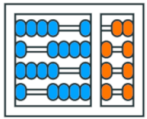


Instituto de Computação

UNIVERSIDADE ESTADUAL DE CAMPINAS

UNICAMP

# MC102 - Aula 15

## Exemplos sobre Objetos Multidimensionais

Algoritmos e Programação de Computadores

Turmas

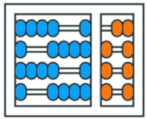
**OVXZ**


Prof. Lise R. R. Navarrete

lrommel@ic.unicamp.br


Quinta-feira, 12 de maio de 2022

19:00h - 21:00h (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

- Exemplo 1
- Exemplo 2
- Exemplo 3
- Exemplo 4
- Exemplo 5

# Exemplo 1

# Matriz de zeros

Escreva uma função que receba dois números inteiros positivos $n$ e $m$, e retorne uma matriz de tamanho $n \times m$ talque todos os seus elementos são zero.

Estratégia 1:

Criar linha x linha e adicionar na matriz.
Usar uma variavel auxiliar para cada linha

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

Python 3.6
(known limitations)

```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

Step 1 of 27

Frames Objects

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

Python 3.6
(known limitations)

```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

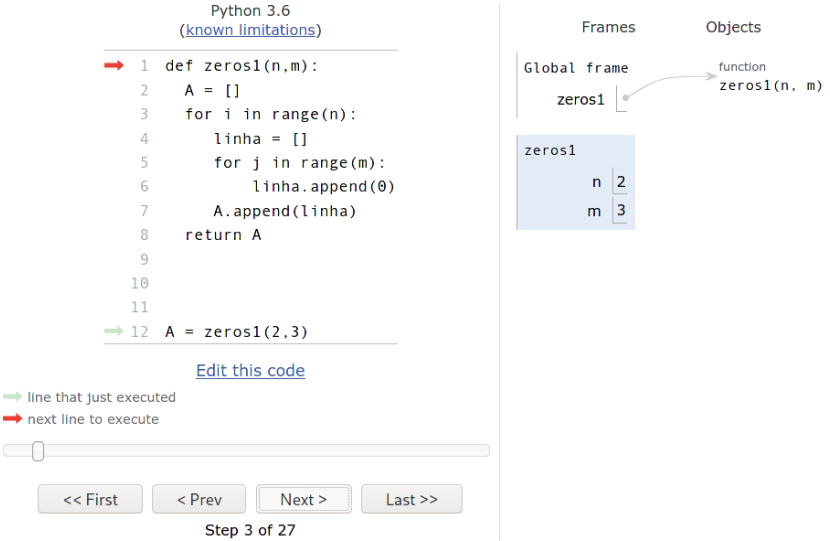
Step 2 of 27

Frames | Objects

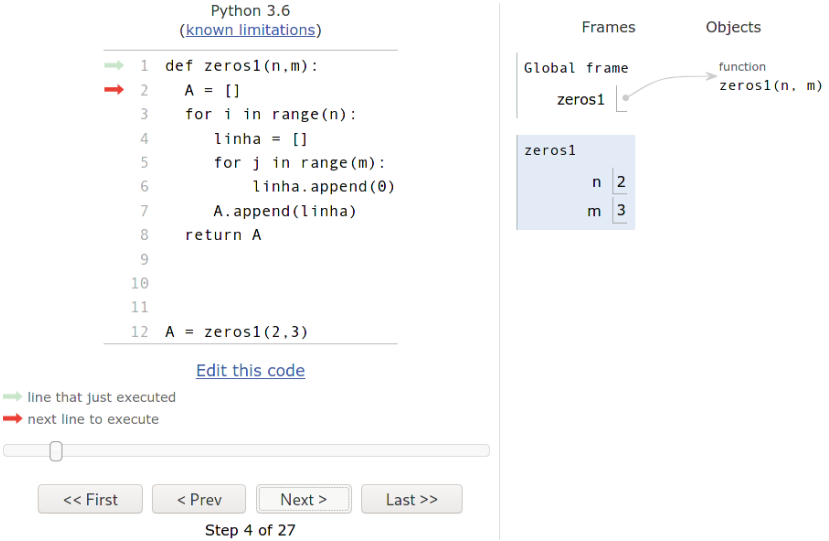
Global frame
zeros1

function
zeros1(n, m)

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)

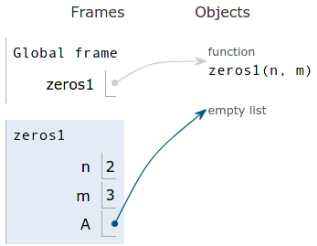
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

Step 5 of 27



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

Python 3.6
(known limitations)

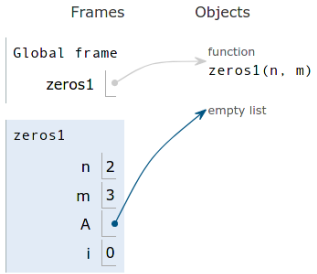
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

Step 6 of 27



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

Python 3.6
(known limitations)

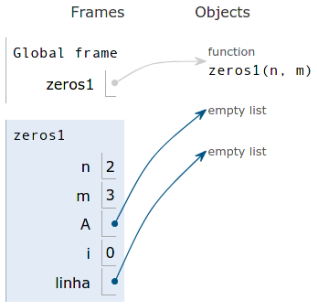
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

Step 7 of 27



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

Python 3.6
(known limitations)

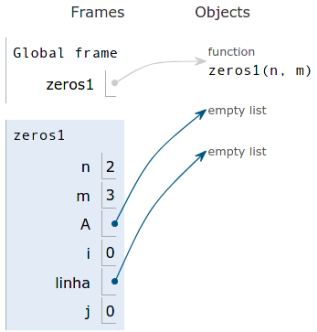
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

Step 8 of 27



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

Python 3.6
(known limitations)

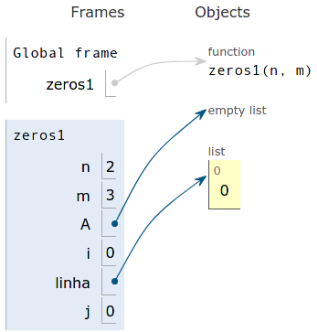
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

Step 9 of 27



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

Python 3.6
(known limitations)

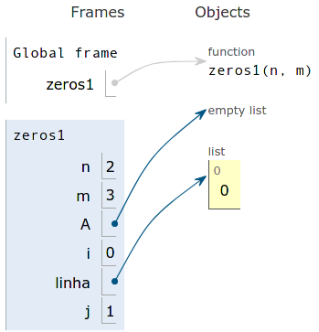
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

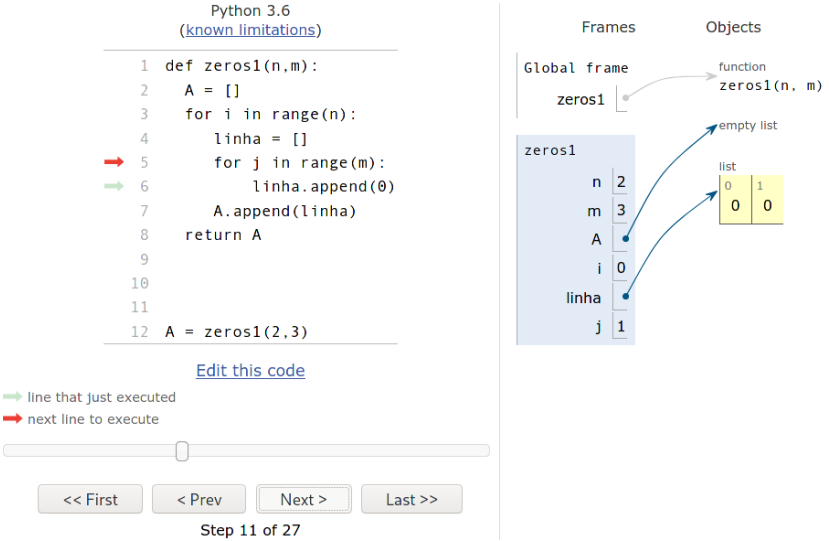
line that just executed
next line to execute

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

Step 10 of 27



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



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

Python 3.6
(known limitations)

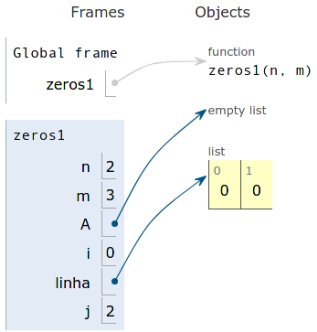
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

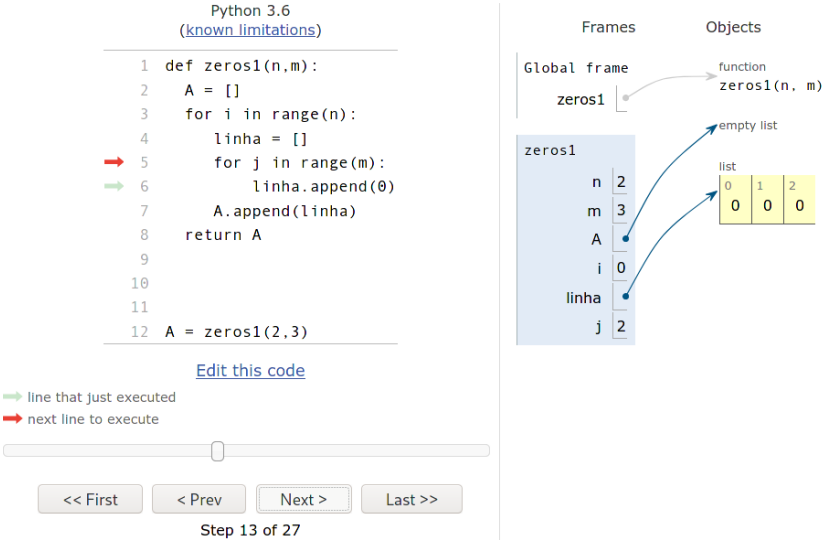
line that just executed
next line to execute

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

Step 12 of 27



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



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

Python 3.6
(known limitations)

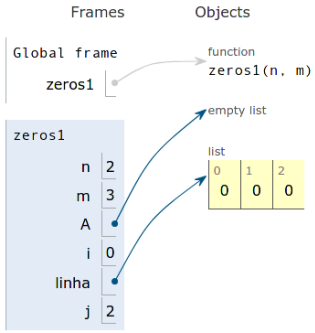
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

Step 14 of 27



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

Python 3.6
(known limitations)

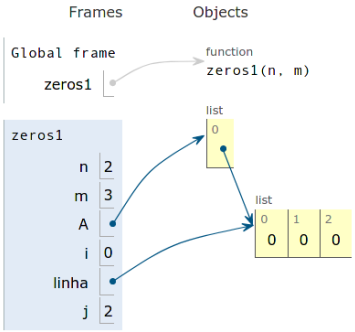
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

Step 15 of 27



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

Python 3.6
(known limitations)

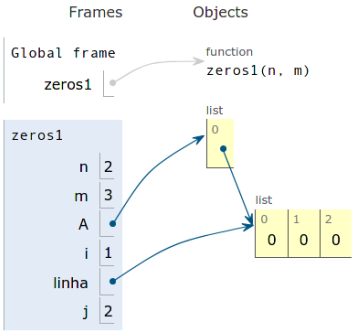
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

Step 16 of 27



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

Python 3.6
(known limitations)

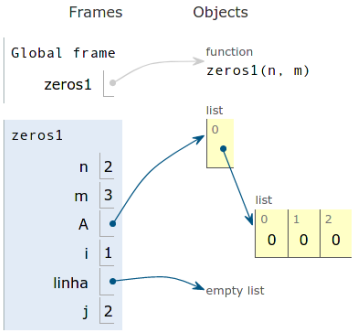
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

Step 17 of 27



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

Python 3.6
(known limitations)

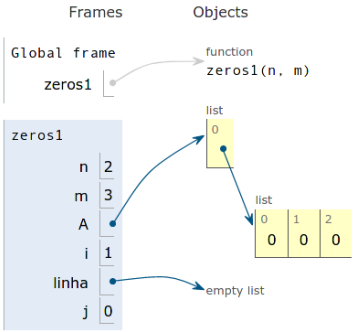
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

Step 18 of 27



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

Python 3.6
(known limitations)

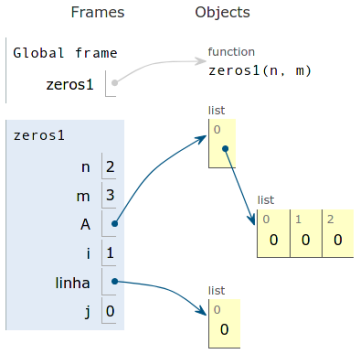
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

