


DEPARTAMENTO
DE INFORMÁTICA
PUC-RIO

Dicionários

1



DEPARTAMENTO
DE INFORMÁTICA
PUC-RIO

Exercício: Índice Remissivo

Uma editora deseja construir o índice remissivo de algumas palavras em um livro técnico do seguinte modo:

palavra_a: pag_i,... pag_n
palavra_b: pag_i,... pag_n
.....
palavra_z: pag_i,... pag_n

Cada página do livro é finalizada pelo símbolo '@'. As palavras podem ser separadas por espaço (' '), vírgula(',') ou ponto('.').

Considere as seguintes palavras para o índice remissivo:

['REPOUSO','INÉRCIA','ATRITO','MOVIMENTO']

Faça um programa para mostrar, na tela, o índice remissivo destas palavras.

2



Exercício: Exemplo de um arquivo TextoGalileu.txt*

Antes de Galileu, a maioria dos pensadores acreditava que um corpo em movimento se encontraria num estado forçado, enquanto que o repouso seria o seu estado natural. A experiência diária parece confirmar essa afirmativa. Quando depositamos um livro sobre uma mesa é fácil constatar seu estado natural de repouso. Se colocarmos o livro em movimento, dando-lhe apenas um rápido empurrão, notamos que ele não irá se mover indefinidamente: o livro deslizará sobre a mesa até parar. Ou seja, é fácil observar que cessada a força de empurrão da mão, o livro retorna ao seu estado natural de repouso. Logo, para que o livro se mantenha em movimento retilíneo uniforme é necessária a ação contínua de uma força de empurrão. Galileu, entretanto, foi contra essa ideia de movimento ser um estado necessariamente forçado, argumentando que o livro só interrompeu seu deslizamento (vindo a parar) em razão da existência de atrito com a mesa. Isto é, se lançássemos o livro sobre uma mesa menos áspera, haveria menos resistência ao seu deslizamento. Se o seu lançamento ocorresse sobre uma mesa perfeitamente polida, livre de atrito, o livro manter-se-ia em movimento retilíneo uniforme indefinidamente, sem a necessidade de estar sendo continuamente empurrado. Em virtude disso, Galileu conclui ser uma tendência natural dos corpos a manutenção de seu estado de repouso ou de seu estado de movimento retilíneo uniforme, promovendo aos corpos uma propriedade denominada inércia.

*<https://www.mundovestibular.com.br/articles/627/1/PRIMEIRA-LEI-DE-NEWTON/Paacutegina1.html>

3

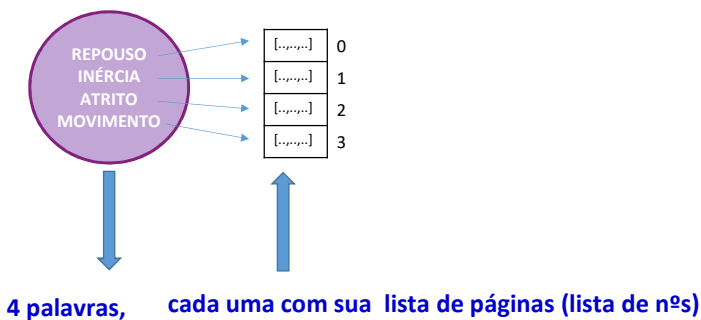


Desenvolvimento da Solução

- Índice remissivo:

sequência de palavras,

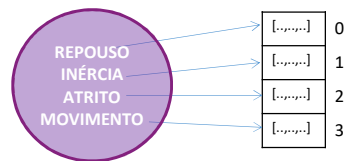
cada uma com a indicação da(s) página(s) onde aparece(m)



4

Desenvolvimento da Solução

- Índice remissivo:
 - sequência de palavras,
 - cada uma com a indicação da(s) página(s) onde aparece(m)
- Agrupar, por palavra, suas ocorrências nas páginas

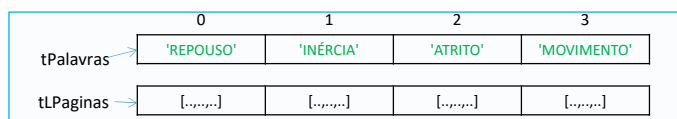


Problema : Como representar este mapeamento?

