

25 Reunião 25: 24/NOV/2020



Figure 1: Hã? Quem? Eu?



Figure 2: Vixe!

25.1 Reuniões passadas

Na última terça-feira começamos a estudar dicionários

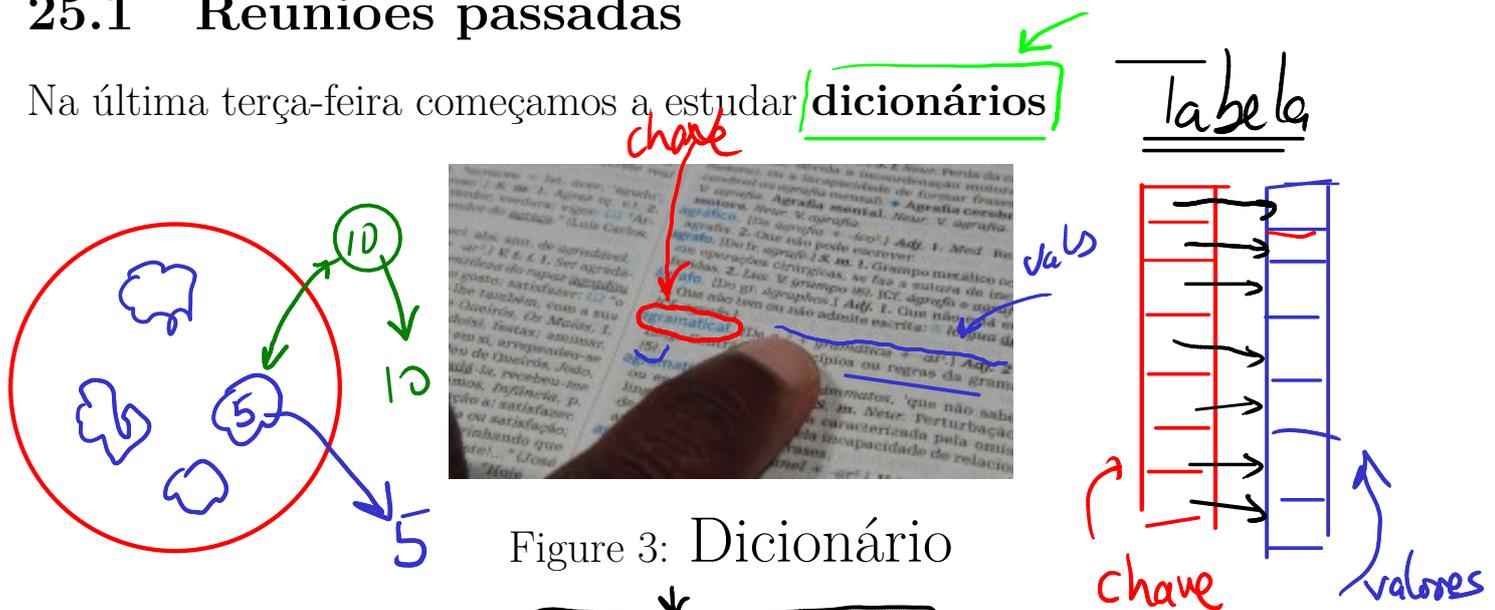


Figure 3: Dicionário

- listas são **mutáveis**, strings são **imutáveis** dicionários são **mutáveis**
- `strip()`: `s.strip()` retorna uma string com `s` sem brancos no início e no final
- `split()`: `s.split()` retorna uma lista de strings

```
In [5]: s = " Como é bom estudar MAC0110! "
```

```
In [6]: s.strip()
```

```
Out[6]: 'Como é bom estudar MAC0110!'
```

```
In [7]: s.split()
```

```
Out[7]: ['Como', 'é', 'bom', 'estudar', 'MAC0110!']
```

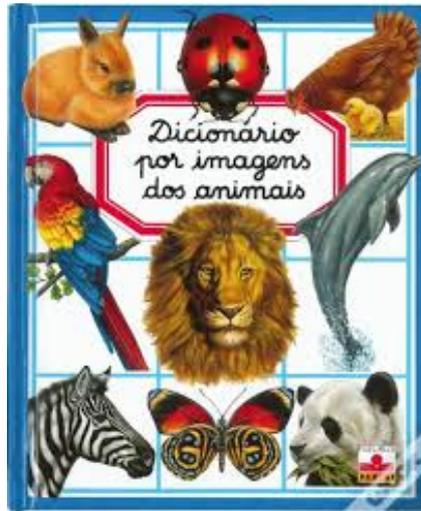
list

str

- operador in: in range(), in str, in list e in dict

tipo	... in <u>not in</u> ...	<u>if</u> ... in ...:	for ... in ..."
int	<u>i in range(10)</u>	i <u>not in</u> range(10)	if i in range(10):	for i in range(10):
str	<u>c in s</u>	c not in s	if c in s:	for <u>c</u> in <u>s</u> :
list	<u>item in lst</u>	item not in lst	if item in lst:	for <u>item</u> in <u>lst</u> :
dict	<u>chave in dicio</u>	chave not in dicio	if chave in dicio:	for <u>chave</u> in <u>dicio</u> :

25.2 Dicionários



Key Value

Figure 4: Que fofo!

Um **dicionário** é um conjunto de objetos ou itens cada um dotado de uma **chave** e de um **valor**.

Um dicionário está sujeito a dois tipos de operações:

- *inserção*: consiste em *introduzir* um objeto na tabela
- *busca*: consiste em *encontrar* um elemento que tenha uma dada chave.

Dicionário em Python

Python possui dicionários como um tipo nativo.

As **chaves** podem ser números inteiros ou strings ou outros tipos de dados **imutáveis**.

