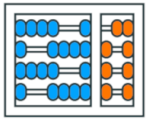


Instituto de Computação

UNIVERSIDADE ESTADUAL DE CAMPINAS

UNICAMP

MC102 - Aula 14

Objetos Multidimensionais

Algoritmos e Programação de Computadores

Turmas

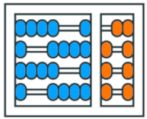
OVXZ

Prof. Lise R. R. Navarrete

lrommel@ic.unicamp.br

Terça-feira, 10 de maio de 2022

21:00h - 23:00h (CB06)



Instituto de Computação

UNIVERSIDADE ESTADUAL DE CAMPINAS

UNICAMP

MC102 – Algoritmos e Programação de Computadores

Turmas

OVXZ

https://ic.unicamp.br/~mc102/

Site da Coordenação de MC102

Aulas teóricas:

Terça-feira, 21:00h - 23:00h (CB06)

Quinta-feira, 19:00h - 21:00h (CB06)

Conteúdo

- Objetos Multidimensionais
  - Exemplo 1: Listas Multidimensionais
  - Exemplo 2: Tuplas Multidimensionais
  - Exemplo 3: Sets Multidimensionais
  - Exemplo 4: Dicionários Multidimensionais
- Matrices
- Hipermatrices, arranjos multidimensionais
- Exercícios

Objetos Multidimensionais

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

- ... objetos multidimensionais são generalizações de objetos simples vistos anteriormente (listas e tuplas).
- Esses tipos de dados nos permitem armazenar informações mais complexas em uma única variável.
- Exemplo de informações/operações que podem ser armazenadas/manipuladas utilizando matrizes e objetos multidimensionais:
  - Matemática: operações com matrizes.
  - Processamento de imagem: cor de cada pixel presente na imagem.
  - Mapas e geolocalização: informação sobre o relevo em cada ponto do mapa.
  - Jogos de tabuleiro: Xadrez, Damas, Go, Batalha Naval, etc.

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

- Uma lista pode conter elementos de tipos diferentes.
- Uma lista pode conter inclusive outras listas.
- Exemplo de declaração de uma lista:

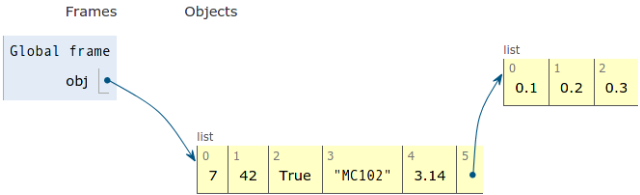
```
obj = [
  7, 42, True, "MC102", 3.14,
  [0.1, 0.2, 0.3]
]
```

- Exemplo de declaração de um objeto multidimensional:

```
obj = [
  [1, 2, 3, 4],
  [5, 6],
  [7, 8, 9]
]
```

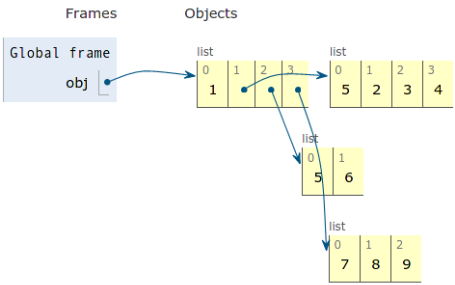
https://pythontutor.com/visualize.html#mode=edit

```
1
2 obj = [
3           7, 42, True , "MC102", 3.14,
4           [0.1, 0.2, 0.3]
5         ]
```



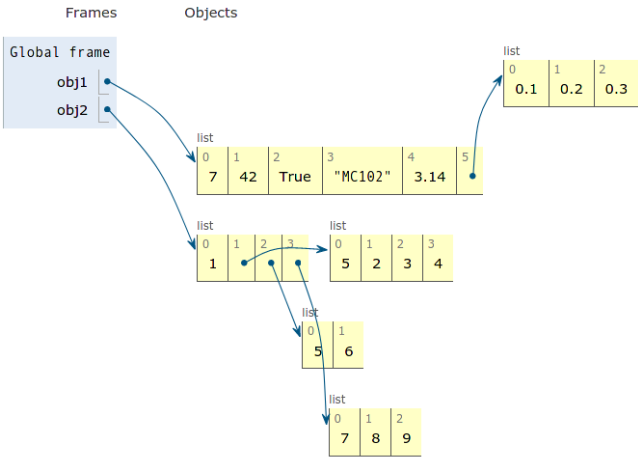
https://pythontutor.com/visualize.html#mode=edit

```
1
2 obj = [ 1,
3           [5, 2, 3, 4],
4           [5, 6],
5           [7, 8, 9]
6         ]
```



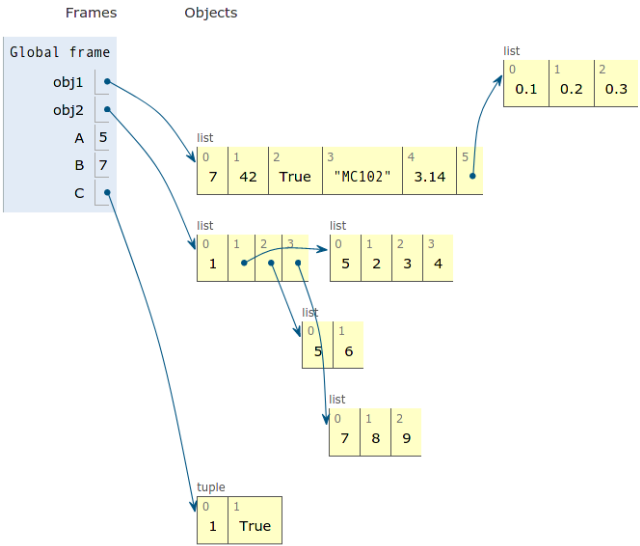
https://pythontutor.com/visualize.html#mode=edit

```
1  obj1 = [
2           7, 42, True , "MC102", 3.14,
3           [0.1, 0.2, 0.3]
4         ]
5
6  obj2 = [ 1,
7           [5, 2, 3, 4],
8           [5, 6],
9           [7, 8, 9]
10        ]
```



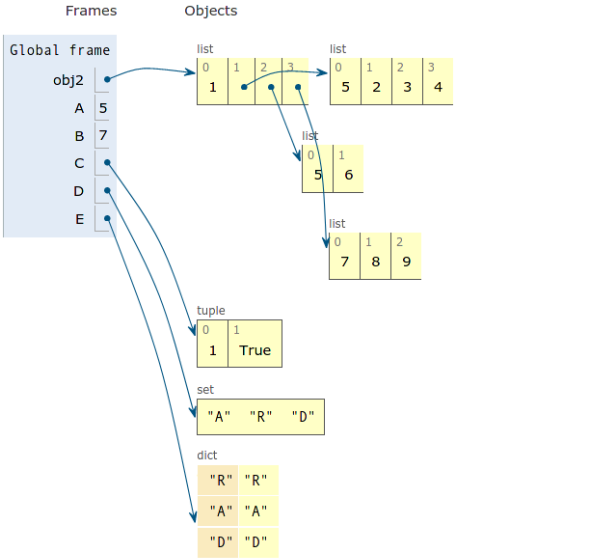
https://pythontutor.com/visualize.html#mode=edit

```
1  obj1 = [
2           7, 42, True , "MC102", 3.14,
3           [0.1, 0.2, 0.3]
4         ]
5
6  obj2 = [ 1,
7           [5, 2, 3, 4],
8           [5, 6],
9           [7, 8, 9]
10        ]
11
12 A = 5
13
14 B = 7
15
16 C = (1,True)
```



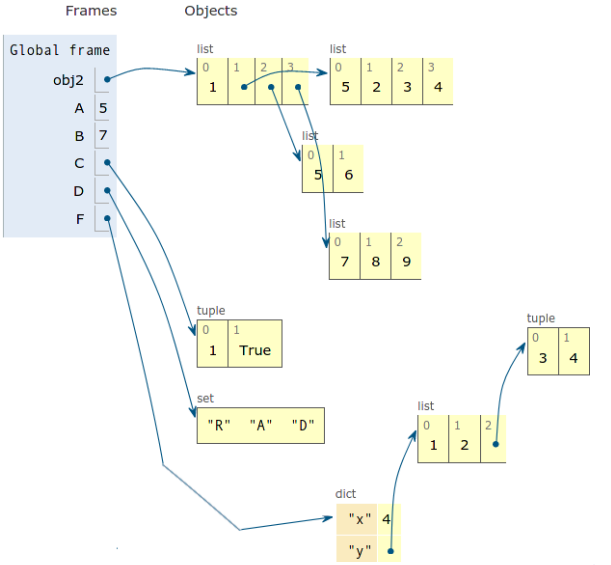
https://pythontutor.com/visualize.html#mode=edit

```
1  obj2 = [ 1,
2           [5, 2, 3, 4],
3           [5, 6],
4           [7, 8, 9]
5         ]
6
7  A = 5
8
9  B = 7
10
11 C = (1,True)
12
13 D = set("RADAR")
14
15 E = dict(zip("RADAR","RADAR"))
```



https://pythontutor.com/visualize.html#mode=edit

```
1  obj2 = [ 1,
2           [5, 2, 3, 4],
3           [5, 6],
4           [7, 8, 9]
5         ]
6
7  A = 5
8
9  B = 7
10
11 C = (1,True)
12
13 D = set("RADAR")
14
15 F = dict(x=4, y=[1,2,(3,4)])
```



Exemplo 1: Listas Multidimensionais

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  X = [ 1,
2          [5, 2, 3, 4],
3          [5, 6],
4          [7, 8, 9]
5        ]
6
7  X[0] = (1,2)
8
9  X[1][2] = "RADAR"
10
11 X[3][2] = ["CAT", ["D","O","G"]]
12
13 X[3][2][1][2] = "T"
```

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 1 of 8

Frames Objects

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  X = [ 1,
2          [5, 2, 3, 4],
3          [5, 6],
4          [7, 8, 9]
5        ]
6
7  X[0] = (1,2)
8
9  X[1][2] = "RADAR"
10
11 X[3][2] = ["CAT", ["D","O","G"]]
12
13 X[3][2][1][2] = "T"
```

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 2 of 8

Frames Objects

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  X = [ 1,
2          [5, 2, 3, 4],
3          [5, 6],
4          [7, 8, 9]
5        ]
6
7  X[0] = (1,2)
8
9  X[1][2] = "RADAR"
10
11 X[3][2] = ["CAT", ["D","O","G"]]
12
13 X[3][2][1][2] = "T"
```

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 3 of 8

Frames Objects

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  X = [ 1,
2          [5, 2, 3, 4],
3          [5, 6],
4          [7, 8, 9]
5        ]
6
7  X[0] = (1,2)
8
9  X[1][2] = "RADAR"
10
11 X[3][2] = ["CAT", ["D","O","G"]]
12
13 X[3][2][1][2] = "T"
```

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 4 of 8

Frames Objects

https://pythontutor.com/visualize.html#mode=edit

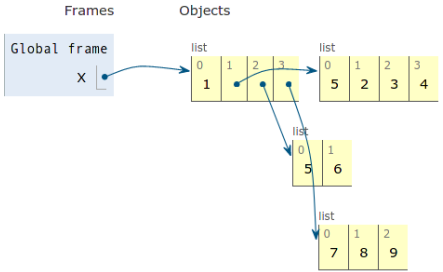
Python 3.6
(known limitations)

```
1  X = [ 1,
2          [5, 2, 3, 4],
3          [5, 6],
4          [7, 8, 9]
5        ]
6
7  X[0] = (1,2)
8
9  X[1][2] = "RADAR"
10
11 X[3][2] = ["CAT", ["D","O","G"]]
12
13 X[3][2][1][2] = "T"
```

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 5 of 8



https://pythontutor.com/visualize.html#mode=edit

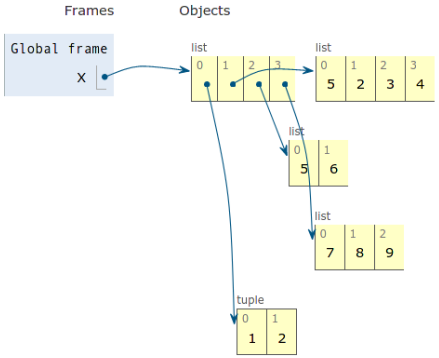
Python 3.6
(known limitations)

```
1  X = [ 1,
2          [5, 2, 3, 4],
3          [5, 6],
4          [7, 8, 9]
5        ]
6
7  X[0] = (1,2)
8
9  X[1][2] = "RADAR"
10
11 X[3][2] = ["CAT", ["D","O","G"]]
12
13 X[3][2][1][2] = "T"
```

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 6 of 8



https://pythontutor.com/visualize.html#mode=edit

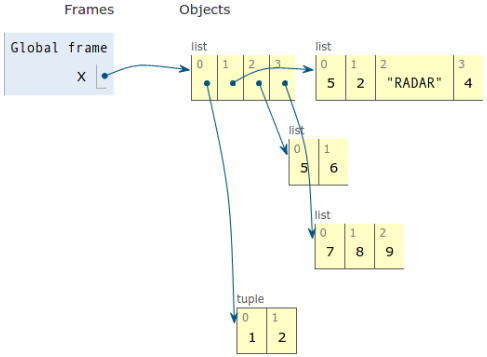
Python 3.6
(known limitations)

```
1  X = [ 1,
2          [5, 2, 3, 4],
3          [5, 6],
4          [7, 8, 9]
5        ]
6
7  X[0] = (1,2)
8
9  X[1][2] = "RADAR"
10
11 X[3][2] = ["CAT", ["D","O","G"]]
12
13 X[3][2][1][2] = "T"
```

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 7 of 8



https://pythontutor.com/visualize.html#mode=edit

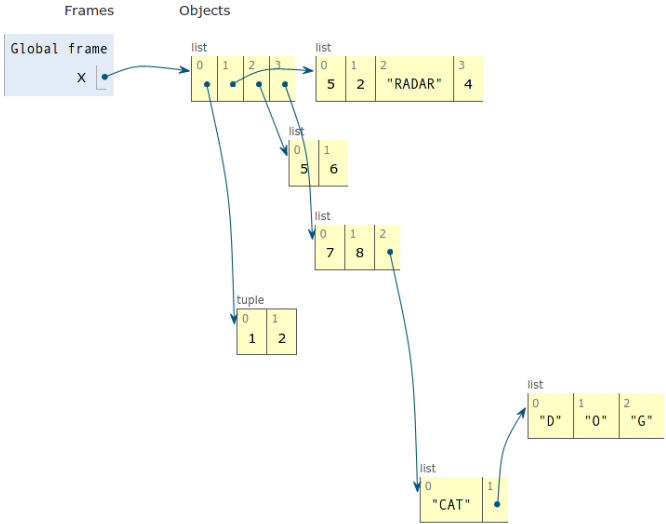
Python 3.6
(known limitations)

```
1  X = [ 1,
2          [5, 2, 3, 4],
3          [5, 6],
4          [7, 8, 9]
5        ]
6
7  X[0] = (1,2)
8
9  X[1][2] = "RADAR"
10
11 X[3][2] = ["CAT", ["D","O","G"]]
12
13 X[3][2][1][2] = "T"
```

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 8 of 8



https://pythontutor.com/visualize.html#mode=edit

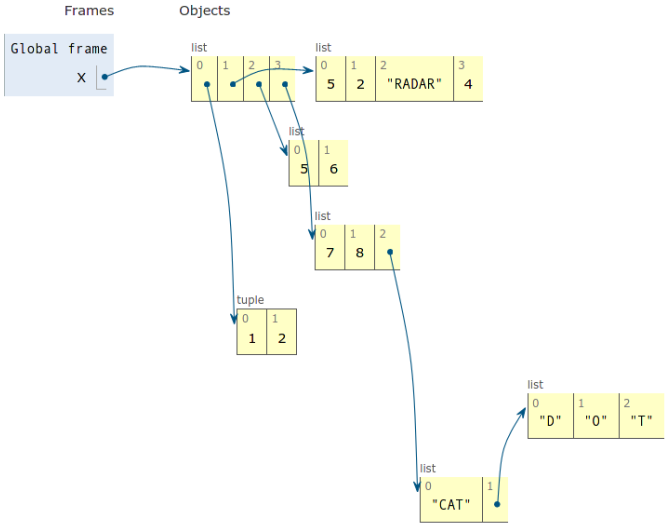
Python 3.6
(known limitations)

```
1  X = [ 1,
2          [5, 2, 3, 4],
3          [5, 6],
4          [7, 8, 9]
5        ]
6
7  X[0] = (1,2)
8
9  X[1][2] = "RADAR"
10
11 X[3][2] = ["CAT", ["D","O","G"]]
12
13 X[3][2][1][2] = "T"
```

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Done running (8 steps)



Exemplo 2: Tuplas Multidimensionais

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  ( 1,
2         [5, 2, 3, 4],
3         [5, 6],
4         (7, 8, 9)
5       )
6  Y[1][0] = "RADAR"
7  Y[1][2] = Y[3][2]
8  Y[2][1] = Y[3]
9  Y[1][1] = Y[2][1][1]
10 Y[1][3] = Y[1]
11 Y[2][0] = Y[1][3][3][3][3][3][3][3][2] + 2
12 Y[2][0] **= 2
13 Y[1][3][3][3][3][3][3][3][2] = Y[2]
14 print(Y[1][3][3][3][3][3][3][3][2][0])
15 print("End")
```

Edit this code

line that just executed
next line to execute

Print output (drag lower right corner to resize)

Frames Objects

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  ( 1,
2         [5, 2, 3, 4],
3         [5, 6],
4         (7, 8, 9)
5       )
6  Y[1][0] = "RADAR"
7  Y[1][2] = Y[3][2]
8  Y[2][1] = Y[3]
9  Y[1][1] = Y[2][1][1]
10 Y[1][3] = Y[1]
11 Y[2][0] = Y[1][3][3][3][3][3][3][3][2] + 2
12 Y[2][0] **= 2
13 Y[1][3][3][3][3][3][3][3][2] = Y[2]
14 print(Y[1][3][3][3][3][3][3][3][2][0])
15 print("End")
```

