Lendo documentos

Arquivos JSON

Lendo documentos

Arquivos JSON

JSON (*JavaScript Object Notation* - Notação de Objetos JavaScript) é uma formatação leve de troca de dados. Para seres humanos, é fácil de ler e escrever. Para máquinas, é fácil de interpretar e gerar. Está baseado em um subconjunto da linguagem de programação *JavaScript, Standard ECMA-262* 3a Edição - Dezembro - 1999. *JSON* é em formato texto e completamente independente de linguagem, pois usa convenções que são familiares às linguagens *C* e familiares, incluindo *C++*, *C#*, *Java*, *JavaScript*, *Perl*, *Python* e muitas outras. Estas propriedades fazem com que *JSON* seja um formato ideal de troca de dados.

Lendo documentos

Arquivos JSON

JSON está constituído em duas estruturas:

- Uma coleção de pares nome/valor. Em várias linguagens, isto é caracterizado como um objeto, registro, estrutura, dicionário, tabela de hashes, lista chaveada, ou matrizes associativas.
- Uma lista ordenada de valores. Na maioria das linguagens, isto é caracterizado como uma matriz, vetor, lista ou sequência.

Estas são estruturas de dados universais. Virtualmente todas as linguagens de programação modernas as suportam, de uma forma ou de outra. É aceitável que um formato de troca de dados que seja independente de linguagem de programação se baseie nestas estruturas.

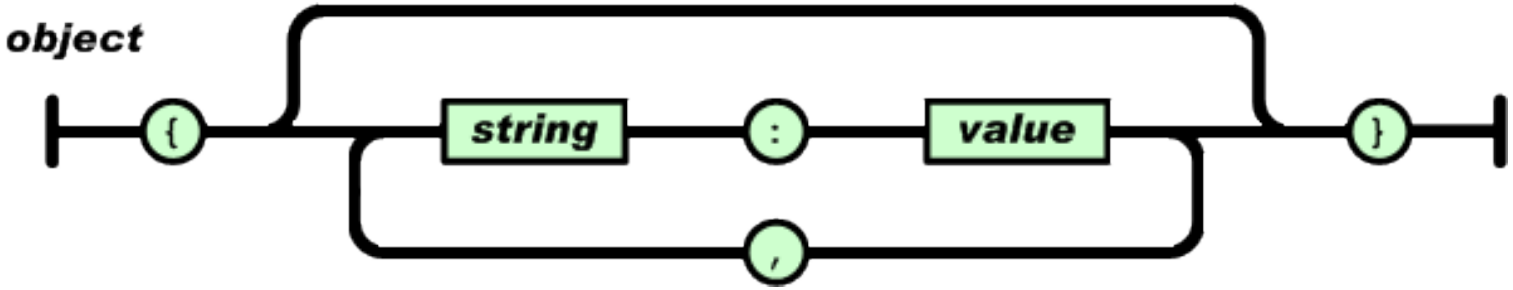
Lendo documentos

Arquivos JSON

Em *JSON*, os dados são apresentados desta forma:

Objeto

Um objeto é um conjunto desordenado de pares nome/valor. Um objeto começa com { (chave de abertura) e termina com } (chave de fechamento). Cada nome é seguido por : (dois pontos) e os pares nome/valor são seguidos por , (vírgula).



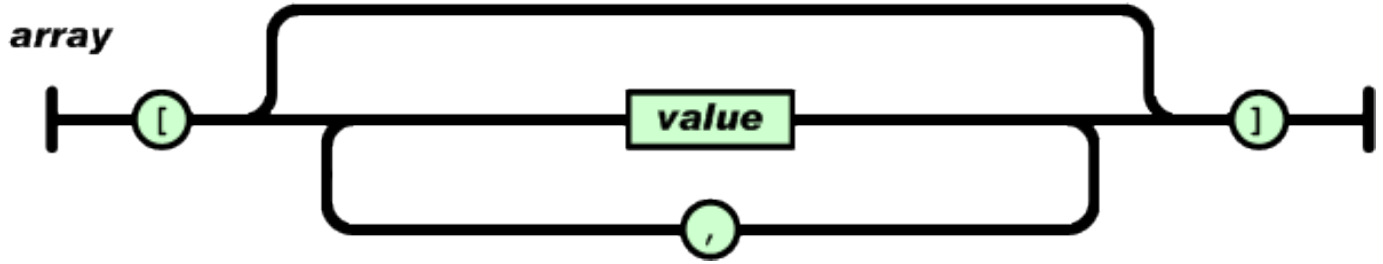
Ex.: {“chave”:valor, “chave”:valor, “chave”:valor}

