

気象情報サイトの構築

地域ごとの気象情報を表示するウェブサイト構築します。各自のペースで進めましょう。このドキュメントや説明動画を見ながら、また、適宜ウェブ検索をして調べながら、実装を進めましょう。

完成させるサイトの例

1. 地域ごとの気象情報を表示するウェブサイト構築します。完成後は以下のURL先のサイトの動作をします。
<http://www.cm.is.ritsumei.ac.jp/class/isseds/webapp/weather.html>

本サイトは以下の流れで気象情報が表示されます

- a. 地域の入力: 入力欄に地域名(日本のTokyo、中国のShanghai)等
- b. 地域情報の送信: 「Show」ボタンをクリック
- c. 地域の気象情報の表示: 画面に地域の天気画像、温度、湿度、が表示される

Webサーバーのセットアップ

1. Python3 のインストールを行う
2. Python3 が PC にインストールされていることを確認してください。
3. このプロジェクトのファイルを保存するフォルダーを作成します。
4. 「weather.html」という名前のファイルを作成します。上記で作成したフォルダーの下に次の内容が含まれます。

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
  </head>
  <body>
    <h1>Weather Information</h1>
    
    <p>
      <b>City:</b> Kyoto <br/>
      <b>Temperature</b>: 30 &#8451; <br/>
      <b>Humidity</b>: 60 %
    </p>
  </body>
</html>
```

5. ターミナルのウィンドウを開き、現在のディレクトリを上記で作成したフォルダーに変更します。cdのコマンドを利用できます。
6. 次のコマンドで Python Web サーバーを起動します。WindowsとMacで異なるコマンドになる可能性があります。Web検索等で自分のPCに適切なコマンドを調べてみましょう(「python http サーバを起動 Windows」等で検索してみよう)
\$ python -m http.server --cgi 8000
7. 必要に応じて、Ctrl+C でサーバーを停止できます。
8. Webブラウザから以下のURLでWebサーバーにアクセスします。
<http://localhost:8000/weather.html>

気象情報を返すサーバー側コードの構築

1. 「cgi-bin」という名前のフォルダーを作成します。上で作成したフォルダーの下にあります。
2. 「weather.py」という名前のファイルを作成します。「cgi-bin」の下に次のコンテンツが含まれるようにします。このプログラムは、現時点ではダミーの気象情報をJSON形式で出力します。

```
#!/usr/bin/env python3

import json

print("Content-type: text/json\n")

weather = {"city": "Kyoto", "icon": "02d", "temp": 30, "humidity": 60}
print(json.dumps(weather))
```

3. 次のコマンドでこのプログラムを実行し、動作を確認します。
以下のURLから確認できます
<http://localhost:8000/cgi-bin/weather.py>

以下のコマンドをターミナルに入力しても確認できます。WindowsとMacで異なるコマンドになる可能性があります。Web検索等で自分のPCに適切なコマンドを調べてみましょう
\$ python cgi-bin/weather.py

HTML ファイルへの動的コンテンツの追加

1. 動的コンテンツを表示できるように HTML ファイルを以下の様に変更します (「span」タグを追加し、ページの動的部分のIDを指定します)。

```
<h1>Weather Information</h1>
<img id="icon" src="" width="64">
<p>
  <b>City:</b> <span id="city"> City </span><br/>
  <b>Temperature</b>: <span id="temp">0</span> &#8451; <br/>
  <b>Humidity</b>: <span id="humidity">0</span> %
</p>
```

2. また、都市名と表示するボタンを指定するためのテキスト入力領域も以下の様に追加します。

```
<p>
  <input type="text" id="city_input">
  <button type="submit" name="submit">Show</button>
</p>
```

3. このページがブラウザ上で正しく表示されるか確認してください。

サーバーから天気情報を受信し、ページ内容を変更します

1. 次の JavaScript コードを HTML ファイルの <head> 内に追加します。

```
<script type="text/javascript">
    var request = new XMLHttpRequest();
    function showWeatherInfo() {
        var url = "/cgi-bin/weather.py";
        url = encodeURIComponent(url);
        request.open("GET", url, true);
        request.onreadystatechange = updatePage;
        request.send(null);
    }
    function updatePage() {
        if (request.readyState == 4) {
            if (request.status == 200) {
                var replyDoc = JSON.parse(request.responseText);
                var city = replyDoc.city;
                var cityElm = document.getElementById("city");
                cityElm.removeChild(cityElm.firstChild);
                cityElm.appendChild(document.createTextNode(city));
                var icon = replyDoc.icon;
                var imgElm = document.getElementById("icon");
                imgElm.src = "http://openweathermap.org/img/w/" +
icon+".png";
            }
        }
    }
}
</script>
```

2. ボタン要素を次のように変更して、JavaScript コードの showWeatherInfo() 関数を呼び出します。
<button type="submit" name="submit" onclick="showWeatherInfo()">Show</button>
3. 「SHOW」のボタンをクリックし、サーバーから返された天気アイコンがページに表示されるかどうかを確認します。表示される気象情報はダミーのままです。
4. 温度と湿度の値を表示するように JavaScript コードを変更し、正しく動作するかどうかを確認します。
 - a. cityを表示するコードの部分を温度や湿度に直してみよう。具体的には、以下をhtmlファイルの適切な位置に張り付けます。

```
var temp = replyDoc.temp;
var tempElm = document.getElementById("temp");
tempElm.removeChild(tempElm.firstChild);
tempElm.appendChild(document.createTextNode(temp));

var humidity = replyDoc.humidity;
var humidityElm = document.getElementById("humidity");
humidityElm.removeChild(humidityElm.firstChild);
humidityElm.appendChild(document.createTextNode(humidity));
```

OpenWeatherMap APIから気象情報を取得する

1. 指定した都市の気象情報は、次のコードで取得できます。これらのコード行をサーバー側の Python プログラムに追加します。

```
import requests
import json
city_name = "Kyoto"
API_KEY = "d01ed74f13d285ba9f785fb49335bf3a"

api =
"http://api.openweathermap.org/data/2.5/weather?units=metric&q={city}&APPID={key}"

url = api.format(city = city_name, key = API_KEY)
response = requests.get(url)
data = response.json()
jsonText = json.dumps(data, indent=4)

print(jsonText)
```

独自の API キーは<https://openweathermap.org/api> で取得できます。

2. 次のコマンドでこのコードを実行し、OpenWeatherMap API からどのようなコンテンツが返されるかを確認します。

以下のURLから確認できます

<http://localhost:8000/cgi-bin/weather.py>

以下のコマンドをターミナルに入力しても確認できます。WindowsとMacで異なるコマンドになる可能性があります。Web検索等で自分のPCに適切なコマンドを調べてみましょう

```
$ python cgi-bin/weather.py
```

※`import requests` を利用するために「requests」をインストールする必要があります。次のコマンドでパッケージ化します。WindowsとMacで異なるコマンドになる可能性があります。Web検索等で自分のPCに適切なコマンドを調べてみましょう(pip install コマンド Windows)等で検索してみましょう)

```
$ pip install requests
```

JSONデータから都市名、天気アイコン、気温、湿度データを抽出

1. OpenWeatherMap API から返される JSON データの内容から、天気アイコン、気温、湿度データの抽出を次のコードで行えます。

```
city = data["name"]
icon = data["weather"][0]["icon"]
temp = data["main"]["temp"]
humidity = data["main"]["humidity"]
```

2. (ダミーの情報ではなく) 天気情報を返すように Python コードを変更します。天気アイコン、気温、湿度データをJson形式で返すようにしましょう。例えば以下の様にできます。なお、Pythonコード変更せずにhtmlコードを変更する方法もあります

```
data = response.json()

responseData = {
    "city": data["name"],
    "icon": data["weather"][0]["icon"],
    "temp": data["main"]["temp"],
    "humidity": data["main"]["humidity"]
}

jsonText = json.dumps(responseData, indent=4)
```

3. Python コードからどのようなコンテンツが返されるかを確認します。

都市名を指定する機能の追加

1. URL 指定時の JavaScript コードを以下のように変更すると、ユーザーが入力した都市名が送信されます。

```
var city_input = document.getElementById("city_input").value;  
var url = encodeURIComponent("/cgi-bin/weather.py?city=" + city_input);
```

2. ブラウザ上で入力した都市名は Python コードで以下のコードで取得できます。以下のコードを適切な位置に書きましょう。

```
import cgi  
  
form = cgi.FieldStorage()  
city_name = form.getvalue('city', 'Kyoto')
```

3. 指定した都市の気象情報が正しく表示されているかを確認します。

自由な作成

- 時間があれば自由な内容でウェブシステムを作成しましょう。例えば、以下があります。
 - 気象情報サイトに必要そうな機能を追加する
 - 気象情報サイト以外のウェブシステムを作成する
 - その他(上記以外の作成でも構いません)