

Flask を用いた Python ベースの Web プログラミング教材開発

Development of Teaching Materials for Python-based Web Programming using Flask

幸谷智紀*

Tomonori KOUYA

Abstract: In 2021, we have started Python-based programming education in our class of "Computer Practical Exercise I" for senior undergraduate students of department of computer system. Our prepared Note PCs, not student's personal Note PC, and our original educational text on Web are used in order to learn how to build web sites with Flask framework. In this paper, we report the meaning of our education and the educational materials. Next, authentication system using facial recognition library is described and we conclude our future plan with it.

1. 初めに

1989年のWorld Wide Web(以下「Web」と略記)の発明以来、TCP/IP通信網を基盤とするThe Internet(以下「ネット」と略記)は、Webを利用したいユーザ数の増加と共に世界中に張り巡らされてきた。現在ではスマートフォンからの利用も増え、回線容量のアップを求める声は引きも切らない。TCP上の主要プロトコルはWebが用いるHTTP/HTTPS以外にも種々のものがあるが、最低限、Webクライアントであるブラウザさえ動作するデバイスがあれば手軽にわかりやすいインターフェースを提供するWebの利用は増えることすらあれ、今後も減ることは想定し難い。少なくとも、シビアリアルタイム応答性が求められないリモート処理のニーズはWebが担っていくことに疑いはない。また、現状でもYouTube, Netflix, Amazon Prime, ニコニコ動画・生放送等の動画配信コンテンツさえWebで問題なく使用できていることを考えると、回線の物理層の容量増加とContents Delivery Network(以下「CDN」と略記)との併用で、リアルタイム応答を求める用途すら、多くをWebが担っていく未来が見えていけると言える。従って、情報基盤技術の教育を柱とする大学教育に置いて、Web関連のプログラミング技法の教育は必要不可欠である。

とはいえ、現在のWeb技術は層が深く、全ての要素技術を網羅的に学ぶことは現実的ではない。HTML/CSSに限っても、タグや書式指定の意味付けを知るだけなら難しいものではないが、デザイン技法に通じていない限り、Bootstrap⁸⁾等、人間工学的に優れ、利用実績のあるフレームワーク以上のものを構築することは難しい。また、JavaScriptはECMAScriptという国際規格として確立したSchemaベースのプログラミング言語であり、静的なWebページに埋め込まれた用途だけでなく、Node.jsのようにデバイスを操作するスタンドアローンの開発環境としての利活用も活発化している。

我々は長年TCP/IPネットワークに関する教育を、主として実験や卒研で行っており、2018年からは独自のPHP, MySQLをベースとしたサーバサイドのWebプログラミング教材を作成し、使用している³⁾。また、2019年度には

Node.js上に構築されたApache CordovaでJavaScriptベースのSingle Page Application(SPA)の開発を実験テーマとして実施しており、Webコンテンツからスマートフォンのネイティブアプリケーションまで包含したクロスプラットフォーム開発が可能であることを学部教育を通じて学生に伝えている⁴⁾。

このように、我々が本学で取り組んできたWebプログラミング教育は決して最先端を狙ったものではなく、Web開発のメインストリームに添うものであり、その流れに沿って、我々は数年単位で実験・卒研テーマを変更している。2021年度から「Pythonプログラミング」をテーマに、コンピュータシステム学科3年生向けの「コンピュータシステム実践演習1」において実習を行うようにしたのもその一環である。

本論文では、まず、その意義と内容を説明する。次に今回開発した顔認証ライブラリを用いたログインシステムについて解説し、次年度以降の教材として活用する意義について議論する。

2. PythonとFlask

Pythonはソースプログラム(スクリプト)をインタプリタが解釈して実行するタイプのコンピュータ言語体系の一つである。Version 2は2000年にリリースされたものの、利用者の増加と共に言語としての不十分さが指摘され、2006年には後方互換性のないVersion 3がリリースされた。現在ではこのVersion 3系統のPythonを使用することを前提としており、2021年1月現在最新版はVersion 3.10.2である。2006年以来、MITではPythonによるプログラミング教育が開始され、Pythonの開発と普及に務めるPSF(The Python Software Foundation)⁹⁾の強力なライブラリ開発サポート体制のおかげか、科学技術計算、テキスト処理、機械学習、深層学習等々、様々なパッケージが無料で提供されており、複雑な処理もパッケージの組み合わせで解決できることが多いことから、現在では世界中で活用されるようになってきている。このように、Pythonは単なる言語というより、Pythonから使用できる様々なモジュールの集まり、すなわち「Python環境体系(ecosystem)」を形成しているといえる。これらのモジュールは、pipなどのツールを使って簡単に導入するこ

