



JavaScript Oppgavebank

I dette dokumentet ligger oppgaver vi jobber med for hvert tema. Det er anbefalt å begynne på starten og jobbe seg igjennom tema for tema. Dersom du selv opplever at du kan et tema, kan du gå videre til neste tema og sjekke om du har kontroll på det temaet. Dersom du finner ting her som du ikke har lært enda, er det tenkt at du skal finne ut av det selv.

Anbefalte strategier for å finne hjelp:

- Sjekk ut slides fra kurset. Det er eksempler der som passer til nesten alle oppgavene.
- Spør Google på engelsk. StackOverflow gir nesten alltid bra svar!
- Sjekk ut [W3 Schools](https://www.w3schools.com/) for syntaks og eksempler.
- Spør AI! Bruk www.ki.osloskolen.no eller andre chatbotter.
- Gjerne spør om hjelp fra andre elever før du spør lærer om hjelp – det er kjempegod læring i å forklare noen andre hvordan noe fungerer. Dette gjelder både store og små ting.

Husk 1 - Vi bruker «vanilla» JavaScript.

Når du søker på ting vil du ofte få svar som inneholder biblioteker eller rammeverk. Det kan være lurt å legge til vanilla JavaScript i søkemeldingen din. Eller lete etter svar på [W3 Schools \(link\)](https://www.w3schools.com/).

Husk 2 – Vær forsiktig med å kopiere kode.

Det kan være fristende å bare copy/paste kode fra internett/AI, men det kan være lurt å tenke seg om. Poenget med dette kurset er å lære JavaScript, og da er det nok lurere å skrive koden selv.

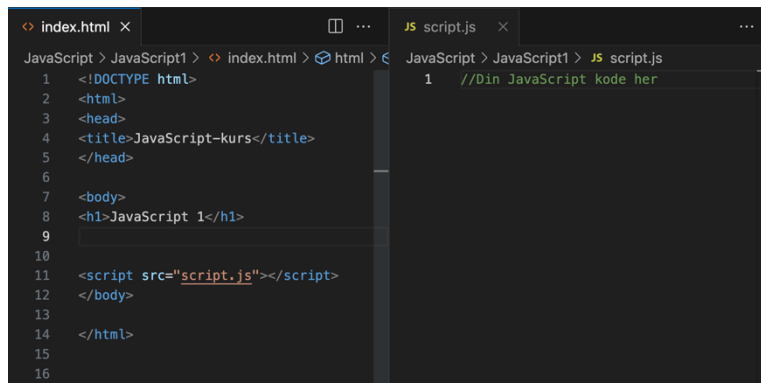
Innhold – Trykk på for å hoppe direkte i dokumentet

Kom i gang!	2
JavaScript 1 – Variabler, objekter, if/else	3
JavaScript 2 – Forms og Løkker	7
JavaScript 3 – Funksjoner	10
JavaScript 4 – Arrays og maps	13
JavaScript 5 – Komponenter, bibliotek og rammeverk	17

Kom i gang!

For å komme i gang med JavaScript trenger du en **mappe** med **to filer** slik:

- JavaScript1
 - index.html
 - script.js
- JavaScript2
 - index.html
 - script.js



```
index.html
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>JavaScript-kurs</title>
5 </head>
6
7 <body>
8 <h1>JavaScript 1</h1>
9
10
11 <script src="script.js"></script>
12 </body>
13
14 </html>
15
16

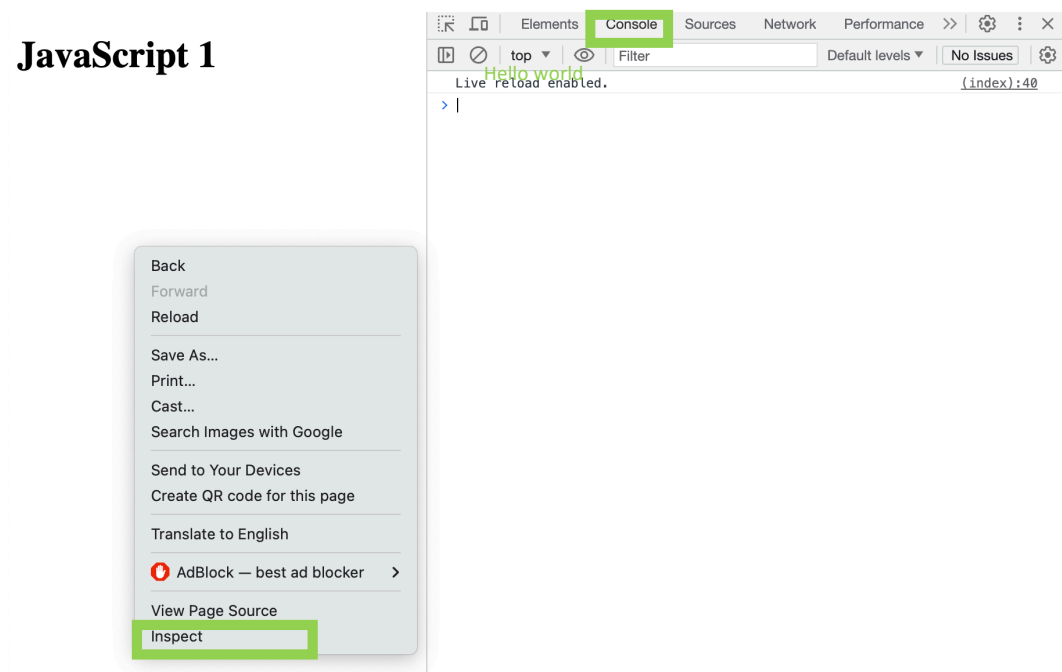
script.js
1 //Din JavaScript kode her
```

Åpne «**index.html**» i en nettleser eller bruk [Live Server extension i VS-code](#) til å kjøre koden.

Vi anbefaler å bruke Chrome som nettleser.

Husk å «**Inspect element**» og trykk på «**console**» for å finne frem til ting som blir skrevet i konsollen.

JavaScript 1



JavaScript 1 – Variabler, objekter, if/else

Variabler, objekter og datatyper

Oppgave 1

Lag en variabel som inneholder strengen «Hello World!» og skriv den ut til konsollen.

Oppgave 2

Lag en variabel med navnet ditt. Skriv ut «Hallo (ditt navn)» ved å kombinere «Hallo» med ditt navn.

Oppgave 3

Lag variablene «radius» og «pi». Pi skal ha verdien 3.14 og radius bestemmer du

Lag variabelen «omkrets» som regner ut omkretsen av en sirkel. $Omkrets = 2 * P * r$

Skriv ut «For radius (radius) er omkretsen (omkrets)»

Oppgave 4

Lag et objekt som heter person. Person-objektet skal inneholde følgende elementer

- Fornavn
- Etternavn
- Alder
- Dato for bursdagen din
- Er du over 18
- Favoritt-farge

Skriv ut hele objektet å se hvordan det ser ut.

Skriv ut «Hallo (ditt navn)» ved å kombinere «Hallo» med hele navnet i objektet.

Oppgave 5

Lag en variabel som inneholder antall bokstaver i navnet ditt. Skriv ut «Hallo (ditt navn). Ditt navn har (ditt antall bokstaver) bokstaver».

Oppgave 6

Lag en variabel som inneholder hele navnet ditt ved å kombinere variablene for fornavn og etternavn fra oppgave 4

Oppgave 7

Bruk template strenger til å skrive ut all informasjon du lagret i objektet om deg selv i oppgave 4

Teksten skal se slik ut «Hei, mitt navn er Madeleine Lorås. Jeg er 32 år gammel og har bursdag 19. januar.»

Kontrollflyt, DOM og eventer

Oppgave 8

Skriv en kodesnutt som ber brukeren om å oppgi to tall ved bruk av prompt(). Sammenlign de to tallene og skriv ut en melding til konsollen som indikerer hvilket tall som er større, eller om de er like.

Oppgave 9

Skriv en kodesnutt som spør brukeren om å oppgi alderen sin ved bruk av prompt (). Hvis alderen er 18 år eller eldre, vis en melding som sier "Du er kvalifisert til å stemme." Ellers vis en melding som sier "Du er ikke kvalifisert til å stemme ennå."

