Diary Entry

Link to SCIS website: http://student.athabascau.ca/~matsph/

Work for this Unit:

Activities:

Scimmed the Unit 4 and 5 JavaScript FAQ

Watched LearnWebCode's Sticky Navigation Tutorial: https://www.youtube.com/watch?v=dZYy72ObKf0&index=43&list=PLr6-GrHUlVf8JIgLcu3sHigvQjTw_aC9C

Watched tutor4u's JavaScript & jQuery Tutorial for Beginners – 1 of 9: https://www.youtube.com/watch?v=VRnQOcVclS8

Added the JavaScript 'floater.js' to the website.

Changed some of the CSS to position the navigation field properly when it changes positioning.

How I have met the Learning Outcomes:

☐ Critique JavaScript code written by others, identifying examples of both good and bad practice.

This is the original javascript code written by LearnWebCode in the Sticky Navigation Tutorial video with the link above:

```
jQuery(document).ready(function() {

	var navOffset = jQuery("nav").offset().top;

	jQuery("nav").wrap('<div class="nav-placeholder"></div>');
	jQuery(".nav-placeholder").height(jQuery("nav").outerHeight());

	jQuery("nav").wrapInner('<div class="nav-inner"></div>');
	jQuery(".nav-inner").wrapInner('<div class="nav-inner-most"></div>');

	jQuery(window).scroll(function() {
		var scrollPos = jQuery(window).scrollTop();

		if (scrollPos >= navOffset) {
			jQuery("nav").addClass("fixed");
		} else {
			jQuery("nav").removeClass("fixed");
		}

	});

});
```

Explanation:

The “ready(function()” function is run when the page is loaded, and it retrieves the y position of the top of the ‘nav’ field. Then, it wraps the ‘nav’ field in a “nav-placeholder” wrapper, to use the “nav-placeholder” to act as a husk to be left over after the ‘nav’ is converted into a fixed field. It then makes an inner-wrapper inside the ‘nav’ field to have different CSS for the outer area and the inner area, and then another inner-wrapper to have even more differentiation between the outer and inner areas for CSS.
The “scroll(function()” function is run every time the user scrolls, and every time it retrieves the y position of the top of the viewport, and then it compares it to the y position of the top of the ‘nav’ field to determine if it makes the ‘nav’ field fixed or not.
Critique:

- Good practice:
  - The code is properly indented and has consistent style
  - Variables use camelCase
  - Each line doesn’t exceed 80 characters
  - Uses proper if statement form
- Bad practice:
  - The code has no comments, but these can be inferred from the tutorial video
  - Blank spaces don’t follow the function keywords
  - Certain ‘=’ operators are not separated from their operands
  - Variables are not declared before use

☐ Use JavaScript to add dynamic content to pages.

The JavaScript above changes the position property of the ‘nav’ element, which is a construction change in the DOM caused by the user.

☐ Modify existing JavaScript code to extend and alter its functionality and, where appropriate, to correct errors and cases of poor practice.

First, I corrected all of the bad practice problems listed above:

```
jQuery(document).ready(function () {
  //Variable declarations:
  var navOffset; //Holds the y position of the nav field
  var scrollPos; //Holds the y position of the top of the viewport

  //Retrieve the y position of the top of the nav field
      navOffset = jQuery("nav").offset().top;

  /*
  * Wraps the nav in a wrapper, so that when the nav is transformed into a
  * fixed nav, the rest of the page doesn't adjust to it being gone
  */
      jQuery("nav").wrap('<div class="nav-placeholder"></div>');

  //Set the height of the wrapper to the same height of the nav section
      jQuery(".nav-placeholder").height(jQuery("nav").outerHeight());

  /*
  * Wraps the fixed floating navigation inside the nav-inner class, so it can
  * be edited by CSS separately from the nav container
  */
      jQuery("nav").wrapInner('<div class = "nav-inner"></div>');
```

```
  /*
  * Wraps inside the nav-inner with another wrappeing of nav-inner-most,
  * so that it again can be edited by CSS separately
  */
      jQuery(".nav-inner").wrapInner('<div class = "nav-inner-most"></div>');

  //Runs every time the page is scrolled
      jQuery(window).scroll(function () {
    //Y pos of current scrolling position
    scrollPos = jQuery(window).scrollTop();

    //Changes the fixed state of the nav if it is above the scrollPos
            if (scrollPos >= navOffset) {
                  jQuery("nav").addClass("fixed");
            } else {
                  jQuery("nav").removeClass("fixed");
            }

      });

});
```

Then, I removed some unneeded statements, such as the ‘nav’ wrappers. I also changed the point at which the position type is changed to fixed (navOffset-10, to give some padding from the top):

```
jQuery(document).ready(function () {
  //Variable declarations:
  var navOffset; //Holds the y position of the nav field
  var scrollPos; //Holds the y position of the top of the viewport

  //Retrieve the y position of the top of the nav field
      navOffset = jQuery("nav").offset().top;

  //Runs every time the page is scrolled
      jQuery(window).scroll(function () {
    //Y pos of current scrolling position
    scrollPos = jQuery(window).scrollTop();

    //Changes the fixed state of the nav if it is above the scrollPos
            if (scrollPos >= navOffset - 10) {//-10 because it floats 10px
from top
                  jQuery("nav").addClass("fixed");
            } else {
                  jQuery("nav").removeClass("fixed");
            }

      });

});
```

Other Requirements

explain how the code improves the experience of the personas you created in Unit 1

Persona 1: Saun Simmerling

Scenario 2: Interesting Facts about the Rubik's Cube

Now that the navigation is only fixed after the user scrolls down the page below its sitting position, it won't be in his way when he navigates through the main pages.

