

Javascript-oppgåver

Programmering. Meir oppgåver blir lagt til fortløpande.

Sjå òg "[Oppsummering og eigenvurdering](#)" for ein gjennomgang av det viktigaste du må kunne etter kvart kapittel (kvar periode, tema el.l.).

Kapittel 1 - HTML og CSS

1. Løys oppgåvene frå boka, og gjennomfør eksempla. Sjå i tillegg oppgåver i presentasjonen om HTML og CSS.
2. Du må forstå kva som skal ligge i HTML, kva som skal ligge i CSS og kva som skal ligge i Javascript. Forsøk å alltid separere desse i 3 (eller fleire) filer.
3. Ha eit eller fleire standardutgangspunkt for vidare arbeid, type "mal" for design og oppsett av nettsider. Tanken er at me hovudsakleg fokuserer på Javascript, og sparar oss for ein del tid for kvar oppgåve me løyser. Altså: kopier og lim inn HTML og CSS for kvar nye oppgåve du løyser.
4. Dersom du bruker Visual Studio Code bør du jobbe slik dei foreslår, med ei mappe for kvar oppgåve du løyser. Jfr. "Open folder". Det blir fort surr og rot for mange når dei opnar ei overordna mappe som inneheld "alt".

Kapittel 2 - Introduksjon

1. Løys oppgåvene frå boka, og gjennomfør eksempla. **Eksperimenter med den enkle koden du ser!**
2. Lag fylgjande oppsummering på nettsida di. NB: Dynamisk. Det vil sei at tala for til dømes bredde, høgde og nettlesarversjon skal endre seg basert på enheten du sit på. Dette kan gjerast på meir eller mindre effektive måtar. Vurder ulike alternativ.

Eigenskapar til "window" (fane, vindu)

Me kan hente ut diverse eigenskapar til fana og nettlesaren som er open. Fana du har open er 1440 piksler brei, og 830 piksler høg. Nettsida som er open er `file:///C:/Users/JoBj%C3%B8rnar/Google%20Drive/programmering/it2/kap02-javascript-intro/intro-javascript.html`. Du bruker nettlesaren Netscape, versjon 5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/53.0.2785.143 Safari/537.36.

Legg til fleire eigenskapar du kan hente ut om du ynskjer.

3. Få til å endre innhald på ei nettside med allerede etablert innhald, jfr. boka og forelesningsnotat. Bytt ut til dømes ei overskrift og innhaldet i ein paragraf, samt eit bilete. Korleis refererer du til eit spesifikt element? `getElement...` `innerHTML...`
4. Bytte bilete: I tillegg til å endre biletet vha. kode skal du endre andre attributter, som `title` og `width`. Kjem du på fleire av desse? Set `width` lik fana sin bredde. Kan du ta utgangspunkt i fana sin bredde og trekke frå ein viss størrelse (tal)?

5. Kan du be brukaren om input til oppgåvene over? Altså, kan du til dømes be brukaren om å spesifisere ein tekst og deretter bruke denne når du byter ut overskrifta (eller paragrafen sitt innhald)?
6. Les inn to tal frå brukaren. Summer desse, og skriv ut resultatet. Sjå eksempelkode som er lagt ut for hjelp.
7. Få eksempla over til å fungere med ein enkel knapp. Sjå presentasjonen for hjelp til korleis me best gjer dette.

Kapittel 3: Variablar

1. Løys oppgåvene frå boka, og gjennomfør eksempla.
2. Hva er feil eller dårlig med variabelnavnene under? Gi en kort forklaring og gi dem til bedre variabelnavn.
 - a. `var Et tall = 123`
 - b. `var 3variabelen = 3`
 - c. `var Variabel1 = "teksten skal hit"`
 - d. `var detførstetallet = "1"`
 - e. `var %AvTallet = 15`
 - f. `var SummenAvRegnestykketEr = 16`
3. Skriv eit program som skriver "5 ganger 10 er 50" til konsoll ved å bruke variablar for tala 5, 10 og 50.
4. Skriv et program som bruker variablene under (erstatt variabelnavnene med mer passende navn) til å skrive forslag til e-post adresser til konsoll. Be gjerne om input fra brukeren.
`var1 = fornavn`
`var2 = etternavn`
`var3 = domene`
Kriterier:
 - Epostadressene skal bestå av fornavn og etternavn, separert med punktum, etterfulgt av @ og et valgfritt domene, for eksempel "test.testesen@bedrift.no".
 - Epostadressene som skrives ut skal være uten mellomrom.
 - Variabelen som inneholder "domene" skal ikke inneholde "@".
 - Epostadressen skal lagres i en egen variabel, og denne nye variabelen skal printes ut
5.
 - a. Hvorfor blir ikke regnestykket under regnet ut når `console.log`-setningen kjøres?
`console.log("4 + 6");`
 - b. Hvorfor blir to forskjellige tall printet når koden under kjøres?
`var a = 3.4;`
`var b = 20.5;`
`var total = a * b;`
`console.log(total);`
`console.log(parselInt(total));`

- c. Hva er innholdet i variabelen "poeng" etter at denne koden kjøres?
 var poeng = 0;
 poeng = poeng + 1;
 poeng++;
 poeng += 1;
- 6.
- a. Hva er feil med koden i "Kodebit 1" under? Rett den opp så den kjører.
 gate = 'Kongens gate'
 husnr = 432
 oppgang = b
 adresse = gate + husnr + oppgang
 print(adresse)
- b. Skriv adressen ut på følgende to måter ved å bruke variabler: "Adressen er Kongens gate 432b" og "Gaten er Kongens Gate, husnummeret er 432, oppgang b"
7. Kva betyr det å sette eit erlikteikn i koden din? Til dømes var **pris = pris + 39;**
8. Wales har ein by ved namn **Llanfairpwllgwyngyllgogerychwyrndrobwyllyllantysiliogogoch**. På New Zealand er det ein ås som har blitt gitt det lange navnet **Taumatawhakatangiangaohauotamateapokaiwhenuakitanatahu**. Lag eit program som printer ut antallet bokstaver i kvart namn. Hint: String har ein funksjon som heiter `length`. Beregn deretter differansen mellom desse to. Få til fylgjande formatering i HTML-dokumentet ditt, og eksperimentar med eigne "vriar" på denne:

Geografitemen

Ein by i Wales heiter **Llanfairpwllgwyngyllgogerychwyrndrobwyllyllantysiliogogoch** og ein ås i New Zealand heiter **Taumatawhakatangiangaohauotamateapokaiwhenuakitanatahu**. Byen i Wales består av 59 teikn, medan åsen i NZ består av 57. Differansen mellom desse er 2.

9. La brukaren skrive inn to eigenvalte (stads)namn og berekn deretter dei same data du gjorde over. Treng du gjere store endringar i programmet ditt? Forsøk gjerne å bruke tekstfelt og ein knapp som i skjermbiletet under ([for starthjelp kan du laste ned min startversjon](#)). Eventuelt **prompt**. Pass i tillegg på at du unngår minusteikn i utskrifta dersom den første staden har færre antall teikn enn den andre.
 (Bonusoppgåve: Handter brukarfeil, som til dømes manglande input. Dette tek me opp i neste kapittel).