Edit this code

line that just executed
next line to execute

Print output (drag lower right corner to resize)

Frames Objects

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  ( 1,
2           [5, 2, 3, 4],
3           [5, 6],
4           (7, 8, 9)
5       )
6  Y[1][0] = "RADAR"
7  Y[1][2] = Y[3][2]
8  Y[2][1] = Y[3]
9  Y[1][1] = Y[2][1][1]
10 Y[1][3] = Y[1]
11 Y[2][0] = Y[1][3][3][3][3][3][3][3][2] + 2
12 Y[2][0] **= 2
13 Y[1][3][3][3][3][3][3][3][2] = Y[2]
14 print(Y[1][3][3][3][3][3][3][3][2][0])
15 print("End")
```

Edit this code

line that just executed
next line to execute

Print output (drag lower right corner to resize)

Frames Objects

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  ( 1,
2         [5, 2, 3, 4],
3         [5, 6],
4         (7, 8, 9)
5       )
6  Y[1][0] = "RADAR"
7  Y[1][2] = Y[3][2]
8  Y[2][1] = Y[3]
9  Y[1][1] = Y[2][1][1]
10 Y[1][3] = Y[1]
11 Y[2][0] = Y[1][3][3][3][3][3][3][3][2] + 2
12 Y[2][0] **= 2
13 Y[1][3][3][3][3][3][3][3][2] = Y[2]
14 print(Y[1][3][3][3][3][3][3][3][2][0])
15 print("End")
```

Edit this code

line that just executed
next line to execute

Print output (drag lower right corner to resize)

Frames Objects

https://pythontutor.com/visualize.html#mode=edit

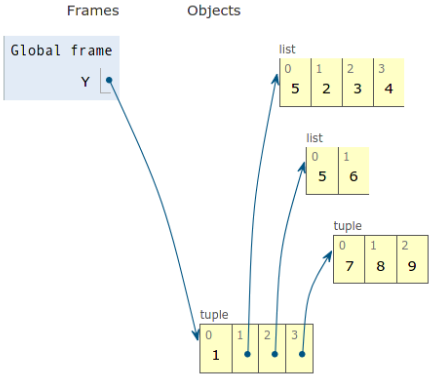
Python 3.6
(known limitations)

```
1  Y =  ( 1,
2         [5, 2, 3, 4],
3         [5, 6],
4         (7, 8, 9)
5       )
6  Y[1][0] = "RADAR"
7  Y[1][2] = Y[3][2]
8  Y[2][1] = Y[3]
9  Y[1][1] = Y[2][1][1]
10 Y[1][3] = Y[1]
11 Y[2][0] = Y[1][3][3][3][3][3][3][3][2] + 2
12 Y[2][0] **= 2
13 Y[1][3][3][3][3][3][3][3][2] = Y[2]
14 print(Y[1][3][3][3][3][3][3][3][2][0])
15 print("End")
```

Edit this code

line that just executed
next line to execute

Print output (drag lower right corner to resize)



https://pythontutor.com/visualize.html#mode=edit

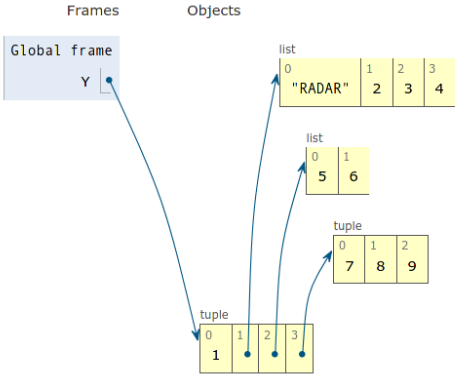
Python 3.6
(known limitations)

```
1  Y =  ( 1,
2         [5, 2, 3, 4],
3         [5, 6],
4         (7, 8, 9)
5       )
6  Y[1][0] = "RADAR"
7  Y[1][2] = Y[3][2]
8  Y[2][1] = Y[3]
9  Y[1][1] = Y[2][1][1]
10 Y[1][3] = Y[1]
11 Y[2][0] = Y[1][3][3][3][3][3][3][3][2] + 2
12 Y[2][0] **= 2
13 Y[1][3][3][3][3][3][3][3][2] = Y[2]
14 print(Y[1][3][3][3][3][3][3][3][2][0])
15 print("End")
```

Edit this code

line that just executed
next line to execute

Print output (drag lower right corner to resize)



https://pythontutor.com/visualize.html#mode=edit

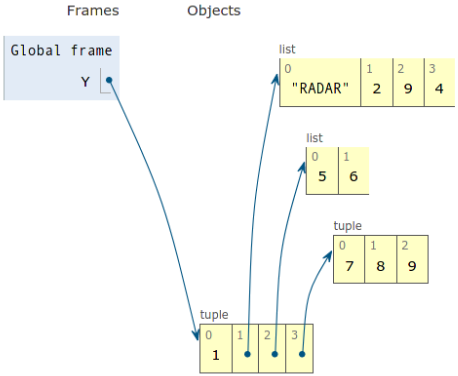
Python 3.6
(known limitations)

```
1  Y =  ( 1,
2         [5, 2, 3, 4],
3         [5, 6],
4         (7, 8, 9)
5       )
6  Y[1][0] = "RADAR"
7  Y[1][2] = Y[3][2]
8  Y[2][1] = Y[3]
9  Y[1][1] = Y[2][1][1]
10 Y[1][3] = Y[1]
11 Y[2][0] = Y[1][3][3][3][3][3][3][3][2] + 2
12 Y[2][0] **= 2
13 Y[1][3][3][3][3][3][3][3][2] = Y[2]
14 print(Y[1][3][3][3][3][3][3][3][2][0])
15 print("End")
```

Edit this code

line that just executed
next line to execute

Print output (drag lower right corner to resize)



https://pythontutor.com/visualize.html#mode=edit

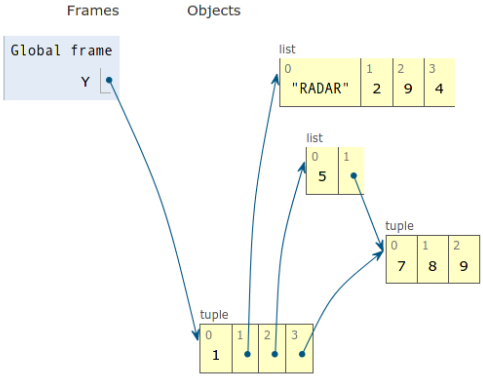
Python 3.6
(known limitations)

```
1  Y =  ( 1,
2         [5, 2, 3, 4],
3         [5, 6],
4         (7, 8, 9)
5       )
6  Y[1][0] = "RADAR"
7  Y[1][2] = Y[3][2]
8  Y[2][1] = Y[3]
9  Y[1][1] = Y[2][1][1]
10 Y[1][3] = Y[1]
11 Y[2][0] = Y[1][3][3][3][3][3][3][3][2] + 2
12 Y[2][0] **= 2
13 Y[1][3][3][3][3][3][3][3][2] = Y[2]
14 print(Y[1][3][3][3][3][3][3][3][2][0])
15 print("End")
```

Edit this code

line that just executed
next line to execute

Print output (drag lower right corner to resize)



https://pythontutor.com/visualize.html#mode=edit

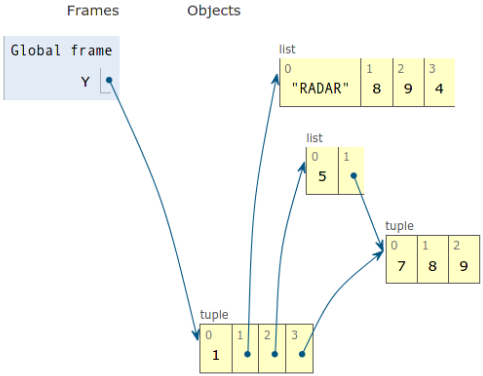
Python 3.6
(known limitations)

```
1  Y =  ( 1,
2         [5, 2, 3, 4],
3         [5, 6],
4         (7, 8, 9)
5       )
6  Y[1][0] = "RADAR"
7  Y[1][2] = Y[3][2]
8  Y[2][1] = Y[3]
9  Y[1][1] = Y[2][1][1]
10 Y[1][3] = Y[1]
11 Y[2][0] = Y[1][3][3][3][3][3][3][3][2] + 2
12 Y[2][0] **= 2
13 Y[1][3][3][3][3][3][3][3][2] = Y[2]
14 print(Y[1][3][3][3][3][3][3][3][2][0])
15 print("End")
```

Edit this code

line that just executed
next line to execute

Print output (drag lower right corner to resize)



https://pythontutor.com/visualize.html#mode=edit

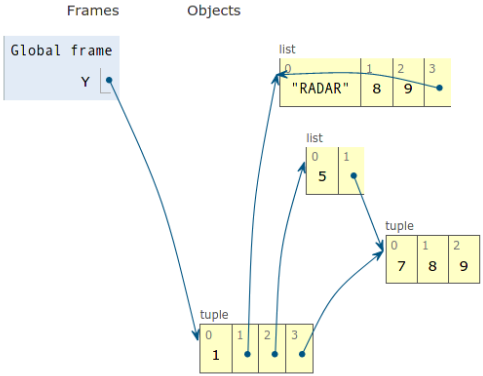
Python 3.6
(known limitations)

```
1  Y =  ( 1,
2         [5, 2, 3, 4],
3         [5, 6],
4         (7, 8, 9)
5       )
6  Y[1][0] = "RADAR"
7  Y[1][2] = Y[3][2]
8  Y[2][1] = Y[3]
9  Y[1][1] = Y[2][1][1]
10 Y[1][3] = Y[1]
11 Y[2][0] = Y[1][3][3][3][3][3][3][3][2] + 2
12 Y[2][0] **= 2
13 Y[1][3][3][3][3][3][3][3][2] = Y[2]
14 print(Y[1][3][3][3][3][3][3][3][2][0])
15 print("End")
```

Edit this code

line that just executed
next line to execute

Print output (drag lower right corner to resize)



https://pythontutor.com/visualize.html#mode=edit

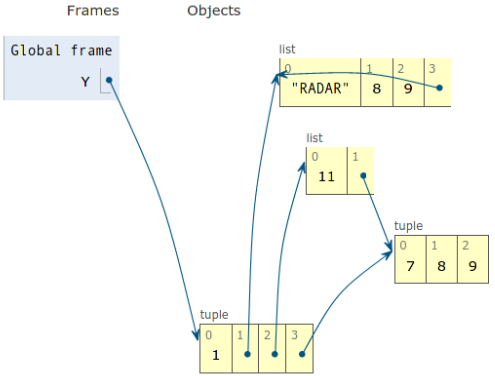
Python 3.6
(known limitations)

```
1  Y =  ( 1,
2         [5, 2, 3, 4],
3         [5, 6],
4         (7, 8, 9)
5       )
6  Y[1][0] = "RADAR"
7  Y[1][2] = Y[3][2]
8  Y[2][1] = Y[3]
9  Y[1][1] = Y[2][1][1]
10 Y[1][3] = Y[1]
11 Y[2][0] = Y[1][3][3][3][3][3][3][3][2] + 2
12 Y[2][0] **= 2
13 Y[1][3][3][3][3][3][3][3][2] = Y[2]
14 print(Y[1][3][3][3][3][3][3][3][2][0])
15 print("End")
```

Edit this code

line that just executed
next line to execute

Print output (drag lower right corner to resize)



https://pythontutor.com/visualize.html#mode=edit

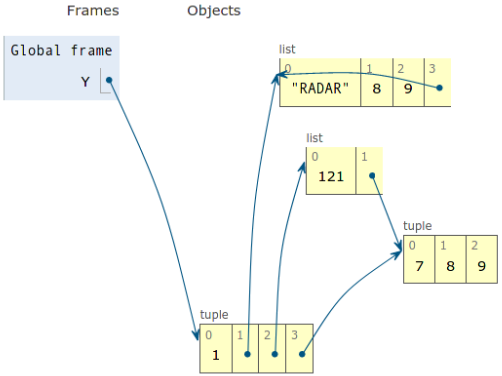
Python 3.6
(known limitations)

```
1  Y =  ( 1,
2         [5, 2, 3, 4],
3         [5, 6],
4         (7, 8, 9)
5       )
6  Y[1][0] = "RADAR"
7  Y[1][2] = Y[3][2]
8  Y[2][1] = Y[3]
9  Y[1][1] = Y[2][1][1]
10 Y[1][3] = Y[1]
11 Y[2][0] = Y[1][3][3][3][3][3][3][3][2] + 2
12 Y[2][0] **= 2
13 Y[1][3][3][3][3][3][3][3][2] = Y[2]
14 print(Y[1][3][3][3][3][3][3][3][2][0])
15 print("End")
```

Edit this code

line that just executed
next line to execute

Print output (drag lower right corner to resize)



https://pythontutor.com/visualize.html#mode=edit

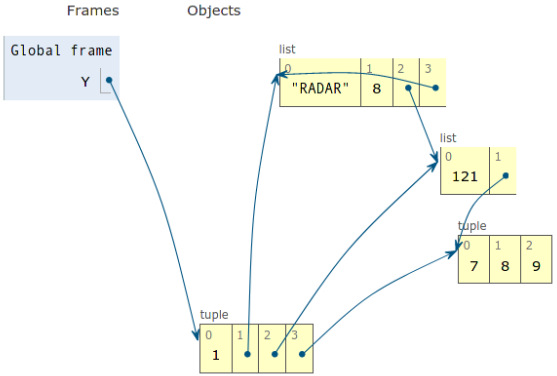
Python 3.6
(known limitations)

```
1  Y =  ( 1,
2         [5, 2, 3, 4],
3         [5, 6],
4         (7, 8, 9)
5       )
6  Y[1][0] = "RADAR"
7  Y[1][2] = Y[3][2]
8  Y[2][1] = Y[3]
9  Y[1][1] = Y[2][1][1]
10 Y[1][3] = Y[1]
11 Y[2][0] = Y[1][3][3][3][3][3][3][3][2] + 2
12 Y[2][0] **= 2
13 Y[1][3][3][3][3][3][3][3][2] = Y[2]
14 print(Y[1][3][3][3][3][3][3][3][2][0])
15 print("End")
```

Edit this code

line that just executed
next line to execute

Print output (drag lower right corner to resize)



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

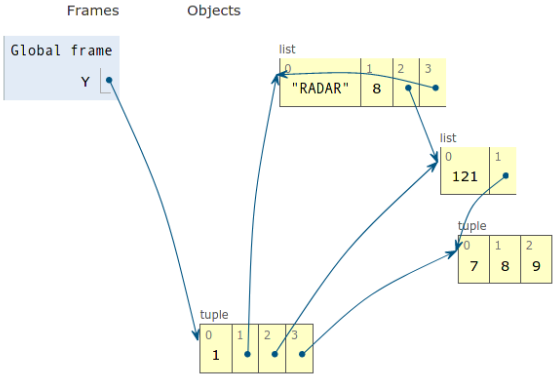
```
1  Y =  ( 1,
2         [5, 2, 3, 4],
3         [5, 6],
4         (7, 8, 9)
5       )
6  Y[1][0] = "RADAR"
7  Y[1][2] = Y[3][2]
8  Y[2][1] = Y[3]
9  Y[1][1] = Y[2][1][1]
10 Y[1][3] = Y[1]
11 Y[2][0] = Y[1][3][3][3][3][3][3][3][2] + 2
12 Y[2][0] **= 2
13 Y[1][3][3][3][3][3][3][3][2] = Y[2]
14 print(Y[1][3][3][3][3][3][3][3][2][0])
15 print("End")
```

Edit this code

line that just executed
next line to execute

Print output (drag lower right corner to resize)

```
121
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

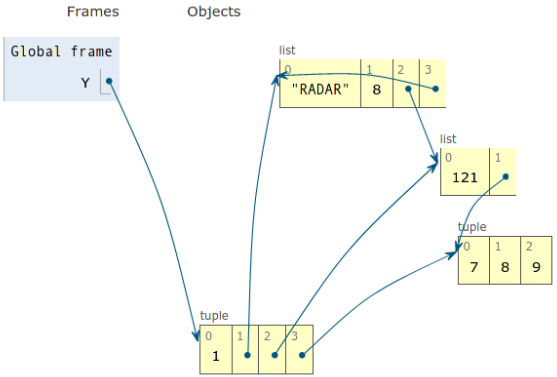
```
1  Y =  ( 1,
2         [5, 2, 3, 4],
3         [5, 6],
4         (7, 8, 9)
5       )
6  Y[1][0] = "RADAR"
7  Y[1][2] = Y[3][2]
8  Y[2][1] = Y[3]
9  Y[1][1] = Y[2][1][1]
10 Y[1][3] = Y[1]
11 Y[2][0] = Y[1][3][3][3][3][3][3][3][2] + 2
12 Y[2][0] **= 2
13 Y[1][3][3][3][3][3][3][3][2] = Y[2]
14 print(Y[1][3][3][3][3][3][3][3][2][0])
15 print("End")
```

Edit this code

line that just executed
next line to execute

Print output (drag lower right corner to resize)

