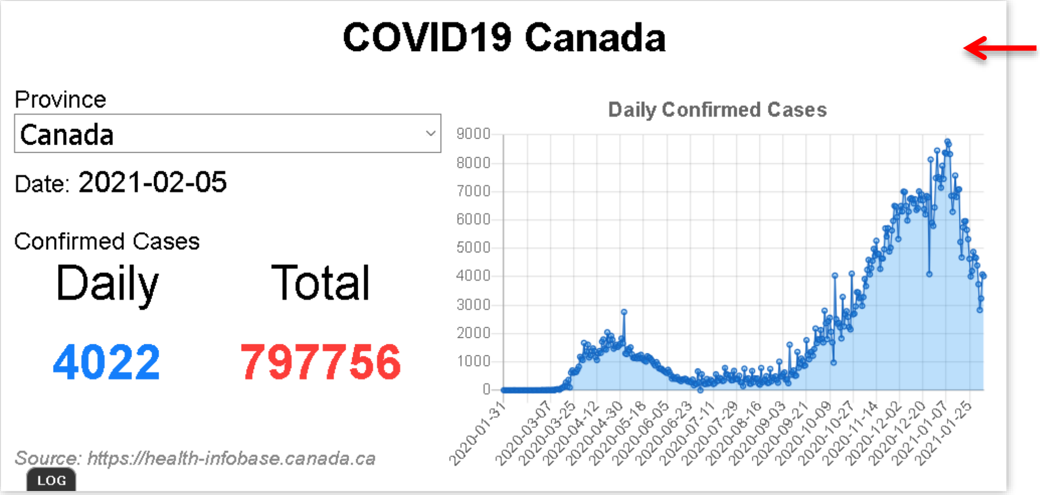
Assignment 2: COVID-19 Canada

Description

Write a responsive web application to report the number of COVID-19 confirmed cases in Canada and provinces since Jan 31 2020. When the HTML document is loaded, you request a web service to get the JSON data from a remote server asynchronously. Then, you display the Today's (the last date) daily and cumulative confirmed cases for the selected province. Also, you draw a line graph of the daily confirmed cases since 2020-01-31.



