第56回 プログラミング･シンポジウム 2015.1

# 視覚化と穴埋め問題を組み合わせた初学者のためのプログラミング学習ツールの提案

大泉 良介[1,a)] 岩崎 英哉[1,b)]

**概要**：本稿では，プログラムの穴埋め問題とプログラム実行の視覚化を組み合わせた，初学者向けのプログラミング学習ツールである拡張 Online Python Tutor（拡張 OPT) を提案する．拡張 OPT は，web 上で動作し，プログラムの穴埋め問題を，プログラム実行の視覚化とともに出題する．学習者は視覚化されたプログラム実行の過程を確認し，その実行過程と合致するように穴埋め問題に解答する．誤答の場合には，誤答プログラムによる実行過程を正答プログラムの実行過程と並べて視覚化して提示する．これにより，正答と誤答の振る舞いの違いを明確にし，学習者の理解を促す．学習者はこれを繰り返すことによって，プログラミングに関する理解を深めることができる．穴埋めの正解判定は，正解プログラムと学習者からの解答のプログラムの実行トレースを比較することで行う．この方法により，正解が唯一に定まるとは限らないという問題に対処した．

**キーワード**：プログラミング学習，Python，視覚化，穴埋め問題

## 1. はじめに

初等プログラミング学習において初学者に求められるのは，変数の値の変化やループ処理などのプログラムの実行過程を理解することである．プログラムの実行を，例えば，ソースコードをなぞる，あるいは比喩を用いて概念的，抽象的に解説されることがある．しかし機械上での動作や，学習者にとって慣れない比喩表現の想像は，初学者にとって難しい．その結果としてプログラミングの本質を理解出来ず，挫折してしまう学習者を今まで見てきた．

このような初学者に対し，より直接的に実機上での実行過程を理解できるようにプログラムの視覚化を行うツール [2] [3] の開発が行われてきた．これにより，実行状態への理解力を高めることはできるものの，ソースコードを読むことは出来ても目的のプログラムを実装することが出来ない学習者が出てきている．

初学者向けの学習方法として，ソースコードに対する穴埋め問題を用いるものがある．しかし，従来のシステムは，プログラムの実行によらず，文字列の比較で答え合わせをするものがほとんどである [3][10]．このようなシステムにおいて学習者は，プログラムの字面および実行結果だけから穴埋め問題を行わなければならない．そのため，特に初学者や新しい単元の学習を行う人にとってハードルが高く，学習者の意欲をそぐことになりかねない．さらに誤答の場合，なぜ間違っているのか，プログラムの振る舞いは正答とどのように違うのかがすぐにはわからず，学習，理解を効率

[1] 電気通信大学
[a)] oizumi@ipl.cs.uec.ac.jp
[b)] iwasaki@cs.uec.ac.jp

よく進めることが難しい．また，一般に，同じ振る舞いをするプログラムの書き方は，穴埋めのレベルであっても複数通りがあり得る．単純な文字列比較あるいは（「x+y」と「y+x」を同じとみなすといったような）ヒューリスティクスでは，正答が複数あり得るような状況に完全に対応することができない．また，正答判定が可能な穴埋めだけに問題範囲を狭める結果につながりかねない問題点がある．

そこで本研究では，Python 言語のプログラム視覚化ツールである Online Python Tutor [1]（OPT）を拡張し，プログラムの視覚化とソースコードに対する穴埋め問題を組み合わせた学習ツールを設計し作成した．OPT 本来の機能である強力なプログラム視覚化機能を利用し，問題作成者向けとして，穴埋め問題の容易な作成機能と問題の配布機能を，学習者向けとして，複数解を許容する穴埋め問題の演習機能と，学習者による誤答プログラムの視覚化機能を実装した．これにより，プログラミングの学習において，プログラムの視覚化を用いてコードの読み書きを同時に行う学習を実現した．

問題作成者はプログラムのソースコードと穴埋め指示を入力して，穴埋め問題を容易に作成することが出来る．また，作成した問題のデータは URL によって管理されるため，学習者への問題の受け渡しも容易である．学習者は問題作成者から提供された URL にアクセスするだけで問題を入手し，解くことが出来る．問題に対し誤答を答えた場合には，誤答プログラムによる実行過程と正答の実行過程を並べて視覚化する．このことにより，学習者の書いたプログラムとその実行結果の結びつきを強め，プログラミング言語に対する理解度を向上させることが期待できる．

本稿では，2 章にプログラミング学習の手法を述べ，3 章で目的と方針を述べる．4 章に Online Python Tutor について述べる．5 章で拡張 Online Python Tutor の概要と利用方法について述べ，6 章で実装を述べる．7 章で関連研究を紹介し，8 章で本稿をまとめる．

## 2. プログラミング学習に用いられる手法

プログラミングの初学者にとって，プログラミングを理解するための課題の一つは，プログラムの実行状態を理解することである [8]．ここでプログラムの実行状態とは，変数の値の変化やループ処理などのことを指す．

