

Fundamentos da Programação LEIC/LETI

Aula 13

Dicionários

O tipo dicionário. Exemplos. Frequência de letras num texto. Dicionários de dicionários.

Alberto Abad, Tagus Park, IST, 2021-22

Dicionários

O tipo Dicionário em Python

- Os dicionários, também conhecidos como mapas (*mapping*), são entidades formadas por um conjunto de pares *chave/valor*, em que cada chave é associada a um valor.
 - Outros nomes: Associative arrays (Perl/PHP), HashMaps (Java).
- Em Python existe o tipo `dict`, um tipo **mutável** que representa um dicionário. Em BNF:

```
<dicionário> ::= {} | {<pares>}
<pares> ::= <par> | <par>, <pares>
<par> ::= <chave> : <valor>
<chave> ::= <expressão>
<valor> ::= <expressão> | <tuplo> | <lista> | <dicionário>
```

- Em Python os dicionários são como listas em que:
 - Os elementos também são referenciados por indexação, mas são acedidos por chave
 - As chaves tem de ser de um tipo imutável (e são únicas)
 - Ao contrário das listas, os elementos de um dicionário não estão ordenados

Dicionários

Dicionários em Python - Exemplos

```
>>> vazio = {} # definir dicionarios vazios
>>> outrovazio = dict()
>>> notasFP = {'ist40000':14, 'ist40001':9, 'ist40002':17} #init dicio
narios
>>> notasFP['ist40000'] #indexar elementos
>>> notasFP['ist40008'] #e se não existem!?
>>> notasFP['ist40001'] += 1 #alterar elementos

>>> notasFP['ist40001'] = [9, 10] # o tipo dos valores heterogêneos
>>> notasFP['ist40001'][0]
>>> notasFP['ist40007'] = 12 #inserir novos elemetos
>>> notasFP[('ist40008', 'Primeiro exame')] = 13 # tipo chaves hegerog
éneo
>>> outrasnotas = dict((('ist40000', 14),('ist40001',9),('ist40002',17)
))
```

In [101]:

```
Out[101]: {'ist40000': 14,
'ist40001': 9,
'ist40002': 17,
'ist30000': 15,
'ist400020': 'NA',
(400021, '1a epoca'): 13}
```

Dicionários

Operações básicas com dicionários

Operação	Tipo dos argumentos	Valor
<code>del(d[e])</code>	Elemento de dicionário	Remove do dicionário <i>d</i> o elemento com índice <i>e</i> .
<code>c in d</code>	Chave e dicionário	True se a chave <i>c</i> pertence ao dicionário <i>d</i> ; False em caso contrário.
<code>c not in d</code>	Chave e dicionário	A negação do resultado da operação <code>c in d</code> .
<code>len(d)</code>	Dicionário	O número de elementos do dicionário <i>d</i> .

Dicionários

Operações básicas com dicionários - Exemplos

```
>>> notasFP = {'ist40000':14, 'ist40001':9, 'ist40002':17}
>>> 'ist40000' in notasFP # True or False?
>>> 14 in notasFP        # True or False?
>>> del notasFP['ist40000'] #apagar elementos
>>> 'ist40000' in notasFP
>>> 'ist40000' not in notasFP
>>> len(notasFP)
```

```
In [104]: notasFP = {'ist40000':14, 'ist40001':9, 'ist40002':17}
notasFP['ist40000']
```

```
Out[104]: 14
```