Uma maneira de criar um dicionário é começar com o dicionário vazio e adicionar pares chave-valor. O dicionário vazio é denotado por {}

```
In [1]: d = {} # dicionário vazio
```

lst = []
s = ""

Para inserirmos um par **chave-valor** no dicionário ou alterar o valor associado a uma chave, fazemos simplesmente

```
d[chave] = valor
```

Exemplos de uso

```
In [3]: eng2port = {} # dicionário vazio
```

```
In [4]: eng2port['one'] = 'um'
```

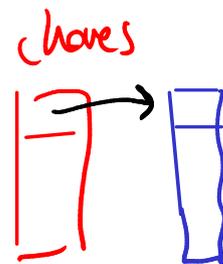
```
In [5]: eng2port['two'] = 'dois'
```

```
In [6]: eng2port['three'] = 'treiss'
```

```
In [8]: eng2port
```

```
Out[8]: {'one': 'um', 'two': 'dois', 'three': 'treiss'}
```

```
In [9]: eng2port['um']
```



KeyError

Traceback (most recent call last):

```
<ipython-input-9-69ed764ca20e> in <module>
```

```
----> 1 eng2port['um']
```

KeyError: 'um'

```
In [10]: eng2port['one']
```

```
Out[10]: 'um'
```

```
In [11]: eng2port['three']
```

```
Out[11]: 'treiss'
```

```
In [12]: eng2port['three'] = 'três' # são mutáveis
```

```
In [13]: eng2port
```

```
Out[13]: {'one': 'um', 'two': 'dois', 'three': 'três'}
```


txt → "como, 10" \n "é, 20" \n "bom, 30" \n

25.3 Exercício: consultas interativas

lst → []

Escreva um programa que leia um arquivo .csv e responda várias consultas sobre o seu conteúdo. O arquivo .csv contém **dois campos** por linha. O primeiro campo contém uma **palavra** (str sem espaços) e o segundo campo um **número inteiro** com, digamos que seja a frequência de cada palavra em algum texto. Um exemplo de arquivo csv é o seguinte

como, 10
é, 20
bom, 30
estudar, 45
MAC0110!, 1000

dict

	10
	20
	30
	45
MAC0110!	1000

lst → ["como, 10"]
"é, 20",
"bom, 30",
:
s.split()

O programa deve ser capaz de responder interativamente a consultas como:

- Qual a frequência de uma dada palavra? Para o arquivo acima e a palavra 'bom' a resposta é 30.
- Qual o número de pares palavra-frequência? Para o arquivo acima a deve ser 5.
- Qual a palavra com a maior frequência? Para o arquivo acima a resposta deve ser 'MAC0110!'.



Figure 5: Hã? Quem? Eu?

Solução

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun Nov 22 14:18:13 2020

@author: fradim
"""

MOSTRE = "mostre"
SAIR = 'sair'
CHAVES = 'chaves'
VALORES = 'valores'
ITENS = 'itens'
LEN = 'len'
MAX = 'max'

#-----
def main():
    # leia o dicionário
    dicio_palavras = crie_dicionario()

    # responda a consulta interativamente
    i = 1
    cmd = input(f"In [{i}]: ")
    while cmd != SAIR:
        if cmd in dicio_palavras:
            print(f"Out [{i}]: {dicio_palavras[cmd]}")
        elif cmd == CHAVES:
            print(f"Out [{i}]: {dicio_palavras.keys()}")
        elif cmd == VALORES:
            print(f"Out [{i}]: {dicio_palavras.values()}")
```

```

elif cmd == LEN:
    print(f"Out [{i}]: {len(dicio_palavras)}")
elif cmd == MAX:
    print(f"Out [{i}]: {mais_frequente(dicio_palavras)}")
elif cmd == ITENS:
    print(f"Out [{i}]: {dicio_palavras.items()}")
else:
    print(f"Out [{i}]: ERRO")
    help()

i += 1
cmd = input(f"In [{i}]: ")
print(f"Out [{i}]: tchau")

```

#-----

```

def mais_frequente(dicio):
    '''(dict) -> str

    RECEBE um dicionário `dicio`
    RETORNA a chave de maior valor.

    Pré-condição: supõe que os valores podem ser comparados
                    com >.

    '''
    max_chave = ''
    max_valor = 0
    for chave in dicio:
        if dicio[chave] > max_valor:
            max_chave = chave
            max_valor = dicio[chave]
    return max_chave

```

```

#-----
def init_dicionario():
    '''(None) -> dict
    LÊ o nome de um arquivo csv. Esse arquivo possui dois campos

    - chave (str)
    - valor (int)

    RETORNA um dicionário com esses pares chave-valor.
    '''
    dicio = {}
    nome = input("Digite o nome do arquivo: ")
    arq = open(nome, 'r', encoding="utf-8")
    for linha in arq:
        lst = linha.split(",")
        if len(lst) == 2:
            chave = lst[0].strip()
            valor = int(lst[1])
            dicio[chave] = valor
    return dicio

```

```

#-----
def help():
    '''(None) -> None

    EXIBE uma mensagem com ajuda para usar o programa.
    '''
    s = "Digite:\n" +\
        "    - um chave no dicionário para saber o valor associad\n"
    f"    - {SAIR}      : para sair\n" +\
    f"    - {CHAVES}   : para ver todas as chaves no dicionário\n"
    f"    - {VALORES}  : para ver todos os valores no dicionário\n"
    f"    - {ITENS}    : para ver todos os itens no dicionário\n"

```

```
f" - {LEN}      : número de itens no dicionário\n" +\  
f" - {MAX}      : chave de maior valor\n"  
print(s)
```

```
#-----  
if __name__ == "__main__":  
    main()
```