COVID19 Canada

Province: Canada

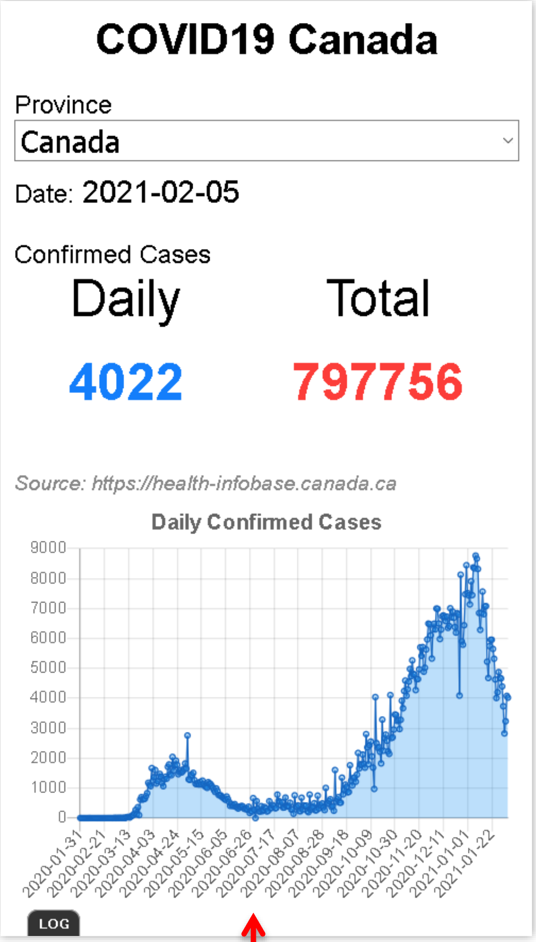
Date: 2021-02-05

Confirmed Cases

Daily	Total
4022	797756

Source: https://health-infobase.canada.ca

Display Width > 600 px



COVID19 Canada

Province: Canada

Date: 2021-02-05

Confirmed Cases

Daily	Total
4022	797756

Source: https://health-infobase.canada.ca

Display Width < 600 px

The original source data is available from https://health-infobase.canada.ca, and it is converted to JSON format at http://ejd.songho.ca/ios/covid19.json. The JSON data structure is an array of dictionaries, and it looks like;

```
[
    {
        "pruid": 35,
        "prname": "Ontario",
        "date": "2020-01-31",
        "numconf": 3,
        "numdeaths": 0,
        "numtotal": 3,
        "numtested": 0,
        "numrecover": 0,
        "numtoday": 3,
        "numdeathstoday": 0,
        "numtestedtoday": 0,
        "numrecoveredtoday": 0,
        "numactive": 3
    },
    ...
]
```

NOTE: You need only 4 properties from the JSON data:

prname	: Name of the province
date	: Date of the data entered
numtotal	: Cumulative numbers up to date
numtoday	: Number of cases on the date

Requirements

- Must be <u>responsive</u> width, height, etc. using 2-column CSS flex layout
- Must be <u>unobtrusive</u> (no inline CSS and JavaScript)
- Must be <u>validated</u> (no errors or warnings in HTML, CSS & JavaScript)
- Can use native JavaScript, jQuery and/or third-party libraries or frameworks
- Must load the JSON <u>asynchronously</u> using <u>Promise + Arrow</u> from http://ejd.songho.ca/ios/covid19.json
- Must include <u>Canada (default)</u>, <u>Ontario</u>, <u>Quebec</u> and <u>British Columbia</u> into <select> options
- Must highlight the daily/total numbers with <u>bigger font size and accent colours</u>
- Must display the <u>latest values</u> in the daily/total numbers
- Must Include the <u>title</u> and <u>labels</u> in the chart
- Must be <u>responsive width/height</u> for the chart (fit in the container block)

Deliverables

An archive file, **Assignment2_*<yourname>*.zip**, which contains all the files (HTML, CSS, JavaScript).
NOTE: *You must include the file header at the beginning of each file. The header must contain a short description, your name, email, date, etc.*

Submission and Due Date

Submit your deliverables to SLATE/Assignments/Assignment2 by **Saturday, Mar. 6, 11:59 PM**.
You may submit multiple versions, but only the latest version will be evaluated.
NOTE: *Late submission will be deducted 10% per day. (max. 3 days)*
NOTE: *Partial implementation will be accepted.*
NOTE: *This assignment is individual work and subject to Sheridan Academic Integrity Policy.*

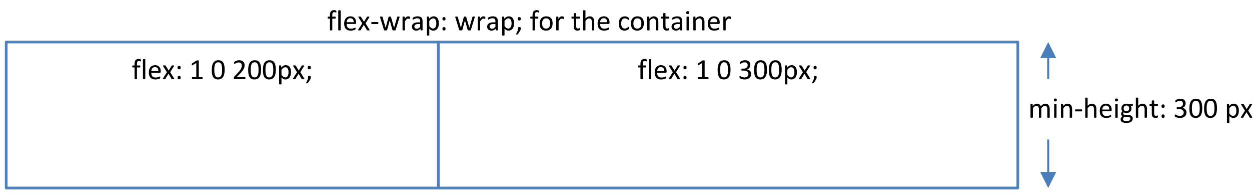
Tasks and Evaluations (Total 100 points)

Task 1: HTML (20 points)

1. Construct static DOM elements and placeholders in HTML page
2. Must have the title <h1> of the page, <u>COVID-19 Canada</u>
3. Load necessary CSS and JavaScript files
4. <u>Not allowed inline</u> CSS styles and JavaScript codes
5. Must contain "Canada" (default), "Ontario", "Quebec" and "British Columbia" <option> tags in the province <select> element

Task 2: CSS (20 points)

1. Use 2-column <u>flex</u> layout for the main block, allow automatically to wrap the columns for the width



2. The first column item contains province names in <select>, the latest date and the numbers for the daily/total confirmed cases
3. The second column item contains the chart, must be responsive (fit the chart in this block)
4. The daily and total numbers are aligned center
5. Use bigger font sizes for province names, date, daily and total numbers
6. Use accent colours for daily and total numbers

