



Trường ĐH Khoa Học Tự Nhiên Tp. Hồ Chí Minh
TRUNG TÂM TIN HỌC

Lập trình Python nâng cao

Bài 2: *Làm việc với dữ liệu JSON*

Phòng LT & Mạng

<http://csc.edu.vn/lap-trinh-va-csdl>

Nội dung

1. Giới thiệu
2. Làm việc với dữ liệu JSON

Giới thiệu



- JSON là dạng dữ liệu lightweight text dựa trên open standard được thiết kế để trao đổi dữ liệu.
- JSON format được đưa ra bởi Douglas Crockford, và được mô tả trong tiêu chuẩn RFC 4627.
- JSON được mở rộng từ JavaScript scripting language.
- Tập tin JSON có phần mở rộng là .json
- JSON được sử dụng trong hầu hết các NNLT như PHP, PERL, Python, Ruby, Java...
- JSON Internet Media type là application/json.

Giới thiệu

❑ Đặc điểm

- Rất dễ đọc và viết.
- Là một định dạng trao đổi văn bản siêu nhẹ
- JSON là ngôn ngữ độc lập
- Dễ dàng được sử dụng trên Internet
- Hỗ trợ nhiều ứng dụng

Giới thiệu

□ JSON syntax

- Xuất phát từ cú pháp ký hiệu của JavaScript:
 - Dữ liệu theo định dạng cặp name/value như sau:
“name” : value
 - Dữ liệu được phân cách nhau bằng dấu phẩy ,
 - Ngoặc nhọn (curly brace) {} chứa các đối tượng
 - Ngoặc vuông (square bracket) [] chứa các mảng

Giới thiệu

□ JSON Syntax

- Ví dụ

```
{  
    "books": [  
        { "language": "Java" , "edition": "first" },  
        { "language": "Python" , "edition": "second" },  
        { "language": "Kotlin" , "edition": "third" }  
    ]  
}
```

Chú ý: trong JSON, name (*key*) *phải là* string, được viết trong dấu nháy đôi double quote (“”)

Giới thiệu

□ JSON hỗ trợ 2 kiểu cấu trúc dữ liệu:

- Tập hợp các cặp **name/value** – cấu trúc dữ liệu này được hỗ trợ bởi rất nhiều ngôn ngữ lập trình khác nhau
- Danh sách các **value có thứ tự** – bao gồm array, list, vector, hoặc sequence...

Giới thiệu

□ Các kiểu dữ liệu hỗ trợ

- string
- number
- object (JSON object)
- array
- boolean
- null

Không hỗ trợ các kiểu: function, date,
undefined

Giới thiệu

❑ So sánh JSON và XML

- Giống nhau:

- Điều dùng để nhận và chuyển dữ liệu
- Điều là ngôn ngữ tự mô tả ("self describing" - human readable)
- Điều theo cấu trúc phân cấp (hierarchical - values within values)
- Điều có thể phân tích (parse) và được sử dụng bởi rất nhiều ngôn ngữ lập trình khác nhau
- Điều có thể được tìm và nạp bởi XMLHttpRequest

Giới thiệu

❑ So sánh JSON và XML

- Khác nhau:

- JSON không sử dụng end tag
- JSON ngắn gọn hơn
- JSON đọc và ghi nhanh hơn
- JSON có thể sử dụng array
- *XML được phân tích (parse) với một XML parser. JSON có thể phân tích (parse) bởi một standard JavaScript function.*

Nội dung

1. Giới thiệu
2. Làm việc với dữ liệu JSON

Làm việc với dữ liệu JSON

❑ Đọc dữ liệu JSON từ Internet

- Python cung cấp thư viện urllib giúp làm việc với đường dẫn url trên internet
- Và cung cấp thư viện json để xử lý dữ liệu JSON
- `import urllib.request, json`

Làm việc với dữ liệu JSON

❑ Đọc dữ liệu JSON từ Internet

- Bước 1: Mở url dẫn đến nơi chứa nội dung JSON

```
url = urllib.request.urlopen("tên đường dẫn")
```

- Ví dụ:

```
url =  
urllib.request.urlopen("http://maps.googleapis.com/m  
aps/api/geocode/json?address=google")
```