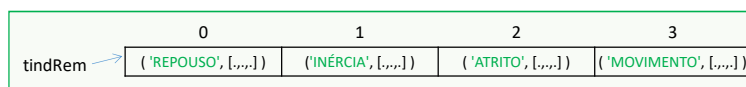
5

Desenvolvendo a Solução

Duas possíveis alternativas:



Tuplas de tuplas:



6

Desenvolvendo a Solução

Como acessar uma palavra?



Tuplas de tuplas:



Como em listas e tuplas a **indexação** é por números inteiros (posição na sequência), exige [uma busca](#) para encontrar a posição a ser usada (mesmo que seja pelo método index)

7

Resumindo_{2/2}

I. IDÉIA DE UMA SOLUÇÃO PARA A 2ª ALTERNATIVA

A PARTIR DAS PALAVRAS DO ÍNDICE REMISSIVO

Criar a estrutura do índice remissivo (tupla de tuplas):

uma tupla onde cada elemento é (palIndRem, [])

Para cada palavra do índice remissivo:

Para cada página:

Se palavra do índice remissivo está na página

Inclui nº da página na lista de ocorrências

Exibir o índice remissivo

8



Resumindo:

II. IDÉIA DE UMA SOLUÇÃO PARA A 2ª ALTERNATIVA

A PARTIR DAS PALAVRAS DO ARQUIVO

Criar uma tupla onde cada elemento é (palavra,[])

Para cada página

Para cada palavra da página

*Se palavra pertence ao índice remissivo (**buscar na tupla**)*

*encontra a tupla da palavra (posição retornada pela **busca**)*

*Inclui nº da página na lista de ocorrências(**se não estiver**)*

Exibir o índice remissivo

9



Uma Solução: tupla de tuplas

```
"""
FUNÇÃO cria_lista_paginas:
    -recebe o nome de um arquivo texto no formato descrito
    -retorna uma lista com as páginas(texto da página já sem
    pontuação e em letra maiúscula)
"""

def cria_lista_paginas(nomeArq):
    arq = open(nomeArq, 'r')
    texto = arq.read() # lê todo o arquivo
    # substitui ', ' '. ' por ' ' colocando em maiúscula
    texto = (texto.replace(', ', ' ').replace('.', ' ')).upper()
    # separa as páginas do arquivo
    lPaginas = texto.split('@')
    arq.close()
    return lPaginas
```

10



Uma Solução: tupla de tuplas

```

"""
FUNÇÃO prepara:
- recebe o nome de um arquivo com o texto
- constrói a partir do arquivo lido uma lista (lpalPag)
onde o elemento de cada posição da lista(lpalPag[i]) é uma
lista com as palavras da página i: LISTA de LISTAS
"""
def prepara(nome):
    lPaginas = cria_lista_paginas(nome) # arruma texto e
    # separa em páginas
    lpalPag = []
    for el in lPaginas: # separa as palavras de uma página
        lpalPag.append(el.split())
    return lpalPag

```

11



Uma Solução: tupla de tuplas FAZENDO A BUSCA NO ÍNDICE REMISSIVO

```

"""
FUNÇÃO busca:
- recebe um índice remissivo como uma tupla de tuplas e uma
palavra, onde tupla[i] é a tupla ('PalavraIndice',
[lista das paginas onde a palavra ocorre]).
- busca a palavra no índice remissivo. Retorna a posição no
índice remissivo onde a palavra aparece ou None se
ela não pertencer ao índice
OBS: cada elemento el da tupla recebida é uma tupla, onde
el[0] é a palavraIndice e el[1] é a lista das páginas onde
a palavra aparece
"""
def busca(tIndRem, pal):
    for i, el in enumerate(tIndRem):
        if pal == el[0]:
            return i
    return None