今まで，プログラミングの初学者の理解度向上のために，多くの教材や学習ツール [1][2][3][10] が開発されてきた．以下では，学習者の理解度をより高めるために用いられている手法と考えられる問題点について述べる．

### 2.1 穴埋め問題

穴埋め問題は広く用いられる学習法であり，暗記的な知識の習得を高速に行うことができるメリットがある [6]．プログラミング言語の学習においても，文法の習得には暗記的な要素もあるため，プログラミング学習に対しても穴埋め問題は有効である．

プログラミング言語の学習においては，ソースコード中のキーワードをマスクする，あるいはアルゴリズムの一部をマスクすることが考えられる．これにより，フルスクラッチでプログラムを書くよりも難易度を抑え，学習者が行うプログラム実装の負担を減らすことができる．さらに，穴の場所によって難易度をある程度調節できることから，学習者のレベルに合わせた継続的な学習を可能とする．

しかしプログラミングにおいては，覚えた文法を用いてコードを書けるようにならなくてはならない．学習者によっては文法を暗記的にのみ習得し，それを実装に用いることができずに理解が追い付かなくなってしまう．

また 1 節で述べたように，穴埋め問題の答え合わせには文字列比較を用いることが多い．一般に同じ振る舞いをするプログラムの書き方は複数通りがあり，これに完全に対応しようとすることは困難である．穴埋め問題を提供するツールとして JPLAS [10] があるが，これは穴埋め問題の対象を

キーワードのみとすることで，この問題を回避している．

### 2.2 プログラム実行過程の視覚化

プログラム中の変数の値の変化を表に表し，実行過程を見えるようにした視覚化，あるいは図やアニメーションを用いることでより直感的に学習者が理解できるようにした視覚化を行うツールの研究・開発 [2][3] も，今まで数多く行われてきた．視覚化を行うことによって，実機上での実行過程の理解を深めることができ，またツールを用いることで学習者のペースに合わせた学習を可能とする．しかし，視覚化対象のプログラムを教師側が用意するだけでは，学習者はプログラムの動作を理解することはできても，プログラム作成能力に直接は結びつかないおそれがある．だからといってプログラムを一から作成させ，視覚化により作成の助けにしようとしても，初学者にはハードルが高くなることも予想される．

### 2.3 コンテスト形式

近年，プログラミング学習の手法としてコンテスト形式を採用する試みが行われている [9]．他の人の成績を確認することができ，競争をすることで学習意欲を向上させることができる．また，完答ではなくても，問題の一部の満たす解答には部分点を与えるなど，初心者に対する柔軟な配慮も可能である．

しかし，コンテストツールの多くはソースコードを正誤判定を行うサーバに送信するため，フルスクラッチでソースコードを書く技術力が前提となっている．成績下位である学生はこの力が不足しており，学習に有効に活用するにはハードルが高い．

## 3. 本研究の目的と方針

前節にあげたプログラミングの学習法では，プログラムを書くことと読むことが独立している．そのため，読み書き能力の間にギャップが生まれ，結びつきが薄い．このことから，学習者によっては，文法だけを覚えプログラムの動作が理解できない，またはあるプログラムの動作はわかるが他のプログラムへの応用ができないことが，挫折の要因になっていると考えられる．また，このような挫折を経験した学習者を対象とした補助ツールがないことも，学習を諦める要因の 1 つと考えられる．

そこで本研究では，初学者を対象とした，プログラムの読み書き能力を同時に養うことを目的とした，プログラム視覚化と穴埋め問題を組み合わせた学習ツールを提案する．視覚化されたプログラムのソースコードの一部を隠し，穴埋め問題とすることで，実機上での動作とソースコードの読み書きを関連させ，学習者の理解度の向上を目指す．

問題の答え合わせには，視覚化に用いる実行トレースを利用することにより，正答が複数ある場合に対処する．こうしてプログラミングの穴埋め問題の課題点である，複数の記述方式に対しても正しい評価が可能になる．また，学習者による解答が誤答であった場合には，その誤答コードの実行過程を，正答コードの実行過程を並べて視覚化する．こうすることで，自分の書いたプログラムが実機上でどのように動作するか，さらに正答との動作の違いを確認することができ，コードと動作の対応を再確認し，理解が深まることが期待できる．

以上を実現するために，既存のプログラムの視覚化ツールである Online Python Tutor [1] を拡張した．

## 4. Online Python Tutor

Online Python Tutor [1][5]（以下 OPT）は Python プログラムの視覚化ツールである．フロントエンドは HTML と JavaScript によって構成されているため，プラグインなどの導入が必要なく，利用者はホームページにアクセスするだけで利用することができる．

利用者は，エディタ画面にソースコードを記述する．また，ソースコードを書く以外にも，多くのサンプルプログラムが用意されているため，それを選択することも可能である．記述したソースコードをサーバに送信することでプログラムが実

行され，その実行過程を視覚化したものを利用者に提示する．その様子を図 1 に示す．

OPT ではプログラムの実行を行単位で管理し，プログラムの行単位の実行経過を表示する．画面の右側にヒープ領域や変数の保持する値，関数のコールスタックが表示される．利用者は視覚化画面の左側にあるボタンを操作することによって，前後の行の時点での変数の値，スタック等の実行状態を確認することができる．また，スライダを操作することで，任意のステップにおけるプログラムの実行状態を確認することができる．