Step 19 of 27



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

Python 3.6
(known limitations)

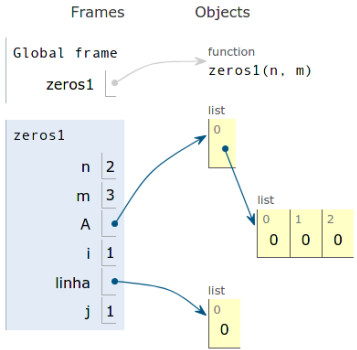
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

Step 20 of 27



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

Python 3.6
(known limitations)

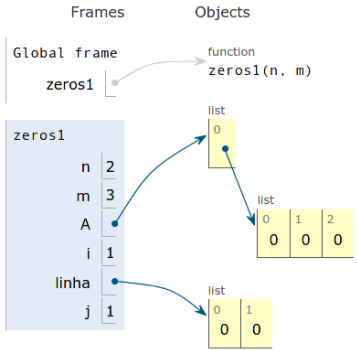
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

Step 21 of 27



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

Python 3.6
(known limitations)

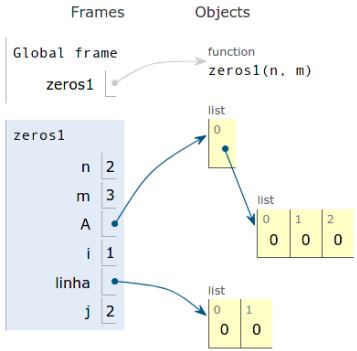
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

Step 22 of 27



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

Python 3.6
(known limitations)

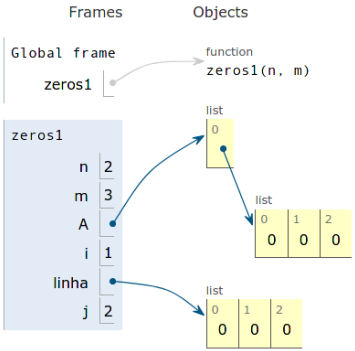
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

Step 23 of 27



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

Python 3.6
(known limitations)

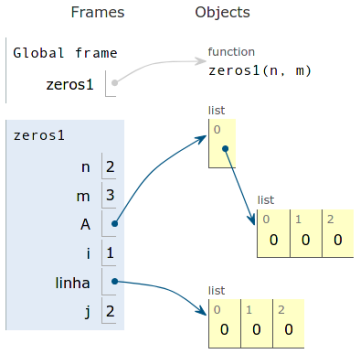
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

Step 24 of 27



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

Python 3.6
(known limitations)

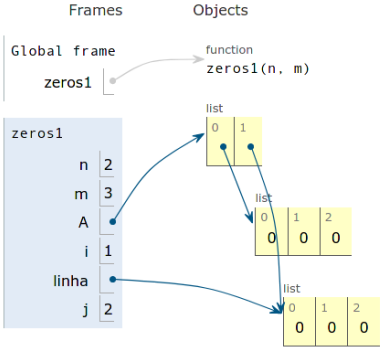
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

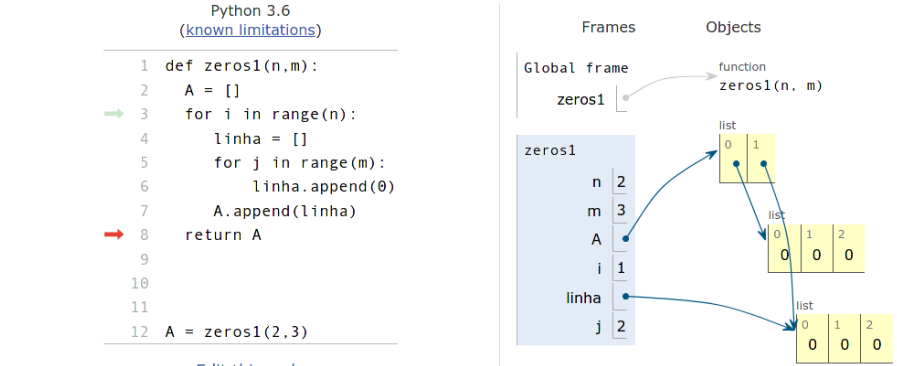
line that just executed
next line to execute

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

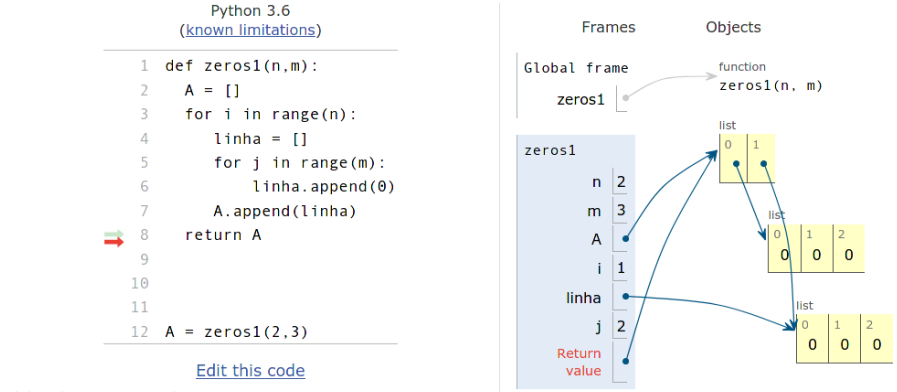
Step 25 of 27



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)

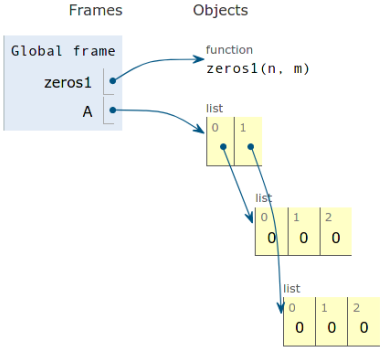
```
1  def zeros1(n,m):
2    A = []
3    for i in range(n):
4       linha = []
5       for j in range(m):
6           linha.append(0)
7       A.append(linha)
8    return A
9
10
11
12 A = zeros1(2,3)
```

Edit this code

line that just executed
next line to execute

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

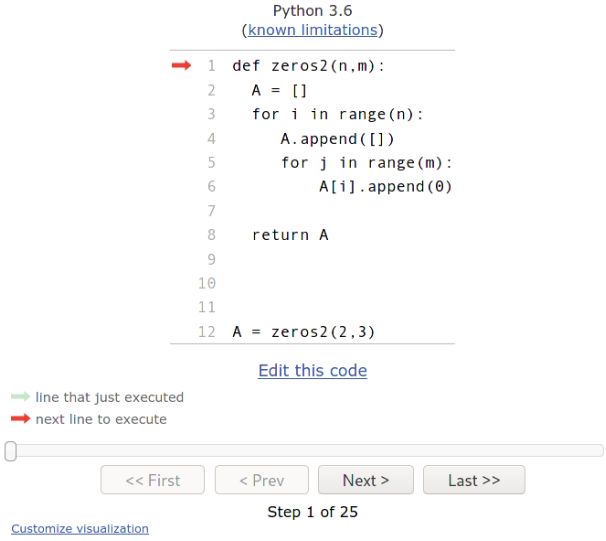
Done running (27 steps)



Estratégia 2:

Criar linha x linha e adicionar na matriz.
Sem usar uma variavel auxiliar para crear as linhas

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



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

Edit this code

line that just executed
next line to execute

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

Step 2 of 25

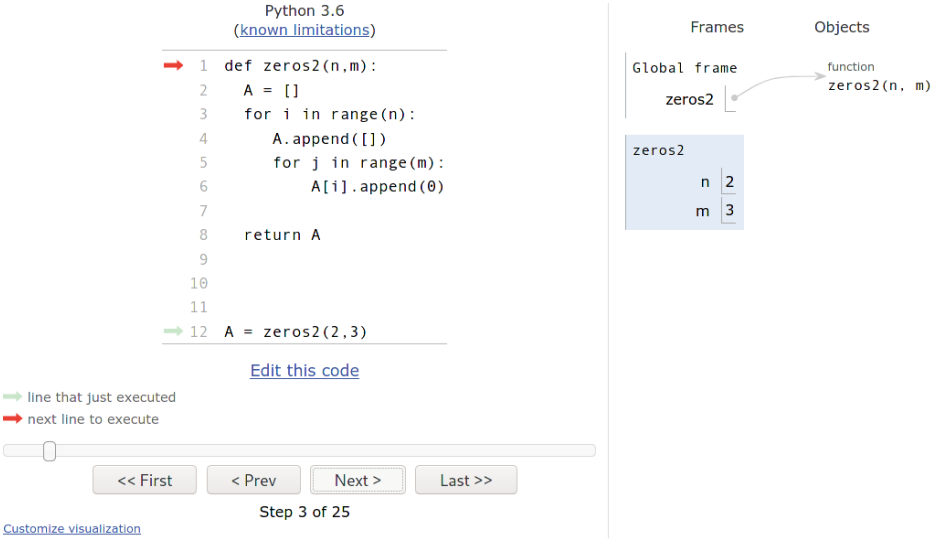
Customize visualization

Frames Objects

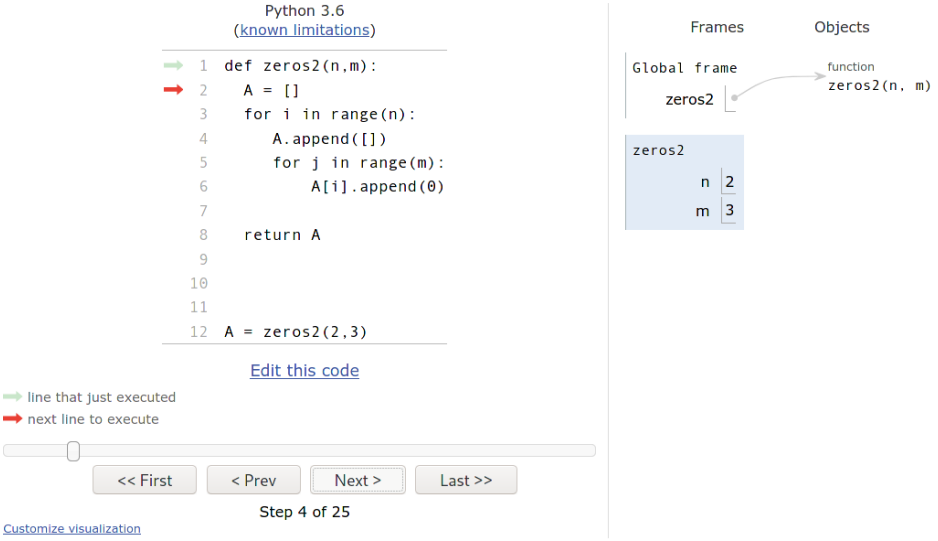
Global frame
zeros2

function
zeros2(n, m)

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 zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