Oppgave 10

Skriv en kodesnutt som ber brukeren om å oppgi et passord ved bruk av prompt (). Hvis det oppgitte passordet er "passord123," vis en melding som sier "Tilgang innvilget." Ellers vis en melding som sier "Tilgang nektet."

Oppgave 11

Skriv en kodesnutt som sjekker om det er fredag og om du kan gå hjem. Hvis det er fredag skal du skrive «Det er fredag!» . Hvis det er fredag og klokka er 15:30 skal du skrive ut «God helg – sees på mandag! Husk stol på fredag da»

Tips! Sjekk ut Date()-objektet her: https://www.w3schools.com/jsref/jsref_getday.asp

Oppgave 12

Opprett en knapp i index.html som heter «Klikk meg»

```
<button>Klikk meg!</button>
```

Lag en klikk-funksjon i script.js

```
function handleClick() {  
    //Din kode her  
}
```

Få knappen til å skrive ut hvor mange ganger den er blitt trykket på ved å opprette en variabel som du plusser på med én hver gang

Oppgave 13

Skriv en kodesnutt som gjør at du kan velge farge på bakgrunnen.

Lag 3 knapper, med 3 forskjellige farger. Når brukeren trykker på en knapp skal bakgrunnen endres.

Oppgave 14

Skriv en kodesnutt som lager en liten tekst formaterer.

Brukeren skal kunne skrive inn valgfri tekst med bruk av `prompt()` og så endre tekstens font, fontstørrelse, font farge og farge på bakgrunnen.

Appen skal se ca. slik ut →

Din text her!

Velg font:

ArialTimes New RomanCourier New

Velg fontstørrelse:

162030

Velg farge på tekst:

HvitSvart

Velg farge på bakgrunnen:

BlåRosaLilla

Klikk her for å skrive inn tekst

Skriv inn valgfri tekst i boksen som popper opp

Utfordringsoppgave 15 - Kalkulator

Bruk triksene vi har lært til å lage en kalkulator!

Begynn med å lage to tekstbokser hvor brukeren kan skrive inn to tall (sett gjerne et plusstall mellom for å gjøre det enklere å forstå), en utregningsknapp, og en ut-tekstboks som er deaktivert.

Utregningsfunksjonen skal legge sammen de to tallene, og skrive det ut i ut-boksen.

Hvis du blir ferdig, her er noen forslag til hvordan du kan gjøre kalkulatoren bedre, i ingen spesiell rekkefølge:

- Gjør kalkulatoren penere med CSS
- Skriv en kalkulator for volum eller omkrets av geometriske former
- Legg til flere regneoperasjoner (enten under hverandre, eller kanskje man kan gjøre noe med https://www.w3schools.com/tags/tag_select.asp ?)
- Finn ut hvordan du kan gjøre så man kun kan skrive tall i tekst-boksene
- Legg til støtte for desimal-tall

Utfordringsoppgave 16 - Animasjoner

Trykk deg gjennom denne informasjonen om interaksjoner og animasjon:

https://www.w3schools.com/js/js_htmlDOM_animate.asp

Bruk dette som utgangspunkt for å animere en «loading-animation» eller såkalt spinner. Det kan være hva som helst, men skal indikere at noe laster.

Klarer du å legge til dette på «onload-event»?

Utfordringsoppgave 17 - Passordgenerator

Bruk innbygde JavaScript-funksjoner for strenger [1] til å lage en enkel passordgenerator som hjelper brukerne å lage trygge passord.

Nivå 1 – Bruk `prompt()` til å hente et substantiv fra brukeren og lagre det i en variabel.

- Bytt ut alle e-er med 3-tall
- Bytt ut allt i-er med 1-tall
- Bytt ut alle o-er med 0-er
- Lag noen av bokstavene STORE og noen små
- Legg til noen tilfeldige tegn på slutten, for eksempel `!/?#`

*For eksempel: Brukeren skriver inn **hettegenser**. Programmet ditt gjør det om til **h3TtegeN53r**?*

Nivå 2 - Bruk `prompt()` til å hente et substantiv og et adjektiv fra brukeren og lagre det i en variabel.