2022年5月27日受理

* 情報学部 コンピュータシステム学科

とができるようになる。

Flask¹⁰⁾ は、Python で記述できる Web 開発ツールの一つである。コントローラーとなる最小限の Python コードを記述すれば良いので、比較的小さい単機能の Web サービスを作るには Flask が向いている。参考までに Flask のトップページを Fig.1 に上げておく。

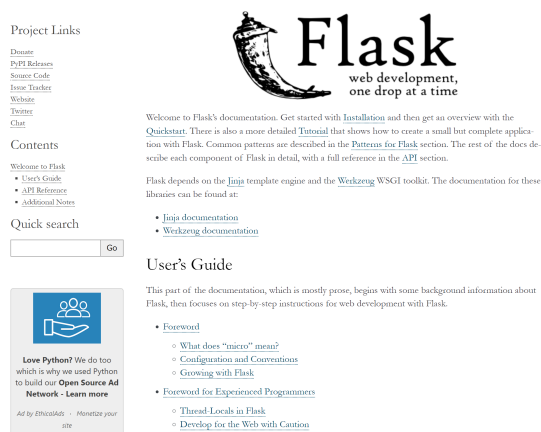


Fig.1 Flask のトップページ¹⁰⁾

現在の多くの Web 開発用フレームワークは、GUI の構成要素である HTML + CSS をフロントエンドとし、ユーザの入力に応じた動作をコントローラーに担わせることで動作するようになっている。Flask では前者を Jinja(ジンジャ)という HTML テンプレートエンジンで、後者を WSGI(Web Service Gateway Interface) をサポートする Werkzeug(ワークツェウグ(ドイツ語), Work+Stuff=Tool の意味)が担っている。Flask は両者の機能を融合し、統一的な Web サービスを作ることができるようになる。また、追加のパッケージを使うことで、より高度な機能を記述することができる。

Flask も Python パッケージの一種なので、pip コマンドで簡単にインストールすることができる。導入後、作成する Web サイトごとにフォルダを掘り、その中でコントローラと HTML テンプレートを作成し、目指す Web サービスを作り上げていく。

3. Python ベースの Web プログラミング教材

最初に述べたように、Web 開発技法は日々変化しており、特に基盤技術を使いやすくまとめ上げてより複雑なサービスの開発を可能とするフレームワークの主導権争いは耐えることがない。

Web クライアントサイドでは、CSS レベルでも Bootstrap 以外に大小様々なデザインテンプレートが提供されている。JavaScript では Vue.js, React, Angular といった Web/Node.js 上の開発フレームワークが定番となっており、複数の HTML ファイルの遷移を前提としたインターフェースから、1 ページ内で JavaScript による SPA として動的にインターフェースを変更するものが増えている。

Web サーバサイドでも、PHP, Java, C#, Python, Ruby 等の言語の上に構築されたフレームワークを使うことが標準となって久しい。PHP では CakePHP や Laravel, Java で

は Spring, Python では前述の Flask や Django, Ruby では Ruby on Rails といったものを利用している Web サーバサイドのサービス構築が定番である。我々が使用している PHP ベースの教材はフレームワークを使用しない、完全にスクラッチからのサービス構築を行うものになっているが、これは要素技術を学ぶには適しているものの、ある程度大規模なシステムを構築するための「大局観」を培うには向いていない。現在ではむしろ、複雑なシステムを、これだけ普及し開発・使用実績のあるフレームワークの力を借りることで簡単に実現できるという体験をさせることが望ましい。