OPT はプログラム視覚化の共有機能を二つ備えている．まず第一に，ある利用者が生成したプログラム視覚化に対し，固有の URL を生成する．この URL を他者と共有し，別の利用者がこの URL を用いて OPT にアクセスすることで，まったく同一のプログラム視覚化の状況を再現することができる．第二に，プログラム視覚化の他の web ページへの埋め込み機能が提供されている．これにより，利用者が OPT のサイトに直接アクセスする必要なく，視覚化されたプログラムの実行過程を確認することができる．この機能はいくつかの web 上の教科書において利用されている [7]．その様子を図 2 に示す．図 2 では，画面の上部にテキストが記述されており，その下に OPT によって作成したプログラムの視覚化を同一ページ内に埋め込んで表示している．このように，埋め込み機能を利用することで，教材のテキストとサンプルプログラムを同時に確認することができる．以上のように，OPT は視覚化ツールとして強力な機能を持ち合わせているが，視覚化以外の機能は持ち合わせていない．

一方，プログラミング教育・学習に Python を利用する点については，近年 Python がプログラミングの基礎教育に用いられることが増えている．米国の主要大学では，基礎プログラミングの科目に Python が最も用いられているという調査結果 [4] がある．また国内においては九州大学 *1 や愛知大学 *2 などで，Python が基礎プログラミングの科目に用いられている．

このような傾向から，より柔軟なプログラミング教育のために，OPT には問題演習や掲示板の埋め込み機能などの実装が今後の課題として挙げられている．しかし，現在のところまだ実現はされていない．

## 5. 拡張 Online Python Tutor の概要

本節では，OPT に対し行った拡張に関して述べる．OPT のソースコードはオープンソースとして公開されており，改変も自由に行うことができる．本研究ではこれを元に拡張を行った．

拡張 OPT は OPT 本来の機能のうち，プログラムコード入力と入力プログラムの視覚化の生成を用いて，問題作成者による穴埋め問題の作成と問題の配布機能を実装したものである．また学習者向けの機能として，提供された問題に対する答え合わせ機能と解答に対するフィードバック機能を，プログラム視覚化の埋め込み機能を用いて実装した．問題作成から学習者による解答までの一連の動きを図 3 に示す．図中の黒文字で書かれた部分は OPT 本来持っている機能を表す．赤文字で書かれた部分が本研究による拡張である．

問題作成者は，拡張 OPT に対し正答となるプログラムと穴埋め問題とする場所のマスク指定を送信する．これに基づいて拡張 OPT は穴埋め問題を含むプログラムの視覚化と，これを再現する URL を問題作成者に提供する．問題作成者はこの URL を他の web ページに埋め込んで作成した学習用ページの URL，を学習者に提供する．学習者は問題作成者から提供された URL を用いて拡張 OPT によって作成された問題にアクセスし，問題を解く．

プログラミングの学習支援ツールの中には，学習者の理解度の向上だけでなく，科目の評価を対象とした評価支援などを目的としたツールも多く存在する．pgtracer [3] は Moodle プラグインとして作成されたツールであり，その傾向が強い．一方，OPT は初学者の理解度の向上を主目的としており，評価支援は対象とはしていない．そのため，

