Vår 2019 - INFO134 Semesteroppgave: Folk

Kandidatnummer

- 242
- 222
- 267

Filbeskrivelser

index.html : Den eneste html filen i innleveringen som inneholder all markup for applikasjonen, utenom det som blir generert av javascript.

folk.css : All stil-informasjon for applikasjonen.

folk.js : Inneholder hovedlogikk for hvordan data skal presenteres for det forskjellige seksjonene som; oversikt, detaljer osv.

CommonDataset.js : Inneholder kontruktør for fellesdataset-objekt, fungerer som grensesnitt for populasjons- og sysselsettings-dataene.

EduDataset.js : Inneholder kontruktør for utdanningdataset-objekt, fungere som grensesnitt for utdanningsdataene.

Spørsmål og svar

1. **Lastes datasettene ned samtidig eller etter hverandre av deres program? Begrunn svaret ditt. Henvis gjerne til koden og forklar når de tre forespørslene blir sendt. (Du trenger ikke å rettferdiggjøre hvorfor deres program laster inn dataene på denne måten.)**

   Svar: Datasettene blir lastet ned etter hverandre i vårt program, i en eller annen rekkefølge siden selve "forespørslene" er asynkrone. Om man ser i init() funksjonen som er definert i folks.js så vil man få øye på at onclick() funksjoner blir definert for alle navigasjonsknappene. For oversikt, detaljer og sammenligning vil onclick() laste datasettene som er nødvendige for den gitte seksjonen av programmet, den skjekker dette ved å teste om de globale variablene populationData, employmentData, educationData er definert.

   Merk at oversikt-navigasjonsknapp laster kun populasjonsdata, sammenligningsknapp laster kun sysselsettingsdata og detaljerknapp laster alle 3 datasettene. Om en gitt seksjonen har lastet data nødvendig for en annen seksjon så vil ikke de dataene bli lastet ned på nytt.

```
/**
 * Sets the initial condition for the application
 *  1. Add eventlisteners where needed which fetch datasets and handles searches
 *  2. Hide all sections except the introduction
 */
function init() {
    document.getElementById("introButton").onclick = function () {
        toggleSectionVisibility(introSection);
    }
    document.getElementById("overviewButton").onclick = function () {
        toggleSectionVisibility(overviewSection);
        if (!populationData) populationData = fetchStatisticalData(FetchType.population, CommonDataset, populateOverview);
        else if (!overviewPopulated) populateOverview();
    }
    document.getElementById("detailsButton").onclick = function () {
        toggleSectionVisibility(detailsSection);
        if (!populationData) populationData = fetchStatisticalData(FetchType.population, CommonDataset);
        if (!employmentData) employmentData = fetchStatisticalData(FetchType.employment, CommonDataset);
        if (!educationData) educationData = fetchStatisticalData(FetchType.education, EduDataset);
    }
    document.getElementById("comparisonButton").onclick = function () {
        toggleSectionVisibility(comparisonSection);
        if (!employmentData) employmentData = fetchStatisticalData(FetchType.employment, CommonDataset);
    }
```

2. **Hvordan vet programmet deres når det tredje (siste) datasettet er lastet ned. Begrunn svaret deres. (Henvis gjerne til en variabel, eller et sted i koden der dette er sikkert.)**

   Svar: Navigasjonsknappene setter i gang lasting av de ulike datasettene (se bruk av fetchStatisticalData() i init() i folk.js). På søkeknappene er onclick() definert slik at den sjekker om de nødvendige datasettene er definert. Skulle de ikke være definert når brukeren trykker på knappene vil ikke funksjoner som er avhengig av disse bli kjørt.

   Vi kan være sikker på at alle datasettene er lastet når handleDetailsSearch() blir kallet på via "Finn Kommune" knappen, siden dette er den eneste funksjonen avhengig av alle 3 datasettene. Se gjerne: "document.getElementById("detailsSubmit").onclick = ..." i init() i folk.js.

```
    // Both events check that the required data has loaded before trying to execute after button-click
    document.getElementById("detailsSubmit").onclick = function() {
        if (populationData && employmentData && educationData) handleDetailsSearch();
    }
    document.getElementById("compareSubmit").onclick = function() {
        if (employmentData) handleComparisonSearch();
    }
```

   En alternativ løsning kunne f.eks være å la onload-funksjoner på datasettene si i fra via variabler når det gitte datasettet er klart, og i etterkant skjekke verdien av disse.

3. **På små skjermer skal de historiske dataene presenteres vertikalt. På store skjermer skal de presenteres horisontalt. Forklar hvordan dere har løst dette. (Henvis gjerne til CSS-koden deres.)**

   Svar: Vi har laget "div"-er med klassen "historyElement" som vi igjen har lagt inn i "div"-er som enten har id "detailsHistory" eller "compareHistory". Disse 2 "foreldre"-elementene (detailsHistory og compareHistory) er satt i CSS til å være flexbox-er med en flex-wrap satt til "wrap". Dette gjør at elementene sprer seg ut horisontalt så langt det går når skjermen vokser, men skyver "historyElement"-ene våre under hverandre til et vertikalt perspektiv når skjermen krymper.

