

Webから天気情報を取得しよう

インターネット上では、さまざまな情報が公開されています。それが、プログラミングによって、手軽に取得できるように工夫されている場合があります。

例えば、筆者が運営する「クジラWeb API」では、**JSON**データで天気予報の情報を手軽に取得できるようになっています。



図5-2-1：筆者が運営する「クジラWeb API」のサイト

そして、以下を見ると、どのようなURLにアクセスすれば、どのような情報がどんな形式で得られるのか詳しく情報が掲載されています。

●クジラWeb API > 使い方(APIのエンドポイント)

[URL] <https://api.aoikujira.com/index.php?tenki&show#hb3802071>

例えば、以下のURLにアクセスすると、JSONというデータ形式で東京の天気情報を取得できます。ちなみに、JSON形式とは、PythonをはじめJavaScriptなど、プログラミング言語でネット上のデータを受け渡すのによく使われるデータ形式です。

●東京の天気情報がJSON形式で取得できる

[URL] <https://api.aoikujira.com/tenki/week.php?fmt=json&city=319>

以下は、WebブラウザでURLにアクセスしたところです。生のJSON形式なので、よく分からない文字列が表示されますが、これをPythonで取得して、エンコードすれば天気情報を得ることができます。

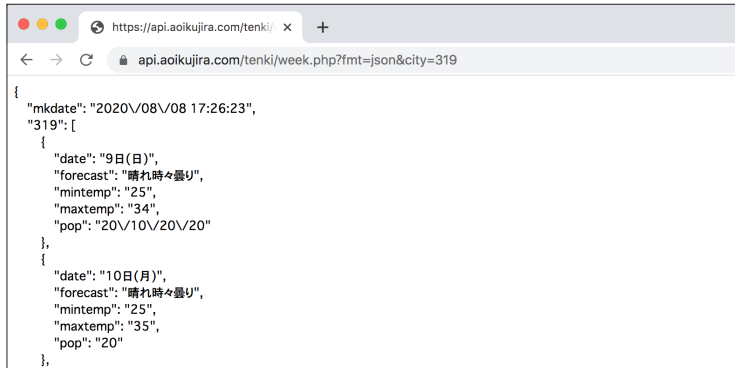


図5-2-2 : WebブラウザでURLにアクセスしたところ - JSON形式で天気情報を取得

ちなみに、URLの末尾の「city=319」の部分を「city=306」と書き換えると札幌の天気情報を取得することができます。また、「city=329」に変えると名古屋の天気になります。この都市のID情報は、以下に記されています。

●クジラWeb API > 対応都市一覧

[URL] <https://api.aoikujira.com/index.php?tenki&show#h39c8f4bf>

▶ 天気情報を表示するプログラムを作ろう

それでは、Weather Hacksからデータを取得して、画面に出力してみしましょう。以下は、Webから天気予報情報を取得するプログラムです。

IDLEのテキストエディタに入力 - サンプルプログラム : src/ch5/show_tenki.py

```

01 # 東京の天気を JSON で取得できる URL --- (*1)
02 url = "https://api.aoikujira.com/tenki/week.php?fmt=json"
03 url += "&city=319"
04
05 # Web から天気情報を取得する --- (*2)
06 import urllib.request as req
07 res = req.urlopen(url)
08 json_data = res.read()
09
10 # JSON データを Python のデータ型 (今回は辞書型) に変換 --- (*3)
11 import json
12 data = json.loads(json_data)
13
14 # 結果を表示 --- (*4)
15 for row in data["319"]: # 「319」を1行ずつ読み込む
16     date = row["date"] # 読み込んだ行の「date」の値を代入
17     forecast = row["forecast"] # 読み込んだ行の「forecast」の値を代入
18     print(date + " : " + forecast)

```

IDLEで実行してみましょう。実行結果は、その日の天気によって変わります。

プログラムを見てみましょう。(※1)の部分では、天気予報のJSONデータが取得できるURLを指定します。2行目では、

都市IDを指定することで、別の都市の天気予報を取得できます。

それで、(※2)の部分では、Webからデータを取得します。ここが今回のプログラムのポイントですが、モジュール「`urllib.request`」を別名のreqという名前で利用できるように宣言しています。そして、その`urlopen()`関数を使ってWebサイトに接続し、`read()`メソッドでデータを取得します。

そして、(※3)の部分で、`json.loads()`関数を利用して、取得したJSONデータをPythonのデータ型に変換します。最後の(※4)の部分で、データの中から、今日の天気と明日の天気を抽出して表示します。