```

12



Uma Solução: tupla de tuplas

```

"""
PROGRAMA:
"""

ltPalsPorPag=prepara('TextoGalileu.txt')
tIndRem=((('REPOUSO',[]),('INÉRCIA',[]),('ATRITO',[]),('MOVIMENTO',[]))
for n,pg in enumerate(ltPalsPorPag):
    npag=n+1
    for pal in pg:
        pos = busca(tIndRem,pal)
        if pos != None:
            if npag not in tIndRem[pos][1]:
                tIndRem[pos][1].append(npag)
print(tIndRem)

(('REPOUSO', [1, 2, 4 ]), ('INÉRCIA', [4]), ('ATRITO', [3]), ('MOVIMENTO', [1, 2, 3, 4]))

```

13



Uma Solução: tupla de tuplas

```

def cria_lista_paginas(nomeArq):
    arq = open(nomeArq, 'r')
    texto = arq.read() # lê todo o arquivo
    # substitui ', ' por ' ' colocando em maiúscula
    texto=((texto.replace(', ', ' ')).replace('.', ' ')).upper()
    # separa as páginas do arquivo
    lPaginas = texto.split('@')
    arq.close()
    return lPaginas

def busca(tIndRem, pal):
    for i,el in enumerate(tIndRem):
        if pal == el[0]:
            return i
    return None

def prepara(nome):
    lPaginas = cria_lista_paginas(nome) # arruma texto e separa em páginas
    lpalPag = []
    for el in lPaginas:
        # separa as palavras de uma página
        lpalPag.append(el.split())
    return lpalPag

ltPalsPorPag=prepara('textogalileu.txt')
tIndRem=((('REPOUSO',[]),('INÉRCIA',[]),('ATRITO',[]),('MOVIMENTO',[]))
for n,pg in enumerate(ltPalsPorPag):
    npag=n+1
    for pal in pg:
        pos = busca(tIndRem,pal)
        if pos != None:
            if npag not in tIndRem[pos][1]:
                tIndRem[pos][1].append(npag)
print(tIndRem)

```

14



Analizando os dados

Cada item do índice remissivo armazena:

- ✓ palavra,
- ✓ lista de páginas nas quais a palavra ocorre

A palavra não se repete e é um identificador do elemento que a armazena (como o CPF de uma pessoa).

Quando um atributo possui **valor único** para cada item, ele pode ser usado para localizar o item (como seu índice) e é chamado de **campo chave**.

Ex: o “atributo” PLACA de um carro possui valor único para cada “item” CARRO. Poderia, se desejado, ser considerado um “campo chave” e servir de índice em um sistema de monitoramento de veículos na hora de localizar um carro.

15



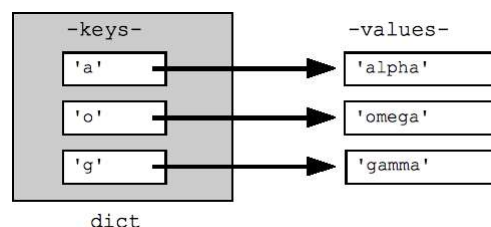
Alternativa: Dicionário

Dicionário: Estrutura de dados que implementa mapeamentos entre uma chave (*key*) e algum conteúdo (*value*)


Mapeamentos também são chamados de pares chave-valor

A chave funciona como um **índice** para acessar o conteúdo

Conteúdo pode ser qualquer coisa, inclusive outro dicionário



16




Exemplos

```
dicPortIngles={'um':'one','dois':'two','tres':'three'}
```



```
dicInscr={ '172354881' : ['INF1111','MAT5555'] ,
           '194354884' : ['INF1711','MAT6655','FIS7676'] ,
           '232354672' : ['FIS7672','MAT3232']
         }
```

17



Desenvolvendo a solução: ÍNDICE REMISSIVO

Quem deve ser a chave?

- ✓ *O acesso é por qual elemento?*

O que deve ser o conteúdo?

- ✓ *Qual(is) valor(es) associar à chave?*

0	1	2	3
('REPOUSO', [.,.,.])	('INÉRCIA', [.,.,.])	('ATRITO', [.,.,.])	('MOVIMENTO', [.,.,.])

