

<image/nation>

Tech Training Series

Exclusive for



# JavaScript Programming

(2023/2024)

## Intermediate Level

*by*

Sunny NG

<image/nation>

# In this workshop (3 hours)

- JavaScript Syntax
  - Variables
  - Operators
  - Arithmetic
  - Data Types
  - Functions
  - Objects
  - Array
- fetch API
- JSON Processing
- JS Data Visualization
  - Apache eCharts
  - Google Charts
  - D3.js
  - Observable HQ

# Required Software

Make sure you have the following software installed

1. Google Chrome Browser
2. Visual Studio Code
3. Node js

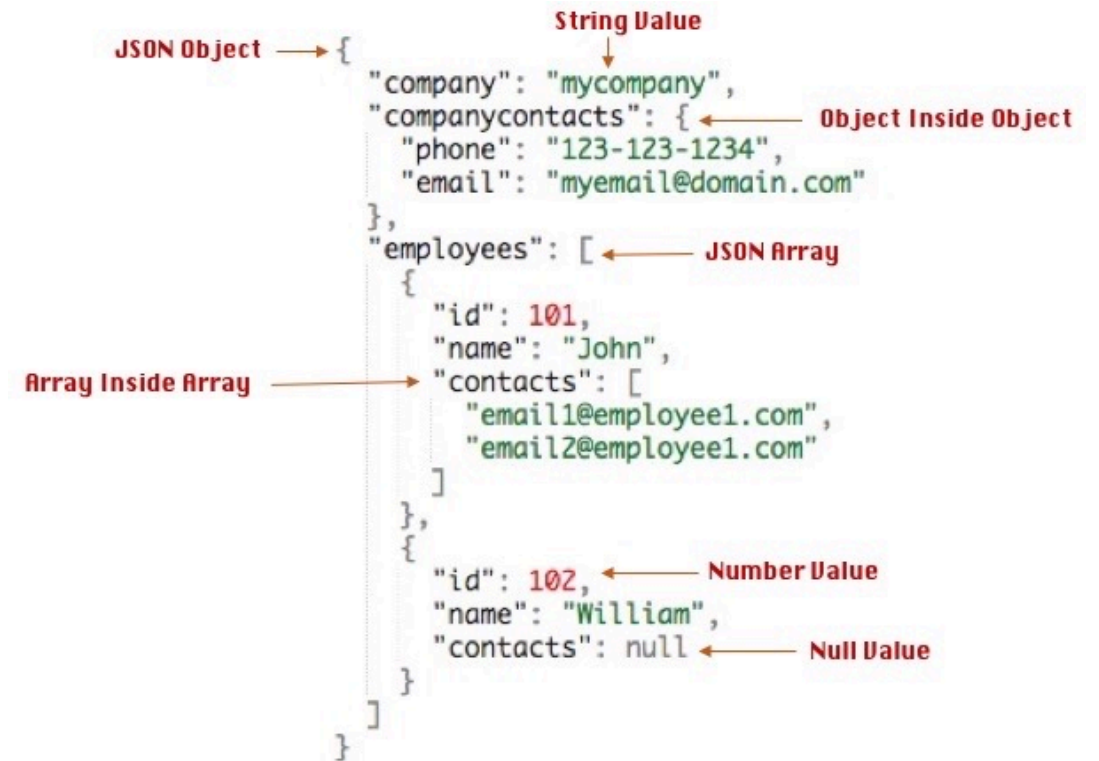
# Sunny Ng



- Founder / Master Trainer Image Nation
- Developer Web, Mobile, WeChat & IoT
- Content Creator Video producing / Live streaming
- AWS Solution Architect – Associate
- Alibaba Cloud Professional
- AWS Academy Educator
- Email: [sunny.ng@imagenation.com.hk](mailto:sunny.ng@imagenation.com.hk)
- 🐙 [github.com/ngsanluk](https://github.com/ngsanluk)

# JSON

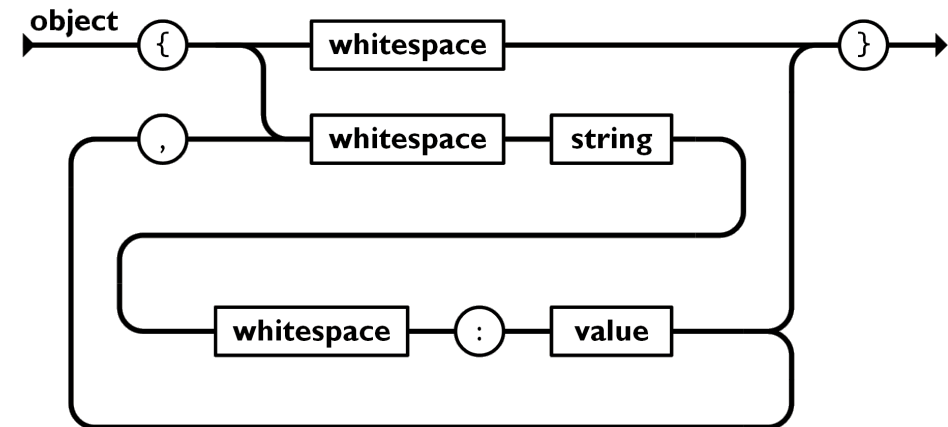
- JSON (**JavaScript Object Notation**) is an open standard file format and data interchange format that are user friendly
- JSON is a language-independent data format.
- It was derived from JavaScript, but many modern programming languages include code to generate and parse JSON-format data.
- JSON filenames use the extension **.json**