- Kombiner ordene og bruk de samme metodene som i nivå 1.

*For eksempel: brukeren skriver inn **hettegenser** og **prikkete**. Programmet ditt gjør det om til **Pr1KK3tEh3TtegeN53r***

Nivå 3 - Kan du tenke på flere ting du kan legge til eller endre på for å lage sterkere passord?

[1] Innbygde JavaScript-funksjoner for strenger `str.toLowerCase()`

Convert string str to lower case

`str.toUpperCase()`

Convert string str to upper case

`str.length`

Return length of string str

`str1.concat(str2)`

Combine text of strings str1 and str2 and return a new string

`str.match(regex)`

Match a regular expression regex against string str

`str1.replace(regex, str2)`

Replace matched regex elements in string str1 with string str2

`str.search(regex)`

Return position of search for a match between regex and string str

Sjekk også ut denne «Cheet-Cheet» <https://dev.to/devsmitra/javascript-string-methods-a-cheat-sheet-for-developer-4kbk>

BONUS - Bli bedre på CSS!

Sjekk ut følgende treningssider for å bli god på CSS:

- Flex box froggy – et spill som lærer deg flex
<https://flexboxfroggy.com/>
- CSS diner – et spill som lærer deg CSS selectors
<https://flukeout.github.io/>
- Grid garden – et spill som lærer deg grid
<https://cssgridgarden.com/>

Når du har klikka deg gjennom dette kan du også se på noen CSS rammeverket Tailwind som vi skal lære mer om senere i år: <https://tailwindcss.com/docs/installation/play-cdn>

JavaScript 2 – Forms og Løkker

Forms – input fra brukeren

Oppgave 0 – Recap fra JS1

Lag et program hvor en bruker kan finne ut om de kan øvelseskjøre.

For å øvelseskjøre i Norge må du være minst 16 år og bestått trafikalt grunnkurs. Hvis du er over 25 år trenger du ikke ha bestått trafikalt grunnkurs.

Bruk en input med type=number for å skrive inn alder, og en input type=checkbox for om man har bestått trafikalt grunnkurs.

Løkker og iterasjoner

Oppgave 1

Skriv en kodesnutt som skriver ut i konsollen alle tall mellom 1 og 10. Lag en løsning uten løkker, en løsning med for-løkke og en med while-løkke.

Oppgave 2

Skriv en kodesnutt som skriver ut til HTML alle tall mellom 1 og ett tallet brukeren skriver inn. Her er det lurt å sjekke hva brukeren skriver inn før du kjører løkken.

Du kan velge selv om du vil bruke prompt() eller lage inputfelt og knapp.

Oppgave 3

Bruk person-objektet du lagde i JavaScript 1. Skriv en kodesnutt som skriver ut til HTML informasjonen i person-objektet.

Hint! For in

Oppgave 4

Bruk løkker til å lage en passordgeneratoren fra forrige kurs!

Bruk prompt() eller input til å hente et substantiv fra brukeren og lagre det i en variabel.

- Bytt ut alle e-er med 3-tall
- Bytt ut allt i-er med 1-tall
- Bytt ut alle o-er med 0-er
- Lag noen av bokstavene STORE og noen små
- Legg til noen tilfeldige tegn på slutten, for eksempel !?#

For eksempel: Brukeren skriver inn **hettegenser**. Programmet ditt gjør det om til **h3TtegeN53r?**

Hint! For of

Oppgave 6

Skriv en kodesnutt som skriver ut en pyramide av stjerner ved hjelp av løkker. Brukeren skal kunne skrive inn et tall som sier hvor mange rader stjernepyramiden skal ha.

For eksempel: Brukeren skriver 4. Konsollen ser da slik ut:

```
*  
**  
***  
****
```

Oppgave 7 – Hangman!

Skriv et program som lager et enkelt hangman-spill ved bruk av løkker.

