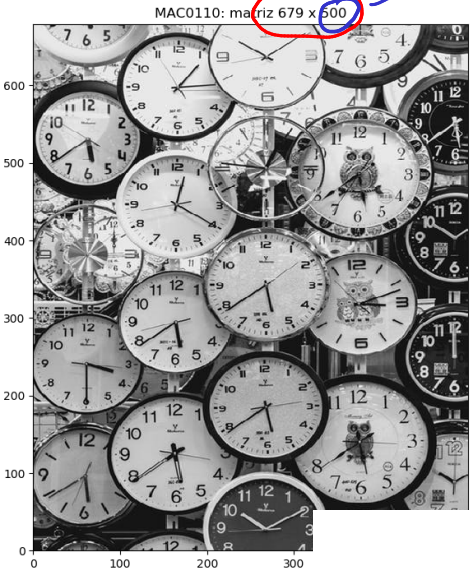
# 29 Reunião 29: 08/DEZ/2020

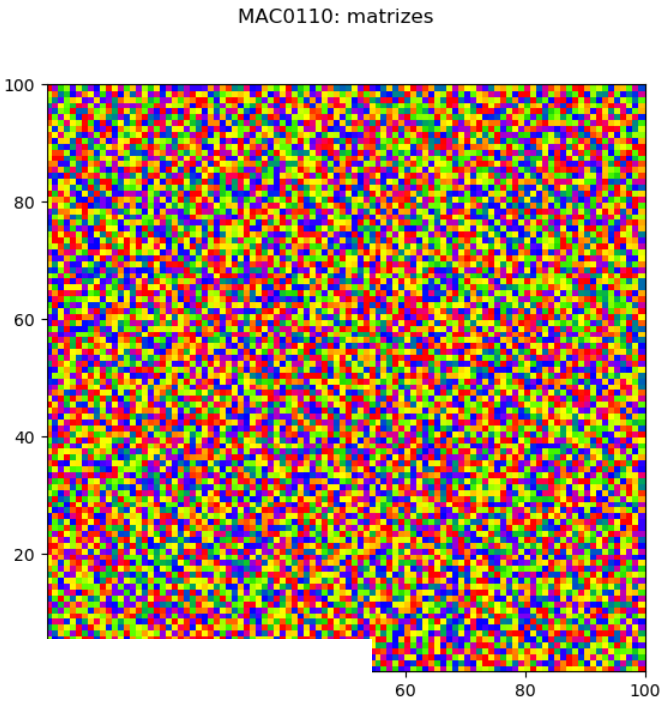
Figure 1: http://www.supercoloring.com/

## 29.1 Reuniões passadas

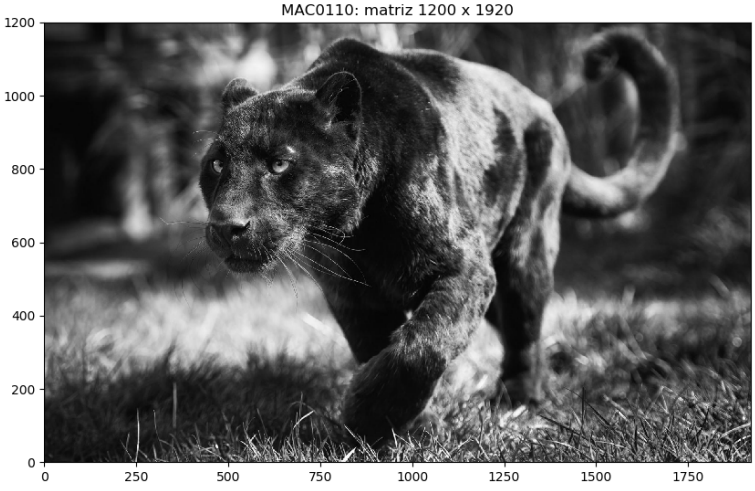
- matrizes
- matrizes em Python: lista de listas
- `init_matriz()`: variáveis, referências, apelidos,...
- `to_str()`: produz uma string usada para exibir a matriz
- `exiba_matriz()`: invólocro dafunção `to_string()`