18



Desenvolvendo a solução: ÍNDICE REMISSIVO

Quem deve ser a chave?


✓ O acesso é por qual elemento? **Palavra**

O que deve ser o conteúdo?

✓ Qual(is) valor(es) associar à chave? **Lista de páginas**

0	1	2	3
('REPOUSO', [...])	('INÉRCIA', [...])	('ATRITO', [...])	('MOVIMENTO', [...])

19



ÍNDICE REMISSIVO como Dicionário

Dicionário onde

a chave: **palavra**

o conteúdo: **a lista de páginas**

-chaves-

'REPOUSO'

'INÉRCIA'

'ATRITO'

'MOVIMENTO'

→

valores

[1,2,3,4,5]

[1,2,3,5]

[1,3]

[1,2,5]

dicionário

20



Dicionários

Listas → indexadas por **inteiros**

Dicionários → (*hash tables*) indexados por **chaves** (keys), que podem ser de qualquer tipo imutável (como strings e inteiros)

Têm comprimento variável, são heterogêneos e podem ser aninhados

São delimitados por **{ }**

Os elementos são pares **chave:valor** separados por vírgulas

21



Criando um dicionário por enumeração

Dicionário vazio:

```
nomeVariávelDic = {}  
ou  
nomeVariávelDic = dict()
```


Dicionário com elementos enumerados:

```
nomeVariávelDic = { el1, el2, ..., eln }
```

Elemento:

chave:valor

22




Exemplo

```
dicIndRem = {'REPOUSO': [1,2,3,4,5], 'INÉRCIA': [1,2,3,5]}
```

↙ ↘

chave valor

23



Pertinência da chave no dicionário

chave in varDicionário

Retorna True se chave pertence ao dicionário e False caso contrário

Exemplos:

```
>>>dicIndRem = {'REPOUSO': [1,2,3,4,5], 'INÉRCIA': [1,2,3,5]}

>>>'REPOUSO' in dicIndRem
True

>>>'up' in dicIndRem
False
```

24



Acesso ao valor de um elemento do dicionário

Pelo operador de indexação `[]`.

`varDicionário[chave]`

Chaves inexistentes provocam um erro

Exemplos:

```
>>>dicIndRem = {'REPOUSO': [1,2,3,4,5], 'INÉRCIA': [1,2,3,5]}
>>>dicIndRem['REPOUSO']
[1,2,3,4,5]
>>>dicIndRem['up']
Traceback (most recent call last):
  File , line 1, in <module>
    dictIndRem['up']
KeyError: 'up'
```

25



Iterando sobre os elementos de um dicionário

for chave in dicionário:
CORPO

A iteração em elementos de um dicionário é feita a partir da chave

Obs: não tem ordem pré-definida

Exemplos:

```
dicIndRem = {'REPOUSO':1,2,3,4,5}, 'INÉRCIA': [1,2,3,5], 'ATRITO': [1,3]}
for pal in dicIndRem:
    print(pal, ': ', dicIndRem[pal])
```

REPOUSO: [1, 2, 3, 4, 5]

INÉRCIA: [1, 2, 3, 5]

ATRITO: [1, 3]

26



Mãos na massa!!

Crie um dicionário com os nomes, altura e peso abaixo:

Huguinho	1.20	45
Luisinho	1.10	60
Zezinho	1.00	100
Patinhas	1.10	40
Donald	1.20	50

a) Mostre o nome e o IMC ($\text{peso}/\text{altura}^2$) de cada um

27



Acesso ao valor de um elemento do dicionário

`varDicionário.get(chave, valor)`

Obtém o conteúdo associado à chave

Se chave não existe:

- a) retorna valor ou
- b) **None**, se valor não for especificado

Exemplos:

```
dicIndRem={ 'REPOUSO': [1,2,3,4,5], 'INÉRCIA': [1,2,3,5], 'ATRITO': [1,3] }

>>> dicIndRem.get('REPOUSO')
[1,2,3,4,5]
>>> dicIndRem.get('Sequência', 'Não Existe')
Não Existe
>>> dicIndRem.get('Sequência')
None
```

28



Acesso às chaves do dicionário

`varDicionário.keys()`

Retorna uma estrutura iterável com as chaves do dicionário.