Task 3: JavaScript (60 points)

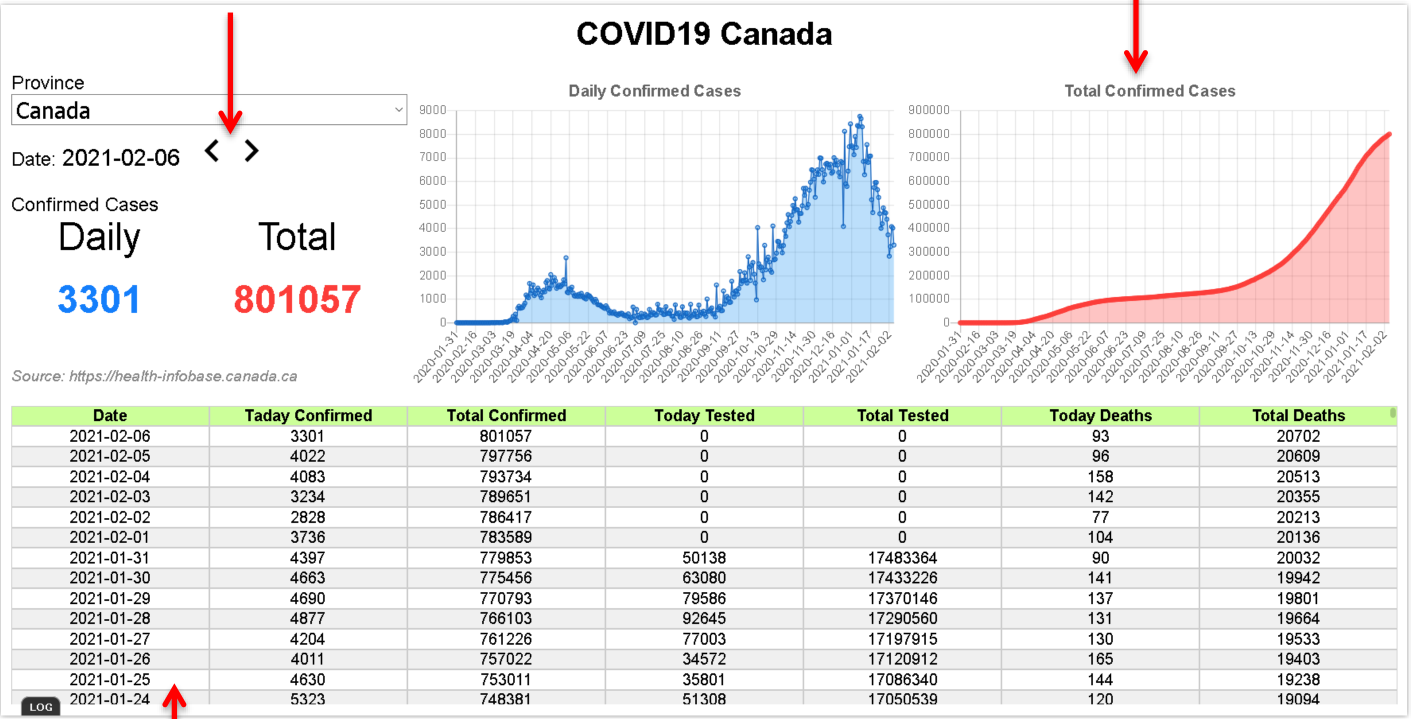
1. Load covid19.json from http://ejd.songho.ca/ios/covid19.json only once when the document is loaded
2. Must use Promise and Arrow function to load the JSON asynchronously
3. After the JSON loaded, create an array store the data for the selected province only, then sort the array by the date string in ascending order
   HINT: Use **filter()** and **sort()** array functions (Declarative Programming)
4. Display the daily and total numbers from the last element of the array
5. Generate 2 additional arrays for drawing the line graph; “values” array is to store all daily cases since 2020-01-31, and “dates” array is to store all date strings since 2020-01-31
   NOTE: The original JSON data contains only when the data are entered and skips if there is no records. However, these arrays must be continuous from the first date to up to date without hopping in order to draw the chart accurately.
6. Must compute the number of dates from the first to the last date to determine the size of “values” and “dates” arrays
7. Draw the line graph corresponding the selected province
   HINT: Use 3rd-party chart library, such as Chart.js. An example of Chart.js, **test_chart.html** is available in SLATE/Assignment2 folder.

