

Python Tutor での Python プログラム実行

(人工知能, プログラミング等のオンラインサービス)

URL: <https://www.kkaneko.jp/pro/online/index.html>

金子邦彦



Python 言語の特徴



- 「入門者に学習しやすい」とされる
- **多数の拡張機能**（外部プログラムのインポートによる）
- **Python 言語システムのソースコードは公開**されている
- **オブジェクト指向**の機能を持つ
オブジェクト, クラス, メソッド, 属性, クラス階層, 継承

- プログラミング学習を行える**オンラインサービス**

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

- Web ブラウザを使う

- たくさんの言語を扱うことができる

Python, Java, C, C++, JavaScript, Ruby など

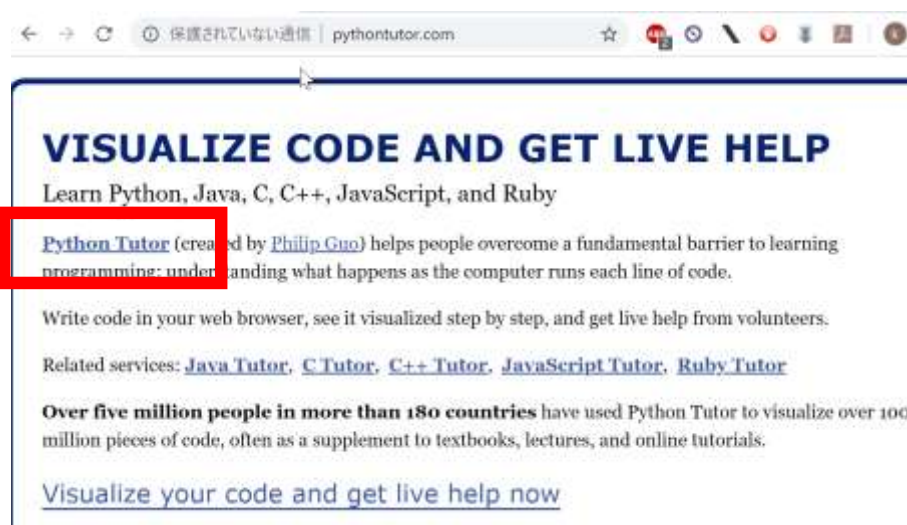
Python Tutor の起動



① ウェブブラウザを起動する

② Python Tutor を使いたいのので, 次の URL を開く
<http://www.pythontutor.com/>

③ 「Python Tutor」をクリック ⇒ 編集画面が開く



Python Tutor の編集画面



Python debugger - [pdb](#) interface to Python Tutor - Learn Python by visualizing code (also debug [JavaScript](#), [Java](#), [C](#), and [C++](#) code)

Write code in Python 3.6 「Python 3.6」になっている

エディタ

(プログラムを書き換えることができる)

Visualize Execution

実行のためのボタン

hide exited frames [default] ▾

inline primitives, don't nest objects [default] ▾

draw pointers as arrows [default] ▾

[Show code examples](#)

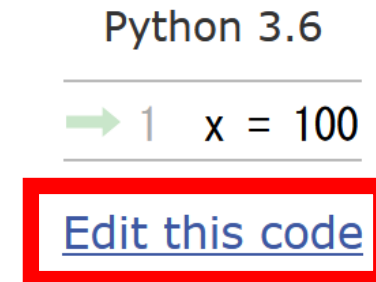
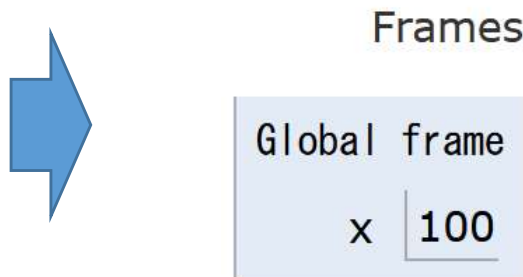
Generate permanent link

