

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](#) 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 \(link\)](#).

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	11
JavaScript 4 – Arrays og maps	Error! Bookmark not defined.
JavaScript 5 – Komponenter, bibliotek og rammeverk	Error! Bookmark not defined.

Kom i gang!

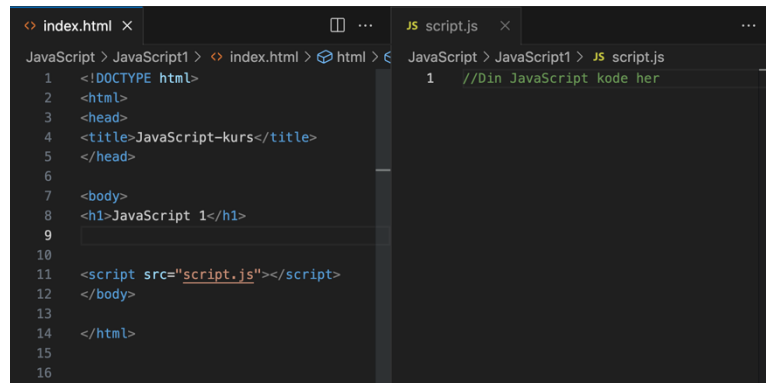
For å komme i gang med JavaScript trenger du en **mappe** med **to filer** slik:

