



Assignment 2

Working with the Data

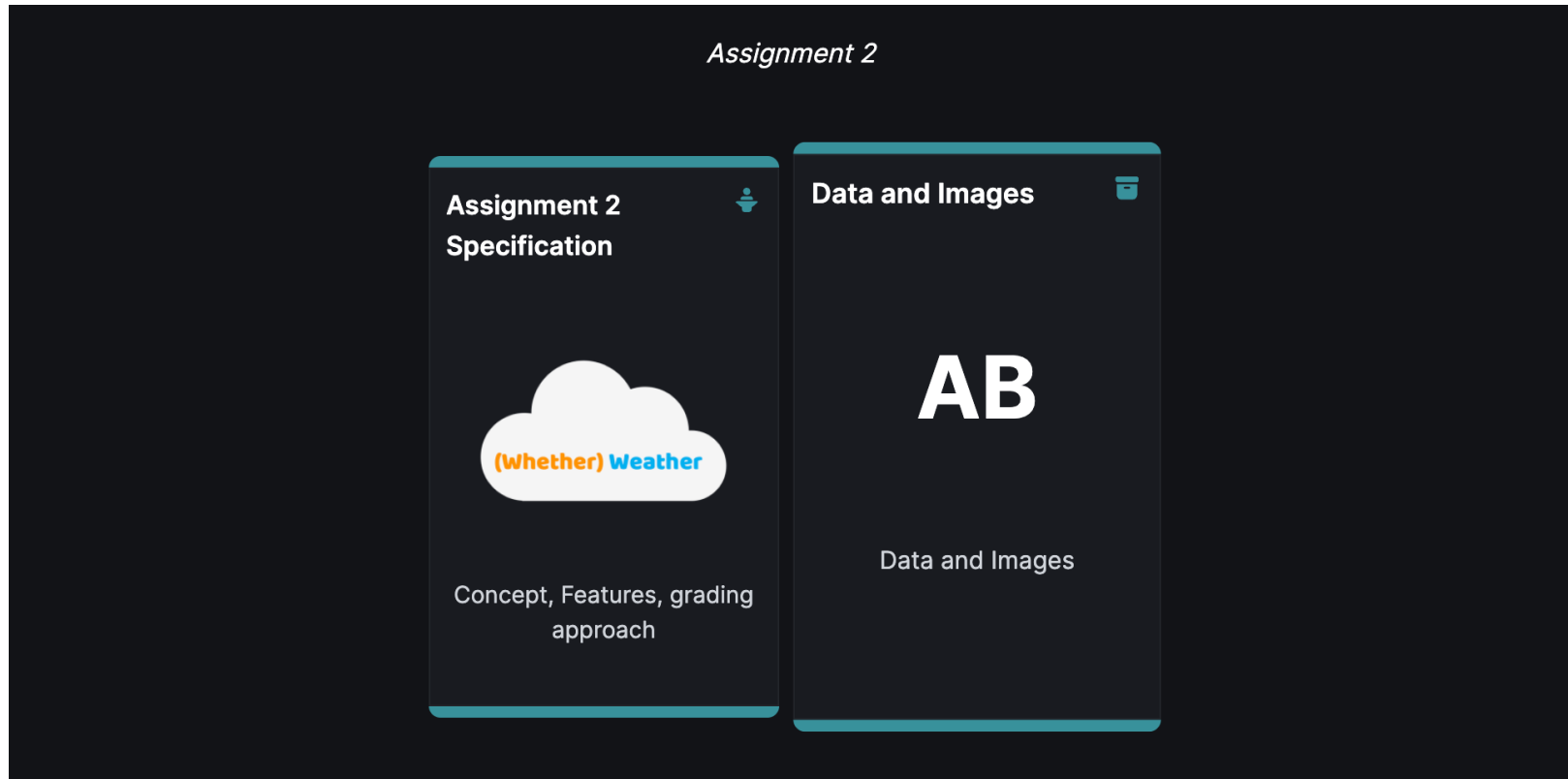
There is a video to accompany this!