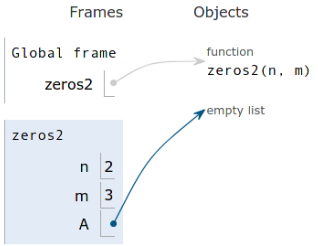
Edit this code

line that just executed
next line to execute

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

Step 5 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

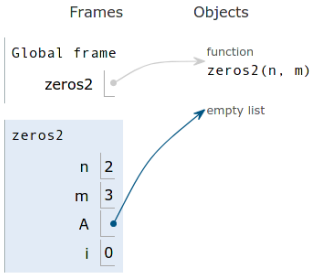
Edit this code

line that just executed
next line to execute

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

Step 6 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

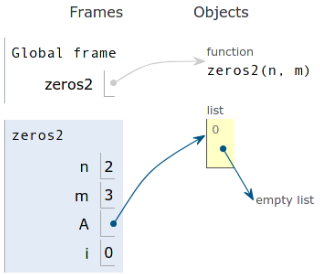
Edit this code

line that just executed
next line to execute

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

Step 7 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

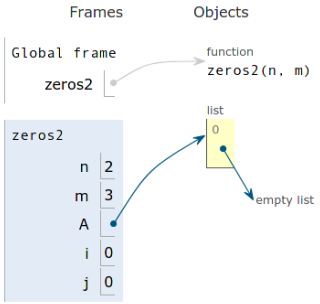
Edit this code

line that just executed
next line to execute

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

Step 8 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

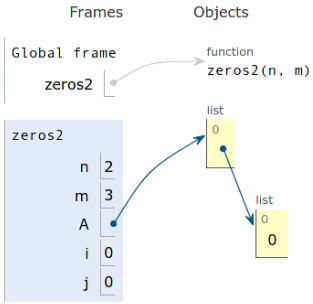
Edit this code

line that just executed
next line to execute

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

Step 9 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

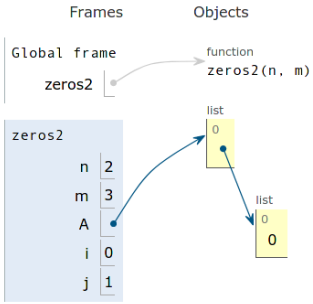
Edit this code

line that just executed
next line to execute

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

Step 10 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

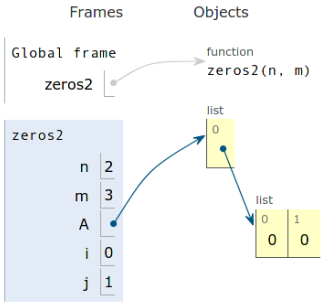
Edit this code

line that just executed
next line to execute

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

Step 11 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

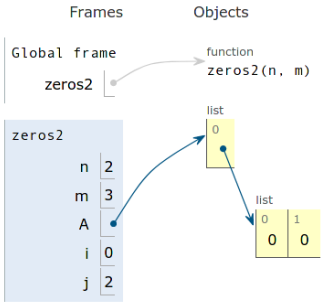
Edit this code

line that just executed
next line to execute

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

Step 12 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

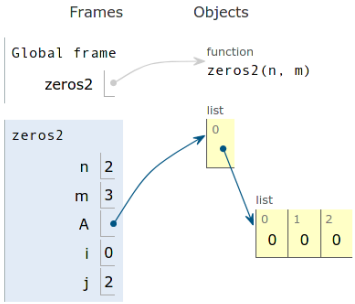
Edit this code

line that just executed
next line to execute

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

Step 13 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

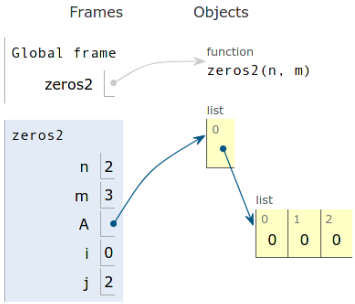
Edit this code

line that just executed
next line to execute

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

Step 14 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

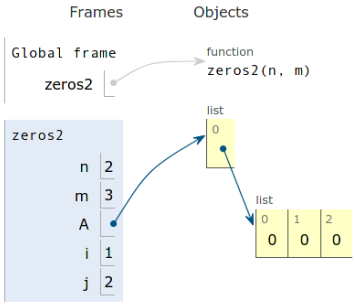
Edit this code

line that just executed
next line to execute

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

Step 15 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

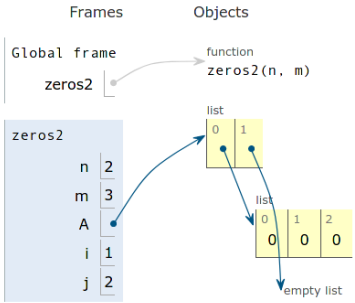
Edit this code

line that just executed
next line to execute

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

Step 16 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

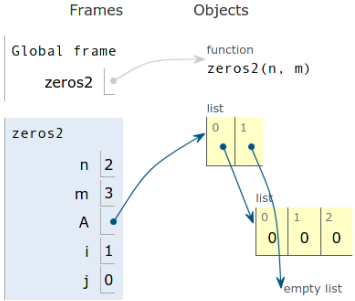
Edit this code

line that just executed
next line to execute

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

Step 17 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

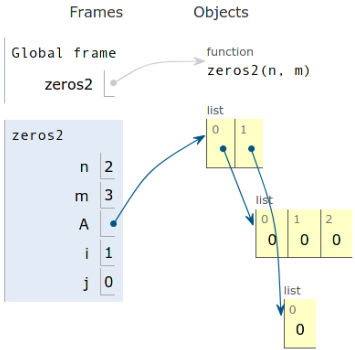
Edit this code

line that just executed
next line to execute

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

Step 18 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

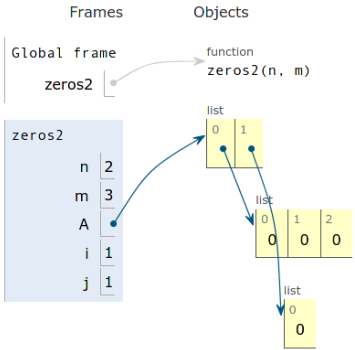
Edit this code

line that just executed
next line to execute

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

Step 19 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

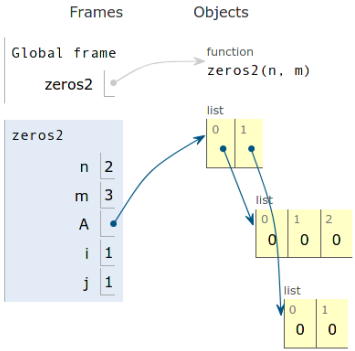
Edit this code

line that just executed
next line to execute

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

Step 20 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

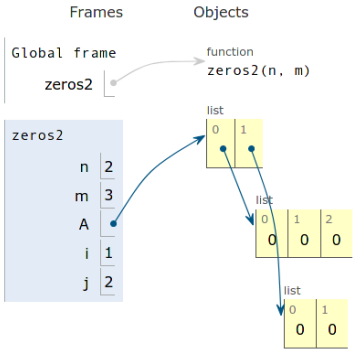
Edit this code

line that just executed
next line to execute

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

Step 21 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

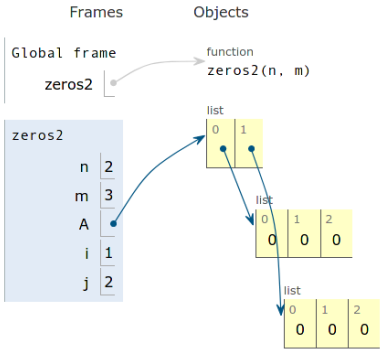
Edit this code

line that just executed
next line to execute

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

Step 22 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

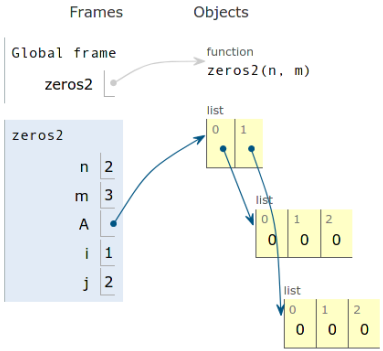
Edit this code

line that just executed
next line to execute

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

Step 23 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

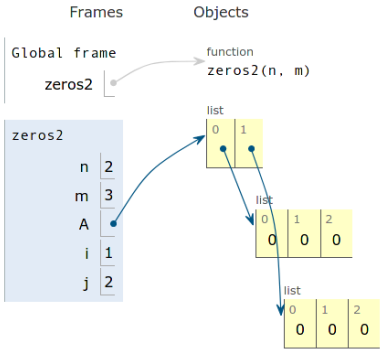
Edit this code

line that just executed
next line to execute

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

Step 24 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

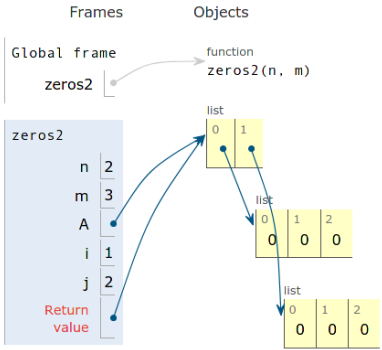
Edit this code

line that just executed
next line to execute

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

Step 25 of 25

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros2(n,m):
2    A = []
3    for i in range(n):
4       A.append([])
5       for j in range(m):
6           A[i].append(0)
7
8    return A
9
10
11
12 A = zeros2(2,3)
```

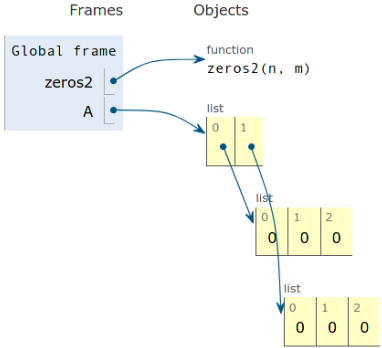
Edit this code

line that just executed
next line to execute

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

Done running (25 steps)

Customize visualization



Estratégia 3:

Criar linha x linha e adicionar na matriz.
Sem usar uma variavel auxiliar para crear as linhas.
Usando "List Comprehensions"para criar de forma concisa "inline"as linhas

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

Python 3.6
(known limitations)

```
1  def zeros3(n,m):
2    A = []
3    for i in range(n):
4       A.append([0 for j in range(m)])
5
6
7
8    return A
9
10
11
12 A = zeros3(2,3)
```

Edit this code

line that just executed
next line to execute

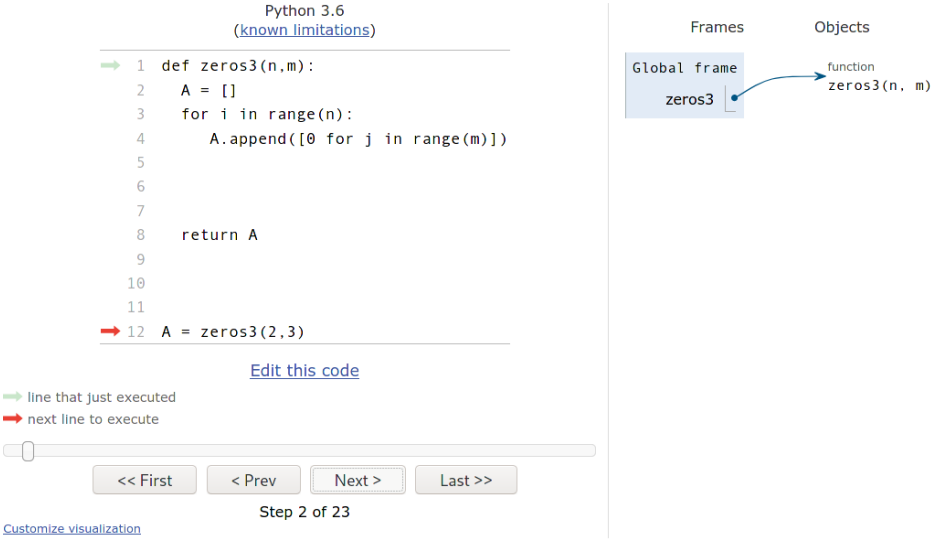
<< First | < Prev | Next > | Last >>

Step 1 of 23

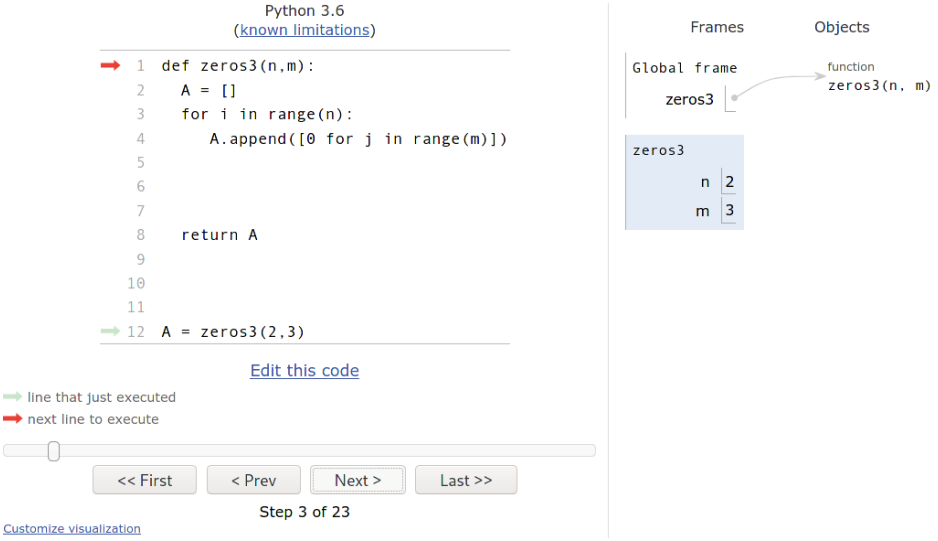
Customize visualization

Frames Objects

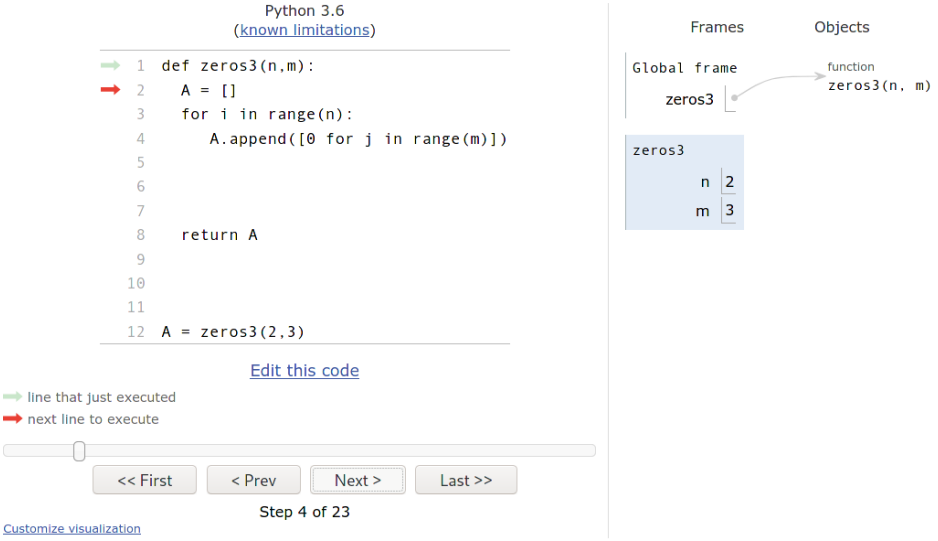
https://pythontutor.com/visualize.html#mode=edit



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 zeros3(n,m):
2    A = []
3    for i in range(n):
4       A.append([0 for j in range(m)])
5
6
7
8    return A
9
10
11
12 A = zeros3(2,3)
```