```
121
End
```



Exemplo 3: Sets Multidimensionais

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 1 of 26

Print output (drag lower right corner to resize)

Frames Objects

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 2 of 26

Print output (drag lower right corner to resize)

Frames Objects

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 3 of 26

Print output (drag lower right corner to resize)

Frames Objects

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 4 of 26

Print output (drag lower right corner to resize)

Frames Objects

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 5 of 26

Print output (drag lower right corner to resize)

Frames Objects

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 6 of 26

Print output (drag lower right corner to resize)

Frames Objects

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8      }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 7 of 26

Print output (drag lower right corner to resize)

Frames Objects

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

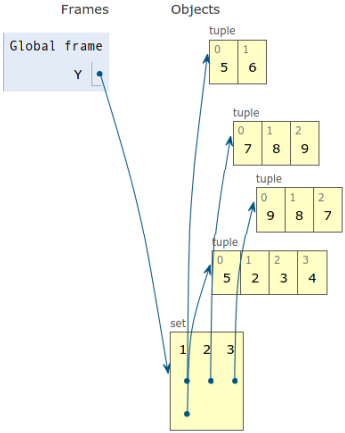
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 8 of 26

Print output (drag lower right corner to resize)



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

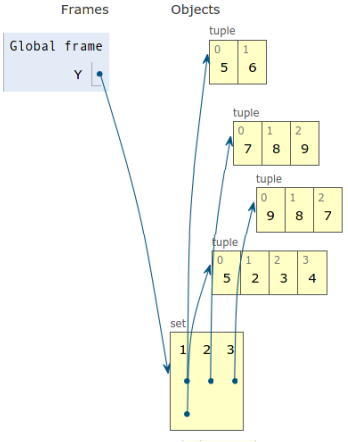
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 9 of 26

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

line that just executed
next line to execute

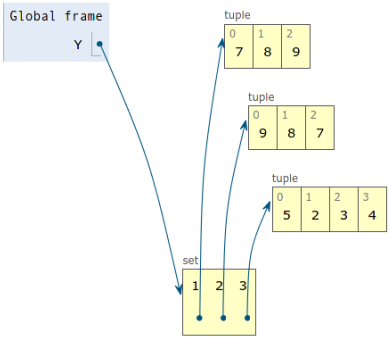
<< First | < Prev | Next > | Last >>

Step 10 of 26

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
```

Frames Objects



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

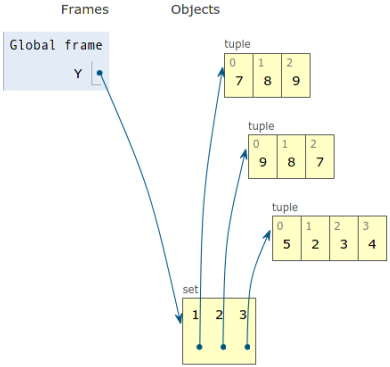
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 11 of 26

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
{1, 2, 3, (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

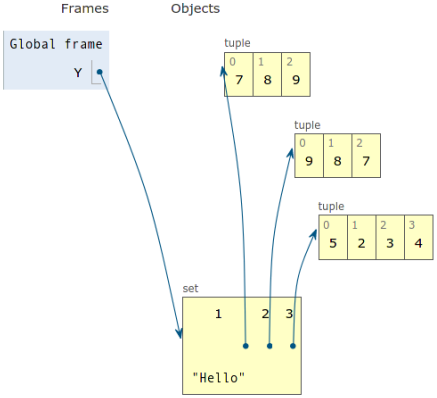
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 12 of 26

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
{1, 2, 3, (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

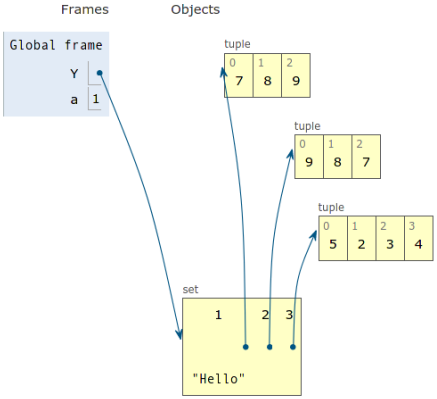
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 13 of 26

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
{1, 2, 3, (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

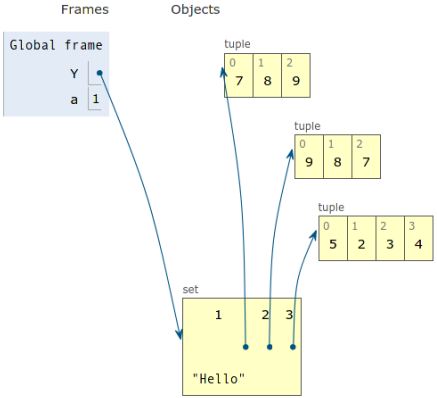
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 14 of 26

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
{1, 2, 3, (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
1
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

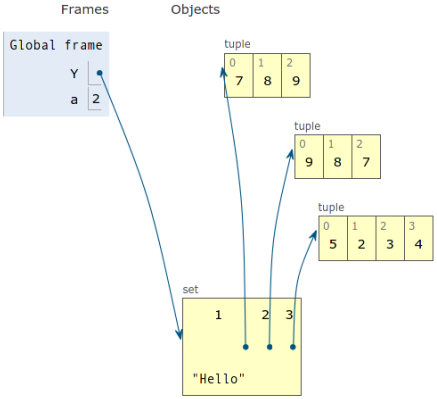
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 15 of 26

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
{1, 2, 3, (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
1
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

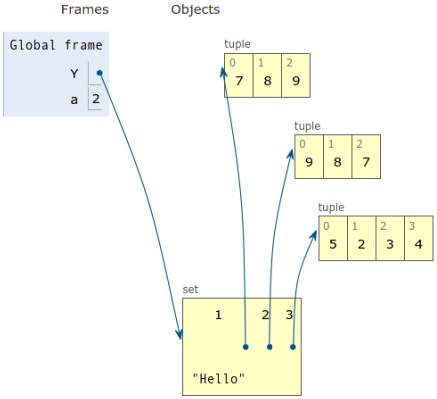
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 16 of 26

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
{1, 2, 3, (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
1  2
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

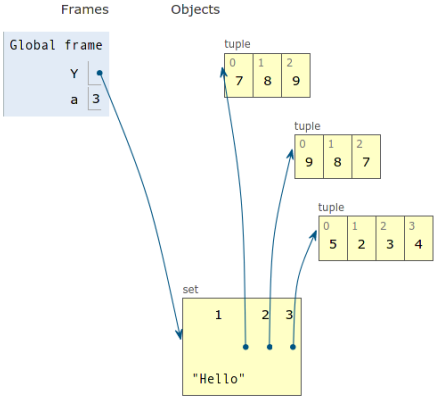
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 17 of 26

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
{1, 2, 3, (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
1  2
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

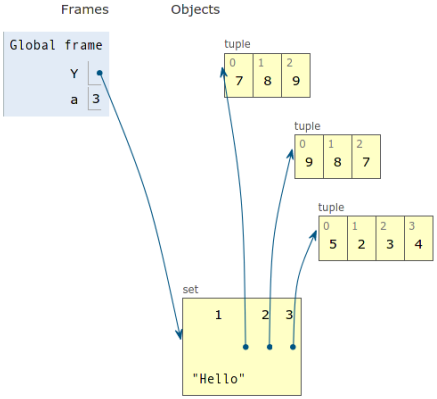
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 18 of 26

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
{1, 2, 3, (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
1  2  3
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

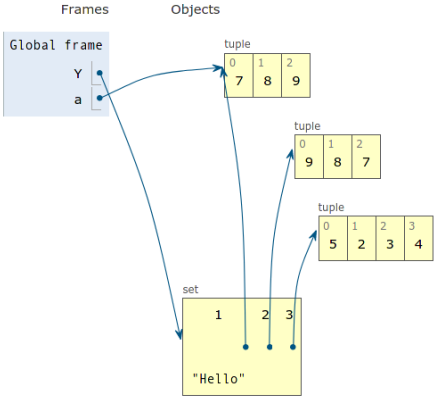
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 19 of 26

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
{1, 2, 3, (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
1  2  3
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

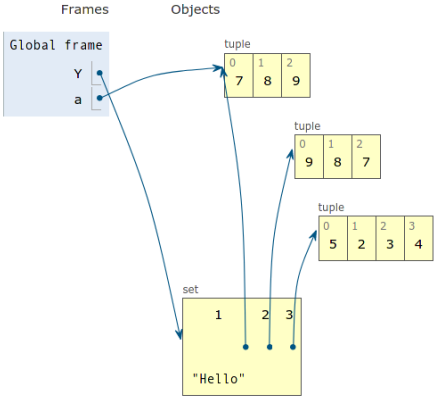
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 20 of 26

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
{1, 2, 3, (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
1  2  3  (7, 8, 9)
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

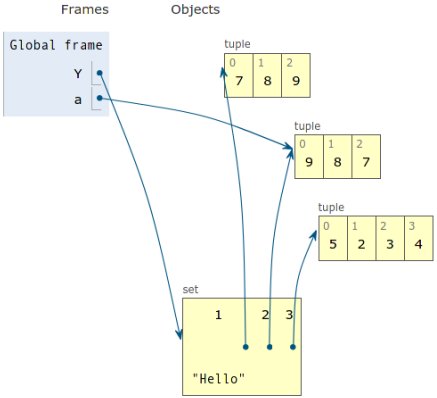
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 21 of 26

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
{1, 2, 3, (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
1  2  3  (7, 8, 9)
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

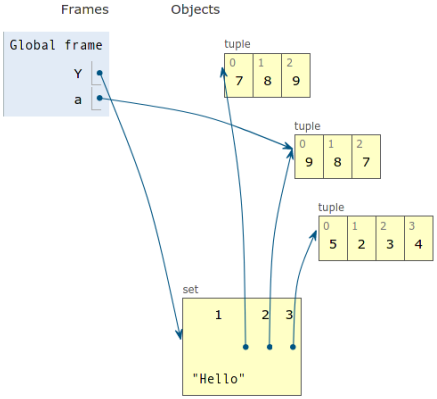
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 22 of 26

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
{1, 2, 3, (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
1  2  3  (7, 8, 9)  (9, 8, 7)
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

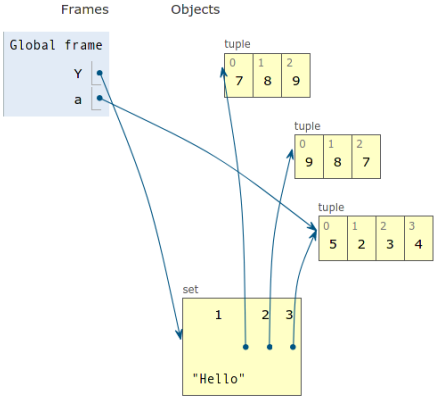
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 23 of 26

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
{1, 2, 3, (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
1  2  3  (7, 8, 9)  (9, 8, 7)
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

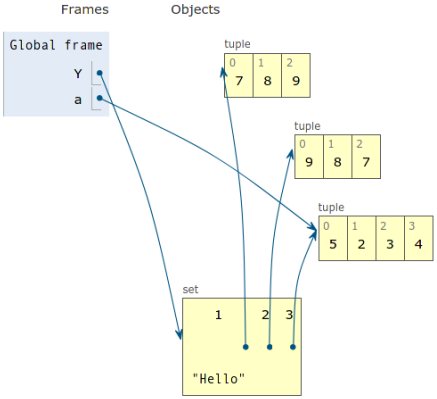
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 24 of 26

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
{1, 2, 3, (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
1  2  3  (7, 8, 9)  (9, 8, 7)  (5, 2, 3, 4)
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

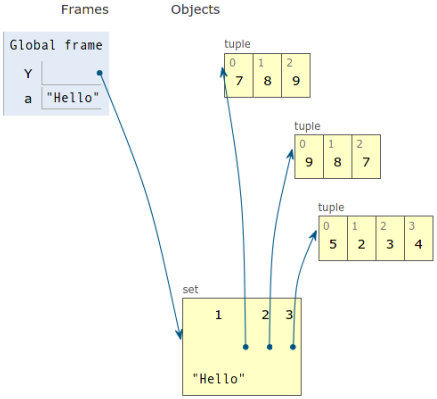
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 25 of 26

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
{1, 2, 3, (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
1  2  3  (7, 8, 9)  (9, 8, 7)  (5, 2, 3, 4)
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

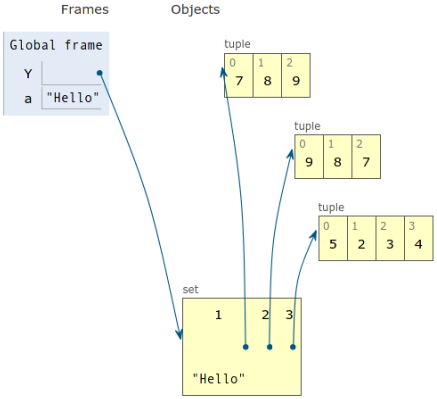
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 26 of 26

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
{1, 2, 3, (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
1  2  3  (7, 8, 9)  (9, 8, 7)  (5, 2, 3, 4)  Hello
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  Y =  { 1, 2, 3,
2         (5, 2, 3, 4),
3         (5, 6),
4         (7, 8, 9), (9, 8, 7),
5         3,
6         (5, 6),
7         2,
8       }
9
10 print(Y)
11 Y.discard((5,6))
12 print(Y)
13 Y.add("Hello")
14 for a in Y:
15     print(a,end='  ')
```

Edit this code

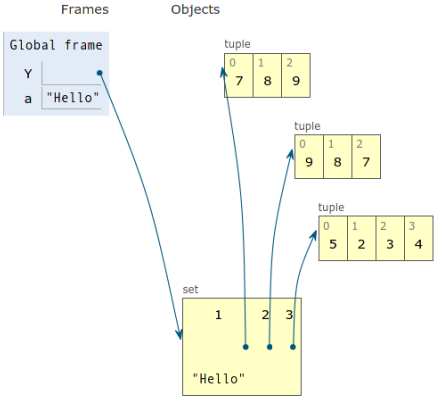
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Done running (26 steps)

Print output (drag lower right corner to resize)

```
{1, 2, 3, (5, 6), (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
{1, 2, 3, (7, 8, 9), (9, 8, 7), (5, 2, 3, 4)}
1  2  3  (7, 8, 9)  (9, 8, 7)  (5, 2, 3, 4)  Hello
```



Exemplo 4: Dicionários Multidimensionais

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1 X = {   0 : 1,
2         1 : [5, 2, 3, 4],
3         2 : [5, 6],
4         3 : [7, 8, 9]
5       }
6 X[0] = (1,2)
7 X[1][2] = "RADAR"
8 X[3][2] = ["CAT",["D","O","G"]]
9 X[3][2][1][2] = "T"
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 1 of 8

Frames Objects

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1 X = {   0 : 1,
2         1 : [5, 2, 3, 4],
3         2 : [5, 6],
4         3 : [7, 8, 9]
5       }
6 X[0] = (1,2)
7 X[1][2] = "RADAR"
8 X[3][2] = ["CAT",["D","O","G"]]
9 X[3][2][1][2] = "T"
```

Edit this code

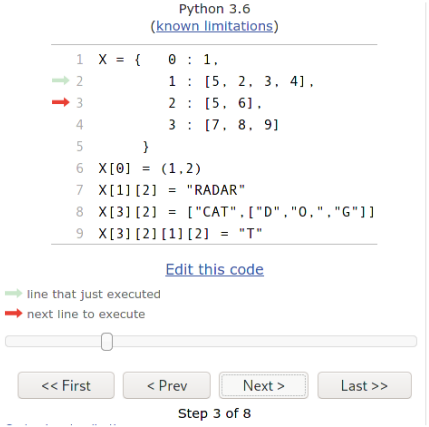
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 2 of 8

Frames Objects

https://pythontutor.com/visualize.html#mode=edit



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  X = {   0 : 1,
2          1 : [5, 2, 3, 4],
3          2 : [5, 6],
4          3 : [7, 8, 9]
5       }
6  X[0] = (1,2)
7  X[1][2] = "RADAR"
8  X[3][2] = ["CAT",["D","O","G"]]
9  X[3][2][1][2] = "T"
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 4 of 8

