Rapport semesteroppgave INFO134

# Beskrivelse av filer

|  |  |
| --- | --- |
| Index.html | Dette er HTML-filen som alle de andre filene benytter til å presentere alle delene av oppgaven. |
| css/style | Stilark for alle felles regler som er uavhengige av skjermstørrelser. |
| css/styleMax700 | Dette er css-dokumentet som benyttes for å presentere HTML-dokumentet for mobil-enheter. |
| css/styleMin700 | Dette er css-dokumentet som lastes uten media-querie. Det vil si for alle skjermer over 700 piksler. |
| images/favicon | Bilde-fil som benyttes helt øverst i fanen på nettleseren. |
| images/search | Bilde-fil som brukes til å signalisere at søking kan foretas. |
| js/app | Hovedfilen som kaller funksjoner i global kontekst. |
| js/canvasGraphRender | Tegner grafer for historisk data. |
| js/renderComparison | Lager tabeller for sammenligning mellom to ulike kommuner. |
| js/renderDetails | Lager tabell for å presentere detaljerte data om en kommune. |
| js/renderOverview | Lager tabell med alle kommunenavnene og tilhørende kommunenummer samt befolkningstall. |
| js/dataset | Fil med konstruktør som fungerer som ett grensesnitt mot hvert datasett. Konstruktøren oppretter tre instanser, ett for hvert datasett. |
| js/details | Fil med konstruktør som tar inn alle datasettene og samler de i ett objekt. Gjør beregninger på dataene i det nye objektet. De fleste funksjonene bruker denne istedenfor dataset.js, da den har mer data i objektet. |
| js/formFunctions | Tre funksjoner, en for hver av Detaljar, Oversikt og Samanlikning. Sjekker at riktig kommunenummer er tastet inn. |
| js/graphHelpFunc | Hjelpefunksjoner for å ta inn data og tegne grafen. Brukes i canvasGraphRender.js. |
| js/graphObjectClassIds | Objekt som brukes til å printe ut element id og klasser. |
| js/htmlHelperFunctions | Hjelpefunksjoner som benytter HTML DOM til å opprette og manipulere elementer. |
| js/navigation | Styrer navigasjonen i HTML-dokumentet. Henter inn alle knappene i navigasjons-feltet og tilordner oppførsel. |

# Spørsmål 1

Datasettene lastes ned etter hverandre i programmet. Dette skjer i filen app.js. Hvert datasett har hver sin callback-funksjon som trigger innlastingen av neste datasett-objekt etter at det gjeldende datasettets objekt er ferdig innlastet.

# Spørsmål 2

Når callback-funksjonen til utdanning kjøres, vet programmet at det er ferdig å laste inn datasettene da dette er den siste callback-funksjonen. Ingenting presenteres før dette er gjort. Load og onload benyttes, hvor onload kun slår ut når objektet er ferdig lastet inn. Dermed kan en se i linje 47 i app.js at programmet vet at alle datasettene er ferdig lastet inn.

# Spørsmål 3

I utgangspunktet er fordelingen lagt til å være horisontal, som vanlig med tabeller. Media queries er benyttet i mobilstyle.css, slik at når skjermen er under 700 piksler bred gjør CSS-koden tabellene om til blokker med display: block og bruker float: left til å tvinge elementene til å legge seg mot venstre, slik at det presenteres nedover vertikalt.

# Spørsmål 4

Utdanning går tilbake i tid til 1970 og inneholder derfor flere kommuner enn de andre datasettene. Dette fant vi ut ved å bruke <https://jsonformatter.org/> til å telle antall kommuner i hvert objekt. Kommunen Rindal har to kommunenummer i befolkning og sysselsetting. Vi har valgt å ikke ta med dataene til det gamle kommunenummeret, da det var ufullstendig data. Details.js tar kun med detaljer om årstall som ligger i datasettet «befolkning».