

# Høyskolen Kristiania

## Eksamen

### PGR102 Introduksjon til programmering

Høst 2021

Løses og leveres individuelt eller i gruppe på 2 eller i gruppe på 3

---

#### Merknader:

- Du/dere har kun lov å bruke JavaScript, HTML og CSS. Programmering med de JavaScript-teknikker som er lært i emnet er hovedpoenget med eksamenen. CSS er kun poenggivende i kontekst av JavaScript, for eksempel ved at du bruker «element.style.property» eller involverer JavaScript på annet vis for å legge til CSS.
- Du/dere skal kode i vedlagte HTML-filer
- All kode skal være din/deres egen. Copy paste og avskrift av andres funksjoner/kodeblokker, for eksempel fra nettet eller fra medstudenter, er ulovlig, dvs. regnes som plagiat. Hele gruppen er ansvarlig for at denne regelen følges.
- Tips: les gjennom alle oppgavene før du begynner å kode for å få en oversikt over hvilke oppgaver du synes er vanskeligst og hvilke som er enklest.
- Tips 2: noen ganger kan det være greit å gjøre mindre tester av enkelte teknikker før man går i gang med en oppgave.
- Tips 3: all kode kan gi poeng, selv om man ikke skulle bli ferdig med en oppgave. Hvis man ikke får til noe på en måte så test å gjøre det på en annen måte.
- Zip/rare besvarelsen din/deres før opplasting til WISEFlow
- Karakterform er Godkjent/Ikke Godkjent. Vurderingen settes basert på en helhetsvurdering av hvordan studenten(e) behersker grunnleggende programmering.
- Les sensurveiledningen som ligger integrert i denne oppgaveteksten.

## Oppgave 1. Modern Art Maker (20%)

Her skal du lage en applikasjon hvor en bruker skal kunne lage sine egne kunstverk.

1. Bruker angir bredde, høyde, farge, left og top og «boks» (<div></div>) legges til på nettsiden, dvs. skrives ut til «output-div», evt. på annet vis ved behov.
2. Det skrives ut hvor mange bokser som er lagt til på nettsiden.
3. Legg også til funksjon for å fjerne alle boksene for å kunne starte på nytt.

I denne oppgaven skal du kommentere kodelinjene/kodeblokkene/funksjonene. Du skal forklare hva koden gjør.

Oppgave 1

←

→

↺

🏠

🔒

📄

127.0.0.1:5500/oppgave1.html

☆

🔒

⚙️

▼

☰

# Modern Art Maker

Bredde

10

Høyde

500

Farge

red

Posisjon fra venstre (left)

200

Posisjon fra toppen (top)

50

Legg til

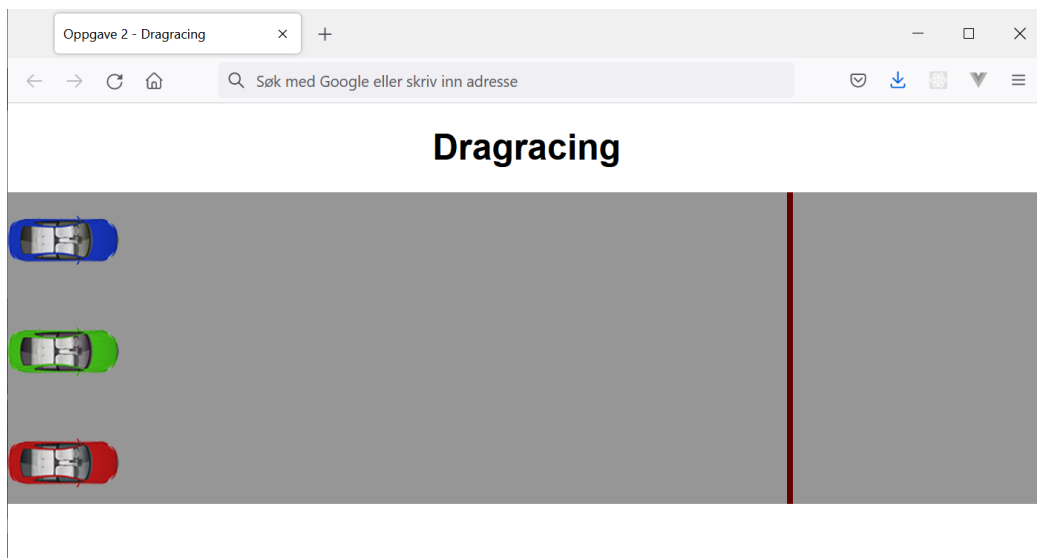
Antall bokser lagt til: 3

## Oppgave 2. Dragracing (25%)

Vedlagt ligger en HTML-fil som heter setInterval-eksempel.html. Les og kjør koden og forstå hvordan den fungerer.

I denne oppgaven skal du simulere et billøp. Lag følgende funksjonalitet (det skal gå fint å utvide med 1 og 1 funksjonalitet):

1. På nettsiden er det 3 biler. Hvert 2de sekund (setInterval brukes for dette) skal hver bil kjøre fremover en tilfeldig strekning mellom 0 og 200 px.
2. Når første bil kommer i mål (700px eller over) skal det skrives ut hvem som vant og alle tre biler skal stoppe å bevege seg.
3. Gjør at billøpet ikke starter før du klikker på en knapp (du må legge til HTML også for dette).



### Oppgave 3. Middelalder eiendomsmegler (25%)

Vedlagt ligger 6 bilder av middelalderbygninger. I denne oppgaven skal du lage en applikasjon for å vise dem frem på nettsiden.

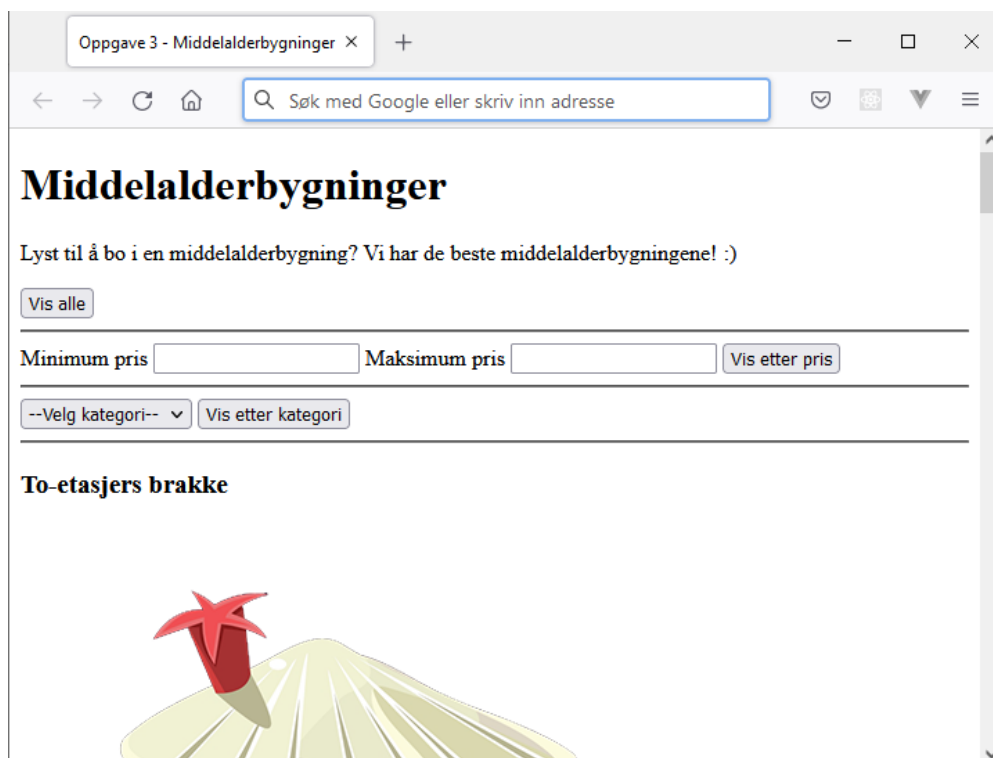
Du skal opprette ett array som inneholder 6 objekter (dvs. 1 objekt per bygning). Hvert bygningsobjekt skal ha følgende informasjon:

- name
- price
- imageName
- category (du kan finne på dette selv, men eksempelvis «Vakttårn», «Brakke» osv.)

Følgende funksjonalitet skal lages her:

1. Vise alle bygninger
2. Vise de bygningene som er innenfor en minimums- og maksimumspris
3. Vise bygninger i en kategori (inkludert å vise antallet bygninger som ble funnet i kategorien)

Hvis brukeren skriver inn ugyldige verdier, eller har glemt å fylle ut/velge noe skal det sies fra til bruker.



## Oppgave 4. Last Mission (30%)

Hvis du jobber alene: velg 1 av de 3 oppgavene nedenfor.

Hvis dere jobber 2 sammen: velg 2 forskjellige oppgaver av de 3 oppgavene nedenfor.

Hvis dere jobber 3 sammen: alle 3 oppgaver skal løses.

Følgende oppgaver er ganske åpne oppgaver hvor man kan bestemme mye selv mht. funksjonalitet.

### Oppgavene å velge mellom:

1. **Minibank-simulator** med minimum følgende funksjonalitet:
  - a. Må taste inn riktig pin-kode for å få lov til å ta ut penger.
  - b. Bruker skal starte med 2000 kroner som saldo. Bruker kan ta ut 500, 1000, 1500 eller 2000 kroner. Bilde av penger (vedlagt) skrives ut til bruker.
  - c. Man skal kunne se visuelt (med bilder) hvor mye penger bruker har, til å begynne med 4 500-lapper, og hvor mye penger bruker tar ut.
2. **Minnespill:** Et spill for å øve opp minnet. Brukeren vises et tilfeldig tall. Etter noen sekunder tas tallet vekk og brukeren må taste det inn. Brukeren får tilbakemelding om det var riktig eller feil, og får så opp neste tall. Man får vite hvor mange riktige og gale man har.
3. **«Lær et språk»-spill:** Et språklæringspill hvor man skal kunne lære opptil 10 ord i et språk. Her kan du være kreativ med tanke på hvordan spillet skal fungere. Du har også lov å legge til egne bilder for denne oppgaven.

### Påminnelse:

- All kode skal være din/deres egen. Copy paste og avskrift av andres koder, for eksempel fra nettet eller fra medstudenter, er ulovlig, dvs. regnes som plagiat. Hele gruppen er ansvarlig for at denne regelen følges.

-- Slutt oppgavesett --

## Sensurveiledning for sensorer (interne og eksterne) og studenter

Vurderingen settes på grunnlag av følgende ting som gjelder for alle oppgavene:

- Variabel og funksjonsnavn
- Ryddighet og kodestruktur
- God og riktig bruk av JavaScript-teknikker (funksjoner, betingelser, løkker, bruk av arrays osv.)
- Antallet på gruppen
- Modularisering og unngåelse av koderepetisjon gjennom bruk av funksjoner
- Omfang og kompleksitet; merk spesielt dette i sammenheng med oppgave 4 som er friere definert enn de andre oppgavene
- **Karakteren settes basert på en helhetsvurdering av hvordan studenten(e) behersker grunnleggende programmeringsteknikker lært i emnet.**