Stadsnamn, med brukarinput

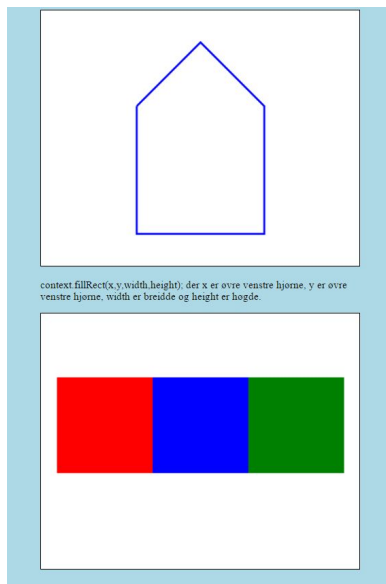
Ulvik Granvin Differanse i antall teikn

Ein stad heiter **Ulvik**, og ein annan heiter **Granvin**. Den første staden består av 5 teikn, medan den andre består av 7. Differansen mellom desse to er 2.

10. **Enkel kalkulator:** Les inn to tal frå brukaren og rekn ut resultatet av desse.

Utviding: Kan du og spesifisere rekneoperasjonen, dvs. $+/-/*$, til dømes? Kva alternative måtar kan du gjere dette på?

11. Array: La brukaren skrive inn X antall element (tekst, tal eller liknande) - til dømes via prompt - som skal plasserast i ein array. Ver bevisst på nummerering. Kan du la brukaren få bestemme kor mange verdier som skal lesast inn òg?
12. Viktig oppgåve frå boka: 5. side 69. **Tilfeldig bilete skal visast kvar gong du opnar nettsida.** (Hugs òg korleis du tidlegare har brukt knappar for å endre eit bilete. Endre koden slik at du får vist eit tilfeldig bilete kvar gong du trykker på knappen.)
13. Tilpass og utvid oppgåvene om teikning i canvas. Eit konkret tips er å lage eit dokument som samlar fleire av oppgåvene frå side 72, til dømes slik:



Legg spesielt merke til oppgåvene som ber deg lese inn input frå brukaren, som fargar (som i biletet over, der brukaren har skrive inn red, blue og green) og koordinatar (eksempel side 72), og teiknar basert på desse data.

14. **Avansert oppgåve:** Ta utgangspunkt i oppgåva over. Ein avansert oppgåvetype legg opp til at du skal vise korleis noko er ved å bruke eksempel og forklare i koden. [Løysningsforslag](#).
 - a. I dette tilfellet skal du vise at du må “caste” datatype når du les inn frå tekstfelt, sjølv om du bruker input type “number”.
 - i. Bruk to tekstfelt og ein knapp, igjen - gjerne frå oppgåva over.
 - ii. Lagre variablar lest inn frå tekstfelt både med og utan casting.
 - iii. Rekn saman variablane både med og utan casting - og vis resultata.
 - iv. Samanlikn ein av variablane (eller begge) både med og utan casting, til dømes om det eine talet er større enn eller lik 5 (bruk \geq), og om det er akkurat likt 5 (bruk $===$). Bruk funksjonar dersom du kan.
15. Få eksempla over til å fungere med ein enkel knapp. Sjå tidlegare presentasjon for hjelp til korleis me best gjer dette.

Kapittel 4: If-setningar og løkker

1. Løys oppgåvene frå boka, og gjennomfør eksempla.
2. (Repetisjon, feilsøking og kontrollering av programflyt) Bruk `console.log("Utskrift...")`; som eit alternativ for å sjå utdata. Dette kjem i konsollen, same plass du mellom anna ser etter feilmeldingar. Spør om du ikkje kjenner til dette.

3. Gjer det **likegyldig om personen skriv inn fornavnet sitt med store eller små bokstavar** i det første eksempelet, side 75, eller quizprogrammet på side 77. Du skal bruke både logiske operatorar og metodar til String-datatypen (sjå kapittel 2 og W3Schools; toUpperCase eller toLowerCase).
4. Utvid eksempelet på side 81, der du legg inn else òg. Til dømes på første sjekk skal det stå "Du fikk ikke to like tall." osv.
5. **NB!** Brukaren skal kunne variere mellom to ulike bileter som blir vist fram berre ved å bruke ein knapp. Pseudokode:
 - a. Programmet startar opp med eit bilete som blir vist fram, samt ein knapp.
 - b. Dersom brukaren trykker på knappen så endrer biletet seg til eit anna bilete.
 - c. Dersom brukaren trykker på knappen igjen så endrer biletet seg til der originale.
 - d. Slik held programmet fram; kvar gong du trykker på knappen hoppar du mellom dei to bileta.
 - e. NB: Det går òg an å kutte ut knappen og berre la brukaren trykke på biletet. Du kan registrere "onclick" på eit bilete på akkurat same måte som med knappen. Du treng heller ikkje fleire img-taggar, du byter berre ut src. ID til img blir det same gjennom heile programmet.
6. **NB!** Lag ein enkel app som kan hjelpe folk med å få vite kva setifikat dei kan ta. Du ber om alderen og presenterer deretter bileter og tekst som fortel kva dei kan ta sertifikat på (meir avansert om du vil:
<https://lovdata.no/forskrift/2004-01-19-298/§3-1>):
 - a. Yngre enn 16. Ingenting, brukaren er for ung.
 - b. 16 år: moped
 - c. 18 år: bil
 - d. 21 år: lastebil
 - e. 24 år: buss
7. Hent ut tidspunktet for maskina du sit på i eksempelet på side 76 (slik som me har gjort tidlegare, ved å bruke Date-objektet). Vis vha. if-else ein generell helsing for ulike intervall for dagen, til dømes når klokka er mellom 6-9 så får du "God morgon", 9-12 så får du "God formiddag" osv.
8. **Lag eit enkelt terningspel.** Ta utgangspunkt i 2 terningar, eller fleire. Når sida lastast på nytt, eller når ein trykker på ein knapp, skal terningane få kvar sin verdi mellom 1-6. Skriv ut kva desse er. Deretter kan du til dømes sjekke om alle er like, om dei totalt overstig ein viss sum osv. Gje deg sjølv nokre utfordringar her. Du kan få hjelp frå eksempelet i boka på side 82, men forsøk å skriv dette sjølv.

Kast terningar

Resultat: 1, 3 og 1

Desse terningane er ikkje like kvarandre.

Summen av terningane overstig ikkje 15, den er berre 5.

9. **Utvid terningspelet.** Bruk ein knapp og enkle funksjonar for å kaste og teste resultatet. Sjå [løysningsforslag](#) (tidleg versjon med forbedringspotensiale) for hjelp, men som vanleg bør du gjere mest mogleg sjølv.