Dicionários

Mais operações com dicionários

Operação	Tipo dos	Valor
<code>d[k] = v</code>	Dicionário, hashable, universal	Define <code>d[k]</code> igual ao valor <code>v</code> .
<code>del d[k]</code>	Dicionário, hashable	Remove a entrada <code>k</code> do dicionário <code>d</code> .
<code>d.clear()</code>	Dicionário	Remove todos os itens do dicionário <code>d</code> .
<code>d.copy()</code>	Dicionário	Devolve uma cópia (<i>shallow</i>) do dicionário <code>d</code> .
<code>d.fromkeys(l[, v])</code>	Dicionário, iterável, universal (opcional)	Cria um novo dicionário com as chaves os elementos de <code>l</code> e valor <code>v</code> .
<code>d1.update(d2)</code>	Dicionário, dicionário	Atualiza <code>d1</code> com os pares chave/valor de <code>d2</code> , sobrescrevendo chaves existentes.
<code>d.get(k[, v])</code>	Dicionário, hashable, universal	Devolve o valor da entrada <code>k</code> se existe, caso contrário devolve <code>v</code> .
<code>d.setdefault(k[, v])</code>	Dicionário, hashable, universal	Devolve o valor da entrada <code>k</code> se existe, caso contrário insere <code>d[k]</code> igual a <code>v</code> e devolve <code>v</code> .
<code>d.keys()</code>	Dicionário	Devolve um objeto <code>view</code> contendo todas as chaves do dicionário <code>d</code> .
<code>d.values()</code>	Dicionário	Devolve um objeto <code>view</code> contendo todos os valores do dicionário <code>d</code> .
<code>d.items()</code>	Dicionário	Devolve um objeto <code>view</code> contendo todos os pares chave/valor do dicionário <code>d</code> .

Referência: <https://docs.python.org/3/library/stdtypes.html#mapping-types-dict>
(<https://docs.python.org/3/library/stdtypes.html#mapping-types-dict>)

Dicionários

Iterar dicionários

- Iterar um dicionário (sobre as chaves):

```
for key in notasFP:  
    print(key, "--> ", notasFP[key])
```

- Iterar chaves, valores ou itens:
 - `notasFP.keys()`, `notasFP.values()` e `notasFP.items()`

```
In [115]: notasFP = {'ist40000':14, 'ist40001':9, 'ist40002':17}  
for item in notasFP.items():  
    print(item[0], '->', notasFP[item[0]])  
tuple(notasFP.items())  
notasFP.get('ola')
```

```
ist40000 -> 14  
ist40001 -> 9  
ist40002 -> 17
```

Dicionários

Sobre a mutabilidade dos dicionários

<http://pythontutor.com/visualize.html> (<http://pythontutor.com/visualize.html>)

The image shows a screenshot of the Python Tutor website. On the left, there is a code editor with Python 3.6 code. The code defines a dictionary `d1`, creates a shallow copy `d3`, and then mutates `d1` by adding a new key-value pair. It then prints `d1`, `d2`, and `d3`. The code is as follows:

```
Python 3.6  
1 from pprint import pprint  
2  
3 d1 = {'um': 1, 2: ['dois']}  
4 d2 = d1  
5 d3 = d1.copy() #shallow copy  
6  
7 d2[3] = ['tres']  
8 d3[4] = ['cuatro']  
9  
10 d3[2] = ['novodois']  
11 d1[1] += ['outro']  
12  
13 pprint(d1)  
14 pprint(d2)  
15 pprint(d3)
```

On the right, there is a memory graph showing the state of memory. It includes a 'Global frame' containing `pprint` (imported object), `d1`, `d2`, and `d3`. The 'Objects' section shows the internal structure of these dictionaries. `d1` is a dict with keys 1 and 2, and values ['um', 'outro'] and ['dois']. `d2` is a dict with keys 0, 1, 2, and 3, and values ['tres'], ['dois'], ['dois'], and ['tres']. `d3` is a dict with keys 0, 1, 2, 3, and 4, and values ['tres'], ['tres'], ['cuatro'], ['tres'], and ['novodois']. Arrows indicate the references between the variables in the frames and the objects in memory.

Dicionários

Exemplo 1: Reverse Lookup

- Encontrar a chave que corresponde a um valor:
 - Pode existir mais de uma e então voltamos uma qualquer
 - Pode não existir LookupError
- Alterar para retornar lista com todas as chaves

```
In [121]: def reverse_lookup(d, value):
          lista = []
          for key in d:
              if d[key] == value:
                  lista.append(key)

          return lista

d = {'1': 'um', '2': 'dois', '3': 'tres', '4': 'dois'}
for key in reverse_lookup(d, 'dois'):
    del d[key]
d
```

```
Out[121]: {'1': 'um', '3': 'tres'}
```