# JSON Syntax

<https://www.json.org/json-en.html>

- **{ }**
  - Object
- **[ ]**
  - Array of objects
- JSON is nested, one can have
  - An object that contains array
  - An array that contains objects



# JavaScript Object $\neq$ JSON

## JavaScript object

```
{firstname : "Sam",  
  lastname : "Fernandes"}
```

↑  
Key

↑  
Value

(Need not be enclosed  
within double quotes)

## JSON object

```
{"firstname" : "Sam",  
  "lastname" : "Fernandes"}
```

↑  
Key

↑  
Value

(Must be enclosed  
within double quotes)

# Dummy JSON Data

https://jsonplaceholder.typicode.com/

## Resources

JSONPlaceholder comes with a set of 6 common resources:

<a href="#">/posts</a>	100 posts
<a href="#">/comments</a>	500 comments
<a href="#">/albums</a>	100 albums
<a href="#">/photos</a>	5000 photos
<a href="#">/todos</a>	200 todos
<a href="#">/users</a>	10 users

```
{
  "id": 1,
  "name": "Leanne Graham",
  "username": "Bret",
  "email": "Sincere@april.biz",
  "address": {
    "street": "Kulas Light",
    "suite": "Apt. 556",
    "city": "Gwenborough",
    "zipcode": "92998-3874",
    "geo": {
      "lat": "-37.3159",
      "lng": "81.1496"
    }
  },
  "phone": "1-770-736-8031 x56442",
  "website": "hildegard.org",
  "company": {
    "name": "Romaguera-Crona",
    "catchPhrase": "Multi-layered client-server neural-net",
    "bs": "harness real-time e-markets"
  }
}
```



# Real-world JSON Data Source

Weather API by Hong Kong Observatory

- HK Weather Data Dictionary

- <https://data.weather.gov.hk/weatherAPI/opendata/weather.php>

- HK Weather Resource Endpoint / Link

- <https://data.weather.gov.hk/weatherAPI/opendata/weather.php?dataType=flw&lang=tc>

# HK Observatory Open API

## Data Dictionary

- [https://www.hko.gov.hk/en/weatherAPI/doc/files/HKO\\_Open\\_Data\\_API\\_Documentation.pdf](https://www.hko.gov.hk/en/weatherAPI/doc/files/HKO_Open_Data_API_Documentation.pdf)

### Request Example

<https://data.weather.gov.hk/weatherAPI/opendata/weather.php?dataType=flw&lang=en>

### Request

Parameter	Accepted values	Description
dataType	flw fnd rhrread warnsum warningInfo swt	flw: Local Weather Forecast fnd: 9-day Weather Forecast rhrread: Current Weather Report warnsum: Weather Warning Summary warningInfo: Weather Warning Information swt: Special Weather Tips
lang	en tc sc	en: English tc: Traditional Chinese sc: Simplified Chinese Default language: en

# 9-day Weather Forecast (Raw Data)

<https://data.weather.gov.hk/weatherAPI/opendata/weather.php?dataType=fnd&lang=en>