*1 http://syllabus.kyushu-u.ac.jp
*2 http://ar.aichi-u.ac.jp/python/

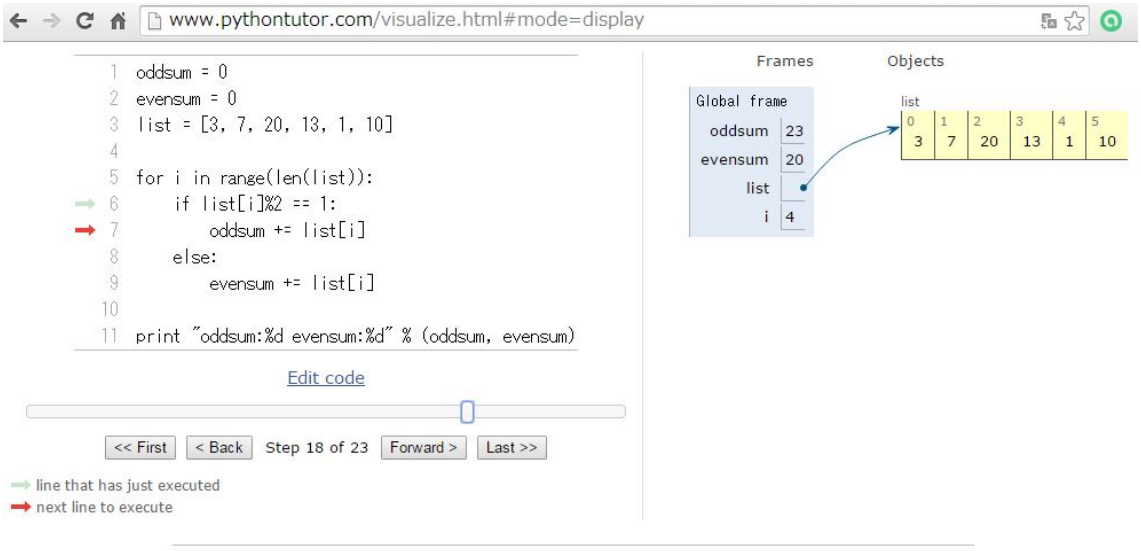


図 1　Online Python Tutor によるプログラム視覚化

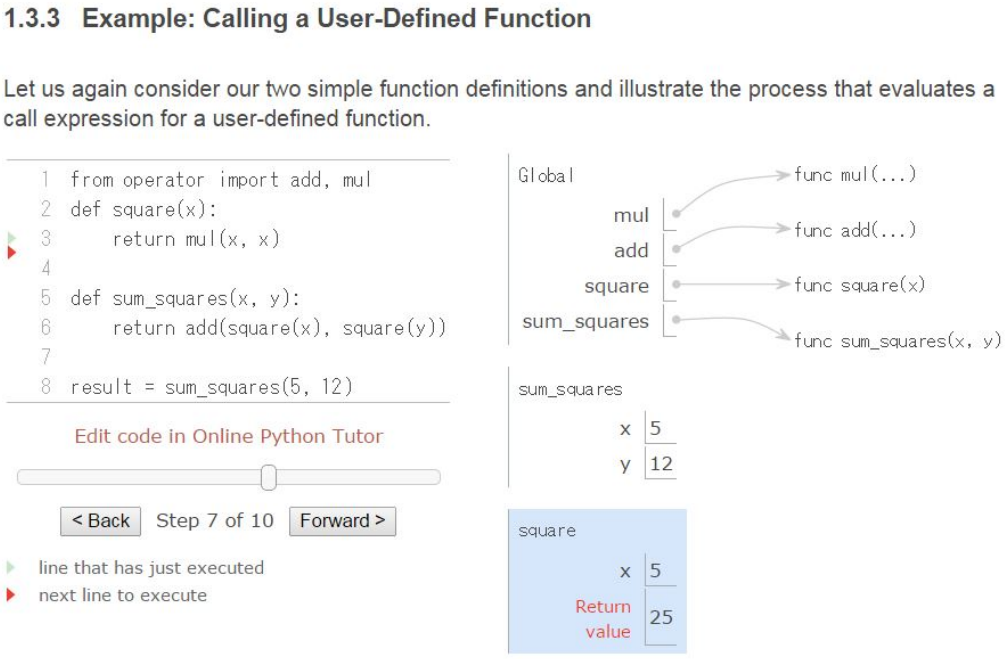


図 2　OPT を埋め込んだ web 教科書 ([7] より転載)

拡張 OPT においても，教師側の評価支援は考慮せず，学習者がより理解しやすくなることを目標とした．

以下では，問題作成者と学習者双方の拡張 OPT の利用方法について述べる．拡張 OPT は Google App Engine を利用して公開しており [*3]，誰でも利用できる．

[*3] http://ipl-extend-opt.appspot.com/

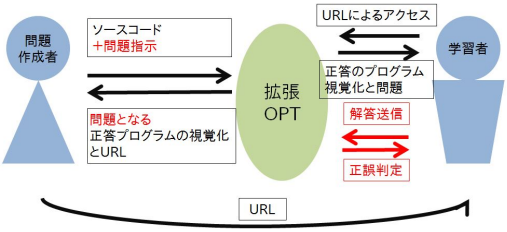


図 3　拡張 OPT の使用概観図

### 5.1　問題作成者

問題作成者は，拡張 OPT のコード入力機能とマスク指定機能，プログラムの視覚化機能を用いて問題を作成する．また，必要に応じ，生成した問題を含めた web ページを作成し，学習者に配布する．

#### 5.1.1　プログラムと秘匿部分の指定

拡張 OPT による問題作成の様子を図 4 に示す．

問題作成者は，問題の元となるソースコードに加え，穴埋め問題とするマスク部分を指定する．マスク部分の指定は，コードエディタ上で隠す部分をドラッグし，ボタンをクリックすることで行う．指定した部分は，コードエディタの下部に行，開始位置，終了位置を記した表として表示され，確認することが出来る．また，表の中の Highlight ボタンを押すことによって，指定したマスク部分に対し背景色が付き，コードエディタ上で視覚的に確認することが出来る．

マスク部分の指定については，OPT の視覚化のステップ実行が行単位であることから，単一行に対してのみ可能とした．そのため，複数行にまたがる部分をマスクとしたい場合には，行毎に個別のマスクとして用意する必要がある．

プログラムの記述とマスク指定を終えたら実行ボタンを押し，問題の確認へ移る．

#### 5.1.2　問題の確認

問題の確認画面を図 5 に示す．この画面では，学習者に提供するものとほぼ同一の画面が表示される．

OPT と同じく，ボタンやスライダの操作によって，プログラムの実行の様子を動的に確認することが出来る．学習者と同じように問題に対する解答も行うことができ，問題のプレビューを行うことが可能である．

