

# Lærerveiledning - JS: Trykkomania

Skrevet av: Lars Klingenberg og Susanne Rynning Seip

Kurs: Web

Tema: Tekstbasert, Nettside, Spill

Fag: Matematikk, Programmering, Kunst og håndverk

Klassetrinn: 1.-4. klasse, 5.-7. klasse, 8.-10. klasse, Videregående skole

## Om oppgaven

Denne oppgaven viser deg hvordan du kan lage et spill med JavaScript og dele det med vennene dine. Spillet kalles *Trykkomania* fordi det handler om å trykke på en ball flest mulig ganger før tiden renner ut.

Oppgaven henter ideer fra utvikling av web-applikasjoner med bibliotek som React (<https://facebook.github.io/react/>) og Mithril (<http://mithril.js.org/>), der elementer i spillet lages som individuelle komponenter. Komponentene ligner på objektorientert programmering, men bruker teknikken "closures" (funksjoner som husker konteksten de ble laget i).

## Oppgaven passer til:

**Fag:** Matematikk, Programmering, Kunst og håndverk, IT1

**Trinn:** 4. trinn - VG3

**Tema:** Javascript, Web, Spill, Variabler, Funksjoner, Closures, Objekter

**Nivå:** Nybegynner

**Tidsbruk:** Dobbeltime eller mer.

## Kompetansemål



**Kunst og håndverk, 7. trinn:** bruke programmering til å skape interaktivitet og visuelle uttrykk

- ☐ **Matematikk, 7. trinn:** representere og bruke brøk, desimaltall og prosent på ulike måter og utforske de matematiske sammenhengene mellom disse representasjonsformene
- ☐ **Matematikk, 8. trinn:** lage og forklare regneuttrykk med tall, variabler og konstanter tilknyttet praktiske situasjoner
- ☐ **Fordypning i matematikk, 10. trinn:** diskutere, planlegge, lage og vurdere spilldesign og egne spill
- ☐ **Programmering, 10. trinn:** bruke flere programmeringsspråk, deriblant minst ett som er tekstbasert
- ☐ **IT1, VG2:** lage og bruke egne og andres funksjoner med og uten parametre og returverdier
- ☐ **IT1, VG2:** lese, strukturere, analysere og kommentere programkode
- ☐ **IT2, VG3:** lage objektorienterte programmer som benytter klasser med metoder

## Forslag til læringsmål

- ☐ Eleven kan bruke enkle matematiske uttryksmåter for å øke eller minke variabler i JavaScript.
- ☐ Eleven kan bruke JavaScript til å tegne en sirkel.
- ☐ Eleven kan plassere et element i på en nettside ved hjelp av koordinater på x- og y-aksen.
- ☐ Eleven kan skrive kommentarer til sin egen kode i JavaScript.

- ☐ Eleven kan bruke variabler, løkker og funksjoner til å manipulere elementer i JavaScript.
- ☐ Eleven kan videreutvikle sitt ferdige produkt ved hjelp av egenprodusert JavaScript-kode.

## Forslag til vurderingskriterier

- ☐ Eleven oppnår middels måloppnåelse ved å fullføre oppgaven.
- ☐ Eleven oppnår høy måloppnåelse ved å videreutvikle egen kode basert på oppgaven.

## Forutsetninger og utstyr

- ☐ **Forutsetninger:** Oppgaven er *kun* javascript, men det lønner seg å ha kjennskap til HTML og CSS.
- ☐ **Utstyr:** Datamaskin med internett.

## Konsepter brukt i oppgaven

- ☐ Variabler (<https://developer.mozilla.org/en-US/docs/Glossary/Variable>)
- ☐ Objekter (<https://developer.mozilla.org/en-US/docs/Glossary/Object>)
- ☐ Funksjoner (<https://developer.mozilla.org/en-US/docs/Glossary/Function>)

- ☐ Funksjoner (<https://developer.mozilla.org/en-US/docs/Glossary/Function>)
- ☐ Closures (<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Closures>), funksjoner som husker konteksten de ble laget i.
- ☐ HTML-elementer (<https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement>) via javascript
- ☐ CSS-stil (<https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement/style>) via javascript
- ☐ `onClick` (<https://developer.mozilla.org/en-US/docs/Web/API/GlobalEventHandlers/onclick>)
- ☐ `setInterval` (<https://developer.mozilla.org/en-US/docs/Web/API/WindowTimers/setInterval>)

## Løsning

Her er en full løsning av oppgaven. ([losning.js](#))

## Fremgangsmåte

Her kommer tips, erfaring og utfordringer til de ulike stegene i den faktiske oppgaven. Klikk her for å se oppgaveteksten. ([trykkomania.html](#))

## Generelt i oppgaven

- ☐ Elevene bør ha god og oversiktlig kode sånn at det er enkelt å finne feil. Dette oppnås ved å kommentere koden, samt bruke inntrykk og mellomrom mellom funksjoner og annen kode.
- ☐ Elevene må passe på at variabelnavn og tegnsetting er riktig.

☐ Elevene må passe på at funksjonen og regning er riktig.

## Steg 2: Lage en ball

- ☐ Elevene kjenner kanskje igjen CSS-elementer når de skal lage funksjonen `Ball()`. Her vises det at HTML og CSS kan programmeres gjennom JavaScript.

## Steg 3: Flytte ballen

- ☐ Elevene kan lure på hvor de skal legge til koden i dette steget, den skal legges til i funksjonen `Ball()` fordi `e1` er en lokal variabel til funksjonen `Ball()`.

## Steg 4: Flytte ballen med en funksjon

- ☐ Elevene kan være forvirret hva `x` og `y` er i denne oppgaven så her er det viktig å poengtere at dette er verdier som blir sendt inn senere i programmet, som vi ser rett før *Steg 5*.

## Steg 5: Velg en tilfeldig plassering

- ☐ Her ser vi at vi kan legge til *strenger* bak tall som er blitt regnet ut: `Math.random() * 100 + '%'`; . Dette kan være svært nyttig for elevene å vite i senere oppgaver.
- ☐ Elever kan lure på hva `Math.random()` er. Og ved å si `Math.random()` så kaller vi på et bibliotek (*Math*), altså en innebygget JavaScript-fil, som inneholder funksjonen `random()`.

## Steg 7: Poeng

- ☐ Elevene må kopiere koden akkurat som den står oppført, hvis ikke blir det fort feil.

- ☐ **Merk:** Det er en liten forskjell i koden gitt i dette steget og koden som det er linket til i fasitene.

`e1.style.bottom` er satt til `50px` (i motsetning til `5px`). Dette er for at den ikke skal ende opp bak reklamen som kan dukke opp på bunnen av editoren.

## Steg 8: Begrense tiden

- ☐ Elevene må kopiere koden akkurat som den står oppført, hvis ikke blir det fort feil.

## Steg 9: Omstarte spillet

- ☐ Elevene må kopiere koden akkurat som den står oppført, hvis ikke blir det fort feil.

### Variasjoner

- ☐ *Vi har dessverre ikke noen variasjoner tilknyttet denne oppgaven enda.*

Lisens: CC BY-SA 4.0 (<http://creativecommons.org/licenses/by-sa/4.0/deed>)