

Web and App Programming Exam 2023

Rules:

You have until **15:00** to complete this exam. Feel free to upload multiple versions of it, if you want to make sure you get something returned in time. There is a return area in our Moodle for it.

You can use the internet freely, including AI tools like chatGPT and knowledge bases like Stack Overflow. You can also refer to our files created during the lessons freely.

Real time communication with other humans is forbidden. So no chatting or talking with others during the exam.

Monitoring will be done via microphones, so you need to keep your mic open during the exam (your speakers can be muted though so possible sounds from others will not disturb you).

Good luck and do your best!

Task 1: How Do You Do?

Preparations:

Go to the folder called Task_HowDoYouDo.

You will find an HTML-file called howDoYouDo.html which contains some basic HTML tags and a set of empty script tags.

JQuery has NOT been added to this file, but you shouldn't really need it for this task.

Task:

Create a variable called **answer** and set it to contain the string **fine**.

Create a variable called **pleasantry** and set it to contain the string **thank you**.

Create a variable called **together** and add the values of both **answer** and **pleasantry** to it. If you want, you can also add one whitespace character between the two strings.

Make the page pop up a little window on load, that displays the content of the **together** variable.

Remember to save your changes to the file.

Task 2: Color Me Surprised!

Preparations:

Go to the folder called Task_ColorMeSurprised.

You will find an HTML-file called colorMeSurprised.html which contains three divs with a class of "color".

JQuery has NOT been added to that file, but feel free to do so if you prefer using it. You can also just use vanilla JavaScript if you like.

Task:

Modify the code so that when the mouse cursor moves over one of the divs with the class of color, the body of the web page changes its background-color property to the corresponding color (written inside of each div).

Extra point if you are able to take the color value dynamically from either the id or the text (inside of the div) instead of hard coding it.

Remember to save your changes to the file.

Task 3: Map

Preparations:

Go to the folder called Task_Map.

You will find an image of a map and an image of a character.

There is also an HTML-file called map.html which contains both images.

JQuery has already been linked already to that file for your convenience, but you can also use vanilla JavaScript if you prefer.

Task:

Change the map.html file so that when the user clicks on the map where there is a circle, the position of the character is animated to that location. Note that you need to implement something that creates "hotspots" where the circles are in the image.

Remember to save your changes to the file.

Task 4: Lock

Preparations:

Go to the folder called Task_Lock.

You will find two image files: a closed lock and an open lock.

There is also an HTML-file called lock.html.

JQuery has already been linked already to that file for your convenience, but you can also use vanilla JavaScript if you prefer.

Task:

Change the lock.html file so that when the three numbers under the lock are switched to the combination 404, the closed lock image on the page gets replaced with the open lock image.

Remember to save your changes to the file.

Task 5: Objection!

Preparations:

Go to the folder called Task_Objection.

There is an HTML-file called objection.html.

JQuery has already been linked already to that file for your convenience, but you can also use vanilla JavaScript if you prefer.

Task:

The HTML file contains an object representing a movie rental shop and some buttons and one input field.

Implement the following features:

- When the user clicks on "List All Movies", write all the of movies from the object and their details (name, cost, year) on the page.
- When the user clicks on "Sort by Price", write all the of movies from the object and their details (name, cost, year) on the page but order them based on their costs. Can be ascending or descending order.
- When the user clicks on "Sort by Year", write all the of movies from the object and their details (name, cost, year) on the page but order them based on their year. Can be ascending or descending order.
- When the user enters characters that match the name of an existing movie in the input field and clicks on the "Search by Name" button, write the names of the matching movies on the page.

Remember to save your changes to the file.

Task 6: Memeory

Preparations:

Go to the folder called Task_Memeory.

You will find a folder with some images, jQuery and an HTML-file called memeory.html.

jQuery has already been linked already to that file for your convenience, but you can also use vanilla JavaScript if you prefer.

