

## Øvingsoppgaver

# Repetisjonsforelesning før kontinuasjonseksamen

Løsningsforslag finner du mot slutten av dokumentet.

## Oppgave 1

I denne oppgaven skal du lage et lite program som lar bruker søke etter en norsk by/tettsted basert på navn. Informasjon om byene finnes i tabellen under. Denne informasjonen bør gjøres om til JavaScript-objekter, hvor navnene på objektenes egenskaper (properties) står i parentes i tabellen. Alle objektene bør deretter legges inn i et array. For denne oppgaven bør du ha kjennskap til arrays, objekter, løkker, kontrollstrukturer og betingelser, input fra bruker, samt DOM-manipulasjon.

Navn (name)	Fylke (county)	Innbyggertall (population)	URL til bilde (imageUrl)
Alta	Finnmark	20000	<a href="https://goo.gl/i5QZJT">https://goo.gl/i5QZJT</a>
Lillehammer	Oppland	28000	<a href="https://goo.gl/YnvQwT">https://goo.gl/YnvQwT</a>
Svolvær	Nordland	5000	<a href="https://goo.gl/8rZWqg">https://goo.gl/8rZWqg</a>

### Kravspesifikasjon

Søkes det etter en by som finnes i arrayet, slik som "Lillehammer", skal det, via DOM-manipulasjon, vises frem en informasjonstekst om byen, samt et bilde (`imageUrl` i tabellen over), slik som vist under:


An embedded page at run.scrimba.com says  
Hvilken by vil du søke etter?  
  

Cancel OK

→

index.html

Lillehammer har 28000 innbyggere, og ligger i fylket Oppland



Søkes det derimot om en by som *ikke* finnes i arrayet, skal bruker få beskjed om dette, slik som vist under:

An embedded page at run.scrimba.com says  
Hvilken by vil du søke etter?  
  

Cancel OK

→

index.html

Denne byen finnes ikke!

## Oppgave 2

I denne oppgaven skal det lages et lite spill vi tidligere har kodet *live* i forelesning. Spillet går ut på å hente inn et tall fra spiller 1, som spiller 2 skal gjette seg frem til basert på tilbakemeldingene "tallet du gjettet er for høyt" eller "tallet du gjettet er for lavt". Spillet skal fortsette helt til spiller 1 eller 2 enten skriver "Avbryt" i `prompt()`, eller vinner spillet ved å gjette riktig tall. Under er en visuell kravspesifikasjon med de tre scenarier det skal kontrolleres for i spillet. Ønsker du mer kompleksitet kan du implementere spillet ved bruk av DOM-manipulasjon med `<input />`-bokser istedenfor `prompt()`.

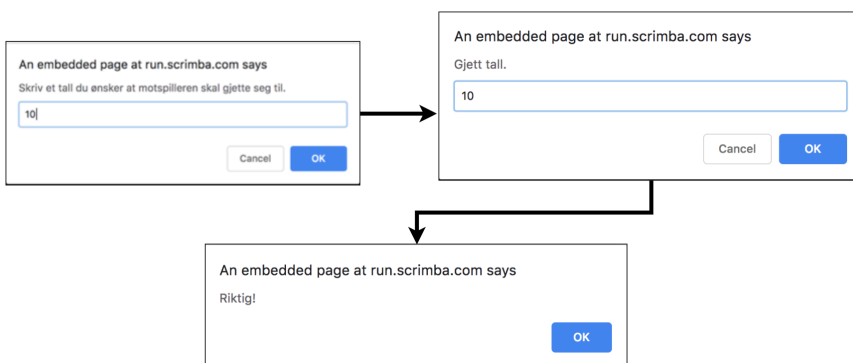
### Scenario 1: Spiller gjetter for lavt