JSON	<b>Raw Data</b>	Headers
Save	Copy	Pretty Print

```

{"generalSituation":"A fresh to strong easterly airstream will affect the coast of southern China in the next couple of days. A cold front is expected to reach the coast of Guangdong on Saturday. Temperatures will fall appreciably over the region. Under the influence of an intense winter monsoon and upper-air disturbance, it will remain cold with some rain over southern China during the weekend to midweek next week.", "weatherForecast": [{"forecastDate": "20220216", "week": "Wednesday", "forecastWind": "East force 4 to 5, occasionally force 6 offshore later.", "forecastWeather": "Mainly cloudy. Sunny periods during the day.", "forecastMaxtemp": {"value": 19, "unit": "C"}, "forecastMintemp": {"value": 16, "unit": "C"}, "forecastMaxrh": {"value": 90, "unit": "percent"}, "forecastMinrh": {"value": 70, "unit": "percent"}, "ForecastIcon": 51, "PSR": "Low"}, {"forecastDate": "20220217", "week": "Thursday", "forecastWind": "East force 5, force 6 offshore and on high ground.", "forecastWeather": "Cloudy with one or two rain patches.", "forecastMaxtemp": {"value": 19, "unit": "C"}, "forecastMintemp": {"value": 17, "unit": "C"}, "forecastMaxrh": {"value": 95, "unit": "percent"}, "forecastMinrh": {"value": 80, "unit": "percent"}, "ForecastIcon": 60, "PSR": "Low"}, {"forecastDate": "20220218", "week": "Friday", "forecastWind": "East force 4 to 5, force 6 offshore and on high ground at first.", "forecastWeather": "Cloudy with a few rain and mist patches.", "forecastMaxtemp": {"value": 19, "unit": "C"}, "forecastMintemp": {"value": 17, "unit": "C"}, "forecastMaxrh": {"value": 95, "unit": "percent"}, "forecastMinrh": {"value": 85, "unit": "percent"}, "ForecastIcon": 62, "PSR": "Medium Low"}, {"forecastDate": "20220219", "week": "Saturday", "forecastWind": "North to northeast force 4 to 5.", "forecastWeather": "Becoming cold appreciably. Cloudy with occasional rain.", "forecastMaxtemp": {"value": 17, "unit": "C"}, "forecastMintemp": {"value": 12, "unit": "C"}, "forecastMaxrh": {"value": 95, "unit": "percent"}, "forecastMinrh": {"value": 80, "unit": "percent"}, "ForecastIcon": 92, "PSR": "High"}, {"forecastDate": "20220220", "week": "Sunday", "forecastWind": "North to northeast force 4 to 5, occasionally force 6 offshore.", "forecastWeather": "Cloudy to overcast with occasional rain. It will be cold.", "forecastMaxtemp": {"value": 13, "unit": "C"}, "forecastMintemp": {"value": 11, "unit": "C"}, "forecastMaxrh": {"value": 95, "unit": "percent"}, "forecastMinrh": {"value": 80, "unit": "percent"}, "ForecastIcon": 93, "PSR": "High"},

```

# Use **Firefox** for better JSON viewing

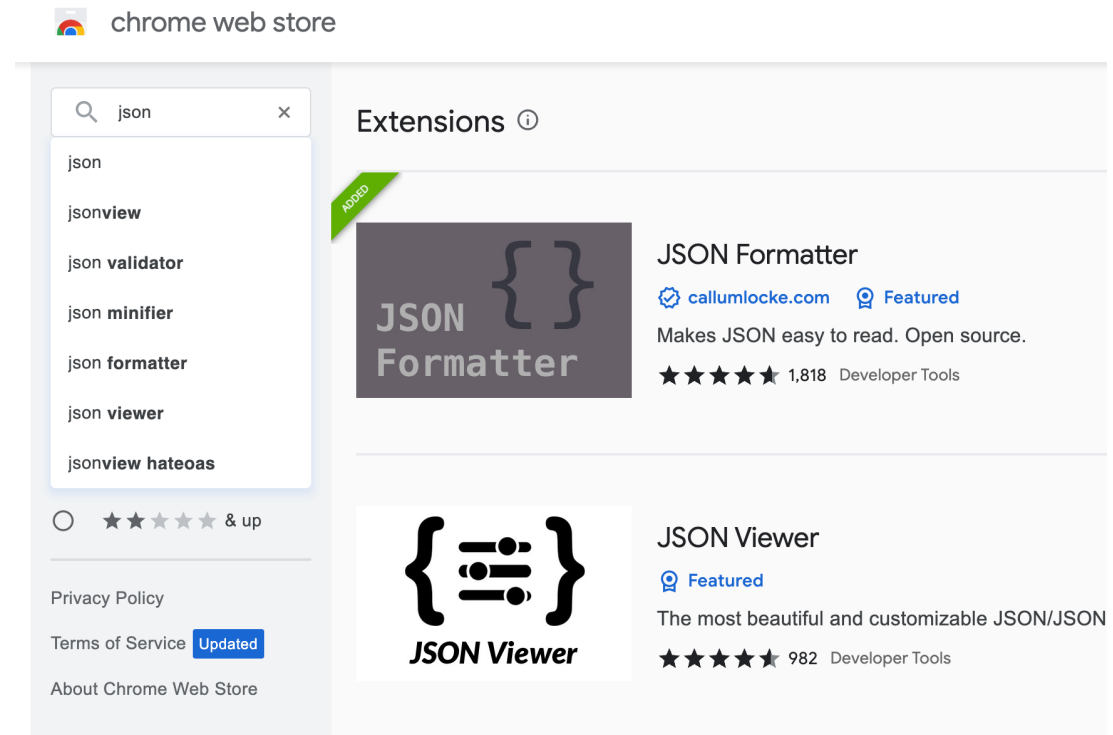
The screenshot shows the Firefox JSON viewer interface. At the top, there are tabs for 'JSON', 'Raw Data', and 'Headers'. Below the tabs is a toolbar with buttons for 'Save', 'Copy', 'Collapse All', 'Expand All', and a search icon labeled 'Filter JSON'. The main content area displays a JSON object with two main properties: 'generalSituation' and 'weatherForecast'. The 'generalSituation' property is expanded, showing a long text description. The 'weatherForecast' property is also expanded, revealing an array with one element (index 0). This element contains several sub-properties: 'forecastDate', 'week', 'forecastWind', 'forecastWeather', 'forecastMaxtemp', and 'forecastMintemp'. The 'forecastMaxtemp' and 'forecastMintemp' properties are further expanded to show their 'value' and 'unit' sub-properties.

```
JSON Raw Data Headers
Save Copy Collapse All Expand All Filter JSON
▼ generalSituation: "A fresh to strong easterly airstream will affect the coast of southern China in the next couple of days. A cold front is expected to reach the coast of Guangdong on Saturday. Temperatures will fall appreciably over the region. Under the influence of an intense winter monsoon and upper-air disturbance, it will remain cold with some rain over southern China during the weekend to midweek next week."
▼ weatherForecast:
  ▼ 0:
    forecastDate: "20220216"
    week: "Wednesday"
    ▼ forecastWind: "East force 4 to 5, occasionally force 6 offshore later."
      forecastWeather: "Mainly cloudy. Sunny periods during the day."
    ▼ forecastMaxtemp:
      value: 19
      unit: "C"
    ▼ forecastMintemp:
      value: 16
      unit: "C"
```

