24 Reunião 24: 19/NOV/2020

Figure 1: Fonte: https://www.pinterest.es/

24.1 Reuniões passadas

- listas são mutáveis
- strings são imutáveis
- `strip()`: `s.strip()` retorna uma string com `s` sem brancos no inicio 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!']
```

- operador `in`: `in str` e `in list`

```
# c é substring de s
c in s

# item é um elemento da lista?
item in list

# i é um valor entre 0 e n-1?
i in range(n)

# percorre/itera sobre todos os caracteres
for c in str:
    print(c)

# percorre/itera sobre todos os itens
for item in lista:
```

```
    print(item)

# percorre/itera sobre todos os inteiros entre 0 e n-1
for i in range(n):
    print(i)
```

24.2 Exercício: frequências de palavras

Escreva um programa que leia um arquivo com palavras separadas por espaços e determine a frequência de cada palavra.

24.2.1 Exemplos

Arquivos para testes podem ser copiados daqui.

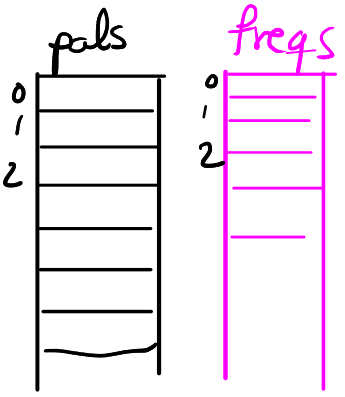
```
Digite o nome do arquivo: darci_sem_pontuacao.txt
Conteudo do arquivo:
Fracassei em tudo o que tentei na vida
Tentei alfabetizar as crianças brasileiras  não consegui
Tentei salvar os índios  não consegui
Tentei fazer uma universidade séria e fracassei
Tentei fazer o Brasil desenvolver se autonomamente
e fracassei
Mas os fracassos são minhas vitórias
Eu detestaria estar no lugar de quem me venceu
Darci Ribeiro

