

指導資料



鹿児島県総合教育センター
令和2年10月発行

情報教育 第155号

対象
校種

高等学校
特別支援学校



「Python」を活用したデータ分析 —高等学校「情報Ⅰ」でのプログラミング②—

Society5.0と呼ばれる新たな時代においては、ビッグデータを有効に活用し意思決定を行うことが、これまで以上に重要になる。インターネット上のサービスやビッグデータへのアクセス機能を提供するWebAPI^{*1}の活用により、データサイエンス^{*2}への興味・関心を高める指導の工夫例を紹介する。

1 はじめに

Webサイトを閲覧していると、過去に閲覧したサイトの広告、類似した商品やサービスの広告が表示されることがある。これは「追跡型広告」、「行動ターゲティング広告」などと呼ばれ、閲覧履歴をもとにアカウントの所有者が興味・関心を抱きそうな広告を自動的に配信するものである。また、カーナビゲーションシステムでは、インターネットを通じて交通情報を随時収集し、交通状況に応じて案内ルートを自動的に変更するものがある。これまでの情報社会（Society 4.0）では、膨大なデータから必要な情報を自ら収集してきた。今後、到来するSociety 5.0と言われる超スマート社会では、人間の能力を超えたAIが膨大なビッグデータを解析し、その結果についてロボットなどを通して人間にフィードバックすることになる（図1）。すなわち、大量のデータを処理し、分析することで、何らかの意味のある情報や価値を導き出すデータサイエンスの重要性が高まっているのである。

新学習指導要領では、情報活用能力を「学習の基盤となる資質・能力」と位置付け、その育成のための学習活動の充実を図ることとした。また、今年度から小学校ではプログラミ

ング教育が始まり、高等学校では共通必修修科目「情報Ⅰ」（令和4年度開始）においてプログラミングを全ての高校生が学ぶことになる。そのため、指導資料情報教育第153号「意外と簡単！『Python』—高等学校「情報Ⅰ」でのプログラミング—」において、高等学校で扱うプログラミング言語として注目されている「Python」の基礎と活用例について紹介した。本稿では、「Python」を使用言語として、WebAPIの機能を活用する方法を示し、生徒たちのデータサイエンスへの興味・関心を高めるための工夫例を紹介する。



図1 Society 5.0「科学技術イノベーションが拓く新たな社会」説明資料（内閣府Webサイト）

*1) 2(1)参照

*2) データから科学的及び社会に有益な知見を引き出す学問

2 WebAPIの基本的な考え方

(1) WebAPIとは

API (Application Programming Interface) とは、「ある一つの機能に特化したプログラムで共有可能なもの」や「ソフトウェアの機能を共有する仕組み」のことである。よく使う機能がAPIとして用意されていれば、わざわざ一からプログラムを開発する必要はない。必要に応じてAPIを利用し、効率的にシステム開発が進められる。

WebAPIの場合、プログラムはWeb上に公開され、**図2**のように外部から呼び出して利用できる。様々な分野でWebAPIが公開され、多くは無料で利用可能だが、登録が必要な場合

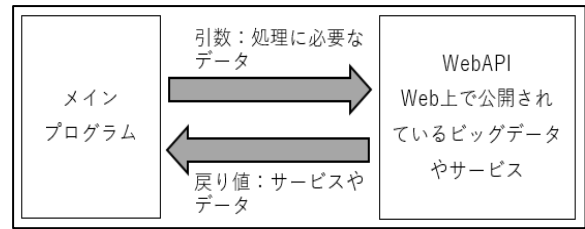


図2 WebAPI の概念

もある。

(2) 基本的なプログラムの例

WebAPIを活用した基本的なプログラムを示す。このプログラムで使用するWebAPIは、登録が不要なため容易に実行が可能であり、Web上のデータの取得を実感することができる。また、Pythonの基本的なコードや環境設定については、指導資料情報教育第153号を参照してほしい。

【処理内容】 「WebAPIを利用し、郵便番号からそれに対応する住所を表示する。」^{*3}

【処理条件】 ・必要なサービスやデータ：郵便番号に係るデータ

使用するWebAPIのURL 「<https://zipcloud.ibsnet.co.jp/api/search>」

・引数（処理に必要なデータ）：郵便番号 ⇒ パラメータ名「zipcode」

・戻り値（処理結果のデータ）：郵便番号に対応する住所

※検索結果が複数存在する場合は、以下の項目が配列として返される。

| フィールド名 | | 項目名 |
|---------|----------|---------|
| results | zipcode | 郵便番号 |
| | prefcode | 都道府県コード |
| | address1 | 都道府県名 |
| | address2 | 市区町村名 |
| | address3 | 町域名 |
| | kanal | 都道府県名カナ |
| | kana2 | 市区町村名カナ |
| | kana3 | 町域名カナ |

```
1 import requests } ①
2 import json
3 url = "http://zipcloud.ibsnet.co.jp/api/search" ②
4 yban = input("郵便番号:") ③
5 param = {"zipcode": yban} ④
6 res = requests.get(url,params=param) ⑤
7 response = json.loads(res.text) ⑥
8 address = response["results"][0] ⑦
9 print(address["address1"] + address["address2"] + address["address3"]) ⑧
```

【実行結果】

実行中: 入力検索.py

郵便番号:890-0053
鹿児島県鹿児島市中央町
>>>

引数

戻り値

- ① PythonでWebAPIを扱う場合に必要なrequestモジュールとjsonモジュールの取り込み
- ② 使用するWebAPIのURLを指定
- ③ コンソールから入力された郵便番号をybanへ
- ④ WebAPIの引数として郵便番号を指定
- ⑤ WebAPIにおいて郵便番号（引数）をもとに検索、処理結果（戻り値）をresへ
- ⑥ 戻り値をjsonデータからPythonオブジェクトへ変換
- ⑦ 先頭のデータ（["results"][0]）をaddressへ ※対応する住所が複数存在した場合のため
- ⑧ 住所データを表示

*3) 文部科学省「高等学校情報科『情報Ⅰ』教員研修用教材」の例示を基に作成

3 WebAPI活用の具体例

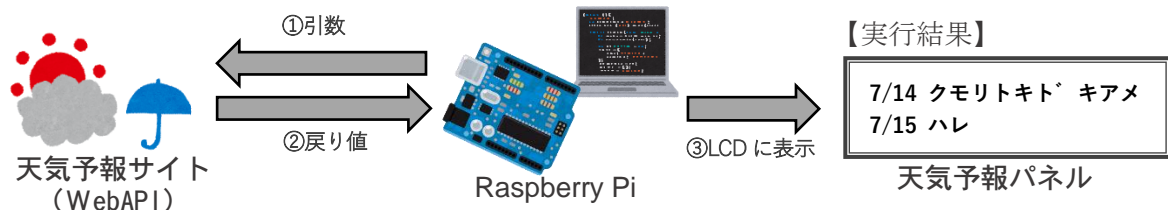
本稿 2 で示した、郵便番号からそれに対応する住所を表示するプログラムでは、WebAPI を活用する意義はそれほど強く感じず、自分のPC端末に住所データを保存し検索することや、住所検索用のWebサイトを利用することで十分と考えるかもしれない。それは、データ量が膨大でない、データの更新が頻繁でない、

そして処理が複雑でないためである。しかし、情報社会が進展する中、ビッグデータと言われる多種多様で膨大なデータは日々刻々と蓄積され、目まぐるしく変化している。そこで、Society5.0の情報システムの入口として、人間が必要とする情報を自動的に提供するプログラムの作成に取り組んでほしい。ここでは、天気情報に関するWebAPIの活用と機器の制御を行うプログラムの具体例を紹介する。

【処理内容】「鹿児島県の今日と明日の天気予報および最高・最低気温をLCDパネルに表示する。」
(1時間ごとに更新)

【使用機材】・Raspberry Pi 4 Model B (ARM プロセッサ搭載のシングルボードコンピュータ)
・LCD(LCM1602C) ・ブレッドボード ・半固定抵抗 ・ジャンパ線

※ 鹿児島県立隼人工業高等学校 永倉紀仁教諭
高等学校情報教育継続研修における作成プログラム



【プログラム (一部省略)】 #からの文はプログラムに関する作成者の注釈文

(1) プログラムに必要なモジュールの取り込み

```
1  #!/usr/bin/python
2  # Example using a character LCD connected to a Raspberry Pi or BeagleBone Black.
3  import time
4  import datetime
5  import requests
6  import json
7  import Adafruit_CharLCD as LCD
```

(以下省略。LCD に関する設定を 38 行目まで記述)

(2) リストweatherCharに天気ワードを16進数の文字コードで、リストweatherNumに天気予報パターンをリストweatherCharの添字の組み合わせで設定

```
39 # 天気ワード7つ (ハレ、クモリなど) を 16 進数の文字コードで設定
40 weatherChar = (['¥xca', '¥xda'], ['¥xb8', '¥xd3', '¥xd8'], ['¥xb1', '¥xd2'],
41                ['¥xd5', '¥xb7'], ['¥xc4', '¥xb7', '¥xc4', '¥xde', '¥xb7'],
42                ['¥xc9', '¥xc1'], ['¥xc2', '¥xd6', '¥xb2'])
43 weatherNum = ["0", "041", "042", "043", "051", "052", "053",
44               "1", "140", "142", "143", "150", "152", "153",
45               "2", "240", "241", "243", "250", "251", "253", "62",
46               "3", "340", "341", "342", "350", "351", "352", "63"]
```

「アメ」を表す

「クモリノチユキ」を表す

(3) WebAPIを利用する手続きを実行するget関数の作成

```
48 def get():
49     # 天気情報の取得 (「livedoor 天気」から データフォーマット:json)
50     global url
51     global locale
52     global data
53
54     url = 'http://weather.livedoor.com/forecast/Webservice/json/v1'
55     locale = {'city': '460010'}
56     data = requests.get(url, params = locale).json()
57     print(datetime.datetime.now())
58     print('60 分で更新')
```

関数の定義




WebAPI の URL を指定

引数として鹿児島の番号を指定

現在の日時を表示

戻り値を data へ

(4) 天気予報をLCDパネルに表示

| | | |
|--|---|--|
| <pre> 61 while True: 62 get() 63 for i in range(4 * 60): 64 print(str(i + 1).rjust(3) + '回目') 65 lcd.clear() 66 67 # タイトル表示: Weather forecast テンキヨホウ ヒョウジ 68 lcd.message(' ¥x57¥x65¥x61¥x74¥x68¥x65¥x72¥xa0¥x66¥x6f¥x72¥x65¥x63¥x61¥x73¥x74') 69 lcd.message(' ¥n¥xc3¥xdd¥xb7¥xd6¥x65¥xb3¥xa0¥xcb¥xae¥xb3¥x65¥x63¥x61¥x73¥x74') 70 time.sleep(5) 71 lcd.clear() 72 lcd.show_cursor(0) 73 74 # 天気予報の表示処理 75 for forecast in data['forecasts'][0:2]: 76 weather = int(forecast['image']['url'][37:-4]) - 1 77 # 例) 'url': 'http://weather.livedoor.com/img/icon/17.gif', 'title': '雨時々曇' 78 # プロパティ url の左から 37 文字目以降右から 4 文字目まで (天気アイコンの番号 **.gif) を取得 79 month = int(forecast['date'][5:7]) # プロパティ date から月を取得 80 day = int(forecast['date'][8:10]) # プロパティ date から日を取得 81 lcd.message(' {:2d}'.format(month) + '/' + ' {:2d}'.format(day) + ' ') # 先に日付を表示 82 for i in weatherNum[weather]: 83 for txt in weatherChar[int(i)]: 84 lcd.message(txt) 85 lcd.message(' ¥n') 86 time.sleep(5) 87 lcd.clear() 88 89 # 最高・最低気温の表示処理 (以下省略。処理内に time.sleep(5)があるため、最高・最低気温の表示処理に 5 秒要する) </pre> | <p>get関数で表示内容の更新</p> <p>タイトル表示 5 秒+天気予報表示 5 秒+気温表示 5 秒=15 秒</p> <p>60 分間同じ内容の表示を繰り返す</p> <p>0 から 2 未満の要素 [0]今日のデータ [1]明日のデータ</p> <p>40 から 46 行目で設定した、天気コード・天気パターンを抽出・表示</p> |    |
|--|---|--|

このプログラムは県立隼人工業高等学校における「課題研究」の題材として制作されたものである。WebAPIを利用して大量の天気予報データから必要な情報を抽出することと、LCDパネルは表示できる文字数が限られるため、表示内容が簡潔になるようにデータを変換することが要求される。このような要求を把握し、これに応えるよう取り組むことにより、情報活用能力と問題発見・解決能力の育成を図ることができる。

情報活用能力は教科等横断的な視点に立った育成が求められている。この事例は専門教科におけるものであり、WebAPIの利用と外部接続した機器へのデータ出力をするため、手続きがやや複雑になっている。しかし、小学校において実際に機器を動作させるフィジカルプログラミングを体験し、中学校においてプログラムによる計測・制御とネットワークを利用した双方向性のあるコンテンツのプログラミングを学習したことを生かして、このようなシステムの構築を目指す探究的な学習の場面が学科に関わらずあるべきだと考える。

4 おわりに

WebAPIの中には、生徒の興味を引くものが多くある。顔認識、言語や画像解析などAIを使えるAPI、地図表示や経路案内など日常的に活

用している利便性の高いAPIまでである。また、オープンデータとして活用が推進されている政府統計の総合窓口(e-Stat)で提供している統計データについても、API機能が提供され活用可能である。Pythonのインストールを含め、様々な環境設定が必要になるが、生徒の知的好奇心を満足させる環境作りをしてほしい。今後、PC端末が随時更新され、Webカメラやマイクを内蔵したものになっていくとAPIの活用の可能性は更に高まるだろう。

生徒には、その環境の中で行われるプログラミングやデータを活用する体験により、数多くのシステムで共通して使用されている機能はAPIとして提供されていることを知り、効果的にプログラムと連携させてほしい。特に、WebAPIを活用すると外部のサービスやビッグデータを使用することができ、様々なシステム開発の場面で活用されていることを生徒が理解し、プログラミングとデータサイエンスへの興味・関心を高め、Society5.0社会に向けて大きく踏み出すことを期待している。

ー引用・参考文献ー

- 文部科学省「高等学校学習指導要領解説 情報編」平成30年 7 月
- 文部科学省「高等学校情報科『情報Ⅰ』 教員研修用教材」平成31年 3 月29日
(情報教育研修課 青木 誠)