

(GOOGLE TRANSLATE)

Kristiania University

Exam

Introduction to programming

Fall 2021

Notes:

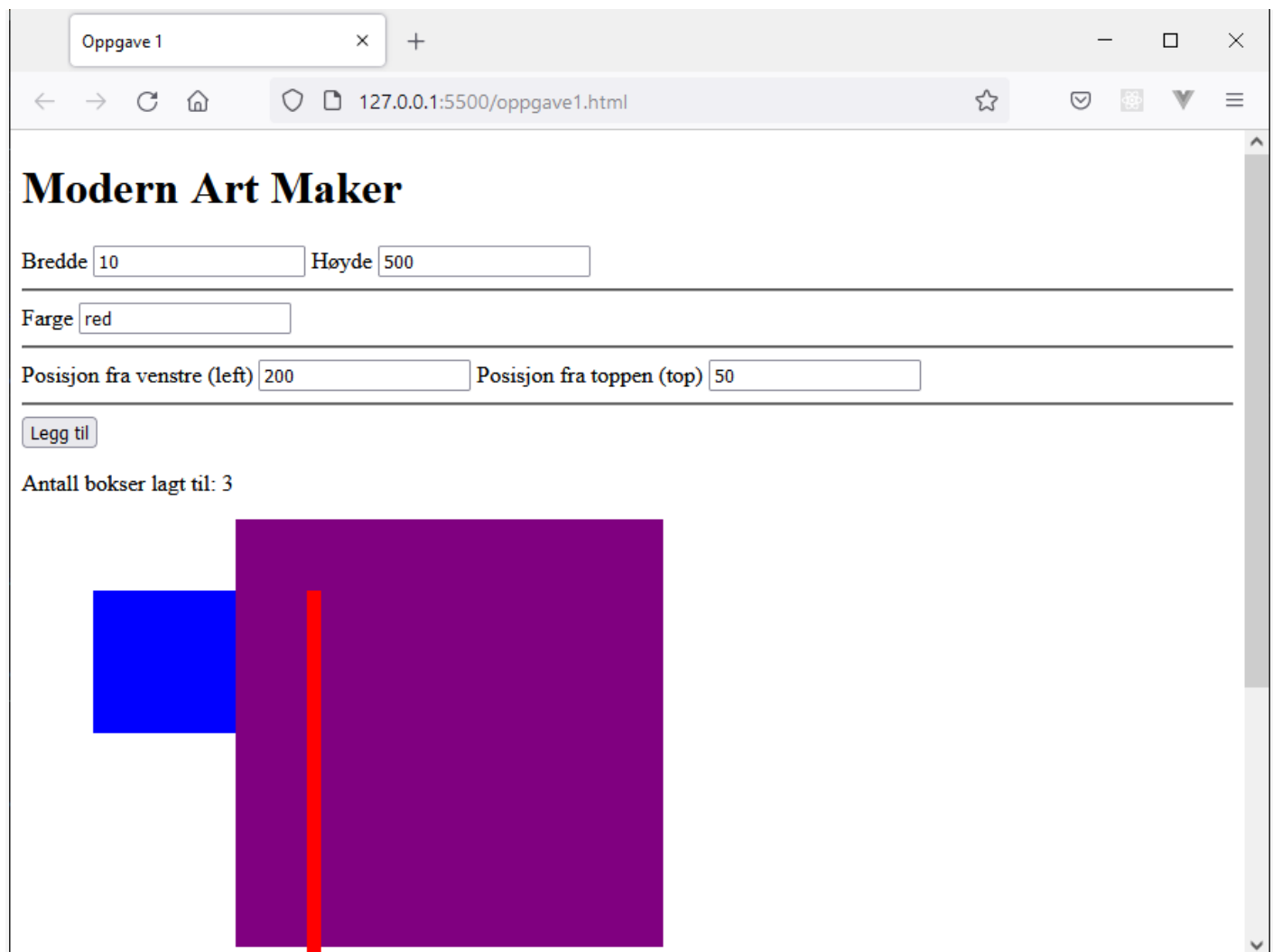
- You/you are only allowed to use JavaScript, HTML and CSS. Programming with the JavaScript techniques learned in the course is the main point of the exam. CSS is only scoring in the context of JavaScript, for example by using "element.style.property" or otherwise involving JavaScript to add CSS.
- You/you must code in the attached HTML files
- All code must be your/their own. Copy paste and copying of other people's functions/code blocks, for example from the internet or from fellow students, is illegal, i.e. considered plagiarism. The entire group is responsible for following this rule.
- Tip: read through all the tasks before you start coding to get an overview of which tasks you think are the most difficult and which are the easiest.
- Tip 2: sometimes it can be good to do smaller tests of certain techniques before starting a task.
- Tip 3: all code can give points, even if you don't finish a task. If you can't get something done in one way, try doing it in another way.
- Zip/rare your/their answer before uploading to WISEFlow
- Grade form is Approved/Not Approved. The assessment is based on an overall assessment of how the student(s) master basic programming.
- Read the censorship guide that is integrated in this assignment text.

Assignment 1. Modern Art Maker (20%)

Here you will create an application where a user will be able to create their own works of art.

1. User specifies width, height, colour, left and top and "box" (<div></div>) is added to the website, i.e. printed to "output-div", possibly in another way if necessary.
2. It prints how many boxes have been added to the website.
3. Also add function to clear all the boxes to be able to start over.

In this assignment, you must comment on the code lines/code blocks/functions. You must explain what the code does.



ts

Task 2. Drag racing (25%)

Attached is an HTML file called setInterval-eksempel.html. Read and run the code and understand how it works.

In this task, you will simulate a car race. Create the following functionality (it should be fine to expand with 1 and 1 functionality):

1. There are 3 cars on the website. Every 2nd second (setInterval is used for this) each car must drive forward a random distance between 0 and 200 px.
2. When the first car reaches the finish line (700px or above) it must be printed who won and all three cars must stop moving.
3. Make the car race not start until you click a button (you need to add HTML for this as well).



Task 3. Medieval estate agent (25%)

Attached are 6 photos of medieval buildings. In this task, you will create an application to display them on the website.

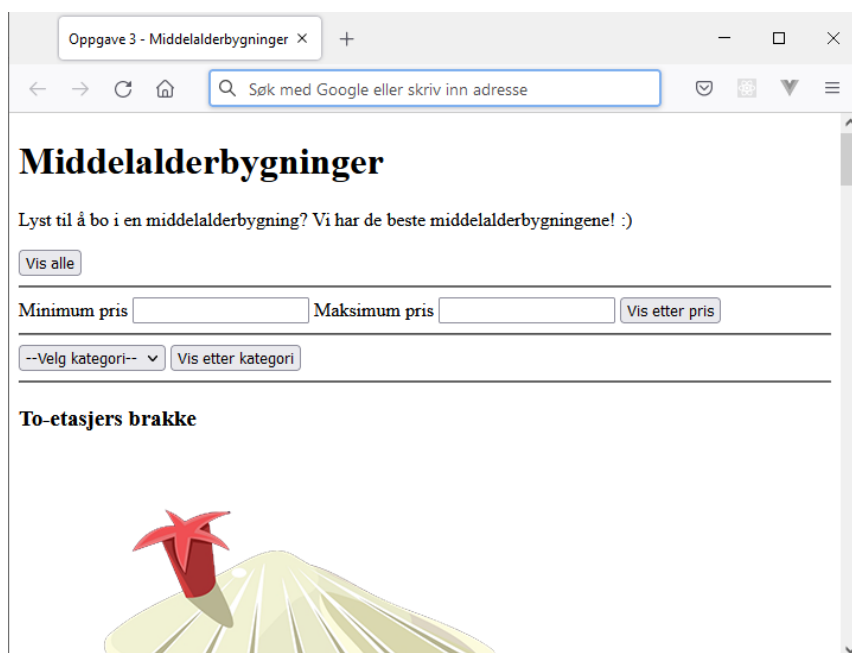
You will create an array containing 6 objects (ie 1 object per building). Each building object must have the following information:

- name
- price
- imageName
- category (you can come up with this yourself, but for example "Watchtower", "Barracks", etc.)

The following functionality must be created here:

1. Show all buildings
2. Show the buildings that are within a minimum and maximum price
3. Displaying buildings in a category (including displaying the number of buildings found in the category)

If the user enters invalid values, or has forgotten to fill in/select something, the user must be notified.



Task 4. Last Mission (30%)

If you work alone: choose 1 of the 3 tasks below.

If you work 2 together: choose 2 different tasks from the 3 tasks below.

If you work 3 together: all 3 tasks must be solved.

The following tasks are fairly open-ended tasks where you can decide a lot yourself regarding functionality.

The tasks to choose from:

1. ATM simulator with at least the following functionality:

a. Must enter the correct pin code to be allowed to withdraw money.

b. The user must start with a balance of NOK 2,000. Users can withdraw NOK 500, 1000, 1500 or 2000. Image of money (attached) is printed for the user.

c. You must be able to see visually (with pictures) how much money the user has, starting with 4,500 notes, and how much money the user withdraws.

2. Memory game: A game to practice your memory. The user is shown a random number. After a few seconds, the number is taken away and the user must enter it. The user receives feedback as to whether it was correct or incorrect, and then receives the next number. You get to know how many right and wrong ones you have.

3. "Learn a language" game: A language learning game where you should be able to learn up to 10 words in a language. Here you can be creative with regard to how the game should work. You are also allowed to add your own photos for this assignment.

Reminder:

- All code must be your/their own. Copying and copying other people's codes, for example from the internet or from fellow students, is illegal, i.e. considered plagiarism. The entire group is responsible for following this rule.

-- 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.**