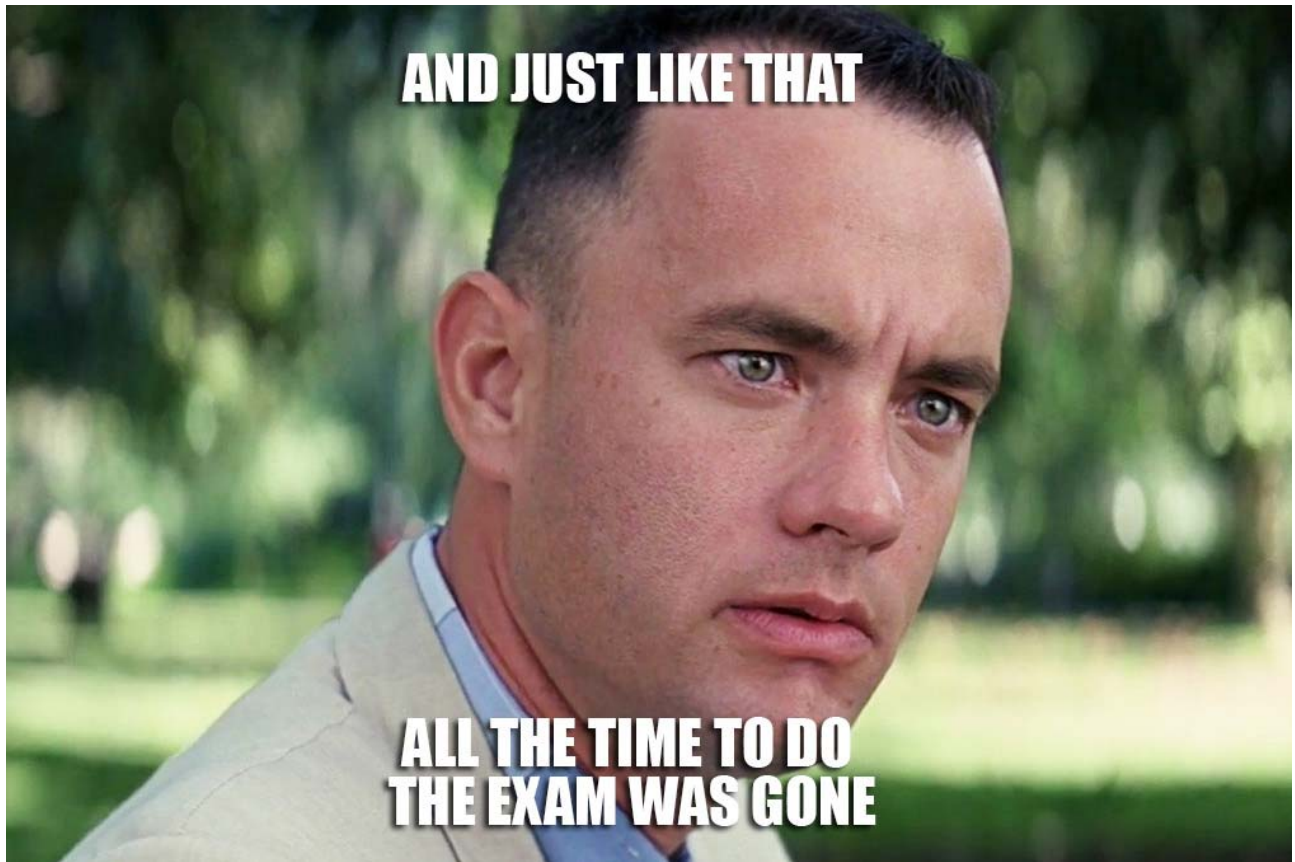
Task:

Inside the HTML-file there is a grid with 12 cells. There is also an array called imagesArray with 12 image filepaths.

Develop this into a memory game in which a pair of each image is attached to the cells and the user needs to click and open two of the same images at the same time to remove those images from the game. When the user has found all the pairs, congratulate them with a pop-up message. Extra point if you are able to make the order of the images random on each load.

Hint: I have written an example function for changing the background image of a cell. You can use it, but there are many many different ways to approach this task! You can create variables to keep track of things, you could use objects with keys for things like if the card is currently open, the name of the card etc.. Try your best with the knowledge that you have! You could also google around for ideas, I'm sure many have created similar memory games in the past.

Remember to save your changes to the file.



At the end of the exam:

When you are ready or when time is about to run out, simply create a .zip archive out of the exam folder and upload it to Moodle.