Obs: Use o operador *list* para transformá-las em lista

Exemplos:

```
dicIndRem={'REPOUSO':[1,2,3,4,5], 'INÉRCIA':[1,2,3,5], 'ATRITO':[1,3]}
>>> dicIndRem.keys()
dict_keys(['ATRITO', 'REPOUSO', 'INÉRCIA'])
>>> list(dicIndRem.keys())
['ATRITO', 'REPOUSO', 'INÉRCIA']
```

29



Acesso aos valores do dicionário

`varDicionário.values()`

Retorna uma estrutura iterável com todos os valores do dicionário

Obs: Use o operador *list* para transformá-las em lista

Exemplos:

```
dicIndRem={'REPOUSO':[1,2,3,4,5], 'INÉRCIA':[1,2,3,5], 'ATRITO':[1,3]}
>>> dicIndRem.values()
dict_values([ [1, 2, 3, 4, 5], [1, 2, 3, 5], [1, 3] ])
>>> list(dicIndRem.values())
[ [1, 2, 3, 4, 5], [1, 2, 3, 5], [1, 3] ]
```

30



Acesso ao pares chave-valor do dicionário

`varDicionário.items()`

Retorna uma estrutura iterável de tuplas com os pares chave-valor do dicionário

Obs: Use o operador *list* para transformá-la em lista

Exemplos:

```
dicIndRem={'REPOUSO':[1,2,3,4,5], 'INÉRCIA':[1,2,3,5], 'ATRITO':[1,3]}
>>> dicIndRem.items()
dict_items([('REPOUSO', [1, 2, 3, 4, 5]), ('INÉRCIA', [1, 2, 3, 5]), ('ATRITO', [1, 3])])
>>> list(dicIndRem.items())
[('REPOUSO', [1, 2, 3, 4, 5]), ('INÉRCIA', [1, 2, 3, 5]), ('ATRITO', [1, 3])]
```

31



Alteração de um elemento do dicionário

`varDicionário[chave] = valor`

O valor do elemento indexado pela *chave* é substituído

Exemplos:

```
dicIndRem={'REPOUSO':[1,2,3,4,5], 'INÉRCIA':[1,2,3,5], 'ATRITO':[1,3]}
>>> dicIndRem['INÉRCIA']=[]
>>> dicIndRem
{'ATRITO': [1, 3], 'REPOUSO': [1, 2, 3, 4, 5], 'INÉRCIA': []}
```

32



Uma Solução do Exemplo com Dicionários - Alternativa I

```
lpalsPag=prepara('TextoGalileu.txt')
dicIndRem={ 'REPOUSO': [], 'INÉRCIA': [], 'ATRITO': [], 'MOVIMENTO': [] }
for n,pg in enumerate(lpalsPag):
    npag=n+1
    for pal in pg:
        if pal in dicIndRem:
            if npag not in dicIndRem[pal]:
                dicIndRem[pal].append(npag)
print(dicIndRem)
```

#Para cada palavra do texto, inclui sua página no dicionário se for uma das selecionadas que não tenha já ocorrido na página

33



Uma Solução do Exemplo com Dicionários - Alternativa II

```
lpalsPag=prepara('TextoGalileu.txt')
dicIndRem={ 'REPOUSO': [], 'INÉRCIA': [], 'ATRITO': [], 'MOVIMENTO': [] }
for palIR in dicIndRem:
    for (npag,pag) in enumerate(lpalsPag):
        if palIR in pag:
            dicIndRem[palIR].append(npag+1)
print(dicIndRem)
```

#Para cada chave do dicionário do índice remissivo, verifica se a mesma ocorre em cada uma das páginas, incluindo o número da página se ocorrer

34



Outra Solução do Exemplo com Dicionários

```
lpalsPag=prepara('TextoGalileu.txt')
dicIndRem={'REPOUSO':[], 'INÉRCIA':[], 'ATRITO':[], 'MOVIMENTO':[]}
lChaves=dicIndRem.keys()
for chave in lChaves:
    for npag,pag in enumerate(lpalsPag):
        if chave in pag:
            dicIndRem[chave].append(npag+1)
print(dicIndRem)
```

#Para cada palavra selecionada, verifica em cada página do texto se está presente, incluindo o número em caso afirmativo

35



Mãos na massa!!