10. **Simuler så mange terningkast som brukaren ynskjer** å gjennomføre ved å bruke løkker. Ta utgangspunkt i førre oppgåve. Eksempelvis: Legg til ein knapp og eit tekstfelt slik at brukaren kan skrive inn kor mange terningkast han eller ho ynskjer å gjennomføre, for så å trykke på knappen. Tel opp kor mange gonger ein får tre like. Legg til meir funksjonalitet etter behov.
11. Lag ei side som viser fram eit bilete, med ein knapp under. Når brukaren trykker på knappen bytes biletet ut med eit anna. Trykker brukaren ein gong til går me tilbake til det originale biletet. Slik skal det halde fram, skift mellom dei to statusane for kvar gong du trykker på knappen. Hint: If-else og boolean. [Løysningsforslag](#).
12. Du skal lage deler av ein app for ungar. Nødvendige bileter skal du laste ned sjølv. Appen (nettsida) opnar med at eit bilete av eit egg vises fram, med forklarande tekst til brukaren.



Funksjonalitet vidare, når brukaren trykker på egget:

- Klikk nr. 1: Egget får ei lita sprekke tvers over.
- Klikk nr. 2: Større sprekke.
- Klikk nr. 3: Egget opnar seg endå meir, og du ser deler av ein kylling innanfor egget.
- Klikk nr. 4: Den øvste delen av egget ligg på bakken/golvet, der resten av egget står med kyllingen oppi.
- Klikk nr. 5: Brukaren får beskjed om at ein ikkje kan trykke på egget meir, og at ein er ferdig.

Presenter innhaldet på en oversiktlig og god måte. Bileta skal vere tilpassa på ein fornuftig måte, med tanke på til dømes format, størrelse og oppløysning.

13. Du skal lage ein løysning for ein nettstad som tilbyr nedlasting av den spennande programvaren "X". Dei har versjonar for Windows (både x64 og x32) og MacOS. Lag ei enkel landingsside ("startside") som kort informerer om programmet og viktigast: tilbyr for nedlasting den versjonen som passar for operativsystemet. Det vil sei at du må avgjere kva OS brukaren bruker og ta val basert på dette ved hjelp av din kode.

NB: Det er ikkje eit krav at du gjer noko anna enn å endre teksten på knappen til det

riktige, til dømes “Last ned til Windows (x64)” eller “Last ned til MacOS”. Sjølve nedlastinga blir ikkje vurdert direkte i denne omgang.

14. Insekt:

Oppgåvetekst: Insektssamleren (for og while)

Oppgåvetekst: En entomolog skal skrive et program som lagrer informasjonen om lengden på larver fra en insektsart han skal forske på. Han trenger 100 målinger. Han vil legge inn denne informasjonen daglig, som en rekke med tall separert med komma, og hver dag få beskjed om hvor mange målinger som mangler før han har nok. Når han har nok målinger skal han få beskjed om det, listen med målinger skal skrives ut til konsoll og programmet avsluttes.

Hjelp entomologen å skrive programmet.

Tips: Her kan man generere en liste med tilfeldige tall (random-biblioteket) av ulike lengder for å teste at programmet fungerer.

Antall registreringar som gjenstår:

Du har 2 igjen å registrere til du har 100.

Du har 1 igjen å registrere til du har 100.

Du har 0 igjen å registrere til du har 100.

Du har no registrert 100 lengder, og er ferdig. Her er lista med verdier: 5, 7, 9, 8, 5, 4, 6, 8, 5, 6, 4, 6, 7, 7, 10, 3, 6, 10, 4, 3, 3, 2, 2, 5, 4, 3, 5, 4, 8, 9, 5, 2, 8, 6, 2, 6, 2, 6, 9, 8, 6, 5, 2, 3, 5, 8, 2, 5, 3, 5, 8, 4, 9, 4, 10, 3, 8, 4, 3, 9, 10, 6, 3, 8, 10, 2, 10, 5, 5, 8, 7, 6, 3, 6, 9, 3, 9, 3, 3, 8, 9, 8, 8, 3, 7, 10, 2, 7, 2, 3, 5, 2, 7, 4, 2, 9, 6, 56, 45, 34.

Gjennomsnittslengden av 100 stk. larver er 6.81.

[Løysningsforslag.](#)

15. Lag eit program for å bestille billetter til tog/buss/kva som helst. Du skal lage eit flytskjema for korleis du handterer dette. Ta gjerne tida på deg sjølv når du løyser denne oppgåva.

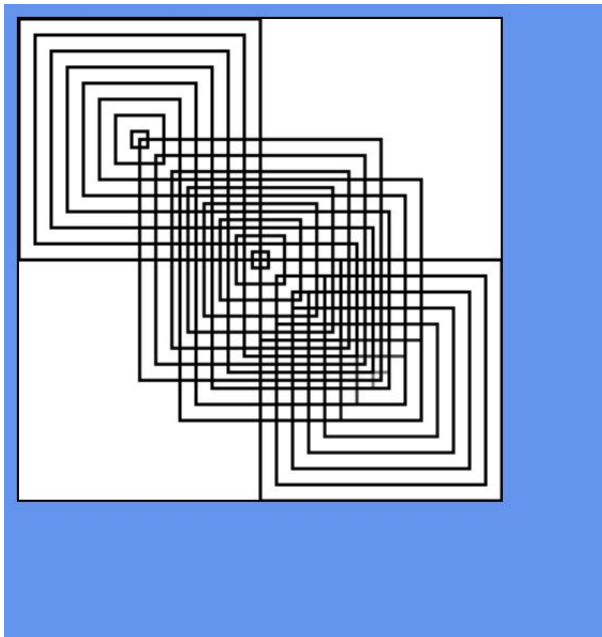


Kjøp

Du skal lese inn navn til personen som bestiller og antall billetter (bruk **prompt** eller tekstfelt og knapp). Dersom personen bestiller mellom 1-5 får han eller ho ingen rabatt, 6-10 gjer 20%, 11-15 gjer 30% og 16 eller meir gjer 40%. Andre verdier skal gje ein beskjed om at noko har gått gale (med tilhøyrande feilmelding). Det skal kome ein oppsummering av ordren, i same gata som “Du har bestilt X antall billetter til ein pris på X kroner, noko som utgjør ein totalpris på X kroner”. Legg til liknande form for

informasjon om eventuelle rabattar. Skriv ein hyggeleg melding som avslutning ("Takk for besøket, Jo Bjørnar Hausnes, kom snart igjen."). Utvid og tilpass etter behov. Etter at den grunnleggjande løysninga er slik du ynskjer legg du programmet inn i eit meir "heilhetlig" og tiltalande design. Dette er ei typisk oppgåve på prøvar og eksamen, så bruk litt tid på å vere grundig.