Frames Objects

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

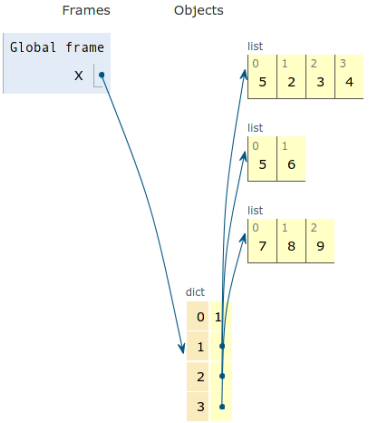
```
1  X = {   0 : 1,
2          1 : [5, 2, 3, 4],
3          2 : [5, 6],
4          3 : [7, 8, 9]
5      }
6  X[0] = (1,2)
7  X[1][2] = "RADAR"
8  X[3][2] = ["CAT",["D","O","G"]]
9  X[3][2][1][2] = "T"
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 5 of 8



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

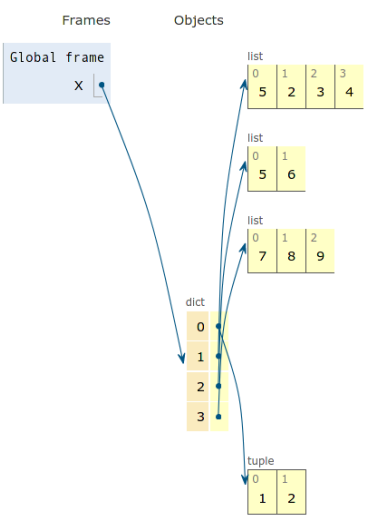
```
1  X = {   0 : 1,
2          1 : [5, 2, 3, 4],
3          2 : [5, 6],
4          3 : [7, 8, 9]
5       }
6  X[0] = (1,2)
7  X[1][2] = "RADAR"
8  X[3][2] = ["CAT",["D","O","G"]]
9  X[3][2][1][2] = "T"
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 6 of 8



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

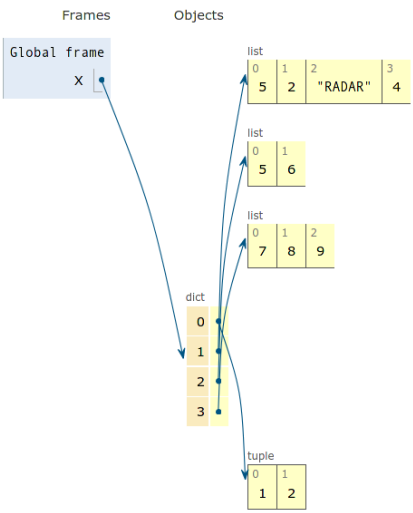
```
1  X = {   0 : 1,
2          1 : [5, 2, 3, 4],
3          2 : [5, 6],
4          3 : [7, 8, 9]
5       }
6  X[0] = (1,2)
7  X[1][2] = "RADAR"
8  X[3][2] = ["CAT",["D","O","G"]]
9  X[3][2][1][2] = "T"
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 7 of 8



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

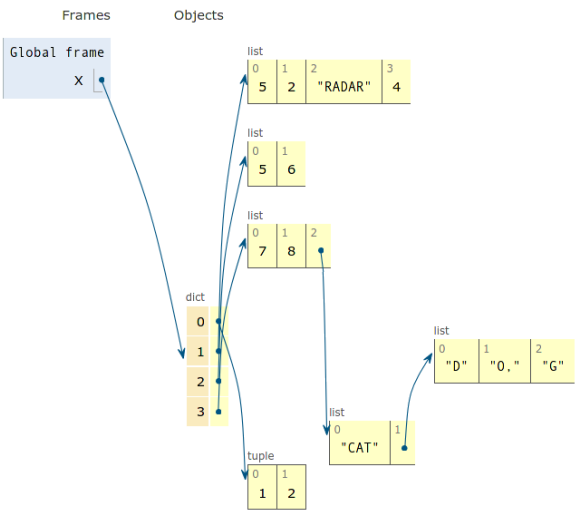
```
1  X = {   0 : 1,
2          1 : [5, 2, 3, 4],
3          2 : [5, 6],
4          3 : [7, 8, 9]
5       }
6  X[0] = (1,2)
7  X[1][2] = "RADAR"
8  X[3][2] = ["CAT",["D","O,","G"]]
9  X[3][2][1][2] = "T"
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 8 of 8



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

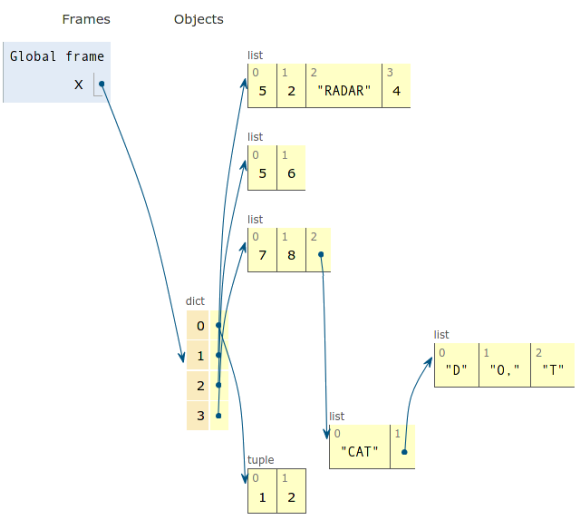
```
1  X = {   0 : 1,
2          1 : [5, 2, 3, 4],
3          2 : [5, 6],
4          3 : [7, 8, 9]
5       }
6  X[0] = (1,2)
7  X[1][2] = "RADAR"
8  X[3][2] = ["CAT",["D","O,","G"]]
9  X[3][2][1][2] = "T"
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Done running (8 steps)



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

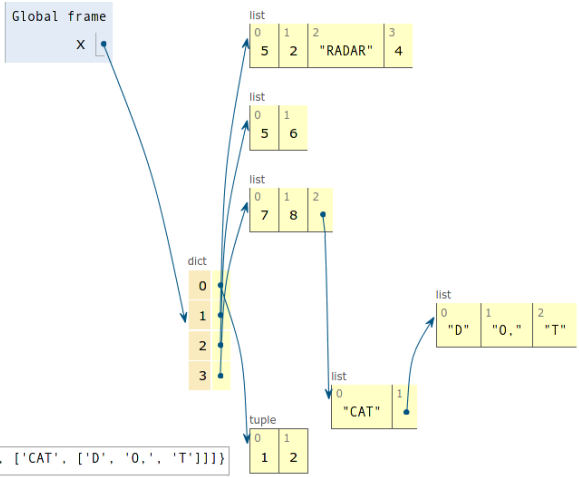
```
1  X = {   0 : 1,
2          1 : [5, 2, 3, 4],
3          2 : [5, 6],
4          3 : [7, 8, 9]
5       }
6  X[0] = (1,2)
7  X[1][2] = "RADAR"
8  X[3][2] = ["CAT",["D","O,","G"]]
9  X[3][2][1][2] = "T"
10
11 print(X)
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Frames Objects



Print output (drag lower right corner to resize)

```
{0: (1, 2), 1: [5, 2, 'RADAR', 4], 2: [5, 6], 3: [7, 8, ['CAT', ['D', 'O,', 'T']]]}
```

Matrices

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

- Uma matriz é um objeto bidimensional, formada por listas, todas do mesmo tamanho.
- Sua representação é dada na forma de uma lista de listas (a mesma ideia pode ser aplicada para tuplas).
- Exemplo de declaração de uma matriz 2×2:

```
matriz = [
  [1, 2], # linha 1
  [3, 4]  # linha 2
]
```

- Exemplo de declaração de uma matriz 3×4:

```
matriz = [
  [11, 12, 13, 14], # linha 1
  [21, 22, 23, 24], # linha 2
  [31, 32, 33, 34]  # linha 3
]
```

https://pythontutor.com/visualize.html#mode=edit

(known limitations)

```
1  M1 = [
2          [1, 2], # linha 1
3          [3, 4], # linha 2
4        ]
5
6  M2 = [
7          [11, 12, 13, 14], # linha 1
8          [21, 22, 23, 24], # linha 2
9          [31, 32, 33, 34], # linha 3
10       ]
```

Edit this code

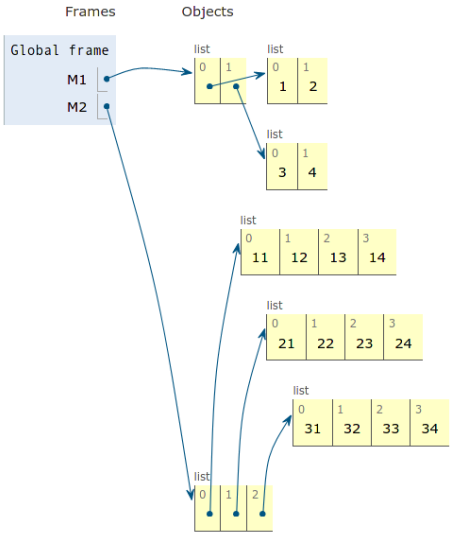
line that just executed

next line to execute

<< First | < Prev | Next > | Last >>

Done running (5 steps)

Customize visualization



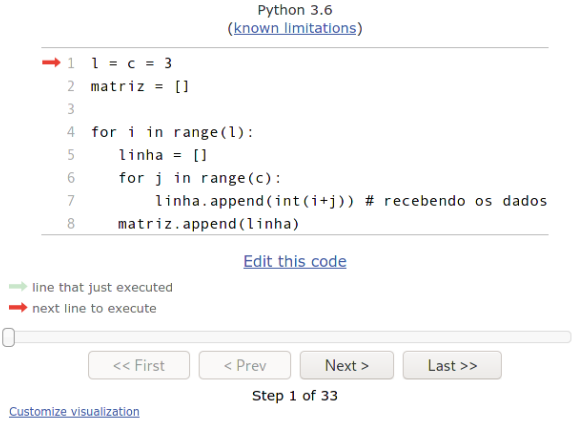
https://ic.unicamp.br/~mc102/aulas/aula09.pdf

- Podemos criar uma matriz com as informações fornecidas pelo usuário.
- Exemplo de como receber uma matriz de dimensões $l \times c$ como entrada:

```
l = int(input("Entre com o número de linhas: "))
c = int(input("Entre com o número de colunas: "))
matriz = []

for i in range(l):
  linha = []
  for j in range(c):
    linha.append(int(input())) # recebendo os dados
  matriz.append(linha)
```

https://pythontutor.com/visualize.html#mode=edit



https://pythontutor.com/visualize.html#mode=edit

https://pythontutor.com/visualize.html#mode=edit

https://pythontutor.com/visualize.html#mode=edit

https://pythontutor.com/visualize.html#mode=edit

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

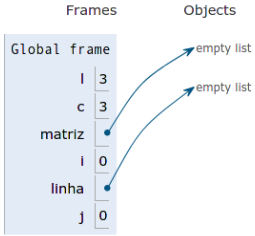
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 6 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

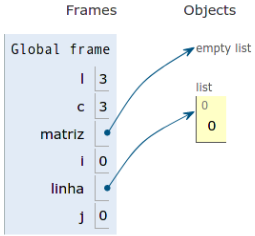
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 7 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

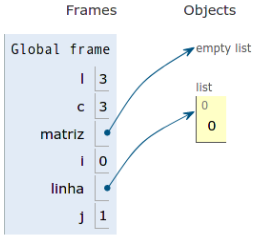
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 8 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

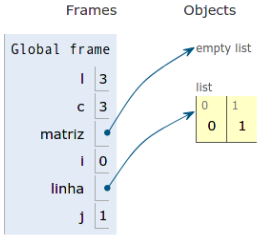
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 9 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

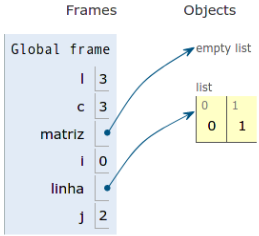
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 10 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

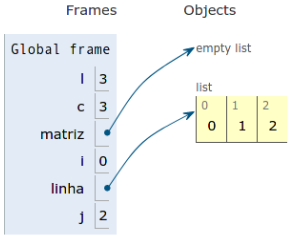
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 11 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

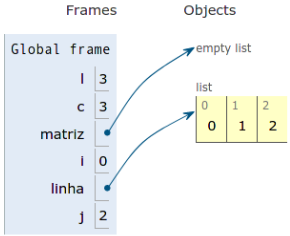
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 12 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

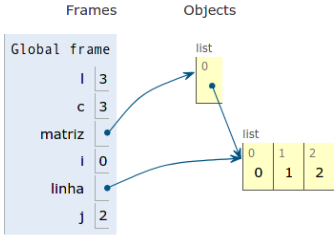
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 13 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5     linha = []
6     for j in range(c):
7         linha.append(int(i+j)) # recebendo os dados
8     matriz.append(linha)
```

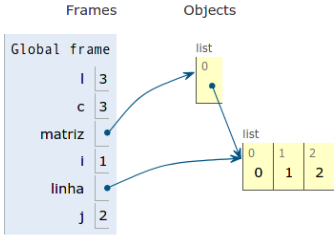
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 14 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

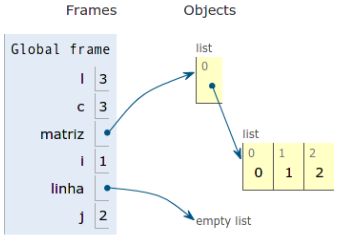
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 15 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

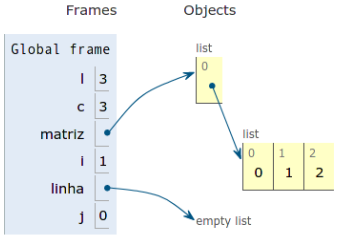
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 16 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

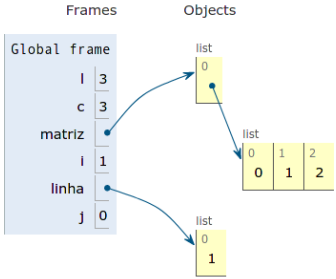
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 17 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

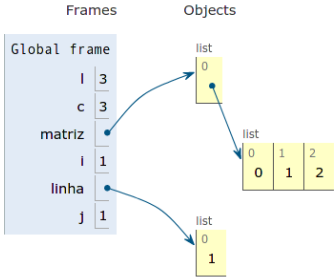
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 18 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

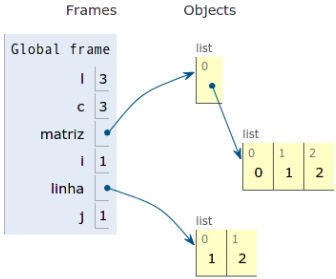
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 19 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

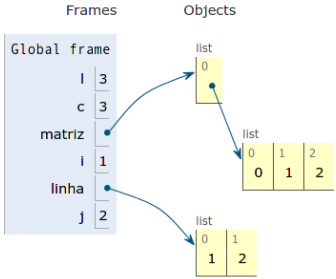
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 20 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

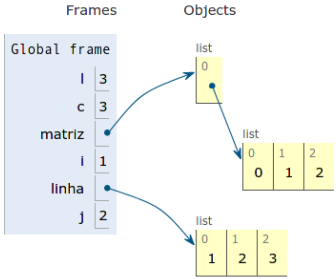
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 21 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

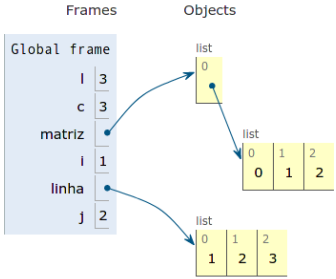
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 22 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

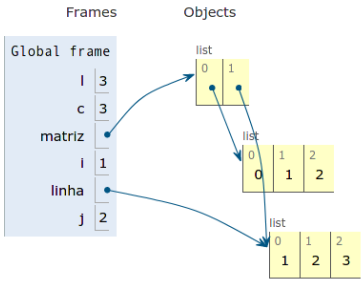
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 23 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

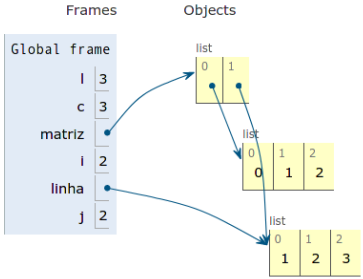
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 24 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

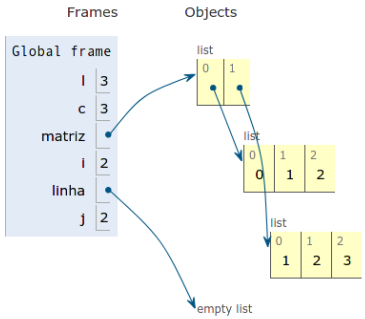
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 25 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

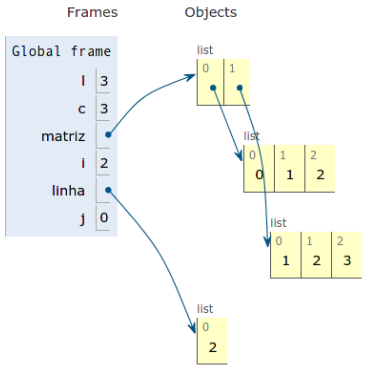
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 27 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

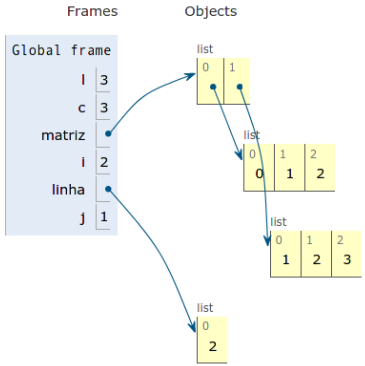
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 28 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

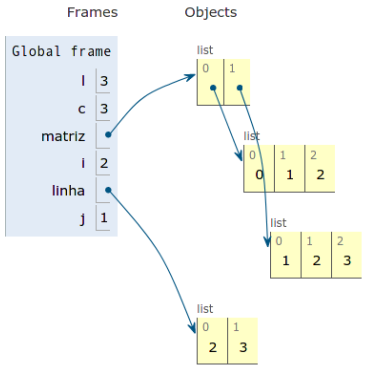
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 29 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

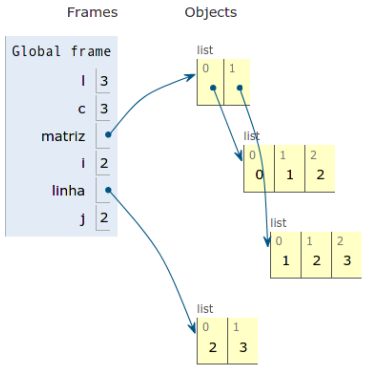
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 30 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

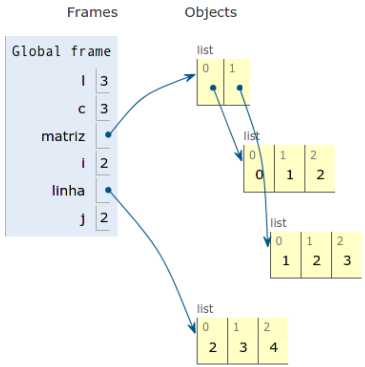
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 31 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

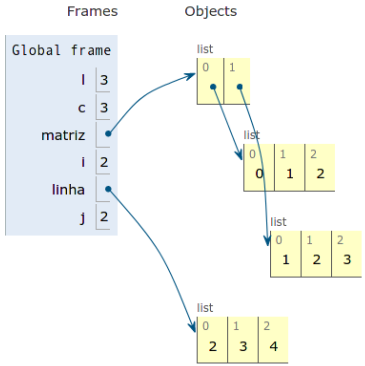
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 32 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

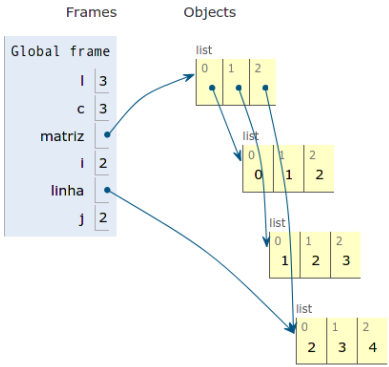
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 33 of 33

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  for i in range(l):
5      linha = []
6      for j in range(c):
7          linha.append(int(i+j)) # recebendo os dados
8      matriz.append(linha)
```