Lendo documentos

Arquivos JSON

Array

Um array é uma coleção de valores ordenados. O array começa com [(colchete de abertura) e termina com] (colchete de fechamento). Os valores são separados por , (vírgula).



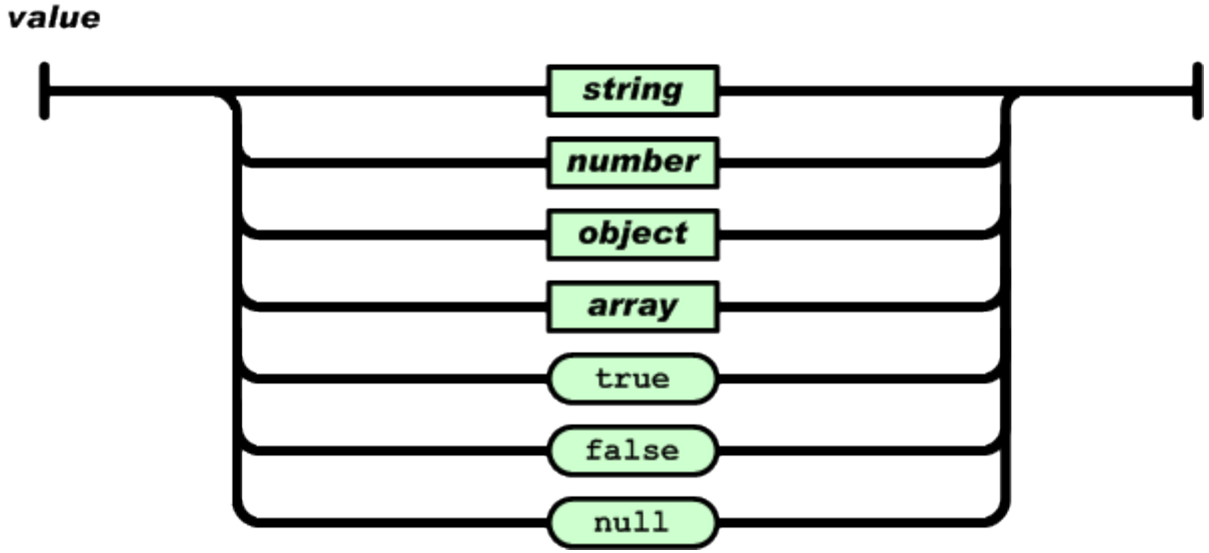
Ex.: [valor, valor, valor]

Lendo documentos

Arquivos JSON

Valor

Um valor (value) pode ser uma cadeia de caracteres (string), ou um número, ou true ou false, ou null, ou um objeto ou um array. Estas estruturas podem estar aninhadas.