16. Ta utgangspunkt i eksempelet på side 88, "Telle fra null til 10". La brukaren bestemme kor langt eit skal telle. Fiks på det at resultatet kjem på ei lang linje, bruk linjeskift `
` eller andre teknikkar.
17. Ta utgangspunkt i eksempelet over og la brukaren bestemme kor store hopp ein skal ta. For ordens skuld, over er hoppa på 1 om gongen.
18. Ta utgangspunkt i "Tegne i canvas med en while-løkke" på side 89, og legg til fleire teikningar slik at du kan lage "flotte" mønster. Legg samtidig til ein bakgrunnsfarge for canvas og body, slik at du får eit liknande resultat som meg:



Kapittel 5: Funksjonar

1. Løys oppgåvene frå boka, og gjennomfør eksempla.
2. Opprett eit par knappar som gjer ulike innstillingar på "style" for til dømes paragrafar, bileter, lenker, overskrifter og liknande. Forsøk òg å slå av og på visninga av ulike typar innhald (til dømes tekst og bileter). Sjå side 98 i boka, samt http://www.w3schools.com/js/js_html_dom_css.asp for meir hjelp.
3. Ta utgangspunkt i programmet me laga i kapittel 2, som kunne [vise klokka](#). Denne vil til dømes vise at klokka er 9.31.9 (timar, minutt, sekund). Legg til funksjonalitet slik at det i dette tilfellet skal vise 09.31.09, men det skal ikkje stå 010.031.09. [Løysningsforslag for å få deg i gang med timar](#) (getHours), der du må lage resten sjølv. Merk at det finst andre teknikkar enn i dette løysningsforslaget.
4. Lag eit program som lar brukaren trykke på ein knapp. Når det skjer skal du køyre to funksjonar; ein som skriv ut eit tilfeldig tal mellom 1-20, og ein som gjer ein utrekning med to ulike tilfeldige tal (i første omgang mellom 1-20 her òg). Du skal altså bruke 3

ulike tilfeldige tal (dersom ikkje tilfeldighetene tilseier at det blir nokre like). Kvar av desse funksjonane skal bruke ein og same generelle funksjon som returnerer eit tilfeldig tal basert på krava over. [Løysningsforslag](#).

Funksjonar: Tilfeldige tal

Nytt tall

Tilfeldig tal: 8.

Tilfeldig utrekning: 1 pluss 9 er lik 10.

5. Utvid eksempelet over til å fortsatt gje eit tilfeldig tal mellom 1-20 i første del, og mellom 1-100 i utrekninga.
6. Utvid eksempelet på side 113 til å få ein ball (foto, bilete, grafikk) til å flytte seg rundt basert på kvar du trykker. Dersom du til dømes vel ein fotball som det grafiske elementet skal der du trykker med muspeikaren plassere midtpunktet til ballen (ikkje øverst i venstre hjørne). Kan du òg få bekreftelsar på om ballen er plassert ein viss stad (til dømes om ballen ligg i mål)?
7. Lag ein funksjon som returnerer ein tilfeldig fargeverdi. Dette kan du til dømes bestemme at skal vere type:
 - red, blue, green osv.
 - RGB-referansar
 - hexadesimale kodarSlå opp på W3Schools korleis du refererer til fargar i dei ulike “systema”. Kva er lettest å generere tilfeldig synes du?
8. **Anbefalt oppgåve:** Opprett eit canvas-element som reagerer på fylgjande måte kvar gong du trykker inne i det:
 - a. Teiknar opp ein firkanta boks der du trykte, og skriv ut x- og y-koordinatane (console eller i HTML, alt etter kva du ynskjer).
 - b. Fyll- og omrissfarge skal vere tilfeldig kvar gong, til dømes mellom raud, grønn og blå.
 - c. Problem med at teikningen blir feil plass i forhold til mouseX og mouseY når du til dømes midtstiller designet ditt? Sjå:
<https://stackoverflow.com/questions/17130395/real-mouse-position-in-canvas> og
<https://developer.mozilla.org/en-US/docs/Web/API/Element/getBoundingClientRect> eller unngå midtstilling av canvas.
 - d. Legg til moglegheit for å teikne sirklar o.l.
 - e. Integrer din løysning på neste oppgåve (oppg. 9) som eit eige val, type “tilfeldige formar og fargar”.
 - f. Lat brukaren velgje farge sjølv, eller få tilfeldige fargar.
 - g. [Løysningsforslag](#) og [animasjon.js](#) dersom du vil.
Sjå òg [korleis du kan teikne med “frihand”](#).
9. Utvid førre eksempel ved å bruke ei løkke som fyller inn 50 ulike firkantar med tilfeldig plassering, farge og størrelse (innan eit visst intervall). [Løysningsforslag](#), der det ikkje er med ein løysning på ulik størrelse. Lag det sjølv.
10. Lag eit “JS drum kit” (avspelar av ulike samples). NB: Egnar seg godt som heimeoppgåve.

- a. Startdokument, HTML:

<https://drive.google.com/file/d/0B-FkOPa5sF50QUo3cDJWOXIFMTQ/view?usp=sharing>

Startdokument, CSS:

<https://drive.google.com/file/d/0B-FkOPa5sF50bk5lb1NkMlI0WjA/view?usp=sharing>

Lydar:

<https://drive.google.com/file/d/0B-FkOPa5sF50LWZmbGc5Q2owd1U/view?usp=sharing>

- b. Du kan følge videoen for korleis dette blir gjort [her](#).

11. **Funksjonar og tabellar. Anbefalt oppgåve.** Relevant oppgåve som sett mange gonger i ulike variantar på eksamenar og prøvar. Det blir etterkvart meir avansert, og det er fare for at det blir uoversikteleg dersom du ikkje har god orden og struktur på koden din. Tenk derfor òg på namngjeving og kommentarar. Eg viser sentrale delar av løysninga etterkvart, samt at du får eit løysningsforslag utlevert til slutt.

- a. I korte trekk skal du begynne med å fylle ein tabell med tilfeldige tal. Om det er 20 tal eller 20 000 er likegyldig, du skal lage eit generelt program. Funksjonar skal “tilpasse” seg ynskja til brukaren, til dømes kor mange tilfeldige tal. Bruk funksjonar alle stader det er hensiktsmessig. Her må du naturleg nok bruke ei løkke òg.
- b. Lag ein funksjon som skriv ut ein tilfeldig array sitt innhald grafisk. Få først til å teikne opp ein vanleg tabell (studer strukturen), få deretter dette til å fungere i ein meir dynamisk situasjon. Bruk ei løkke.



