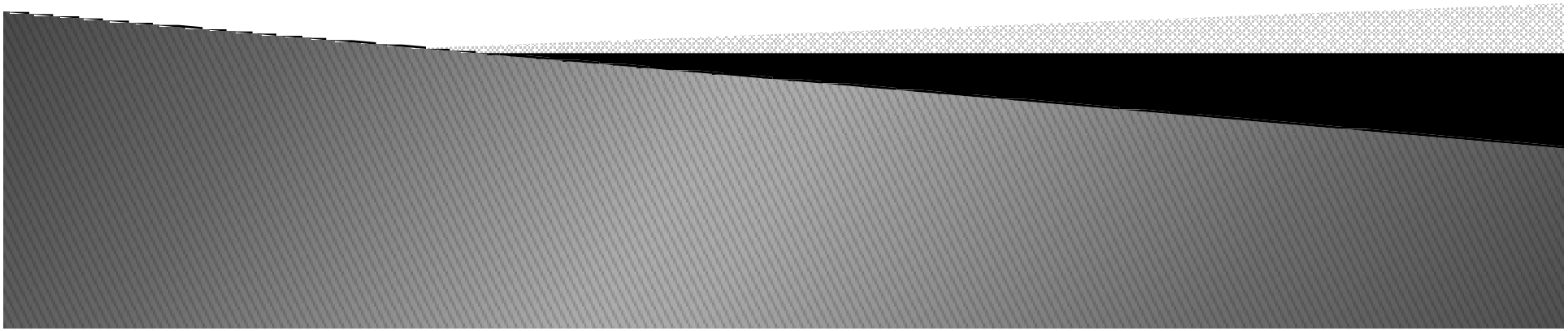
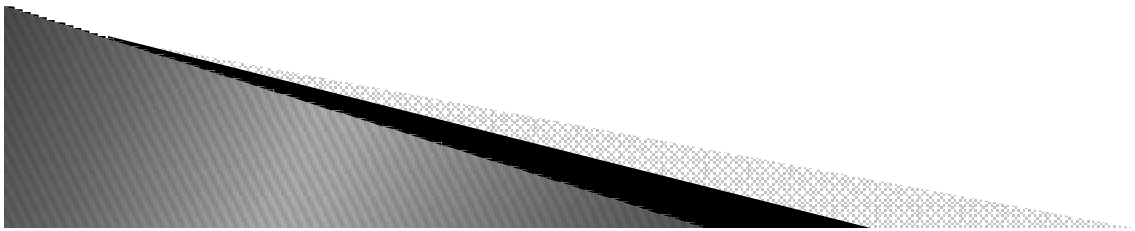
Dicionários e Arquivos

Gustavo Callou
gcallou@gmail.com



Dicionários

- Dicionários representam outra estrutura de dados interna de Python.
- Hash tables
- Listas → indexadas por inteiros
- Dicionários → indexados por chaves (keys), que podem ser de qualquer tipo imutável (como strings e inteiros)
- Têm comprimento variável, são heterogêneos e podem ser aninhados arbitrariamente.

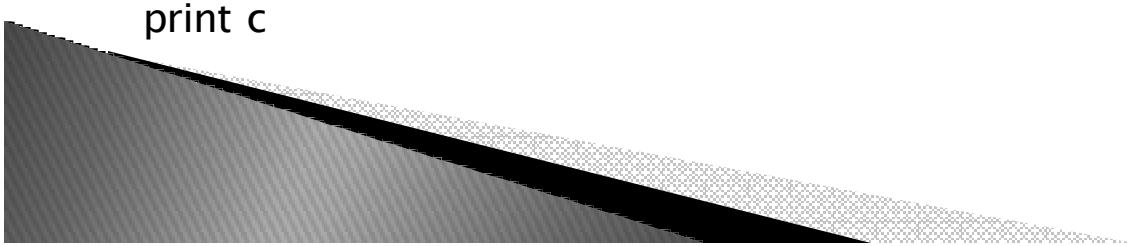


Dicionários

- São delimitados por : {}.
- Lista de pares chave → valor separada por vírgulas dentro desse delimitadores ({})

- Exemplo:

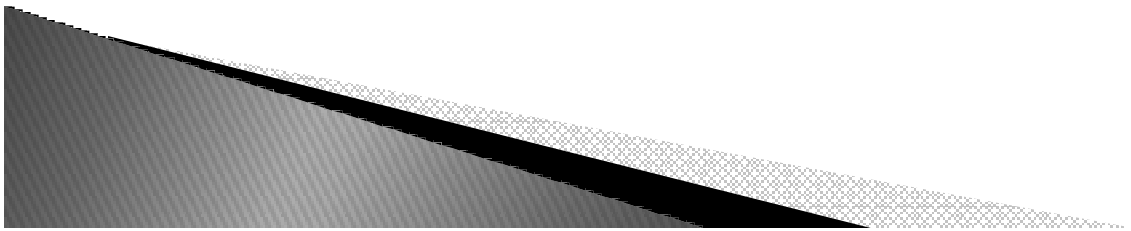
```
tel = {'jack': 4098, 'sape': 4139}
tel['guido'] = 4127
print tel
tel['irv'] = 4127
print tel
tel['irv'] = 1234
print tel
del tel['sape']
print tel
a = tel.keys()
print a
b = tel.has_key('guido')
print b
c = tel.has_key('amanda')
print c
```



Dicionários

Saídas:

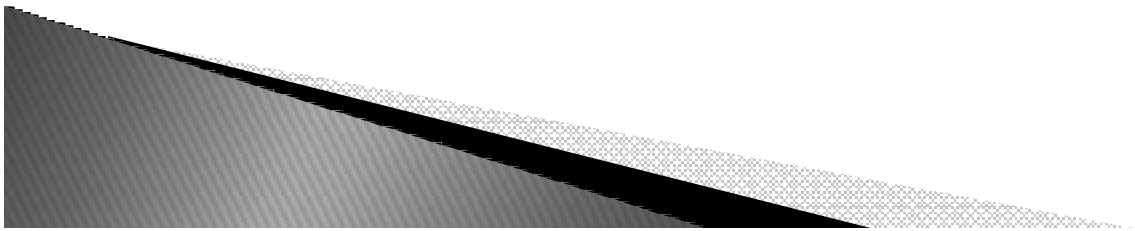
```
{'sape': 4139, 'jack': 4098, 'guido': 4127}
{'sape': 4139, 'jack': 4098, 'irv': 4127, 'guido': 4127}
{'sape': 4139, 'jack': 4098, 'irv': 1234, 'guido': 4127}
{'jack': 4098, 'irv': 1234, 'guido': 4127}
['jack', 'irv', 'guido']
True
False
```



Dicionários

- Mais exemplos:

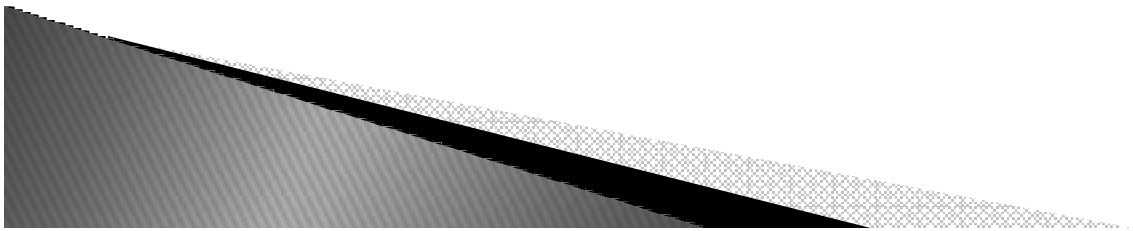
- D1 = {} # dicionario vazio
- D2 = {'spam' : 2, 'eggs' : 3} # dicionario de tamanho = 2
- D3 = {'food': {'ham': 1, 'egg': 2}} # aninhado
- print D3['food']
- print D3['food']['ham']
- print D2.has_key('eggs')
- print 'eggs' in D2
- print "keys", D2.keys()
- print "values", D2.values()
- print D2
- D2.get('eggs')
- D2.update(D3)
- print D2
- print len(D2)
- del D2['eggs']
- print D2



Dicionários

Saídas:

```
{'egg': 2, 'ham': 1}
1
True
True
keys ['eggs', 'spam']
values [3, 2]
{'eggs': 3, 'spam': 2}
{'food': {'egg': 2, 'ham': 1}, 'eggs': 3, 'spam': 2}
3
{'food': {'egg': 2, 'ham': 1}, 'spam': 2}
```



Dicionários

- Utilizado também para criar estruturas usadas em manipulaçao de arquivos e bancos.
- Exemplo:

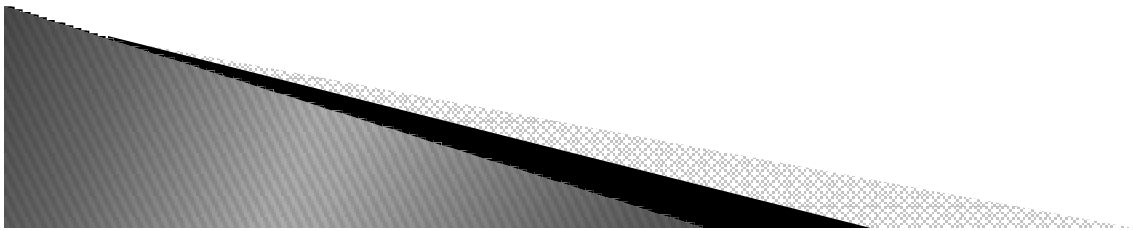
```
# records
bob = {'name': 'Bob Smith', 'age': 42, 'pay': 30000, 'job': 'dev'}
sue = {'name': 'Sue Jones', 'age': 45, 'pay': 40000, 'job': 'mus'}
tom = {'name': 'Tom',       'age': 50, 'pay': 0,     'job': None}

# database
db = {}
db['bob'] = bob
db['sue'] = sue
db['tom'] = tom
```

Matrizes com Dicionários

- Matrix = {}
- Matrix[(2,3)] = 88
- Matrix[(7,8)] = 99
- ...

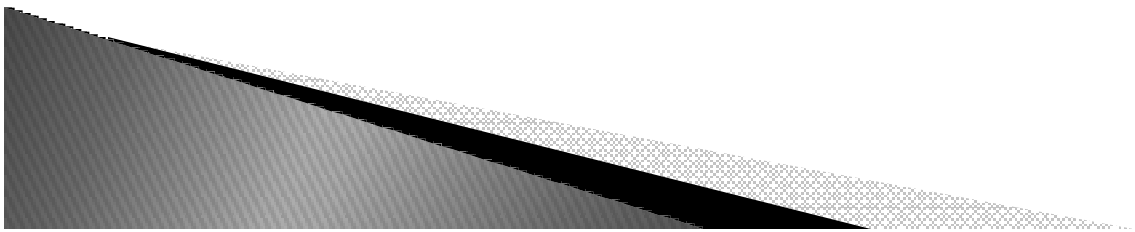
- Inicializando uma matriz:

```
m1 = {}
i, j = 3, 2
for a in range(0,i):
    for b in range(0,j):
        m1[(a,b)] = 0
```



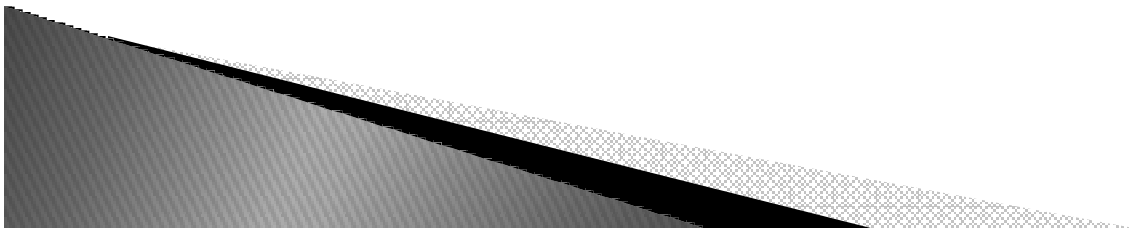
Dicionários

- Exercício:
- 1 - Fazer um programa que calcule e exiba a soma de duas matrizes.

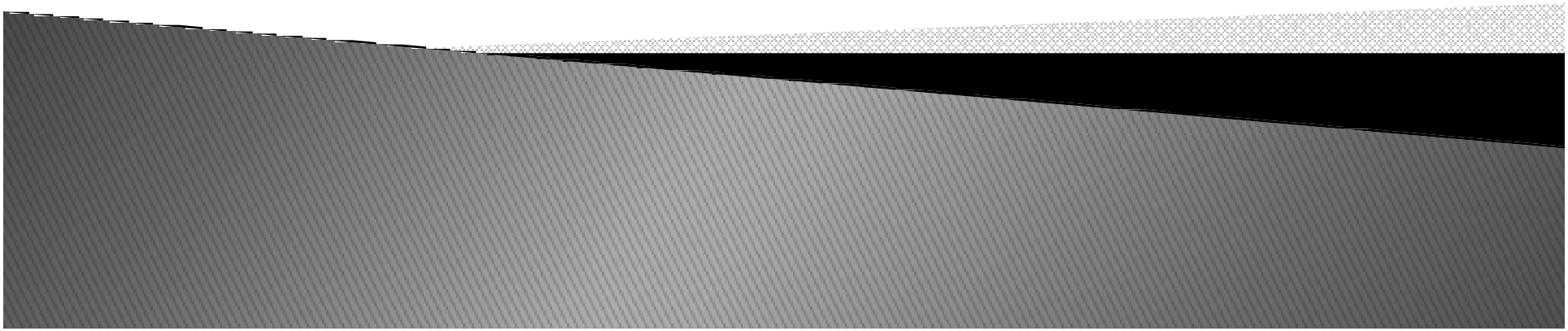


Exercício

- 2 – Fazer um programa que realiza a multiplicação entre duas matrizes.