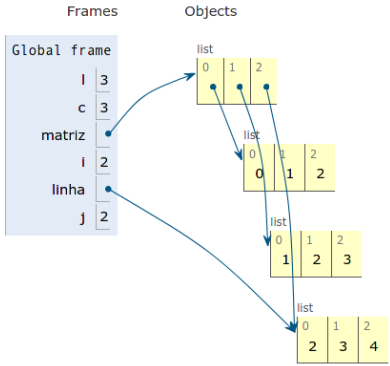
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Done running (33 steps)

Customize visualization



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  l = c = 3
2  matriz = []
3
4  matriz = [[i+j for j in range(c)] for i in range(l)]
```

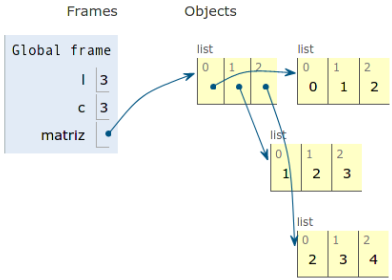
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Done running (27 steps)

Customize visualization



unsupported features

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

- Podemos ainda inicializar uma matriz com valores pré-definidos.
- Inicializando uma matriz de dimensões $l \times c$ e atribuindo valor zero para todos os elementos:

```
l = int(input("Entre com o número de linhas: "))  # l = 3
c = int(input("Entre com o número de colunas: ")) # c = 4
matriz = []
for i in range(l):
  linha = []
  for j in range(c):
    linha.append(0)
  matriz.append(linha)
print(matriz)
# [[0, 0, 0, 0], [0, 0, 0, 0], [0, 0, 0, 0]]

# Forma alternativa/compacta de inicializar uma matriz
matriz = [[0 for j in range(c)] for i in range(l)]
```

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

- Inicializando uma matriz de dimensões $l \times c$ e atribuindo valores de 1 até $l \times c$ para os elementos da matriz:

```
l = int(input("Entre com o número de linhas: "))  # l = 3
c = int(input("Entre com o número de colunas: ")) # c = 4
matriz = []

for i in range(l):
  linha = []
  for j in range(c):
    linha.append(i * c + j + 1)
  matriz.append(linha)

print(matriz)
# [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12]]
```

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

- Note que uma matriz é que uma lista de listas.
- Podemos acessar um elemento de uma matriz, localizado em uma determinada linha e coluna, da seguinte forma:

```
matriz[linha][coluna]
# Lembrete: linhas e colunas são numeradas
#           a partir da posição zero
```

- Exemplo:

```
matriz = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
print(matriz[0][2])
# 3
print(matriz[2][1])
# 8
```

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

- Similar ao que vimos em listas e tuplas, caso ocorra uma tentativa de acessar uma posição inexistente da matriz, um erro será gerado.
- Exemplo:

```
1 matriz = [[1, 2], [3, 4]]
2 print(matriz[0][0])
3 # 1
4 print(matriz[1][1])
5 # 4
6 print(matriz[2][2])
7 # IndexError: list index out of range
```

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

- Podemos alterar um elemento de uma matriz, localizado em uma determinada linha e coluna, da seguinte forma:

```
matriz[linha][coluna] = valor
```

- Exemplo:

```
matriz = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
matriz[0][0] = 0
matriz[2][2] = 10
print(matriz)
# [[0, 2, 3], [4, 5, 6], [7, 8, 10]]
```

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

- Para criar uma cópia de uma matriz, precisamos criar uma nova matriz com as cópias de cada uma das linhas da matriz original.
- Exemplo:

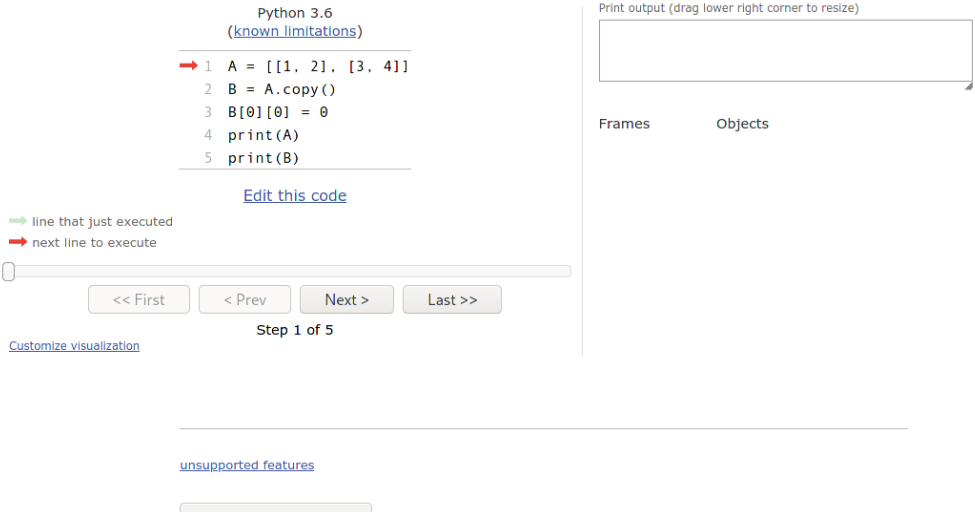
```
A = [[1, 2], [3, 4]]

B = A.copy()

B[0][0] = 0

print(A)
# [[0, 2], [3, 4]]
print(B)
# [[0, 2], [3, 4]]
```

https://pythontutor.com/visualize.html#mode=edit



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1 A = [[1, 2], [3, 4]]
2 B = A.copy()
3 B[0][0] = 0
4 print(A)
5 print(B)
```

Edit this code

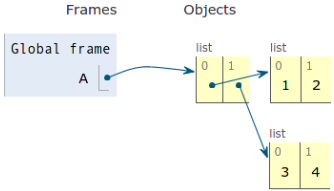
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 2 of 5

Customize visualization

Print output (drag lower right corner to resize)



unsupported features

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  A = [[1, 2], [3, 4]]
2  B = A.copy()
3  B[0][0] = 0
4  print(A)
5  print(B)
```

Edit this code

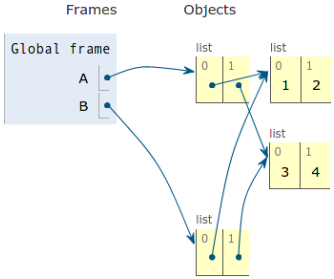
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 3 of 5

Customize visualization

Print output (drag lower right corner to resize)



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  A = [[1, 2], [3, 4]]
2  B = A.copy()
3  B[0][0] = 0
4  print(A)
5  print(B)
```

Edit this code

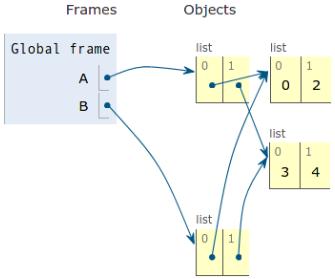
line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 4 of 5

Customize visualization

Print output (drag lower right corner to resize)



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  A = [[1, 2], [3, 4]]
2  B = A.copy()
3  B[0][0] = 0
4  print(A)
5  print(B)
```

Edit this code

line that just executed
next line to execute

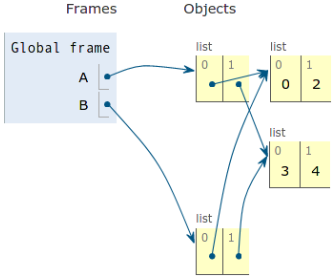
<< First | < Prev | Next > | Last >>

Step 5 of 5

Customize visualization

Print output (drag lower right corner to resize)

```
[[0, 2], [3, 4]]
```



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  A = [[1, 2], [3, 4]]
2  B = A.copy()
3  B[0][0] = 0
4  print(A)
5  print(B)
```

Edit this code

line that just executed
next line to execute

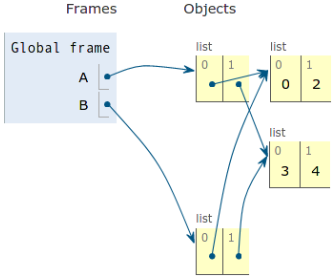
<< First | < Prev | Next > | Last >>

Done running (5 steps)

Customize visualization

Print output (drag lower right corner to resize)

```
[[0, 2], [3, 4]]
[[0, 2], [3, 4]]
```



https://ic.unicamp.br/~mc102/aulas/aula09.pdf

- Para criar uma cópia de uma matriz, precisamos criar uma nova matriz com as cópias de cada uma das linhas da matriz original.
- Exemplo:

```
1  A = [[1, 2], [3, 4]]
2
3  B = [linha.copy() for linha in A]
4
5  B[0][0] = 0
6
7  print(A)
8  # [[1, 2], [3, 4]]
9  print(B)
10 # [[0, 2], [3, 4]]
```

https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  A = [[1, 2], [3, 4]]
2
3  B = [linha.copy() for linha in A]
4
5  B[0][0] = 0
6  print(A)
7  print(B)
```

Edit this code

line that just executed
next line to execute

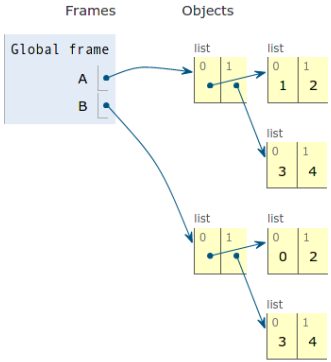
<< First | < Prev | Next > | Last >>

Done running (10 steps)

Customize visualization

Print output (drag lower right corner to resize)

```
[[1, 2], [3, 4]]
[[0, 2], [3, 4]]
```



https://ic.unicamp.br/~mc102/aulas/aula09.pdf

- Para criar uma cópia de uma matriz, precisamos criar uma nova matriz com as cópias de cada uma das linhas da matriz original.
- Exemplo:

```
1  A = [[1, 2], [3, 4]]
2
3  B = [linha[:] for linha in A]
4
5  B[0][0] = 0
6
7  print(A)
8  # [[1, 2], [3, 4]]
9  print(B)
10 # [[0, 2], [3, 4]]
```

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

- Para criar uma cópia de uma matriz, precisamos criar uma nova matriz com as cópias de cada uma das linhas da matriz original.
- Exemplo:

```
1  A = [[1, 2], [3, 4]]
2
3  B = [list(linha) for linha in A]
4
5  B[0][0] = 0
6
7  print(A)
8  # [[1, 2], [3, 4]]
9  print(B)
10 # [[0, 2], [3, 4]]
```

Hipermatrices, arranjos multidimensionais

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

- Até agora criamos matrizes bidimensionais, mas podemos criar objetos com mais dimensões.
- Podemos criar objetos com d dimensões utilizando a mesma ideia de listas de listas.
- Exemplo de um objeto com dimensões $2 \times 2 \times 2$:

```
obj = [
  [[1, 2], [3, 4]],
  [[5, 6], [7, 8]]
]
```

- Podemos acessar um elemento em um objeto com dimensões $d_1 \times d_2 \times \cdots \times d_n$ da seguinte forma:

```
objeto[index_1][index_2]...[index_n]
```

- Exemplo:

```
obj = [[[1, 2], [3, 4]], [[5, 6], [7, 8]]] # 2 x 2 x 2
print(obj[0][0][0])
# 1
print(obj[1][0][0])
# 5
print(obj[1][1][0])
# 7
print(obj[1][1][1])
# 8
```

- Podemos alterar um elemento em um objeto com dimensões $d_1 \times d_2 \times \cdots \times d_n$ da seguinte forma:

```
objeto[index_1][index_2]...[index_n] = valor
```

- Exemplo:

```
obj = [[[0, 0], [0, 0]], [[0, 0], [0, 0]]] # 2 x 2 x 2
obj[1][0][1] = 5
obj[0][1][0] = 3
print(obj)
# [[[0, 0], [3, 0]], [[0, 5], [0, 0]]]
```

Exercícios

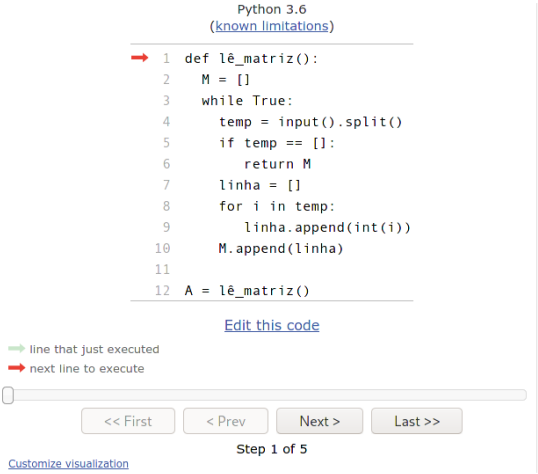
https://ic.unicamp.br/~mc102/aulas/aula09.pdf

1. Escreva uma função que leia e retorne uma matriz de inteiros fornecida pelo usuário. Sua matriz deve ler os números linha a linha. Os números devem estar separados por espaços em branco. Sua função deve interromper a leitura ao receber uma linha em branco.
2. Escreva uma função que, dada uma lista bidimensional (lista de listas), verifique se ela é uma matriz. Em caso positivo, sua função deve retornar uma tupla com o número de linhas e de colunas da matriz. Em caso negativo, deve retornar uma tupla vazia.
3. Escreva uma função que imprime, linha a linha, os valores de uma matriz bidimensional dada como argumento.

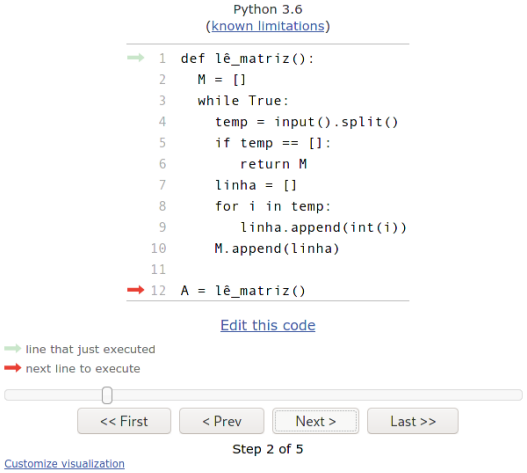
https://ic.unicamp.br/~mc102/aulas/aula09.pdf