このような判断により、現状最も活発にコミュニティが形成されている Python エコシステム上で動作し、手軽に利用可能な Flask を用いた、サーバサイドの Web プログラミング教育が適当であるという結論に至り、2020 年度の卒研⁷⁾の指導を行いつつ Flask の利用技法を習得し、教材を作成することができた (Fig.2)。

教材作成にあたっては、本教材作成と並行して執筆した数値計算プログラミングテキスト⁵⁾のソースプログラム⁶⁾も利用している。以下、2021 年度「コンピュータシステム実践演習 1」の実習環境と、使用した教材の詳細について述べる。

3.1 実習環境について

実習環境としては Windows 10 の Note PC を受講生一人 1 台配布できるよう準備を行った。一教員の予算の範囲内で買い揃えてきた Celeron マシン 7 台に加え、教員個人が使用してきた新旧の Note PC 3 台を揃えたものの、Celeron マシンのうち 3 台が故障のため電源すら入らない状態になり、急遽、B5 サイズの安い Note PC を 3 台新規購入して 2021 年度前期の「コンピュータシステム実践演習 1」を乗り切ることができた。なお、2021 年度の教育プロジェクトとして獲得した予算は、故障したものと同タイプ Note PC の代替機購入に利用した。

Windows 10 であれば、学生個人の Note PC を利用する手もあるが、Python 環境だけでなく、後述する SQLite コマンドの利用が必要であり、テキストエディタについても Visual Code Editor で統一することで環境を統一することでサポートを用意する目的もあり、教員側で実験用 Note PC を準備することにし、インストール作業に伴う混乱と時間の浪費を防ぐことが可能となった。また、学生個々の Note PC は作成したスクリプトや動作時のスナップショットファイルのバックアップとして活用することとし、作業で使用したファイルが紛失することを防ぐことができた。

なお、実習用 Note PC には後片付け用のバッチファイルを用意し、実験終了後は C ドライブ下の Flask ファイルを自動的に消去できるようにした。

3.2 最初の Flask アプリケーション

「最初の Flask アプリケーション」の解説ページを Fig.3 に示す。

受講生の Web プログラミングスキルは相当のばらつきがあるものの、概して HTML/CSS については大体理解していると思われた。実際、最初の Flask アプリケーションである「hellow」については概ね戸惑うことなく実行できていた (Fig.4 上)。

次に、Flask でフォーム入力値を利用する方法を習得する



Fig.2 Flask プログラミング実習のためのフォルダ構成 (左) とトップ画面 (右)



Fig.3 「最初の Flask アプリケーション」解説ページ

ため、一部を白抜きにしてスクリプトの意味付けを理解するように仕向けたが、口頭でヒントを出すことで、概ねこれもクリアできていた。

3.3 SQLite を用いた住所録

次に Web プログラミングでは定番の、SQL データベースを利用した住所録の作成を行わせた。その解説ページを Fig.5 に示す。

今回作成した教材の中では、受講生が一番時間を要したものである。DBMS には何を使うか考えたが、XAMPP for Windows の MySQL+phpMyAdmin の利用よりは、ファイル一つがデータベースとなる SQLite の利用が簡易で良いと判断し、Windows 用のコマンドラインツールをインストールしてデータベースファイルとテーブルの作成を CUI で行うことにした。

当初作成した Web インターフェースは Fig.6 の左図に示すように箇条書きのメニュー形式で、複数の HTML テンプレートを使い分ける必要があったものの、どの機能も実装は容易いようにしてある。しかし、ユーザ側としてはデータの状態が分かりづらいという点で使い勝手に難があり、TA よりから改良版の具体的な提案があり、簡素でわかりやすいテーブル型の HTML テンプレート (Fig.6 右図) であったことから、2 グループ目以降にはこれを採用し、データベー

最初の Flask アプリケーション

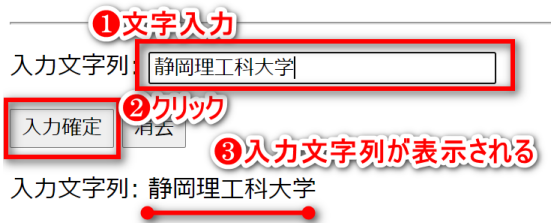
自分の学籍番号 氏名

ようこそ、Flaskの世界へ！

Copyright (c) 20xx 学籍番号, 氏名

フォーム入力の出力

自分の学籍番号 氏名



Copyright (c) 20xx 学籍番号, 氏名

Fig.4 「hellow」実行画面 (上) とフォーム入力値の利用画面 (下)