- Velg et ord brukeren skal gjette, for eksempel ord = «Elvebakken»
- Brukeren skal gjette en og en bokstav - gjettetBokstav
- Brukeren ha 10 liv – antallLiv = 10
- Bruk en forof-løkke til å sjekke om gjettet bokstav ligger i ordet du har valgt
- Bruk en while-løkke til å passe på hvor mange liv brukeren har

Ikke tenk på CSS eller utssende før etter du har laget logikken.

Oppgave 8 – Pokemon-nettside

Kopier Pokemon-objektet fra [dette dokumentet](#) eller lag ditt eget ved hjelp av en AI.

Bruk dette til å lage en kul nettside som presenterer pokemon. Du kan for eksempel lage:

- Lag en tabell eller lignende som presenterer Pokemon-data
- Vis en tilfeldig Pokémon ved hver lasting av siden, hvor farge og design på hele siden også oppdateres
- Lag en quiz for Pokémon-kunnskap
- Lag en søkefunksjon for Pokémon:

JavaScript 3 – Funksjoner

Oppgave 1

Skriv en funksjon som skriver ut teksten "Hello World!" til konsollen.

Endre funksjonen til å ta inn navn som argument og skriv ut «Hello Navn» til konsollen.

Oppgave 2

Skriv en funksjon med navn "erPalindrom", som tar en streng som parameter og returnerer true eller false avhengig av om strengen er en palindrom eller ikke. Et palindrom er en streng som leses likt fra begge retninger.

Eksempel på et funksjonskall: erPalindrom("level") skal gi true.

Oppgave 3

Skriv to funksjoner: "regnutAreal" og "regnutOmkrets". "regnutAreal" skal ta to tall som parametere (lengde og bredde) og returnere arealet av et rektangel, mens "regnutOmkrets" skal ta de samme parametrene og returnere omkretsen.

Eksempel på funksjonskall:

let length = 5;

let width = 3;

regnutAreal(length, width) → Output: 15

regnutOmkrets(length, width) → Output: 16

Oppgave 4 - Terning

I denne oppgaven skal du lage en gjenbrukbar terning-funksjon.

For å lage en terning trenger vi å lage et tilfeldig tal. For å gjøre i dette i javascript kan vi bruke den innebygde funksjonen Math.random(). Det er en funksjon som ikke tar noen parameter, og som gir én returverdi: et tilfeldig desimaltall fra 0 til, men ikke med, 1.

For å gjøre dette til et større tall kan vi rett og slett gange. Math.random() * 6, gir oss et tall fra og med null til men ikke med 6.

For å lage terningen kommer vi også til å trenge Math.floor(number), som runder ned et tall. Det er en funksjon som har ett **parameter**, tallet som skal rundes ned, og gir én **returverdi**, tallet avrundet.

Funksjonen du skal skrive, die(), skal ta ett **parameter**, hvor mange sider terningen skal ha, og gi én **returverdi**, tallet på terningen.

Test funksjonen med konsollet.

Oppgave 5 - Celcius til Fahrenheit

I denne oppgaven skal du lage en funksjon som konverterer mellom Celcius (C) og fahrenheit (F).

For å konvertere fra C til F, tar du C temperaturen og ganger med 1.8, for så å legge til 32.

Funksjonen "celciusToFahrenheit" skal ta en **parameter**: temperaturen celcius som et **number**, og gi én returverdi: fahrenheit som et **number**.

Test funksjonen med konsollet.

Oppgave 6 - Temperatur-kalkulator

I denne oppgaven skal vi bruke celcius-funksjonen i en kalkulator.

Lag en `<input type="number">` for å skrive inn tallet, en `<button>` for å aktivere utregningen, og en `<input>` med disabled for å vise output.

Onclick i knappen skal ikke kalle celciusToFahrenheit direkte, men kalle en funksjon "handleCalculate()" som henter verdier fra HTML, kaller funksjonen fra forrige oppgave, og skriver verdien ut til HTML igjen.

Oppgave 7 - Fahrenheit til Celcius

I denne oppgaven skal du utvide temperatur-kalkulatoren til å også håndtere F til C.

Lag en ny funksjon fahrenheitToCelcius som regner ut motsatt vei (-32, så dele på 1.8).