```
def lê_matriz():
  M = []
  while True:
    temp = input().split()
    if temp == []:
      return M
    linha = []
    for i in temp:
      linha.append(int(i))
    M.append(linha)
```

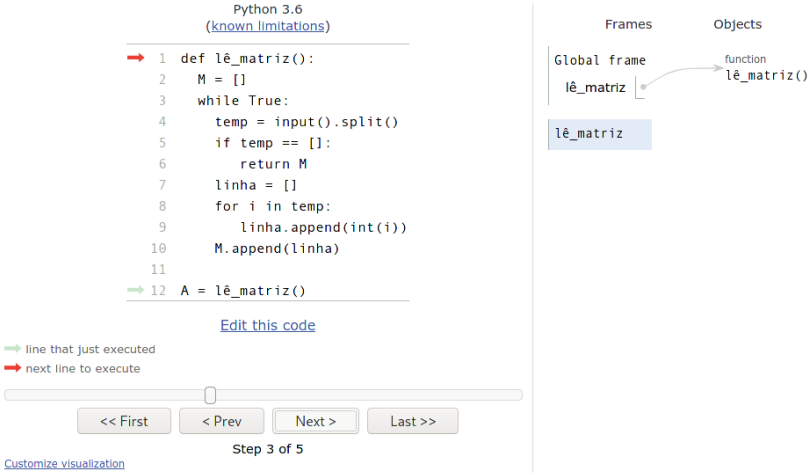
https://pythontutor.com/visualize.html#mode=edit



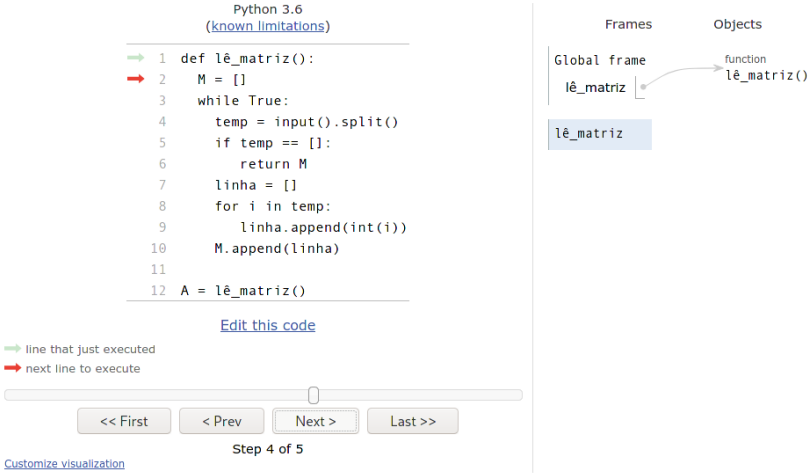
https://pythontutor.com/visualize.html#mode=edit



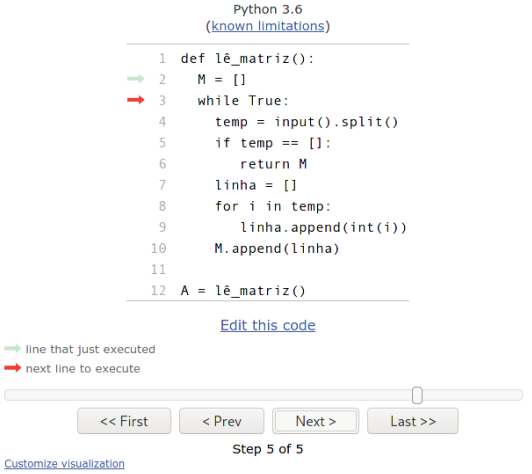
https://pythontutor.com/visualize.html#mode=edit

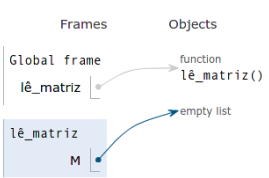


https://pythontutor.com/visualize.html#mode=edit

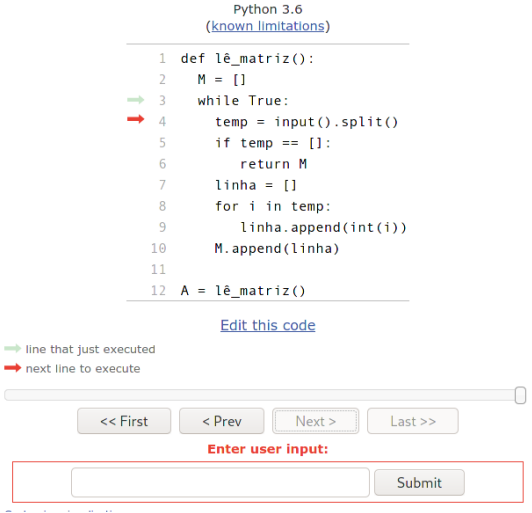


https://pythontutor.com/visualize.html#mode=edit

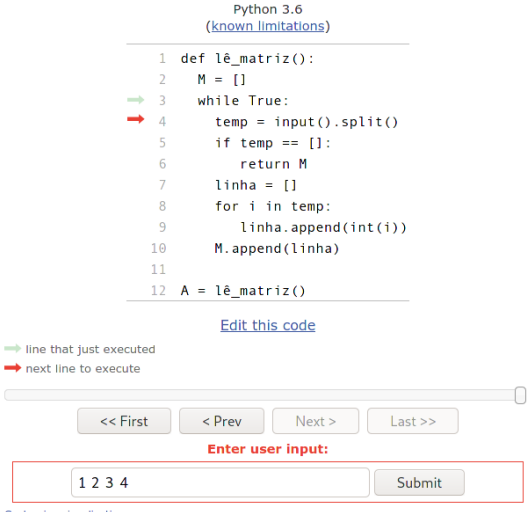


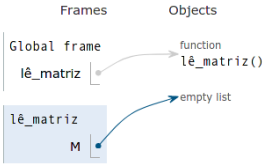


https://pythontutor.com/visualize.html#mode=edit

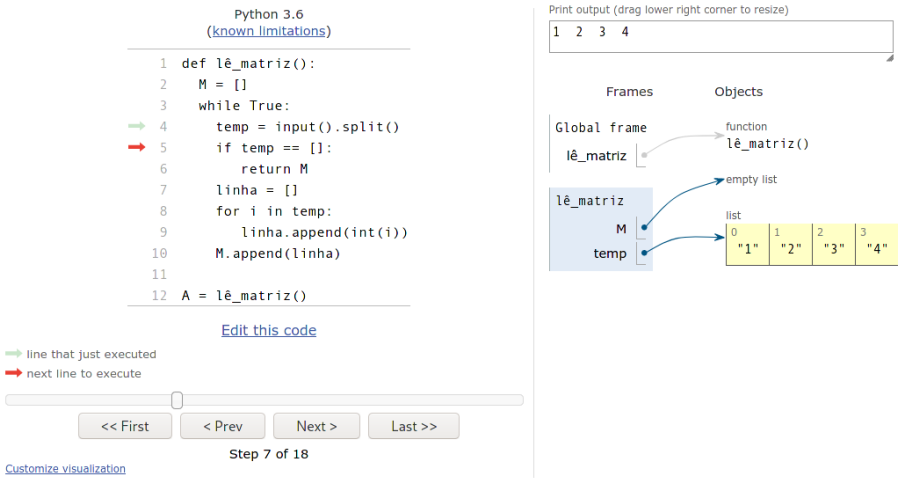


https://pythontutor.com/visualize.html#mode=edit

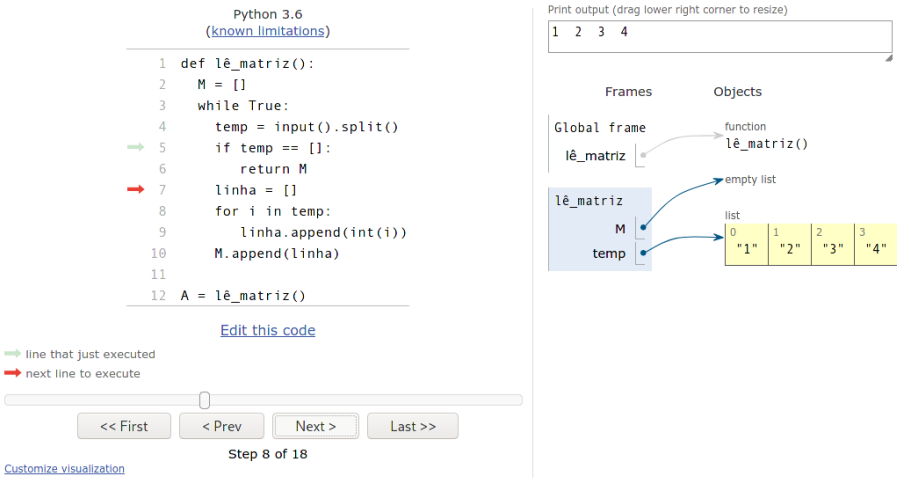




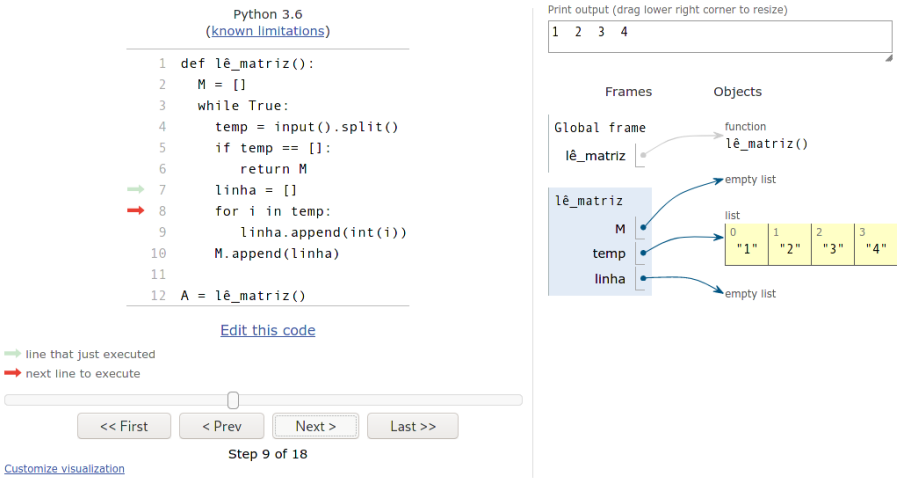
https://pythontutor.com/visualize.html#mode=edit



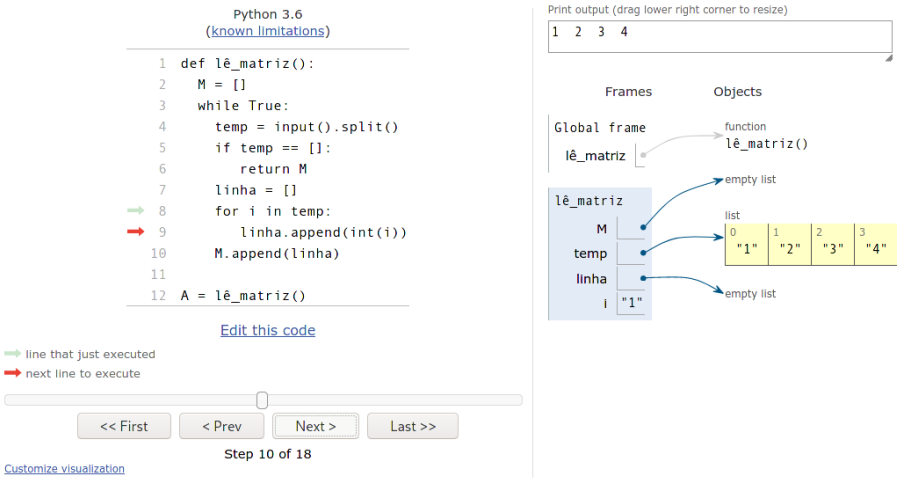
https://pythontutor.com/visualize.html#mode=edit



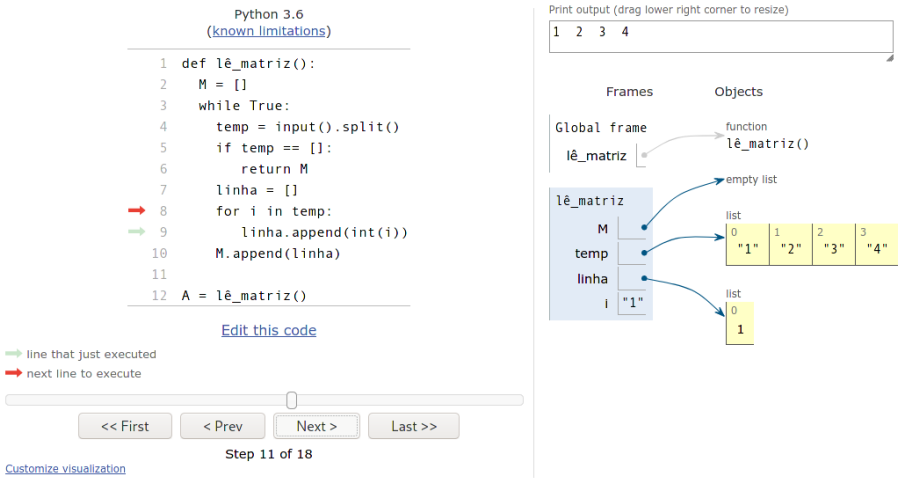
https://pythontutor.com/visualize.html#mode=edit



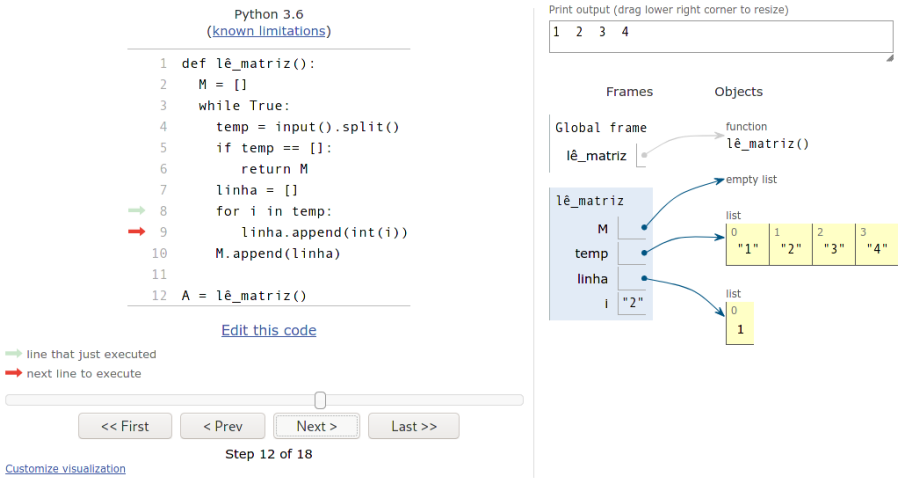
https://pythontutor.com/visualize.html#mode=edit



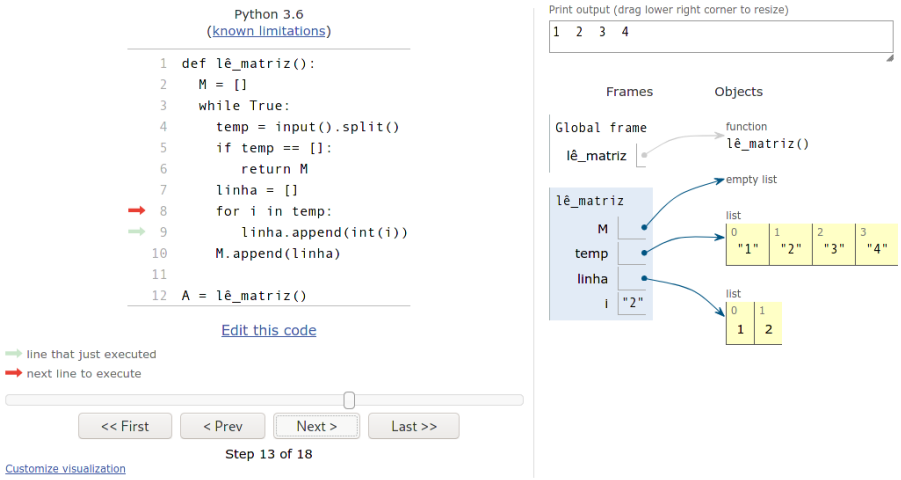
https://pythontutor.com/visualize.html#mode=edit



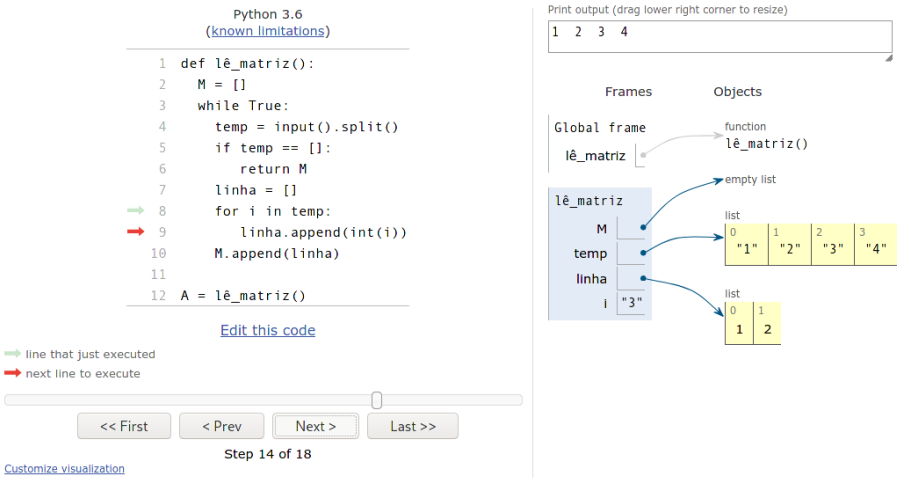
https://pythontutor.com/visualize.html#mode=edit



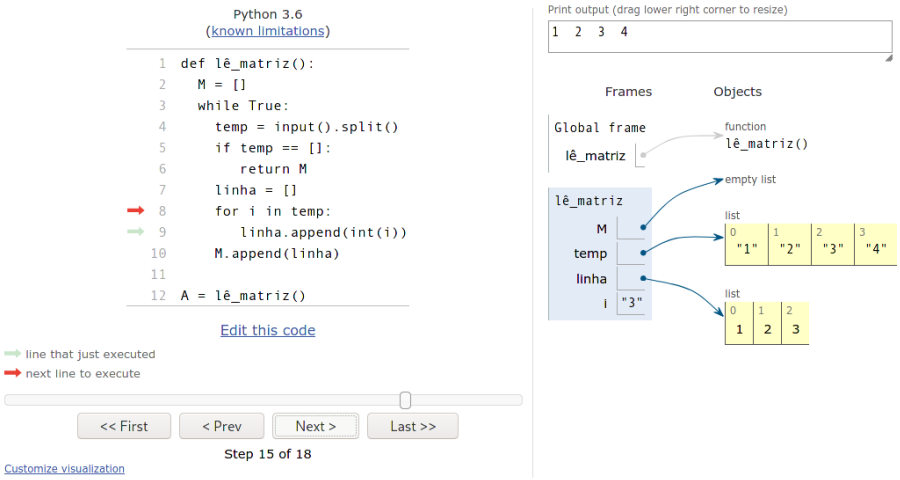
https://pythontutor.com/visualize.html#mode=edit



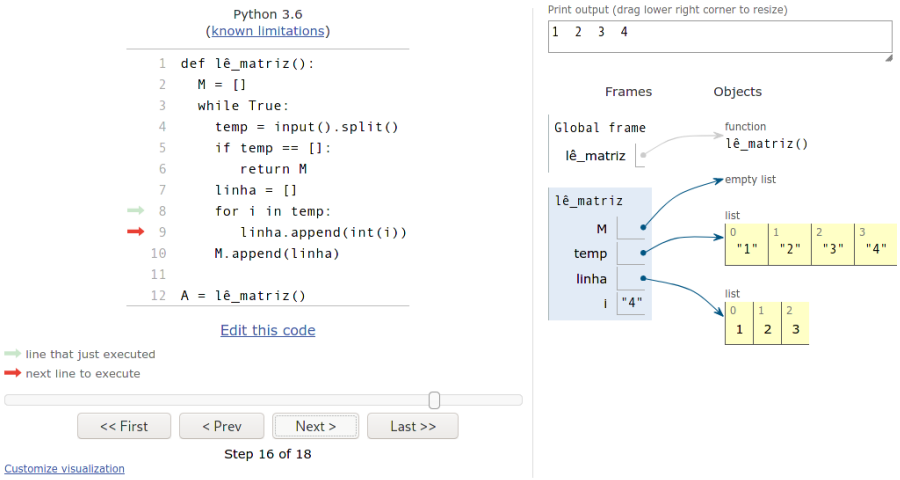
https://pythontutor.com/visualize.html#mode=edit



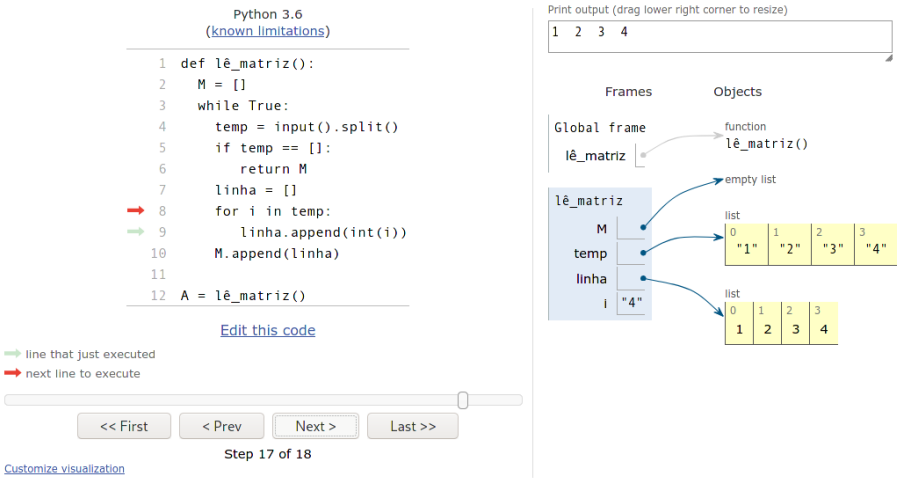
https://pythontutor.com/visualize.html#mode=edit



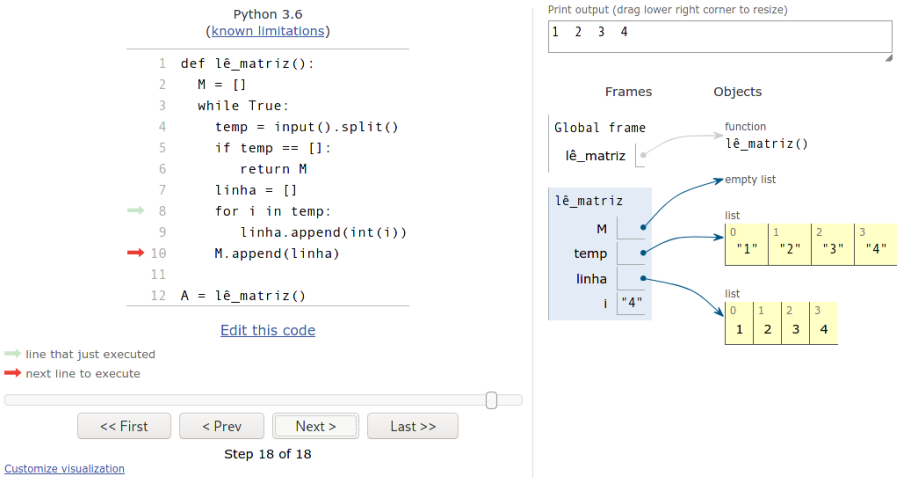
https://pythontutor.com/visualize.html#mode=edit



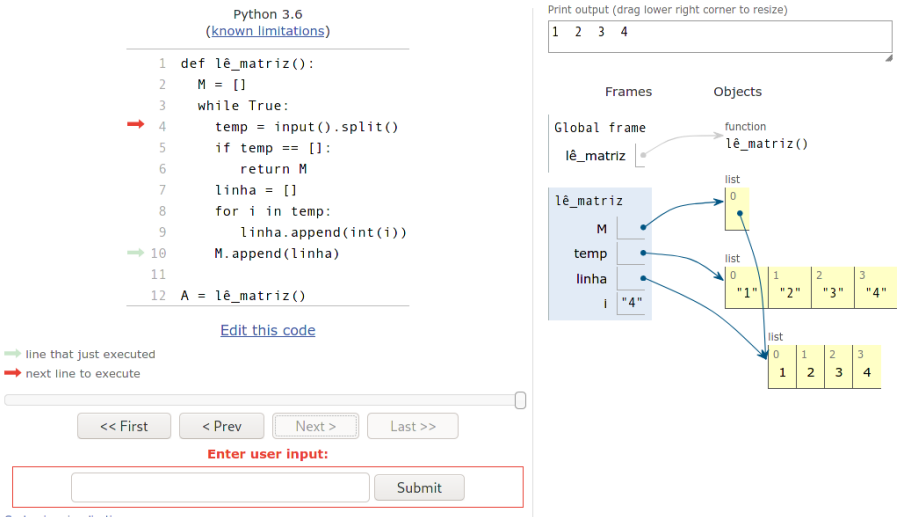
https://pythontutor.com/visualize.html#mode=edit



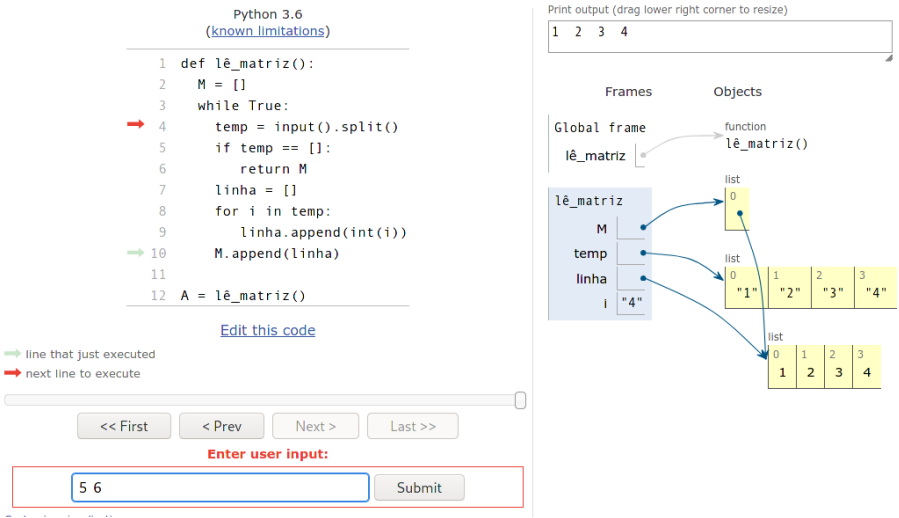
https://pythontutor.com/visualize.html#mode=edit



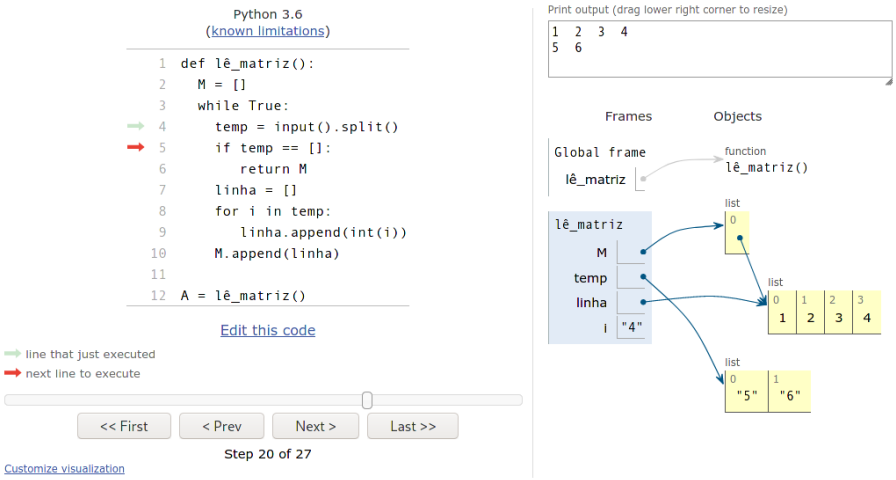
https://pythontutor.com/visualize.html#mode=edit



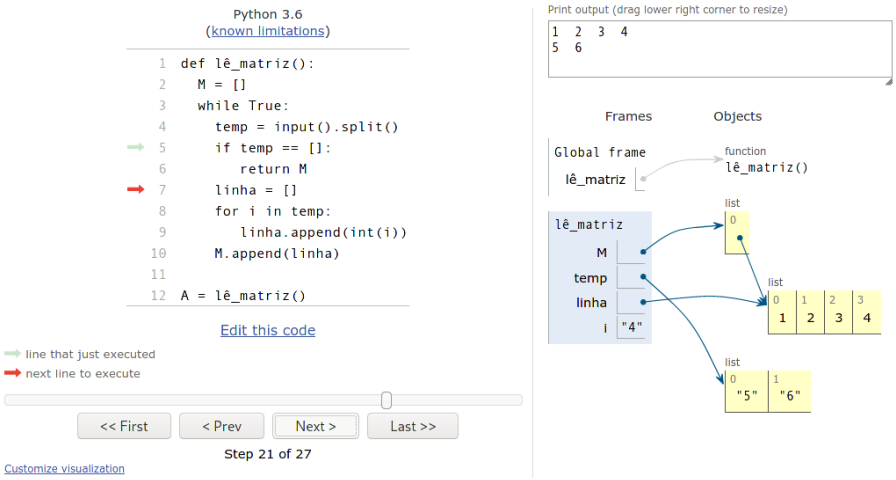
https://pythontutor.com/visualize.html#mode=edit



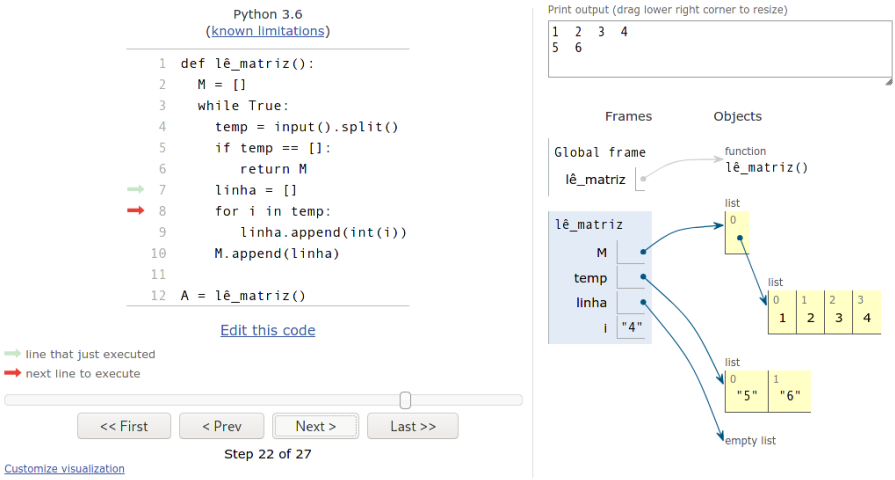
https://pythontutor.com/visualize.html#mode=edit



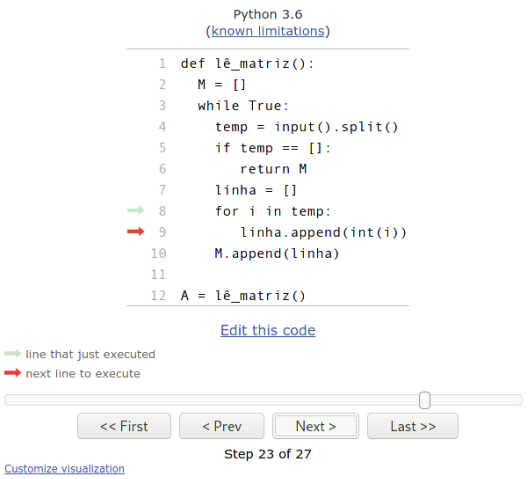
https://pythontutor.com/visualize.html#mode=edit



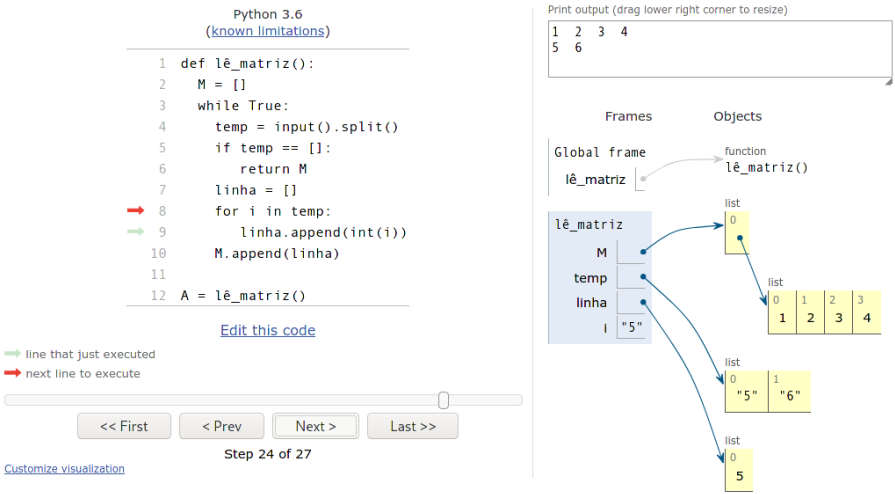
https://pythontutor.com/visualize.html#mode=edit



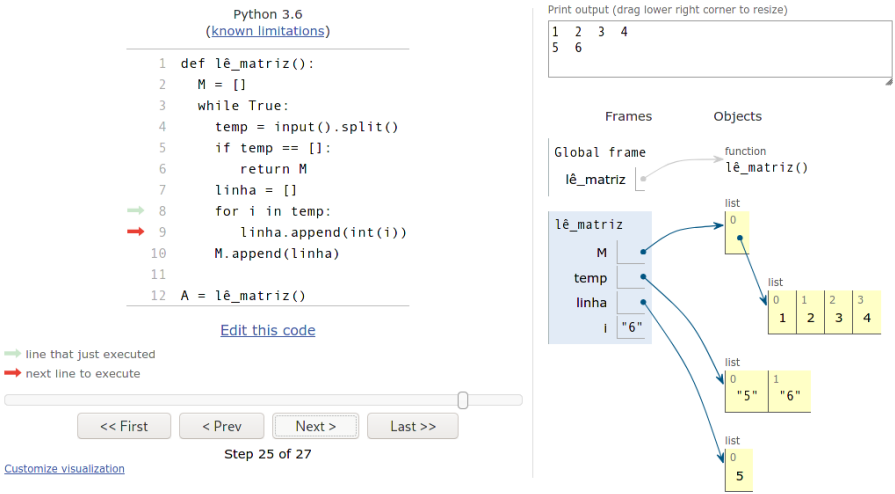
https://pythontutor.com/visualize.html#mode=edit



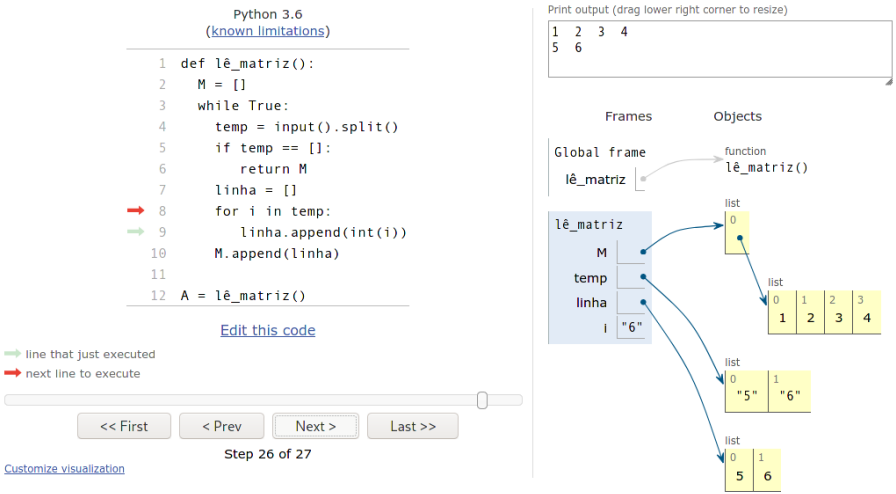
https://pythontutor.com/visualize.html#mode=edit



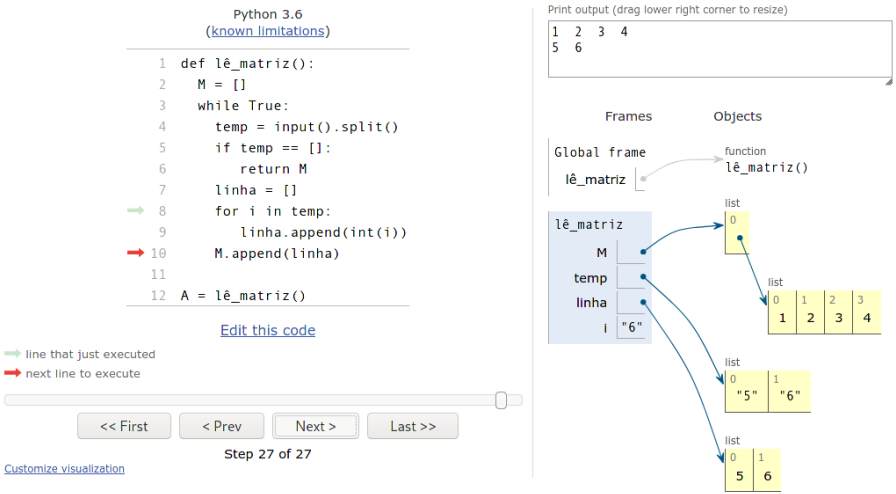
https://pythontutor.com/visualize.html#mode=edit



https://pythontutor.com/visualize.html#mode=edit



https://pythontutor.com/visualize.html#mode=edit



https://pythontutor.com/visualize.html#mode=edit

Python 3.6
(known limitations)

