

Code Improvements and Feedback

Improving the Grand Prix experience
for F1 viewers at home

S8 Graduation FHICT

4 Sept 2023 - 16 Jan 2024

By Jordi Franssen

Introduction	3
Feedback on Commits	4
Template Literals	4
Do more checks to prevent errors	5
Define variables for colors	6
General improvements	7
Summary	9
Learning Outcome Clarification	9

Introduction

Halfway through the development phase, I realized that I haven't done any code reviews with other colleagues. I'm fairly confident in my coding skills and, as described in my deliverable about the use of AI, ChatGPT is able to refactor code. Which I sometimes did to get rid of some messy code.

Although ChatGPT can fix most of the code within a single document, it doesn't always pick up things like file structures and global variables. So, therefore I thought it would be a good idea to do a code review by a colleague.

I asked John to have a look at my commits in Gitlab. John made a few comments and we discussed some general good coding habits.

Feedback on Commits

Template Literals

```
text = [  
  {  
    title: `Red Flag - Lap ${lapNumber}`,  
    sentence: 'Red flag in lap ' + lapNumber + '.',  
  },  
  {  
    title: `Race Halted - Red Flag in Lap ${lapNumber}`,  
    sentence: 'Race halted, red flag shown in lap ' +  
lapNumber + '.',  
  },  
  {  
    title: `Race Stoppage - Red Flag in Lap ${lapNumber}`,  
    sentence: 'Lap ' + lapNumber + ' under red flag, race  
stopped.',  
  },  
]
```

Before I started this semester, I didn't know what template literals were. I often used + to make a string containing a variable. However, the `${}` syntax is much more readable and it's also possible to combine it with logical operators.

Do more checks to prevent errors

```
import Base from './Base.svelte';

const liveblogEventReceiverElement =
document.querySelector('.js-event-receiver');

/**
 * Initialize liveblog
 */
const init = () => {
  // eslint-disable-next-line no-new
```

I should do more checks to prevent unexpected errors. In this case, I assume that `liveblogEventReceiverElement` always exists. This might be the case, but just to be sure it's always a good habit to catch an unexpected error.

```
if (!liveblogEventReceiverElement) {
  return;
}
```

Define variables for colors

```
function trackStatusColor(trackStatus) {  
    switch (trackStatus) {  
        case 'AllClear':  
            return '#82CD47';  
        case 'Red':  
            return '#FE0000';  
        default:  
            return '#F4CE14';  
    }  
}  
  
function checkTrackStatusChanged() {  
    if (contentItem.trackstatusChanged) {  
        return trackStatusColor(contentItem.trackStatus);  
    } else {  
        return '#555';  
    }  
}
```

In this case, I should have defined the colors in variables. Just in case the colors need to be changed, it only has to be changed in one place. The other place cannot be forgotten.

General improvements

John discussed his findings with me and we also concluded some general improvements. Overall he didn't find any concerning things with my code, so that's positive I guess.

Overall, I could pay more attention to code structure. I sometimes define variables in multiple places in a file. It would be better to have all variables defined in the same place. This also applies to defining functions.

```
57
58 function googleEmojis(index) {
59   switch (index) {
60     case 0:
61       return '😬';
62
63     case 1:
64       return '😬';
65
66     case 2:
67       return '😬';
68
69     case 3:
70       return '😬';
71
72     case 4:
73       return '😬';
74
75     case 5:
76       return '😬';
77
78     case 6:
79       return '😬';
80
81     case 7:
82       return '😬';
83
84     case 8:
85       return '😬';
86
87     case 9:
88       return '😬';
89
90     case 10:
91       return '😬';
```

Another thing that I could improve on is avoiding large chunks of code in a single file.

Here I made a switch case for the emojis in the liveblog item. This could also have been a simple array, and use indexing to get the desired emoji.

Summary

I'm always keen to improve my coding skills. Therefore, I asked John to have a look at my commits in Gitlab. John gave feedback on some specific things in my code and later we discussed some general improvements and good habits.

John didn't find any major faults in my code. I should use more template literals, do more checks to prevent errors and potentially define more variables, just in case a color needs to be changed, it needs only to be changed in one place.

In general I could pay more attention to code structure. Things like defining variables and functions in one place would make code more readable. This also applies to avoiding large chunks of code.

Learning Outcome Clarification

- Learning Outcome 5: Personal Leadership

Learning outcome 5 applies for this deliverable as I pay close attention to my own learning ability and I received feedback on my work.