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

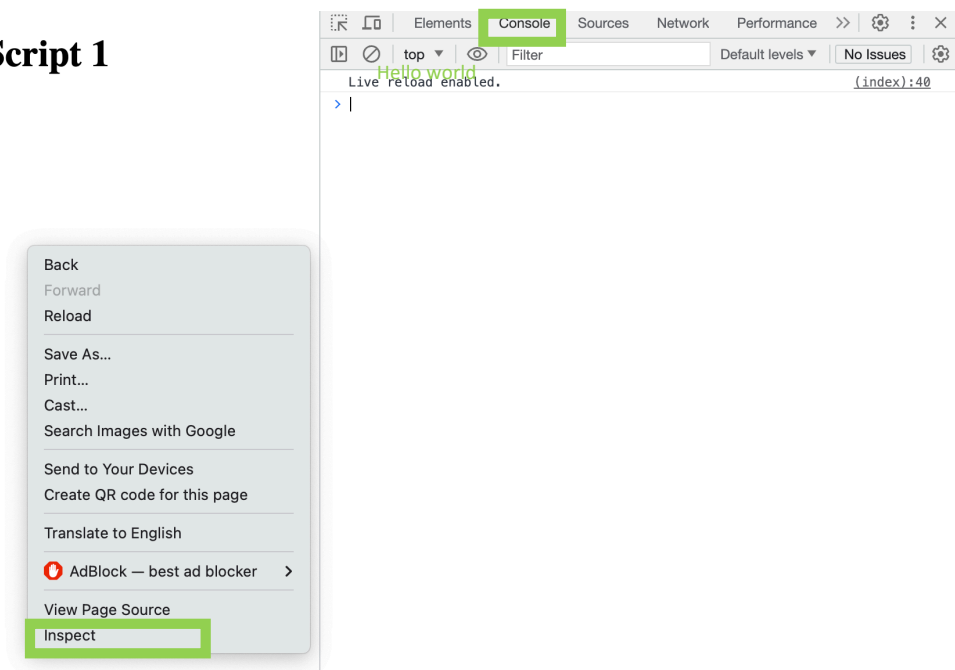
script.js
1 //Din JavaScript kode her
```

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 – Input og Løkker

Forms – input fra brukeren

Oppgave 1 – Kan du øvelsesskjøre

Lag en side hvor en bruker kan finne ut om de kan øvelsesskjøre. Siden skal ha følgende felter:

- Alder – input type = «number»
- Har trafikalt – input type = «checkbox»
- Sjekk – button type = «button»

Skriv inn din alder: Har trafikalt: ☒

Du kan øvelsesskjøre

For å øvelsesskjøre 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.

Skriv inn din alder: Har trafikalt: ☐

Du kan ikke øvelsesskjøre

Oppgave 2 - Kontaktskjema

Lag en side som tar imot henvendelser fra en bruker. Siden skal ha følgende felter:

- Navn - input
- Epost - input
- Tema - input
- Melding - input
- Send inn – knapp
- Avbryt – knapp

Når brukeren trykker på «Send inn»-knappen så skal verdiene i alle feltene lagres i et objekt som heter «henvendelse»

BONUS – Legg på litt CSS så det blir et fint kontaktskjema.

Oppgave 3 – Enkel logg in

Lag en side som tar inn brukernavn og passord. Dersom passordet er «passord» eller «1234» skal man få en melding om at det er et dårlig passord.

UTFORDRING – Gjør denne oppgaven med kun JavaScript. Altså, alle HTML-elementer skal genereres av JavaScript-kode.

UTFORDRING – Quiz

Lag en flervalgsquiz om et valgfritt tema. Her kan du bruke input type radiobutton for å registrere svar.

Løkker og iterasjoner

Oppgave 4

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 5

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.

Hint! `parseInt(streng)` gjør om strenger til tall

Oppgave 6

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

Hint! For in

Oppgave 7

Bruk løkker til å lage en passordgeneratoren fra forrige kurs!

Bruk `prompt()` eller `input` til å hente et ord 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 8

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:

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

UTFORDRING – 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 for-of-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.

UTFORDRING – Pokemon-nettside

Kopier Pokemon-objektet fra [dette dokumentet \(www.krt.im/POKEMON\)](http://www.krt.im/POKEMON) 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 og lister

Oppgave 1 – Kom i gang med funksjoner

Skriv en funksjon som heter **printMelding**. Funksjonen skal skrive ut teksten “Hello World!” til konsollen.

Endre funksjonen til å sende inn **navn** som argument. Da skal meldingen i konsollen se slik ut «Hallo Madeleine»

Oppgave 2 - Terning

Skriv en funksjon som heter **kastTerning**. Funksjonen skal ta inn **antallSider** som parameter, og returnere et tilfeldig tall for en slik terning.

En vanlig terning har 6 sider og returnerer tall mellom 1-6. Da ser funksjonskallet slik ut **kastTerning(6)**

Tips! Bruk de innebygde funksjonene `Math.random()` og `Math.floor(number)`.

- `Math.random()` er en funksjon som returnerer et tilfeldig desimaltall fra 0 til, men ikke med, 1. `Math.random()*6`, gir oss et tall fra og med null til, men ikke med 6.
- `Math.floor(number)` er en funksjon som runder ned et tall. For eksempel vil `Math.floor(3.444)` returnere 3.

Oppgave 3 – Adjektivhistorie

Skriv en funksjon som heter **genererHistorie**. Funksjonen skal ta inn ett navn og 4 adjektiv som parametere, og returnere følgende historie til konsollen:

Det var en gang en <ADJEKTIV1> IM-elev som het <NAVN>. I det <ADJEKTIV2> JavaScript-kurset lærte <NAVN> om variabler og funksjoner sammen med <ADJEKTIV3> klassekamerater. Læreren i kurset er veldig <ADJEKTIV4>, noe som gjør kursdagene ekstra spennende.

Oppgave 4 - Celsius til Fahrenheit

Lag en liten nettside som konverterer fra Celsius til Fahrenheit. Formler du trenger:

$\text{fahrenheit} = (\text{celsius} * 1.8) + 32$ og $\text{celsius} = (\text{fahrenheit} - 32) * 0.56$

Du skal lage følgende funksjoner

- `celsiusToFahrenheit(tempInC)` – tar inn temperatur i C og returnerer F
- `fahrenheitToCelsius(tempInF)` – tar inn temperatur i F og returnerer i C
- `handleCalculate()` – eventhandler for knappen «Konverter temperatur». Denne funksjonen må sjekke om det er trykket av for Fahrenheit eller Celsius, og deretter kalle riktig funksjon.

Temperatur

Fahrenheit ☐

Celsius ☐

Konvertert temperatur vises her

UTFORDRING – Arrow functions

Gjør om alle funksjonene i oppgave 1-4 du har skrevet til *arrow functions*.

Lister og map

Oppgave 5 – Kom i gang med lister

- Lag en liste med fruktene eple, pære og banan.
Bruk en for-løkke for å skrive ut hver frukt til konsoll.
- Legg til kiwi og fjern eple.
- Lag en ny liste til for bær: 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 6 - Palindromer

Skriv en funksjon som heter **erPalindrom**. Funksjonen skal ta inn streng som parameter og returnerer true eller false avhengig av om strengen er et palindrom eller ikke. Et palindrom er en streng som leses likt fra begge retninger.

Eksempel: erPalindrom("level") skal gi true. erPalindrom("hei") skal returnere false.

Oppgave 7 - High-score liste

Lag en side som tar inn og presenterer en liste med high scores. Brukeren skal kunne skrive inn en score og et navn. Når en ny high score legges til i lista skal den sorteres slik at den med høyeste score kommer øverst. Lag egne variabelnavn, funksjoner og passende design.

UTFORDRING - Søkefelt

Lag en side som presenterer følgende liste med helter. Lag et søkefelt hvor brukeren skal kunne søke etter helter.