Tal
8
7
4

- c. Lag ein funksjon som summerer alle tala i tabellen. Ta som input ein tabell.
function f_summerTallTabell(tabInn) ?
- d. Lag ein funksjon som lar deg leite etter eit spesifikt tal i tabellen, tel kor mange det finnes av dette og viser fram resultatet (eksempelvis “Det er 2 stk. av talet 3 i din tabell”). Eit forslag er å bruke ein funksjon som tek som input ein tabell og kva tal ein ynskjer å leite etter. **function f_tellTalet(tabInn, tallInn) ?**
- e. **Avansert:** Lag ein funksjon som tel kor mange det er av kvart tal i tabellen. Pass på at det ikkje er fleire enn ei utskrift av informasjon om kvart tal. Utnytt funksjonen du laga for å telle kor mange det er av eit spesifikt tal i **del b**.
function f_tellAlleTala(tabInn) ?
- f. **Avansert** (me har om dette i seinare kapittel): Få til ein visualisering av nokre av data du presenterer, til dømes histogram.
- g. Legg til meir funksjonalitet som du synes er interessant. ... Fleire oppgåver kjem.
- h. [Løysningsforslag](#)

12. Lag eit program som kan kryptere og dekryptere tekst. Eksempelet me går gjennom i timane baserer seg på “Cæsar”-kryptering (alfabetforskyvning). Dersom brukaren

skriv inn 'a' og vel ein forskyvning på 2, vil ein få den krypterte bokstaven 'c'.
Igjen, denne går me gjennom i fellesskap. Men, forsøk deg gjerne på eiga hand.
[Løysingsforslag](#).

Kapittel 6: Interaktivitet og brukergrensesnitt

1. Løys oppgåvene frå boka, og gjennomfør eksempla.
2. Forbetre oppgåve 10 frå kapittel 4 i dette oppgåvedokumentet; "billettbestilling".



Kjøp

Lag eit brukargrensesnitt som består av dei tekstboksane og knappane du treng (navn, antall billetter m.m). I tillegg legg du til ei moglegheit for å velge utreisedato. Meir avanserte oppsett og eksempel finn du seinare i kapittel 6, merk at det ikkje er nødvendig å bruke sjekklister, avkryssingsboksar og liknande i første omgang. Set deg som mål å lage den mest brukarvennlege løysinga. Legg til fleire krav dersom du ynskjer ein større utfordring.

3. Lag designet for "Kvitter", den norske Twitter-kлона som har stengt ute Amerikanske presidentar. Krav:
 - Autofokus på tekstfeltet ein kan skrive i ved oppstart, slik at brukaren ikkje må "klikke" seg inn i dette.
 - Endre størrelsen på tekstfeltet sitt innhald så lenge dette er aktivt. Større font-størrelse, til dømes, slik at folk verkeleg ser kva dei skriv inn. Tilbake til vanleg størrelse dersom tekstfeltet mistar fokus (brukaren trykker ein annan stad).
 - Bestem deg for ei grense for antall teikn. Til dømes 30. Fortløpande, altså medan ein skriv inn teksten sin, skal ein få oppdateringar på kor mange teikn av dei tilgjengelege ein har brukt. Når ein har "brukt opp" skal det kome tydeleg fram for brukaren.
 - Før ein får lov til å sende inn Kvitter-meldinga si må ein oppfylle to punkt:
 - i. fyller inn svaret på ei matematikkoppgåve i eit input-felt. Det skal berre vere mogleg å skrive inn tal i dette svarfeltet.
 - ii. fyller inn dagens dato frå eit input-felt. Det skal vere ein datovelger.
4. Utvid eksempelet på side 131 i boka der du skal opprette ei dataliste med alternativ, og få på plass label frå side 132 òg. Bruk det du har lært om å få noko til å skje når du har skrive noko i tekstfeltet. Bonus: Få tekstfeltet til å få fokus på ein betre måte når brukaren gjer det aktivt (trykker inne i boksen) = onfocus. Eit eksempel er å gjere tekstboksen og fontstørrelsen større. Gjerne med animasjon. Leik òg med dei andre

innstillingane/vala.

5. Utvid eksempelet på side 133 i boka der du skal lage ein enkel kalkulator.

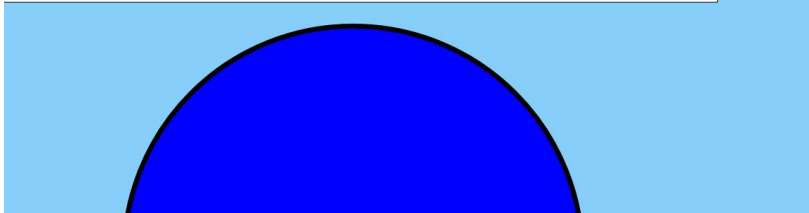
- La brukaren velgje kva type matematisk operasjon han eller ho vil gjere. Du kan til dømes bruke knappar for å oppnå dette. Legg til subtraksjon, ganging og deling, med fleire etter behov. NB: Du treng ikkje fleire enn 2 funksjonar for å handtere dette.
 - Fokuser ellers på brukarvennlegheit. Gjer det vanskeleg å gjere feil for brukaren. Gje tydeleg instruksar.
 - [Løysingsforslag](#).
6. Utvid eksempla som omhandlar **radiobuttons** og **checkbox**. Finn på eit bruksområde sjølv, og gjennomfør det. Utvid gjerne bestillingsskjemaet for din løysning av oppgåve 6.2. “Skjema for bestilling av en flytur” på side 149 er eit anna godt døme på dette, og er viktig at alle gjennomfører på ein grundig måte.
7. Gjer oppgåva “Ekstrastoff” på side 149-152. Denne er eksamensrelevant. Alle detaljar er ikkje med, og det er i så måte ein god øving å få på plass bruken av komponentar og ei meir “heilthetlig” oppgåve.
8. **“Tilfeldig puls”**: Utvid eksempelet på side 121 der du skal endre på størrelsen til den raude sirkelen (til ein tilfeldig størrelse mellom 0-600). Lag eit meir omfattande brukergrensesnitt, og få all funksjonaliteten du ser i skjermbiletet under på plass. Her skal du bruke “timer-events” for å løyse dei fleste av utfordringane som kjem. [Sjå W3Schools for hjelp til å kome i gang](#). Dersom du treng hjelp til å kome i gang kan du ta utgangspunkt i [denne fila](#), som ikkje er ferdig, men har eit “skall” for det meste. Eg legg ut løysningsforslag etter behov.
- Du skal kunne starte og stoppe den repeterande animasjonen som lar deg gå frå ein størrelse til ein annan. Kula vil typisk “pulsere” mellom å vere liten og stor.
 - Du skal kunne auke og minke farten på størrelsesendringen.
 - “Endre farge!” skal gje ein tilfeldig farge der du hoppar tilfeldig mellom 4-5 fargar som du sjølv vel ut.
 - Legg i tillegg til moglegheit for å endre fargen på kula til det brukaren ynskjer frå ein fargevelgar (jfr. side 136).
 - Bruk console.log for å kontrollere data under køyring av programmet (dette er òg ein aktuell teknikk for å feilsøke og kontrollere programflyten din). NB: Hugs tipsa du har fått tidlegare; legg til litt og litt funksjonalitet og få det til å fungere før du går vidare til neste punkt. Dersom å endre fargen er det du ser

på som lettast så er det det du begynner med.

