

Unit Five

Three Site Proposals

In all cases it is important to remember the coding might not accurately be an exact copy of pseudocode if the pseudocode leads to bugs, but it provides a rough idea of implementation.

Proposal One

For the first proposal I would like a situation where the menu is changed to include JavaScript. The JavaScript will display a different information box as you mouse over the different headings so that people know what is included in that section of the website. This is appropriate for all of the personas as the common theme was to have information organized well so they can efficiently browse the site.

If a person moves the mouse over the category it will do something like the below image:



This saves the user time.

Pseudocode

Set Objects in Array to pair menu items and description

1. On mouse over of a menu item
 - a. Clear any menuInfoBox displaying information about a menu item
 - b. Display menuInfoBox for paired menuItem of Object
2. On mouse off of a menu item
 - a. Delay one second
 - b. Clear any menuInfoBox with fade over 1.5 seconds

Proposal Two

The second proposal I have enhances the display of the review sections. Right now, all of the images are at the bottom and the appearance is kind of clunky and unprofessional. I have engaged friends to review my site and provide feedback. The feedback of display and how the page flows has been mostly positive, however one consistent piece of feedback remains that the design is clunky around the area of images on the review page. In order to secure return visitors from the personas and build potential trust issues,

Learning Diary

this must be resolved. I plan to implement scripting that will scroll through images and captions and stick it on the left side of the review text. This will free up the awkwardness of where the pictures currently lie at the bottom and really pull everything together. It will be important to address the current format of the picture boxes to ensure that they will work with a dark or light theme.

Pseudocode:

Define 6-9 Objects of Images and Captions

1. Display Object
2. Count 3 Seconds
3. Move Forward
4. Forward is Change Object +
 - a. Check Object location in Array of Objects
 - i. If Object is at end, set Object to First Object in Array
 - ii. Else Object ++
5. Return to 1

I hope to implement a tiny arrow right and left people can click on to move forward or backward in objects. Moving forward in object is defined in #4 of pseudocode above.

Move backwards would be:

1. Backwards is Change Object –
 - a. Check Object location in Array of Objects
 - i. If Object is at start, set Object to Last Object in Array
 - ii. Else Object –
2. Return to #1 in above pseudocode.

In both cases the object allows for the user to click and bring up a larger version of the image.

Proposal Three

In this final proposal I would like to make the Reviews section sortable so people can sort by distance, review ratings, and elevation. This fits strongly with the personas outlined who are after information in an organized and readily available manner. Being able to sort through the reviews will allow quicker access to the information required.

Pseudocode:

Create objects for each review containing link, name, distance, elevation, and rating.

Define variables of 0,1,2 for distanceSort, elevationSort, and reviewSort ratings where 0= no sort, 1= descending, 2= ascending

1. If click on
 - a. Elevation heading
 - i. If elevationSort = 0
 1. Set distanceSort = 0
 2. Set ratingSort = 0
 3. Sort objects by descending
 4. Set elevationSort = 1
 - ii. Else If elevationSort = 1
 1. Sort objects by ascending
 2. Set elevationSort = 2
 - iii. Else (elevationSort = 2)
 1. Sort objects by descending
 2. Set elevationSort = 1
 - b. Distance heading
 - i. If distanceSort = 0
 1. Set elevationSort = 0
 2. Set ratingSort = 0
 3. Sort objects by descending
 4. Set distanceSort = 1
 - ii. Else If distanceSort = 1
 1. Sort objects by ascending
 2. Set distanceSort = 2
 - iii. Else (distanceSort = 2)
 1. Sort by descending
 2. Set objects distanceSort = 1
 - c. Rating heading
 - i. If ratingSort = 0
 1. Set elevationSort = 0
 2. Set distanceSort = 0
 3. Sort by descending
 4. Set objects ratingSort = 1
 - ii. Else If ratingSort = 1
 1. Sort by ascending
 2. Set objects ratingSort = 2
 - iii. Else (ratingSort = 2)
 1. Sort by descending
 2. Set objects ratingSort = 1

The sorting algorithm will be based off on the sorting algorithm at the following site:

<http://math.hws.edu/eck/cs124/javanotes7/c7/s4.html>

This is the most ambitious of the three implementations but will provide the best challenge and a good test of skills.

