

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

A screenshot of a web browser showing the Python Tutor website. The browser's address bar displays "pythontutor.com". The main heading is "VISUALIZE CODE AND GET LIVE HELP". Below it, the text reads "Learn Python, Java, C, C++, JavaScript, and Ruby". A red box highlights the text "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." Below this, there is a link "Visualize your code and get live help now" which is underlined and blue. Other text on the page includes "Write code in your web browser, see it visualized step by step, and get live help from volunteers." and "Related services: Java Tutor, C Tutor, C++ Tutor, JavaScript Tutor, Ruby Tutor". At the bottom, it states "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."/>

← → ↻ 保護されていない通信 | pythontutor.com ☆ 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](#)

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」になっている

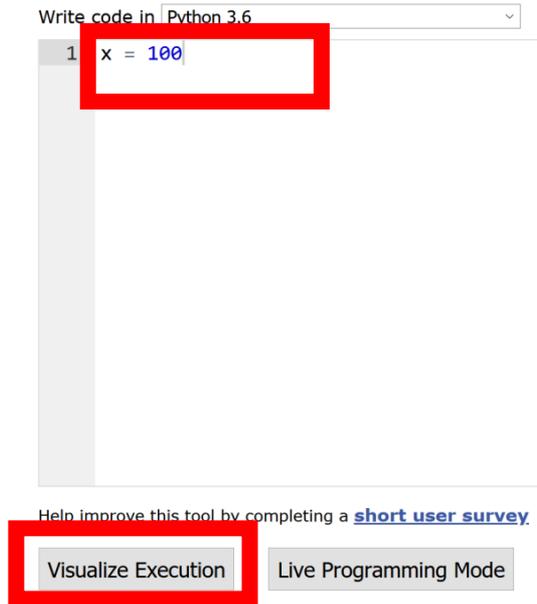
1 |

エディタ
(プログラムを書き換えることができる)

実行のためのボタン

[Show code examples](#)

Python Tutor でのプログラム実行手順

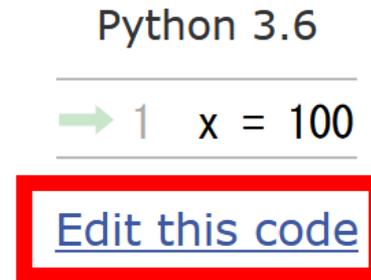


(1) 「**Visualize Execution**」をクリックして**実行画面**に切り替える

(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」

The screenshot shows the Python Tutor interface. On the left, a Java code snippet is displayed with line numbers 1 through 7. Line 3 is highlighted with a red arrow, indicating the next line to execute. The code is as follows:

```
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 }
```

Below the code is a link to "Edit this code".

On the right, the "Print output" area is empty. Below it, the "Frames" and "Objects" sections are visible. A red-bordered error message box is displayed, containing the following text:

You just fixed the following error:

```
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 }
```

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?

Buttons for "Submit", "Close", and "Hide all of these pop-ups" are visible at the bottom of the error message box.

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, the code editor contains the code `print(100 * 200)`. Below the editor, the **Visualize Execution** button is highlighted with a red box. In the center, the execution progress is shown as `1 print(100 * 200)` with a red arrow pointing to the first line. Below this, the **Last >>** button is highlighted with a red box. On the right, the **Print output (drag)** window displays the result `20000`, which is also highlighted with a red box. The interface includes navigation buttons: `<< First`, `< Prev`, `Next >`, and `Last >>`. The text `Step 1 of 1` is visible at the bottom.

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





④ Python Tutor のエディタで次のプログラムを入れる

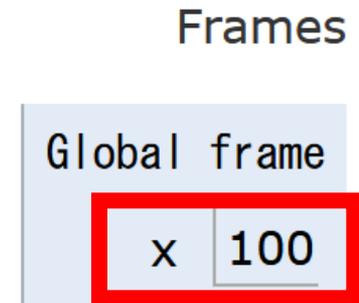
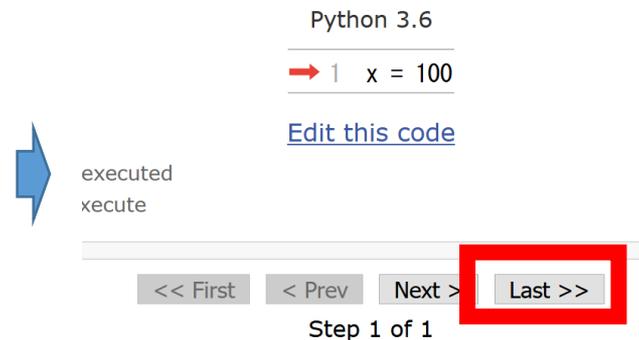
x = 100

