Candidate numbers: 134, 136, 149, 117

Report

1. In our program the datasets are not downloaded at the same time. XMLHttpRequset is not synchronous, this means that the function gives a callback when all data is recieved. The data is loaded in line 5-7 in `LoadData.js`, that makes the three objects with the data. They all have their own constructor. This is since one of them had a bit different setup than the two others.
2. On line 17 in `LoadData.js` we check if all the objects to the datasets are loaded, and if they are that means that the datasets have been downloaded. We had to put in a set timeout (10 ms) to handle the `Uncaught RangeError: Maximum call stack size exceeded`.
3. We are using @media in our CSS to check if the size of the screen is below 700 pixels and if so display the content vertically, and if not horizontally. (we decided ourselves that 700 pixels and less is a small screen)

```
@media (max-width:700px){
    #row, #compareSpan {
        display: flex;
        flex-direction: column;
    }
}
```

4. The three datasets doesn't have the same amount of municipalities. The education data have more municipalities than the other two. Used a `Python`-program that counted all municipalities in each set. The education set had more then the rest. Lardal is an example of a municipality that exists in the third dataset, but not in the other two.

Description of files

DataLoader.js

- Class that loads a dataset from a given URL

Population.js

- Class which uses DataLoader to save the population-dataset. It has functions to receive/get information such as total population and population growth.

Education.js

- Class which uses DataLoader to save the education-dataset. The class has functions that receive percentage or amount of higher-educated given the municipality, and the education percentage given the municipality, category and gender.

Employment.js

- Class which uses DataLoader to save the employed-dataset. Has functions that receive the percentage or amount of employed given the municipality.

Overview.js

- Makes the overviewtable with municipalityname, municipalitynumber, population and population-growth.

Details.js

- Makes overview over the recent data, and a table with data from the last ten years.

Compare.js

- Receives education data for two given municipalities, makes tables for both of them and the municipality with the highest score in most of the categories wins.

LoadData.js

- Makes datasets with the three contructions, and has the function of displaying the correct page given which button is pressed. This file also makes it possible to press `enter` on the keyboard insted of pressing the buttons on screen. And removes the loading messages when the datasets are fully loaded.

index.html

- Displays the correct content. Most of the content is attributed to Compare.js, Overview.js and Details.js

style.css

- This file styles the HTML file. This file is among other things responsible for the table going from horizontal to vertical when the screen size goes from big to small.

loadingGIF.gif

- Gif that shows while the page is loading the data from the API.