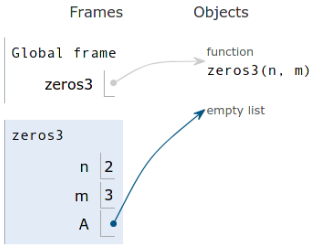
Edit this code

line that just executed
next line to execute

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

Step 5 of 23

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros3(n,m):
2    A = []
3    for i in range(n):
4       A.append([0 for j in range(m)])
5
6
7
8    return A
9
10
11
12 A = zeros3(2,3)
```

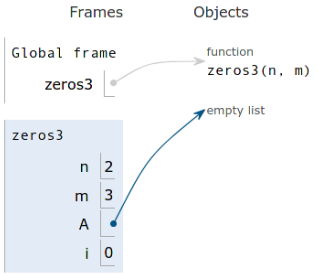
Edit this code

line that just executed
next line to execute

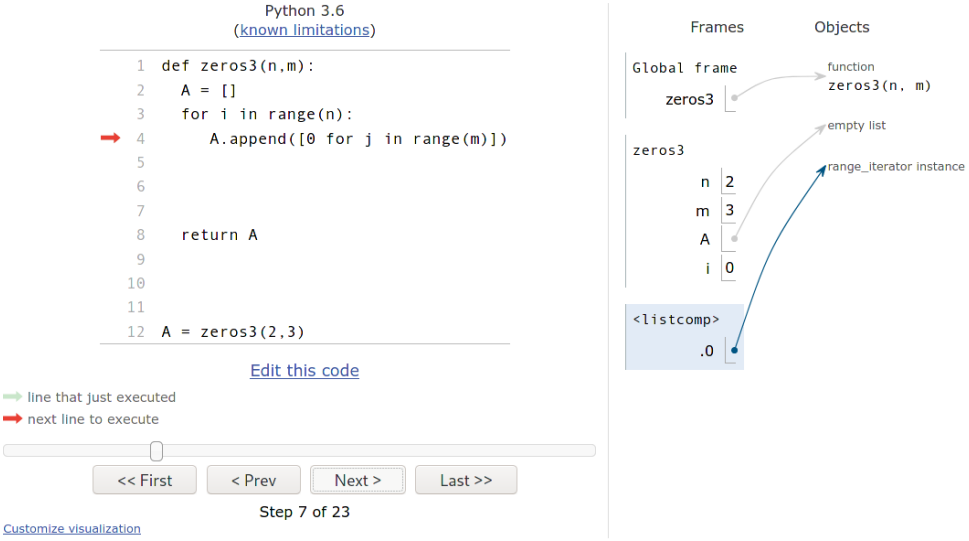
<< First | < Prev | Next > | Last >>

Step 6 of 23

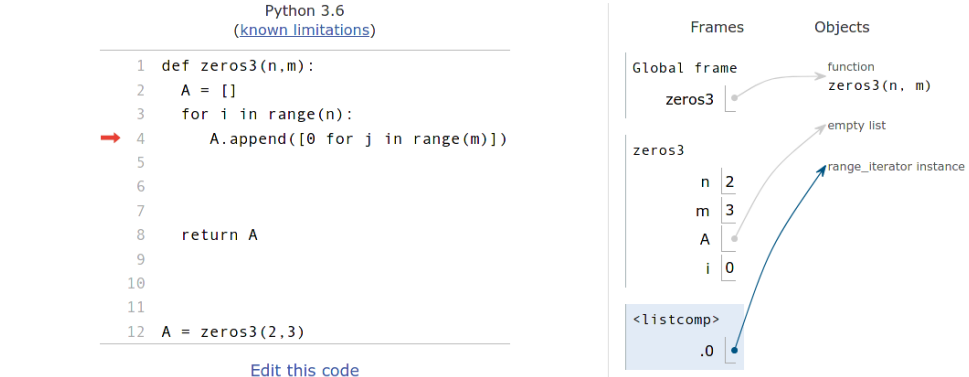
Customize visualization



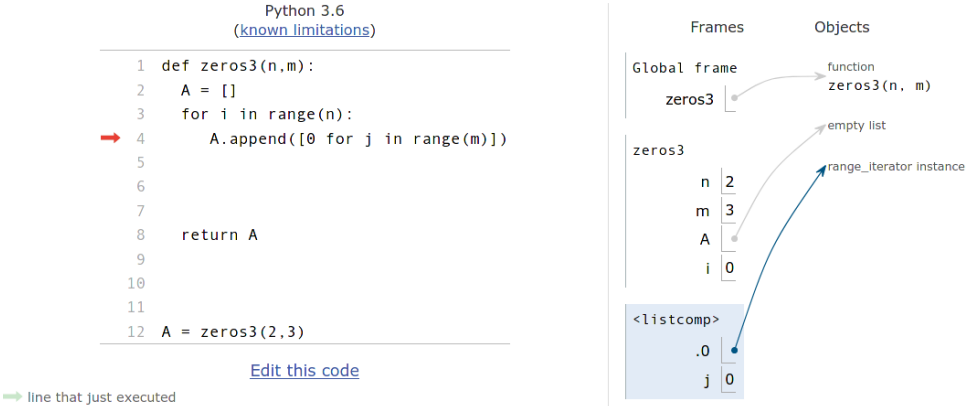
https://pythontutor.com/visualize.html#mode=edit



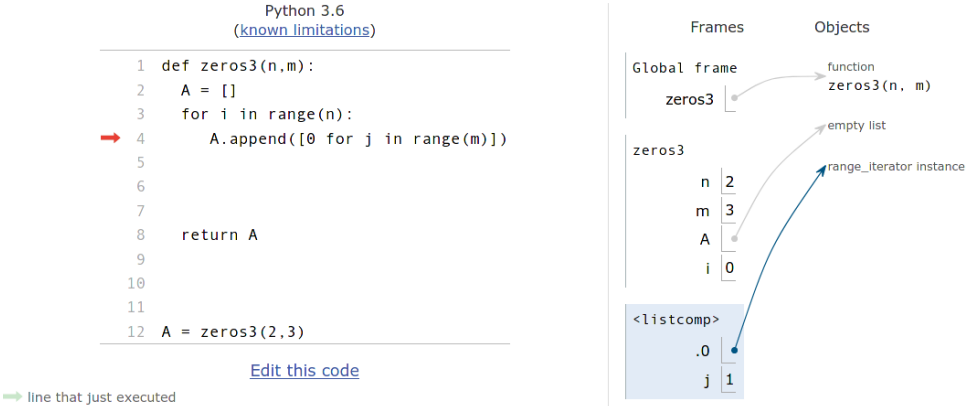
https://pythontutor.com/visualize.html#mode=edit



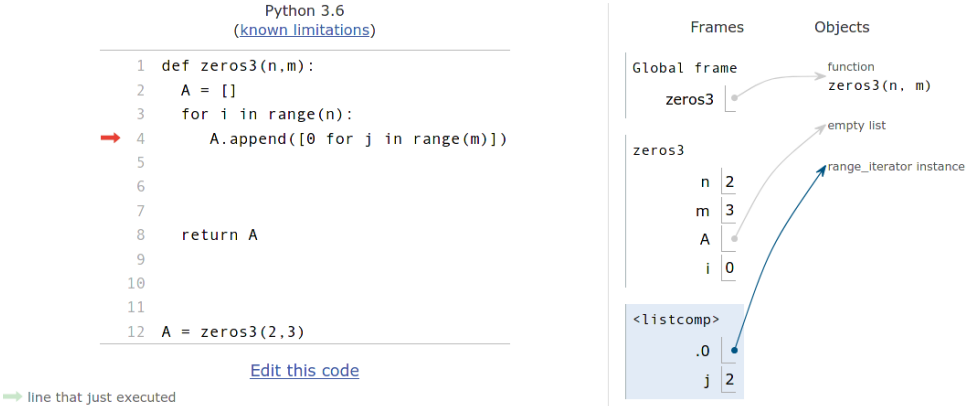
https://pythontutor.com/visualize.html#mode=edit



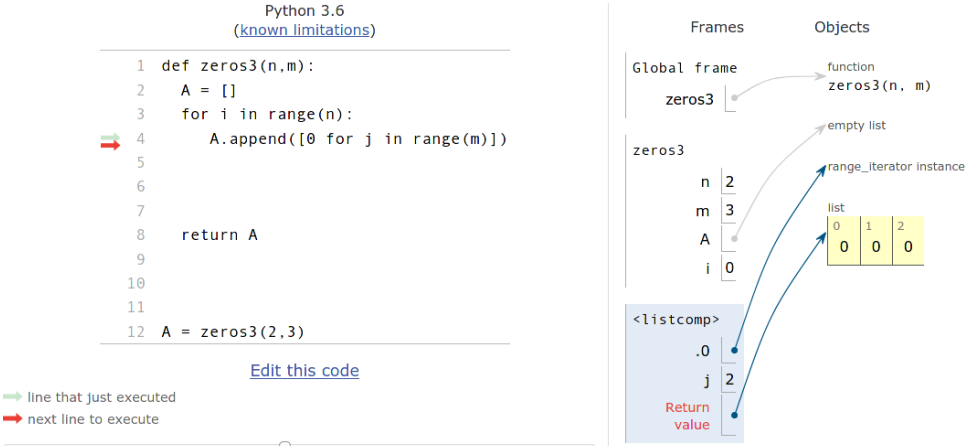
https://pythontutor.com/visualize.html#mode=edit



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 zeros3(n,m):
2    A = []
3    for i in range(n):
4       A.append([0 for j in range(m)])
5
6
7
8    return A
9
10
11
12 A = zeros3(2,3)
```

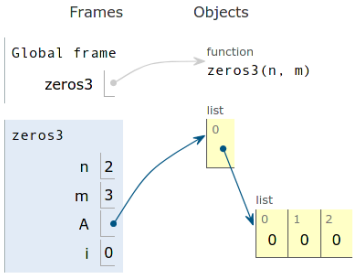
Edit this code

line that just executed
next line to execute

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

Step 13 of 23

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros3(n,m):
2    A = []
3    for i in range(n):
4       A.append([0 for j in range(m)])
5
6
7
8    return A
9
10
11
12 A = zeros3(2,3)
```

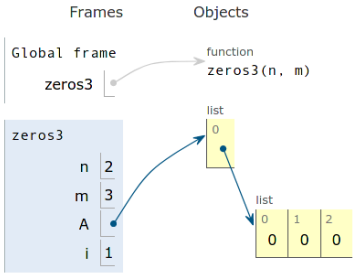
Edit this code

line that just executed
next line to execute

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

Step 14 of 23

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros3(n,m):
2    A = []
3    for i in range(n):
4       A.append([0 for j in range(m)])
5
6
7
8    return A
9
10
11
12 A = zeros3(2,3)
```