# Or install Google Extensions

Go to **Chrome Web Store** and search “**JSON**”

<https://chrome.google.com/webstore/search/json?hl=en>



# JavaScript fetch API

# fetch API

- The fetch API provides a JavaScript interface for accessing and manipulating parts of the HTTP pipeline, such as **requests** and **responses**.
- It also provides a global `fetch()` method that provides an easy, logical way to fetch resources (usually in JSON format) asynchronously across the network.
- [https://developer.Mozilla.Org/en-us/docs/web/API/fetch\\_api/using\\_fetch](https://developer.mozilla.org/en-us/docs/web/API/fetch_api/using_fetch)

# fetch Sample Project

<https://jsfiddle.net/sunnyng/bf0k2ezm/27/>

The screenshot shows a JSFiddle environment with the following content:

- HTML:**

```
1 <h1>Weather Forecast</h1>
2 <hr>
3 <div id="output"></div>
```
- CSS:**

```
1
```
- JavaScript + No-Library (pure JS):**

```
1 fetch('https://data.weather.gov.hk/weatherAPI/opendata/weather.php?dataType=fnd&lang=tc')
2   .then(response => response.json())
3   .then(data => {
4     console.log(data)
5     document.querySelector('#output').innerHTML = data.generalSituation;
6   })
7 );
8
```

**Weather Forecast**

一股清勁至強風程度的偏東氣流會在未來兩三日影響華南沿岸。預料一道冷鋒會在星期六抵達廣東沿岸，該區氣溫顯著下降。受高空擾動及季候風影響，週末至下週中期華南地區持續寒冷及有雨。



# fetch Sample Codes

```
fetch('https://data.weather.gov.hk/weatherAPI/opendata/weather.php?dataType=fnd&lang=tc')  
  .then(response => response.json())  
  .then(data => {  
    console.log(data.generalSituation)  
  })  
);
```

# fetch usage

- We are fetching a JSON file across the internet and printing it to the console (the windows for debug/dev)
- The simplest use of `fetch()` takes one argument — the path to the web resource (URL) you want to fetch
- The Response object, in turn, does not directly contain the actual JSON response body but is instead a representation of the entire HTTP **response**.
- So, to extract the JSON body content from the Response object, we use the `json()` method which returns the response body text as JS object

# Processing JSON data

- JSON is nested data structure
- Use `.` (dot syntax) to refer to data attribute and sub-attributes
  - `data.generalSituation`
  - `data.seaTemp.place`
- Use index number in square bracket `[]` to access member element in array
  - `data.weatherForecast[0]`
  - `data.weatherForecast[0].forecastDate`