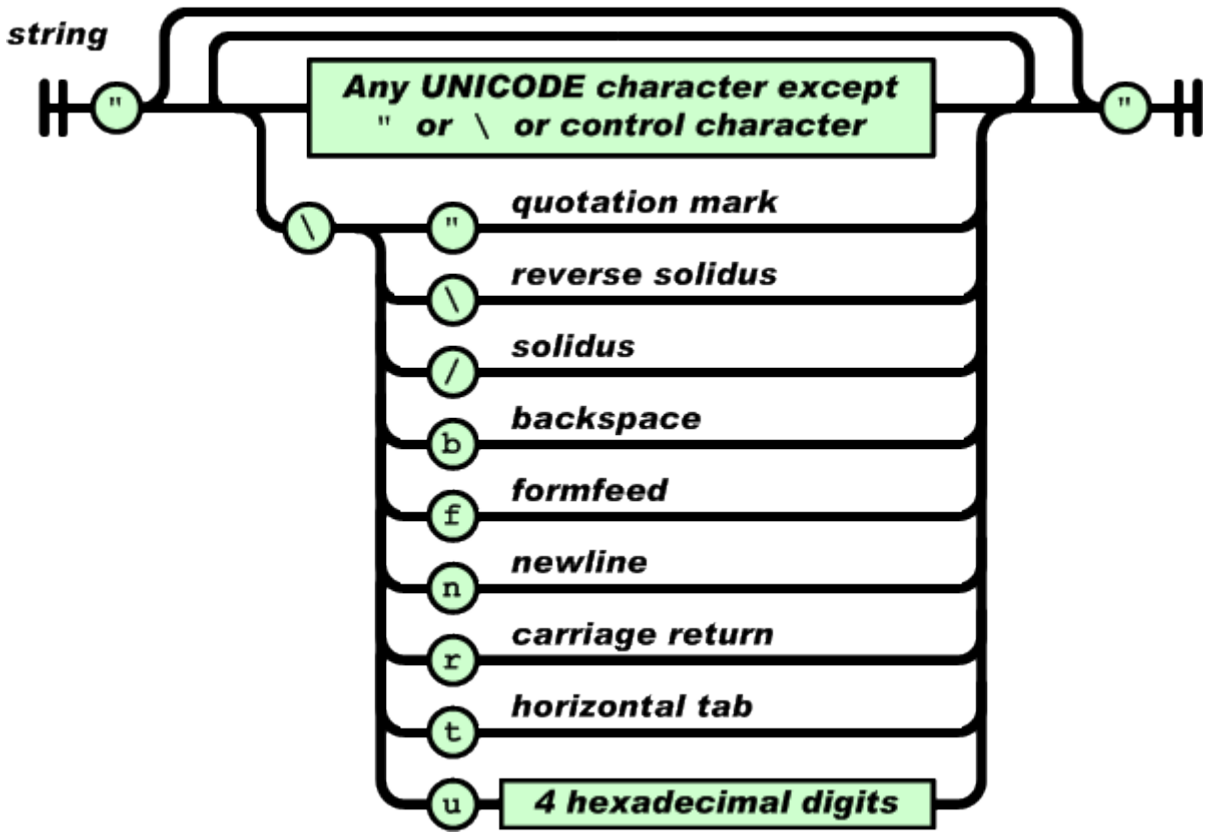


Lendo documentos

Arquivos JSON

String

Uma *string* é uma coleção de nenhum ou mais caracteres *Unicode*, envolvido entre aspas duplas usando barras invertidas como caracter de escape. Um caracter está representado como um simples caracter de *string*. Uma cadeia de caracteres é parecida com uma cadeia de caracteres em *C* ou *Java*.