Tilfeldig puls

Dette programmet endrer på radiusen med ein verdi mellom 1-500 piksler og animerer dette heilt til du vel å stoppe det med å trykke på knappen under.

Velg farge:

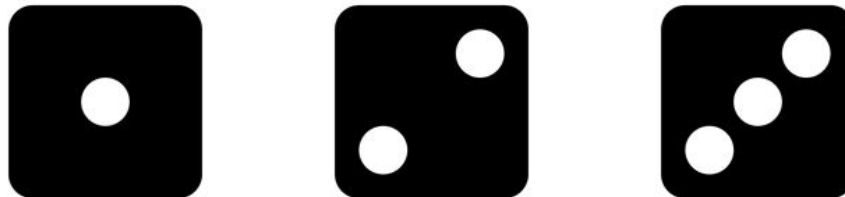


```
Farten etter at t_senkFa  
køyr er no 1000.  
Inne i funksjonen  
f_startAnimasjon().  
Farten inne i funksjonen  
f_animasjon() er no 1000  
Farten etter at f_senkFa  
køyr er no 1500.  
Farten etter at f_senkFa  
køyr er no 2000.  
Inne i funksjonen  
f_startAnimasjon().  
Farten inne i funksjonen  
f_animasjon() er no 2000  
Inne i funksjonen  
f_startAnimasjon().  
Farten inne i funksjonen  
f_animasjon() er no 2000  
Farten etter at f_aukFar  
køyr er no 1500.  
Farten etter at f_aukFar
```

9. **Lag eit terningspel.** Eksempelen kan gjerne utvidast til til dømes Yatzy. Prinsippet er dessutan heilt likt poker, eller andre liknande spel.

Terningspel

Forsøk å få tre like terningar. Du har tre forsøk, og kan halde terningar dersom du vil.



- Begynn typisk med å få 3 stk. tilfeldige tal mellom 1-6 etter at du trykker på knappen. Dersom du vil begynne enklare så tek du i første omgang utgangspunkt i 2 terningar, eller kanskje berre 1..?
- Basert på kva tal som dukkar opp får du bileter av terningar til å dukke opp, som over.
- Kontroller om terningane er like, og gje i så fall ein beskjed til brukaren om det. Du kan legge til fleire slike alternativ etter behov, til dømes kan det vere eit mål å få tre tal etter kvarandre, eller samla sum av terningane over ein viss sum. Gjer dette så avansert du vil i første omgang, og hugs at du kan legge til meir avansert funksjonalitet etterkvart.
- La det vere mogleg å halde ein terning, og å ikkje halde lenger. Det vil sei at den ikkje skal trillast når ein trykker på knappen dersom ein tidlegare har haldt den. Visualiser denne haldinga ved å til dømes ha ei kantlinje rundt grafikken.
- Legg til funksjonalitet som gjer at ein ikkje har fleire forsøk enn til dømes 3 før ein får eit endeleg resultat.

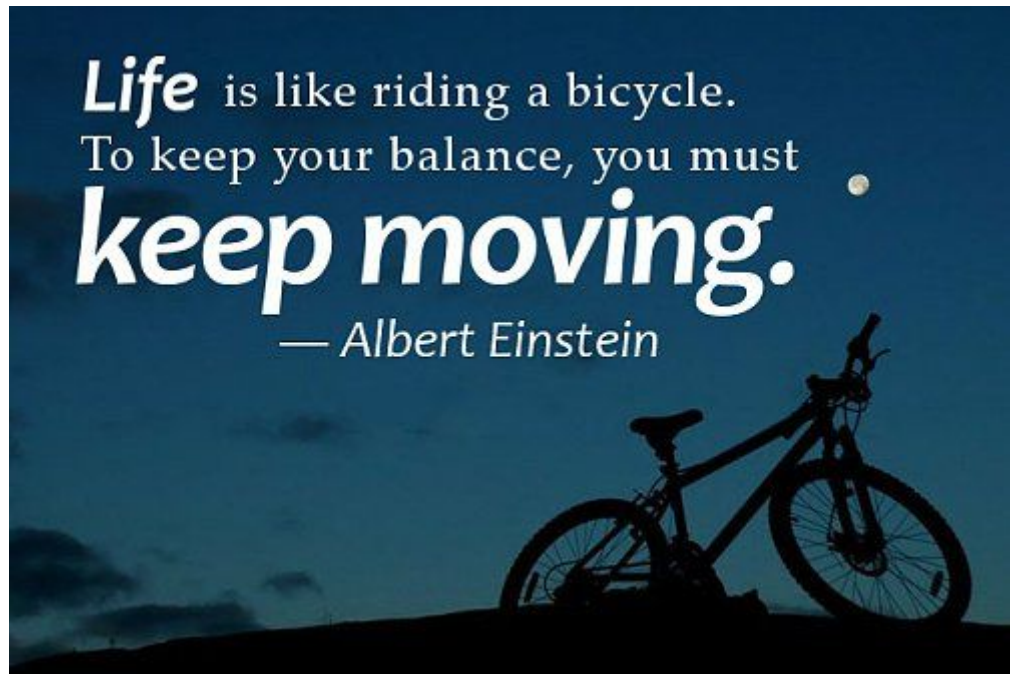
- f. Gje generelt gode tilbakemeldingar i brukergrensesnittet for kva som skjer til ei kvar tid.
 - g. Legg til meir funksjonalitet etter eigne ynskjer. Til dømes:
 - i. Det skal vera mogleg å bestemme kva ein ynskjer å oppnå, til dømes ved å velje mellom like terningar, over ein viss sum, terningar som ligg etter kvarandre (eksempelvis 1-2-3) osv. Dette kan til dømes ligge i ein nedtrykksmeny (select-element), eller liknande.
 - ii. Trillande terningar, med tanke på visuell effekt.
 - iii. Lydeffektar.
10. **“Google Keep”-klone**, eller liknande: Du skal lage ein tjeneste som lar brukaren registrere data (til dømes noko ein skal hugse, ein “post it”) via eit brukargrensesnitt. Bruk form og andre teknikkar frå kapittel 6 (m.m.). Ha gjerne som fokus at det skal fungere på ulike typar skjermar/oppløysningar. NB: Skjermbiletet under er ein prototype, set deg høgare mål.