Làm việc với dữ liệu JSON

❑ Đọc dữ liệu JSON từ Internet

- Bước 2: Đọc dữ liệu từ url => decode dữ liệu. Sau đó dùng phương thức **loads()** để parse dữ liệu JSON

```
data = json.loads(url.read().decode())
```

- Bước 3: Xử lý dữ liệu theo yêu cầu
 - Ví dụ: In dữ liệu theo định dạng JSON (import pprint)

```
pprint(data)
```

Làm việc với dữ liệu JSON

```
{'results': [{ 'address_components': [{ 'long_name': '1600',  
    'short_name': '1600',  
    'types': ['street_number']},  
    { 'long_name': 'Amphitheatre Parkway',  
    'short_name': 'Amphitheatre Pkwy',  
    'types': ['route']},  
    { 'long_name': 'Mountain View',  
    'short_name': 'Mountain View',  
    'types': ['locality', 'political']},  
    { 'long_name': 'Santa Clara County',  
    'short_name': 'Santa Clara County',  
    'types': ['administrative_area_level_2',  
    'political']},  
    { 'long_name': 'California',  
    'short_name': 'CA',  
    'types': ['administrative_area_level_1',  
    'political']},  
    { 'long_name': 'United States',  
    'short_name': 'US',  
    'types': ['country', 'political']},  
    { 'long_name': '94043',  
    'short_name': '94043',  
    'types': ['postal_code']}],  
    'formatted_address': '1600 Amphitheatre Pkwy, Mountain View, CA '  
    '94043, USA',  
    'geometry': { 'location': { 'lat': 37.4219999, 'lng': -122.0840575},  
    'location_type': 'ROOFTOP',  
    'viewport': { 'northeast': { 'lat': 37.4233488802915,  
    'lng': -122.0827085197085},  
    'southwest': { 'lat': 37.4206509197085,  
    'lng': -122.0854064802915}}},  
    'place_id': 'ChIJj61dQgK6j4AR4GeTYWZsKlww',  
    'types': ['establishment', 'point_of_interest']},  
    'status': 'OK'}
```

Làm việc với dữ liệu JSON

❑ Đọc dữ liệu JSON từ Internet

- Trong trường hợp nội dung theo định dạng UNICODE thì cần phải khai báo thêm DEFAULT_ENCODING.

■ Ví dụ:

```
DEFAULT_ENCODING = 'utf-8'
url = 'https://api.github.com/users?since=100'
urlResponse = urllib.request.urlopen(url)

if hasattr(urlResponse.headers, 'get_content_charset'):
    encoding = urlResponse.headers.get_content_charset(DEFAULT_ENCODING)
else:
    encoding = urlResponse.headers.getparam('charset') or DEFAULT_ENCODING

data = json.loads(urlResponse.read().decode(encoding))
pprint(data)
```


Làm việc với dữ liệu JSON

❑ Đọc dữ liệu JSON từ tập tin .json

● Bước 1: Mở tập tin JSON

```
f = open(filename[, encoding = "utf-8"])
```

■ Ví dụ:

```
data_file = open("QLCT_1.json", encoding = "utf-8")
```

● Bước 2: Dùng phương thức **load()** để parse dữ liệu JSON từ tập tin đã mở

```
data = json.load(data_file)
```