設定した問題やプログラムに不備があれば，ソースコードの下部にある [Edit code] のリンクから，問題作成画面に戻り，問題を修正することができる．

#### 5.1.3　問題 URL の生成

作成した問題が正しければ，問題の URL を生成する．問題の URL には，解答ソースコードと問題の指定，図示の手法を示した視覚化のオプションが含まれる．図 6 に生成した穴埋め問題の URL を示す．図に示す通り，URL にはソースコードが直接埋め込まれているため，生成された URL を直接学習者に配布すると，URL から問題を解析される可能性がある．これを防ぐためには，ソースコードや問題の情報を暗号化して URL に埋め込む，あるいは学習者に対して問題を提供する際に，短縮 URL サービスなどのリダイレクトを利用して生成した URL の文字列を学習者が直接見られないようにすることが考えられる．

しかし，拡張 OPT は教師側の評価支援を対象とはしていないため，無理に URL を隠す必要はないと考えている．学習者は URL から穴埋めの正答を得たとしても，学習にはならず，学習者自身のためにならないからである．

視覚化された問題は HTML の iframe タグを用いて他の web ページに埋め込んで使用することを想定している．作成した複数のプログラム視覚化による穴埋め問題を，同一の web ページに埋め込むことも可能である．問題作成者は問題を埋め込んだ web ページの URL を学習者に配布する．

### 5.2　学習者

学習者は，問題作成者から提供された URL を用いて問題演習を行う．図 7 に学習者による演習画面の様子を示す．学習者用のプログラムの視覚化画面では，図 5 で示した中の [Edit code] のリンク以外の機能が与えられる．その上で問題プログラムの視覚化を操作し，実行トレースを確認することで，マスク部分に記述するべきコードを推測する．

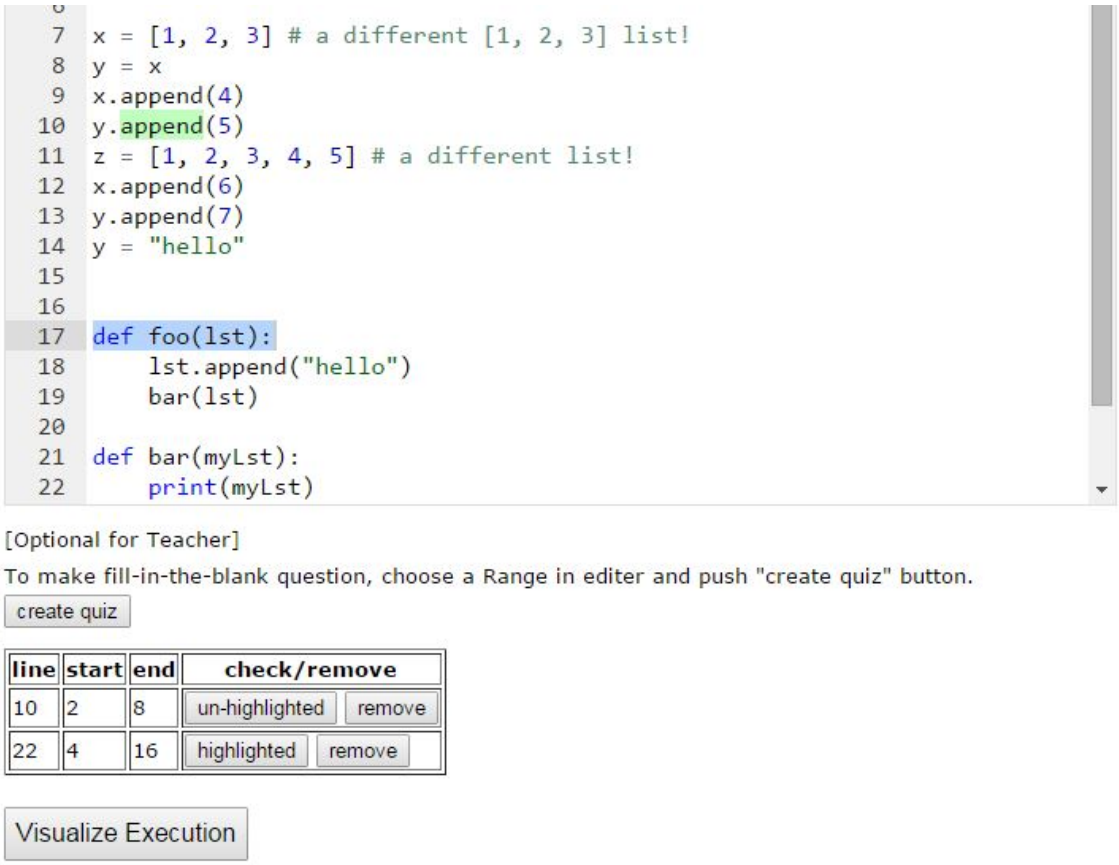


**図 4** 拡張 OPT による問題作成

穴埋め問題への解答は，マスクされた部分それぞれに対して行う．解答欄左側のプルダウンメニューから答える問題番号を選択し，右側のテキストボックスに答を記述する．Python はブロックをインデントによって管理しているため，マスクの範囲によっては if などの制御文や，関数定義などのインデントを含むマスクが問題となっている場合がある．この場合には，1 インデントを 4 スペースとして入力を行う．これは学習者用の解答欄を HTML の textbox タグによって実装しており，textbox タグの性質により，タブ文字を入力できないからである．解答欄にコードを記述したら，テキストボックスの右側にある，[send answer] ボタン押して答え合わせを行う．