Python Tutor でのプログラム実行手順



(1) 「**Visualize Execution**」をクリックして**実行画面**に切り替える

(2) 「**Last**」をクリック。



(3) 実行結果を確認する。

(4) 「**Edit this code**」をクリックして**編集画面**に戻る

Python Tutor 使用上の注意点①



- 実行画面で、次のような**赤の表示**が出ることがある →
無視してよい

過去の文法ミスに関する確認表示
邪魔なときは「Close」

Java Tutor - Visualize Java code execution to learn Java online (also visualize [Python2](#), [Python3](#), [Java](#), [JavaScript](#), [TypeScript](#), [Ruby](#), [C](#), and [C++](#) code)

Get live help for free in the [Python tutoring Discord](#) chat room

Java 8
(known limitations)

```
1 public class YourClassNameHere {  
2     public static void main(String[] args) {  
→ 3         int x = 100;  
4         int y = 200;  
5         System.out.printf("%d\n", x + y);  
6     }  
7 }
```

Edit this code

Print output (drag lower right corner to resize)

Frames Objects

You just fixed the following error:

```
1 public class YourClassNameHere {  
2     public static void main(String[] args) {  
3         int x = 100;  
4         int y = 200;  
5         System.put.printf("%d\n", x + y);  
6     }  
7 }
```

Error: cannot find symbol
symbol: variable put
location: class java.lang.System

Please help us improve this tool with your feedback.
What misunderstanding do you think caused this error?

Submit Close Hide all of these pop-ups

Step 1 of 5

Customize visualization (NEW!)

Python Tutor 使用上の注意点②



「please wait ... executing」のとき，10秒ほど待つ。

```
Write code in Java 8
1 public class YourClassNameHere {
2     public static void main(String[] args) {
3         int x = 100;
4         int y = 200;
5         System.out.printf("%d\n", x * y);
6     }
7 }
```

Please wait ... executing (takes up to 10 seconds)

→ 混雑しているときは，「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）



① Python Tutor のエディタで次のプログラムを入れる

print(100 * 200)