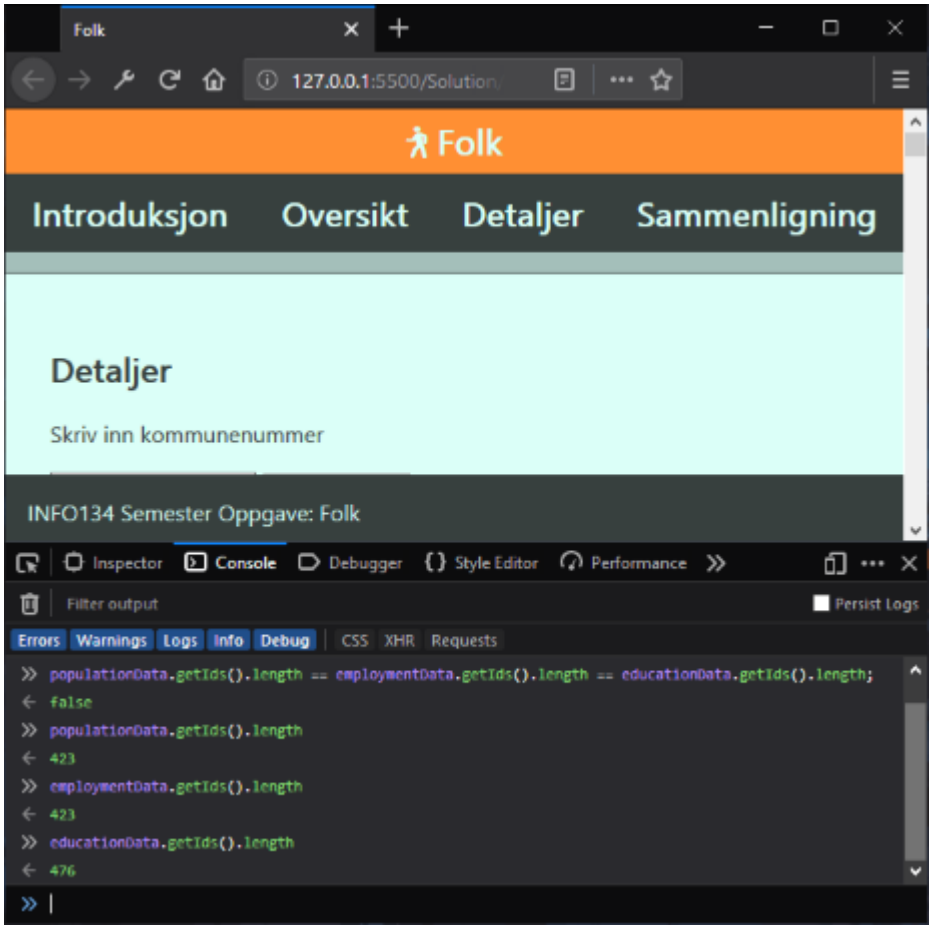
   Se hovedsakling #detailsHistory, #compareHistory og .historyElement i folk.css.

```css
#detailsHistory, #compareHistory {
    display: flex;
    flex-wrap: wrap;
}

.historyElement {
    padding: 1em;
    margin-right: 1em;
    margin-bottom: 1em;
    border-radius: 0.5em;
    border: solid 2px white;
    box-shadow: 0 8px 10px 0 black;
    color: #DBFFF8;
    background-color: #37403E;
}
```

4. **Har alle tre datasett nøyaktig de samme kommunene? Forklar kort hvordan dere fant dette svaret. Dere trenger ikke å legge ved ekstra kode hvis dere har skrevet kode for å svare på dette spørsmålet, men bare forklare fremgangsmåten deres.**

   Svar: Alle datasettene har ikke nøyaktig de samme kommunene, dette kan man f.eks skjekke på variablene vi selv initialiserer i "detaljer"-seksjonen. Om du i konsollen her skriver: "populationData.getIds().length == employmentData.getIds().length == educationData.getIds().length;" Så vil dette returnere "false", og de er dermed ulike - siden getIds() innehold alle kommune nummer for de ulike settene. (Utdanningdatasettet er forsåvidt det største etter det vi undersøkte)



Håndteringen for dette skjer i handleDetailsSearch() og handleComparisonSearch() i folk.js, om getInfo(enGittId) på et av datasettene ikke finner data for en kommune så vil retur verdien bli undefined. I de 2 metodene vil du finne kode som viser brukeren et error om ugyldig id om dette er tilfelle.

```
128     // Error-handling if one of the data-lookups end up as undefined
129     if (!disPopulationData || !disEmploymentData || !disEducationData) {
130         document.getElementById("detailsError").innerText = "Kommune-nummer ikke gyldig";
131         return;
132     } else {
133         document.getElementById("detailsError").innerText = "";
134     }
```

```
235     // Display error to user if one of the data-lookups end up as undefined
236     if (!firstDistrict) {
237         document.getElementById("compareError").innerText = "Første kommune-nummer ikke gyldig";
238         return;
239     } else if (!secondDistrict) {
240         document.getElementById("compareError").innerText = "Andre kommune-nummer ikke gyldig";
241         return;
242     } else {
243         document.getElementById("compareError").innerText = "";
244     }
```