Structured Code Review

Copy this form, and fill it for each program that you review – give the resulting document to the author.

|  |  |  |  |
| --- | --- | --- | --- |
| Filename: | \_bar\_chart.html\_ | Git hash: | <https://pbettio.github.io/JSExercises/bar_chart.html> |
| Author: | \_\_\_\_\_\_\_\_\_\_\_Patricia\_\_\_\_\_\_ | |  |
| Reviewers: | \_Luciano\_ Patricia | \_\_\_\_Megane | \_\_Kerttu\_\_\_\_\_ |

**Instructions:**

Go through the form in order from top to bottom. Every section has a list of questions. Initial questions are bold and all the way to the left, sub questions are indented with one tab.

If you can answer “yes” to an initial question, that aspect of the code is good, and you can skip the sub questions. If not, go through each sub question. Here we look for problems – a “no” means that there is no problem, and you can go to the next sub question.

When you find a problem, discuss with each other what it is, and suggest how it could be remedied. The author shouldn’t fix the problems during the review, but is allowed to, if it is quick and easy.

# Code structure

*Is the code structured well enough, does it fit well together, is it easy to navigate around it, and follow the flow? Can you glance at it, and understand what it is supposed to do? Is it built in a solid way, or does it seem like it could break at any time?*

## Globals

**Are there as few global variables as possible?**

Are there a lot of const document.querySelectors?

One document.query.selector

**Is everything (possible) put into functions?**

Are there eventlisteners outside of functions (apart from DOMContentLoaded)?

No

Are there initialisation done outside of functions?

No

Are there other code-lines outside of functions?

No

**Is every function called from another function** (or forEach, setTimeout, eventListeners etc.)?

Are functions called from the global space?

No

## Naming

**Are variables and functions named correctly in camelCase?**

**Yes**

**Are variables named in a meaningful way?**

**Yes**  
**Can you tell what a variable might contain, from the name?**

**Yes**

Are there single-letter variables, like a, e, or h? (x, y, or i should be okay)

For the counter = i

Are there misleading variable names, asdf, elmt or other un-understandable names?

No

**Are variables named from the “problem-area” rather than the implementation-area?**

Are variables called: element, div, or similar?

No

**Are functions named clearly with what they do?**

Do some functions do more than one thing?

Do some functions mislead with their name?

**Do the function names indicate whether the function modifies the model or renders the view?**

**Do the function names indicate whether the function returns a value?**

**Are functions named in a consistent pattern?** (e.g. are all functions handling the view, sharing part of their names, like createView, updateView, animateView)

## Functions

**Can you follow the flow through the program?** (draw a call-graph to help you decide)

**Are functions declared in an order that makes the code easy to follow?**

**Do each function do only one thing?**

If not, could some of them be split into multiple functions?

**Is there enough separation of concerns?**

If not, which functions should concern themselves with less?

**Are functions generic enough, do they use parameters or variables, so they can be re-used?**

**Are functions loosely coupled, so they can be called at (almost) any time, from anywhere?**

Are there function dependencies, where one function always needs to be called before another? Can it be avoided?

# Code design

*Is the code designed in a way that makes sense, or does it seem like it just does the job. Can you talk about the code in abstract terms, as different parts with different responsibilities, or is it just a lot of JavaScript lines?*

## Components

**Does the code seem like it was built in components**, e.g. one part for init, one for looping, one for handling data, one for output, etc.?

## Separation of concerns / Model View Controller

**Does the JavaScript keep to itself, and don’t modify the CSS (except with css variables or classes)**

**Does the JavaScript split responsibilities between model, view and controller?**

# Code style

How is the code written, is it easy to read, can you skip “blocks” as if they were black boxes, are there comments, are things named properly, are there weird solutions you don’t understand?

## Comments

**Are there comments in the code?**

Are there parts of it, where comments would be needed / nice?

Is it hard to understand what is going on, without comments?

Do you need comments in addition to function and variable names?

**Are the comments meaningful?**

Are they describing what the code does, as opposed to what it should do?

Are there comments that no longer apply to the code?

Are they written in plain english?

## Code

**Does the code use mostly known methods and patterns?**

Does it use methods you haven’t learned yet, or methods you have found replacements for?

Does it use strange constructs, rather than well known methods?

Does it look like it is in another language?

"use strict"

window.addEventListener("DOMContentLoaded", init);

let HTML = {}

function init(){

    //conversion();

    showHex();

    //HTML.hex = document.querySelector(".hexCode");

}

function conversion(){

r /= 255;

g /= 255;

b /= 255;

let h, s, l;

const min = Math.min(r,g,b);

const max = Math.max(r,g,b);

if( max === min ) {

  h = 0;

} else

if (max === r) {

  h = 60 \* (0 + (g - b) / (max - min) );

} else

if (max === g) {

  h = 60 \* (2 + (b - r) / (max - min) );

} else

if (max === b) {

  h = 60 \* (4 + (r - g) / (max - min) );

}

if (h < 0) {h = h + 360; }

l = (min + max) / 2;

if (max === 0 || min === 1 ) {

  s = 0;

} else {

  s = (max - l) / ( Math.min(l,1-l));

}

// multiply s and l by 100 to get the value in percent, rather than [0,1]

s \*= 100;

l \*= 100;

console.log("hsl(%f,%f%,%f%)", h, s, l); // just for testing

}

function showHex(){

    let hex = document.querySelector('input[type=color]').value;

    console.log(hex)

    let hexElement = document.querySelector(".getColor");

    let hexCode = document.querySelector(".getColor").textContent;

    //document.querySelector(".getColor").addEventListener("input", hex);

    hexElement.oninput = function() {

        console.log(document.querySelector('input[type=color]').value)

        console.log(hexCode)

}

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    Select a color<input type="color" class="getColor" id="selectColor">

    <p class="hexCode">HEX Code:</p>

    <div class="testHex"></div>

    <script src="color\_selector.js"></script>

</body>

</html>