This concludes my three proposals.

Three JavaScript Implementations

main.js

What an incredible learning journey. The first proposal was relatively simple and I researched how to use JavaScript in conjunction with CSS to create a box that would fade out explaining the section of the site. I was going to make objects and have the text in the JavaScript. Instead, I typed the description directly in the HTML using a function that put the text into the menuItemInfo box. It was just as simple to do it this way and it let me play with having parameters in calling a function. I used <https://www.w3schools.com/js/default.asp> only for this.

berthafalls.js (and other similar named script files)

The second proposal was a bit different. It's scrolling through pictures to make it appear nicer than the page did in the past, and change elements of the picture holder by changing pictures every 3 seconds. Furthermore, I wanted to include a function to go forward or backwards in photos. This went very close to what was done in my pseudocode and I used only <https://www.w3schools.com/js/default.asp>. I did make a mistake in that my code originally said the following:

```
document.getElementById('imageControl').innerHTML = '<h3>CLICK FOR LARGER</h3>';  
  
document.getElementById('imageControl').innerHTML = '<a target="_blank" href=""';  
  
document.getElementById('imageControl').innerHTML = images[x].full + '>';  
  
document.getElementById('imageControl').innerHTML = '<img src=""';  
  
document.getElementById('imageControl').innerHTML = images[x].thumb + '> alt=""';  
  
document.getElementById('imageControl').innerHTML = images[x].description + '></a>';  
  
document.getElementById('imageControl').innerHTML = '<div class="desc">';  
  
document.getElementById('imageControl').innerHTML = images[x].description + '></div>';
```

https://www.w3schools.com/js/js_htmlDOM_document.asp

The same page that talked about how to change CSS style with getElementById was at that page and cleaned up the code nicely and made it easier to follow.

The third one, a table sorter, had me panicked and worried. When I looked at my pseudocode, it had me a bit worried, and when I googled ideas I saw

https://www.w3schools.com/howto/howto_js_sort_table.asp and I thought to myself I should not cheat

by reviewing another website. I did implement that code on that site, but it didn't really work for sorting numbers. A google search found a way to make it sort numbers, but only if they were the same digits in length.

themePicker.js (bonus Script)

Feeling guilty for implementing code from online, I devised and made code to provide a dark theme at night, and a light theme during the day. This was beneficial as I learned when using `document.write()` might be more appropriate than using `getElementById` and trying to replace elements.

https://www.w3schools.com/js/js_output.asp provided a clear warning that "Using `document.write()` after an HTML document is loaded, will delete all existing HTML". What about before? That was what I needed to wrap my head around, and I quickly changed code and got it to work. For the record, I also referenced https://www.w3schools.com/jsref/jsref_gethours.asp on how to get the script to get the hour of the day for use in a test. I then changed the image to be a transparent png to work with either theme. Really happy how this turned out, even though I did not originally plan this.

tableSort.js

I decided to go back to the table and closely followed my pseudocode. What a relief that I found https://www.w3schools.com/js/js_array_sort.asp as it showed me I would not need to use two variables to juggle positions throughout all the objects of the trails like used in the Java textbook. Instead I could use the sort function. This wouldn't seem to work for Strings, and even though I didn't say I would sort by string, I found a way to add that in by learning the appropriate function at <https://stackoverflow.com/questions/1129216/sort-array-of-objects-by-string-property-value> through a google search. Because I used phrases instead of a rating system for difficulty and to describe how busy a trail is, I also added in a couple of functions to convert a number rating (the program used for sorting) into a string for when the table of table was rendered.

I am super happy with this programming. Even though this may not be the prettiest way to sort data, I think it demonstrates a few different items such as using arrays, for loops, and multiple functions.

I also made sure to thoroughly comment so anyone could pick it up and I contributed to the Landing to help others test their code and validate their code to reduce errors and become a better coder.