# querySelector ()

## Manipulating HTML Contents

# Document.querySelector()

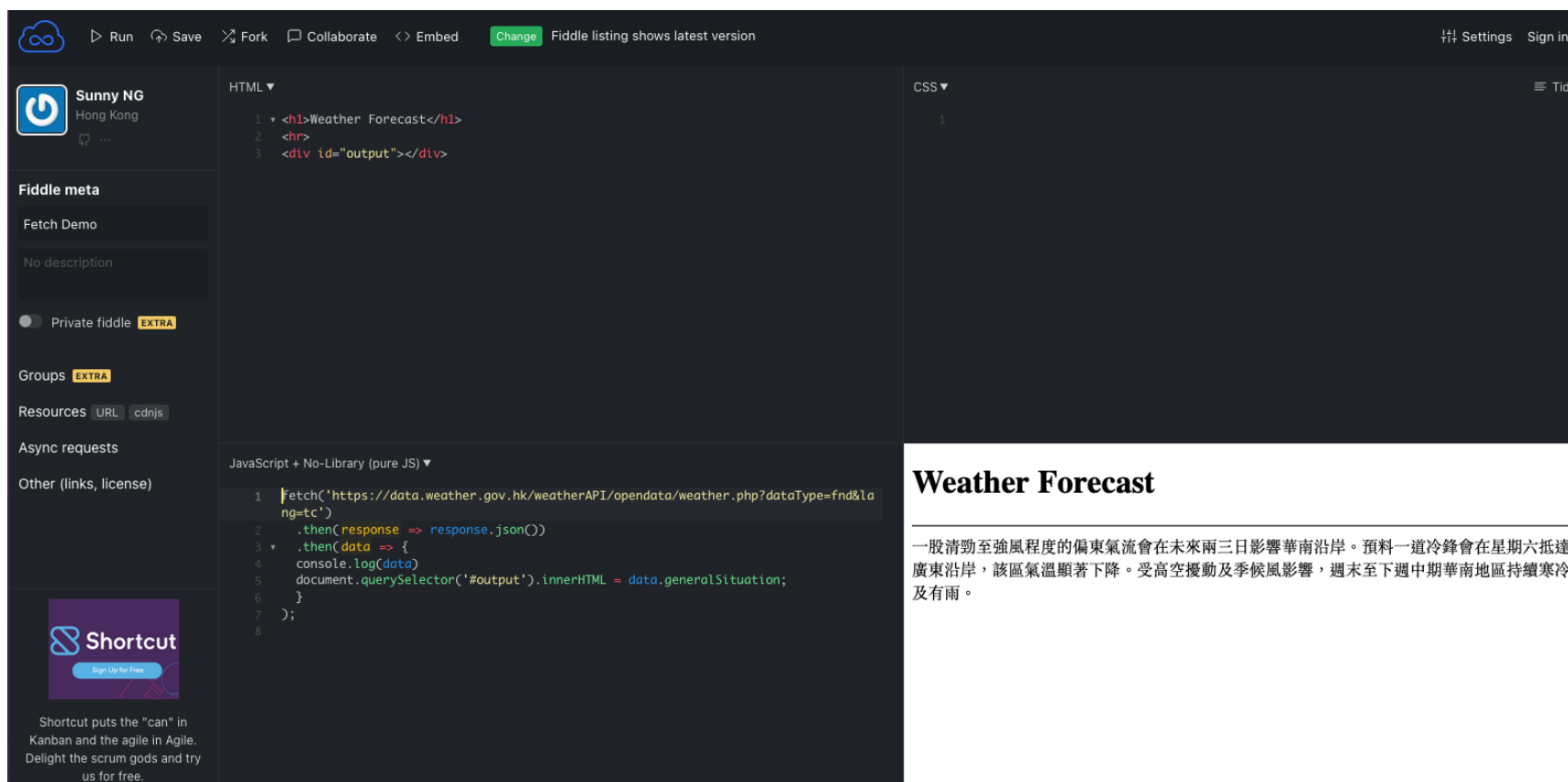
- `querySelector()` returns the **first** Element within the HTML document that matches the specified selector
  - If no matches are found, null is returned
- **Syntax** (case sensitive)
  - `element = document.querySelector(selectors);`
- **Example**
  - `output = document.querySelector('#output')`
  - `product = document.querySelector('.product')`

# Example

```
fetch('https://data.weather.gov.hk/weatherAPI/opendata/weather.php?dataType=fnd&lang=tc')
  .then(response => response.json())
  .then(data => {
    document.querySelector('#output').innerHTML
      = data.generalSituation;
  })
);
```

