



# pf-6. 関数呼び出し, スコープ

(Python プログラミング入門演習, 全6回)

<https://www.kkaneko.jp/cc/pf/index.html>

金子邦彦



## 実習の指示

- 資料：3～14
- 次のことを理解しマスターする
  - 関数呼び出しにおけるジャンプ

# 実習



① ウェブブラウザを起動する

② Python Tutor を使いたないので、次の URL を開く

<http://www.pythontutor.com/>

※ Internet Explorer でうまく動かない場合がある

→ うまく動かないときは Google Chrome を試してください

※ 途中で「Server Busy . . .」というメッセージが出る  
ことがある。

→ 混雑している。少し（数秒から数十秒）待つと自動で表示が変わる（変わらない場合には、操作をもう一度行ってみる）

※ 日本語モードはない。英語で使う



### ③ 「Python Tutor」 をクリック

← → ↻ ⓘ 保護されていない通信 | pythontutor.com ☆ APP 2

## VISUALIZE CODE AND GET LIVE HELP

Learn Python, Java, C, C++, JavaScript, and Ruby

**Python Tutor** (created by [Philip Guo](#)) helps people overcome a fundamental barrier to learning programming: understanding what happens as the computer runs each line of code.

Write code in your web browser, see it visualized step by step, and get live help from volunteers.

Related services: [Java Tutor](#), [C Tutor](#), [C++ Tutor](#), [JavaScript Tutor](#), [Ruby Tutor](#)

**Over five million people in more than 180 countries** have used Python Tutor to visualize over 100 million pieces of code, often as a supplement to textbooks, lectures, and online tutorials.

[Visualize your code and get live help now](#)



Get live help!

Start private chat

These Python Tutor users are asking for help right now. Please volunteer to help!  
user\_041 from Singapore, Singapore needs help with Python3 - 2 people chatting - requested 12 minutes ago)  
user\_91c from Vélizy-Villacoublay, France needs help with Python3 - [click to help](#) minutes ago)  
user\_985 from Seattle, Washington, US needs help with Python3 - [click to help](#) (-

「Python 3.6」になっている

Write code in Python 3.6

```
1 |
```

エディタ

Help improve this tool by completing a [short user survey](#)

実行のためのボタン

Visualize Execution

Live Programming Mode

hide exited frames [default] inline primitives but don't nest objects [default]

draw pointers as arrows [default]



Write code in Python 3.6

```
1 def foo(a):  
2     return (a * 1.08)  
3  
4 p = 120  
5 print(foo(p))  
6 p = 200  
7 print(foo(p))  
8 |
```

「def foo(a)」の  
直後に「:」

字下げも正確に！  
「return (a \* 1.08)」の前に、  
「タブ」を1つだけ



Write code in Python 3.6

```
1 def foo(a):
2     return (a * 1.08)
3
4     p = 120
5     print(foo(p))
6     p = 200
7     print(foo(p))
8
```

これでは動かない

「delキー」などを使いながら書き換えてください

Write code in Python 3.6

```
1 def foo(a):
2     return (a * 1.08)
3
4 p = 120
5 print(foo(p))
6 p = 200
7 print(foo(p))
8
```

これは動く



## ② 「Visualize Execution」 をクリック

Write code in

```
1 def foo(a):
2     return (a * 1.08)
3
4 p = 120
5 print(foo(p))
6 p = 200
7 print(foo(p))
8
9
10
```

Help improve this tool by completing a [short user survey](#).  
Keep this tool free by making a [small donation](#) (PayPal, Patreon, c

Visualize Execution

Live Programming Mode





### ③ 「Last」 ボタンをクリック

Python 3.6

```
→ 1 def foo(a):  
2     return (a * 1.08)  
3  
4 p = 120  
5 print(foo(p))  
6 p = 200  
7 print(foo(p))
```

[Edit this code](#)

t has just executed  
e to execute

if code to set a breakpoint; use the Back and Forward buttons to jump there.