- <https://www.youtube.com/watch?v=eU0APT5Xgfg>



(Whether) Weather

Where is the data?

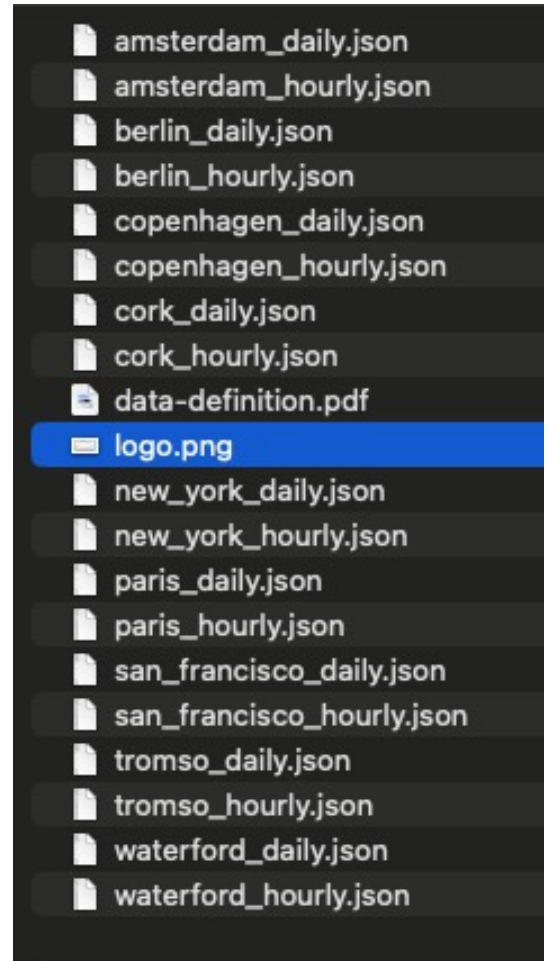


<https://wit-hdip-comp-sci-2024-web-dev-1.netlify.app/topic---assignments/unit-b-assign-2/archive/data.zip>

The Data

```
{
  "latitude": 52.5,
  "longitude": 13.259998,
  "generationtime_ms": 0.28693675994873047, "generationtime": Unknown v
  "utc_offset_seconds": 0,
  "timezone": "GMT",
  "timezone_abbreviation": "GMT",
  "elevation": 62.0,
  "daily_units": {
    "time": "iso8601",
    "weather_code": "wmo code",
    "temperature_2m_max": "°C",
    "temperature_2m_min": "°C",
    "apparent_temperature_max": "°C",
    "apparent_temperature_min": "°C",
    "sunrise": "iso8601",
    "sunset": "iso8601",
    "precipitation_hours": "h",
    "precipitation_probability_max": "%",
    "wind_speed_10m_max": "km/h",
    "wind_gusts_10m_max": "km/h",
    "wind_direction_10m_dominant": "°"
  },
  "daily": {
    "time": [
      "Today",
      "Today+1",
      "Today+2",
      "Today+3",
      "Today+4",
      "Today+5",
      "Today+6"
    ],
    "weather_code": [
      3,
      0,
      61,
      3,
      61,
      61,
      3
    ],
    "temperature_2m_max": [
      8.0,
      8.0,
      10.5,
      13.3,
      7.9,
      9.7,
      11.4
    ],
    "temperature_2m_min": [
      3.0,
      2.0,
      4.0,
      5.0,
      4.0,
      3.0,
      2.0
    ],
    "apparent_temperature_max": [
      8.0,
      8.0,
      10.5,
      13.3,
      7.9,
      9.7,
      11.4
    ],
    "apparent_temperature_min": [
      3.0,
      2.0,
      4.0,
      5.0,
      4.0,
      3.0,
      2.0
    ],
    "precipitation_hours": [
      0,
      0,
      0,
      0,
      0,
      0,
      0
    ],
    "precipitation_probability_max": [
      0,
      0,
      0,
      0,
      0,
      0,
      0
    ],
    "wind_speed_10m_max": [
      10,
      10,
      10,
      10,
      10,
      10,
      10
    ],
    "wind_gusts_10m_max": [
      15,
      15,
      15,
      15,
      15,
      15,
      15
    ],
    "wind_direction_10m_dominant": [
      0,
      0,
      0,
      0,
      0,
      0,
      0
    ]
  }
}
```

```
"daily": {
  "time": [
    "Today",
    "Today+1",
    "Today+2",
    "Today+3",
    "Today+4",
    "Today+5",
    "Today+6"
  ],
  "weather_code": [
    3,
    0,
    61,
    3,
    61,
    61,
    3
  ],
  "temperature_2m_max": [
    8.0,
    8.0,
    10.5,
    13.3,
    7.9,
    9.7,
    11.4
  ],
  "temperature_2m_min": [
    3.0,
    2.0,
    4.0,
    5.0,
    4.0,
    3.0,
    2.0
  ],
  "apparent_temperature_max": [
    8.0,
    8.0,
    10.5,
    13.3,
    7.9,
    9.7,
    11.4
  ],
  "apparent_temperature_min": [
    3.0,
    2.0,
    4.0,
    5.0,
    4.0,
    3.0,
    2.0
  ],
  "precipitation_hours": [
    0,
    0,
    0,
    0,
    0,
    0,
    0
  ],
  "precipitation_probability_max": [
    0,
    0,
    0,
    0,
    0,
    0,
    0
  ],
  "wind_speed_10m_max": [
    10,
    10,
    10,
    10,
    10,
    10,
    10
  ],
  "wind_gusts_10m_max": [
    15,
    15,
    15,
    15,
    15,
    15,
    15
  ],
  "wind_direction_10m_dominant": [
    0,
    0,
    0,
    0,
    0,
    0,
    0
  ]
}
```



What is the weather code for today?

```
You, last month • added data
{
  "latitude": 52.5,
  "longitude": 13.259998,
  "generationtime_ms": 0.28693675994873047, "generationtime": Unknown v
  "utc_offset_seconds": 0,
  "timezone": "GMT",
  "timezone_abbreviation": "GMT",
  "elevation": 62.0,
  "daily_units": {
    "time": "iso8601",
    "weather_code": "wmo code",
    "temperature_2m_max": "°C",
    "temperature_2m_min": "°C",
    "apparent_temperature_max": "°C",
    "apparent_temperature_min": "°C",
    "sunrise": "iso8601",
    "sunset": "iso8601",
    "precipitation_hours": "h",
    "precipitation_probability_max": "%",
    "wind_speed_10m_max": "km/h",
    "wind_gusts_10m_max": "km/h",
    "wind_direction_10m_dominant": "°"
  },
  "daily": {
    "time": [
      "Today",
      "Today+1",
      "Today+2",
      "Today+3",
      "Today+4",
      "Today+5",
      "Today+6"
    ],
    "weather_code": [
      3,
      0,
      61,
      3,
      61,
      61,
      3
    ],
    "temperature_2m_max": [
      8.0,
      8.0,
      10.5,
      13.3,
      7.9,
      9.7,
      11.4
    ]
  }
}
```

```
"daily": {
  "time": [
    "Today",
    "Today+1",
    "Today+2",
    "Today+3",
    "Today+4",
    "Today+5",
    "Today+6"
  ],
  "weather_code": [
    3,
    0,
    61,
    3,
    61,
    61,
    3
  ],
  "temperature_2m_max": [
    8.0,
    8.0,
    10.5,
    13.3,
    7.9,
    9.7,
    11.4
  ]
}
```

- The weather code for today is 3
- The weather code for tomorrow is 0
- The weather code for the day after tomorrow is 61

What is the max temperature for today?

```
You, last month • added data
{
  "latitude": 52.5,
  "longitude": 13.259998,
  "generationtime_ms": 0.28693675994873047, "generationtime": Unknown v
  "utc_offset_seconds": 0,
  "timezone": "GMT",
  "timezone_abbreviation": "GMT",
  "elevation": 62.0,
  "daily_units": {
    "time": "iso8601",
    "weather_code": "wmo code",
    "temperature_2m_max": "°C",
    "temperature_2m_min": "°C",
    "apparent_temperature_max": "°C",
    "apparent_temperature_min": "°C",
    "sunrise": "iso8601",
    "sunset": "iso8601",
    "precipitation_hours": "h",
    "precipitation_probability_max": "%",
    "wind_speed_10m_max": "km/h",
    "wind_gusts_10m_max": "km/h",
    "wind_direction_10m_dominant": "°"
  },
  "daily": {
    "time": [
      "Today",
      "Today+1",
      "Today+2",
      "Today+3",
      "Today+4",
      "Today+5",
      "Today+6"
    ],
    "weather_code": [
      3,
      0,
      61,
      3,
      61,
      61,
      3
    ],
    "temperature_2m_max": [
      8.0,
      8.0,
      10.5,
      13.3,
      7.9,
      9.7,
      11.4
    ],
    "temperature_2m_min": [
      -0.5,
      -0.5,
      -0.5,
      -0.5,
      -0.5,
      -0.5,
      -0.5
    ],
    "apparent_temperature_max": [
      8.0,
      8.0,
      10.5,
      13.3,
      7.9,
      9.7,
      11.4
    ],
    "apparent_temperature_min": [
      -0.5,
      -0.5,
      -0.5,
      -0.5,
      -0.5,
      -0.5,
      -0.5
    ],
    "precipitation_hours": [
      0,
      0,
      0,
      0,
      0,
      0,
      0
    ],
    "precipitation_probability_max": [
      0,
      0,
      0,
      0,
      0,
      0,
      0
    ],
    "wind_speed_10m_max": [
      3,
      0,
      61,
      3,
      61,
      61,
      3
    ],
    "wind_gusts_10m_max": [
      3,
      0,
      61,
      3,
      61,
      61,
      3
    ],
    "wind_direction_10m_dominant": [
      0,
      0,
      0,
      0,
      0,
      0,
      0
    ]
  }
}
```

```
"daily": {
  "time": [
    "Today",
    "Today+1",
    "Today+2",
    "Today+3",
    "Today+4",
    "Today+5",
    "Today+6"
  ],
  "weather_code": [
    3,
    0,
    61,
    3,
    61,
    61,
    3
  ],
  "temperature_2m_max": [
    8.0,
    8.0,
    10.5,
    13.3,
    7.9,
    9.7,
    11.4
  ],
  "temperature_2m_min": [
    -0.5,
    -0.5,
    -0.5,
    -0.5,
    -0.5,
    -0.5,
    -0.5
  ],
  "apparent_temperature_max": [
    8.0,
    8.0,
    10.5,
    13.3,
    7.9,
    9.7,
    11.4
  ],
  "apparent_temperature_min": [
    -0.5,
    -0.5,
    -0.5,
    -0.5,
    -0.5,
    -0.5,
    -0.5
  ],
  "precipitation_hours": [
    0,
    0,
    0,
    0,
    0,
    0,
    0
  ],
  "precipitation_probability_max": [
    0,
    0,
    0,
    0,
    0,
    0,
    0
  ],
  "wind_speed_10m_max": [
    3,
    0,
    61,
    3,
    61,
    61,
    3
  ],
  "wind_gusts_10m_max": [
    3,
    0,
    61,
    3,
    61,
    61,
    3
  ],
  "wind_direction_10m_dominant": [
    0,
    0,
    0,
    0,
    0,
    0,
    0
  ]
}
```

- The max temp for today is 8.0
- The max temp for tomorrow is 8.0
- The max temp for the day after tomorrow is 10.5

Time index and properties

- The index of the “date” is the same index of the data
- For Today+5
 - The wind speed is 11.2
 - The wind gusts are 25.2
 - The dominant wind direction is 247 degrees (you should convert to compass)

```
"time": [  
  "Today",  
  "Today+1",  
  "Today+2",  
  "Today+3",  
  "Today+4",  
  "Today+5",  
  "Today+6"  
],
```

```
],  
"wind_speed_10m_max": [  
  15.0,  
  14.6,  
  17.2,  
  13.5,  
  10.9,  
  11.2,  
  10.2  
],  
"wind_gusts_10m_max": [  
  33.1,  
  31.7,  
  41.0,  
  32.4,  
  24.8,  
  25.2,  
  25.9  
],  
"wind_direction_10m_dominant": [  
  76,  
  94,  
  95,  
  88,  
  19,  
  247,  
  219  
]  
]
```

What else is in there? – Data Definition

data-definition.pdf × test.html U JS copenhagen.daily.js U JS createObject.js U {} complete_data.json

topic---assignments > unit-b-assign-2 > data > data-definition.pdf

1 of 14 Automatic Zoom

05/04/2024, 16:05 Docs | Open-Meteo.com

API Documentation

The API endpoint `/v1/forecast` accepts a geographical coordinate, a list of weather variables and responds with a JSON hourly weather forecast for 7 days. Time always starts at 0:00 today and contains 168 hours. If `&forecast_days=16` is set, up to 16 days of forecast can be returned. All URL parameters are listed below:

| Parameter | Format | Required | Default | Description |
|----------------------------|----------------|----------|---------|--|
| latitude, longitude | Floating point | Yes | | Geographical WGS84 coordinates location. Multiple coordinates can be separated. E.g. <code>&latitude=52.52,48.85&longitude=10.00,10.00</code> . To return data for multiple locations, the output changes to a list of structures. For JSON and XLSX formats add a column <code>location</code> to the list. |
| elevation | Floating point | No | | The elevation used for statistical downscaling. Per default, a 90 meter digital elevation model is used. You can manually set the elevation to correctly match mountain peaks. If <code>&elevation=nan</code> is specified, downscaling will be disabled and the API uses the average grid-cell height. For multiple locations, the elevation can also be comma separated. |

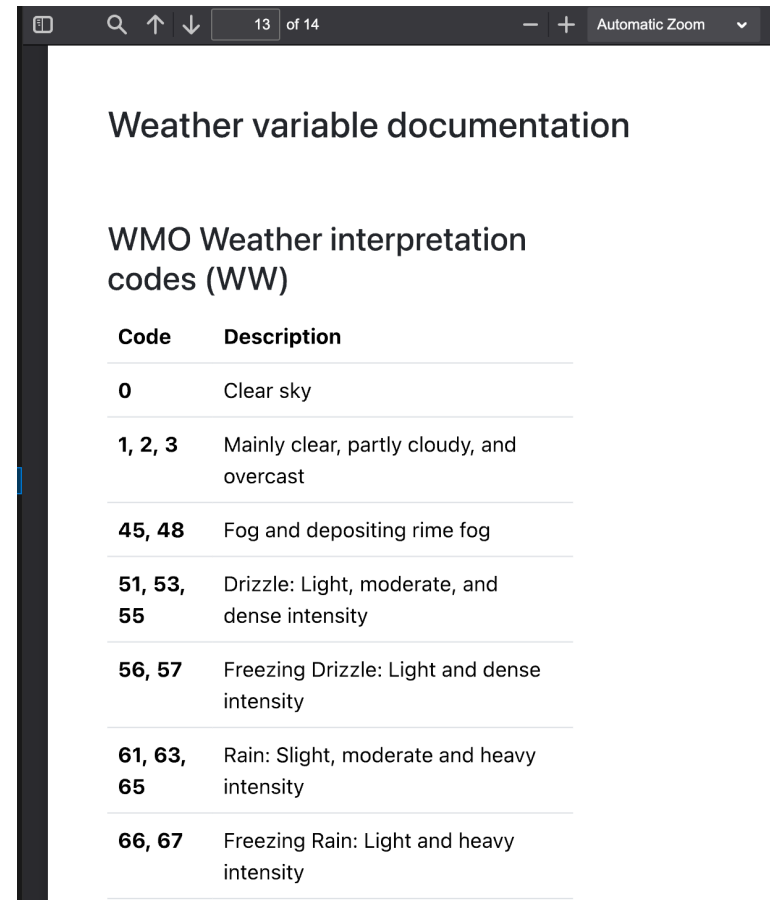
Hourly Parameter Definition

The parameter `&hourly=` accepts the following values. Most weather variables are given as an instantaneous value for the indicated hour. Some variables like precipitation are calculated from the preceding hour as an average or sum.

| Variable | Valid time | Unit | Description |
|--|------------|---------|---|
| temperature_2m | Instant | °C (°F) | Air temperature at 2 meters above ground |
| relative_humidity_2m | Instant | % | Relative humidity at 2 meters above ground |
| dew_point_2m | Instant | °C (°F) | Dew point temperature at 2 meters above ground |
| apparent_temperature | Instant | °C (°F) | Apparent temperature is the perceived feels-like temperature combining wind chill factor, relative humidity and solar radiation |
| pressure_msl surface_pressure | Instant | hPa | Atmospheric air pressure reduced to mean sea level (msl) or pressure at surface. Typically |

What is a weather code?

- Check the data definition documentation

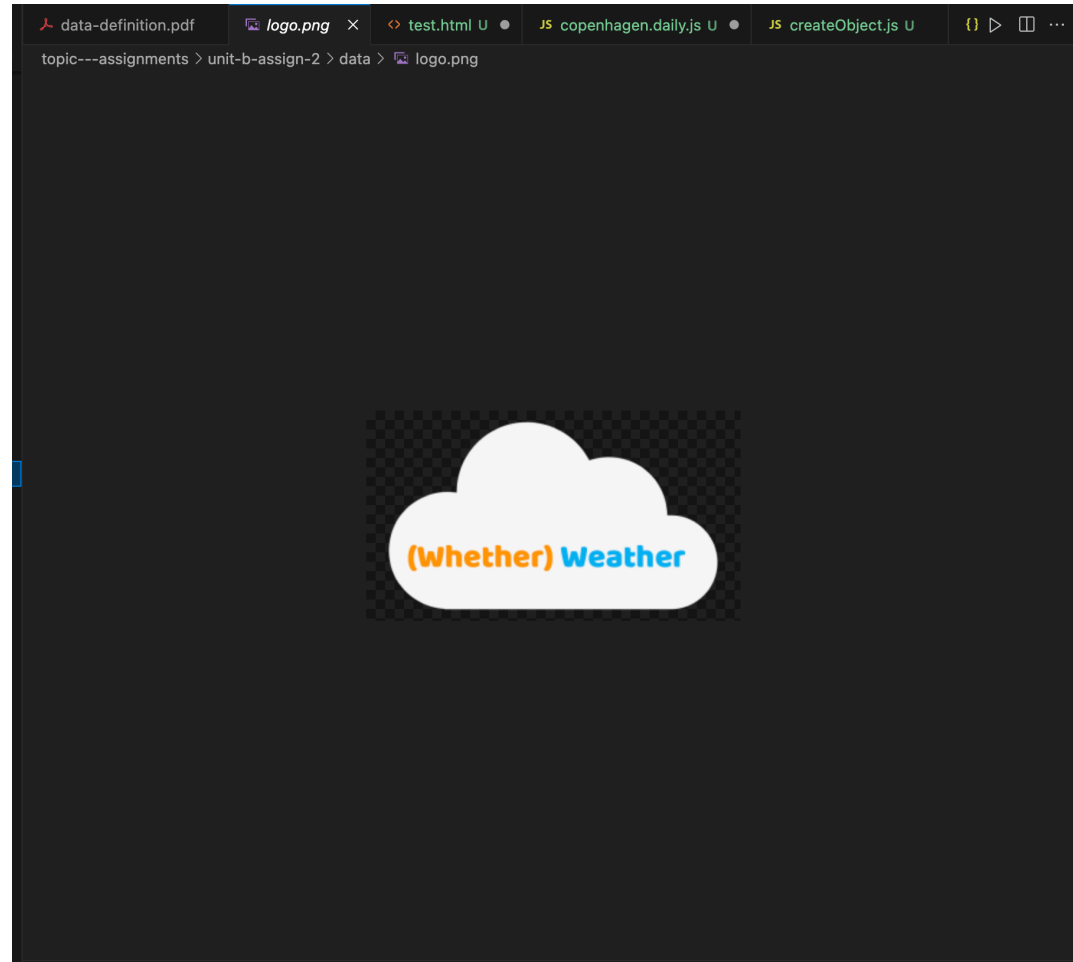


Weather variable documentation

WMO Weather interpretation codes (WW)

| Code | Description |
|------------|---|
| 0 | Clear sky |
| 1, 2, 3 | Mainly clear, partly cloudy, and overcast |
| 45, 48 | Fog and depositing rime fog |
| 51, 53, 55 | Drizzle: Light, moderate, and dense intensity |
| 56, 57 | Freezing Drizzle: Light and dense intensity |
| 61, 63, 65 | Rain: Slight, moderate and heavy intensity |
| 66, 67 | Freezing Rain: Light and heavy intensity |

What else is in there? – Logo



How do we use the data?

- The data is in JSON format
 - https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/JSON
- This stands for JavaScript Object Notation
- JSON is a standard across APIs for delivering data over the network
- You can think of it as JavaScript arrays and Objects

How can we use JSON in our Assignment?

- I have provided a complete_data.json file, this is for reference
- This is duplicated by weather_data.js and is a JavaScript file that we can use to import into our HTML pages
- This creates an object called weatherData that has
 - The keys of the object are file name e.g. amsterdam_hourly, amsterdam_daily
 - The value is the weather data

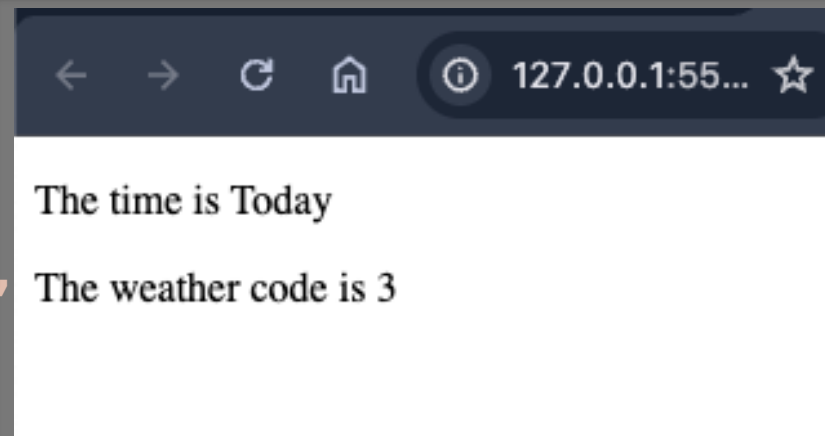
```
const weatherData = {
  "amsterdam_daily": {
    "latitude": 52.36,
    "longitude": 4.82,
    "generationtime_ms": 0.18203258514404297,
    "utc_offset_seconds": 0,
    "timezone": "GMT",
    "timezone_abbreviation": "GMT",
    "elevation": 1,
    "daily_units": {
      "time": "iso8601",
      "weather_code": "wmo code",
      "temperature_2m_max": "°C",
      "temperature_2m_min": "°C",
      "apparent_temperature_max": "°C",
      "apparent_temperature_min": "°C",
      "sunrise": "iso8601",
      "sunset": "iso8601",
      "precipitation_hours": "h",
      "precipitation_probability_max": "%",
      "wind_speed_10m_max": "km/h",
      "wind_gusts_10m_max": "km/h",
      "wind_direction_10m_dominant": "°"
    },
    "daily": {
    }
  },
  "amsterdam_hourly": {
    "latitude": 52.36,
    "longitude": 4.82,
    "generationtime_ms": 0.24902820587158203,
    "utc_offset_seconds": 0,
    "timezone": "GMT",
    "timezone_abbreviation": "GMT",
    "elevation": 1,
    "hourly_units": {
    },
    "hourly": {
    }
  },
  "berlin_daily": {
```

