



気象庁APIを使った 天気Webアプリ



①目的・背景



■ 目的

Pythonを活用してWebサイトのデータ取得する方法を理解

■ 背景

- ・当初はスクレイピングによるデータ取得を検討
- ・しかし、スクレイピングの制限やサイトの構造の変更に影響を受けやすい課題
- ・安定したデータを取得するために、気象庁 APIを利用する方法へ変更
- ・複数の都道府県の天気から地域を選択し、画面に天気情報を表示するアプリを作成



②アプリの概要



■ 目的


複数の都道府県から選択した地域の3日間の天気や週間天気など
気象庁のAPIでデータ取得して、Webアプリの画面に表示

■ 機能

- ・複数の都道府県から知りたい都道府県を選択
- ・選んだ都道府県の天気情報を取得
- ・日付ごとの天気や最高・最低気温を表示



③システム構成・使用技術



ブラウザ(ユーザー)



Flaskアプリ(サーバー処理)



気象庁API(天気データ取得)

■ブラウザ


- ・複数の都道府県を画面に表示
- ・ユーザーが選択した都道府県の天気情報を表示

■Flaskアプリ

- ・気象庁APIから天気情報を取得
- ・取得した情報から必要なデータを抽出し、画面へ渡す

■気象庁API

- ・天気情報を提供する外部サービス
- ・JSON形式でデータを返す



分類	技術	役割	選定理由
言語	Python	Webアプリの処理 APIデータ処理	コードの記述が簡潔で、 ライブラリが豊富
フレームワーク	Flask	Webアプリの構築 (ルーティング・画面表示)	シンプルで軽量で扱いやすい
外部API	気象庁 天気予報	天気データ取得	公開APIでデータを扱える
HTTP	requests	API通信	実装が簡単で定番
フロント	HTML/CSS	天気情報の画面表示	基本構造がシンプルでわかりやすい
仮想環境	uv	依存管理・実行	シンプルな操作で、環境構築を簡単にできる



④実装内容・工夫点



天気情報

東京

大阪

愛知

■実装内容

- ・地域を選択して天気情報を表示
- ・気象庁APIから天気情報を取得
- ・日付・天気・気温をWeb画面に表示

■工夫した点

- ・地域コードを切り替えて複数地域に対応
- ・天気コードを天気表示やアイコンに変換
- ・日付から曜日を計算して表示



⑤実装結果・動作例

地域名：東京

東京

大阪

愛知

2日間の天気情報

2026-02-09 (月)



天気：晴れ

最高：13°C

最低：-°C

2026-02-10 (火)



天気：晴れ時々曇り

最高：11°C

最低：4°C

週間の天気情報

2026-02-10 (火)



天気：晴れ時々曇り

最高：13°C

最低：0°C

2026-02-11 (水)



天気：曇り時々雨

最高：11°C

最低：4°C

2026-02-12 (木)



天気：曇り時々晴れ

最高：12°C

最低：4°C

2026-02-13 (金)



天気：晴れ時々曇り

最高：13°C

最低：2°C

- ・気象庁APIから取得した天気情報を表示
- ・必要な情報(日付・天気・気温)を一覧で表示
- ・天気情報が正しく表示されることを確認




⑥ 成果・動作結果



- ・気象庁APIからデータを取得し、Web画面へ表示するアプリを作成
- ・JSONのデータを分析し、必要な情報を抽出して、Web画面に表示
- ・Flaskを使ったAPI連携の基本的な流れを理解



⑦ 課題・改善点

- 
- ・表示情報が日付・天気・気温のみのため、降水量や風速などの情報も追加
 - ・API取得失敗時のエラー処理やメッセージ表示
 - ・ユーザーが見やすい画面にするため、UIの改善



⑧ まとめ・学んだこと



- ・Flaskを使ったWebアプリの簡単な作り方
- ・外部APIからデータを取得し、画面に表示するまで処理の流れ
- ・JSONデータを分析して必要な情報を取得する方法