Dicionários

Exemplo 2: Contagem de símbolos/letras

- Programa que conta o número de ocorrências de cada símbolo em uma sequência de caracteres.
- Alterar para não diferenciar minúsculas e maiúsculas.
- Alterar para ignorar pontuação e espaços em branco.
- Mostrar o resultado, em que ordem aparece?

```
In [125]: import string
def symbolstable(s):
    table = {}

    for c in s.lower():
        table[c] = table[c] + 1 if c in table else 1

    return table

# Lorem ipsum
res = symbolstable("Sed ut perspiciatis unde omnis iste natus error
sit voluptatem accusantium doloremque laudantium, totam rem aperiam
, eaque ipsa quae ab illo inventore veritatis et quasi architecto b
eatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia v
oluptas sit aspernatur aut odit aut fugit, sed quia consequuntur ma
gni dolores eos qui ratione voluptatem sequi nesciunt.")
for k in res:
    print(k, '-->', res[k])
```

```
s --> 22
e --> 34
d --> 8
  --> 53
u --> 28
t --> 35
p --> 11
r --> 15
i --> 31
c --> 9
a --> 34
n --> 17
o --> 20
m --> 14
v --> 7
l --> 10
q --> 9
, --> 3
b --> 3
h --> 1
x --> 1
. --> 2
f --> 1
g --> 2
```

Dicionários

Exemplo 2 cont.: Contagem de símbolos/letras

- Mostrar/printar o resultado, em que ordem aparecem?
- Como mostrar em ordem?
- Como contar o total de símbolos?
- Como alterar (ou gerar um novo dicionário) com as frequências?
- Como ordenar por frequência (com sorted)?

```
In [51]: res = symbolstable("Sed ut perspiciatis unde omnis iste natus error  
sit voluptatem accusantium doloremque laudantium, totam rem aperiam  
, eaque ipsa quae ab illo inventore veritatis et quasi architecto b  
eatae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia v  
oluptas sit aspernatur aut odit aut fugit, sed quia consequuntur ma  
gni dolores eos qui ratione voluptatem sequi nesciunt.")  
  
# How can we print the results in alphabetical order?  
# for k in sorted(res, reverse=False):  
#     print(k, '->', res[k])  
  
# Count the total number of symbols  
soma = 0  
for v in res.values():  
    soma += v  
  
# with sum  
# print(soma, sum(res.values()))  
  
# Create a new dict (or modify the original one) to store the frequ  
encies  
newres = {}  
for k in res:  
    newres[k] = res[k]/soma  
newres = dict([(k,v/soma) for k, v in res.items()])  
  
for f,k in sorted([(f,k) for k,f in newres.items()], reverse=True):  
    print(k, '->', f)
```

t --> 0.11217948717948718
e --> 0.10897435897435898
a --> 0.10897435897435898
i --> 0.09935897435897435
u --> 0.08974358974358974
s --> 0.07051282051282051
o --> 0.0641025641025641
n --> 0.05448717948717949
r --> 0.04807692307692308
m --> 0.04487179487179487
p --> 0.035256410256410256
l --> 0.03205128205128205
q --> 0.028846153846153848
c --> 0.028846153846153848
d --> 0.02564102564102564
v --> 0.022435897435897436
b --> 0.009615384615384616
g --> 0.00641025641025641
x --> 0.003205128205128205
h --> 0.003205128205128205
f --> 0.003205128205128205

Dicionários

Exemplo 3: Contagem de palavras


```

In [198]: import string

def wordstable(s):
    table = {}

    toignore = string.punctuation + string.whitespace

    s = s.lower()

    # if the string does not end with a punctuation/whitespace, I add one
    # this is done to guarantee that I don't miss the last word
    if s[-1] not in toignore:
        s += '.'

    start = 0 #store the position where the current word starts
    for i in range(len(s)):
        if s[i] in toignore: ## search for possible word ending
            if i > start: # if it is only one character, it must be
                a punctuation/whitespace symbol
                    # and needs to be escaped
                    # otherwise, it is a valid word to add to
our table
                table[s[start:i]] = 1 if s[start:i] not in table else
table[s[start:i]] + 1
                start = i + 1 # reset the start of next word

    return table

res = wordstable("Sed ut perspiciatis unde omnis iste natus error s
it voluptatem accusantium doloremque laudantium, totam rem aperiam,
eaque ipsa quae ab illo inventore veritatis et quasi architecto bea
tae vitae dicta sunt explicabo. Nemo enim ipsam voluptatem quia vol
uptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni
dolores eos qui ratione voluptatem sequi nesciunt.")
print(res)