<< First

< Back

Step 1 of 11

Forward >

Last >>



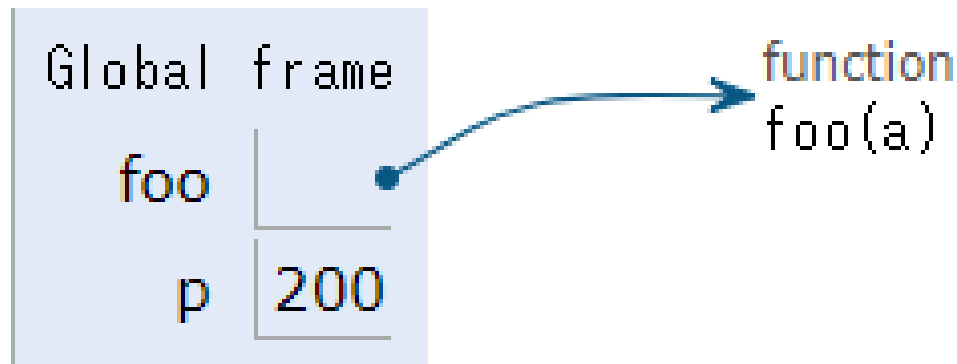
## ④ 表示を確認しなさい

Print output (drag lower right corner to resize)

```
129.600000000000002  
216.0
```

Frames

Objects





## ⑤ 「First」 をクリックして，最初の行に戻しなさい

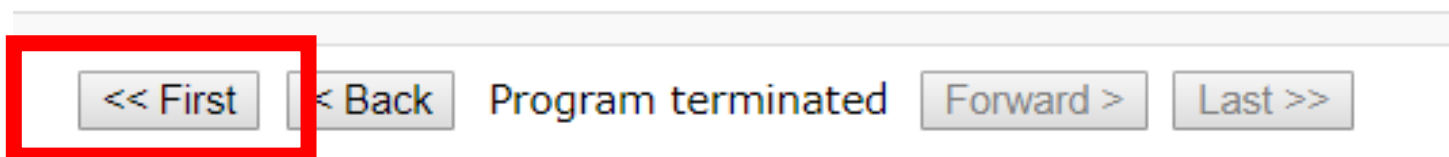
Python 3.6

```
1 def foo(a):  
2     return (a * 1.08)  
3  
4 p = 120  
5 print(foo(p))  
6 p = 200  
→ 7 print(foo(p))
```

[Edit this code](#)

⌘ that has just executed  
⌘ line to execute

line of code to set a breakpoint; use the Back and Forward buttons to jump there.





- ⑥ 「Step 1 of 11」と表示されているので、  
全部で、ステップ数は 11 あることが分かる  
(ステップ数と、プログラムの行数は違うもの)

Python 3.6

```
→ 1 def foo(a):  
2     return (a * 1.08)  
3  
4 p = 120  
5 print(foo(p))  
6 p = 200  
7 print(foo(p))
```

[Edit this code](#)

re that has just executed  
ext line to execute

a line of code to set a breakpoint; use the Back and Forward buttons to jump there.

<< First < Back **Step 1 of 11** Forward > Last >>



⑦ ステップ実行したいので、「Forward」をクリックしながら、緑の矢印の動きと、オブジェクト p, a の変化を確認しなさい。

※「Forward」ボタンを何度か押し、それ以上進めなくなったら終了

見どころ

Python 3.6

```
1 def foo(a):
2     return (a * 1.08)
3
4 p = 120
5 print(foo(p))
6 p = 200
7 print(foo(p))
```

[Edit this code](#)

that has just executed  
: line to execute

ne of code to set a breakpoint; use the Back and Forward buttons to jump there.

<< First < Back Step 6 of 11 Forward > last >>

Print output (drag lower right corner to resize)

Frames	Objects
Global frame	function foo(a)
foo	
p	120
foo	
a	120
Return value	129.6

見どころ



## ⑧ 終わったら「Edit this code」をクリックして、元の画面に戻りなさい

```
Python 3.6
-----
1 def foo(a):
2     return (a * 1.08)
3
4 p = 120
5 print(foo(p))
6 p = 200
→ 7 print(foo(p))
-----
Edit this code
```

Line that has just executed  
Next line to execute

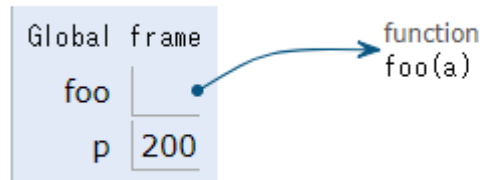
Click on a line of code to set a breakpoint; use the Back and Forward buttons to jump there.

Print output (drag lower right corner to resize)

```
129.60000000000002
216.0
```

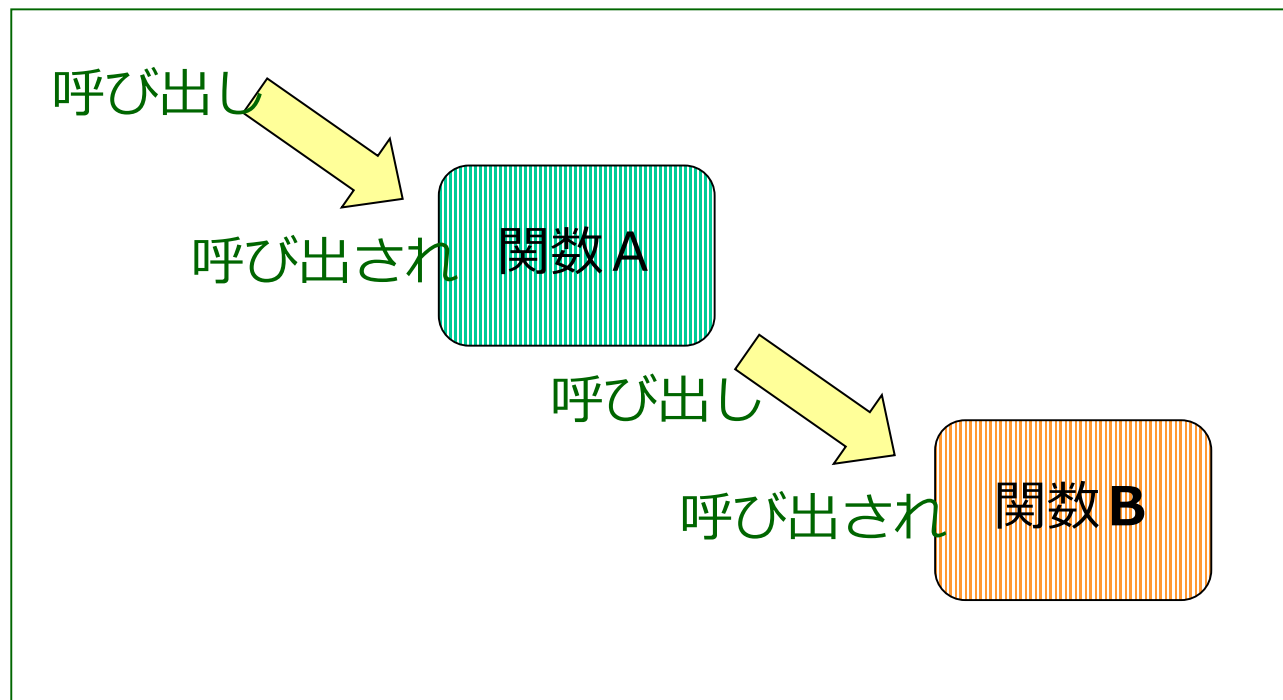
Frames

Objects



<< First   < Back   Program terminated   Forward >   Last >>

# 関数呼び出しの例



プログラムは、しばしば、複数の  
関数に「分割」される



- 関数の中で関数を呼び出す場合

```
def foo(a):  
    return a * 1.08  
def bar(x):  
    return foo(x) * 100  
  
p = 12  
print(bar(p))  
  
p = 20  
print(bar(p))
```



foo の呼び出し

bar の呼び出し





## 実習の指示

- 資料：18～26
- 次のことを理解しマスターする
  - 関数呼び出しにおけるジャンプ



# ① Python Tutor のエディタで次のように書きなさい

Write code in Python 3.6

```
1 def foo(a):
2     return a * 1.08
3 def bar(x):
4     return foo(x) * 100
5 p = 12
6 print(bar(p))
7 p = 20
8 print(bar(p))
9 |
```

「def foo(a)」と「def bar(x)」の直後に「:」

字下げも正確に。  
「タブ」を1つだけ



Write code in Python 3.6

```
1 def foo(a):
2     return a * 1.08
3     def bar(x):
4         return foo(x) * 100
5         p = 12
6         print(bar(p))
7         p = 20
8         print(bar(p))
9
```

これでは動かない

「delキー」などを使いながら書き換えてください

Write code in Python 3.6

```
1 def foo(a):
2     return a * 1.08
3 def bar(x):
4     return foo(x) * 100
5 p = 12
6 print(bar(p))
7 p = 20
8 print(bar(p))
9
```

これは動く



## ② 「Visualize Execution」 をクリック

Write code in Python 3.6

```
1 def foo(a):
2     return a * 1.08
3 def bar(x):
4     return foo(x) * 100
5 p = 12
6 print(bar(p))
7 p = 20
8 print(bar(p))
9
10
11
12 |
```

Help improve this tool by completing a [short user survey](#).

Keep this tool free by making a [small donation](#) (PayPal, Patreon, cre

Visualize Execution

Live Programming Mode



### ③ 「Last」 ボタンをクリック

Python 3.6

```
→ 1 def foo(a):  
2     return a * 1.08  
3 def bar(x):  
4     return foo(x) * 100  
5 p = 12  
6 print(bar(p))  
7 p = 20  
8 print(bar(p))
```

[Edit this code](#)

at has just executed

ne to execute

of code to set a breakpoint; use the Back and Forward buttons to jump there.

<< First

< Back

Step 1 of 18

Forward >

Last >>

... @code... Comment with a small description



## ④ 表示を確認しなさい。

Python 3.6

```
1 def foo(a):  
2     return a * 1.08  
3 def bar(x):  
4     return foo(x) * 100  
5 p = 12  
6 print(bar(p))  
7 p = 20  
→ 8 print(bar(p))
```

[Edit this code](#)

cutted

breakpoint; use the Back and Forward buttons to jump there.



[Back](#) Program terminated [Forward >](#) [Last >>](#)

Support with a [small donation](#).

Print output (drag lower right corner to resize)

```
1296.0  
2160.0
```

Frames

Objects

Global frame

foo

bar

p

20

function

foo(a)

function

bar(x)



## ⑤ 「First」 をクリックして，最初の行に戻しなさい

Python 3.6

```
1 def foo(a):  
2     return a * 1.08  
3 def bar(x):  
4     return foo(x) * 100  
5 p = 12  
6 print(bar(p))  
7 p = 20  
→ 8 print(bar(p))
```

[Edit this code](#)

e that has just executed  
xt line to execute

line of code to set a breakpoint; use the Back and Forward buttons to jump there.

<< First

Back

Program terminated

Forward >

Last >>



- ⑥ 「Step 1 of 18」と表示されているので、  
全部で、ステップ数は 18 あることが分かる  
(ステップ数と、プログラムの行数は違うもの)

Python 3.6

```
→ 1 def foo(a):  
2     return a * 1.08  
3 def bar(x):  
4     return foo(x) * 100  
5 p = 12  
6 print(bar(p))  
7 p = 20  
8 print(bar(p))
```

[Edit this code](#)

ie that has just executed  
xt line to execute

line of code to set a breakpoint; use the Back and Forward buttons to jump there.

<< First < Back **Step 1 of 18** Forward > Last >>





⑦ ステップ実行したいので、「Forward」をクリックしながら、緑の矢印の動きと、オブジェクト p, x, a の変化を確認しなさい。

※ 「Forward」ボタンを何度か押し、それ以上進めなくなったら終了

見どころ

Python 3.6

```
1 def foo(a):
2     return a * 1.08
3 → def bar(x):
4     return foo(x) * 100
5 p = 12
6 → print(bar(p))
7 p = 20
8 print(bar(p))
```

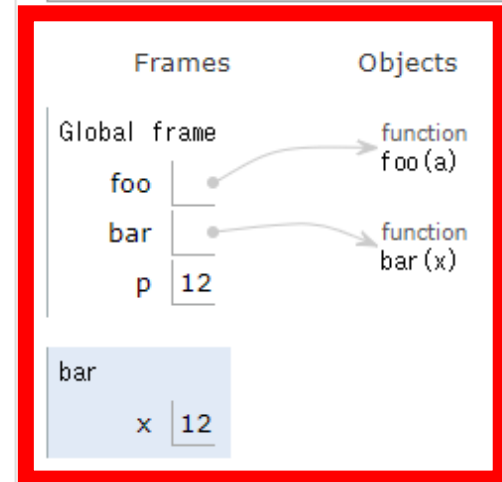
[Edit this code](#)

that has just executed  
line to execute

ne of code to set a breakpoint; use the Back and Forward buttons to jump there.

<< First < Back Step 5 of 13 Forward > Last >>

Print output (drag lower right corner to resize)



見どころ 25



## ⑧ 終わったら「Edit this code」をクリックして、元の画面に戻りなさい

```
Python 3.6
```

---

```
1 def foo(a):  
2     return a * 1.08  
3 def bar(x):  
4     return foo(x) * 100  
5 p = 12  
6 print(bar(p))  
7 p = 20  
→ 8 print(bar(p))
```

---

[Edit this code](#)

at has just executed  
ine to execute

: of code to set a breakpoint; use the Back and Forward buttons to jump there.

Progress bar: [-----○-----]

<< First   < Back   Program terminated   Forward >   Last >>

Print output (drag lower right corner to resize)

```
1296.0  
2160.0
```