Làm việc với dữ liệu JSON

❑ Đọc dữ liệu JSON từ tập tin .json

● Bước 3: Xử lý dữ liệu

■ Ví dụ: In dữ liệu `pprint(data)`

```
{'CONG_TY': [{'Dia_chi': '11223 Trần hưng Đạo Q.1 TP HCM ',  
              'Dien_thoai': '08-83222145',  
              'ID': 1,  
              'Mail': 'hhsom2016@gmail.com',  
              'Muc_Luong_Toai_thieu': 3500000,  
              'Ten': 'Công ty Dịch vụ Hoàng hôn Sớm',  
              'Tuoi_Toai_da': 50,  
              'Tuoi_Toai_thieu': 20}],  
'DON_VI': [{'ID': 1,  
             'ID_CHI_NHANH': 1,  
             'So_Nhan_vien': '14',  
             'Ten': 'Đơn vị A1',  
             'Ty_le': '14.58'},  
            {'ID': 2,  
             'ID_CHI_NHANH': 1,  
             'So_Nhan_vien': '15',  
             'Ten': 'Đơn vị A2',  
             'Ty_le': '15.62'},  
            {'ID': 3,  
             'ID_CHI_NHANH': 2,  
             'So_Nhan_vien': '14',  
             'Ten': 'Đơn vị B1',  
             'Ty_le': '14.58'},  
            {'ID': 4,  
             'ID_CHI_NHANH': 2,  
             'So_Nhan_vien': '20',  
             'Ten': 'Đơn vị B2',  
             'Ty_le': '20.83'}]}
```

Làm việc với dữ liệu JSON

❑ Ghi dữ liệu vào tập tin .json

- Ghi mới/ ghi đè: Mở tập tin .json, nếu chưa có tập tin này thì tạo tập tin mới và ghi nội dung vào.

```
f = open('tên_tập_tin.json', 'w')  
json.dump(dữ_liệu_theo_định_dạng_JSON, f[,  
indent=4])
```

Làm việc với dữ liệu JSON

```
import json
from textwrap import indent

data = {}
data['people'] = []
data['people'].append({
    'name': 'Thuy Phuong',
    'website': 'csc.edu.vn',
    'from': 'Trung tam tin hoc'
})
data['people'].append({
    'name': 'Larry',
    'website': 'google.com',
    'from': 'Michigan'
})
data['people'].append({
    'name': 'Tim',
    'website': 'apple.com',
    'from': 'Alabama'
})

f = open('people.json', 'w')
json.dump(data, f, indent=4)
```



```
1 {
2     "people": [
3         {
4             "website": "csc.edu.vn",
5             "name": "Thuy Phuong",
6             "from": "Trung tam tin hoc"
7         },
8         {
9             "website": "google.com",
10            "name": "Larry",
11            "from": "Michigan"
12        },
13        {
14            "website": "apple.com",
15            "name": "Tim",
16            "from": "Alabama"
17        }
18    ]
19 }
```

Làm việc với dữ liệu JSON

❑ Ghi dữ liệu vào tập tin .json

- Ghi thêm (append):

- Chuẩn bị dữ liệu thêm mới (theo định dạng một item mới kiểu JSON)
- Mở tập tin .json nguồn, load dữ liệu ra
- Gọi phương thức update(dữ liệu thêm mới) để đưa dữ liệu mới vào
- Mở tập tin .json nguồn/đích theo chế độ ghi
- Gọi phương thức dump() để ghi dữ liệu vào tập tin .json

Làm việc với dữ liệu JSON

```
company = {"name": "Trung tam tin hoc", "address": "137E Nguyen Chi Thanh"}
```

```
list = []
```

```
list.append(company)
```

```
new = {"company": list}
```

```
f = open('people.json')
```

```
data = json.load(f)
```

```
data.update(new)
```

```
f = open('people.json', 'w')
```

```
json.dump(data, f, indent=4)
```

```
people.json
1 {
2   "people": [
3     {
4       "website": "csc.edu.vn",
5       "name": "Thuy Phuong",
6       "from": "Trung tam tin hoc"
7     },
8     {
9       "website": "google.com",
10      "name": "Larry",
11      "from": "Michigan"
12    },
13    {
14      "website": "apple.com",
15      "name": "Tim",
16      "from": "Alabama"
17    }
18  ]
19 }
```

```
people.json
1 {
2   "company": [
3     {
4       "address": "137E Nguyen Chi Thanh",
5       "name": "Trung tam tin hoc"
6     }
7   ],
8   "people": [
9     {
10      "from": "Trung tam tin hoc",
11      "website": "csc.edu.vn",
12      "name": "Thuy Phuong"
13    },
14    {
15      "from": "Michigan",
16      "website": "google.com",
17      "name": "Larry"
18    },
19    {
20      "from": "Alabama",
21      "website": "apple.com",
22      "name": "Tim"
23    }
24  ]
25 }
```