```
let heroes = [ "Luke Skywalker", "Tony Stark", "Steve Rogers", "Diana Prince",  
  "Harry Potter", "Frodo Baggins", "Indiana Jones", "Mario", "Zelda", "Lara Croft",  
  "Kratos", "Nathan Drake", "Geralt of Rivia"];
```

Steg 0 – Presenter listen i HTML.

Steg 1 – Lag input-feltet og en søk-knapp

Steg 2 – Skriv en funksjon **updateHeroesList("search")** som kjører når du klikker på søk-knappen. Denne funksjonen tar inn søke-ordet som parameter, og oppdaterer listen i HTML-dokumentet.

BONUS! For hver bokstav som skrives inn skal listen med helter oppdateres

Oppgave 9 – Kom i gang med map()

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

Eksempel:

Med listen `[2, 4, 6]`, skal kodesnutten returnere `[4, 8, 12]`.

Oppgave 10 – Pokemon-lister

Kopier Pokemon-objektet fra [dette dokumentet \(www.krt.im/POKEMON\)](http://www.krt.im/POKEMON) eller lag ditt eget ved hjelp av en AI.

Bruk `map()` til å lage nye lister med følgende info:

- Alle unike typer
- Alle pokemon med health points over 60
- En sorter liste basert på attack points

UTFORDRING - 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 alle 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

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`

Eksempeldata:

```
const aksjeListe = [  
  {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},  
  {navn: 'EcoPower', prisPerAksje: 60, sektor: 'energi', antallAksjerTilgjengelig: 1500},  
  {navn: 'AutoDrive', prisPerAksje: 120, sektor: 'transport', antallAksjerTilgjengelig: 2500},  
  {navn: 'FinWealth', prisPerAksje: 90, sektor: 'finans', antallAksjerTilgjengelig: 4000},  
  {navn: 'InnoTech', prisPerAksje: 85, sektor: 'teknologi', antallAksjerTilgjengelig: 1800},  
  {navn: 'MediCare', prisPerAksje: 140, sektor: 'helse', antallAksjerTilgjengelig: 4500},  
  {navn: 'TeachSmart', prisPerAksje: 35, sektor: 'utdanning', antallAksjerTilgjengelig: 300},  
  {navn: 'TasteGood', prisPerAksje: 220, sektor: 'mat', antallAksjerTilgjengelig: 2800}  
];
```