

Gestion du tas

Python 3.6
([known limitations](#))

```
→ 1 def allouer(x, y) :  
→ 2     t = [y] * x  
  3     u = [x] * y  
  4     return (t, u)  
  5  
  6 mon_tuple = allouer(2,3)  
  7  
  8 print(mon_tuple[0])  
  9  
10 print(mon_tuple[1])
```

[Edit this code](#)

→ line that just executed

→ next line to execute



<< First < Prev Next > Last >>

Step 4 of 9

[Customize visualization](#)

Print output (drag lower right corner to resize)

Frames

Objects

Global frame

allouer

function

allouer(x, y)

allouer

x | 2

y | 3

Gestion du tas

Python 3.6
([known limitations](#))

```
1 def allouer(x, y) :  
2     t = [y] * x  
→ 3     u = [x] * y  
→ 4     return (t, u)  
5  
6 mon_tuple = allouer(2,3)  
7  
8 print(mon_tuple[0])  
9  
10 print(mon_tuple[1])
```

[Edit this code](#)

→ line that just executed

→ next line to execute

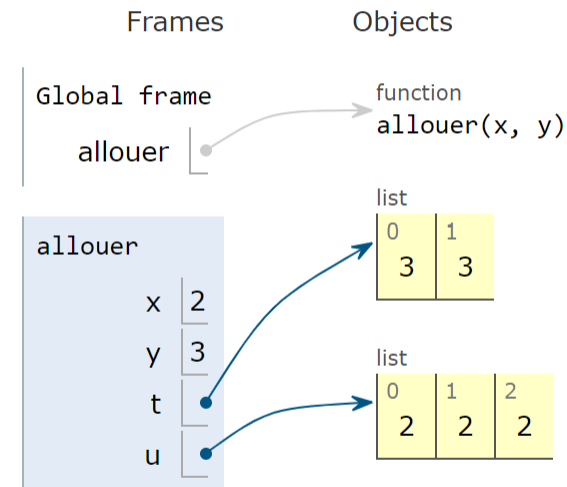


<< First < Prev Next > Last >>

Step 6 of 9

[Customize visualization](#)

Print output (drag lower right corner to resize)



Gestion du tas

Python 3.6
([known limitations](#))

```
1 def allouer(x, y) :  
2     t = [y] * x  
3     u = [x] * y  
➡ 4     return (t, u)  
5  
6 mon_tuple = allouer(2,3)  
7  
8 print(mon_tuple[0])  
9  
10 print(mon_tuple[1])
```

[Edit this code](#)

➡ line that just executed

➡ next line to execute

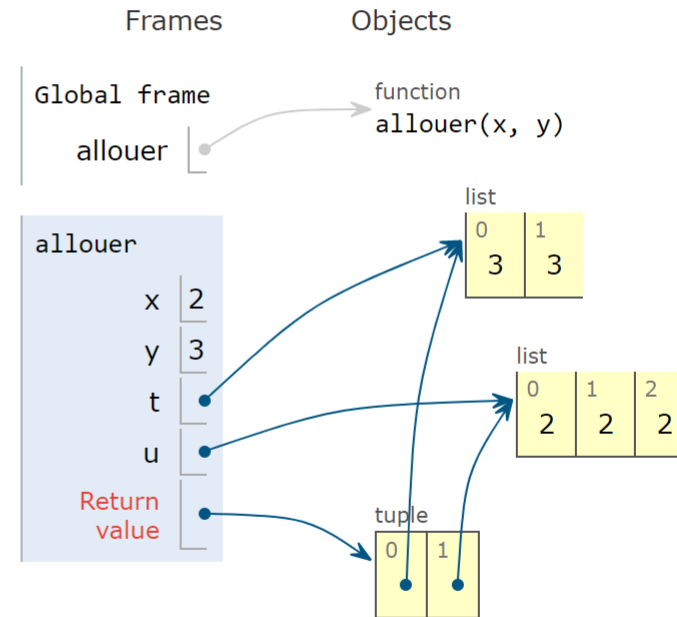


<< First < Prev Next > Last >>

Step 7 of 9

[Customize visualization](#)

Print output (drag lower right corner to resize)



Gestion du tas

Python 3.6
([known limitations](#))

```
1 def allouer(x, y) :  
2     t = [y] * x  
3     u = [x] * y  
4     return (t, u)  
5  
6 mon_tuple = allouer(2,3)  
7  
→ 8 print(mon_tuple[0])  
9  
→ 10 print(mon_tuple[1])
```

[Edit this code](#)

→ line that just executed

→ next line to execute



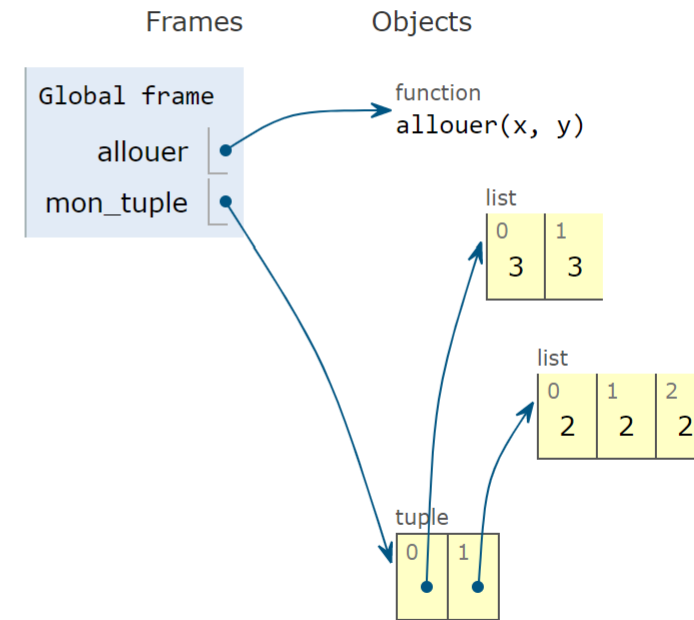
<< First < Prev Next > Last >>

Step 9 of 9

[Customize visualization](#)

Print output (drag lower right corner to resize)

[3, 3]



Gestion du tas

Python 3.6
([known limitations](#))

```
1 def allouer(x, y) :  
2     t = [y] * x  
3     u = [x] * y  
4     return (t, u)  
5  
6 mon_tuple = allouer(2,3)  
7  
8 print(mon_tuple[0])  
9  
10 print(mon_tuple[1])
```

```
11  
→ 12 mon_tuple[0].append(9)
```

[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)

```
[3, 3]  
[2, 2, 2]
```

Frames

Objects

Global frame

allouer

mon_tuple

function

allouer(x, y)

list

| 0 | 1 | 2 |
|---|---|---|
| 3 | 3 | 9 |

list

| 0 | 1 | 2 |
|---|---|---|
| 2 | 2 | 2 |

tuple

| 0 | 1 |
|------|------|
| list | list |