Crie um dicionário com os nomes, altura e peso abaixo:

Huguinho	1.20	45
Luisinho	1.10	60
Zezinho	1.00	100
Patinhas	1.10	40
Donald	1.20	50

- Mostre o nome e o IMC ($\text{peso}/\text{altura}^2$) de cada um
- Crie uma lista com as chaves e outra lista com os valores.
- Calcule o peso médio e acrescente ao peso de todos os patinhos que tem U no nome, 10% do peso médio

36



Inclusão de um elemento no dicionário

`varDicionário[chave inexistente] = valor`

O valor da *chave* não deve existir no dicionário

As chaves são armazenadas **em qualquer posição** (implementados por tabelas de espalhamento - HashTables) e a falta de ordem é proposital!

Exemplos:

```
dicIndRem={'REPOUSO':[1,2,3,4,5], 'INÉRCIA':[1,2,3,5], 'ATRITO':[1,3]}
>>> dicIndRem['Sequência']=[]
>>> dicIndRem
{'ATRITO': [1, 5], 'REPOUSO': [1, 2, 3, 4, 5], 'INÉRCIA': [], 'Sequência': []}
>>> dicIndRem['Set']=[]
>>> dicIndRem
{'Set': [], 'ATRITO': [1, 5], 'REPOUSO': [1, 2, 3, 4, 5], 'INÉRCIA': [], 'Sequência': []}
```

37



Mãos na massa!!

Crie um dicionário com os nomes, altura e peso abaixo:

Huguinho	1.20	45
Luisinho	1.10	60
Zezinho	1.00	100
Patinhas	1.10	40
Donald	1.20	50

- Mostre o nome e o IMC ($\text{peso}/\text{altura}^2$) de cada um
- Crie uma lista com as chaves e outra lista com os valores.
- Calcule o peso médio e acrescente ao peso de todos os patinhos que tem U no nome, 10% do peso médio
- Inclua a Clarabela que mede 2.30m e pesa 80kg e o Peninha com 1.20m e 60 kg

38



Exclusão de um elemento no dicionário

`del varDicionário[chave]`

Chaves inexistentes provocam um erro

Exemplos:

```
dicIndRem={'Set':[1], 'REPOUSO':[1,2,3,4,5], 'INÉRCIA':[1,2,3,5], 'ATRITO':[1,3]}
>>>del dicIndRem['Set']
>>>dicIndRem
{'ATRITO': [1, 5], 'REPOUSO': [1, 2, 3, 4, 5], 'INÉRCIA': []}
>>>del dicIndRem['Set']
>>>dicIndRem
Traceback (most recent call last):
  File "<pyshell#76>", line 1, in <module>
    del dictIndRem['Set']
KeyError: 'Fiat'
```

39



Exclusão com retorno do valor

`varDicionário.pop(chave, valorAlt)`


Obtém o valor correspondente à **chave** e remove o par chave/valor do dicionário
Se chave não existe:

- a) retorna valorAlt ou
- b) provoca um erro se valor não for especificado

Exemplos:

```
dicIndRem={'Set':[1], 'REPOUSO':[1,2,3,4,5], 'INÉRCIA':[1,2,3,5], 'ATRITO':[1,3]}
>>>el =dictIndRem.pop('Set')
>>>dictIndRem
{'ATRITO': [1, 5], 'REPOUSO': [1, 2, 3, 4, 5], 'INÉRCIA': []}
>>> el
[1]
>>>el = dictIndRem.pop('Set','')
>>>el
''
```

40




Mãos na massa!!