答え合わせの結果，学習者の解答が正答であればマスクを外し，その部分の正解プログラムコードを表示する．正解のプログラムコードは問題作成者による想定解のコードを表示するため，学習者が正答と異なる記述で解答した場合には，同じ処理を行う記述が複数存在することを確認でき，より深い理解ができる．

誤答であれば，誤答による実行過程を正答による実行過程と並べて表示する．この様子を図 8 に示す．そのため，学習者は誤答行の前後の実行状態を正答のそれと見比べることによって，何らかの値が正答と誤答で異なることがわかる．このように，学習者は実行された誤答の結果を確認することで，自分が書いたコードと実機上の動作を結び付けることでより理解が深まる．

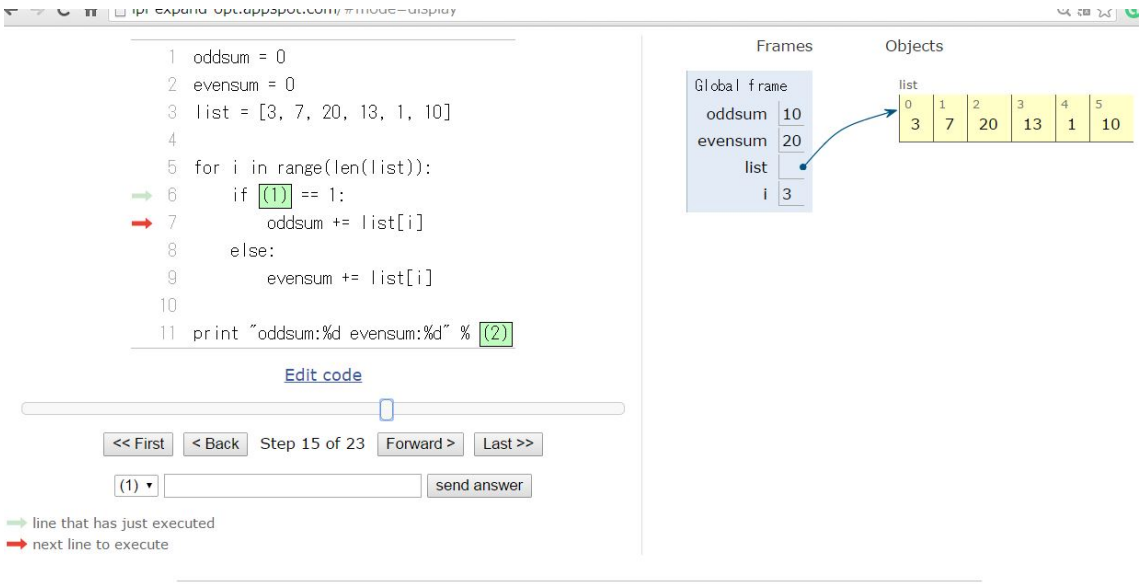


**図 5** 問題作成者によるプレビューの様子

```
http://ipl-extend-opt.appspot.com/quiz-
embed.html#code=lis+%3D+%5B1,+2,+4,+5,
+4%5D%0A%0Alis.append(6)+
%23+6%E3%82%92%E6%9C%AB%E5%B0%BE
%E3%81%AB%E8%BF%BD%E5%8A%A0%0Alis.insert
(2,3)+%23+3%E3%82%922%E3%81%A84%E3%81%AE
%E9%96%93%E3%81%AB%E8%BF%BD%E5%8A
%A0%0Alis.pop(5)+%23+5%E3%81%AE%E5%BE%8C
%E3%82%8D%E3%81%AB
%E3%81%82%E3%82%8B4%E3%82%92%E5%89%8A
%E9%99%A4%0A%0Aprint+lis&origin=opt-
frontend.js&cumulative=false&heapPrimitiv
es=false&drawParentPointers=false&textRef
erences=false&showOnlyOutputs=false&py=2&
rawInputLstJSON=%5B
%5D&curInstr=0&makeBlankRange=%5B
%5B2,4,13%5D,%5B3,4,15%5D,%5B4,4,10%5D
%5D&codeDivWidth=450&codeDivHeight=600
```

**図 6** 生成される URL の例

## 6. 拡張 Online Python Tutor の実装

本節では，OPT に施した拡張の実装について述べる．

OPT では，フロントエンドの構成に JavaScript を用いており，web ページの大部分を動的に生成している．本ツールで実装する機能についても動的に HTML を変更するものなので，JavaScript コードを拡張することで実現した．