Write code in Python 3.6

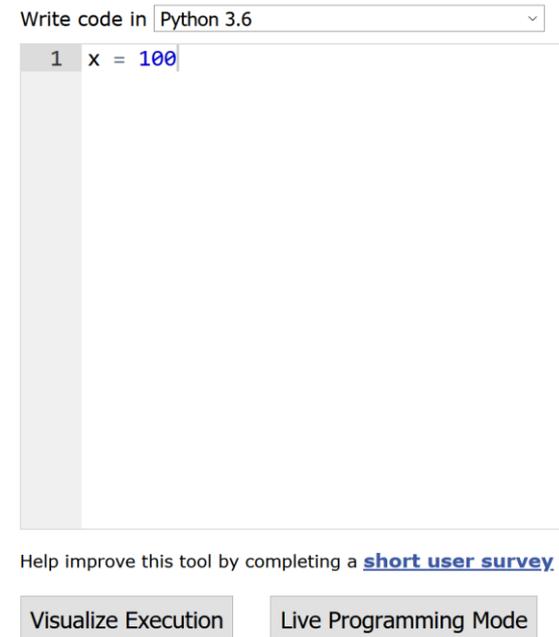
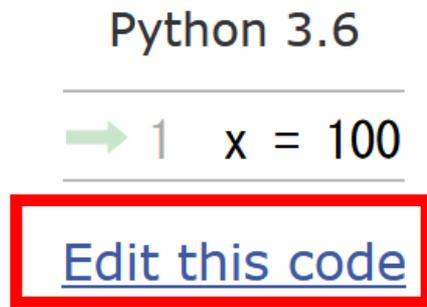
```
1 x = 100
```

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



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 shows the following code:

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

The code is highlighted with a red box. A green arrow points to line 2, and a red arrow points to line 5. Below the code is a link that says "Edit this code".

To the right of the code editor is a "Print output (drag lower right)" panel, which is currently empty. Below this panel is a "Frames" panel showing the "Global frame" with the variable "age" set to 30.

At the bottom of the interface is a control bar with a progress slider and four buttons: "<< First", "< Prev", "Next >", and "Last >>". The "Next >" button is highlighted with a red box, and a mouse cursor is hovering over it. Below the buttons, it says "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、最初の一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



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**)

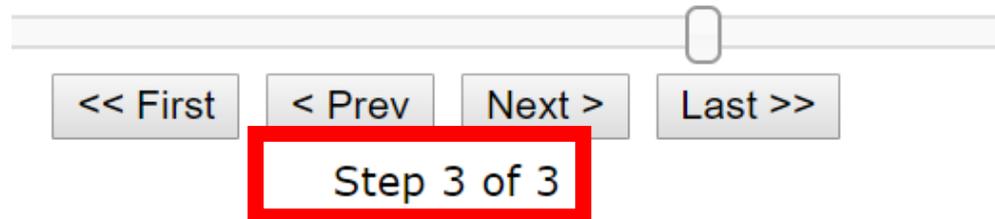
Python 3.6

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

[Edit this code](#)

uted

ie



A horizontal progress bar with a slider knob is shown above a set of navigation buttons. The buttons are: '<< First', '< Prev', 'Next >', and 'Last >>'. Below these buttons, the text 'Step 3 of 3' is displayed and highlighted with a red rectangular border.

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

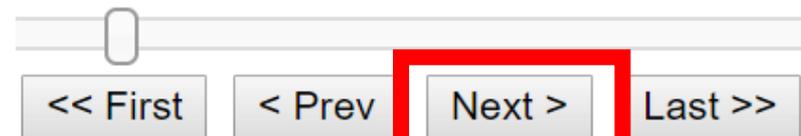
見どころ
2行目から 5行目へ
ジャンプするところ



```
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)