Crie um dicionário com os nomes, altura e peso abaixo:

Huguinho	1.20	45
Luisinho	1.10	60
Zezinho	1.00	100
Patinhas	1.10	40
Donald	1.20	50

- a) Mostre o nome e o IMC ($\text{peso}/\text{altura}^2$) de cada um
- b) Crie uma lista com as chaves e outra lista com os valores.
- c) Calcule o peso médio e acrescente ao peso de todos os patinhos que tem U no nome, 10% do peso médio
- d) Inclua a Clarabela que mede 2.30m e pesa 80kg e o Peninha com 1.20m e 60 kg
- e) Retire o Peninha exibindo seus dados

41



Exclusão de todos

varDicionário.clear()

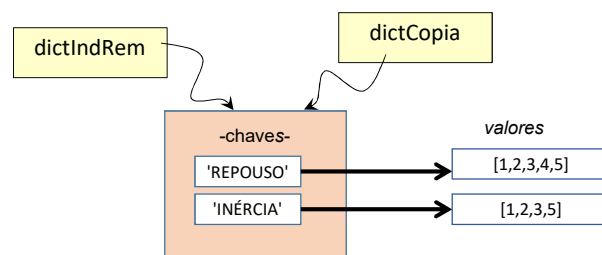
Exclui todos os elementos do dicionário

42

Copiar x Clonar

Comportam-se do mesmo modo que as listas. Na atribuição, o que é copiado é o endereço de memória, e portanto, alterações nas cópias são refletidas umas nas outras

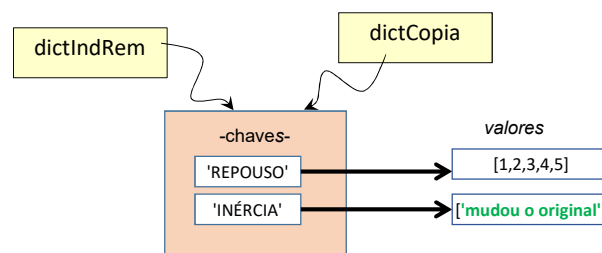
```
>>>dictIndRem = { 'REPOUSO': [1,2,3,4,5], 'INÉRCIA': [1,2,3,5] }
>>>dictCopia = dictIndRem
```



43

Copiar x Clonar

```
>>>dictIndRem = { 'REPOUSO': [1,2,3,4,5], 'INÉRCIA': [1,2,3,5] }
>>>dictCopia = dictIndRem
>>>dictCopia['INÉRCIA']=[ 'mudou o original' ]
```



44

DEPARTAMENTO DE INFORMÁTICA PUC RIO

Copiar x Clonar

varDicionário.copy()

Retorna um outro dicionário com os mesmos pares chave/valor

```
>>> dictClone = dictIndRem.copy()
```

dictIndRem

dictClone

'REPOUSO'

'INÉRCIA'

[1,2,3,4,5]

'mudou o original'

-chaves-

'REPOUSO'

'INÉRCIA'

45

DEPARTAMENTO DE INFORMÁTICA PUC RIO

Copiar x Clonar

varDicionário.copy()

Retorna um outro dicionário com os mesmos pares chave/valor

```
>>> dictClone = dictIndRem.copy()
```

dictIndRem

dictClone

'REPOUSO'

'INÉRCIA'

[1,2,3,4,5]


'mudou o original'

-chaves-

'REPOUSO'

'INÉRCIA'