ス住所録作成のための実習時間を短縮することができた。

後述するように、次年度以降にはこのアプリケーションに対して認証機能を持たせていく予定である。

3.4 関数グラフ

Python で利用できる、初等関数と四則演算の組み合わせで表現できる実 1 変数関数のグラフを描画する Web アプリケーションを作成するのが Fig.7 のページで解説する「関数グラフ」である。



準備

SQLiteがその環境にインストールされていること、そして、SQLiteを使用するためのPythonモジュールである'sqlite'がPython環境にインストールされていることを確認しておいて下さい。入っていないようでしたらインストールしておきましょう。

次に、住所録を作成するフォルダ(ディレクトリ)を作成します。今回は「address」フォルダを作成し、ここに全てのファイルを保存しておくようにします。

データベースを作成する

下記の書録で住所録用のデータベースである「address.db」を作成します。

```
PS C:\Users\kikouya\Google ドライブ\kna\python3\source\flask\address> sqlite3 address.db
SQLite version 3.35.4 2021-04-02 15:20:15
Enter ".help" for usage hints.
```

Fig.5 「SQLiteを用いた住所録」解説ページ

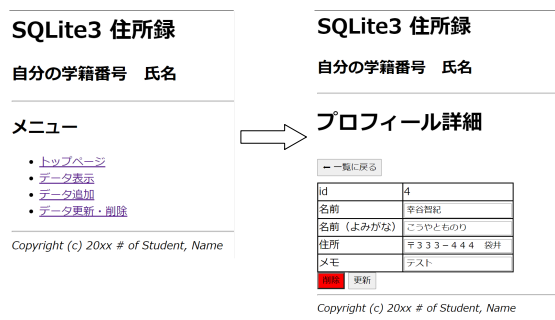


Fig.6 「SQLiteを用いた住所録」実行画面：当初作成版(左)、改良版(右)

ここでは Matplotlib モジュールを用いて描画した関数グラフを PNG ファイルとして出力し、その結果を HTML ファイルに埋め込んで出力している。以下、常微分方程式や偏微分方程式の解の描画でもこの方法を踏襲している。

3.5 常微分方程式の解

2021 年はコロナ禍ということもあり、感染症数理モデルの基本である SIR モデルを用いた常微分方程式の解を求めてグラフに描画するアプリケーションを Fig.8 に示す。

テキストには結果を載せていないが、解説の際に実行しながら、パラメータの違いによって感染者数の推移が変化することを説明している。

3.6 偏微分方程式の解法

差分法で 1 次元熱方程式を解き、その解を描画する Web アプリケーションを Fig.9 で解説した。

偏微分方程式そのものに馴染みがなく、差分化した結果が疎行列になり、SciPy の sparse ライブラリを使用していることから、内容に関しては理解が及ばなかったと吐露するレポートが多かった。

4. 顔認証ログイン機能の開発

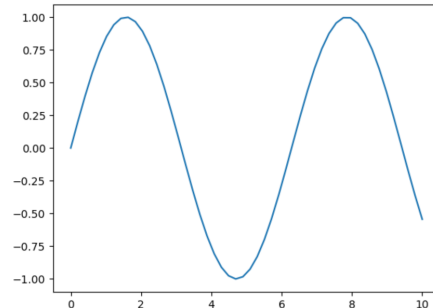
2021 年度は初回ということもあり、教材を適宜改変しながらも、当初の予定通り実施できたが、関数グラフ以降の数

関数グラフの描画

自分の学籍番号 氏名

関数

function: sin(x)



x : [0, 10]
 y =
 x = [,]