I HTML, bruk en `<select>` med "C -> F" og "F -> C" som alternativer.

I javascript, legg til en **switch**-statement i handleCalculate som sjekker hvilket alternativ som er valgt og kaller riktig funksjon.

Oppgave 8 - Farge-knapper + UU

I denne oppgaven skal du endre på fargene på en tekst ved hjelp av knapper.

Lag en h1 med valgfri tekst, og tre knapper med tekstene "Rød", "Grønn", "Blå".

Lag en funksjon "setToColor()" med ett parameter en farge som en **streng**.

Du kan sette fargene på teksten med å hente ut elementet fra HTML, og så sette f.eks `element.style.color = "yellow"`

Bruk dette til å få funksjonen til å sette farge på teksten.

Nå kan det være lurt å teste funksjonen med konsollet før du går videre.

Så skal du kalle funksjonen fra de tre knappene ved å sette onclick, men skal utnytte av vi kan sende et argument fra HTML til javascript ved å skrive det mellom parantesene. Så vi kan skrive f.eks.:

```
onclick="someFunction("argument")
```

Bruk dette til å få hver av knappene til å sette teksten til sin farge.

Utfordringsoppgave! Farge-sliders

Endre så teksten i forrige oppgave til å bli endret av tre `<input type="range">` hvor brukeren kan velge en verdi for hver av fargene mellom 0-255. Når du setter fargen må du skrive:

```
rgb([rødverdi], [grønnverdi], [blåverdi])
```

Istedenfor en onclick, bruk onchange, slik at endringene kommer med én gang. Du kan ikke sende argumentet direkte fra html til javascript i denne oppgaven.

JavaScript 4 – Arrays og maps

Oppgave 1 – Kom i gang med lister

- Opprett en liste som inneholder navnene på tre frukt: eple, pære og banan. Bruk en for-løkke for å skrive ut hver frukt til konsollen.
- Legg til kiwi og fjern eple.
- Lag en liste til for jordbær, bringebær og blåbær. Kombiner de to listene i en ny liste slik at det blir annenhver frukt og bær.
- Skriv en funksjon som tar inn en liste som parameter og printer den som en punktliste i HTML.

- Eple
- Jordbær
- Pære
- Blåbær
- Kiwi
- Bringebær

Du skal kunne se en liste med frukt og bær slik som bildet her →

Oppgave 2 - Snu slangen

I denne oppgaven skal du skrive en funksjon som snur en "slange". Slangen er en liste, hvor første element er hodet, og siste element er slangen. Mellom hodet og halen er det et ukjent antall kroppsdelar.

Funksjonen skal sette det første elementet som det siste, og det siste elementet som det første, sånn at slangen er "snudd". Funksjonen skal ta ett parameter: listen som representerer slangen, og gi én returverdi: listen som er slangen snudd.

Oppgave 3 - Minste tall

Skriv en funksjon som finner det minste tallet i en liste av tall.

Funksjonen skal ta inn en parameter: en liste over tall; og gi én returverdi: det minste tallet.

En algoritme (fremgangsmåte) for å finne det minste tallet i en liste, er å opprette en variabel med en verdi høyere enn alle elementene i listen, og så for hvert element, sette variabelen til verdien av dette elementet om den er lavere enn den nåværende verdien av variabelen.

Bonusutfordring!

Endre funksjonen til å gi en liste med to verdier: den minste verdien, og *index* til den minste verdien.

Oppgave 4 - Elev-register

I denne oppgaven skal du lage et oppdaterbart register over elever. Listen med elever skal listes opp på HTML siden og oppdateres.

Lag en liste som heter `elevListe`.

Lag et inputfelt for å skrive inn nye navn, og en knapp for å oppdatere listen. Skriv en funksjon som oppdaterer listen når knappen trykkes. Funksjonen skal sjekke om navnet fra inputfeltet ligger i listen fra før, og bare legge til nye, unike elev navn.

Representer listen i HTML med en `` med en `` for hver elev.

Oppgave 5 - High-score liste

I denne oppgaven skal du lage en oppdaterbart high score liste. La brukeren skrive inn både en score og et kallenavn i hver sin input-boks. Når det er blitt lagt til en ny score skal listen sorteres sånn at høyeste score kommer øverst.