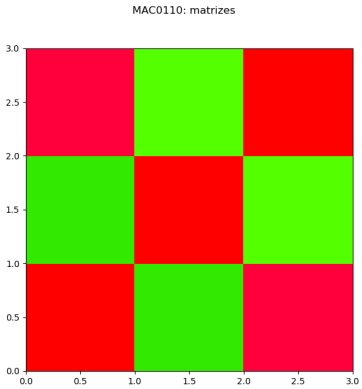
## 29.2 Matrizes







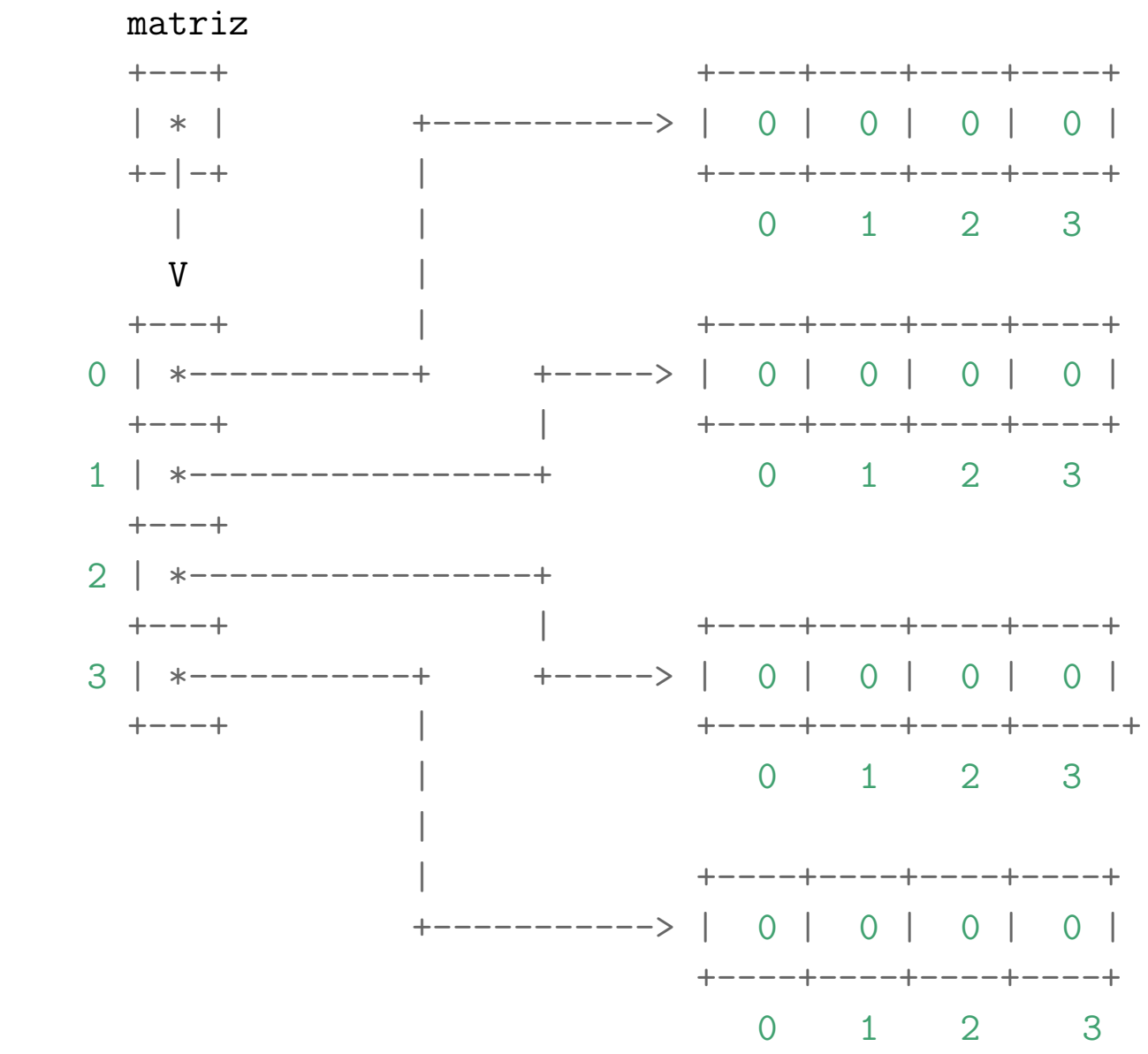
## 29.3 Manipulação de matrizes

Nossa biblioteca para manipulação de matrizes:

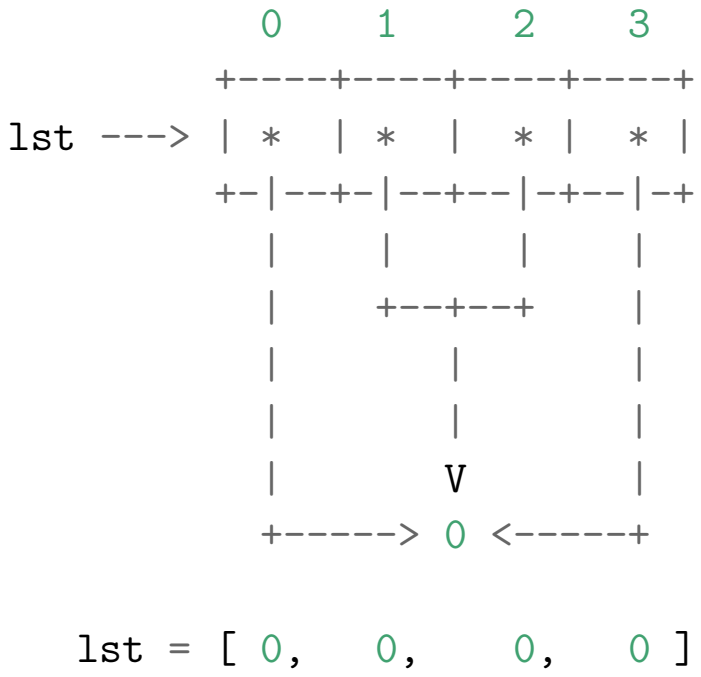
- `init_matriz(nlins, ncols, val=0)`: cria uma matriz de dimensão `nlins` × `ncols` `# REUNIÃO 28`
- `to_str(mt)`: cria uma string para ser usada por `print()` para exibir a `mt` `# REUNIÃO 28`
- `exiba_matriz(mt):` exibe `mt` na tela `# FEITA`
- `grave_matriz(mt, nome_arq)`: grama `mt` em um arquivo `# FAZER`
- `exiba_imagem(mt)`: mostra `mt` como imagem `# FAZER`
- `simetrica(mt)`: verifica se `mt` é simetrica `# REUNIÃO 27`
- `linhas_val(mt, val)`: conta numero de linhas com todos valores iguais a `val` `# REUNIÃO 27`
- `colunas_val(mt, val)`: conta numero de linhas com todos valores iguais a `val` `# REUNIÃO 27`
- `diagonais(mt, val)`: verifica se ambas diagonais têm apenas valor `val` `# REUNIÃO 27`
- `prod(mtA, mtB)`: retorna o produto de `mtA` por `mtB` `# FAZER`
- `leia_matriz(None)`: leitura de matriz `# FAZER`
- `leia_matriz_teclado()`: lê matriz do teclado `# FAZER`
- `leia_matriz_arquivo(nome_arq)`: lê matriz de arquivo `# FAZER`
- `rode_dir(mt)`: roda matriz para direita `# FAZER`
- `rode_esq(mt)`: roda matriz para esquerda `# FAZER`
- `gire_horizontal(mt)`: reflete `mt` horizontamente `# FAZER`
- `give_vertical(mt)`: reflete `mt`verticalmente `# FAZER`

