





BOOTCAMP CIÊNCIA DE DADOS

Numpy

por Thais Viana





**NumPy, abreviação de
Numerical Python, é um dos
mais importantes e
fundamentais pacotes de
computação numérica em
Python.**

PYTHON FOR DATA ANALYSIS, WES MCKINNEY







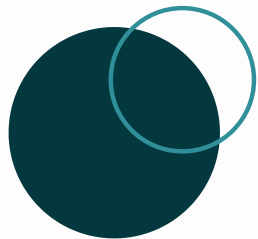
**A maioria dos pacotes
voltados para
funcionalidades científicas
usam NumPy's array objects
como lingua franca para
troca de dados.**

PYTHON FOR DATA ANALYSIS, WES MCKINNEY





NOSSA MOTIVAÇÃO



PORQUE NÓS ESTUDAMOS NUMPY?

Biblioteca otimizada para funções
matemáticas básicas para tudo que
faremos!



PONTOS PRINCIPAIS

ndarray

Funções Matemáticas

Entrada e saída de dados do array

Algebra Linear

C API

ndarray

An array object represents a multidimensional, homogeneous array of fixed-size items. An associated data-type object describes the format of each element in the array (its byte-order, how many bytes it occupies in memory, whether it is an integer, a floating point number, or something else, etc.)

COMPARANDO

```
In [7]: import numpy as np
```

```
In [8]: my_arr = np.arange(1000000)
```

```
In [9]: my_list = list(range(1000000))
```

```
In [10]: %time for _ in range(10): my_arr2 = my_arr * 2
```

```
CPU times: user 20 ms, sys: 10 ms, total: 30 ms
```

```
Wall time: 31.3 ms
```

```
In [11]: %time for _ in range(10): my_list2 = [x * 2 for x in my_list]
```

```
CPU times: user 680 ms, sys: 180 ms, total: 860 ms
```

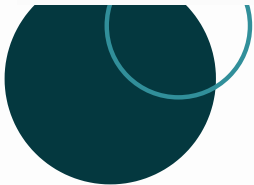
```
Wall time: 861 ms
```

GERAÇÃO DE NÚMEROS ALEATÓRIO

```
In [12]: import numpy as np

# Generate some random data
In [13]: data = np.random.randn(2, 3)

In [14]: data
Out[14]:
array([[ -0.2047,  0.4789, -0.5194],
       [-0.5557,  1.9658,  1.3934]])
```



OPERAÇÕES MATEMÁTICAS

```
In [15]: data * 10  
Out[15]:  
array([[ -2.0471,   4.7894,  -5.1944],  
       [ -5.5573,  19.6578,  13.9341]])
```

```
In [16]: data + data  
Out[16]:  
array([[ -0.4094,   0.9579,  -1.0389],  
       [ -1.1115,   3.9316,   2.7868]])
```



DATATYPES

```
In [33]: arr1 = np.array([1, 2, 3], dtype=np.float64)
```

```
In [34]: arr2 = np.array([1, 2, 3], dtype=np.int32)
```

```
In [35]: arr1.dtype
```

```
Out[35]: dtype('float64')
```

```
In [36]: arr2.dtype
```

```
Out[36]: dtype('int32')
```





ATENÇÃO

ARRAY INDEXING

```
>>> a[0]
0
>>> a[0] = 10
>>> a
array([10, 1, 2, 3])
```

FILL

```
# set all values in an array
>>> a.fill(0)
>>> a
array([0, 0, 0, 0])

# this also works, but may
# be slower
>>> a[:] = 1
>>> a
array([1, 1, 1, 1])
```



BEWARE OF TYPE COERCION

```
>>> a.dtype
dtype('int32')
```

```
# assigning a float into
# an int32 array truncates
# the decimal part
```

```
>>> a[0] = 10.6
>>> a
array([10, 1, 2, 3])
```

```
# fill has the same behavior
```

```
>>> a.fill(-4.8)
>>> a
array([-4, -4, -4, -4])
```

SLICING

```
>>> a[0, 3:5]  
array([3, 4])
```

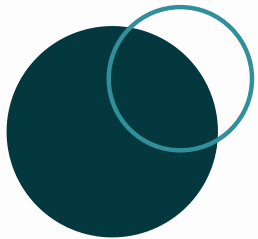
```
>>> a[4:, 4:]  
array([[44, 55],  
       [54, 55]])
```

```
>>> a[:, 2]  
a([2, 12, 22, 32, 42, 52])
```

```
>>> a[2::2, ::2]  
array([[20, 22, 24],  
       [40, 42, 44]])
```

0	1	2	3	4	5
10	11	12	13	14	15
20	21	22	23	24	25
30	31	32	33	34	35
40	41	42	43	44	45
50	51	52	53	54	55

SLICING



INSTALAÇÃO

PYTHON + MINICONDA

<https://docs.conda.io/en/latest/miniconda.html>

#1 DESAFIO

Contar o número de ocorrências de cada elemento em uma matriz.

