

# Introdução à programação para não programadores

## Aula 3



Esta obra foi licenciada sob uma Licença  
Creative Commons Atribuição 3.0 Não  
Adaptada.

# Dicionários

- Estrutura de dados chave/valor
- Eficiente na recuperação do valor

```
1 pessoa = {  
2     'nome': 'Gustavo Fonseca',  
3     'idade': 28,  
4     'telefones': ['(11)99999999', '(11)88888888'],  
5 }
```

A ordem **não** é preservada

[ ] são usados para recuperar os valores

```
PAT113-SCIELO:src gustavofonseca (master) python -i dict_basico.py  
>>> pessoa  
{'idade': 28, 'telefones': ['(11)99999999', '(11)88888888'], 'nome': 'Gustavo Fonseca'}  
>>> pessoa['nome']  
'Gustavo Fonseca'  
>>> pessoa['telefones'][0]  
'(11)99999999'  
>>> for tel in pessoa['telefones']:  
...     print tel  
...  
(11)99999999  
(11)88888888  
>>> █
```



# JSON – Javascript Object Notation

- Formato, baseado em texto, para intercâmbio de dados
- Bastante parecido com **dict** do Python

```
1 {  
2     "nome": "Gustavo Fonseca",  
3     "idade": 28,  
4     "telefones": ["(11) 99999999", "(11) 88888888"]  
5 }
```

# JSON e Python

Comentário funcional que informa ao interpretador qual o **charset** deste **código-fonte**

Biblioteca que sabe lidar com JSON

```
1 # coding: utf-8
2 import json
3
4 dados_json = """
5 {
6     "nome": "Gustavo Fonseca",
7     "idade": 28,
8     "telefones": ["(11)99999999", "(11)88888888"]
9 }
10 """
11
12 dados_dict = json.loads(dados_json)
13
14 print dados_dict
15 print u'Meu nome é', dados_dict['nome']
```

Função que recebe uma string no formato JSON e retorna sua representação em estruturas de dados do Python

Marcador de texto **Unicode**



```
PAT113-SCIELO:src gustavofonseca (master) python json_dict.py
{'idade': 28, 'telefones': [u'(11)99999999', u'(11)88888888'], 'nome': u'Gustavo Fonseca'}
Meu nome é Gustavo Fonseca
```

# Leitura de arquivos

```
1 # coding: utf-8
2 import json
3
4 dados_json = open('dados.json')
5
6 dados_dict = json.loads(dados_json.read())
7
8 print dados_dict
9 print u'Meu nome é', dados_dict['nome']
```

Abre o arquivo para leitura ou gravação. Retorna um **file descriptor\***.

O método **read** lê os dados associados ao **file descriptor**

A biblioteca **json** possui a função **load** que recebe **file descriptors** diretamente. Ex.:

```
dados_dict = json.load(dados_json)
```



\*[http://en.wikipedia.org/wiki/File\\_descriptor](http://en.wikipedia.org/wiki/File_descriptor)

# Acessando dados na Web

```
1 # coding: utf-8
2 import urllib
3 import json
4
5
6 API_URL = 'http://search.twitter.com/search.json?q=%23'
7 HASHTAG = 'scielo'
8
9 # pesquisando no Twitter pela hashtag
10 response = urllib.urlopen(API_URL + HASHTAG)
11
12 # transformando o JSON retornado em estruturas nativas do Python
13 tweets = json.loads(response.read())
14
15 # geração da saída HTML
16 html = u'<table>'
17
18 for tweet in tweets['results']:
19     html += u'<tr>'
20     html += u'<td></td>'
21     html += u'<td>' + tweet['from_user_name'] + u'</td>'
22     html += u'<td>' + tweet['text'] + u'</td>'
23     html += u'<td>' + tweet['created_at'] + u'</td>'
24     html += u'</tr>'
25
26 html += '</table>'
27
28 # saída dos dados utilizando a codificação utf-8
29 print html.encode('utf-8')
```

Biblioteca para requisições na web.

Efetua uma requisição HTTP e retorna um **file descriptor**

**bla += 'foo'** é um açúcar sintático para **bla = bla + 'foo'**

Codifica os dados em **utf-8** e imprime na saída padrão

SciELO

# Resumo

- Geral
  - JSON é um formato de intercâmbio de dados, baseado em texto, comumente utilizado em APIs públicas
  - O código-fonte deve informar ao interpretador qual conjunto de caracteres foi utilizado na sua escrita
- Estruturas de dados
  - **dict** – Dicionários chave/valor
  - Dicionários são mutáveis
  - Dicionários não preservam a ordem de inserção dos itens



# Resumo

- Tipos de dados
  - **unicode** – Sequência de caracteres textuais, definida pelo marcador **u''**, chamado **unicode literal marker**
  - **File-like objects** – Objetos que “se parecem” com **file descriptors**, pois possuem o método **read** para recuperar seu conteúdo. Primeiro contato com **duck-typing**



# Links para APIs

- <https://dev.twitter.com/docs/api/1>
- <http://docs.scielo.org>