Bonus (20 points max)

- Add additional provinces into <select> (1 point per province)
- Add the previous/next buttons to change the date (5 points)
- Add an additional graph with 3-column flex layout, e.g.: chart for the total confirmed cases (5 points)
- Add a table to display all data for the selected province (5 points)
- Add additional features to enhance the application (5 points)

Navigate other dates with buttons

Line graph for cumulative cases



COVID19 Canada

Province

Canada

Date: 2021-02-06 < >

Confirmed Cases

Daily Total

3301 801057

Source: https://health-infobase.canada.ca

Date	Taday Confirmed	Total Confirmed	Today Tested	Total Tested	Today Deaths	Total Deaths
2021-02-06	3301	801057	0	0	93	20702
2021-02-05	4022	797756	0	0	96	20609
2021-02-04	4083	793734	0	0	158	20513
2021-02-03	3234	789651	0	0	142	20355
2021-02-02	2828	786417	0	0	77	20213
2021-02-01	3736	783589	0	0	104	20136
2021-01-31	4397	779853	50138	17483364	90	20032
2021-01-30	4663	775456	63080	17433226	141	19942
2021-01-29	4690	770793	79586	17370146	137	19801
2021-01-28	4877	766103	92645	17290560	131	19664
2021-01-27	4204	761226	77003	17197915	130	19533
2021-01-26	4011	757022	34572	17120912	165	19403
2021-01-25	4630	753011	35801	17086340	144	19238
2021-01-24	5323	748381	51308	17050539	120	19094

Display all the data into a table for the selected province

Q & A (Extra Notes)

Q1. How to find the data for the selected province and create an array?

You may use your own for-loop, find the date where “**prname**” equals to the selected province and append it to array (Imperative programming). However, you can also use Array.**filter**() function instead (Declarative programming).
Reference: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/filter

```
go.json = …;  // suppose it stores the original JSON data
go.provData = go.json.filter( e => … );
```

Q2. How to sort an array by the date string in ascending order?

The date string the JSON data is in ISO format, yyyy-MM-dd. You can use String.**localeCompare**() and Array.**sort()** functions.
Reference: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/String/localeCompare

```
let date1 = “2020-01-31”;
let date2 = “2020-02-08”;
let result = date1.localeCompare(date2); // return -1, means date1 comes first

go.provData.sort( (a,b) => … );
```

Q3. How to calculate the number of elements for “values” and “dates” arrays?

These arrays should be continuous from the first date to the last date without skipping to draw an accurate chart. Use the difference of the milliseconds between 2 dates, then divide it by the ms per day.

```
const MS_PER_DAY = 24 * 60 * 60 * 1000; // ms
let time1 = new Date(“2020-01-31”).getTime();
let time2 = new Date(“2020-02-08”).getTime();
let dateCount = (time2 – time1) / MS_PER_DAY + 1; // should be 9 days, NOT 8
```

Q4. How to allocate an array with a given size and initialize it?

Use array constructor with the size parameter, then use Array.**fill**() to initialize all elements.
Reference: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/Array

```
let dateCount = 10;
let values = new Array(dateCount).fill(0); // 10 elements
```

Q5. How to populate “values” and “dates” arrays?

Use a for-loop and compute the index of the array using the similar algorithm in Q3.

```
let firstTime = new Date(go.provData[0].date).getTime(); // ms of 1st element of array
for(let e of go.provData)  // you may use forEach()
{
    let currTime = new Date(e.date).getTime(); // ms
    let index = (currTime – firstTime) / MS_PER_DAY;
    values[index] = e.numtoday;
}
```