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
<title>Test Data</title>
<script src="weather_data.js"></script>
<script>
  document.addEventListener("DOMContentLoaded", () => {
    const currentCity = "amsterdam";
    const currentCityData = weatherData[currentCity + "_daily"]
    const time = document.getElementById("time");
    const weatherCode = document.getElementById("weatherCode");

    time.innerHTML = currentCityData.daily.time[0];
    weatherCode.innerHTML = currentCityData.daily.weather_code[0];
  });
</script>
</head>
<body>
  <p>The time is <span id="time"></span></p>
  <p>The weather code is <span id="weatherCode"></span></p>
</body>
</html>

```



This page has been added to the data.zip

Basic

- This would be seen as a basic way of loading data
- To get to the next level we would want to namespace the weatherData object
- We would also like to hide it behind a datastore via a pattern similar to Model View Controller
- This will be shown in the lectures later, however, for a passing mark, accessing the data as above is acceptable

Useful Techniques for Working with Data

You may not have covered all these yet but we will

The Array map function

- `Array.map` applies a function to each item in an array
- It then maps the **return** value of the function to the same index of in completely new array

```
const words = ['metal', 'rock', 'folk']
```

```
const wordToDescription = (word) => {  
  return `The word "${word}" has ${word.length} letters`  
}
```

```
const descriptions = words.map(wordToDescription)
```

```
[  
  'The word "metal" has 5 letters',  
  'The word "rock" has 4 letters',  
  'The word "folk" has 4 letters'  
]
```


The Array map function

```
const words = ['metal', 'rock', 'folk']

const wordToDescription = (word) => {
  return `The word "${word}" has ${word.length} letters`
}

const descriptions = words.map(wordToDescription)
```

- The function is typically declared within the parentheses

```
const descriptions = words.map((word) => {
  return `The word "${word}" has ${word.length} letters`
})
```

```
[
  'The word "metal" has 5 letters',
  'The word "rock" has 4 letters',
  'The word "folk" has 4 letters'
]
```

The Array map function

```
const words = ['metal', 'rock', 'folk']

const wordToDescription = (word) => {
  return `The word "${word}" has ${word.length} letters`
}

const descriptions = words.map(wordToDescription)
```

- This can be shortened further from:

```
const descriptions = words.map((word) => {
  return `The word "${word}" has ${word.length} letters`
})
```

```
[
  'The word "metal" has 5 letters',
  'The word "rock" has 4 letters',
  'The word "folk" has 4 letters'
]
```

- To a one-line assignment:

```
const descriptions = words.map(word => `The word "${word}" has ${word.length} letters`)
```

The Array forEach function

- `Array.forEach` runs a function for each item in an array
- Unlike “map” it has no return value

```
const words = ['metal', 'rock', 'folk']  
  
const descriptions = [] // we use forEach to populate one by one  
  
words.forEach(word => descriptions.push(`The word "${word}" has ${word.length} letters`))  
  
console.log(descriptions)
```

```
[  
  'The word "metal" has 5 letters',  
  'The word "rock" has 4 letters',  
  'The word "folk" has 4 letters'  
]
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/forEach

To map or forEach

- Use map when you want to **transform** each item in an array into a new array, e.g. square every number in an array.
- Use forEach when you want to use each item in an array to act upon another piece of data e.g. add every item of fruit into a shopping basket

```
const words = ['metal', 'rock', 'folk']
```

```
const uppercaseWords = words.map(word => word.toUpperCase());  
// new array with ["METAL", "ROCK", "FOLK"]
```

find and filter

- Use `Array.find` to find the **FIRST ELEMENT** that matches a condition
- Use `Array.filter` to find **ALL THE ELEMENTS** as an array that match a condition

```
const numbers = [5, 12, 8, 130, 44];
```

```
const foundNumber = numbers.find((element) => element > 10); // 12
```

```
const foundNumbers = numbers.filter((element) => element > 10); // [12, 130, 44]
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/find
https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter