

# Python 実行サーバを必要としない オンラインプログラミング環境の開発

周藤祐汰<sup>1</sup> 高木正則<sup>1</sup> 市川尚<sup>1</sup>

**概要**：高等学校の次期学習指導要領で新設される科目「情報 I」の研修用教材に対応した、ブラウザ上で動作するオンラインプログラミング環境の開発を行った。本環境では、WebAssembly 上で実装された Python コンパイラ、Pyodide を用いることにより実行結果がサーバを介さずに受け取ることができる。そのため、生徒全員が一斉に演習用サーバにアクセスするための十分なネットワーク帯域を確保できない場合でも、プログラミングの演習を実施できることが期待される。

**キーワード**：プログラミング教育, Python, プログラミング環境

## Development of an Online Programming Environment that is Python Execution Server Needless

YUTA SUDO<sup>†1</sup> MASANORI TAKAGI<sup>†1</sup>  
HISASHI ICHIKAWA<sup>†1</sup>

### 1. はじめに

高等学校の次期学習指導要領で新設される科目「情報 I」の「コンピュータとプログラミング」では、「目的に応じたアルゴリズムを考える力」、「目的に応じたモデル化やシミュレーションを適切に行う」の 2 点が求められている[1]。これらの能力を養うためには実際にプログラミングを行うことが重要であるが、高校のような高い情報セキュリティ対策が求められる教育現場で、各生徒の端末にプログラミング環境を構築することは難しい。

Google Colaboratory[2]のようなブラウザ上でオンラインプログラミング環境を提供するサービスも多くリリースされている。しかし、学校ごとにアカウントの管理が必要になることや、十分なネットワーク帯域を確保できないなど、オンラインプログラミング環境の導入も簡単ではない。また、サーバ側に遠隔でソースプログラムを実行させる機能を搭載したオンラインプログラミング環境の場合、以下のような対応も求められる。

- セキュリティ対策 (OS コマンドを実行させない等)
- 実行に極端な時間がかかるプログラム (無限ループなど) への対応
- 多人数のプログラムの実行リクエストを処理可能なサーバの管理 (一時的なアクセス集中への対応)

そのため、サーバの能力を考慮すると低コストで運用することが非常に困難である。そこで、本研究では、ブラウザで動作し、実行結果の取得にオンライン接続を必要とし

ない Python3 プログラミング環境を構築する。

### 2. 関連研究

ブラウザ上で動作する Python プログラミング環境として、BitArrow (BA-Python) [3]がある。BitArrow はブラウザ上でソースコードを作成・編集・実行できる環境である。JavaScript による Python トランスパイラを実装しており、サーバとの通信を介さずに手元で Python を実行することができる。また、OS モジュールなどの機能を一部制限して実装したサーバ実行機能も有しており、サーバ実行機能では numpy などの科学計算ライブラリの利用も可能となっている。本研究では、主にオフライン環境下でのプログラムの実行を想定している点で関連研究と異なっており、numpy, matplotlib などの各種ライブラリがオフラインモードで実行できる点やソースコードの自動保存機能を提供している点が本研究の特徴となっている。

### 3. システムの設計・開発

#### 3.1 システムの概要

本システムはサーバ側で特殊な処理を行っておらず、静的な HTML, JavaScript, wasm ファイルの配信のみで動作する。開発環境は Google Chrome 79, Web サーバは Apache を利用した。本システムの対応ブラウザは Microsoft Edge, Google Chrome の最新版で動作することを確認している。図 1 にシステム利用時の画面例を示す。また、表 1 に本システムの最低動作、推奨環境を示す。最低動作環境で本システムを利用した場合、他のアプリケーションを実行するなどしてメモリ使用量が増えると、システムがクラッシュする可能性がある。

<sup>1</sup> 岩手県立大学大学院ソフトウェア情報学研究科  
Graduate School of Software and Information Science, Iwate Prefectural University