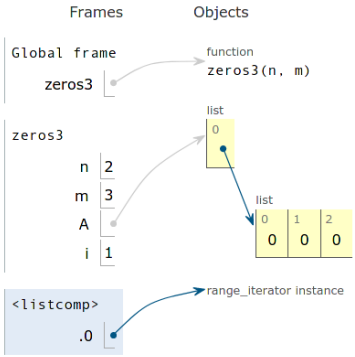
Edit this code

line that just executed
next line to execute

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

Step 15 of 23

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros3(n,m):
2    A = []
3    for i in range(n):
4       A.append([0 for j in range(m)])
5
6
7
8    return A
9
10
11
12 A = zeros3(2,3)
```

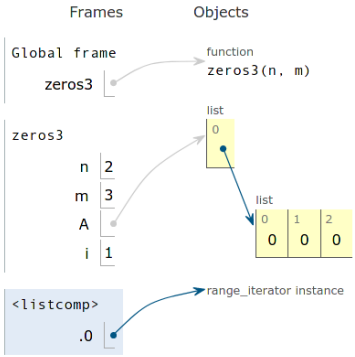
Edit this code

line that just executed
next line to execute

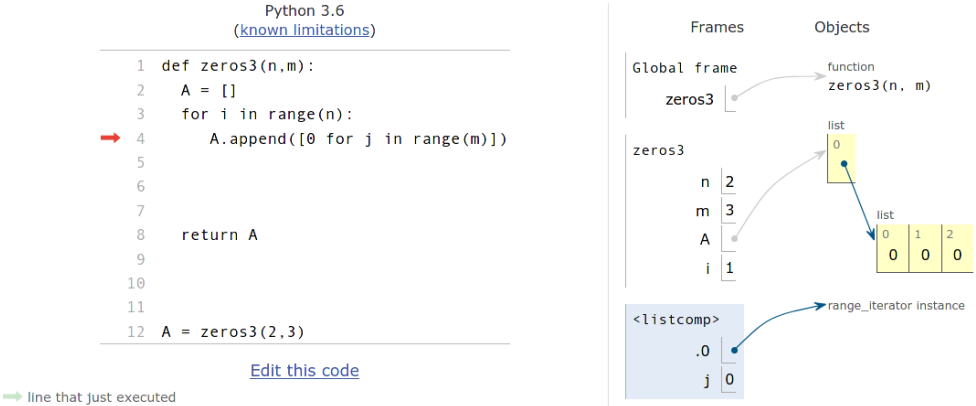
<< First | < Prev | Next > | Last >>

Step 16 of 23

Customize visualization



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



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

Python 3.6
(known limitations)

```
1  def zeros3(n,m):
2    A = []
3    for i in range(n):
4       A.append([0 for j in range(m)])
5
6
7
8    return A
9
10
11
12 A = zeros3(2,3)
```

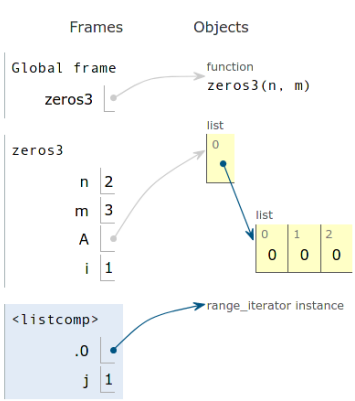
Edit this code

line that just executed
next line to execute

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

Step 18 of 23

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros3(n,m):
2    A = []
3    for i in range(n):
4       A.append([0 for j in range(m)])
5
6
7
8    return A
9
10
11
12 A = zeros3(2,3)
```

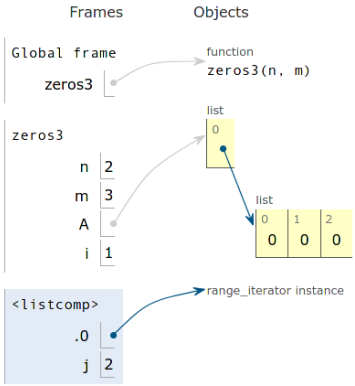
Edit this code

line that just executed
next line to execute

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

Step 19 of 23

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros3(n,m):
2    A = []
3    for i in range(n):
4       A.append([0 for j in range(m)])
5
6
7
8    return A
9
10
11
12 A = zeros3(2,3)
```

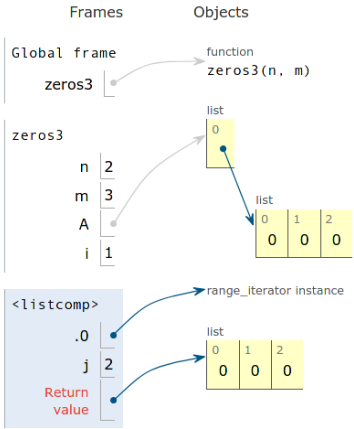
Edit this code

line that just executed
next line to execute

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

Step 20 of 23

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros3(n,m):
2    A = []
3    for i in range(n):
4       A.append([0 for j in range(m)])
5
6
7
8    return A
9
10
11
12 A = zeros3(2,3)
```

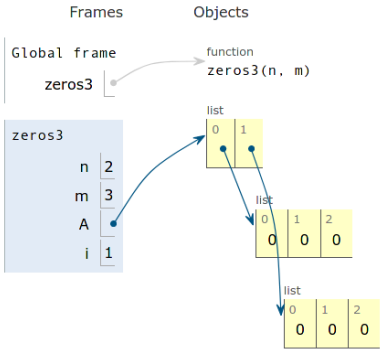
Edit this code

line that just executed
next line to execute

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

Step 21 of 23

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros3(n,m):
2    A = []
3    for i in range(n):
4       A.append([0 for j in range(m)])
5
6
7
8    return A
9
10
11
12 A = zeros3(2,3)
```

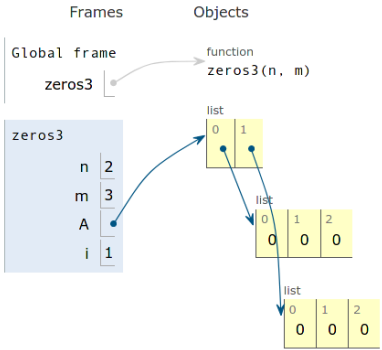
Edit this code

line that just executed
next line to execute

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

Step 22 of 23

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros3(n,m):
2    A = []
3    for i in range(n):
4       A.append([0 for j in range(m)])
5
6
7
8    return A
9
10
11
12 A = zeros3(2,3)
```

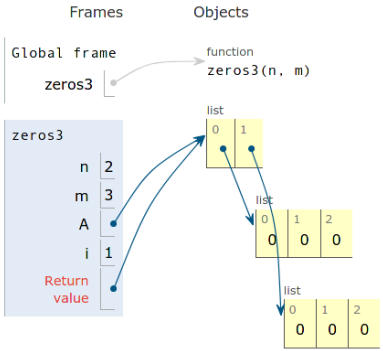
Edit this code

line that just executed
next line to execute

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

Step 23 of 23

Customize visualization



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

Python 3.6
(known limitations)

```
1  def zeros3(n,m):
2    A = []
3    for i in range(n):
4       A.append([0 for j in range(m)])
5
6
7
8    return A
9
10
11
12 A = zeros3(2,3)
```

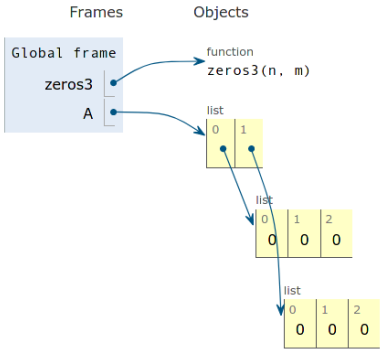
Edit this code

line that just executed
next line to execute

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

Done running (23 steps)

Customize visualization



Estratégia 4:

Usando "List Comprehensions"
para criar de forma concisa "inline"toda a matriz

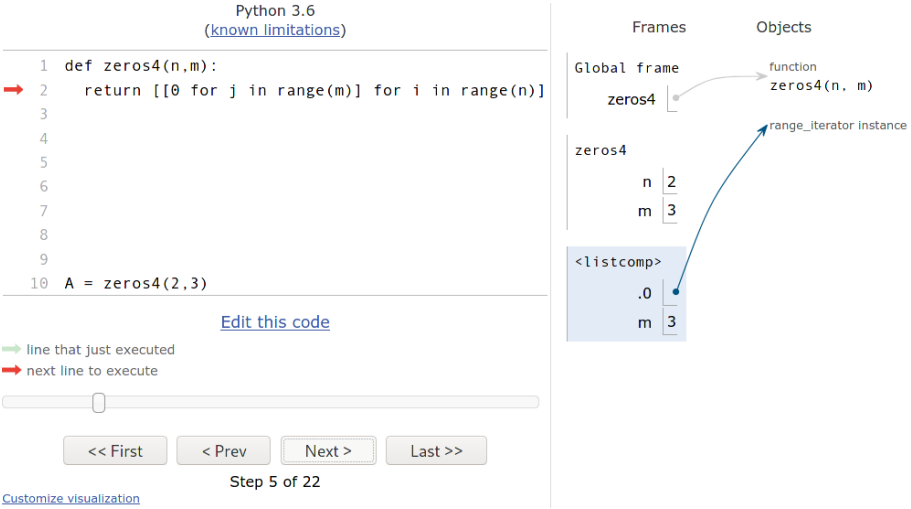
https://pythontutor.com/visualize.html#mode=edit

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

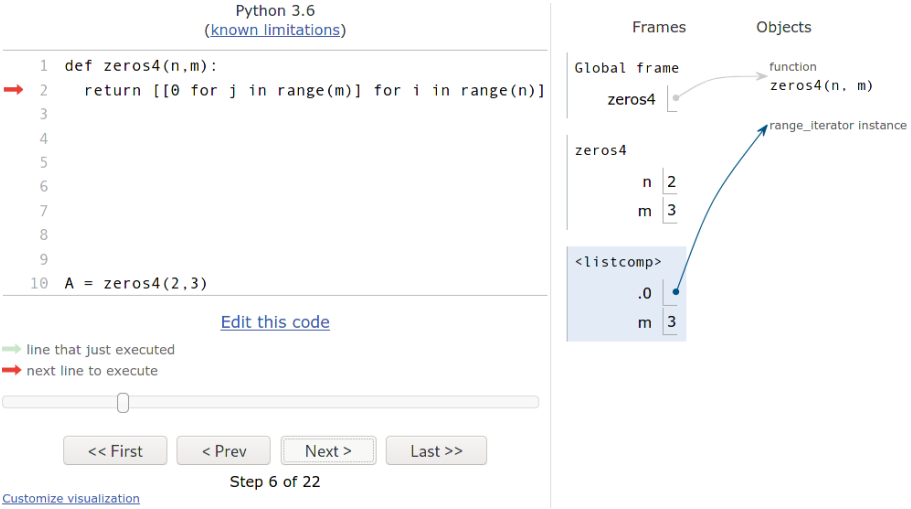
https://pythontutor.com/visualize.html#mode=edit

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

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



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

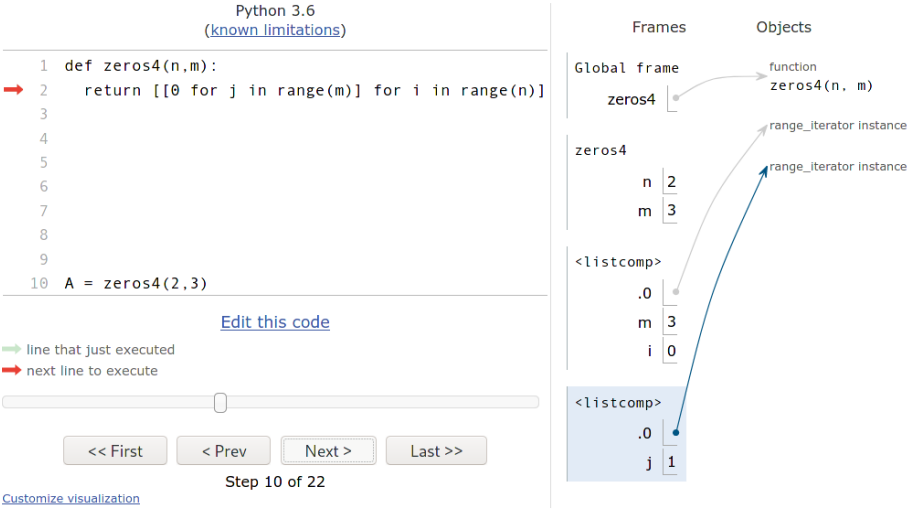


https://pythontutor.com/visualize.html#mode=edit

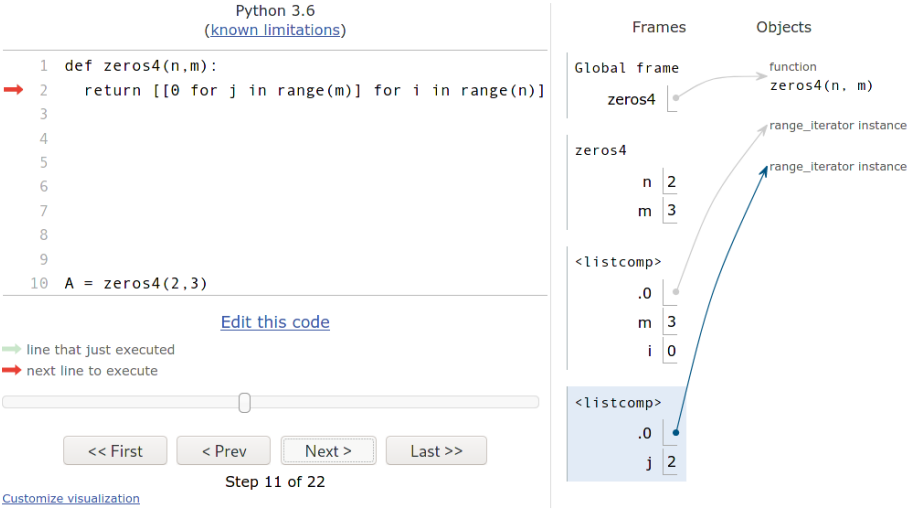
https://pythontutor.com/visualize.html#mode=edit