# Complete Codes

<https://jsfiddle.net/sunnyng/bf0k2ezm/27/>



The screenshot shows a JSFiddle editor interface. On the left, there is a sidebar with the user profile 'Sunny NG Hong Kong', 'Fiddle meta' section (Fetch Demo, No description, Private fiddle toggle, Groups, Resources, Async requests, Other links), and a 'Shortcut' advertisement. The main editor area is divided into three panes: HTML, CSS, and JavaScript. The HTML pane contains three lines of code: an h1 tag with the text 'Weather Forecast', an hr tag, and a div with id 'output'. The CSS pane is empty. The JavaScript pane contains a single line of code using the fetch API to retrieve weather data from the OpenWeatherMap API and update the content of the 'output' div.

```
HTML
1 <h1>Weather Forecast</h1>
2 <hr>
3 <div id="output"></div>

CSS
1

JavaScript + No-Library (pure JS)
1 fetch('https://data.weather.gov.hk/weatherAPI/opendata/weather.php?dataType=fnd&lang=tc')
2   .then(response => response.json())
3   .then(data => {
4     console.log(data)
5     document.querySelector('#output').innerHTML = data.generalSituation;
6   })
7 );
8
```

### Weather Forecast

一股清勁至強風程度的偏東氣流會在未來兩三日影響華南沿岸。預料一道冷鋒會在星期六抵達廣東沿岸，該區氣溫顯著下降。受高空擾動及季候風影響，週末至下週中期華南地區持續寒冷及有雨。

# 9 Day Weather Forecast

Complete Codes - <https://jsfiddle.net/sunnyng/bf0k2ezm/60/>

```
HTML ▼
1 <h1>HK Weather Forecast</h1>
2 <hr>
3 <ul id="output"></ul>
4 <hr>
5 <i>Provided by Hong Kong Observatory</i>

CSS ▼
1

JavaScript + No-Library (pure JS) ▼
1 fetch('https://data.weather.gov.hk/weatherAPI/opendata/weather.php?dataType=fnd&lang=tc')
2 .then(response => response.json()) // getting response
3 .then(data => { // processing JSON data
4   var output = document.querySelector('#output')
5   weatherForecast = data.weatherForecast; // an array
6   weatherForecast.forEach((item) => {
7     //console.log(item.week);
8     const forecast = document.createElement("li");
9     forecast.innerHTML = item.forecastDate
10      + " "
11      + item.week
12      + " : "
13      + item.forecastMaxtemp.value
14      + "°C";
15     output.appendChild(forecast);
16   });
17 }
18 );
19
```

## HK Weather Forecast

- 20220905 星期一 : 34°C
- 20220906 星期二 : 33°C
- 20220907 星期三 : 29°C
- 20220908 星期四 : 29°C
- 20220909 星期五 : 31°C
- 20220910 星期六 : 32°C
- 20220911 星期日 : 31°C
- 20220912 星期一 : 31°C
- 20220913 星期二 : 31°C