Lista de palavras no arquivo txt_sem_pontuacao/darci_sem_pont
0: 'Fracassei' : 1
1: 'em' : 1
2: 'tudo' : 1
3: 'o' : 2
4: 'que' : 1
5: 'tentei' : 1
6: 'na' : 1
7: 'vida' : 1
8: 'Tentei' : 4
9: 'alfabetizar' : 1
10: 'as' : 1
11: 'crianças' : 1
```

darci.txt ← usar no e-Disciplinas

pals

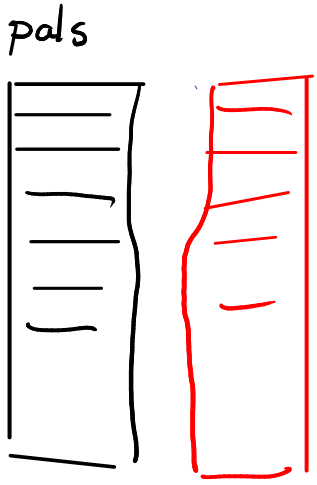
freq

Fazer uma função que recebe uma string txt e retorna uma "tabela"

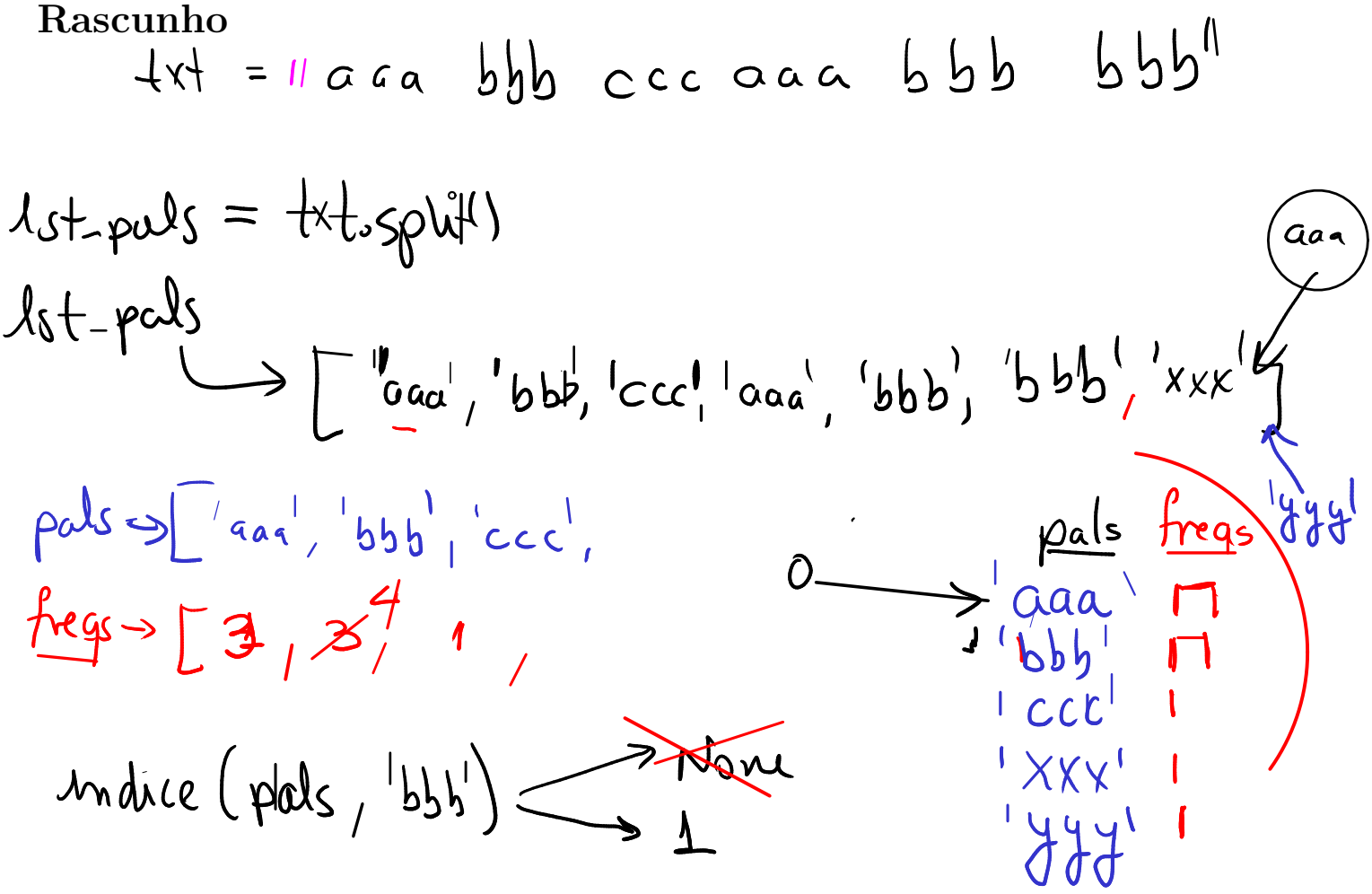


pals[i] ocorre freqs[i] vezes em txt

```
12: 'brasileiras' : 1
13: 'não' : 2
14: 'consegui' : 2
15: 'salvar' : 1
16: 'os' : 2
17: 'índios' : 1
18: 'fazer' : 2
19: 'uma' : 1
20: 'universidade' : 1
21: 'séria' : 1
22: 'e' : 2
23: 'fracassei' : 2
24: 'Brasil' : 1
25: 'desenvolver' : 1
26: 'se' : 1
27: 'autonomamente' : 1
28: 'Mas' : 1
29: 'fracassos' : 1
30: 'são' : 1
31: 'minhas' : 1
32: 'vitórias' : 1
33: 'Eu' : 1
34: 'detestaria' : 1
35: 'estar' : 1
36: 'no' : 1
37: 'lugar' : 1
38: 'de' : 1
39: 'quem' : 1
40: 'me' : 1
41: 'venceu' : 1
42: 'Darci' : 1
43: 'Ribeiro' : 1
```



Rascunho

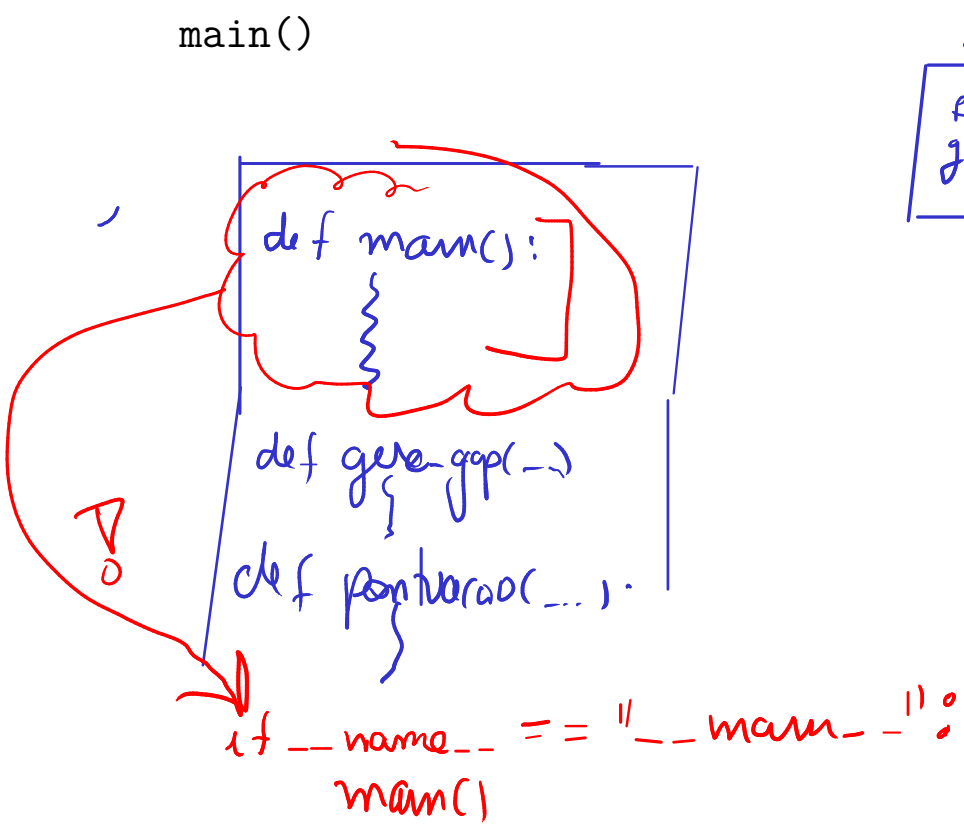

txt = "aaa bbb ccc aaa bbb bbb"
lst_pals = txt.split()
lst_pals
['aaa', 'bbb', 'ccc', 'aaa', 'bbb', 'bbb', 'xxx', 'yyy']
aaa
pals
['aaa', 'bbb', 'ccc',
freqs
pals
freqs
0
'aaa'
'bbb'
'ccc'
'xxx'
'yyy'
indice(pals, 'bbb')
None
1

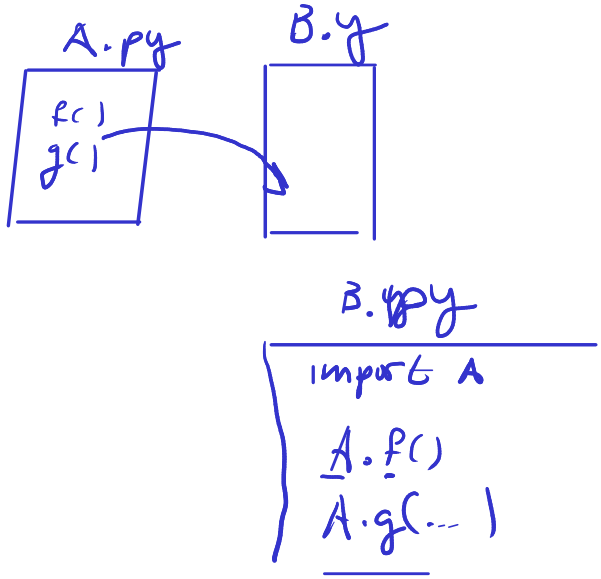
24.2.2 Solução

```
'''
   Programa que lê um arquivo em que as palavras estão separadas
   brancos e conta a frequência de cada palavra.
'''

def main():
    # leitura do texto
    # 1 pegue o nome do arquivo
    nome_in = input("Digite o nome do arquivo: ")
    # 2 abrir o arquivo para leitura: 'r'= read
    arq_in = open(nome_in, "r", encoding="utf-8")
    # 3 leia o conteudo do arquivo
    txt = arq_in.read()
    # 4 fechar o arquivo
    arq_in.close()

    print(f"Conteudo do arquivo '{nome_in}'")
    print(txt)

    lst_pals, lst_freqs = conte_palavras(txt)

    print(f"Lista de palavras no arquivo {nome_in}:")
    for i in range(len(lst_pals)):
        print(f"{i}: '{lst_pals[i]}' : {lst_freqs[i]}")