- a. Kvar gong ein registrerer noko skal det bli framvist i (til dømes) ein ny boks. I eksempelet over er det brukt div-boksar og float. Avansert oppgåve: Forskjellige fargar? Tilpass størrelse til tekst?
- b. I tillegg til å framvise data skal du òg lagre på ein hensiktsmessig måte, til dømes i ein array. Eg har brukt “push” for kvart nye element.
- c. Avansert: Bruk [localStorage](#) og [JSON.stringify](#) og [JSON.parse](#) (du finn eit **godt eksempel i boka vår på side 353**) for å lagre data “permanent”, og gjer det mogleg å hente desse data inn igjen når ein kjem tilbake til nettsida.
- d. Dei som vil kan bruke denne fila som eit utgangspunkt, sjølv om eg for dei fleste som vanleg anbefalar å gjere alt på eiga hand: [Startversjon](#)

Kapittel 7: Multimedier - tekst, bilde, video og lyd

1. Løys oppgåvene frå boka, og gjennomfør eksempla. Legg merke til at det er eit større fokus på teori.
2. Lag eit enkelt program som lar brukaren endre størrelsen på eit bilete vha. skalering. Sjå gjerne s. 166 for litt hjelp. Legg òg til funksjonalitet for å zoome inn på bileter, jfr. s. 168-169.
3. Repetisjon av multimedia.
 - a. Lag eit HTML-dokument der du presenterer eit slitsomt og svulstig “life motto”, som er eit typisk pompøst og “voldsomt” sitat saman med eit (eller fleire) bilete eller film. Dette skal la brukaren få frysningar på ryggen, tenke wow, oi og fytti katta (og dele det vidare til vener som ikkje bryr seg på Facebook). Du skal bruke teknikkar i kodinga, ikkje berre gjer det i

Photoshop.



- b. Det faglege fokuset går på å få til fylgjande teknikkar:
 - i. Tenk gjennom teorien rundt skrifttypar og begrunn vala dine. Bruk skrifttypar som fungerer med tanke på det du skal sei. Ha "fallbacks". Bruk webfonter (i det minste frå Google), og last opp til serveren din (it1.iktv3.no) dersom du ynskjer å forsøke det.
 - ii. Ha fleire paragrafar og overskrifter med ulik utsjånad.
 - iii. Finn nokre bileter som du tilpassar i Photoshop (eller liknande). Ha fokus på innstillingar kapittel 7 snakkar om (og som me hadde om tidlegare dette skuleåret). Kan du begrunne alle val og tenke gjennom kva endringar som skal til dersom du får spesifikke krav om å endre på filstørrelse, til dømes? Tenk filformat, oppløysning, fargedybde m.m. Korleis kan du vise din kompetanse innan desse måla utan at du skal skrive om dei? Ver òg forberedt på å snakke om det på munnleg-praktisk.
 - iv. Sjå på teknikkane for å få tekst som ligger over eit bilete ved å bruke position-attributtet på side 166-167. Bruk dette på deler av ditt arbeid.
 - v. Legg til moglegheit for å skalere bilder i deler av arbeidet ditt. Sjå til dømes "Zoome inn på et bilde" på side 168-169.
 - vi. Ta sjølv utgangspunkt i kapittel 7 og legg til meir innhald der du bruker teknikkane som blir beskrive. Finn gjerne på andre ting òg!
4. Lag ein interaktiv variant (forenkla) av oppgåva over. Forslag: la brukaren sjølv velgje frå nokre bileter, der ein etterpå kan skrive inn visdomsord som blir lagt over.
5. Bruk torrent-teknologien i nettlesaren. Det blir vist eksempel i timen. Mozilla har ein guide på dette.

Kapittel 8: Kolleksjonar

1. Løys oppgåvene frå boka, og gjennomfør eksempla.
2. Frå boka: **Lag biletgalleriet på side 200-201.**
 - a. Legg i tillegg til ei moglegheit for å bla andre vegen.
 - b. La det òg vere ei moglegheit at galleriet “blar framover sjølv”, som ein slider. Timer-funksjon kan vere eit hint her.
3. **Array:** Opprett ein array med ein del **ulike** testdata (tal, tekst, boolean, array m.m.).
 - a. Hent ut **tilfeldige** data frå arrayen (eit og eit element) når du trykker på ein knapp, og vis dette fram på ønska måte (forslag: tekstboks/paragraf, console.log el.l.).
 - b. Lagre data som du henta ut tilfeldig i førre punkt i ein **ny array** fortløpande.
 - c. Til slutt skal det vere mogleg å hente ut (vise fram) data frå den nye arrayen, der du **VALGFRITT** kan legge til meir avansert funksjonalitet etter behov (skriv til ein tabell, kor mange er det av kvar data osv.). NB: [Har du gjort noko liknande før?](#)

Hente ut tilfeldige data frå tabell

Trykk på knappen for å hente ein tilfeldig verdi ut frå tabellen (Array). Du får då ei utskrift av dette.

Her er data du henta ut frå tabellen:
Jo Bjørnar,Hausnes,4x3=12,green,3,green,3,3,green,Jo Bjørnar,Hausnes,4x3=12,pink,red,7

Innhald
Jo Bjørnar,Hausnes,4x3=12
green
3
green
3
3
green
Jo Bjørnar,Hausnes,4x3=12
pink
red
7

4. **Array:** Opprett to stk. array. Same kva innhald. [Starthjelp til dei som treng det.](#)
 - a. La brukaren velge mellom desse to, til dømes ved å bruke ein nedtrykksmeny (select). Kva andre komponentar kunne du brukt?
 - b. La brukaren trykke på ein knapp for å vise fram data frå den valgte tabellen. Er ingenting valgt skal ein få feilmelding.
 - i. Framsyninga av arrayen skal vere i form av ein HTML-tabell.
 - ii. Ekstraoppgåve: Forbetre framsyninga ved å vise fram indeksnummeret ved sida av innhaldet.
 - iii. Ekstraoppgåve, utfordring: Dersom det er fleire enn 10 element i tabellen skal den delast opp i “blokker” på 10 om gongen.
 - c. Lag deretter moglegheiter til å gjere endringar i denne tabellen. eksempelvis ved å spørre kva celle vil du endre og kva skal den endrast til. Du bør kunne handtere eventuelle feil (til dømes at brukaren vel ein indeks utanfor området

til arrayen.

Array: Hente ut og endre data

Instruksjonar til brukaren.

Velg data å vise fram

Velg array ▼

Vis data frå array

Her kjem data du hentar frå tabellen.

Endre data i array

Indeks (NB: Begynner på 0):

Velg data å vise fram

Velg array ▼

Vis data frå array

Her kjem data du hentar frå tabellen.

Du må velge ein array å jobbe med.

Du forsøker å endre data som ikkje er innan rammene for arrayen. Hugs, indekseringa begynner på 0.