Write code in Python 3.6

```
1 print(100 * 200)
```

```
2
```

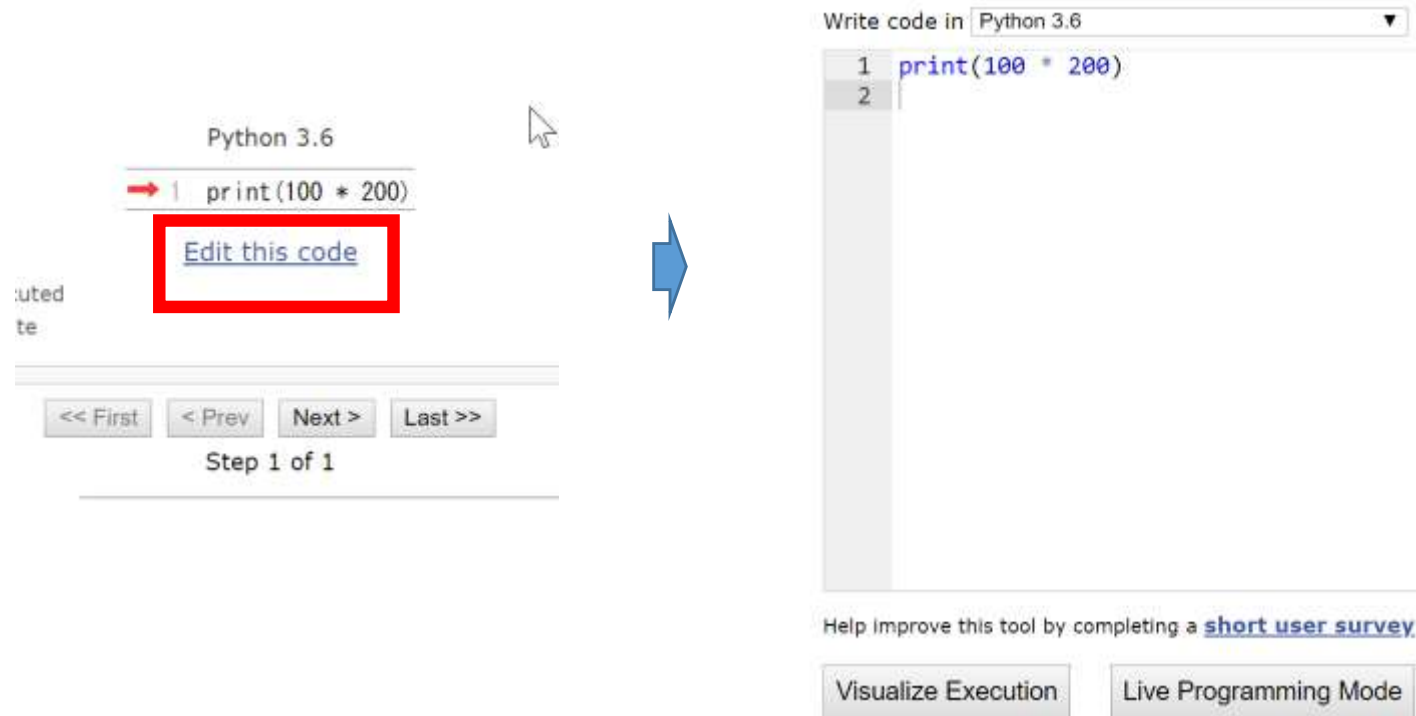
すべて 半角文字

「*」は掛け算の記号

② 「**Visual Execution**」をクリック. そして
「**Last**」をクリック. 結果を確認

The screenshot shows the Python Tutor interface. On the left, a code editor displays the code `1 print(100 * 200)`. Below the editor, a button labeled "Visualize Execution" is highlighted with a red box. In the center, a panel shows the code being executed, with a red arrow pointing to the first line. Below this, a row of navigation buttons includes "<< First", "< Prev", "Next >", and "Last >>". The "Last >>" button is highlighted with a red box. At the bottom of this panel, it says "Step 1 of 1". On the right, a panel titled "Print output (drag)" shows the output "20000" inside a red box.

③ 「**Edit this code**」 をクリックすると, エディタの画面に戻る



The image shows a transition in the Replit web interface. On the left, a small preview of the code execution environment is shown. It displays the code `print(100 * 200)` and a red box highlights the [Edit this code](#) link. Below the code, there are navigation buttons: `<< First`, `< Prev`, `Next >`, and `Last >>`, along with the text `Step 1 of 1`. A blue arrow points from this preview to the right, where a larger screenshot of the code editor is shown. The editor has a dropdown menu set to `Python 3.6` and contains the code `1 print(100 * 200)` on the first line. At the bottom of the editor, there is a link to [Help improve this tool by completing a short user survey](#) and two buttons: `Visualize Execution` and `Live Programming Mode`.



④ Python Tutor のエディタで次のプログラムを入れる

x = 100

Write code in Python 3.6

```
1 x = 100
```

すべて 半角文字

⑤ 「**Visual Execution**」をクリック. そして
「**Last**」をクリック. 結果を確認

Write code in Python 3.6

```
1 x = 100
```

Python 3.6

```
→ 1 x = 100
```

[Edit this code](#)

executed
recute

<< First < Prev Next > **Last >>**

Step 1 of 1

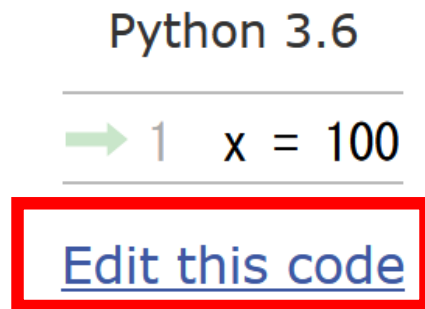
Frames

Global frame

x	100
---	-----

Visualize Execution

⑥ 「**Edit this code**」 をクリックすると、エディタの画面に戻る



ステップ実行

プログラム実行



- **命令（コマンド）** は、順番に並んでいる
- 通常は、上から順に1つずつ実行される（**逐次実行**）
- **条件分岐**や**繰り返し（ループ）** では、逐次実行とは違う実行の流れになる