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

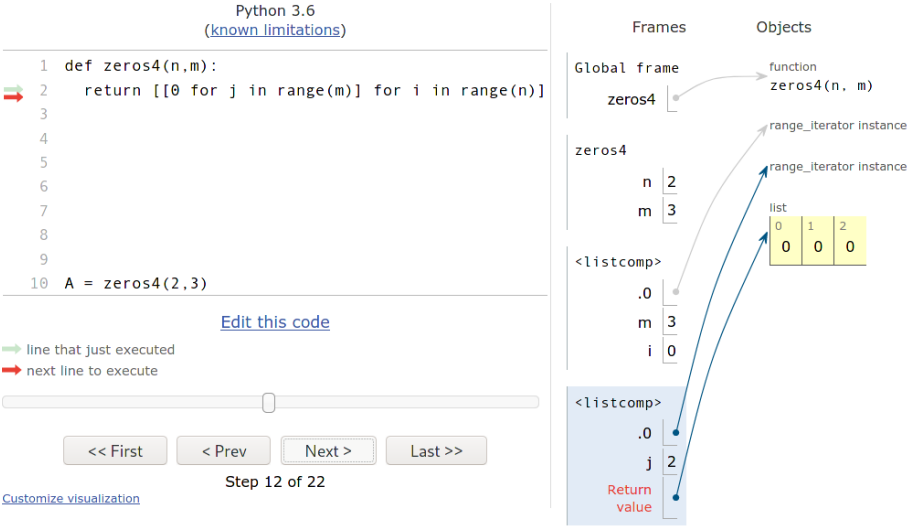
https://pythontutor.com/visualize.html#mode=edit



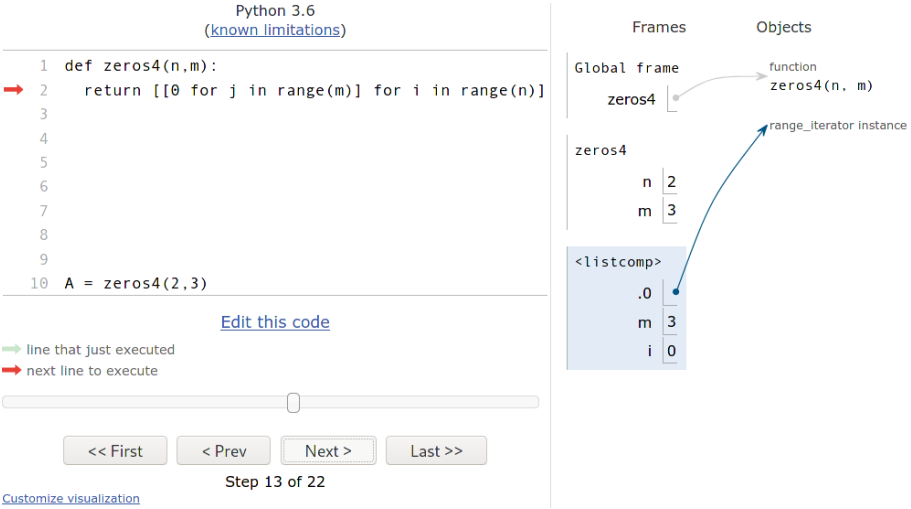
https://pythontutor.com/visualize.html#mode=edit



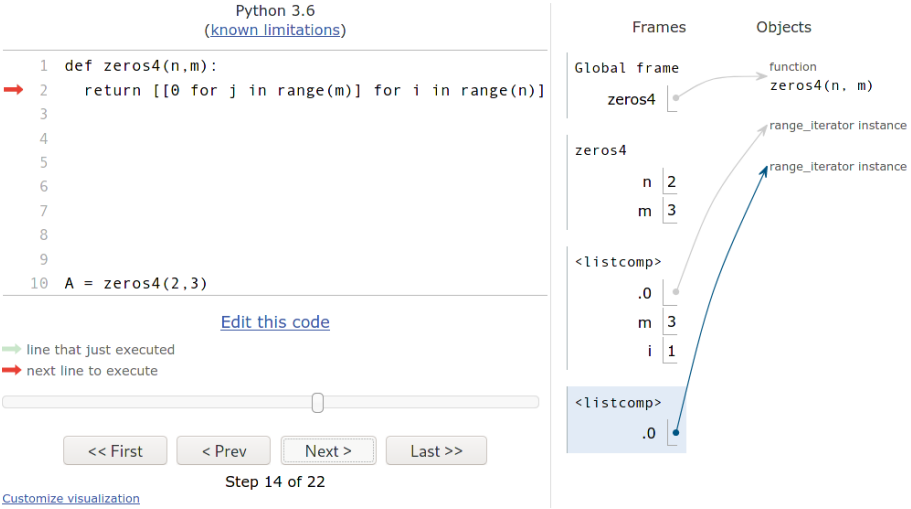
https://pythontutor.com/visualize.html#mode=edit



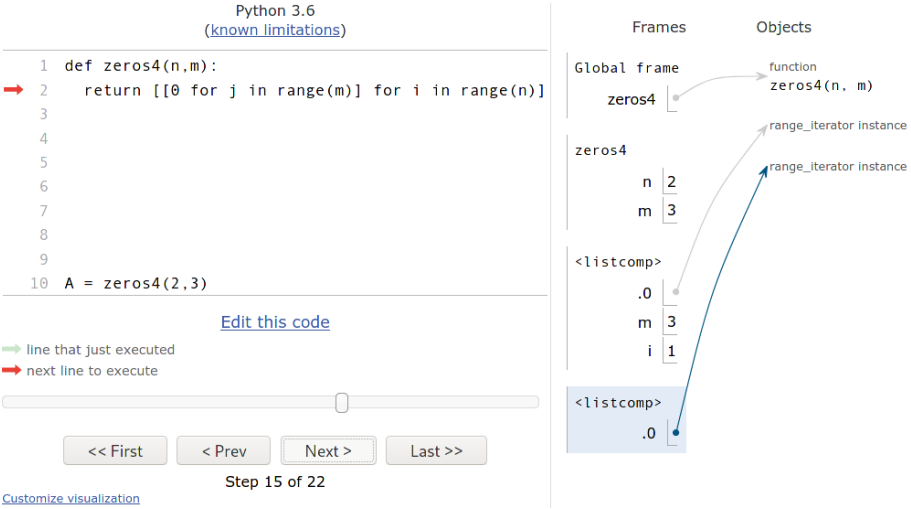
https://pythontutor.com/visualize.html#mode=edit



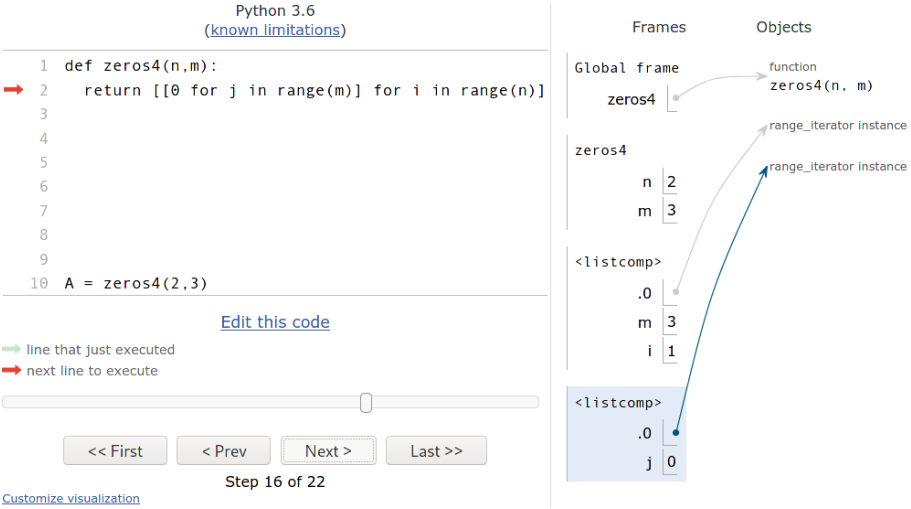
https://pythontutor.com/visualize.html#mode=edit



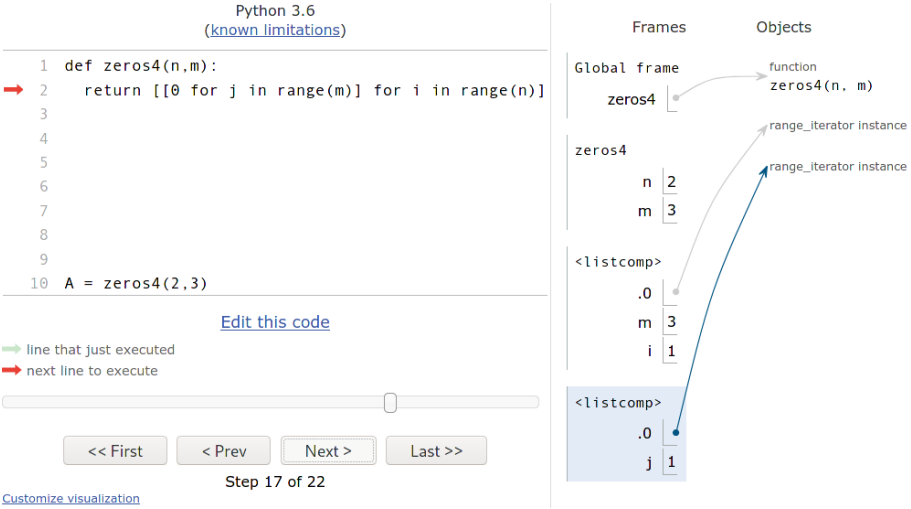
https://pythontutor.com/visualize.html#mode=edit



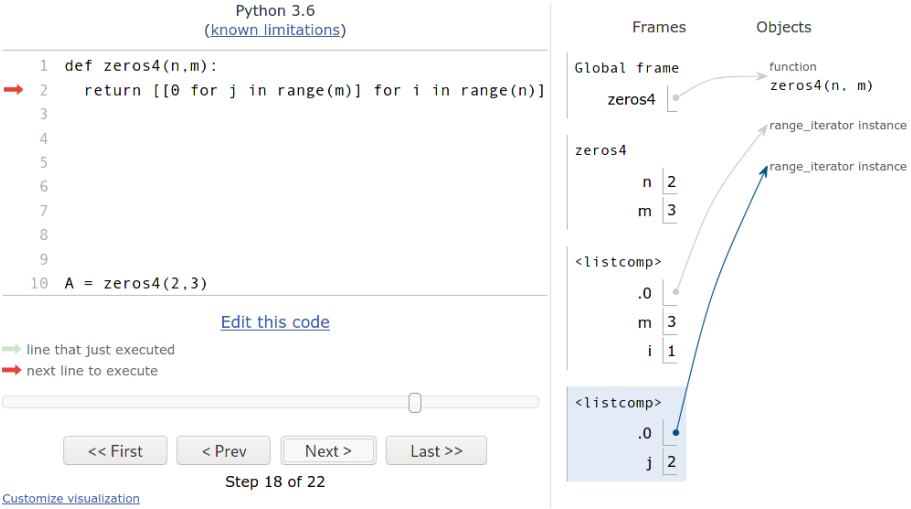
https://pythontutor.com/visualize.html#mode=edit