## 29.4 init_matriz()

```
init_matriz(4, 4, 0)
```

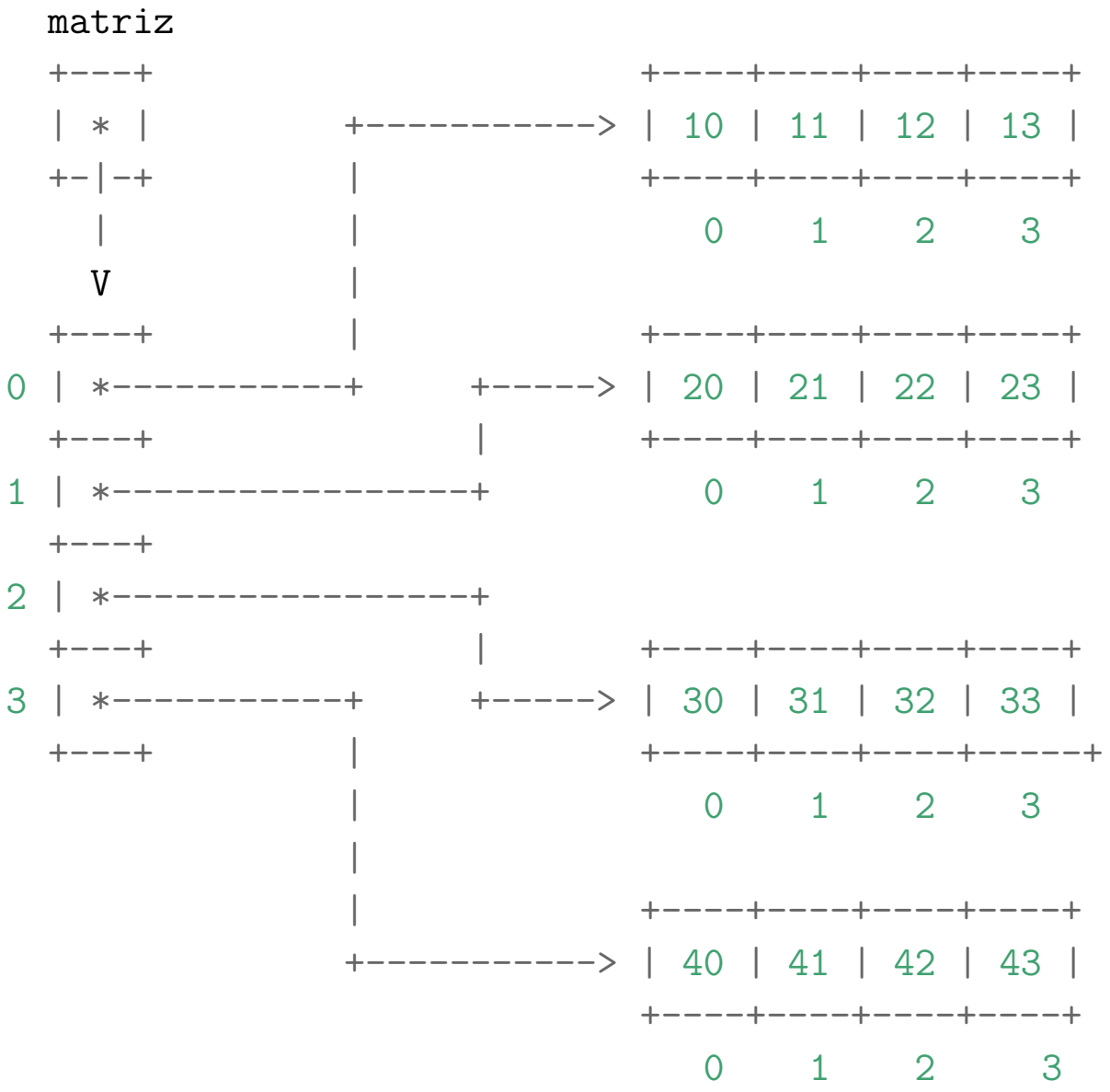


## Rascunhos




```
lst = [ 0,   0,    0,   0 ]
```

## Rascunhos


matriz
0
1
2
3
10 11 12 13
0 1 2 3
20 21 22 23
0 1 2 3
30 31 32 33
0 1 2 3
40 41 42 43
0 1 2 3