※ プログラミング言語の種類によっては、順番に並んでいないという場合もある： SQL, Scheme, Prolog など

ステップ実行



- ステップ実行により, プログラム実行の流れをビジュアルに観察

Python Tutor でのステップ実行



ステップ実行により, プログラム実行の流れを確認できる

The screenshot displays the Python Tutor interface for Python 3.6. The code editor on the left shows a program with five lines. A red box highlights the first four lines, and a green arrow points to line 2. A red arrow points to line 5, indicating the current step in the execution. Below the code editor is a link that says "Edit this code". To the right of the code editor is a panel titled "Print output (drag lower right)" which contains a text area for output and a section titled "Frames" showing the "Global frame" with the variable "age" set to 30. At the bottom of the interface is a navigation bar with buttons for "<< First", "< Prev", "Next >", and "Last >>". A slider bar is positioned above these buttons, and the text "Step 3 of 3" is displayed below the buttons.

```
Python 3.6
```

```
1 age = 30  
2 if age <= 12:  
3     print(500)  
4 else:  
5     print(1200)
```

[Edit this code](#)

Print output (drag lower right)

Frames

Global frame

age 30

<< First < Prev Next > Last >>

Step 3 of 3



① まずは、「Visualize Execution」をクリックして、実行開始

Write code in Python 3.6 ▼

```
1 age = 30
2 if age <= 12:
3     print(500)
4 else:
5     print(1200)
6
```

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

Visualize Execution

Live Programming Mode



すでに、プログラムを実行中だったときは、「**First**」をクリックして、最初の行に戻すことができる

Python 3.6

```
1 age = 30
2 if age <= 12:
3     print(500)
4 else:
→ 5     print(1200)
```

[Edit this code](#)

ited
e

<< First < Prev Next > Last >>

Done running (3 steps)



- ② 「**Step 1 of 3**」と表示されているので、
全部で、**ステップ数**は**3**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 age = 30
→ 2 if age <= 12:
3     print(500)
4 else:
→ 5     print(1200)
```

[Edit this code](#)

uted
:e

<< First < Prev Next > Last >>

Step 3 of 3



③ **ステップ実行**したいので、「Next」をクリックしながら、矢印の動きを確認。

※「Next」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
2行目から 5行目へ
ジャンプするところ

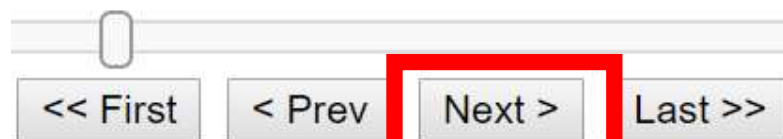


Python 3.6

```
→ 1 age = 30
→ 2 if age <= 12:
3     print(500)
4 else:
5     print(1200)
```

[Edit this code](#)

ed



Step 2 of 3

④ 最後の行まで達したら, 「**Edit this code**」 をクリックして, エディタの画面に戻る



Python 3.6

```
1 age = 30
2 if age <= 12:
3     print(500)
4 else:
5     print(1200)
```

[Edit this code](#)

ited
e

<< First

< Prev

Next >

Last >>

Done running (3 steps)



⑤ 次のように「**age = 10**」と書き換えて、いままでと同じことを行う

見どころ
3行目で実行が止まる

Python 3.6

```
1 age = 10
2 if age <= 12:
→ 3     print(500)
4 else:
5     print(1200)
```

[Edit this code](#)

uted
te

<< First

< Prev

Next >

Last >>

Done running (3 steps)