Arquivos

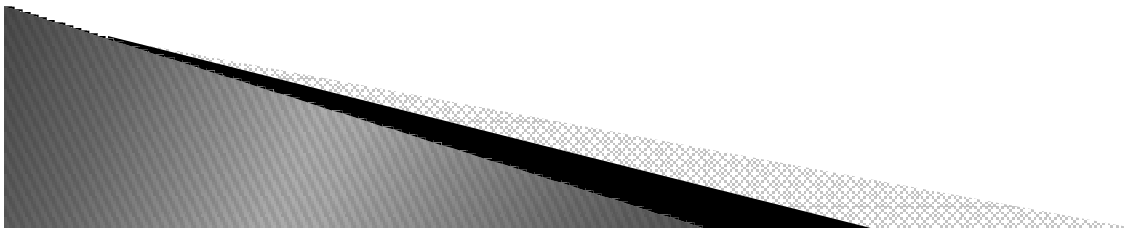


Arquivos

- A função **open** é usada com 2 argumentos:
  - open(filename, mode)

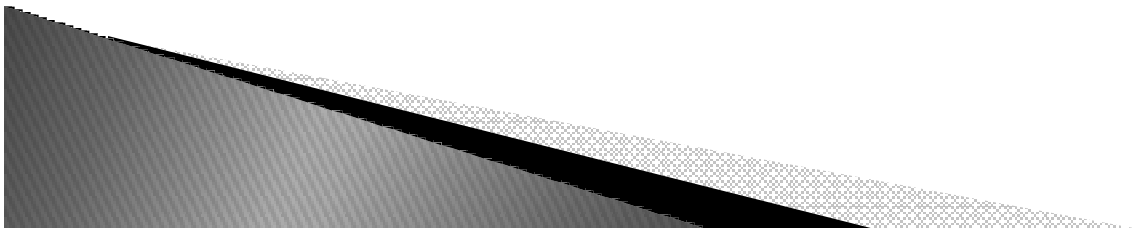
Mode :

- "r" (read) – abre somente para leitura
- "w" (write) –abre para escrita (se o arquivo existir terá o seu conteúdo apagado),
- "a" (append)– abre para anexar no final
- "r+" – abre tanto para escrita como para leitura
- Default: r
- Obs: No Win e no Mac → 'b' no mode indica formato binário
  - Exemplo: rb, wb, r+b



Arquivos

operação	Interpretação
output = open(“c:/teste.txt”,”w”)	Cria arquivo de saída (“w” significa gravação)
input = open(“c:/teste.txt”,”r”)	Cria arquivo de entrada (“r” significa leitura)
S = input.read()	Lê o arquivo inteiro em uma única string
S = input.read(N)	Lê N byte (1 ou mais)
S = input.readline()	Lê a próxima linha
L = input.readlines()	Cria uma lista onde cada elemento é uma linha do arquivo
output.write(S)	Grava a string S no arquivo
output.writelines(L)	Grava no arquivo todas as strings da lista L
output.close()	Fechamento manual do arquivo



Arquivos

- Exemplo:
  - myfile = open(“teste.txt”, “w”)
  - myfile.write(“hello text file\n”)
  - myfile.close()

  - myfile = open(“teste.txt”, “r”)
  - myfile.readline()
  - myfile.readline()

- Função rsplit()
  - “teste.teste1”.rsplit(".")

Exercícios

- Dado um arquivo contendo a identidade e o nome de cada pessoa, gere e exiba o conteúdo de um dicionário onde as chaves são as identidades e os valores os nomes.
  - 5384423 Manoel
  - 4345566 Alberto
  - 3235574 Mariana
  - ...

- Dados os seguintes endereços IPs, mostre os validos e os inválidos (um endereço ip válido não pode ter uma de suas partes maior que 224).
  - 200.135.80.9
  - 192.168.1.1
  - 8.35.67.74
  - 257.32.4.5
  - 85.345.1.2
  - 1.2.3.4
  - 9.8.234.5
  - 192.168.0.256