46



Copiar x Clonar

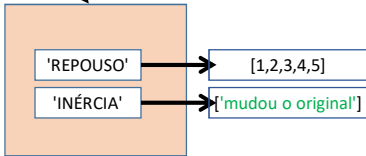
`varDicionário.copy()`

Retorna um outro dicionário com os mesmos pares chave/valor

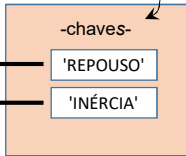
```

>>> dictClone = dictIndRem.copy()
>>> dictClone['INÉRCIA'] = 'só na cópia'
```


dictIndRem



dictClone



47



Copiar x Clonar

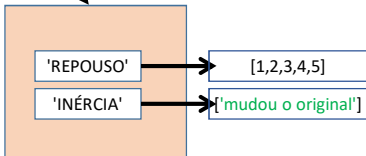
`varDicionário.copy()`

Retorna um outro dicionário com os mesmos pares chave/valor

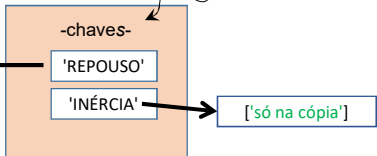
```

>>> dictClone = dictIndRem.copy()
>>> dictClone['INÉRCIA'] = 'só na cópia'
```

dictIndRem



dictClone



48



Mãos na massa!!

Crie um dicionário com os nomes, altura e peso abaixo:

Huguinho	1.20	45
Luisinho	1.10	60
Zezinho	1.00	100
Patinhas	1.10	40
Donald	1.20	50

- Mostre o nome e o IMC ($\text{peso}/\text{altura}^2$) de cada um
- Crie uma lista com as chaves e outra lista com os valores.
- Calcule o peso médio e acrescente ao peso de todos os patinhos que tem U no nome, 10% do peso médio
- Inclua a Clarabela que mede 2.30m e pesa 80kg e o Peninha com 1.20m e 60 kg
- Retire o Peninha exibindo seus dados
- Crie uma cópia deste dicionário, modificando, na cópia, o peso do Donald para 70kg

49



Atualizar um dicionário

varDicionário.update(dic2)

Atualiza um dicionário com os elementos de outro

Os itens em *dic2* que não estão em *varDicionário* são adicionados e os que estão tem seu valor atualizado.

Exemplos:

```
>>> dictIndRem = {'REPOUSO': [1, 2, 3, 4, 5], 'INÉRCIA': [1, 2, 3, 5]}
>>> dictOutro = {'ATRITO': [1, 5], 'MOVIMENTO': [1, 2, 5]}
>>> dictIndRem.update(dictOutro)
>>> dictIndRem
{'ATRITO': [1, 5], 'REPOUSO': [1, 2, 3, 4, 5], 'INÉRCIA': [1, 2, 3, 5], 'MOVIMENTO': [1, 2, 5]}
```

50



Matrizes Esparsas: como listas de listas

Em geral, uma matriz é representada como uma lista de listas.

0	0	0	1	0
0	0	0	0	0
0	2	0	0	0
0	0	0	0	0
0	0	0	3	0

matriz = [[0,0,0,1,0], [0,0,0,0,0], [0,2,0,0,0], [0,0,0,0,0], [0,0,0,3,0]]

Observando esta matriz, nota-se que a maior parte de seus elementos é 0. (matriz esparsa)

Solução: representar esta matriz esparsa como um dicionário

51



Matrizes Esparsas: como dicionário

matriz = [[0,0,0,1,0], [0,0,0,0,0], [0,2,0,0,0], [0,0,0,0,0], [0,0,0,3,0]]



matriz = {(0,3): 1, (2,1): 2, (4,3): 3}

✓ Como acessar um elemento com valor desta matriz? matriz[(tupla)]

>>> matriz[(2,1)] → 2

✓ Como acessar um elemento zero ?

>>> matriz[(1,2)] → **KeyError: (1,2)**

Solução: matriz.get(tupla, valor)

chave

valor quando a chave não existe

>>> matriz.get((0,3), 0) → 0

>>> matriz.get((2,1), 0) → 2

52



Mãos na massa!!

Crie um dicionário com os nomes, altura e peso abaixo:

Clarabela	2.20	78
Margarida	1.10	40
Vovó Donalda	1.00	40

- Mostre o nome e o IMC ($\text{peso}/\text{altura}^2$) de cada um
- Atualize o dicionário copiado dos patinhos com este dicionário criado
- Retire do dicionário copiado todos os elementos que estão no dicionário original
- Atualize o dicionário original com o dicionário resultante do item c)