### Scenario 2: Spiller gjetter for høyt



### Scenario 3: Spiller gjetter riktig



## Oppgave 1: Løsningsforslag

### HTML:

```
<html>
  <body>
    <p id="cityTitleTag"></p>
    <img id="imageTag" style="width: auto; height: 200px;" />
    <script src="index.js"></script>
  </body>
</html>
```

### JavaScript:

```
// Først lager vi referanser til DOM-elementene vi ønsker å manipulere senere i
programmet
var imageTag = document.getElementById("imageTag");
var cityTitleTag = document.getElementById("cityTitleTag");

// Henter inn data fra bruker
var userInput = prompt("Hvilken by vil du søke etter?");

// Variabel som holder på tilstanden i programmet, og endres til "true" senere i
koden hvis byen det søkes etter finnes.
var cityFound = false;

// Lager "databasen" vår, her i form av et array
var cities = [
  {
    name: "Alta",
    county: "Finnmark",
    imageUrl: "https://goo.gl/YnvQwT",
    population: 20000
  },
  {
    name: "Lillehammer",
    county: "Oppland",
    imageUrl: "https://goo.gl/YnvQwT",
    population: 28000
  },
  {
    name: "Svolvær",
    county: "Nordland",
    imageUrl: "https://goo.gl/8rZWqg",
    population: 5000
  },
];
```

```
// Traverserer arrayet (går gjennom hvert element i arrayet)
for (var i = 0; i < cities.length; i++) {

    // Sjekker om navnet brukeren skrev inn i prompt() finnes i et av objektene i
    arrayet, mer spesifikt som en del av name-egenskapen til objektene.
    if (userInput === cities[i].name) {
        // Finnes det, setter vi src-taggen til å være bilde-URL'en (imageUrl) til
        objektet, og navn i tekstelementet.
        cityFound = true;
        imageTag.src = cities[i].imageUrl;
        cityTitleTag.innerHTML = cities[i].name;
        break; // Avslutter for-løkken om vi finner riktig by, for å unngå
        unødvendig traversering og prosessorbruk.
    }
}

// Finnes derimot ikke elementet, sier vi ifra om det til bruker.
if (!cityFound) {
    cityTitleTag.innerHTML = "Denne byen finnes ikke!";
}
```

## Oppgave 2: Løsningsforslag

```
// Først spør vi bruker om tallet den andre brukeren skal gjette på.
var questionInput = prompt("Skriv et tall du ønsker at motspilleren skal gjette
seg til.");

// Variabelen holder tilstanden i spillet, og bestemmer om spillet skal fortsette.
var continueGame = true;

// Her bruker vi en do-while fordi vi vet at vi skal kjøre logikken i spillet
minst én gang.
do {

    // Først blir bruker om å gjette et tall.
    var answerInput = prompt("Gjett tall.");

    /* Siden vi vil at spillet skal kunne avbrytes når som helst, sjekker vi om
    bruker i noen av prompt()-boksene skriver inn Avbryt. I så fall setter vi
    tilstandsvariabelen i spillet til false slik at spillet avbryter. */
    if (questionInput === "Avbryt" || answerInput === "Avbryt") {
        continueGame = false;
    } else {
        // Først gjør vi om spørsmål- og svar-variablene til numeriske verdier
        slik at vi enklere kan sammenligne verdiene deres.
        var questionAsNumber = parseInt(questionInput);
        var answerAsNumber = parseInt(answerInput);

        // Så sjekker vi om spørsmål-variabelen er større enn svarvariabelen...
        if (questionAsNumber > answerAsNumber) {
            alert("Ikke så lavt! Prøv igjen.");

            // ... eller omvendt
        } else if (questionAsNumber < answerAsNumber) {
            alert("Ikke så høyt! Prøv igjen.");

            // ... eller om spørsmål- og svar-variablene er like store, og at
            spilleren dermed har vunnet.
        } else {
            alert("Riktig!")
            continueGame = false;
        }
    }
} while (continueGame); // Spillet fortsetter helt til continueGame-variabelen
settes til false.
```