Introdução à programação para não programadores
Aula 3

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
>>>
```

SciELO

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 }
```

http://www.json.org

JSON e Python

```
 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']
```

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

Biblioteca que sabe lidar com JSON

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
{u'idade': 28, u'telefones': [u'(11)99999999', u'(11)88888888'], u'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)
```

SciELO

*http://en.wikipedia.org/wiki/File_descriptor

Mudança de planos... vamos acessar o SciELO! ;)

http://ref.scielo.org/ddkpmx

Acessando dados na Web

```
 1 # coding: utf-8
 2 import urllib
 3 import json
 4
 5 username = raw_input('Digite seu nome de usuário: ')
 6 user_token = raw_input('Digite seu token de acesso: ')
 7
 8 API_URL = 'http://manager.scielo.org/api/v1/journals/?format=json&username=
%s&api_key=%s&limit=0' % (username, user_token)
 9
10 # acessando a lista de periódicos
11 response = urllib.urlopen(API_URL)
12
13 # transformando o JSON retornado em estruturas nativas do Python
14 journals = json.loads(response.read())
15
16 # geração da saí-da HTML
17 html = u'<table>'
18
19 for journal in journals['objects']:
20     html += u'<tr>'
21     html += u'<td>%s</td>' % journal['acronym']
22     html += u'<td>%s</td>' % journal['short_title']
23     html += u'<td>%s</td>' % journal['use_license']['license_code']
24     html += u'<td>%s</td>' % ', '.join(journal['study_areas'])
25     html += u'</tr>'
26
27 html += '</table>'
28
29 # saída dos dados utilizando a codificação utf-8
30 out = open('journals.html', 'w')
31 out.write(html.encode('utf-8'))
32 out.close()
33
```

Biblioteca para requisições na web.

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

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

Abre um arquivo para escrita

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

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 ítens

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.

Links para APIs

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