## 29.5 exiba_matriz() e to_str()

```
In [2]: m = init_matriz(5,4)

In [3]: exiba_matriz(m)
Matriz: 5 x 4
 0 0 0 0
 0 0 0 0
 0 0 0 0
 0 0 0 0
 0 0 0 0


In [4]: to_str(m)
Out[4]: 'Matriz: 5 x 4\n 0 0 0 0\n 0 0 0 0\n 0 0 0 0\n 0 0 0 0\n 0 0 0

In [5]: m = [[11,12,13], [21, 22, 23], [31, 32, 33], [41, 42,

In [6]: exiba_matriz(m)
Matriz: 4 x 3
 11 12 13
 21 22 23
 31 32 33
 41 42 43


In [7]: m = [[11,12,13], [21, 22, 23], [31, 32, 33], [41, 42,

In [8]: to_str(m)
Out[8]: 'Matriz: 4 x 3\n 11 12 13\n 21 22 23\n 31 32 33\n 41
```

## 29.6 leia_matriz()

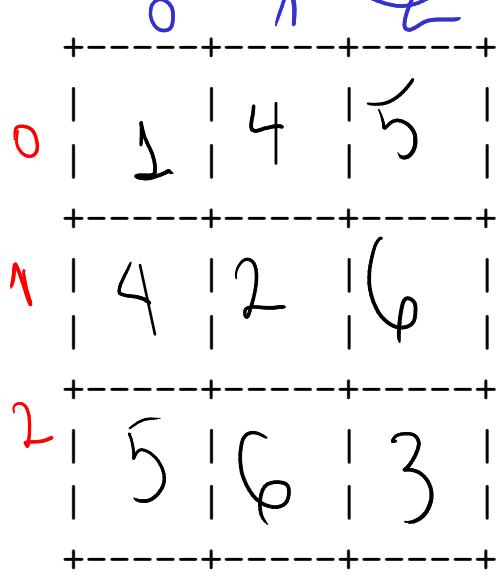
```
def leia_matriz(nome_arq = None):
    '''(None ou str) -> matriz (list[list])

    Se `nome_arq` é `None` lê do teclado os elemento de uma matr
    Em caso contrário os elementos serão lidos do arquivo `noma_
    '''
    if nome_arq == None:
        mt = leia_matriz_teclado()
    else:
        mt = leia_matriz_arquivo(nome_arq)
    return mt
```

## 29.7 leia_matriz_teclado()

### 29.7.1 Exemplos

```
Digite o número de linhas: 3
Digite o número de colunas: 3

Digite o elemento [0][0]: 11
Digite o elemento [0][1]: 12
Digite o elemento [0][2]: 13
Digite o elemento [1][0]: 21
Digite o elemento [1][1]: 22
Digite o elemento [1][2]: 23
Digite o elemento [2][0]: 31
Digite o elemento [2][1]: 32
Digite o elemento [2][2]: 33
Matriz: 3 x 3
    11      12      13
    21      22      23
    31      32      33



Digite o número de linhas: 3
Digite o número de colunas: 3

Digite o elemento [0][0]: 1
Digite o elemento [0][1]: 4
Digite o elemento [0][2]: 5
Digite o elemento [1][0]: 4
Digite o elemento [1][1]: 2
Digite o elemento [1][2]: 6
Digite o elemento [2][0]: 5
Digite o elemento [2][1]: 6
Digite o elemento [2][2]: 3
```

```
Matriz: 3 x 3
     1       4       5
     4       2       6
     5       6       3

init_matriz(nlins,ncols)
```