Endre data i array

Indeks (NB: Begynner på 0):

- d. Legg til annan funksjonalitet som å slette og legge til informasjon.
5. **Array, set, map eller liknande:** Lagre årstall med tilhørende mengder trær som er skjært ned et år i Norge. Bruk en passende kolleksjon for å lagre årstall og antall trær. Gjør ulike operasjoner med data over. Eksempelvis:
- La brukeren regne ut differansen mellom hvor mange trær som er skjært ned i et årstal til et annet, eksempelvis fra 1970 (med 10 000 trær) og 1983 (med 12 000 trær).
 - Prosentvis endring fra et år til et annet.
 - Hvor mange år mengden trær har vært over 8000.
 - Finn på noe selv.
6. **Array, set, map eller liknande:** Lag eit program som inneheld ei rekke med 4 ulike unike knappar. "Innhaldet" kan vere fargar, tal, bokstavar, lydar eller liknande. Når du startar programmet/spelet får du sjå rekka visualisert og skal forsøke å gjenta den. Bommar du får du feilmelding, klarar du heile rekka vinn du spelet. Dersom du

ynskjer å utvide spelet kan du legge til fleire knappar, lengre rekker osv. ([Video](#))



7. **Set:** Utvid programmet som lar deg legge til datoar du er opptatt på.
- a. For kvar gong du legg til ein ny dato skal dette visast fram i ein oversikt under (eg har brukt ei unummerert liste), til dømes slik:

11.10.2017

Du er nå opptatt denne dagen også

Visning av når du er opptatt:

- 2016-05-17
- 2017-03-09
- 2017-03-10
- 2017-03-11
- 2017-03-12
- 2017-04-01
- 2017-10-11

- b. Legg til ei moglegheit til å slette at du er opptatt ein dag, til dømes ved hjelp av å trykke på ein knapp etter at du har valt ein dato. For å forbetre brukaropplevinga må du kanskje gjere om på korleis du legg til avtalar òg? Eksempelvis vha. ein knapp her òg.
- c. Ha ein visning av kor mange dagar ein er satt opp som opptatt.
- Du har totalt 4 opptatte dagar.
- d. Gjer andre tilpassingar og legg til meir funksjonalitet etter behov.

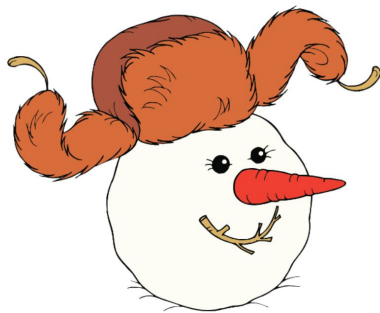
Kapittel 9: Animasjon og programmering

1. Løys oppgåvene frå boka, og gjennomfør eksempla.
2. Gå tilbake til ei tidlegare oppgåve der du har brukt input-felt type tekst/tal for å be om informasjon og gjer det tydeleg kva felt brukaren er "inne i" (som er aktivt). Gjer dette

brukargrensesnittet nødvendigvis betre?



3. Ta utgangspunkt i tidlegare program der du har fått til å spele av ulike lydfiler når du til dømes trykker på ulike knappar på tastaturet. Legg til ulike CSS-stilendringar, animasjonar og liknande for å få ein interaktiv multimedieopplevelse. Tenk blinkande lys og generelle fargeendringar, størrelseendringar osv.
4. La brukaren styre kor mange gonger snøballen skal rulle over skjermen. Kan du la brukaren styre andre innstillingar òg?

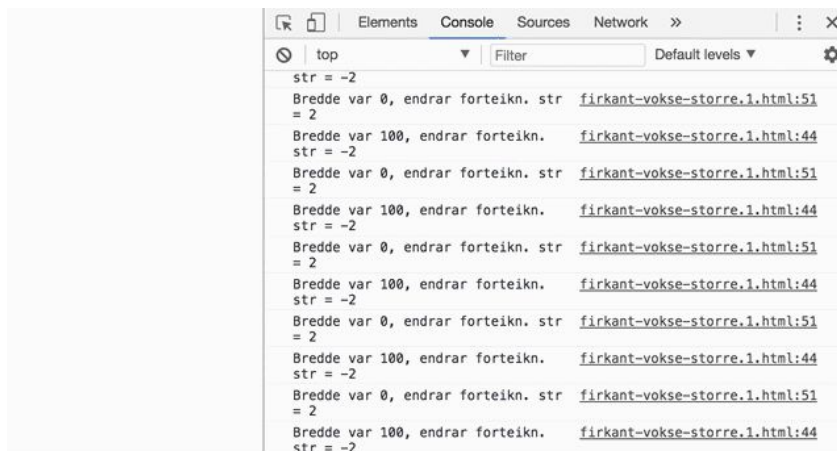


5. Animer ein overgang frå ein sirkel til ein firkant (eller omvendt) når du fører muspeikaren over. [Løysningsforslag](#).
6. Animer ein overgang frå eit bakgrunnselement av typen farge til bakgrunnsbilete, gjerne eit PNG-bilete (eller tilsvarande) med gjennomsiktig bakgrunnsfarge. Dette skal skje når du trykker på eit element, eller fører muspeikaren over. Løysningsforslag, hint (kvit tekst, marker for å vise):
7. Lag eksempelet frå presentasjonen der du tek utgangspunkt i eksempelet på side 260 med ei “vektstang” som roterer litt basert på brukarinput. La brukaren registrere kor mange kaloriar han eller ho får i seg, og basert på dette tippar vektskåla til venstre om du får i deg for mykje, blir liggande vannrett om du et passe, og tippar til høgre dersom du får i deg for mykje.
(Bilete)
8. Fleire kjem.

Kapittel 10: Spillprogrammering

1. Løys oppgåvene frå boka, og gjennomfør eksempla.
2. **Rekursive funksjonar, introduksjon**: Ta utgangspunkt i eksempelet på side 290, der du skal endre på størrelsen på eit div-element. Når elementet har blitt “stort” så kan

du lage ein funksjon som gjer at det blir lite igjen, for så å bli stort igjen osv.



- a. Ekstra utfordring: Lag ein generell funksjon med parameter som til dømes lar deg skifte forteikn for å handtere denne endringa mellom liten og stor.
- b. La brukaren bestemme (til dømes vha. ein knapp) når animasjonen skal snu (bli større/mindre).

3.

Ekstraoppgåver

1. Oppgåvetekst: (Fleire oppgåver: <https://codefights.com/>)

Given a sequence of integers as an array, determine whether it is possible to obtain a strictly increasing sequence by removing no more than one element from the array.

Example

- For sequence = [1, 3, 2, 1], the output should be
- `almostIncreasingSequence(sequence) = false;`
- There is no one element in this array that can be removed in order to get a strictly increasing sequence.
- For sequence = [1, 3, 2], the output should be
- `almostIncreasingSequence(sequence) = true.`
- You can remove 3 from the array to get the strictly increasing sequence [1, 2]. Alternately, you can remove 2 to get the strictly increasing sequence [1, 3].

Input/Output

- **[execution time limit] 4 seconds (js)**
- **[input] array.integer sequence**
- *Guaranteed constraints:*
- $2 \leq \text{sequence.length} \leq 105$,
- $-105 \leq \text{sequence}[i] \leq 105$.

- **[output] boolean**
- Return `true` if it is possible to remove one element from the array in order to get a strictly increasing sequence, otherwise return `false`.

[JavaScript (ES6)] Syntax Tips

```
// Prints help message to the console
// Returns a string
function helloWorld(name) {
  console.log("This prints to the console when you Run Tests");
  return "Hello, " + name;
}
```

[Løysningsforslag frå Simon](#)

2. Meir kjem...