Copyright (c) 20xx # of Student Name

Fig.7 $x \in [0, 10]$ で $\sin x$ を描画した「関数グラフ」の実行例



準備

常微分方程式の初期値問題例: SIRモデル

総人口 N 人の閉じたコミュニティにおいて、伝染病が発生したとする。感染の強さを表す伝染係数が r 、平均感染期間が $1/a$ (日) とし、時刻 t における感染者数を $I(t)$ 、これから感染する可能性のある人数を $S(t)$ 、伝染病が治ったが死亡した人数を $R(t)$ とすると、必ず $I(t) + S(t) + R(t) = N$ が成立する。

このとき、 $S(t)$ 、 $I(t)$ 、 $R(t)$ の導関数、すなわち、変化率は次のように決まっているものとする。

- $S'(t)$ は $S(t)$ と $I(t)$ の積に比例して減っていく。
- $I'(t)$ は $S(t)$ と $I(t)$ の積に比例して増えるが、 $I(t)$ が多くなるとその分減少する。
- $R'(t)$ は $I(t)$ に比例して増えていく。

これを3次元の常微分方程式として表現すると下記のようなになる。

$$\frac{d}{dt} \begin{bmatrix} S \\ I \\ R \end{bmatrix} = \begin{bmatrix} -rSI \\ rSI - aI \\ aI \end{bmatrix}$$

初期条件: $t_0 = 0$ として

$$[S(0) \ I(0) \ R(0)]^T = [S_0 \ I_0 \ R_0]^T$$

とする。

Fig.8 「常微分方程式の解」解説ページ

値計算を題材とした Web プログラミングを行うのは少し無理があると感じた。3 年生向けの専門科目である「数値解析 1」は受講者が 10 数人程度と少なく、コンピュータシステム学科全体の 1/5 以下であることを考えると、数理的な内容よりは実用的なアプリケーションを題材としたものが良い。

そこで、次年度以降は二番目に実行した SQL データベースアプリケーションに機能を追加していく方向で教材を改変することを考えている。Python エコシステム環境であることの利点を最大限生かすべく、深層学習・機械学習の機能を取り込むことも面白いと思う、TensorFlow で学習した



Fig.9 「偏微分方程式の解法」解説ページ

結果を用いた顔認証ライブラリ Face Recognition¹¹⁾ を用いた顔認証ログインシステムを作成し、あわよくば教材として使えないか検討を行った。実際に作成した顔認証ログインシステムのブロック図と実行例を Fig.10 に示す。

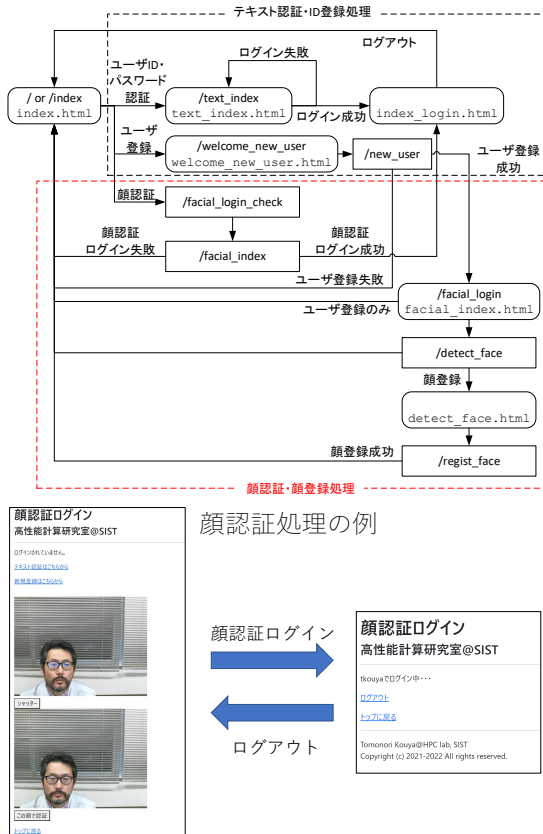


Fig.10 顔認証ログイン機能のフロー(上)と実行例(下)

ブロック図に示す通り、この顔認証ログインシステムは、通常のテキストベースの ID・パスワード認証の上に

構築されている。新規ユーザは最初にテキストベースの ID・パスワードを登録し、成功した暁には自分の顔を Face Recognition の機能を用いて登録できるようになっている。従って、顔認証に失敗した場合でも、テキストベースのログインが可能である。顔認証できた場合は、自動的に ID の特定が可能である。

本機能を実装するにあたり、さまざまな情報を検索してみたが、顔認証の機能を用いた例はたくさん見つかるもの、Web サイトのログイン機能として活用したものは見当たらなかったことから、スクラッチから作成することになった。

顔認証部分は、まず、Media Capture API¹²⁾ の機能を用いて JavaScript ベースのリアルタイム動画を取り込み、静止画をサーバに送信し、一時保存する。それを Face Recognition が取り込んで顔を判断し、既存のユーザの顔と比較照合してユーザ ID を特定する。大体うまくいっているようだが、コロナ禍前の開発であったためか、マスクをしたまでは認証精度が著しく落ちることが判明している。

また、教材としては複雑でコード量が多いことから、まずはテキストベースの認証部分を作成し、顔認証部分は別のアプリケーションとして取り上げた方が良く考えている。

5. まとめと今後の展開

本教材を作成することで、2021 年度の「コンピュータシステム実践演習 1」では Flask による Python ベースのプログラミング演習が可能であることを確認できた。2022 年現在の実践演習 1 は、3 週 6 コマで一つの実験テーマを体験することになっている。2 週 6 コマで実施していたかつての学生実験時よりも、同じコマ数ではあるが比較的時間に余裕をもって取り組む体制ができてはいるが、いかにせん体験以上の、身になる学習ができているかということについては疑問である。それでも Python に触れたことのある受講生が少数派である現状で、データベースや科学技術シミュレーションを題材とした Web プログラミングを体験する意味はあると言える。

今のところ卒業研究として Flask をテーマとしたものは本研究室では 2 件しかないが、最初の卒研ではかなり調べながら少しずつ完成していったものが、この教材を履修した後に実施中の 2 件目の卒研では、データベースの構築まで 2 週間程度でこぎ着けている。その意味では、卒研につながる Flask のお手本として本教材があることの意義は大きいと思われる。

今後は、引き続き教材の改良に努めるとともに、卒研につながる「コンピュータシステム実践演習 2」用の教材としても使用できるよう、さらなる教材の拡大を考えている。

謝辞

本教材作成は静岡理科大学教育プロジェクトの助成を受けて実施されたものである。関係各位に感謝する。教材作成にあたっては、卒研テーマ選定以来の指導教員の無茶振りに耐えて完成させた山口朋也君に感謝する。また、2021 年度実験では SA として井上迪也君が、至らぬ指導教員のフォローを行う追加文書まで作成し、懇切丁寧な指導を行ってくれた。この献身にも感謝したい。また、丁寧な査読を行って頂いた未知の査読者にも感謝する。

参考文献

- 1) Web教材「Flaskを用いたWebアプリケーション作成」,
<https://cs-tklab.na-inet.jp/flask/>
- 2) Web教材「Webアプリケーション開発入門」,
<https://cs-tklab.na-inet.jp/phpdb/>
- 3) 幸谷智紀, 学生による学生のためのWebプログラミング教材開発, 静岡理科大学紀要, Vol.25, pp.169-174, 2018.
- 4) 幸谷智紀, Apache Cordovaを用いたハイブリッドアプリケーション開発教材, 静岡理科大学紀要, Vol.27, pp.73-78, 2019.
- 5) 幸谷智紀, Python 数値計算プログラミング, 講談社, 2021.
- 6) 幸谷智紀, 「Python 数値計算プログラミング」サンプルスクリプト, <https://github.com/tkouya/inapy>
- 7) 山口 朋也, 「Python と Flask を用いた二階偏微分方程式求解 WEB アプリの開発」, 2020 年度卒業研究.
- 8) Bootstrap3 公式サイト, <https://getbootstrap.com/>
- 9) The Python Software Foundation, <https://python.org/>
- 10) Flask, <https://flask.palletsprojects.com/>
- 11) https://github.com/ageitgey/face_recognition
- 12) <https://w3c.github.io/mediacapture-main/>
- 13) <https://flask-login.readthedocs.io/en/latest/>