```
1  def lê_matriz():
2    M = []
3    while True:
4      temp = input().split()
5      if temp == []:
6        return M
7      linha = []
8      for i in temp:
9        linha.append(int(i))
10     M.append(linha)
11
12 A = lê_matriz()
```

Edit this code

line that just executed
next line to execute

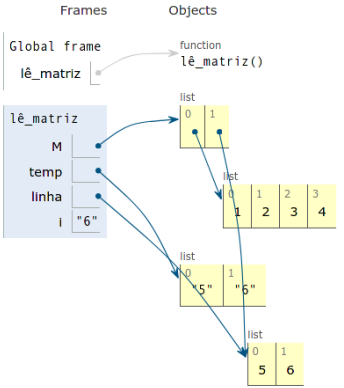
<< First | < Prev | Next > | Last >>

Enter user input:

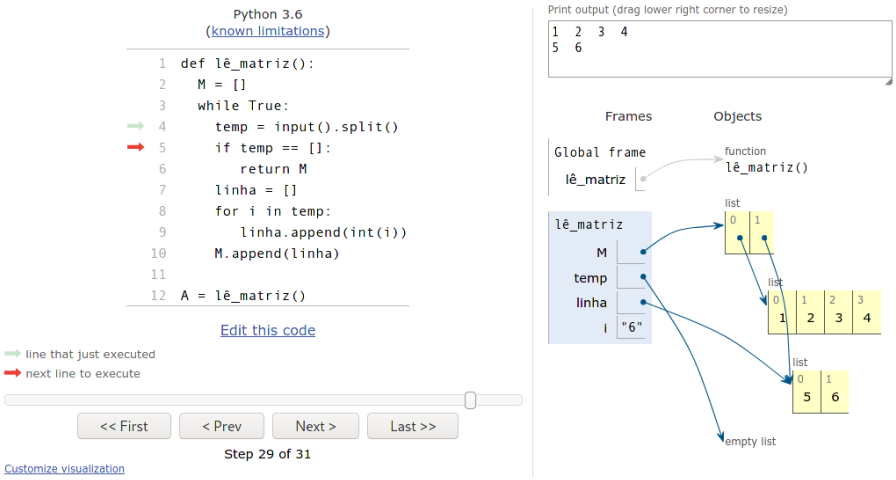
Submit

Print output (drag lower right corner to resize)

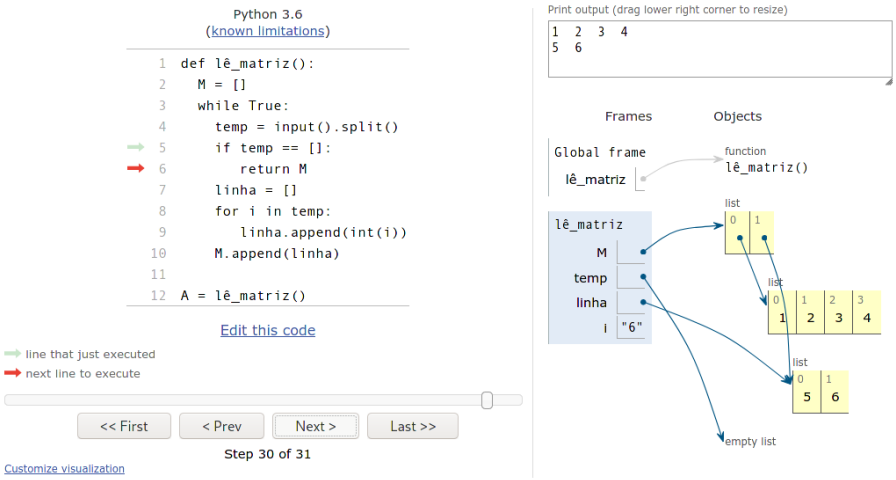
```
1  2  3  4
5  6
```



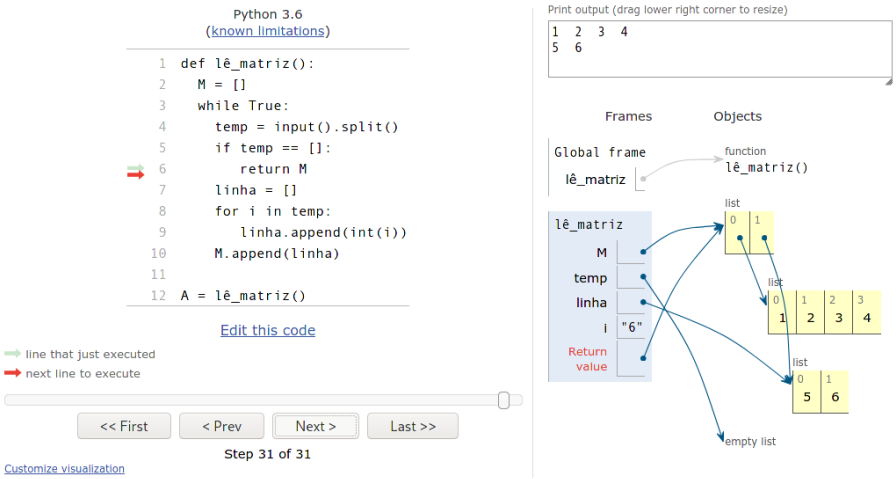
https://pythontutor.com/visualize.html#mode=edit



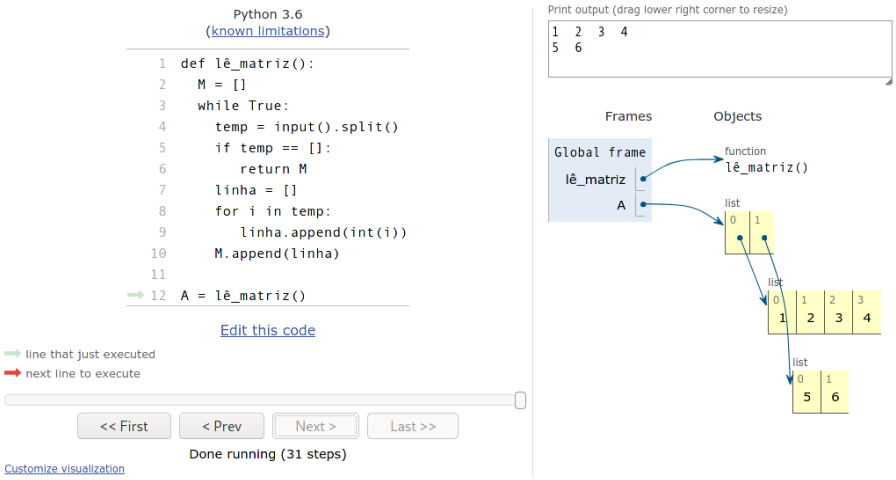
https://pythontutor.com/visualize.html#mode=edit



https://pythontutor.com/visualize.html#mode=edit



https://pythontutor.com/visualize.html#mode=edit



https://ic.unicamp.br/~mc102/aulas/aula09.pdf

