите на экран ip адрес интерфейса. Необходимо получить такой результат:

```
IP адрес интерфейса:
'172.16.0.2'
```

Задание 3. Извлечение вложенных данных с помощью цикла
Создайте файл с именем lab10_3.py. Напишите следующий скрипт:

1. Выполните импорт json модуля и функции pprint() из модуля pprint (см. задание 1).
2. Откройте в режиме чтения файл interfaces.json с помощью with и внутри конструкции with выполните parsing json в python в переменную json_data. Выведите на экран в красивом виде содержимое json_data.

```
{'ietf-interfaces:interfaces': {'interface': [{'enabled': True,
                                             'ietf-ip:ipv4': {'address': [{'ip': '198.18.134.11',
                                                                           'netmask': '255.255.192.0'}]},
                                             'ietf-ip:ipv6': {},
                                             'name': 'GigabitEthernet1',
                                             'type': 'iana-if-type:ethernetCsmacd'},
                                           {'enabled': True,
                                             'ietf-ip:ipv4': {'address': [{'ip': '172.16.255.1',
                                                                           'netmask': '255.255.255.0'}]},
                                             'ietf-ip:ipv6': {},
                                             'name': 'GigabitEthernet2',
                                             'type': 'iana-if-type:ethernetCsmacd'},
                                           {'description': 'loop 0',
                                             'enabled': True,
                                             'ietf-ip:ipv4': {'address': [{'ip': '10.0.0.1',
                                                                           'netmask': '255.255.255.255'}]},
                                             'ietf-ip:ipv6': {},
                                             'name': 'Loopback0',
                                             'type': 'iana-if-type:softwareLoopback'}]}}
```

3. Создайте цикл, с помощью которого можно_получить все имена интерфейсов:

```
for interface in json_data["ietf-interfaces:interfaces"]["interface"]:
    print(interface["name"])
```

Результат работы этого цикла:

```
GigabitEthernet1
GigabitEthernet2
Loopback0
```

4. Создайте цикл, который считывает данные из json_data и выводит на экран информацию о каждом интерфейсе: имя интерфейса, его

ip адрес и маску (для вывода результата на экран используйте форматирование: .format или f строки).

Результат работы цикла должен выглядеть следующим образом:

```
GigabitEthernet1: 198.18.134.11 255.255.192.0  
GigabitEthernet2: 172.16.255.1 255.255.255.0  
Loopback0: 10.0.0.1 255.255.255.255
```