

Gestion du segment de pile

Python 3.6
([known limitations](#))

```
1 def g(x, y) :  
2     return 100 + y  
3  
→ 4 def f(x, y) :  
5     z = x + y  
6     u = g(x - 1, y * 2)  
7     return z + u  
8  
→ 9 f(1, 5)
```

[Edit this code](#)

→ line that just executed

→ next line to execute



<< First

< Prev

Next >

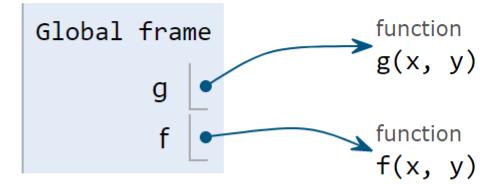
Last >>

Step 3 of 11

[Customize visualization](#)

Frames

Objects



Gestion du segment de pile

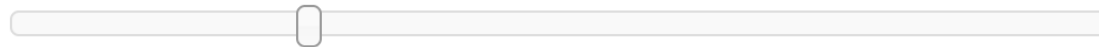
Python 3.6
([known limitations](#))

```
1 def g(x, y) :  
2     return 100 + y  
3  
→ 4 def f(x, y) :  
5     z = x + y  
6     u = g(x - 1, y * 2)  
7     return z + u  
8  
→ 9 f(1, 5)
```

[Edit this code](#)

→ line that just executed

→ next line to execute



<< First

< Prev

Next >

Last >>

Step 4 of 11

[Customize visualization](#)

Frames

Objects

Global frame

g

f

function

g(x, y)

function

f(x, y)

f

x

1

y

5

Gestion du segment de pile

Python 3.6
([known limitations](#))

```
1 def g(x, y) :  
2     return 100 + y  
3  
4 def f(x, y) :  
→ 5     z = x + y  
→ 6     u = g(x - 1, y * 2)  
7     return z + u  
8  
9 f(1, 5)
```

[Edit this code](#)

→ line that just executed
→ next line to execute

[Customize visualization](#)

Step 6 of 11

Frames Objects

Global frame

g → function g(x, y)

f → function f(x, y)

f

x	1
y	5
z	6

Gestion du segment de pile

Python 3.6
([known limitations](#))

```
→ 1 def g(x, y) :  
  2     return 100 + y  
  3  
  4 def f(x, y) :  
  5     z = x + y  
→ 6     u = g(x - 1, y * 2)  
  7     return z + u  
  8  
  9 f(1, 5)
```

[Edit this code](#)

→ line that just executed

→ next line to execute

<< First

< Prev

Next >

Last >>

Step 7 of 11

[Customize visualization](#)

Frames

Objects

Global frame

g

f

function

g(x, y)

function

f(x, y)

f

x | 1

y | 5

z | 6

g

x | 0

y | 10

Gestion du segment de pile

Python 3.6
([known limitations](#))

```
1 def g(x, y) :  
2     return 100 + y  
3  
4 def f(x, y) :  
5     z = x + y  
6     u = g(x - 1, y * 2)  
7     return z + u  
8  
9 f(1, 5)
```

[Edit this code](#)

→ line that just executed
→ next line to execute

Step 9 of 11

[Customize visualization](#)

Frames Objects

Global frame

g → function g(x, y)

f → function f(x, y)

f

x	1
y	5
z	6

g

x	0
y	10
Return value	110

Gestion du segment de pile

Python 3.6
([known limitations](#))

```
1 def g(x, y) :  
2     return 100 + y  
3  
4 def f(x, y) :  
5     z = x + y  
→ 6     u = g(x - 1, y * 2)  
→ 7     return z + u  
8  
9 f(1, 5)
```

[Edit this code](#)

→ line that just executed

→ next line to execute



<< First < Prev Next > Last >>

Step 10 of 11

[Customize visualization](#)

Frames

Objects

Global frame

g

f

function

g(x, y)

function

f(x, y)

f

x	1
y	5
z	6
u	110

Gestion du segment de pile

Python 3.6
([known limitations](#))

```
1 def g(x, y) :  
2     return 100 + y  
3  
4 def f(x, y) :  
5     z = x + y  
6     u = g(x - 1, y * 2)  
7     return z + u  
8  
9 f(1, 5)
```

[Edit this code](#)

→ line that just executed
→ next line to execute



<< First < Prev Next > Last >>

Step 11 of 11

[Customize visualization](#)

Frames

Objects

Global frame

g

f

function
g(x, y)

function
f(x, y)

f

x	1
y	5
z	6
u	110
Return value	116

Gestion du segment de pile

Python 3.6
([known limitations](#))

```
1 def g(x, y) :  
2     return 100 + y  
3  
4 def f(x, y) :  
5     z = x + y  
6     u = g(x - 1, y * 2)  
7     return z + u  
8  
→ 9 toto = f(1, 5)
```

[Edit this code](#)

→ line that just executed
→ next line to execute

<< First < Prev Next > Last >>

Done running (11 steps)

[Customize visualization](#)