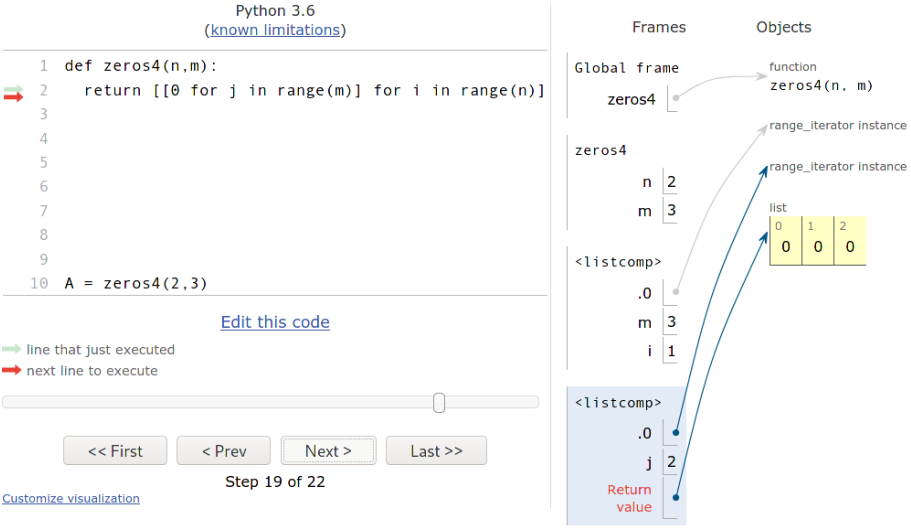
https://pythontutor.com/visualize.html#mode=edit



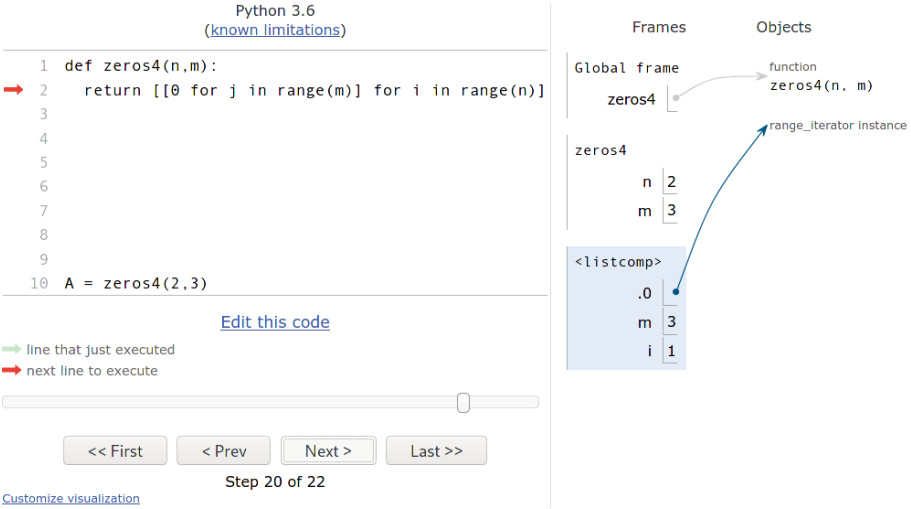
https://pythontutor.com/visualize.html#mode=edit



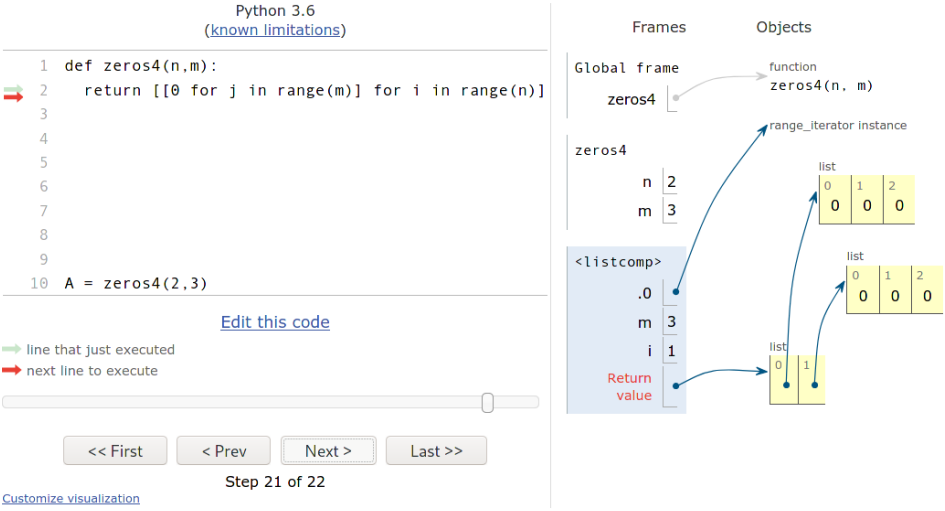
https://pythontutor.com/visualize.html#mode=edit



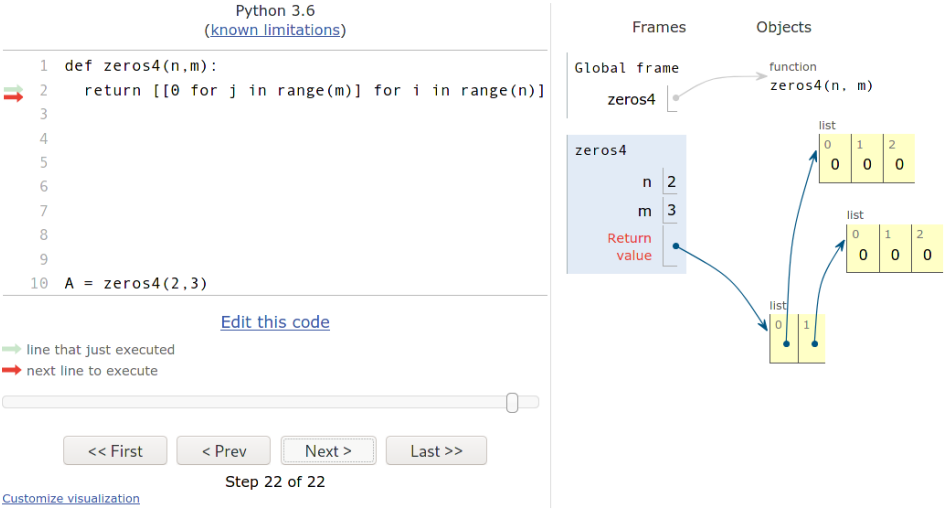
https://pythontutor.com/visualize.html#mode=edit



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 zeros4(n,m):
2    return [[0 for j in range(m)] for i in range(n)]
3
4
5
6
7
8
9
10 A = zeros4(2,3)
```

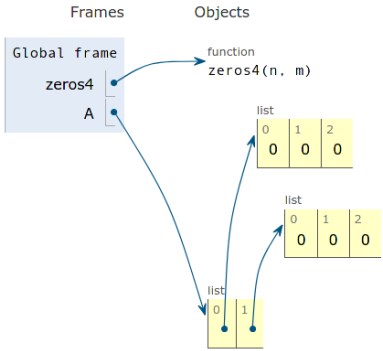
Edit this code

line that just executed
next line to execute

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

Done running (22 steps)

Customize visualization



# Exemplo 2

# Produto de duas matrices

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.

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

## Exemplos

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

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \times \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix} = \begin{bmatrix} 17 & 23 \\ 39 & 53 \\ 61 & 83 \end{bmatrix}$$

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

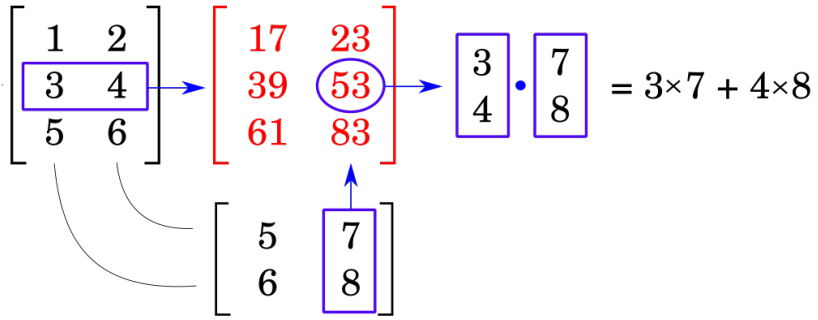
$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \times \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix} = \begin{bmatrix} 17 & 23 \\ 39 & 53 \\ 61 & 83 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \cdot \begin{bmatrix} 17 & 23 \\ 39 & 53 \\ 61 & 83 \end{bmatrix}$$

$$\begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix}$$

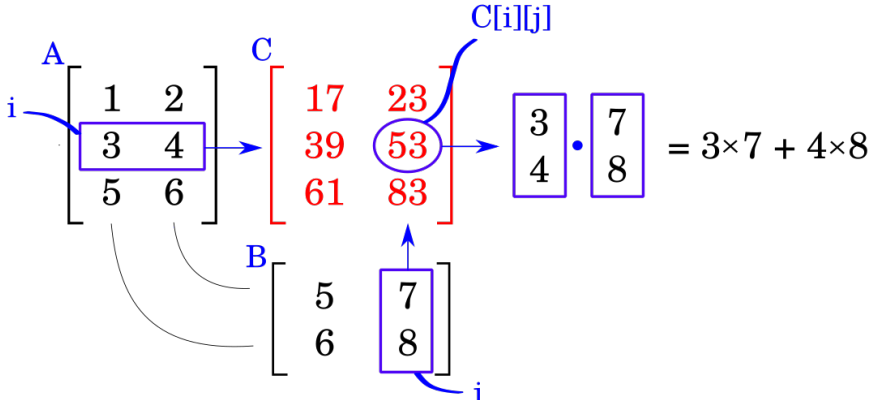
https://pythontutor.com/visualize.html#mode=edit

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \times \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix} = \begin{bmatrix} 17 & 23 \\ 39 & 53 \\ 61 & 83 \end{bmatrix}$$



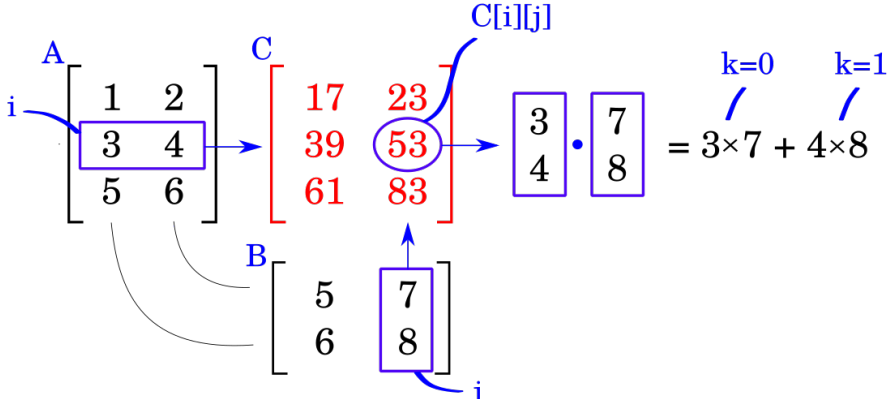
https://pythontutor.com/visualize.html#mode=edit

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \times \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix} = \begin{bmatrix} 17 & 23 \\ 39 & 53 \\ 61 & 83 \end{bmatrix}$$



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

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \times \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix} = \begin{bmatrix} 17 & 23 \\ 39 & 53 \\ 61 & 83 \end{bmatrix}$$



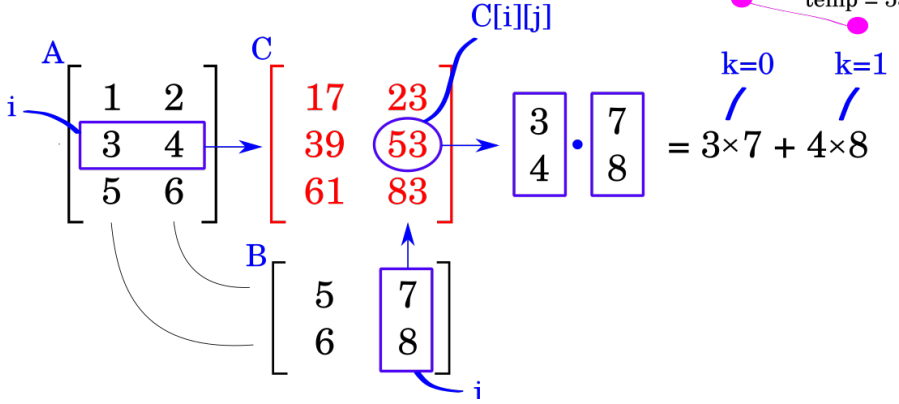
https://pythontutor.com/visualize.html#mode=edit

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \times \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix} = \begin{bmatrix} 17 & 23 \\ 39 & 53 \\ 61 & 83 \end{bmatrix}$$

temp = 0

temp = 21

temp = 53

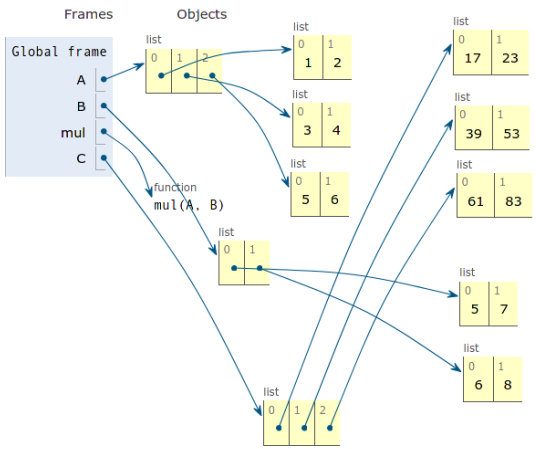


Estratégia 1:

Criar linha a linha e adicionar na matriz,
acumular produto escalar.

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