```
def dimensões(M):
  linhas = len(M)
  colunas = len(M[0])
  for i in range(1, linhas):
    if len(M[i]) != colunas:
      return ()
  return (linhas, colunas)
```

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

```
def imprime_matriz(M):
  (linhas, colunas) = dimensões(M)
  for i in range(linhas):

    for j in range(colunas):
      print(M[i][j], end = " ")
    print()
```

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

```
def imprime_matriz(M):
  (linhas, colunas) = dimensões(M)
  for i in range(linhas):
    print(M[i][0], end = "")
    for j in range(1, colunas):
      print("", M[i][j], end = "")
    print()
```

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

```python
def imprime_matriz(M):
  for linha in M:
    # converte os elementos da lista para string
    aux = [str(i) for i in linha]
    print(" ".join(aux))
```

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

4. Escreva uma função que dada uma matriz (M), calcule a sua transposta (M^t). Exemplo:

$$M = \begin{bmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{bmatrix} \qquad M^t = \begin{bmatrix} 1 & 5 \\ 2 & 6 \\ 3 & 7 \\ 4 & 8 \end{bmatrix}$$

5. Escreva uma função que recebe duas matrizes (A e B). Se as duas matrizes tiverem dimensões compatíveis, sua função deve retornar a soma das duas ($C = A + B$). Caso contrário, sua função deve retornar uma lista vazia. Exemplo:

$$A = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \quad + \quad B = \begin{bmatrix} 5 & 6 \\ 1 & 3 \\ 4 & 2 \end{bmatrix} \quad = \quad C = \begin{bmatrix} 6 & 8 \\ 4 & 7 \\ 9 & 8 \end{bmatrix}$$

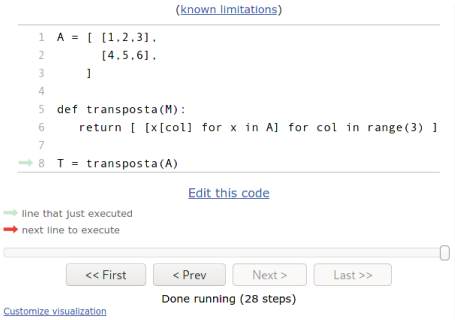
https://ic.unicamp.br/~mc102/aulas/aula09.pdf

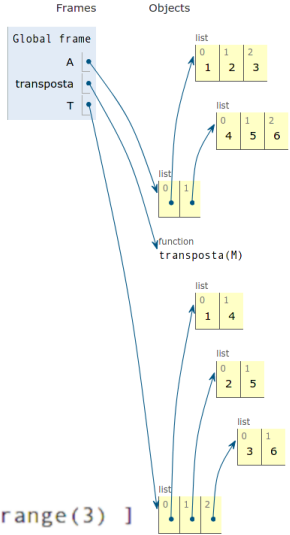
```
def transposta(M):
  T = []
  (linhas, colunas) = dimensões(M)
  for j in range(colunas):
    linha = []
    for i in range(linhas):
      linha.append(M[i][j])
    T.append(linha)
  return T
```

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

```python
def transposta(M):
  T = []
  (linhas, colunas) = dimensões(M)
  for j in range(colunas):
    T.append([])
    for i in range(linhas):
      T[j].append(M[i][j])
  return T
```

https://pythontutor.com/visualize.html#mode=edit






```
[ [x[col] for x in A] for col in range(3) ]
```

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

```
1  def soma(A, B):
2    C = []
3    dim_a = dimensões(A)
4    dim_b = dimensões(B)
5    if dim_a == dim_b:
6      (linhas, colunas) = dim_a
7      for i in range(linhas):
8        linha = []
9        for j in range(colunas):
10         linha.append(A[i][j] + B[i][j])
11       C.append(linha)
12   return C
```

https://pythontutor.com/visualize.html#mode=edit

```
1  A = [ [1,2,3],
2        [4,5,6],
3      ]
4  B = [ [10,20,30],
5        [40,50,60],
6      ]
7
8  def soma(A,B):
9      return [ [A[i][j]+B[i][j] for j in range(3)] for i in range(2) ]
10
11 S = soma(A,B)
```

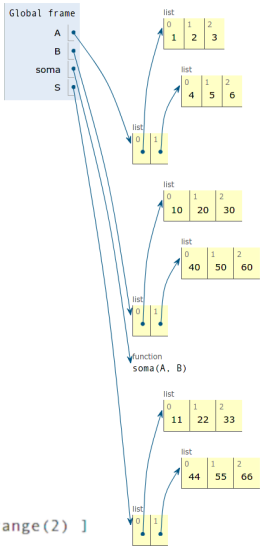
Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Done running (26 steps)

Customize visualization




```
[ [A[i][j]+B[i][j] for j in range(3)] for i in range(2) ]
```

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

6. Escreva uma função que recebe duas matrizes (A e B). Se as duas matrizes tiverem dimensões compatíveis, sua função deve retornar o produto das duas ($C = A \times B$). Caso contrário, sua função deve retornar uma lista vazia. Exemplo:

$$\underset{A}{\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}} \times \underset{B}{\begin{bmatrix} 5 \\ 6 \end{bmatrix}} = \underset{C}{\begin{bmatrix} 17 \\ 39 \end{bmatrix}}$$

7. Escreva uma função que dada uma matriz quadrada, verifique se ela é uma matriz diagonal. Exemplo:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 \end{bmatrix}$$

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

8. Escreva uma função que dada uma matriz quadrada, verifique se ela é uma matriz triangular inferior. Exemplo:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 4 & 4 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 9 & 0 & 2 & 3 \end{bmatrix}$$

9. Escreva uma função que dada uma matriz quadrada, verifique se ela é uma matriz triangular superior. Exemplo:

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

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

10. Uma matriz quadrada de números inteiros é um *quadrado mágico* se o valor da soma dos elementos de cada linha, de cada coluna e da diagonal principal e da diagonal secundária é o mesmo. Além disso, a matriz deve conter todos os números inteiros do intervalo $[1..n \times n]$. Exemplo:

$$\begin{bmatrix} 15 & 8 & 1 & 24 & 17 \\ 16 & 14 & 7 & 5 & 23 \\ 22 & 20 & 13 & 6 & 4 \\ 3 & 21 & 19 & 12 & 10 \\ 9 & 2 & 25 & 18 & 11 \end{bmatrix}$$

A matriz acima é um quadrado mágico, cujas somas valem 65. Escreva um programa que, dada uma matriz quadrada, verifique se ela é um *quadrado mágico*.

https://ic.unicamp.br/~mc102/aulas/aula09.pdf

11. Uma matriz de permutações é uma matriz quadrada cujos elementos são zeros ou uns, tal que em cada linha e em cada coluna exista exatamente um elemento igual a 1. Exemplo:

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Escreva um programa que, dada uma matriz quadrada, verifique se ela é uma matriz de permutações.

Perguntas ....

Referências

- Zanoni Dias, MC102, Algoritmos e Programação de Computadores, IC/UNICAMP, 2021. https://ic.unicamp.br/~mc102/

  Aula Introdutória [slides] [vídeo]
  Primeira Aula de Laboratório [slides] [vídeo]
  Python Básico: Tipos, Variáveis, Operadores, Entrada e Saída [slides] [vídeo]
  Comandos Condicionais [slides] [vídeo]
  Comandos de Repetição [slides] [vídeo]
  Listas e Tuplas [slides] [vídeo]
  Strings [slides] [vídeo]
  Dicionários [slides] [vídeo]
  Funções [slides] [vídeo]
  Objetos Multidimensionais [slides] [vídeo]
  Algoritmos de Ordenação [slides] [vídeo]
  Algoritmos de Busca [slides] [vídeo]
  Recursão [slides] [vídeo]
  Algoritmos de Ordenação Recursivos [slides] [vídeo]
  Arquivos [slides] [vídeo]
  Expressões Regulares [slides] [vídeo]
  Execução de Testes no Google Cloud Shell [slides] [vídeo]
  Numpy [slides] [vídeo]
  Pandas [slides] [vídeo]

- Panda - Cursos de Computação em Python (IME -USP) https://panda.ime.usp.br/

  Como Pensar Como um Cientista da Computação https://panda.ime.usp.br/pensepy/static/pensepy/
  Aulas de Introdução à Computação em Python https://panda.ime.usp.br/aulasPython/static/aulasPython/

- Fabio Kon, Introdução à Ciência da Computação com Python http://bit.ly/FabioKon/
- Socratica, Python Programming Tutorials http://bit.ly/SocraticaPython/
- Google - online editor for cloud-native applications (Python programming) https://shell.cloud.google.com/
- w3schools - Python Tutorial https://www.w3schools.com/python/
- Outros, citados nos Slides.