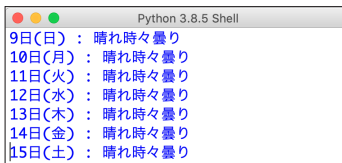
(※3)で取得したdataには、以下のような内容が入っています。

```
{
  'mkdate': '2020/08/08 17:03:52',
  '319': [{
    'date': '9日(日)',
    'forecast': '晴れ時々曇り',
    'mintemp': '25',
    'maxtemp': '34',
    'pop': '20/10/20/20'
  },
  {
    'date': '10日(月)',
    'forecast': '晴れ時々曇り',
    ...
  },
  ...
}]
}
```

ですので、この中から、「date」の値や「forecast」の値を取り出して使っています。

🔍 再利用できそうな部分を関数にしよう

それでは、ここまで作った天気予報を表示するプログラムを、関数にまとめてみましょう。特に、Webからデータを取得する関数や、天気予報を取得する関数を作ってみましょう。



```
Python 3.8.5 Shell
9日(日) : 晴れ時々曇り
10日(月) : 晴れ時々曇り
11日(火) : 晴れ時々曇り
12日(水) : 晴れ時々曇り
13日(木) : 晴れ時々曇り
14日(金) : 晴れ時々曇り
15日(土) : 晴れ時々曇り
```

図5-2-3 : Webから天気予報を取得して表示するプログラムを実行したところ

ヒント !

もしSSLに関するエラーが表示されてしまったら、一度、IDLEを終了させて、コマンドラインから以下のコマンドを実行してください。

```
$ pip install
--upgrade certifi
```

IDLEのテキストエディタに入力 - サンプルプログラム : src/ch5/tenki.py

```

01  # 天気予報を取得するモジュール
02  # URL から JSON データをダウンロードする関数 --- (*1)
03  def get_json(url):
04      import urllib.request as req
05      import json
06      # データをダウンロード
07      res = req.urlopen(url)
08      json_data = res.read()
09      # JSON データを Python で扱えるよう読み込む
10      return json.loads(json_data)
11
12  # 都市 ID を取得することで天気情報を取得する関数 --- (*2)
13  def get_weather(city_id):
14      url = "https://api.aokujira.com/tenki/week.php?fmt=json"
15      url += "&city=" + str(city_id)
16      data = get_json(url)
17      s = ""
18      for row in data[str(city_id)]:
19          date = row["date"]
20          forecast = row["forecast"]
21          s += date + " : " + forecast + "\n"
22      return s
23
24  # モジュールとして利用されるかどうか判定
25  if __name__ == "__main__":
26      res = get_weather(319) # 東京の天気を取得 --- (*3)
27      print(res)

```

IDLEでプログラムを実行してみましょう。結果は実際の天気によって異なるでしょう。

```

9日(日) : 晴れ時々曇り
10日(月) : 晴れ時々曇り
...

```

どうでしょうか、プログラムの動きは、先ほどと全く同じです。しかし、機能ごとに関数を作って処理をまとめています。プログラムの処理(*1)の部分では、`get_json()`関数を定義します。これは、URLを指定してWebサイトからデータをダウンロードし、JSONデータをPythonで扱えるように読み込むものです。

そして、(*2)の部分では、都市IDを指定することで、天気情報を取得する、`get_weather()`関数を定義します。そして、都市IDをもとに作っ

たURLを引数にしてget_json()関数を呼びだし、JSONデータから天気情報を取得して関数の戻り値として返します。

最後に、(*3)の部分で、東京(都市IDが319)の天気情報を、get_weather()関数によって取得して結果を表示します。また、この部分では、そのプログラムがモジュールとして取り込まれたかどうかを判定しています。

[書式]モジュールかどうか確認する

```
if __name__ == "__main__":  
    # ここにモジュールでない場合の処理  
else:  
    # ここにモジュールの場合の処理
```

この部分の動きを調べるために、以下のようなファイルを作成して試してみましょう。

IDLEのテキストエディタに入力 - サンプルプログラム : src/ch5/test_tenki.py

```
01 import tenki  
02 print(tenki.get_weather(306)) # 札幌の天気
```

このプログラムをIDLEで実行してみましょう。札幌の天気予報が表示されます。

このプログラムでは、importで自作のtenkiモジュールを取り込んで、天気を取得します。ただし、この場合、tenki.pyの(*3)の部分は実行されません。都市ID「306」を引数として(*2)のget_weather()関数を実行して、結果を表示します。なお、通常のプログラムとして実行された場合、Pythonの特別関数である「__name__」には「__main__」が代入されます。一方、モジュールとしてtenki.pyが取り込まれるとき、__name__には、モジュール名「tenki」が自動的に設定されます。



まとめ

今回は、モジュールの作り方、そして、外部モジュールを利用する方法について詳しく見てみました。importの書き方によって、外部モジュールの機能をどのような形式で利用するのかを指定できる点も詳しく紹介しました。

【ここで学んだこと】

- モジュールの定義
- importの使い方
- 今日の天気をWebで調べるプログラム