    # criação de um arquivo
    # 1 pegue o nome do arquivo
    nome_out = nome_in[:-3] + "csv"
    # 2 abra o arquivo para escrita: 'w' = write
    arq_out = open(nome_out, "w", encoding="utf-8" ) # write
    # 3 escreva no arquivo: write()
```

```
    for i in range(len(lst_pals)):
        arq_out.write(f"{lst_pals[i]},{lst_freqs[i]}\n"  )
    # 4 feche o arquivo
    arq_out.close()

    print(f"Arquivo {nome_out} foi criado")

#-----------------------------------------------------------------
def conte_palavras(txt):
    '''(str) -> list, list

    RECEBE uma string `txt`.
    RETORNA duas listas de mesmo comprimento:

       * uma lista de strings com todas as palavras que ocorrem
         no texto, sem repeticões;
       * uma lista de inteiros com a respectiva frequência de co
         palavra no texto.
    '''
    pals  = []
    freqs = []
    lst_palavras = txt.split()
    for pal in lst_palavras:
        j = indice(pals, pal)
        if j != None:
            freqs[j] += 1
        else:
            pals += [pal]
            freqs += [1]
    return pals, freqs

#-----------------------------------------------------------------
def indice(lst, val):
```

```
    '''(list, obj) -> int ou None

    RECEBE uma lista `lst` e um valor `val`.
    RETORNA o índice da primeira posição de `lst` que tem o
    valor a `val`.
    '''
    for i in range(len(lst)):
        if lst[i] == val:
            return i
    return None

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





24.3 Dicionários

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

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

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.

24.3.1 Dicionário em Python

Python possui dicionários como um tipo nativo.

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

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

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

Alguns exemplos

```
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 rec
<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'}
```

24.3.2 Métodos de dicionários

Dicionários possuem vários métodos nativos úteis. A seguinte tabela fornece um resumo e mais detalhes podem ser encontrados em Python Documentation.

Método	Parâmetros	Descrição
`keys()`	nenhum	Retorna uma vista das chaves no dicionário
`values()`	nenhum	Retorna uma vista dos valores no dicionário
`items()`	nenhum	Retorna uma vista dos pares chave-valor no dicionário
`get()`	key	Retorna o valor associado com a chave; ou `None`
`get()`	key,alt	Retorna o valor associado com a chave; ou `alt`

24.4 Frequências de palavras com dicionários

```
'''
   Programa que lê um arquivo em que as palavras estão separadas
   brancos e conta a frequência de cada palavra.
'''

def main():
    # leitura do texto
    # 1 pegue o nome do arquivo
    nome_in = input("Digite o nome do arquivo: ")
    # 2 abrir o arquivo para leitura: 'r'= read
    arq_in = open(nome_in, "r", encoding="utf-8")
    # 3 leia o conteudo do arquivo
    txt = arq_in.read()
    # 4 fechar o arquivo
    arq_in.close()

    # crie um dicionário em que as chaves são as palavras e os
    # valores são suas frequências
    dicio_pals = conte_palavras(txt)

    print(f"Lista de palavras no arquivo '{nome_in}':")
    i = 0
    for pal in dicio_pals:
        print(f"{i}: '{pal}' : {dicio_pals[pal]}")
        i += 1

    # criação de um arquivo
    # 1 pegue o nome do arquivo
    nome_out = nome_in[:-3] + "csv"
    # 2 abra o arquivo para escrita: 'w' = write
    arq_out = open(nome_out, "w", encoding="utf-8" ) # write
```

```
    # 3 escreva no arquivo: write()
    for pal in dicio_pals:
        arq_out.write(f"'{pal}',{dicio_pals[pal]}\n")
    # 4 feche o arquivo
    arq_out.close()

    print(f"Arquivo '{nome_out}' foi criado")

#-----------------------------------------------------------------
def conte_palavras(txt):
    '''(str) -> dict

    RECEBE uma string `txt`.
    RETORNA um dicionário em que a CHAVE e cada palavra em txt
    e o VALOR é a sua frequência em txt.
    '''
    dicio = {}
    lst_palavras = txt.split()
    for pal in lst_palavras:
        if pal in dicio:
            dicio[pal] += 1
        else:
            dicio[pal] = 1
    return dicio

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