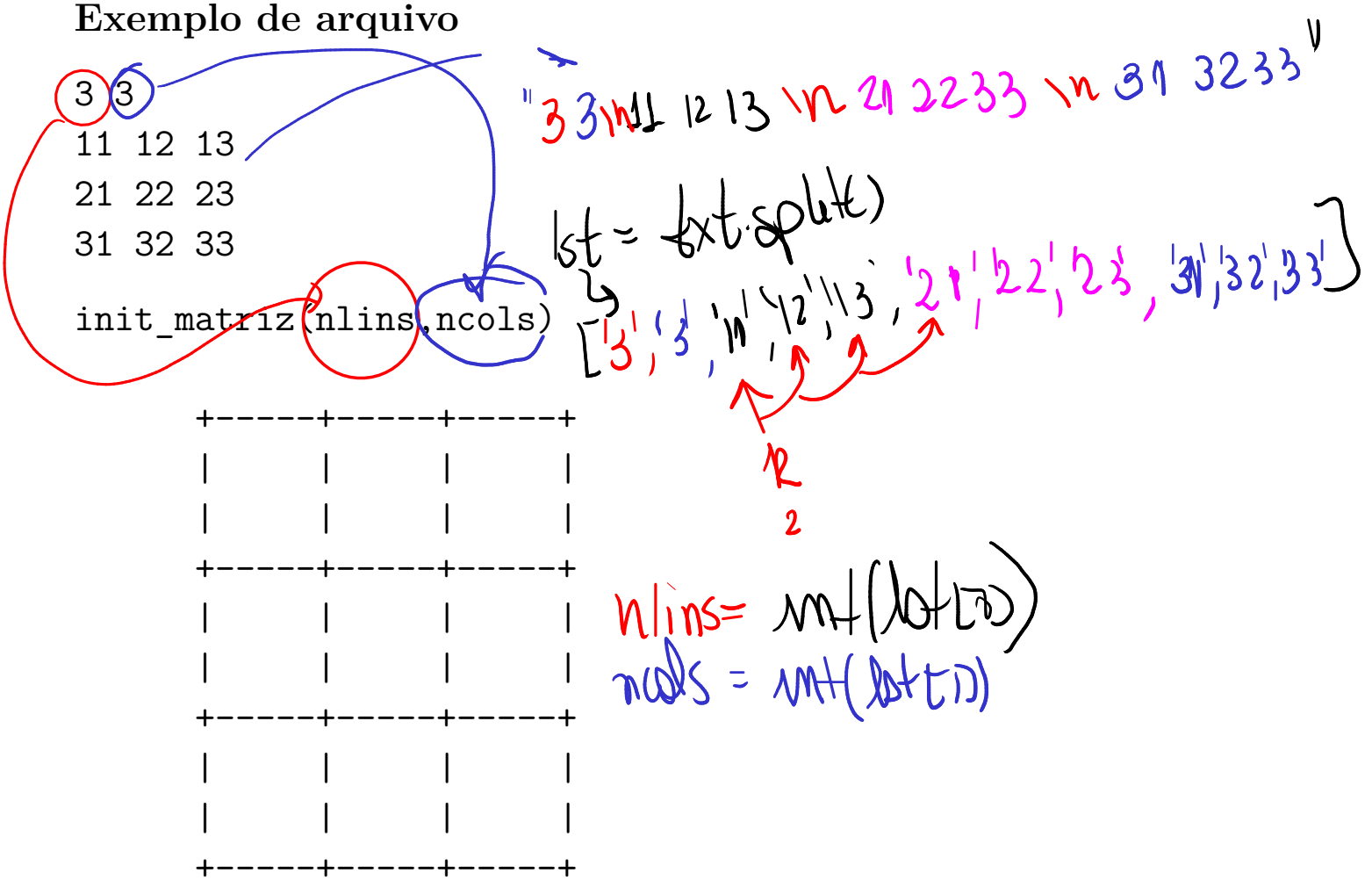


### Solução

```
#-------------------------------------------------------------------
def leia_matriz_teclado():
    '''(None) -> matriz (list[list])

    LÊ do teclado os elementos de uma matriz .
    RETORNA uma lista de listas representando a matriz.
    '''
    # 1 leia as dimensões
    nlins = int(input("Digite o no. de linha: "))
    ncols = int(input("Digite o no. de colunas: "))

    # crie a matriz
    mt = init_matriz(nlins, ncols)

    # preencha a matriz
    for i in range(nlins):
        # leia linha i
        for j in range(ncols):
            valor = int(input(f"Digite elem [{i}][{j}]: "))
            mt[i][j] = valor
    return mt
```

## 29.8 leia_matriz_arquivo()

**Exemplo de arquivo**




No final tem um texto sobre leitura de arquivos.