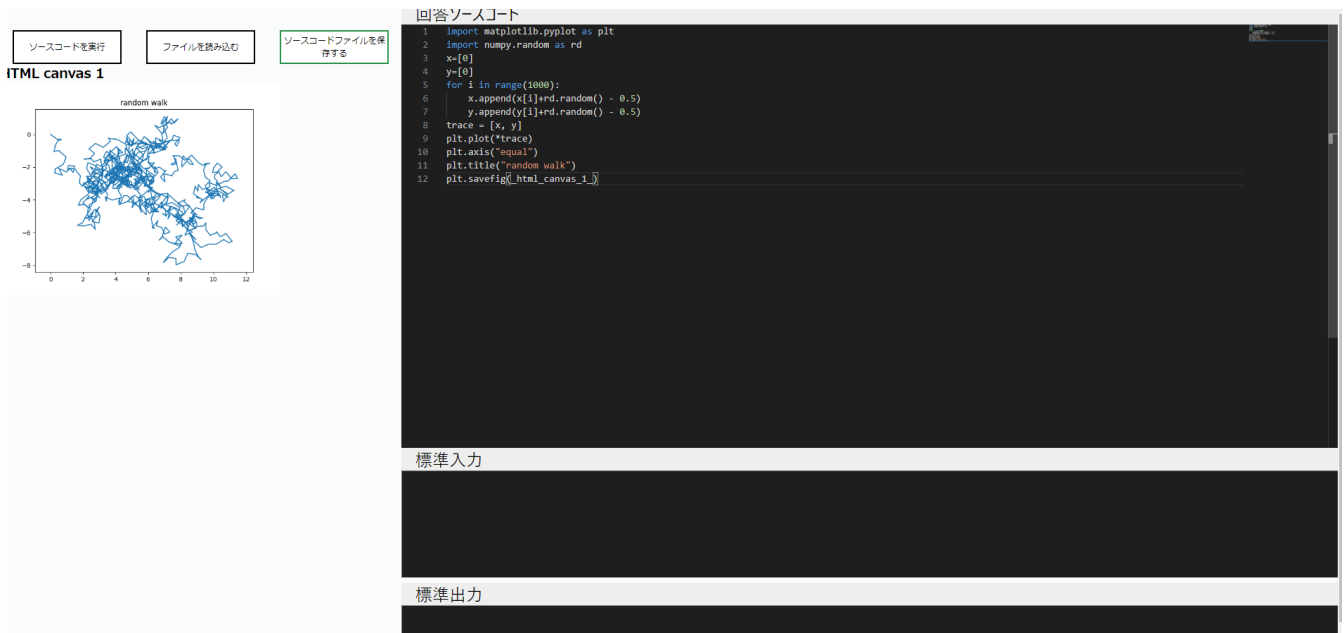


図1 システムの実際の画面

表1 本システムの動作環境

推奨環境	
CPU	Intel 第8世代以降
メモリ	8GB以上
OS	Windows 10
ブラウザ	Google Chrome 80
最低動作環境	
CPU	Intel 第6世代以降
メモリ	4GB以上
OS	Windows 10
ブラウザ	Google Chrome 74

### 3.2 Python 処理系の実装

本システムで採用したブラウザ上で動作する Python 処理系は、前述した BitArrow[3]で採用されている JavaScript のトランスパイラを実装するアプローチとは異なり、Mozilla グループが開発を行っている、WebAssembly で実装されている CPython の処理系である Pyodide[4]を採用した。Pyodide は、Python の処理系そのものをブラウザ上に搭載しているため、JavaScript トランスパイラとは異なり、実際の Python と全く同じ動作を行う。また、numpy[5]や matplotlib[6]などの文部科学省研修用教材[7]で利用されている科学計算ライブラリを内包しており、「情報 I」の対象となるプログラミングの実行を全て網羅できるため採用した。

しかし、ブラウザ上で Python 処理系を取り扱う上でファイルの入出力、標準入出力等の機能は取り扱いが異なる。そのため、別途、本システムの形式に合わせて実装を行った。以下は Pyodide の機能以外に本システムで実装した本環境の独自の機能について説明を行う。

### 3.3 本環境独自の実装機能

#### 3.3.1 標準入出力・標準エラー出力機能

通常 Pyodide の実装では標準入力はブラウザコンソール、標準出力は入力ダイアログで行われているが、プログラミング環境の利便性の問題上、ブラウザ上に標準入力・出力画面を用意し、その画面で行うことにした。標準入力は改行文字ごとに一度の標準入力として取り扱われる。また、標準エラー出力は通常ブラウザコンソール内に表示されるが、標準出力画面に表示するように実装を行った。

#### 3.3.2 matplotlib 作成画像表示機能

文部科学省の「情報 I」の研修用教材[7]にも利用例があるように、Python で行った統計処理をグラフ描画するために用いられるライブラリである matplotlib の画像出力ができる機能を開発した。具体的には、WebAssembly 上で出力される画像のバイナリデータをバッファに書き込み、Python 上のバッファデータを HTML Canvas 上に描画して実現した。図2に出力された matplotlib の画像例を示す。

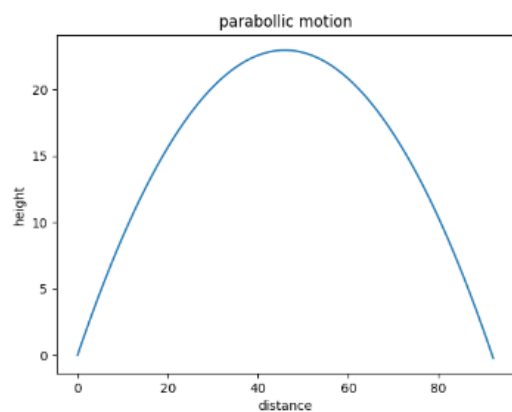


図2 出力された matplotlib の画像例

### 3.3.3 オフライン機能

モダンブラウザに搭載されている機能である、Cache API を用い、初回のファイルダウンロードをサーバ上から行っていれば、オフラインでも本システムで提供するプログラミング環境が動作する機能を開発した。本システムにおいて Python の実行自体はサーバ通信を必要としないため、この機能により、ネットワーク環境に影響されにくい Web プログラミング環境の提供が可能になる。