Persona 2: Janice Miranda

Scenario 5: The 5*5*5

Now that the navigation is working properly, it won't be distracting, and she can see that it is meant to be an index of the different categories, as it sits next to the most popular puzzles. This should help her find what interests her easier.

Persona 3: Saun Simmerling

Scenario 6: Buying a new puzzle

The categories navigation is now properly situated for Robert to see in relation to the rest of the page, and feel comfortable using it as a navigation guide.

Notes:

Went well:

The modification of the JavaScript went the best, as I knew what needed to be changed according to the notes on JavaScript code conventions, and I knew what every line of code did, so I was able to remove lines that were useless to me.

Didn't go well:

Getting the JavaScript code to work on my website, as I needed to reference a JQuery library in order to have the website use the code properly. I ended up having to use a library, even though that is supposed to be covered in a later unit.

What was most difficult and why:

Relating the changes to the Unit 1 personas, scenarios, and site-map. It is only a simple feature, that doesn't directly help or is crucial to any scenario, so I had to find out how it helped the scenarios indirectly.

If done again, would it be done differently and why:

I would have chosen to add code that adds YouTube player widgets, since they are a more direct help to the Unit 1 personas and scenarios, and the code would be much more uniform in structure, and probably easier to find.

How did previous experience help/hinder completing the tasks:

Previous experience with using C++, Java, and Ruby programming languages helped me understand the JavaScript code, as it had the same structure and logic inherent in all of those languages.

Most surprising thing learned:

JavaScript can use libraries, and they can be imported directly from another website.

Most useful thing learned:

JavaScript works mostly by changing the DOM, and can add, remove, or change HTML fields.

Map to Course Outcomes

In the original site mock-up, the ‘Other_Puzzles.png’ has a categories box that was supposed to sit on the left side and follow the user as they scrolled down the page. The JavaScript code added accomplishes that, so that now the categories box isn’t just in a fixed position, but it sits next to the top of the ‘Most Popular’ puzzle list until the user scrolls below the top of the navigation box.

Self-assessment:

Learning Outcome	Evidence of Meeting the Learning Outcome	Your Own Assessment of the Grade You Believe Would Be Appropriate	Tutor’s Justification of Grading (optional)
Apply a structured approach to identifying needs, interests, and functionality of a website.	I have made 7 scenarios that give structured processes of how the anticipated audience will address their needs with the website. I also took into account the constraints users might have, such as different operating systems, slow computers, and browsing the site using mobile devices.	B	
Design dynamic websites that meet specified needs and interests.		A, B, C, D	

Write well-structured, easily maintained, standards-compliant, accessible HTML code.	Sample1.html and Sample2.html are edited to show my ability to write standards-compliant and structured HTML. The errors that used to exist in the code are listed in the learning diary under the section "Other Requirements". The 11 pages I wrote for my website all have links easily clickable on a mobile device, and are all well-structured and standards-compliant.	B	
Write well-structured, easily maintained, standards-compliant CSS code to present HTML pages in different ways.	The CSS I wrote is either in the external CSS file, or in a one-time internal CSS in the 'other_puzzles.html' page. In both, the code is organized with comments and well-structured using proper spacing, character case, property ordering, and naming schemes. It is easily maintained as the external CSS file is less than 130 lines long, and the internal CSS only defines styling for one selector. It is standards compliant with WCAG 2.0 by having no obscure or hard to read text, and proper contrast.	A, B, C, D	

Use JavaScript to add dynamic content to pages.	I have added the JavaScript file 'floater.js' which dynamically changes the positioning of the navigation field depending on where the user's viewport is.	B	
Critique JavaScript code written by others, identifying examples of both good and bad practice.		A, B, C, D	
Select appropriate HTML, CSS, and JavaScript code from public repositories of open source and free scripts that improves your site and that enhances the experience of site visitors.		A, B, C, D	
Modify existing HTML, CSS, and JavaScript code to extend and alter its functionality, and to correct errors and cases of poor practice.		A, B, C, D	
Write well-structured, easily maintained JavaScript code following accepted good practice, including		A, B, C, D	
• general appearance and form: commented, properly laid out, appropriate capitalization		A, B, C, D	
• structure: modular, using functions and objects effectively		A, B, C, D	
• standards-compliant		A, B, C, D	

• accessible		A, B, C, D	
Write JavaScript code that works in all major browsers (including IE, Mozilla-based browsers such as Firefox, Opera, Konqueror, Safari, Chrome).		A, B, C, D	
Effectively debug JavaScript code, making use of good practice and debugging tools.		A, B, C, D	
Use JavaScript libraries (e.g., JQuery) to create dynamic pages.		A, B, C, D	
Use JavaScript to access and use web services for dynamic content (AJAX, JSON, etc.).		A, B, C, D	
Overall		A, B, C, D	

DOM Notes

DOM: Document Object Model

Javascript employes the DOM to describe the page as a hierarchy tree of branches and leaves.

Uses tags and attributes to make hierarchy tree.

Document: The DOM

Window: The displayed page

Other Javascript Notes

Javascript is read by a program, rather an interpreter or a compiler.

Some versions semi-compile it (to a faster format, but not machine code), or remove comments

JavaScript Code conventions

The keyword *function* is always followed by one space

All binary operators besides '.', '(', and '[' Should be separated from their operands by a space

Every ',' should be followed by a space or line break

Each semicolon should be followed by a line break, except in for statements, which are separated by a space

All variables should be declared before use