Rapport Eksamen INFO134

Beskrivelse av filer:

hovedinnhold.html: Html-dokumentet inneholder en navigasjonsmeny med fire knapper, fire blokkelementer der kun 1 er synlig til enhver tid og en «footer». Dokumentet innholder også metadata for informasjonen som skal vises på siden (i <head)

styles.css: Inneholder utformings-data for nettstedet. Her tilpasser vi innholdet til nettsiden og endrer innholdet i forhold til type enhet som blir brukt.

main.js: Inneholder konstruktøren, definerer dataobjekter, angir oppførsel til knappene, lager tabeller og laster ned data.

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.

   Svar: Datasettene lastes ned etter hverandre. Hvert av datasettene blir lastet ned når load-metoden til objektene blir kallet (linje 108, 114, 121, main.js). Vi lager et objekt for hvert datasett via konstruktøren også kaller vi load-metoden for hvert objekt en etter en. Da lastes også datasettet ned.

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: Siden onload-metoden for «utdanning» (linje 113, main.js) blir kallet når det siste datasettet er ferdig å laste ned så er det et tegn på at alle datasettene er lastet ned. I tillegg så blir enableNavigationButtons-funksjonen kallet i onload-metoden(linje 111, main.js), denne funksjonen(linje 29, main.js) aktiverer navigasjonsknappene i nav-bar.

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: For å presentere de historiske dataene vertikalt valgte vi å bruke «media-queries» til gjøre om tabell-radene og tabell-dataen til display:block (linje 111, 112, styles.css). Dette gjør at tabellene kan bevege seg nedover når siden blir veldig liten. For å få tabellen til å ligge ii venstre side brukte vi float:left(linje 111, styles.css) . Siden noe av teksten i tabellen er veldig lang og tar mye plass var jeg også nødt til å gjøre skriften mindre og fjerne «padding» (linje 117, styles.css) på cellene for å få plass til alle radene på veldig små skjermer (Iphone 5 og mindre). Stilen for små skjermer aktiveres når bredden på skjermen er mindre enn 513px som gjør at de fleste mobiler støttes. For innhold som flyter utover beholder sin har vi lagt inn overflow-y:scroll, dette er mest aktuelt på oversikt-delen (linje 146, styles.css)

   For store skjermer skal dataen presenteres horisontalt. For å gjøre dette har vi strukturert tabellen til å opprettes horisontalt. Deretter legger vi litt stil til på de ulike cellene og tabellen i sin helhet(linje 97, styles.css). Siden for eksempel oversikt har alt for mye informasjon til å kunne vise det på en ryddig måte, har vi lagt inn overflow-x:scroll(linje 145, styles.css) for å

kunne bla bortover radene. Teksten blir også gjort litt større på noen vertikale tabellene for å kunne lese det lettere (linje 88-90, styles.css)

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

   Svar:
   For å sjekke at alle tre datasettene inneholder nøyaktig de samme kommunene har vi to linjer i nummerSjekk-funksjonen:

```
//Skjekker at nummeret er det samme i hvert datasett
 if (befolkning.getInfo(nummer).name == sysselsatte.getInfo(nummer).name && befolkning.getInfo(nummer).name == utdanning.getInfo(nummer).name) {
   return true;
```

   Denne linjen sjekker om innholdet i befolkning, sysselsatte og utdanning er det samme ved å ta nummeret til kommunen vi søker på, lager getInfo-objektet og tar navnet på hver kommune i hvert datasett og ser om de ligger i objektet.