```
1  A,B = [[1,2],[3,4],[5,6]],[[5,7],[6,8]]
2
3  def mul(A,B):
4    if len(A[0]) != len(B):
5      return []
6    else:
7      C = []
8
9      for i in range(len(A)):
10         linha = []
11         for j in range(len(B[0])):
12             temp = 0
13             for k in range(len(B)):
14                 temp+=A[i][k]*B[k][j]
15             linha.append(temp)
16         C.append(linha)
17     return C
18
19 C = mul(A,B)
```

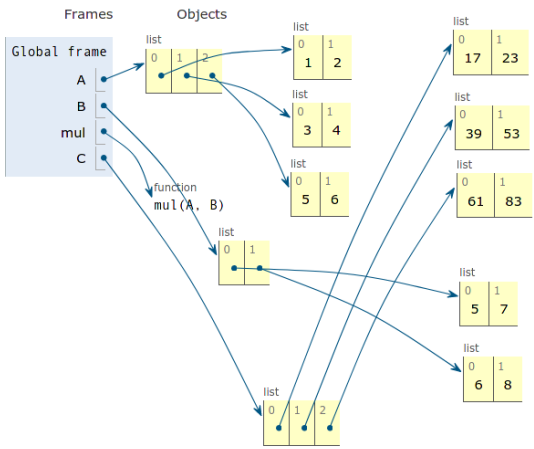



## Estratégia 1:

Criar linha a linha e adicionar na matriz, acumular produto escalar.

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

```
1  A,B = [[1,2],[3,4],[5,6]],[[5,7],[6,8]]
2
3  def mul(A,B):
4    if len(A[0]) != len(B):
5      return []
6    else:
7      C = []
8
9      for i in range(len(A)):
10         linha = []
11         for j in range(len(B[0])):
12             linha.append(0)
13             for k in range(len(B)):
14                 linha[j]+=A[i][k]*B[k][j]
15
16         C.append(linha)
17     return C
18
19 C = mul(A,B)
```

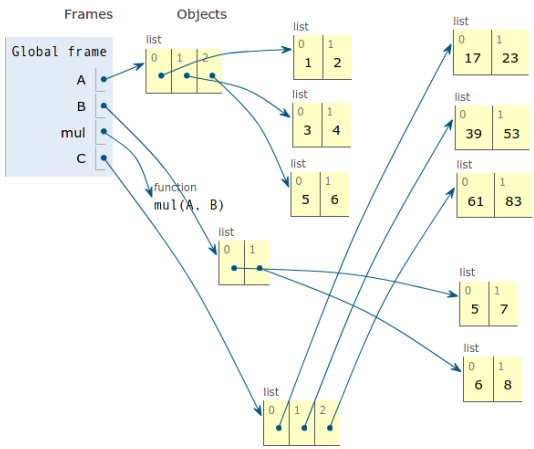



## Estratégia 2:

Criar linha a linha e adicionar na matriz, acumular produto escalar diretamente na linha.

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

```
1  A,B = [[1,2],[3,4],[5,6]],[[5,7],[6,8]]
2
3  def mul(A,B):
4    if len(A[0]) != len(B):
5      return []
6    else:
7      C = []
8
9      for i in range(len(A)):
10         C.append([])
11         for j in range(len(B[0])):
12             C[i].append(0)
13             for k in range(len(B)):
14                 C[i][j]+=A[i][k]*B[k][j]
15
16
17     return C
18
19 C = mul(A,B)
```

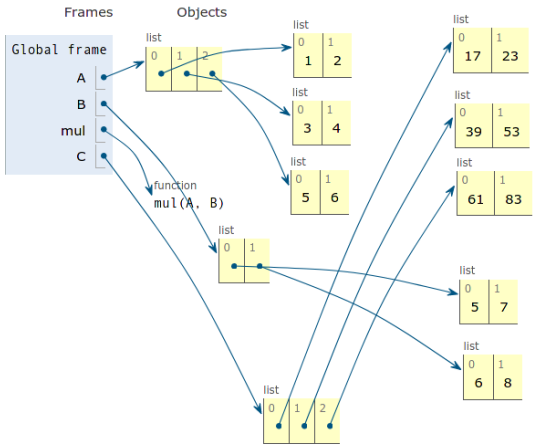



## Estratégia 3:

Criar linha a linha diretamente na matriz,
acumular produto escalar diretamente na linha na matriz.

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

```
1  A,B = [[1,2],[3,4],[5,6]],[[5,7],[6,8]]
2
3  def mul(A,B):
4    if len(A[0]) != len(B):
5      return []
6    else:
7      C = [[0 for j in range(len(B[0]))] for i in range(len(A))]
8
9      for i in range(len(A)):
10
11         for j in range(len(B[0])):
12
13             for k in range(len(B)):
14                 C[i][j]+=A[i][k]*B[k][j]
15
16
17     return C
18
19 C = mul(A,B)
```

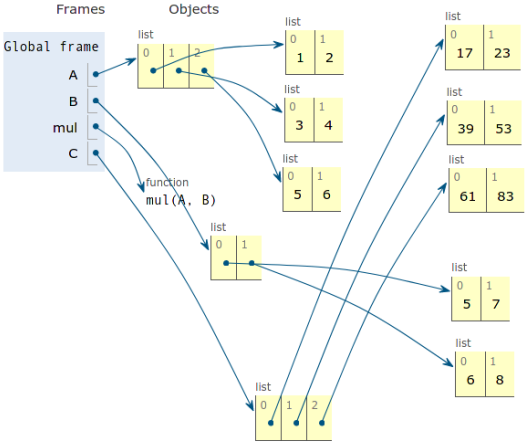



## Estratégia 4:

Inicializar a matriz com zeros,
acumular produto escalar diretamente na matriz.

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

```
1  A,B = [[1,2],[3,4],[5,6]],[[5,7],[6,8]]
2
3  def mul(A,B):
4    if len(A[0]) != len(B):
5      return []
6    else:
7      C = [[0 for j in range(len(B[0]))] for i in range(len(A))]
8
9      for i in range(len(A)):
10
11         for j in range(len(B[0])):
12
13
14             C[i][j]=sum([A[i][k]*B[k][j] for k in range(len(B))])
15
16
17     return C
18
19 C = mul(A,B)
```




## Estratégia 5:

Inicializar a matriz com zeros,
usar "sum"para calcular o produto escalar.

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

```
1  A,B = [[1,2],[3,4],[5,6]],[[5,7],[6,8]]
2
3  def mul(A,B):
4    if len(A[0]) != len(B):
5      return []
6    else:
7
8
9
10
11
12
13
14
15     C = [[ [sum([A[i][k]*B[k][j] for k in range(len(B))]) ] for j in range(len(B[0]))] for i in range(len(A))]
16
17     return C
18
19 C = mul(A,B)
```

## Estratégia 6:

Usando "List Comprehensions"
para criar de forma concisa "inline" toda a matriz

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

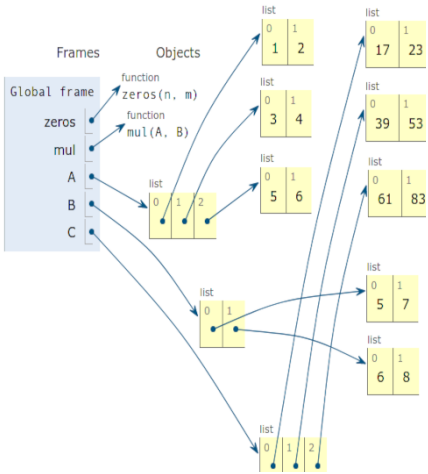
```
1  A,B = [[1,2],[3,4],[5,6]],[[5,7],[6,8]]
2
3  def mul(A,B):
4    if len(A[0]) != len(B):
5      return []
6    else:
7      return [[ [sum([A[i][k]*B[k][j] for k in range(len(B))]) ] for j in range(len(B[0]))] for i in range(len(A))]
8
9
10
11
12
13
14
15
16
17
18
19 C = mul(A,B)
```

## Estratégia 6:

Inicializar a matriz com zeros, construção inline.

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

```
1  def zeros(n,m):
2    return [[0 for j in range(m)] for i in range(n)]
3
4  def mul(A,B):
5    if len(A[0]) != len(B):
6      return []
7    else:
8      n,m = len(A), len(B[0])
9      C = zeros(n,m)
10
11     for i in range(n):
12         for j in range(m):
13             C[i][j]=sum([A[i][k]*B[k][j] for k in range(len(B))])
14
15     return C
16
17
18 A,B = [[1,2],[3,4],[5,6]],[[5,7],[6,8]]
19 C = mul(A,B)
```




## Estratégia 7:

Usando a funções para quebrar o problema.

Percorrer a matriz explicitamente.

# Exemplo 3

# Matriz Diagonal

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

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

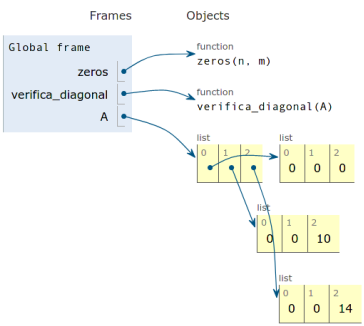
```
1  def zeros(n,m):
2    return [[0 for j in range(m)] for i in range(n)]
3
4  def verifica_diagonal(A):
5      for i in range(len(A)):
6          for j in range(len(A)):
7              if (i!=j) and (A[i][j]!=0):
8                  return False
9      return True
10
11 A = zeros(3,3)
12 A[2][2] = 14
13 print(verifica_diagonal(A))
14
15 A[1][2] = 10
16 print(verifica_diagonal(A))
```

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

Done running (80 steps)

Print output (drag lower right corner to resize)

```
True
False
```




Estratégia:

Percorrer a matriz, reconhecer um elemento não diagonal

# Exemplo 4

# Triangular Inferior

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

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

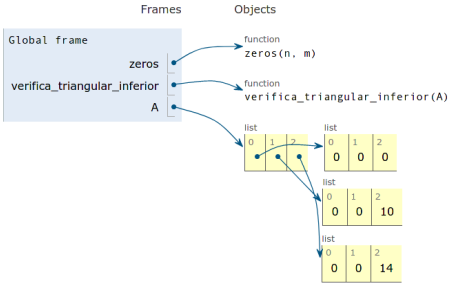
```
1  def zeros(n,m):
2    return [[0 for j in range(m)] for i in range(n)]
3
4  def verifica_triangular_inferior(A):
5      for i in range(len(A)):
6          for j in range(len(A)):
7              if (i<j) and (A[i][j]!=0):
8                  return False
9      return True
10
11 A = zeros(3,3)
12 A[2][2] = 14
13 print(verifica_triangular_inferior(A))
14
15 A[1][2] = 10
16 print(verifica_triangular_inferior(A))
```

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

Done running (80 steps)

Print output (drag lower right corner to resize)

```
True
False
```




Estratégia:

Percorrer a matriz, reconhecer um elemento acima da diagonal

# Exemplo 5

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*.

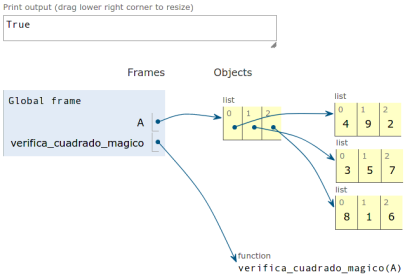
https://pythontutor.com/visualize.html#mode=edit

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

def verifica_cuadrado_magico(A):
    n = len(A)
    X = list(range(1,n*n+1))
    for i in range(n):
        for j in range(n):
            if A[i][j] in X:
                X.remove(A[i][j])
            else:
                return False
    soma = sum(A[0])
    for i in range(1,n):
        if soma != sum(A[i]):
            return False
    for j in range(n):
        if soma != sum([A[i][j] for i in range(n)]):
            return False
    if soma != sum([A[i][i] for i in range(n)]):
        return False
    if soma != sum([A[i][n-1-i] for i in range(n)]):
        return False

    return True

print(verifica_cuadrado_magico(A))
```




## Estratégia:

Reconhecer as somas a validar

# 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.