Exemplo

$$A = \begin{bmatrix} 1 & 2 & 7 & 4 & 5 \\ 1 & 6 & 3 & 4 & 9 \\ 5 & 0 & 7 & 4 & 8 \end{bmatrix}$$

#1 DESAFIO



Contar o número de ocorrências de cada elemento em uma matriz.

Exemplo

$$A = \begin{bmatrix} 1 & 2 & 7 & 4 & 5 \\ 1 & 6 & 3 & 4 & 9 \\ 5 & 0 & 7 & 4 & 8 \end{bmatrix}$$

Resultado:

{0: 1, 1: 2, 2: 1, 3: 1, 4: 3, 5: 2, 6: 1, 7: 2, 8: 1, 9: 1}





2DESAFIO

Dada uma seqüência de informações sobre partidas de Par ou Ímpar (nomes dos jogadores e números que os jogadores escolheram), você deve escrever um programa para indicar o vencedor de cada uma das partidas.



2 DESAFIO



A entrada é composta de vários conjuntos de testes. A primeira linha de um conjunto de testes contém um inteiro N , que indica o número de partidas de Par ou Ímpar que aconteceram. As duas linhas seguintes contêm cada uma um nome de jogador. Um nome de jogador é uma cadeia de no mínimo um e no máximo dez letras (maiúsculas e minúsculas), sem espaços em branco. As N linhas seguintes contêm cada uma dois inteiros A e B que representam o número de dedos que cada jogador mostrou em cada partida ($0 \leq A \leq 5$ e $0 \leq B \leq 5$). Em todas as partidas, o primeiro jogador sempre escolhe Par. O final da entrada é indicado por $N = 0$.



2 DESAFIO



A entrada é composta de vários conjuntos de testes. A primeira linha de um conjunto de testes contém um inteiro N , que indica o número de partidas de Par ou Ímpar que aconteceram. As duas linhas seguintes contêm cada uma um nome de jogador. Um nome de jogador é uma cadeia de no mínimo um e no máximo dez letras (maiúsculas e minúsculas), sem espaços em branco. As N linhas seguintes contêm cada uma dois inteiros A e B que representam o número de dedos que cada jogador mostrou em cada partida ($0 \leq A \leq 5$ e $0 \leq B \leq 5$). Em todas as partidas, o primeiro jogador sempre escolhe Par. O final da entrada é indicado por $N = 0$.



2 DESAFIO (ENTRADA)



A entrada é composta de vários conjuntos de testes. A primeira linha de um conjunto de testes contém um inteiro N , que indica o número de partidas de Par ou Ímpar que aconteceram. As duas linhas seguintes contêm cada uma um nome de jogador. Um nome de jogador é uma cadeia de no mínimo um e no máximo dez letras (maiúsculas e minúsculas), sem espaços em branco. As N linhas seguintes contêm cada uma dois inteiros A e B que representam o número de dedos que cada jogador mostrou em cada partida ($0 \leq A \leq 5$ e $0 \leq B \leq 5$). Em todas as partidas, o primeiro jogador sempre escolhe Par. O final da entrada é indicado por $N = 0$.



Exemplo de entrada

```
3
Pedro
Paulo
2 4
3 5
1 0
2
Claudio
Carlos
1 5
2 3
0
```



2 DESAFIO (SAÍDA)

Para cada conjunto de teste da entrada, seu programa deve produzir a saída da seguinte forma. A primeira linha deve conter um identificador do conjunto de teste, no formato "Teste n", onde n é numerado seqüencialmente a partir de 1. As próximas N linhas devem indicar o nome do vencedor de cada partida. A próxima linha deve ser deixada em branco. A grafia mostrada no Exemplo de 3 Saída, abaixo, deve ser seguida rigorosamente.





Exemplo de Saída

Teste 1

Pedro

Pedro

Paulo

Teste 2

Claudio

Carlos



3 DESAFIO

A OBI (Organização de Bocha Internacional) é responsável por organizar a competição mundial de bocha. Infelizmente esse esporte não é muito popular, e numa tentativa de aumentar a sua popularidade, ficou decidido que seriam chamados, para a Grande Final Mundial, o campeão e o vice-campeão de cada sede nacional, ao invés de apenas o primeiro lugar.

Tumbólia é um país pequeno que já havia realizado a sua competição nacional quando a nova regra foi instituída, e o comitê local não armazenou quem foi o segundo classificado. Felizmente eles armazenaram a pontuação de todos competidores - que foram apenas três, devido ao tamanho diminuto do país. Sabe-se também que as pontuações de todos jogadores foram diferentes, de forma que não ocorreu empate entre nenhum deles. Resta agora descobrir quem foi o vice-campeão e para isso o comitê precisa de ajuda.

Entrada

A primeira e única linha da entrada consiste de três inteiros separados por espaços, A , B e C , as pontuações dos 3 competidores.

Saída

Imprima uma única linha na saída, contendo apenas um número inteiro, a pontuação do vice-campeão.

Restrições

- $1 \leq A \leq 100$
- $1 \leq B \leq 100$
- $1 \leq C \leq 100$

Exemplos

Entrada	Saída
4 5 6	5
Entrada	Saída
10 5 9	9