|  |
| --- |
|  |
| Unit 5 – Writing JavaScript |
| Student 2729329 |

|  |
| --- |
| Dave Boyle  11-2-2020 |

# Table of Contents

# 

|  |  |
| --- | --- |
| Mapped Learning Outcomes | .…………….. [page](#_Theme) 2 |

# Mapped Learning Outcomes

## Potential Grades

## A

## Rich use of a wide variety of code constructs, objects, and functions, each appropriately used and elegantly constructed, with appropriate methods and designs for each task the code performs.

## Effective and intelligent use of advanced techniques including objects, recursion, and regular expressions.

## Well integrated with the HTML and CSS of the site, with good separation of data, process, and presentation.

## Well-commented, well laid-out and maintainable.

## The need for the code is strongly driven by the purposes, personas, and scenarios of Unit 1 and notably enhances the experience of the end user in that context.

## No apparent bugs, logic or run-time errors; any errors or exceptions treated well.

## Good interface design—highly accessible and usable.

## Works (or fails gracefully) with all browsers.

## B

## Use of a varied range of code constructs, objects, and functions, each appropriately used, with appropriate methods and designs for each task the code performs.

## Some use of advanced techniques including objects, recursion, and regular expressions.

## Well integrated with the HTML and CSS of the site.

## Well-commented, well laid-out and maintainable.

## The need for the code clearly derives from the purposes, personas, and scenarios of Unit 1 and is explicitly shown to improve the experience of the end user in that context.

## No glaring bugs, logic errors, or run-time errors; any errors captured and treated appropriately.

## Good interface design—accessible and usable.

## Works with all common desktop browsers.

## C

## A range of common code constructs, objects , and functions, mostly appropriately used, with appropriate methods and designs for each task the code performs.

## Limited use of advanced techniques including objects, recursion, and regular expressions.

## HTML and CSS integration works, with some separation of data, process, and presentation.

## Adequately commented, mostly properly laid out, and generally maintainable.

## Some relationship with the purposes, personas, and scenarios of Unit 1.

## Limited error handling.

## The occasional small bug, logic, or run-time error.

## Mediocre interface design but generally usable and accessible.

## Works in most modern desktop browsers.

## D

## Use of a limited range of common code constructs, objects, and functions.

## Those that are attempted are appropriately used by and large.

## Very limited or buggy use of advanced techniques including objects, recursion, and regular expressions.

## HTML and CSS integration works, but may not be very elegant.

## Some comments and mostly well laid out code, but limited maintainability.

## Tenuous or sketchy but explicit relationship between the purposes, personas, and scenarios of Unit 1.

## Poor error handling, the occasional show-stopping bug, logic, or runtime error, but mostly works well.

## Works in at least two common desktop browsers from different families (e.g., two or more of Firefox, Chrome, Safari, Opera, and IE).

## Mostly accessible and usable but some small problems here and there.

## Recommended Grading

The recommended grade for this unit is an A. I’ve corrected the minor errors I had from Unit 3 so that the HTML and CSS code validate with no errors. I’ve also used online JavaScript validators to make sure my code has no syntax errors. I’ve tried to keep it clean. I feel like my JavaScript implementation aligns to the personas:

* Menu descriptions – allows people to know what each section is about without clicking into it, so they can increase efficient use of time.
* Image Changer – this JavaScript that rolls through images on the trail review pages brings the page together and makes it look more professional than when images were at the bottom of the text, building trust of users as it looks better.
* Table Sorter – Really helps users find more appropriate data quickly. This is super important. Many different techniques are displayed in the code, pushing my grading to an A. My table sorter is both better and worse than the ones found online. It is better as it sorts both Strings and numbers of different digits without issue. It is worse because it requires naming all of the <TD> elements in the HTML code. However, at the end of the day there is value in the different techniques displayed in this code.
* Theme Picker – This implements a dark theme during later hours (8 PM to 7 AM) when people are typically winding down, and when a bright white page might not be appreciated. This helps build trust by increasing usability without personas needing to demand it. This was not originally planned, but I wasn’t sure if I was going to implement a table sorting program myself outside of the one I found online.

The code itself that I have implemented is clean, easy to follow, and well commented on. For the JavaScript that picks a CSS sheet based on time of day, I did default to the lighter CSS theme if the browser is not JavaScript enabled. I also made sure the script that rotates images encourages the user to enable JavaScript if not enabled and the user is alerted that table sorting will not work without JavaScript.

I also encouraged a user that had multiple body and head elements in her code to use a validator as it was her Unit 4 submission and felt she still had time to correct some code errors prior to portfolio submission. I tried to be polite in my response to her post, which is here: <https://landing.athabascau.ca/blog/view/6868159/unit-4-script-use-and-augmentation>

Ironically, this is when I discovered an error in my CSS syntax. I should have used an online validator myself. It’s fixed now.

I also posted on the landing the validators I’ve used to help the broader audience here <https://landing.athabascau.ca/discussion/view/6977473/online-validator-resources> and contributed to <https://landing.athabascau.ca/discussion/view/6961850/testing-sites> to provide a useful tool to test, for free, users websites on older browsers. My code works on all available current browsers I had available: Edge, Opera, Chrome, and Firefox. By far this was the most rewarding Unit of this course so far. Years ago, I had reused other’s scripting, but I have never written my own.