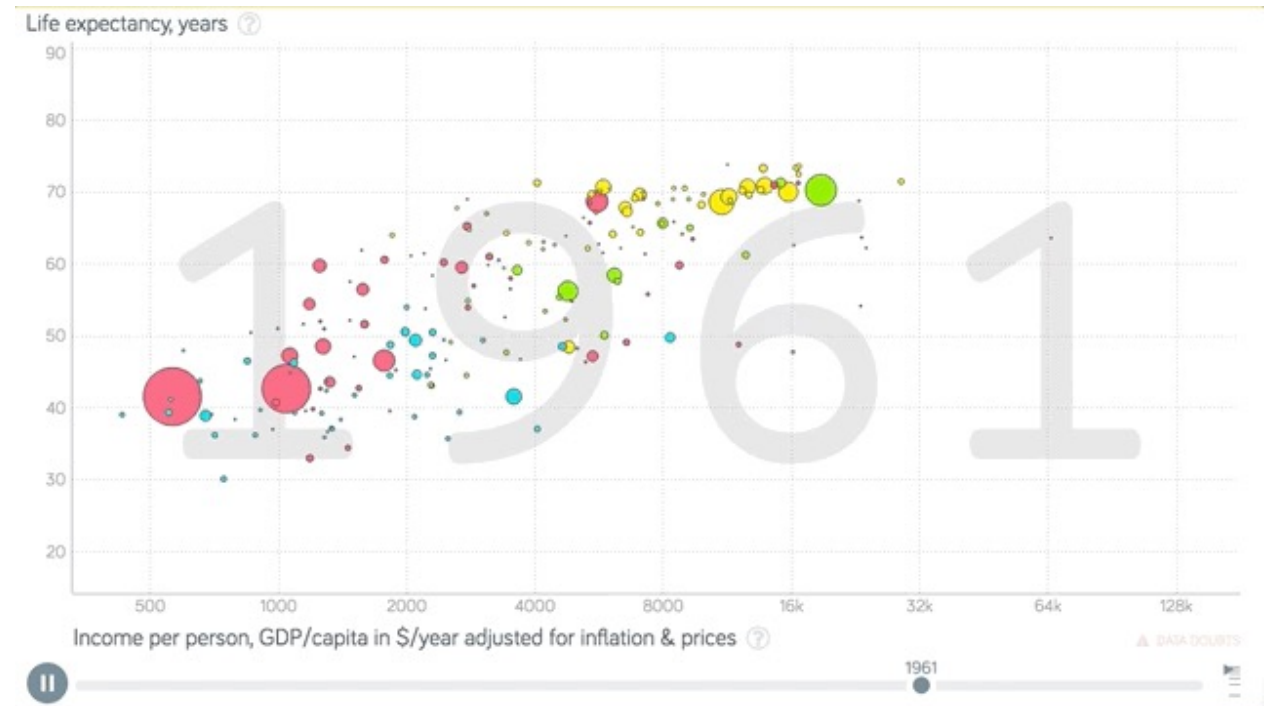
*Provided by Hong Kong Observatory*



# Interactive Data Visualization

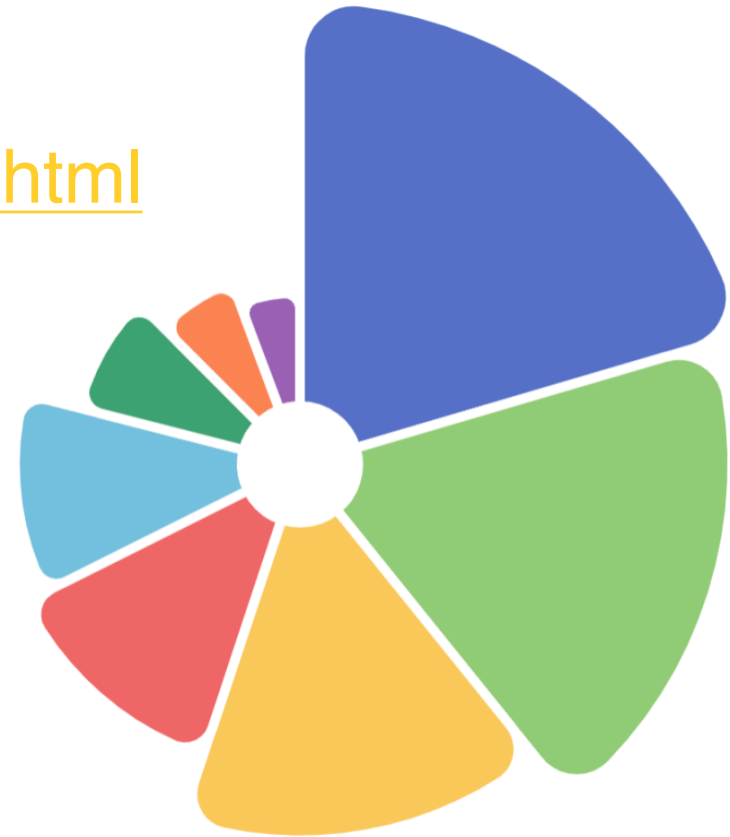
# Interactive Data Visualization

- Web-based
- Powered by JavaScript
- Presentation is dynamic
- Allows user interactions



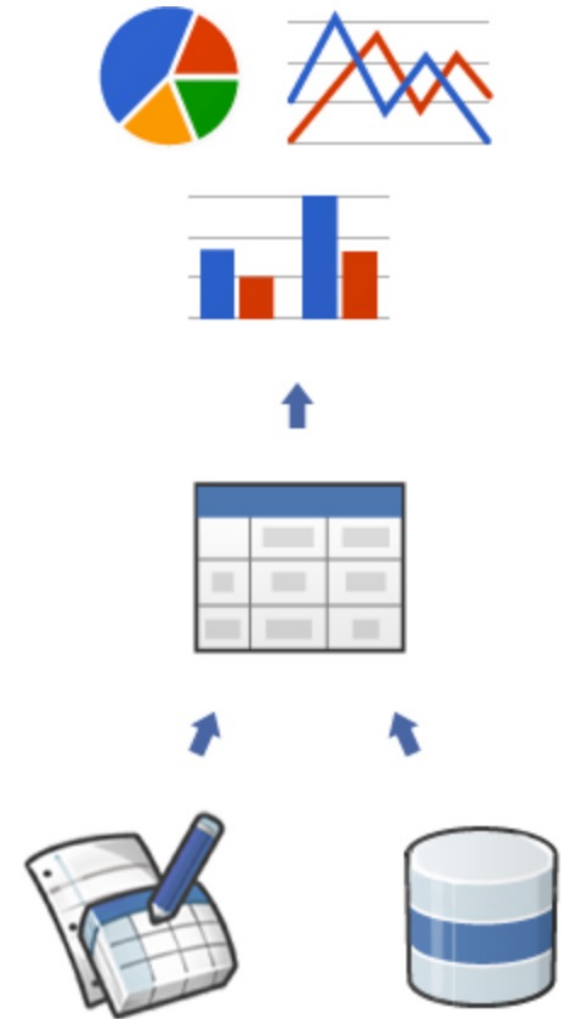
# Apache eCharts

- <http://echarts.apache.org/en/index.html>



# Google Charts

- <https://developers.google.com/chart>



# D3.js

- <https://d3js.org/>



# By using CDN

- Content Delivery Network
- Make the file loading a lot faster than hosting on own web server
- <https://www.jsdelivr.com/>
- You can load required JS from CDN instead of hosting you own copy. It's usually faster to use CND.