Oppgave 6 - Matrise-skattejakt.

Bruk en (for-index) løkke inni en løkke for å lag en liste hvor hvert element er en liste: en matrise. Du kan tenke på matrisen som et rutenett med x- og y-koordinater for bredde og høyde.

Lag en funksjon som tar to parametere: høyden og bredden på matrise.

Det vil bli seende omtrent slik ut:

```
// 3 x 2:  
[[11, 12, 13],  
 [21, 22, 23],  
 [31, 32, 33]]
```

På et tilfeldig sted i matrisen, gjør en av elementene til en "X", dette er skatten!

Skriv en annen funksjon som tar en matrise som parameter. Bruk (for-of) løkke inni løkke til å gå gjennom matrisen og lete etter skatten. Gi som returverdi en liste med to verdier: x-koordinatet og y-koordinatet til skatten.

Oppgave 7 – Kom i gang med map()

Map() er en av de mest brukte innebygde funksjonene i JavaScript fordi den er utrolig nyttig når man skal presentere data i forskjellige formater. Les mer om map() her:

https://www.w3schools.com/jsref/jsref_map.asp

Nivå 1 - Dobling av tall

Skriv et program som tar inn en liste med heltall og lager en ny liste hvor hvert tall er doblet. Begge listene skal printes til konsollen.

Eksempel:

Gitt listen **[2, 4, 6]**, skal programmet returnere **[4, 8, 12]**.

Nivå 2 – Gjør fornavn til initialer

Skriv et program som tar inn en liste med fornavn og lager en ny liste med kun initialene til hvert fornavn. Begge listene skal printes til konsollen.

Eksempel:

For listen **['Alice', 'Bob', 'Charlie']**, skal eleven returnere **['A', 'B', 'C']**.

Oppgave 8 - Aksjeanalyse

I denne oppgaven skal du skrive et program som tar inn data om forskjellige aksjer og presenterer forskjellige deler av dataene som beskrevet under.

Aksjedataene er lagret i en liste med objekter. Hvert objekt representerer en aksje med egenskapene **navn**, **prisPerAksje**, **sektor** og **antallAksjerTilgjengelig**. Se eksempeldata nederst i oppgaven.

Skriv funksjoner som gjør følgende:

1. **Sektordeling:**
Returner en ny liste som inneholder unike sektorer fra alle aksjene.
2. **Aksjefilter:**
Basert på den opprinnelige listen, returner en ny liste med aksjer som koster mindre enn 100 enheter per aksje.
3. **Lite Tilgjengelig:**
Returner en ny liste med aksjenavn som har en tilgjengelighet på 0.
4. **Total Markedsverdi:**
Beregn totalverdien av alle aksjer (pris per aksje multiplisert med antall aksjer tilgjengelig).

Du kan velge å gjøre hele oppgaven i konsollen, eller lage en side i HTML

Eksempeldata:

```
const aksjer = [  
  {navn: 'TechCorp', prisPerAksje: 80, sektor: 'teknologi', antallAksjerTilgjengelig: 1000},  
  {navn: 'HealthMed', prisPerAksje: 150, sektor: 'helse', antallAksjerTilgjengelig: 5000},  
  {navn: 'EduLearn', prisPerAksje: 40, sektor: 'utdanning', antallAksjerTilgjengelig: 0},  
  {navn: 'FoodYum', prisPerAksje: 200, sektor: 'mat', antallAksjerTilgjengelig: 3000}  
]
```

Forventet svar for eksempeldata:

1. ['teknologi', 'helse', 'utdanning', 'mat']
2. [{navn: 'TechCorp', prisPerAksje: 80, sektor: 'teknologi', antallAksjerTilgjengelig: 1000}, {navn: 'EduLearn', prisPerAksje: 40, sektor: 'utdanning', antallAksjerTilgjengelig: 0}]
3. ['EduLearn']
4. $80 * 1000 + 150 * 5000 + 200 * 3000 = 1430000$

JavaScript 5 – Komponenter, bibliotek og rammeverk