### Solução

```
def leia_matriz_arquivo(nome_arq):
    '''(str) -> matriz (list de list)

    RETORNA uma matriz lida de um arquivo.

    A função supõe que os dois primeiros valores do arquivo
    são número inteiros representando a dimensão da matriz:

        número de linha nlins e número de coluna ncols.

    Em seguida devem estar dispostos o nlins por ncols
    elementos da matriz.

    Todos os valores devem estar separados por pelo menos
    um branco.
    '''
    # 1 abra o arquivo para leitura
    arq =  open(nome_arq, 'r', encoding='utf-8')
    # 2 leia o conteúdo
    arq_str = arq.read()
    # 3 feche o arquivo
    arq.close()

    # 4 coloque os elementos no string em uma lista
    arq_lst = arq_str.split()

    # 5 pegue a dimensão da matriz
    nlins = int(arq_lst[0])
    ncols = int(arq_lst[1])

    # 6 crie a matriz
```

```
mt = init_matriz(nlins, ncols)

# coloque os elementos do arquivo em suas posições na matriz
k = 2
for i in range(0, nlins, 1):
    for j in range(0, ncols, 1):
        mt[i][j] = int(arq_lst[k])
        k += 1
return mt
```

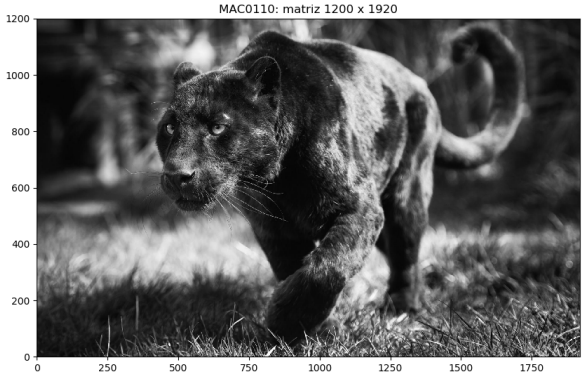
## 29.9 gire_horizontal()

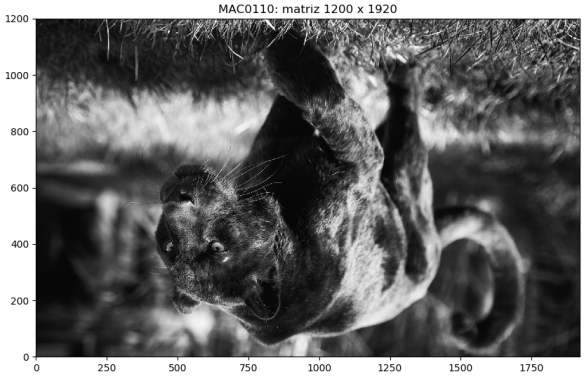
| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | a | b | c | d |
| 1 | e | f | g | h |
| 2 | i | j | k | l |
| 3 | m | n | o | p |
| 4 | q | r | s | t |

flipH() ----->

| | 0 | 1 | 2 | 3 |
|---|---|---|---|---|
| 0 | q | r | s | t |
| 1 | m | n | o | p |
| 2 | i | j | k | l |
| 3 | e | f | g | h |
| 4 | a | b | c | d |

horizontal[i][j] = mat[n_lins - 1 - i][j]



## 29.13 Arquivos

Em Python há pelo menos quatro maneira diferentes de lermos um arquivo. Cada uma delas tem sua utilidade. Até o momento vimos duas delas.

Nos exemplos a seguir utilizaremos o arquivo `jeff.txt`.

```
We hold these truths to be self-evident:
that all men are created equal;
that they are endowed by their Creator
with certain unalienable rights;
that among these are life, liberty,
and the pursuit of happiness.
```

### `read()`

Se `arq` é um arquivo, `arq.read()` retorna uma string com todo o conteúdo do arquivo. Os caracteres de *nova linha* `'\n'` no arquivo estão na string.

Exemplo de leitura com `read()`

```
def leitura_com_read():
    ''' (None) -> None

    Exemplo de leitura de um arquivo com read()
    '''
    print("Leitura com read()")
    # 1 abra o arquivo
    arq = open(NOME_ARQUIVO, 'r', encoding='utf-8')
    # 2 leia o conteúdo do arquivo
    txt = arq.read() # retorna uma string
    # 3 feche o arquivo
    arq.close()
    # 4 exiba o conteúdo do arquivo
    print(f"Conteúdo do arquivo '{NOME_ARQUIVO}'")
    print(txt)
```

Saída produzida

```
Leitura com read()
Conteúdo do arquivo 'jeff.txt'
We hold these truths to be self-evident:
that all men are created equal;
that they are endowed by their Creator
with certain unalienable rights;
that among these are life, liberty,
and the pursuit of happiness.
```

**`for linha in arq:`**

Se **`arq`** é um arquivo, **`for linha in arq:`** lê o conteúdo de um arquivo linha a linha. No caso, a variável **`linha`** recebe em cada iteração uma string com o conteúdo de cada linha do arquivo. Os caracteres de *nova linha* **`'\n'`** no arquivos estão nas strings.

Exemplo de leitura com **`for ...`**

```
def leitura_com_for():
    '''(None) -> None

    Exemplo de leitura de um arquivo percorrendo suas
    linhas com for ...
    '''
    print("Leitura com for ..")
    # 1 abra o arquivo
    arq =open(NOME_ARQUIVO, 'r', encoding='utf-8')
    # 2 leia o arquivo linha a linha
    # 1 abra o arquivo
    arq = open(NOME_ARQUIVO, 'r', encoding='utf-8')
    # 2 leia o conteúdo do arquivo linha a linha
    print(f"Conteúdo do arquivo '{NOME_ARQUIVO}'")
    i = 0
    for linha in arq:
        print(f"{i}: {linha}")
        i += 1
    # 3 feche o arquivo
    arq.close()
```

Saída produzida

```
Leitura com for ..
Conteúdo do arquivo 'jeff.txt'
0: We hold these truths to be self-evident:
```

```
1: that all men are created equal;

2: that they are endowed by their Creator

3: with certain unalienable rights;

4: that among these are life, liberty,

5: and the pursuit of happiness.
```

**readline()**

Se **arq** é um arquivo, **arq.reasline()** retorna uma string com o conteúdo de uma linha do arquivo. Um novo **arq.readline()** retorna a proxima linha, e mais um novo **arq.readline()** retorna a próxima linha e assim até o final do arquivo se atingido. Os caracteres de *nova linha* **'\n'** no arquivo estão nas strings.

Exemplo de leitura com **readline()**

```
def leitura_com_readline():
    '''(None) -> None

    Exemplo de leitura de um arquivo com readline()
    '''
    print("Leitura com readline()")
    # 1 abra o arquivo
    arq = open(NOME_ARQUIVO, 'r', encoding='utf-8')
    # 2 leia o conteúdo do arquivo linha a linha
    print(f"Conteúdo do arquivo '{NOME_ARQUIVO}'")
    i = 0
    linha = arq.readline()
    while linha != '':
        print(f"{i}: {linha}")
        i += 1
        linha = arq.readline()
    # 3 feche o arquivo
    arq.close()
```

Saída produzida

```
Leitura com readline()
Conteúdo do arquivo 'jeff.txt'
0: We hold these truths to be self-evident:
```

```
1: that all men are created equal;

2: that they are endowed by their Creator

3: with certain unalienable rights;

4: that among these are life, liberty,

5: and the pursuit of happiness.
```

### `readlines()`

Se `arq` é um arquivo, `arq.reaslines()` retorna uma lista em que cada item é uma string com o conteúdo de uma linha do arquivo. Os caracteres de *nova linha* `\n` no arquivo estão nas strings.

Exemplo de leitura com `readlines()`

```
def leitura_com_readlineS():
    '''(None) -> None

    Exemplo de leitura de um arquivo com readline()
    '''
    print("Leitura com readlines()")
    # 1 abra o arquivo
    arq = open(NOME_ARQUIVO, 'r', encoding='utf-8')
    # 2 leia o conteúdo do arquivo linha a linha
    lst_linhas = arq.readlines()
    # 3 feche o arquivo
    arq.close()
    # 4 exiba o conteúdo do arquivo
    for i in range(len(lst_linhas)):
        linha = lst_linhas[i]
        print(f"{i}: {linha}")
```

Saída produzida

```
Leitura com readlines()
0: We hold these truths to be self-evident:

1: that all men are created equal;

2: that they are endowed by their Creator

3: with certain unalienable rights;
```

```
4: that among these are life, liberty,

5: and the pursuit of happiness.
```