**JSDELIVR**

# Popular CDN for Web Libs

## d3 CDN Files

The screenshot displays the d3.js CDN file manager interface for version 6.7.0. The interface includes a header with the 'd3' logo and a dropdown menu for the version '6.7.0'. A callout box points to the 'Default file: /dist/d3.min.js' text. Below the header, a list of files is shown, each with a file icon and a toggle switch. A context menu is open over the first file, listing options: 'Copy URL', 'Copy SRI', 'Copy HTML', and 'Copy HTML + SRI'. Another callout box points to the 'Copy to Clipboard' button in the top right corner of the interface.

File Name	Copy Icon	Toggle
/npm/d3@6.7.0/dist/d3.min.js	📄	🔴
/npm/d3@6.7.0/dist	📁	
/npm/d3@6.7.0/CHANGES.md	📄	🔴
/npm/d3@6.7.0/index.js	📄	🔴
/npm/d3@6.7.0/LICENSE	📄	🔴
/npm/d3@6.7.0/package.json	📄	🔴
/npm/d3@6.7.0/README.md	📄	🔴

<https://observablehq.com/>  
Notebook-like interactive environment



# Observable

<https://observablehq.com/>

## Exploring Olympic athletes data

id	name	nationality	sex	date_of_birth	height	weight	sport	gold
736,041,664	A Jesus Garcia	ESP	male	1969-10-17	1.72	64	athletics	1
532,037,425	A Lam Shin	KOR	female	1986-09-23	1.68	56	fencing	1
435,962,603	Aaron Brown	CAN	male	1992-05-27	1.98	79	athletics	1
521,041,435	Aaron Cook	MDA	male	1991-01-02	1.83	80	taekwondo	1
33,922,579	Aaron Gate	NZL	male	1990-11-26	1.81	71	cycling	1
173,071,782	Aaron Royle	AUS	male	1990-01-26	1.8	67	triathlon	1
266,237,702	Aaron Russell	USA	male	1993-06-04	2.05	98	volleyball	1
382,571,888	Aaron Younger	AUS	male	1991-09-25	1.93	100	aquatics	1
87,689,776	Aauri Lorena Bokesa	ESP	female	1988-12-14	1.8	62	athletics	1
997,877,719	Ababel Yeshaneh	ETH	female	1991-07-22	1.65	54	athletics	1
242,604,681	Abadi Haddis	ETH	male	1987-11-06	1.7	62	athletics	1

```
viewof table = Inputs.table(olympians)
```

### Explore Olympic athletes with charts 1 of 7

This guide builds on [First steps with Observable](#).

Let's explore some data! You're probably familiar with the tabular view of data, like the one you see on the left. Each entry (or in our case, athlete) is represented by a row, and each column contains a different piece of data about them.

To see what data in Observable looks like, create a new JavaScript cell by clicking a **+** button on the left and selecting **JavaScript**. Then type the variable name `olympians` into your new cell. This is one of the [sample datasets from our standard library](#).

```
olympians
```

Run the cell by clicking **▶ Run cell** or pressing **shift-return**. You will see the word `Array` with a triangle next to it. An array is a group of objects that usually share the same structure. Click the triangle to expand the array into its objects, then click the triangle next to an object to see its properties.



# Source Codes Download

`bit.ly/3ULJkNk`

# Source Code Download

- <https://github.com/ngsanluk/JavaScriptDataVisualization.git>

The screenshot shows the GitHub repository page for 'ngsanluk / JavaScriptDataVisualization'. The 'Code' button is highlighted with a yellow circle. A dropdown menu is open, showing options: 'Clone' (with sub-options for HTTPS, SSH, and GitHub CLI), 'Open with GitHub Desktop', and 'Download ZIP' (which is also highlighted with a yellow circle). A yellow arrow points from the 'Code' button to the 'Download ZIP' option. The repository's commit history is visible in the background, showing a commit by 'ngsanluk' titled 'add html demofiles' with files like '.gitignore', 'README.md', 'apache\_echarts\_demo.html', 'd3\_demo.html', 'google\_charts\_demo.html', and 'index.html'.