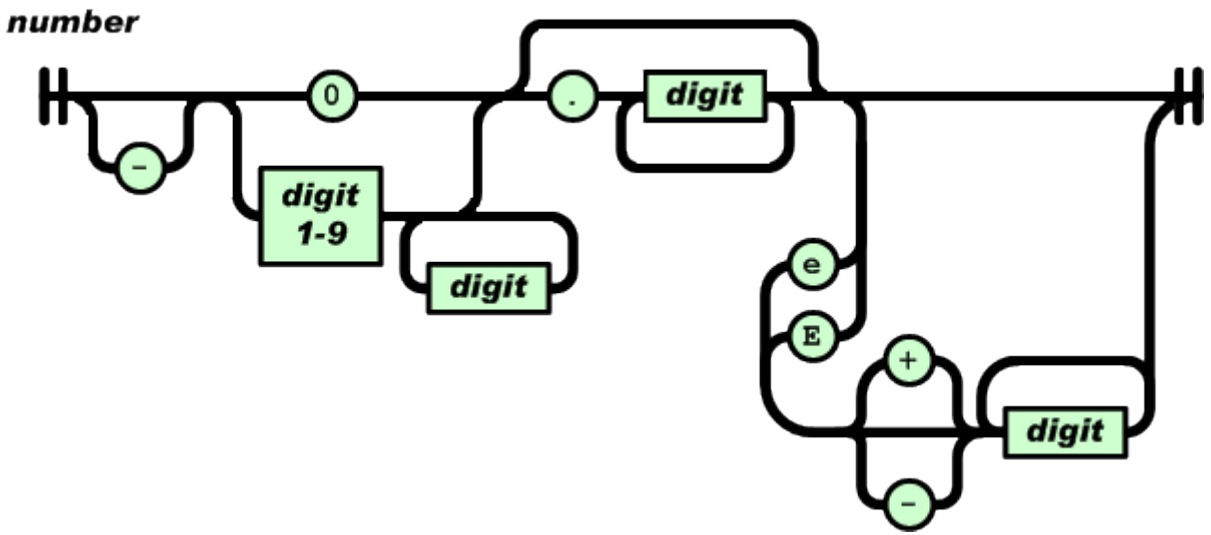
Lendo documentos

Arquivos JSON

Número

Um número é similar a um número em *C* ou *Java*, exceto quando não se usa os números *octais* ou *hexadecimais*.

Espaços em branco podem ser inseridos em qualquer parte dos símbolos.



Fonte:
https://www.json.org/json-pt.html

Lendo documentos

Arquivos JSON

Na seção “Usando APIs” já vimos que o Python possui um módulo em sua biblioteca padrão para trabalharmos com JSON.

Para utilizarmos este módulo, devemos importá-lo usando o comando:

import json

Lendo documentos

Arquivos JSON

Em nosso primeiro exemplo vamos criar um dicionário, realizar a serialização do dicionário para JSON com o método dumps e escrever o resultado em um arquivo chamado “exemplo01.json”.

O método dumps vai converter o dicionário em JSON.

```
import json

dados = {'Curso': 'Python Web Scraping',
         'Secao':'12',
         'Nome Secao': 'Lendo documentos',
         'Numero da Aula':'05',
         'Descricao da Aula':'Arquivos JSON'}
dados_json = json.dumps(dados)
with open('exemplo01.json', 'w') as arq:
    arq.write(json.dumps(dados))
```

Este é o conteúdo do arquivo exemplo01.json:

```
{"Curso": "Python Web Scraping",
"Secao": "12",
"Nome Secao": "Lendo documentos",
"Numero da Aula": "05",
"Descricao da Aula": "Arquivos JSON"}
```

Lendo documentos

Arquivos JSON

No segundo exemplo vamos abrir o arquivo “exemplo01.json” para leitura e usar o método *loads* para deserializar o texto do arquivo para *JSON*. Em seguida vamos imprimir o conteúdo da chave “Curso”, que é “Python Web Scraping”.

```
import json

with open('exemplo01.json', 'r') as arq:
    conteudo = arq.read()
    dados_json = json.loads(conteudo)
    print(dados_json['Curso'])
```

Lendo documentos

Arquivos JSON

No terceiro exemplo vamos abrir o arquivo “exemplo03.json” para leitura, vamos especificar a codificação “*utf-8*” para visualizarmos corretamente os caracteres acentuados. Vamos deserializar o conteúdo do arquivo para *JSON* e vamos realizar algumas impressões. Veja o conteúdo do arquivo:

```
[
  {
    "Curso": "Python Web Scraping","Secao": "12", "Nome Secao": "Lendo documentos",
    "Numero da Aula": "05", "Descricao da Aula": "Arquivos JSON"
  },
  {
    "Curso": "Python Para Todos","Secao": "08", "Nome Secao": "Estrutura de dados",
    "Numero da Aula": "01", "Descricao da Aula": "Listas"
  },
  {
    "Curso": "Usando Python para Análise de dados","Secao": "03", "Nome Secao": "Introdução ao NumPy",
    "Numero da Aula": "04", "Descricao da Aula": "Indexação booleana"
  },
  {
    "Curso": "Terminal Linux e Prompt de Comando do Windows","Secao": "04", "Nome Secao": "Comandos Básicos",
    "Numero da Aula": "03", "Descricao da Aula": "Comandos de manipulação de pastas"
  }
]
```

Lendo documentos

Arquivos JSON

Como vimos na introdução ao formato *JSON*, este arquivo possui um *array* contendo vários objetos e cada objeto *JSON* com vários conjuntos de chave/valor.

```
[
  {
    "Curso": "Python Web Scraping","Secao": "12", "Nome Secao": "Lendo documentos",
    "Numero da Aula": "05", "Descricao da Aula": "Arquivos JSON"
  },
  {
    "Curso": "Python Para Todos","Secao": "08", "Nome Secao": "Estrutura de dados",
    "Numero da Aula": "01", "Descricao da Aula": "Listas"
  },
  {
    "Curso": "Usando Python para Análise de dados","Secao": "03", "Nome Secao": "Introdução ao NumPy",
    "Numero da Aula": "04", "Descricao da Aula": "Indexação booleana"
  },
  {
    "Curso": "Terminal Linux e Prompt de Comando do Windows","Secao": "04", "Nome Secao": "Comandos Básicos",
    "Numero da Aula": "03", "Descricao da Aula": "Comandos de manipulação de pastas"
  }
]
```

Lendo documentos

Arquivos JSON

Podemos acessar os objetos através de seu índice (como se estivéssemos acessando elementos de uma lista).

Para acessar o primeiro objeto usamos dados_json[0], veja:

```
[
  {
0   "Curso": "Python Web Scraping","Secao": "12", "Nome Secao": "Lendo documentos",
    "Numero da Aula": "05", "Descricao da Aula": "Arquivos JSON"
  },
  {
1   "Curso": "Python Para Todos","Secao": "08", "Nome Secao": "Estrutura de dados",
    "Numero da Aula": "01", "Descricao da Aula": "Listas"
  },
  {
2   "Curso": "Usando Python para Análise de dados","Secao": "03", "Nome Secao": "Introdução ao NumPy",
    "Numero da Aula": "04", "Descricao da Aula": "Indexação booleana"
  },
  {
3   "Curso": "Terminal Linux e Prompt de Comando do Windows","Secao": "04", "Nome Secao": "Comandos Básicos",
    "Numero da Aula": "03", "Descricao da Aula": "Comandos de manipulação de pastas"
  }
]
```

Lendo documentos

Arquivos JSON

Veja o exemplo:

```
import json

with open('exemplo03.json', 'r', encoding='utf-8') as arq:
    conteudo = arq.read()
    print("*" * 50)
    print("Conteudo do arquivo:\n", conteudo)
    dados_json = json.loads(conteudo)
    print("*" * 50)
    print("JSON completo:\n", dados_json)
    print("*" * 50)
    print("Primeiro objeto (dados_json[0]):\n", dados_json[0])
    print("*" * 50)
    print("Loop em todos objetos:")
    for dados in dados_json:
        print("Curso:", dados["Curso"], "/", "Descrição da Aula:", dados['Descricao da Aula'])
    print("*" * 50)
```

Lendo documentos

Arquivos JSON

Veja o resultado:

```
**********************************************
Conteudo do arquivo:
 [
  {
   "Curso": "Python Web Scraping","Secao": "12", "Nome Secao": "Lendo documentos", "Numero da Aula": "05", "Descricao da Aula": "Arquivos JSON"
  },
  {
   "Curso": "Python Para Todos","Secao": "08", "Nome Secao": "Estrutura de dados", "Numero da Aula": "01", "Descricao da Aula": "Listas"
  },
  {
   "Curso": "Usando Python para Análise de dados","Secao": "03", "Nome Secao": "Introdução ao NumPy", "Numero da Aula": "04", "Descricao da Aula":
"Indexação booleana"
  },
  {
   "Curso": "Terminal Linux e Prompt de Comando do Windows","Secao": "04", "Nome Secao": "Comandos Básicos", "Numero da Aula": "03", "Descricao
da Aula": "Comandos de manipulação de pastas"
  }
]
...
```

Lendo documentos

Arquivos JSON

Veja o resultado:

```
...
**************************************************
JSON completo:
 [{'Curso': 'Python Web Scraping', 'Secao': '12', 'Nome Secao': 'Lendo documentos', 'Numero da Aula': '05', 'Descricao da Aula': 'Arquivos JSON'}, {'Curso': 'Python Para Todos', 'Secao': '08', 'Nome Secao': 'Estrutura de dados', 'Numero da Aula': '01', 'Descricao da Aula': 'Listas'}, {'Curso': 'Usando Python para Análise de dados', 'Secao': '03', 'Nome Secao': 'Introdução ao NumPy', 'Numero da Aula': '04', 'Descricao da Aula': 'Indexação booleana'}, {'Curso': 'Terminal Linux e Prompt de Comando do Windows', 'Secao': '04', 'Nome Secao': 'Comandos Básicos', 'Numero da Aula': '03', 'Descricao da Aula': 'Comandos de manipulação de pastas'}]
**************************************************
Primeiro objeto (dados_json[0]):
 {'Curso': 'Python Web Scraping', 'Secao': '12', 'Nome Secao': 'Lendo documentos', 'Numero da Aula': '05', 'Descricao da Aula': 'Arquivos JSON'}
**************************************************
Loop em todos objetos:
Curso: Python Web Scraping / Descrição da Aula: Arquivos JSON
Curso: Python Para Todos / Descrição da Aula: Listas
Curso: Usando Python para Análise de dados / Descrição da Aula: Indexação booleana
Curso: Terminal Linux e Prompt de Comando do Windows / Descrição da Aula: Comandos de manipulação de pastas
**************************************************
```

Lendo documentos

Arquivos JSON

Em nosso quarto exemplo vamos acessar um arquivo *JSON online* para obter o seu conteúdo, imprimindo em seguida.

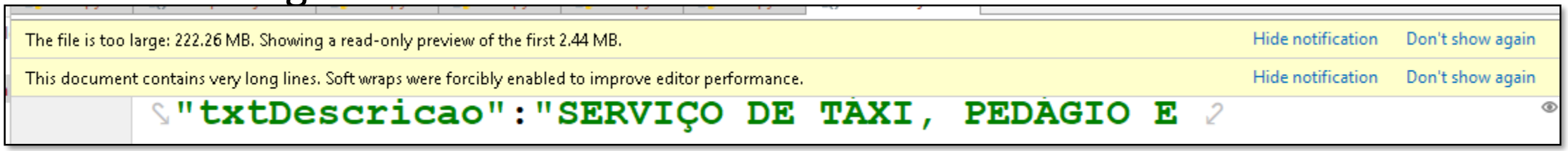
```
from urllib.request import urlopen
import json

dados = urlopen('http://dados.iffarroupilha.edu.br/'
                'api/v1/projetos.json').read()
dados = json.loads(dados)
print(dados)
```

Lendo documentos

Arquivos JSON

Em nosso quinto exemplo vamos trabalhar com um arquivo *JSON* um pouco maior. Estou disponibilizando o arquivo “Ano-2017.json” compactado como “Ano-2017.json.zip” que deve ser descompactado. Você não precisa abrir este arquivo no *PyCharm*, se for aberto diretamente no *PyCharm*, você receberá o seguinte aviso:



Neste aviso o *PyCharm* informa que o arquivo tem 222.26 MB e ele está exibindo apenas os primeiros 2.44 MB. Mas não será necessário abrir este arquivo em nenhum editor porque vamos manipulá-lo com o *Python*. Veja a seguir um exemplo do conteúdo do arquivo com 3 registros apenas (Ano-2017_exemplo.json).

Lendo documentos

Arquivos JSON


```
{
    "DESPESA":
    [
        {"indTipoDocumento":0,"numRessarcimento":5828,"ideDocumento":6266962,"nuCarteiraParlamentar":1,
            "datEmissao":"2017-04-26 00:00:00","vlrGlosa":"0","codLegislatura":55,"numMes":4,"numSubCota":1,
            "sgUF":"RR","idecadastro":178957,"numAno":2017,"txtNumero":"3592","sgPartido":"DEM",
            "txNomeParlamentar":"ABEL MESQUITA JR.","txtDescricao":"MANUTENÇÃO DE ESCRITÓRIO DE APOIO À ATIVIDADE PARLAMENTAR",
            "txtCNPJCPF":"12132854000100","txtFornecedor":"WM PAPELARIA E ESCRITÓRIO","numParcela":0,
            "nuLegislatura":2015,"txtDescricaoEspecificacao":"","vlrDocumento":"296","numEspecificacaoSubCota":0,
            "vlrLiquido":"296","nuDeputadoId":3074,"numLote":1377952,"vlrRestituicao":"0","txtPassageiro":null,"txtTrecho":null},
        {"indTipoDocumento":0,"numRessarcimento":5993,"ideDocumento":6408821,"nuCarteiraParlamentar":1,
            "datEmissao":"2017-10-05 00:00:00","vlrGlosa":"0","codLegislatura":55,"numMes":10,"numSubCota":1,
            "sgUF":"RR","idecadastro":178957,"numAno":2017,"txtNumero":"321","sgPartido":"DEM",
            "txNomeParlamentar":"ABEL MESQUITA JR.","txtDescricao":"MANUTENÇÃO DE ESCRITÓRIO DE APOIO À ATIVIDADE PARLAMENTAR",
            "txtCNPJCPF":"12132854000100","txtFornecedor":"WMS COMERCIO DE ARTIGOS DE PAPELARIA LTDA - ME","numParcela":0,
            "nuLegislatura":2015,"txtDescricaoEspecificacao":"","vlrDocumento":"175","numEspecificacaoSubCota":0,
            "vlrLiquido":"175","nuDeputadoId":3074,"numLote":1430312,"vlrRestituicao":"0","txtPassageiro":null,"txtTrecho":null},
        {"indTipoDocumento":0,"numRessarcimento":6041,"ideDocumento":6450961,"nuCarteiraParlamentar":1,
            "datEmissao":"2017-10-23 00:00:00","vlrGlosa":"0","codLegislatura":55,"numMes":10,"numSubCota":3,
            "sgUF":"RR","idecadastro":178957,"numAno":2017,"txtNumero":"389857","sgPartido":"DEM",
            "txNomeParlamentar":"ABEL MESQUITA JR.","txtDescricao":"COMBUSTÍVEIS E LUBRIFICANTES.",
            "txtCNPJCPF":"00692418000107","txtFornecedor":"AUTO POSTO CINCO ESTRELAS LTDA","numParcela":0,
            "nuLegislatura":2015,"txtDescricaoEspecificacao":"Veículos Automotores","vlrDocumento":"100","numEspecificacaoSubCota":1,
            "vlrLiquido":"100","nuDeputadoId":3074,"numLote":1446095,"vlrRestituicao":"0","txtPassageiro":null,"txtTrecho":null}
    ]
}
```

Lendo documentos

Arquivos JSON

Observe que nós temos um objeto *JSON* com a chave “DESPESA”:

{“DESPESA”: VALOR}

Sendo “VALOR” um array com vários objetos JSON, contendo, cada objeto, vários pares de chave:valor.


```
{
    "DESPESA":
    [
        {"indTipoDocumento":0,"numRessarcimento":5828,"ideDocumento":6266962,"nuCarteiraParlamentar":1,
            ...
            "vlrLiquido":"296","nuDeputadoId":3074,"numLote":1377952,"vlrRestituicao":"0","txtPassageiro":null,"txtTrecho":null},
        {"indTipoDocumento":0,"numRessarcimento":5993,"ideDocumento":6408821,"nuCarteiraParlamentar":1,
            ...
            "vlrLiquido":"175","nuDeputadoId":3074,"numLote":1430312,"vlrRestituicao":"0","txtPassageiro":null,"txtTrecho":null},
        {"indTipoDocumento":0,"numRessarcimento":6041,"ideDocumento":6450961,"nuCarteiraParlamentar":1,
           ...
            "vlrLiquido":"100","nuDeputadoId":3074,"numLote":1446095,"vlrRestituicao":"0","txtPassageiro":null,"txtTrecho":null}
    ]
}
```

Lendo documentos

Arquivos JSON

O conteúdo destacado abaixo é o conteúdo de índice zero e pode ser obtido com:

print(dados_json["DESPESA"][0])


```
{
    "DESPESA":
    [
        {"indTipoDocumento":0,"numRessarcimento":5828,"ideDocumento":6266962,"nuCarteiraParlamentar":1,
            ...
            "vlrLiquido":"296","nuDeputadoId":3074,"numLote":1377952,"vlrRestituicao":"0","txtPassageiro":null,"txtTrecho":null
        },
        {"indTipoDocumento":0,"numRessarcimento":5993,"ideDocumento":6408821,"nuCarteiraParlamentar":1,
            ...
            "vlrLiquido":"175","nuDeputadoId":3074,"numLote":1430312,"vlrRestituicao":"0","txtPassageiro":null,"txtTrecho":null
        },
        {"indTipoDocumento":0,"numRessarcimento":6041,"ideDocumento":6450961,"nuCarteiraParlamentar":1,
           ...
            "vlrLiquido":"100","nuDeputadoId":3074,"numLote":1446095,"vlrRestituicao":"0","txtPassageiro":null,"txtTrecho":null
        }
    ]
}
```

Lendo documentos

Arquivos JSON

Para obter um determinado valor dentro do *JSON* do objeto de índice zero, devemos fazer desta forma:

print(dados_json["DESPESA"][0]["elemento"])

Por exemplo, pra retornar o valor líquido (vlrLiquido), devemos fazer desta forma:

print(dados_json["DESPESA"][0]["vlrLiquido"])


```
{
    "DESPESA":
    [
        {"indTipoDocumento":0,"numRessarcimento":5828,"ideDocumento":6266962,"nuCarteiraParlamentar":1,
            ...
            "vlrLiquido":"296","nuDeputadoId":3074,"numLote":1377952,"vlrRestituicao":"0","txtPassageiro":null,"txtTrecho":null
        },
...     ]
}
```

Lendo documentos

Arquivos JSON

Veja o programa do quinto exemplo:

```
"""
Neste exemplo vamos trabalhar com os dados obtidos
no portal da Câmara dos Deputados:
http://www.camara.leg.br/cotas/Ano-2017.json.zip
"""
import json

with open('Ano-2017/Ano-2017_exemplo.json', 'r', encoding='utf-8') as arq:
    dados_json = json.load(arq)
    print(dados_json["DESPESA"][0])
    print(dados_json["DESPESA"][0]["vlrLiquido"])
```

Lendo documentos

Arquivos JSON

Veja o resultado:

```
Objeto índice zero:
 {'indTipoDocumento': 0, 'numRessarcimento': 5828, 'ideDocumento': 6266962,
'nuCarteiraParlamentar': 1, 'datEmissao': '2017-04-26 00:00:00', 'vlrGlosa': '0', 'codLegislatura': 55,
'numMes': 4, 'numSubCota': 1, 'sgUF': 'RR', 'idecadastro': 178957, 'numAno': 2017, 'txtNumero':
'3592', 'sgPartido': 'DEM', 'txNomeParlamentar': 'ABEL MESQUITA JR.', 'txtDescricao':
'MANUTENÇÃO DE ESCRITÓRIO DE APOIO À ATIVIDADE PARLAMENTAR', 'txtCNPJCPF':
'12132854000100', 'txtFornecedor': 'WM PAPELARIA E ESCRITÓRIO', 'numParcela': 0, 'nuLegislatura':
2015, 'txtDescricaoEspecificacao': '', 'vlrDocumento': '296', 'numEspecificacaoSubCota': 0,
'vlrLiquido': '296', 'nuDeputadoId': 3074, 'numLote': 1377952, 'vlrRestituicao': '0', 'txtPassageiro':
None, 'txtTrecho': None}
Valor líquido do objeto de índice zero:
 296
```

Lendo documentos

Arquivos JSON

No sexto exemplo vamos percorrer todos os objetos, realizar a soma do valor líquido e imprimir a soma no final.

```
import json

soma_valor_liquido = 0.0

with open('Ano-2017/Ano-2017.json', 'r', encoding='utf-8') as arq:
    dados_json = json.load(arq)
    for x in range(0, len(dados_json["DESPESA"])):
        valor_liquido = float(dados_json["DESPESA"][x]["vlrLiquido"].replace(",","."))
        soma_valor_liquido += valor_liquido
print("Soma do valor líquido: %.2f" % soma_valor_liquido)
```

FIM