### 3.3.4 ファイル保存機能

本システムは、オフラインで動作することを想定しているため、サーバ上に学習データとしてソースコードを保存する等の機能を有していない。その代わりに、ソースコードが変更されるたび、ブラウザのローカルストレージ上に逐次自動で変更が保存されるようになっている。また、Google Chrome の利用者に対して、ローカルのファイルをエディタ上に読み込むことができる機能と、エディタ上に記述されたソースコードをローカルの保存領域に保存する機能をブラウザ上のインターフェイスで提供している（図 1）。

### 3.3.5 入力支援機能

Python の仕様上、インデントが非常に重要であるため、特定のコードブロック内ではインデントが維持されるような入力支援機能を提供している。また、一般的なプログラミング用のテキストエディタと同等の構文ハイライト機能も入力支援機能として提供を行っている。

## 4. 評価

本研究で開発したプログラミング環境が、高校の授業で活用可能であるかを検証するため、文部科学省が公開している研修用教材に掲載されているソースコードを実行して検証した。表 2 に BA-Python のブラウザ実行版と本環境の実行結果を示す。各項目に関するソースコードの詳細は研修用教材を参照されたいが、ここでは本システム特有の事項について説明する。

図 3 は表 2 ウ-1 にある、モデル化とシミュレーションで示されるソースコードを本システムで動作する形に書き換えたものである。最終行のみ、本システムで動作するように変更した。図 3 のソースコードを実行すると、図 2 の画像がシステム上の画面に表示される。

表 2 の○の項目については、`plt.show()` とされている関数を `plt.savefig(_html_canvas_1)` に置き換えることで正常に動作する。今後、`plt.show()` と `plt.savefig(_html_canvas_1)` の置換をシステム側で自動的に行うことによって、文科省教材と同等のコードが実行できるよう目指す。

表 2 イ-3 の応用的プログラムについては、WebAPI との連携部分（python の `requests` モジュール）が Pyodide 未対応であったため動作しなかった。Javascript 側の `XMLHttpRequest` モジュールをラッパーとして用意することで、数行程度の変更で同様の結果を得られることができるが、異なるオリジンへの HTTP リクエストを行うため、CORS 制約の回避に注意が必要である。

表 2 文科省教材の動作可否

項目	学習内容	動作可否	
		BA-Python	本システム
ア-2	計算誤差	×	◎
イ-1	外部機器との接続	×	×
イ-2	基本的プログラム	◎	◎
イ-3	応用的プログラム	×	×
イ-4	アルゴリズムとの比較	◎	◎
ウ-1	モデル化とシミュレーション	×	○
ウ-2	確定モデルと確率モデル	×	○
ウ-3	自然現象のモデル化とシミュレーション	×	○

×：実行不可 ○：1～2 行程度の変更で実行可能  
◎：そのまま実行可能

図 3 本システムで動作するソースコードの例

```
import math as math
import matplotlib.pyplot as plt
dt = 0.01
v0 = 30
g = 9.8
x = [0]
y = [0]
angle = 45.0 * math.pi / 180.0
vx = [v0*math.cos(angle)]
vy = [v0*math.sin(angle)]
for i in range(1000):
    vx.append(vx[i])
    vy.append(vy[i]-g*dt)
    x.append(x[i]+vx[i]*dt)
    y.append(y[i]+(vy[i]+vy[i+1])/2.0*dt)
    if y[i] < 0 :
        break
plt.plot(x,y)
plt.title("parabolic motion")
plt.xlabel("distance")
plt.ylabel("height")
plt.savefig(_html_canvas_1)
```

## 5. おわりに

本稿では、次期学習指導要領「情報編」の「コンピュータとプログラミング」に焦点あて、管理・導入コストの低い、ブラウザ単体で動作する Python プログラミング環境の構築を行った。また、文部科学省が公開した研修用教材のソースコードを利用し、動作の確認を行い、本環境がおおよその授業に利用できることを確認した。現在、岩手県内の高校の授業で本環境が利用されている。今後は高校での利用結果を踏まえて機能の拡充や改善を行っていく。なお、SSS2021 では、高校の授業での利用結果も報告予定である。

## 参考文献

- [1] 【情報編】高等学校学習指導要領（平成 30 年告示）, [https://www.mext.go.jp/content/1407073\\_11\\_1\\_2.pdf](https://www.mext.go.jp/content/1407073_11_1_2.pdf),
- [2] Colaboratory, <https://colab.research.google.com>
- [3] オンラインプログラミング環境 Bit Arrow における Python 処理系, 長 慎也, 長島 和平, 間辺 広樹, 兼宗 進, 並木 美太郎, 2019 Information Processing Society of Japan, 122-129
- [4] Pyodide, (<https://pyodide.org/en/stable/>)
- [5] NumPy, (<https://numpy.org/>)
- [6] matplotlib, (<https://matplotlib.org/>)
- [7] 高等学校情報科「情報 I」教員研修用教材（本編）, [https://www.mext.go.jp/content/20200722-mxt\\_jogai02-100013300\\_005.pdf](https://www.mext.go.jp/content/20200722-mxt_jogai02-100013300_005.pdf)