穴埋め問題の生成には，エディタ画面として用いられている AceEditer の API を利用することで，マスク範囲の指定を可能にした．

視覚化画面においては，プログラムコードのマスク部分を JavaScript によって動的に秘匿し，視覚化の動的な生成機構を利用することで，学習者の解答コードの視覚化も正答と並べて表示するようにした．

答え合わせには，正答コードのマスク部分を学習者による解答に置換したものをサーバで実行し，得られた実行過程と正答による実行過程を比較する．プログラムの実行過程は，行単位の実行によるヒープ領域や関数のコールスタックなどの状態が JSON 形式によって構成されている．行単位で正答と解答の JSON を比較し，一致していれば正解とする．既存研究 [3] においては，模範解答と完全に一致していなければ正解とすることができないという問題が挙げられているが，拡張 OPT の方法を用いることによって，一つの穴に対して複数の記述が正解として存在し得る場合でも，そのどれを用いても正解と判定することができる．例

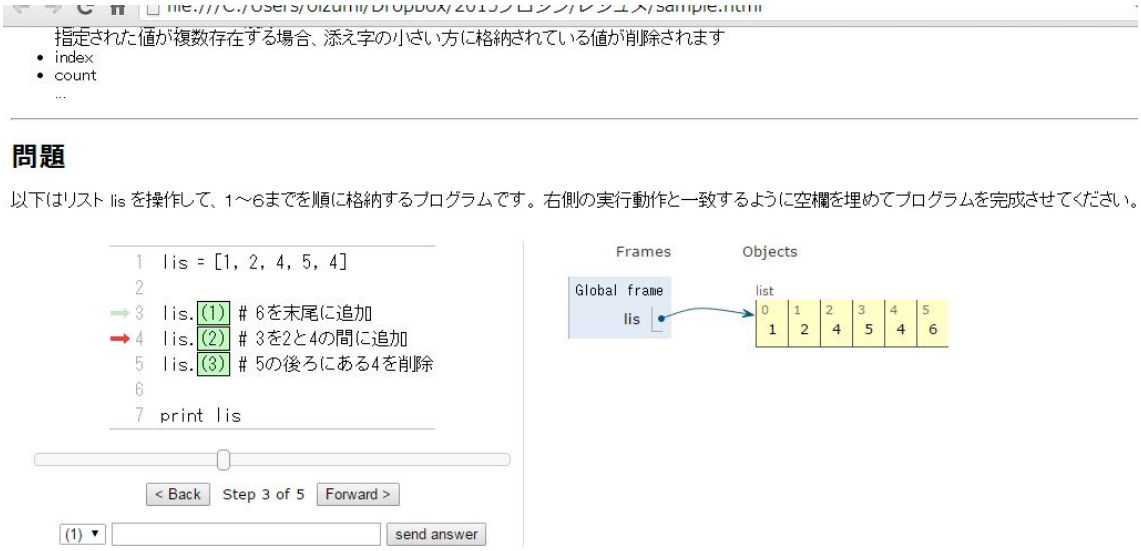


図 7　学習者による解答画面

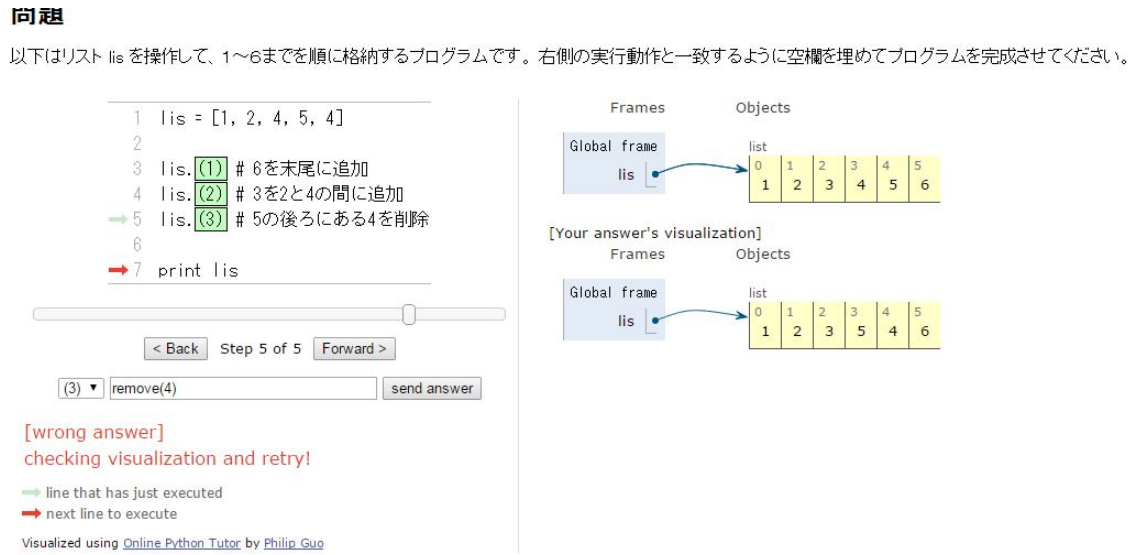


図 8　学習者が誤答を送信した場合

えば，図 7，8 の問題 (2) では，「`insert(2,3)`」と入力することが期待されるが，「`insert(2,2+1)`」や「`insert(-4,3)`」といった入力でも正解と判定される．

しかし，同じ処理を記述したとしても不正解としてしまうケースが存在する．関数の実行について，自作関数の実行は関数のコールスタックを表示し，複数ステップをかけて処理を行っている．これに対し Math ライブラリなどの組み込み関数は，コールスタックに乗せずに 1 ステップで処理している．そのため，視覚化を行った際の実行ステップ数が異なってしまうため，同じ処理を行う記述であっても不正解と判定される．この点への対処は今後の課題とする．

## 7. 関連研究

### 7.1 UUhistle

UUhistle [2] は，Python を対象としたプログラム学習支援ツールである．UUhistle では，プログラムの視覚化機能と問題演習機能を備えている．ここでいう問題は，実行結果を求める選択式問題や，視覚化されているオブジェクトを操作する問題であり，本ツールのような穴埋め問題は実装されていない．

### 7.2 pgtracer

pgtracer [3] は C++言語を対象とした穴埋め問題を用いた，Moodle を基盤とするプログラミング教育支援ツールである．ソースコード，実行トレースの両方に穴を空けることが出来る．また，教師支援の機能として，解答データの収集機能などを備える．1 画面で実行過程がわかるよう，実行トレースは表として視覚化されて与えられるため，本研究とは見せ方が異なっている．また穴埋め問題の解答は文字列の比較で行っているため，複数の正解が存在する場合でも正答プログラムと同一の記述でなければ不正解となる．

## 8. まとめ

本稿では，OPT を拡張し，プログラム視覚化と穴埋め問題を組み合わせたプログラミング学習ツール拡張 OPT を提案した．本ツールでは学習者によって記述されたコードを実行して答え合わせを行うため，正解が唯一に定まらない問題に対しても，正しく評価することを可能とした．

今後の課題としては，まず拡張 OPT を用いた教材の充実がある．現在，視覚化を含んだ初学者向けの教材は少ない．また，拡張 OPT は他の教材に組み込んで使う，補助教材としての利用方法を想定している．そのため，既存の教材に拡張 OPT を組み込んだ，初学者向けの教材の拡充が必要である．

また，一問で複数行をマスクとして指定できるようにする，という点があげられる．現状では，答え合わせは実行過程の JSON 形式が完全に一致しているかを判断しているため，一問題のマスクは単一行内で閉じている必要がある．そのため，連続した行がマスク指定されており，この順番が逆に記述してもプログラムに影響がない場合であっても，問題作成者の意図した通りの順にコードを記述する必要があり，学習者にとっては束縛となってしまう．この制約を取り払うには，答え合わせの手法を改良し，全実行過程の比較ではなく，特定のステップにおいて JSON 形式の状態が一致していれば良いようにする必要がある．

更に，ツールに対する評価方法の策定と評価を行いたいと考えている．

### 参考文献


[1] P. J. Guo. Online *Python Tutor: Embeddable Web-Based Program Visualization for CS education.* In Proceeding of the 44th ACM technical symposium on Computer science education, SIGCSE’3, pp.579–584, New York, NY, USA, 2013.

[2] J. Sorva and T. Sirki. *UUhistle: a software tool for visual program simulation.* In Proceedings of the 10th Koli Calling International Conference on Computing Education Research, pp.49–54, 2010.

[3] 太田，柳田，大月，掛下．穴埋め問題を用いたプログラミング教育支援ツール pgtracer の概要と学生用機能の実装，情報処理学会研究報告 コンピュータと教育 (CE) 2014-CE-124-5 pp.1-8, 2014.

[4] Philip Guo, *Python is Now the Most Popular Introductory Teaching Language at Top U.S. Universities*, ACM BLOG, 2014. http://cacm.acm.org/blogs/blog-cacm/176450-python-is-now-the-most-popular-introductory-teaching-language-at-top-us-universities

[5] Online Python Tutor — Learn programming by visualizing code execution http://pythontutor.com/

[6] Piotr Wozniak, *Effective learning: Twenty rules of formulating knowledge*, 1999. http://www.supermemo.com/articles/20rules.htm

[7] J. Denero. *CS61A: Structure and Interpretation of Computer Programs.* http://www-inst.eecs.berkeley.edu/~cs61a/

[8] B. du Boulay. *Some difficulties of learning to program.* Journal of Educational Computing Research, 2(1):57–73, 1986.

[9] 富永，西村．実行テスト系列を取り入れた小コン

テスト形式の初級 C 演習―学生の特典状況の時系列分析による活性度の推定―，情報処理学会研究報告　コンピュータと教育 (CE) 2012-CE-116-15 pp.1–8, 2012.

[10] 伊永，舩曳，中西，渡邊，天野．Java プログラミング学習支援システムの穴埋め問題機能の拡張と授業への適用，電子情報通信学会技術研究報告 ET, 教育工学 111(473), pp.7–12, 2012.