

## Oppgave 11

### Liste over HTML-filer

- Index.html
- Anbefalinger.html
- Toaletter.html
- Lekeklass.html
- minfavorittlekeklass.html
- Universitet.html

### Liste over JavaScript-filer

- Js.js

### Liste over bilde-filer

- Anbefalinger.png
- Anbefalingerliten.png
- Bergenlogoblack.png
- Favoritt.png
- Lekeklass.png
- Lekeklasserliten.png
- Minfavorittlekeklassliten.png
- Toaletter.png
- Toaletterliten.png
- Universitet.png
- Universitetliten.png

### Liste over CSS-filer

- Style.css

Grunnen til at vi kun bruker en CSS-fil er fordi vi gjenbruker mange av klassene på tvers av forskjellige sider og at det gjør det lettere å gjøre sidene responsive. Gjenbruken av klasser gjør at designet er konsistent på tvers av sidene. Vi følte ikke et behov for å ha flere CSS-filer av disse grunnene.

Grunnen til at vi kun har en JS-fil er for det meste av samme grunn til at vi kun har en CSS-fil. Mange av funksjonene blir brukt på tvers av forskjellige sider. Variabler blir også brukt av forskjellige metoder.

Oppgave 1 er løst på index.html.

Oppgave 2 er løst i js.js, funksjonen har navn loadJSON().

Oppgave 3 er løst i js.js, vi har flere init() funksjoner, f.eks. initToaletter() i toaletter.html, som bruker loadJSON.

Oppgave 4 er løst i js.js og toaletter.html. Funksjonene har navn populateList(), som genererer listen, og populateMap(), som generer markers på kartet.

Oppgave 5 er løst på toaletter.html, hvor man kan utføre hurtig-, og avanserte søk. Dette gjøres med funksjonene search() og quickSearch().

Oppgave 6 er løst i js.js og lekeplass.html. Funksjonene har navn populateList(), som genererer listen, og populateMap(), som generer markers på kartet.

Oppgave 7 er løst i js.js, inne i getClosest()-funksjonen. Her tar pytagoras()-funksjonen to variabler som vi får ved å hente ut høydegrad og breddegrad fra to objekter og subtrahere dem. Verdien vi får tilbake er distansen vi måler.

Oppgave 8 er løst i js.js og minfavorittlekeplass.html. Funksjonen populateSelect() tar elementet med id="select" og legger ett <option>-element for hver lekeplass i JSON.

Oppgave 9 er løst i js.js og minfavorittlekeplass.html av funksjonen getClosest(), som sammenligner alle toalettene med den valgte lekeplassen. Det toaletter som er nærmest blir representert i <div> elementet med id="closest".

Oppgave 10 er løst i js.js og universitet.html. Den tar rominformasjon fra UiBs åpne datasett på [opendata.app.uib.no](https://opendata.app.uib.no) og lar brukeren velge hvilke romtype og hvilket fakultet de vil se rommene til. Funksjonen som blir brukt er finnRom().

Oppgave 11 er løst i denne PDFen.