```

```

{'sed': 2, 'ut': 1, 'perspiciatis': 1, 'unde': 1, 'omnis': 1, 'iste': 1, 'natus': 1, 'error': 1, 'sit': 2, 'voluptatem': 3, 'accusantium': 1, 'doloremque': 1, 'laudantium': 1, 'totam': 1, 'rem': 1, 'aperiam': 1, 'eaque': 1, 'ipsa': 1, 'quae': 1, 'ab': 1, 'illo': 1, 'inventore': 1, 'veritatis': 1, 'et': 1, 'quasi': 1, 'architecto': 1, 'beatae': 1, 'vitae': 1, 'dicta': 1, 'sunt': 1, 'explicabo': 1, 'nemo': 1, 'enim': 1, 'ipsam': 1, 'quia': 2, 'voluptas': 1, 'aspernatur': 1, 'aut': 2, 'odit': 1, 'fugit': 1, 'consequuntur': 1, 'magni': 1, 'dolores': 1, 'eos': 1, 'qui': 1, 'ratione': 1, 'sequi': 1, 'nesciunt': 1}

```

Dicionários

Exemplo 4: Dicionário de dicionários

- Representar fichas académicas de alunos duma universidade: número do aluno (index), nome (primeiro nome e apelido) e disciplinas que frequentou, contendo ano(s) letivo(s) e classificação.

```
In [1]: from pprint import pprint
alunos = {1000:
          {'nome': {'primeiro nome': 'John', 'apelido': 'Boy'},
           'disc': {'FP': {'2017-2018': 8, '2018-2019': 13}, 'IEI': {'2017-2018': 17}}}}
alunos[1001] = {'nome': {'primeiro nome': 'Sarah', 'apelido': 'Girl'},
               'disc': {'FP': {'2017-2018': 18}, 'IEI': {'2017-2018': 15}}}
alunos[1000]['disc']['FP']
```

```
Out[1]: {'2017-2018': 8, '2018-2019': 13}
```

- Função que recebe estrutura como a anterior e uma pauta e insere:
 - Ex: pauta = ('PO', '2018-2019', ((1000, 'RE'), (1001, 15)))

```
In [2]: def insere_notas(alunos, pauta):
        disc = pauta[0]
        ano = pauta[1]

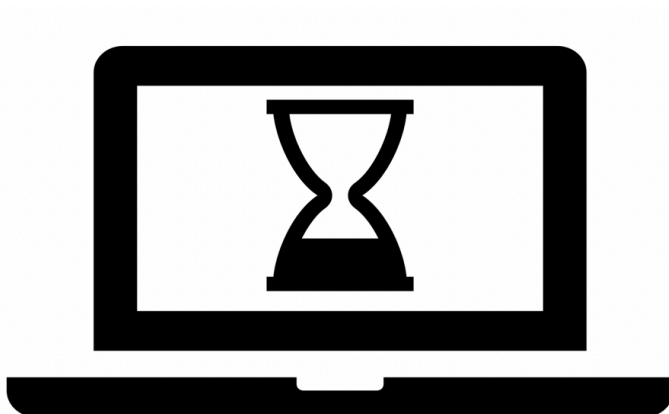
        for aluno, nota in pauta[2]:
            if aluno not in alunos:
                raise LookupError("o aluno não existe")
            if disc not in alunos[aluno]['disc']: # primeira inscricao
                alunos[aluno]['disc'][disc] = {ano : nota}
            else:
                alunos[aluno]['disc'][disc][ano] = nota

        pauta = ('PO', '2018-2019', ((1000, 'RE'), (1001, 15)))
        insere_notas(alunos, pauta)
        pprint(alunos)
```

```
{1000: {'disc': {'FP': {'2017-2018': 8, '2018-2019': 13},
                'IEI': {'2017-2018': 17},
                'PO': {'2018-2019': 'RE'}},
        'nome': {'apelido': 'Boy', 'primeiro nome': 'John'}},
 1001: {'disc': {'FP': {'2017-2018': 18},
                'IEI': {'2017-2018': 15},
                'PO': {'2018-2019': 15}},
        'nome': {'apelido': 'Girl', 'primeiro nome': 'Sarah'}}}
```

Dicionários - Tarefas

- Estudar matéria de dicionários:
 - Completar exemplos
 - Explorar um outro tipo relacionado do Python: set
- Próxima aula ==> Projeto e revisões matéria anterior



In []: