

実践報告 Python で学ぶ Processing.py 入門

東京都立つばさ総合高等学校 主幹教諭
横枕 雄一郎

1. はじめに

コンピュータはツールであり、道具として利用することに意味がある。ソフトウェアの操作習得に重点を置いたり、決められた枠の作業にのみ集中することが可能である反面、クリエイティブな仕事、新しいことを生み出すことに利用するためには、それなりの準備、段取り、企画など高度なことが必要となってくる。そのための学習、課題解決能力を身に付けていくことが目標となる。

本校は総合学科である。情報科では、様々な工夫や、使っていくことの楽しさが伝わるような教材開発ができないか検討してきた。

平成25年度より1年次に必修科目「社会と情報」(2単位)を設置している。学校設定科目として8つの「情報演習」(各2単位)を設けている。様々な課題に対応し解決していく力を身に付けて行けるように、様々な教材を利用している。

また情報科の夏期講習では「フリーソフトウェア Blender」「3ds max 3DCG アニメーション」「CG プログラミング」を開講している。ディズニー・ピクサー・アニメーションで利用されている CG アニメーションの基礎基本的な技術である RenderMan について体験学習を行っている。ディズニー・ピクサーの映画「カールじいさんの空飛ぶ家」でも使用されたオープンソースの物理シミュレーション ODE を使って python cgkit による再現もできた。興味・関心がある生徒にとっては、進路選択のきっかけになるように紹介している。

今回、多くの教育現場で利用されている Processing を Python で記述する processing.py を紹介する。主な利点として、①手続きが簡単、②可読性が高い、③集中して取り組めるなど、視覚的に興味・関心・意欲を引き出す教材である。

2. Python のすすめ

様々なプログラミング言語がある中で、Python は、ユーザがしたいことを効率的に助けてくれる言

語である。教育用として、プログラミングを学びたい人向けであり、オープンソースであり、フリーで入手可能なプログラミング言語である。

生徒に簡単に体験してもらい導入として、コマンドプロンプトより python と打ち込み、インタラクティブモードにて

```
>>>2**10000
```

と実行すると、2の1万乗が表示される。電卓では表示できないので驚きの声が上がります。

(1) Python の長所

- ①クロスプラットフォームである。Windows, MacOS X, Linux で使用できる。
- ②記述性が高い。他の言語と比べてコードの行数が少ない、シンプルに組める。豊富なモジュールがある。
- ③Python は可読性が高い。読みやすくなっている。大きな特長としてコードブロックはインデントを使って表すことになっている。インデントをタブまたは半角スペースで表すが、混在はエラーとなる。授業では半角スペース4つをインデントにするようにしている。

例えば下記のようにインデント(半角スペース4つ)を使う。例：反復処理

```
for i in range(10):
    print i
```

比較処理をインタラクティブモードで記述すると、

```
>>> a=10
>>> if a<0:
...     print "red"
... elif a==5:
...     print "green"
... else:
...     print "blue"
...
```

結果：blue
となる。

(2) Python で簡易 Web サーバ

Python はライブラリが豊富であり、様々なモジュールを利用できる。コンピュータ教室に Python をインストールすれば、ネットワーク接続、HTTP 処理のためのライブラリが標準装備されているので、ディレクトリ(フォルダ)内のファイルを共有可能な Web サーバを簡単に構築できる。方法は 1 行のコマンドを実行するだけである。

コマンドプロンプトを開き、ローカル Web サーバの root ディレクトリにしたいフォルダまで移動後、
`>python -m SimpleHTTPServer`
 を実行する。

本校の授業「情報演習 C(ネットワーク)」において、HTML・CSS を理解する実習に利用している。コンピュータ教室の中で、各生徒同士の Web ページをお互いに確認でき、リンクを作成し、楽しく実習ができる。

Python は、ビジネスでも使われており、Google や映画制作を行っている Industrial Light & Magic や Pixar などの CG 業界で多く利用されている。

(3) Python の短所

- ① Python は実行速度が他の言語より低速である。
- ② Python では型はオブジェクトに関連付けられ、変数には関連付けられていない。多言語を経験していると戸惑う。ミスマッチしている型エラーについては修正していくことができる。
- ③ また様々なバージョンがあり、主に Python2.7 での記述と、Python3 以降では、微妙に異なる。互換性は完全ではない。モジュールも各バージョンに合わせたものを使用しなければならない。

3. Processing でのビジュアル・デザイン

Processing は、MIT メディアラボで開発され、視覚的な表現を得意とするプログラミング言語である。美術系の大学や多くの高校で利用されており、プログラミングを学ぶ初心者にとって最適なツールである。記述は java をベースとしているが、現在公開(ver2.2.1)されている Processing では、java モードから python モードに変更することができる。開発環境 IDE で Add Mode で python モードを追加することができる。エディタ部分に記述し、実行ボタンを押すだけで描画される。

4. 授業実践

本校の学校設定科目「情報演習 D(プログラミング)」の授業でプログラミングの学習・体験を行った。1 学期に Excel VBA を利用して簡単な流れを学び、2 学期中旬から Python を実習として取り入れていた。しかし、エディタとコマンドプロンプトを使っている Python の実習に、取り組めない生徒がでてきた。単にソースを打ち込むだけにとどまり、与えられた課題のみこなす。自分で考えない、工夫しない、面倒と感じるといったことだけになり、プログラミングについて難しいと感じている生徒がいた。そこで、何かしら取り組めるきっかけを与える授業が必要と感じ、Processing の python モードを利用することにした。

現在、3 種類の python モードの方法がある。

- ① processing.exe を起動して開発環境 IDE で切替える python モードを利用する方法
- ② Jonathan Feinberg 氏が開発した processing.py をエディタとコマンドプロンプトで利用する方法
<https://github.com/jdf/processing.py> (web ページ中段からダウンロードできる。)
- ③ Python のモジュールとして開発された pyprocessing を利用する方法
<https://code.google.com/p/pyprocessing/>

この中で授業に導入しやすいのは①の開発環境 IDE であるが、今回は、②のエディタとコマンドプロンプトを用いた processing.py の実習を検討した。①②は、java で開発された Processing の豊富な既存ライブラリを利用することができる。これはとても便利であり、様々な試みを実践できる。③については Python にモジュール pyglet をインストールしなくてはならない。pip を利用すると楽にインストールできる。

(1) エディタとコマンドプロンプトで processing.py の利用

はじめにコマンドプロンプトでどこでも起動できるようにするためには、環境変数 PATH にインストールした processing.py のディレクトリを追記しておかなければならない。

例えば、C:\processing.py-0202 を PATH に追記する。起動確認はコマンドプロンプトから

>processing-py

と実行することで、画面全体がグレーで白い円をマウスで描画できれば、processing.py が起動できている。まずは Processing の入門サイトや付属のサンプルを動かしてみると良い。エディタでソースを記述し保存、コマンドプロンプトから下記を実行する。

>processing-py sample.py

エラーと出力情報は、テキストファイル(err.txt と out.txt)で毎回出力される。また Processing についての参考書籍もたくさん出ており、python モードで記述しなおして確認すると文法の勉強になる。基本図形や 3 次元モデルを描画すると面白くなる。

(2) タートル・グラフィックス

シーモア・バパートはプログラミング言語 LOGO を開発した。学習者が抽象的な概念を学習する上で「シンボルの活用」として「タートル・グラフィックス」を用いた。タートル・グラフィックスを processing.py で試することができる。

ライブラリの Terrapin.jar¹⁾をダウンロードし、C:\processing.py-0202\libraries にコピーしておく、タートル・グラフィックスが可能となる。次のプログラム 1 は、1 行目でライブラリを呼び出し、辺の長さが 2 倍となる正方形を描画する。最後に描画した正方形を画像(jpeg)として保存している。

プログラム 1

```
from terrapin import Terrapin

def square(haba):
    for i in range(4):
        t.forward(haba)
        t.right(90)

def setup():
    global t
    size(400,400)
    colorMode(RGB,256)
    background(0)
    strokeWeight(6)
```

```
c1 = color(0, 0, 255)
fill(c1)
#create Terrapin
t = Terrapin(this)

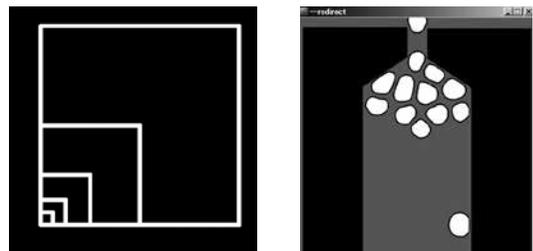
def draw():
    global t
    t.setLocation(50,350)
    t.setRotation(-90)
    print t.x,t.y,t.getRotation()

    haba=20
    for i in range(5):
        square(haba)
        haba=haba*2

    save("terra06.jpg");
```

(3) 2次元物理シミュレーション

PC・スマートフォン等のゲームでも利用されている物理エンジン Box2D を簡易的に扱える Fisica を利用することで様々な事象について試みることができる。processing.py には標準で Fisica のサンプルが付属しているので体験してみると良い。物体落下のシミュレーション後、マウスを使ってドラッグ移動することができるので興味・関心を高めるライブラリである。注意点としては Fisica と他の物理ライブラリ(box2d_processing)が同じ libraries フォルダにあると Fisica は動かない。



タートル・グラフィックスで作成した正方形(左)と 2次元物理シミュレーション(右)

(4) 授業感想

・かめがすぐに動いて楽しかった。すぐに動作が確認できるのでいいなと思った。文字を打ち間違

えてちゃんとプログラムが起動しないときも、どこが間違ってるのか書いてあったから、間違えを見つけた。

- ・点とか線とかがいっぱいでくるのが面白かったです。でもすぐ引数がどうとかエラーが出て、いくつかはちゃんとできませんでした。pu ? とか wa ? など、今までやってきたやつよりも、書くことがわかりにくい気がします。
- ・全然終わりませんでした・・・カメの現在の位置からいくつ移動する&回転するという感覚の違いにひっかかり、なかなか進みませんでした。居残ってがんばります。
- ・今回はなんか慣れ始めたのかわからないけどミスったところなど自分で見つけることができました。ちょっとこれは進歩だなとすこし思いました。
- ・今回は新しいものをやった。個人的には、今までやった中で一番面白く感じたので、暇があったら、家でもやってみたくと思った。世の中にはいろいろな面白いものがあるのだなと思った。
- ・いろいろと数字を変えたりしながらやった。うまくいかないこともあったけど、楽しかった。
- ・最初はなかなか理解するのが難しかったけれど、だんだん慣れていくにつれてできた時のうれしさや感動することができました。
- ・細かい作業でとてもしんどかった。ゲーム制作者やプログラマーなどはこのようなことをしているのだと思うと大変だなと思ひ、自分は消費者のほうがいいと思った。
- ・最後の課題が、さまざまな数字を変えることで面白い形を作ることができたのでとても面白かった。
- ・前半は曲線や円を用いて、さまざまな絵を描いた。
- ・人生でプログラミングを初めてやったが、自分が入力し、それをコンピュータが計算するという一見簡単そうに見えてとても根気のいる作業だったのでなかなか大変でした。1学期はほとんど打つだけで簡単だったが2学期になると Python で新しくやるが増えて少し大変だった。こういう機会はその道に行かない限りあまりできない内容だったので少しやれただけでもよかったです。
- ・周りの人がすごいなと思った。
- ・課題作成では、色がそれぞれついたものを作ろうと思っていた。なかなかうまくいかず、適当に引数で数字を入れて何とか作ることができた。

・Python で書きました。楽しい。打ち直したりするのが面倒くさいなと思った。マークを作るのが難しい。でもできた時は達成感がすごい。

5. 今後の課題

Processing は、世界中で様々な表現活動として使われている。教育現場での利用は、まず自学自習できるリソースが豊富にある。様々な事象や原理を理解する道具として使えば、面白いことができると感じている。3次元コンピュータグラフィックスで取り扱うデータを、タートル・グラフィックスを使って出力させてみたり、群集シミュレーションのデータを作成し、可視化させることも可能であろう。応用範囲はたくさんあると思う。総合学科高校で MSWLogo を授業実践した藤川氏²⁾は次のように述べている。

「英語マニュアルを使って、基本的な使い方を英語で学習することも貴重な経験になる。今まで学んだ英語の知識が、他の授業で役に立つということを経験することも、学習意欲を高めるためのひとつの方法と考えることができる。」²⁾

「プログラムを英語で書くことは、英単語のもつ深い意味を知ることとも密接に関係している。教科書の中で覚えた英単語の意味とプログラミングに使われる英単語の意味の違いを学んだり、文学的な表現で使われる単語の意味と、理科系の文書に現れる単語の意味の違いを理解することも、知識を得るだけでなく文化の違いをも理解する総合的な学習になってくる。」²⁾

こんなことが実現できないかどうか検討し、実際にできたときの喜びは、次のステップへと進んでいく力となる。成長するための道具として、今後も様々な気づきを与えてくれる教材開発をしていきたい。生徒が何を表現し、考え、形あるものを制作し、納得できるようサポートしていきたいところである。

参考 URL

- 1) <http://terrapin.sourceforge.net/>
- 2) 藤川清一「Logo & Physics」,
元東京都立晴海総合高等学校,2003, <http://web.archive.org/web/20030726181131/http://fjkw.zive.net/logo/>