Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-js-and-css-challenge/grade/ekh3

IT202-008-S2024 - [IT202] JS and CSS Challenge

Submissions:

Submission Selection

1 Submission [active] 2/12/2024 9:36:57 PM

Instructions

COLLAPSE

- Reminder: Make sure you start in dev and it's up to date
    1. `git checkout dev`
    2. `git pull origin dev`
    3. `git checkout -b M3-Challenge-HW`

1. Create a copy of the template given here: https://gist.github.com/MattToegel/77e4b66e3c73c074ea215562ebce717c
2. Implement the changes defined in the body of the code
    1. Hint: You may want to use your browser's developer tools to see the script that's pulled in, this may help with a few challenges
3. **Do not** edit anything where the comments tell you not to edit, you will lose points for not following directions
4. Make changes where the comments tell you (via TODO's or just above the lines that tell you not to edit below)
    1. **Hint**: Just change things in the designated `<style>` and `<script>` tags
    2. **Important**: The function that drives one of the challenges is `updateCurrentPage(str)` which takes 1 parameter, a string of the word to display as the current page. This function is not included in the code of the page, along with a few other things, are linked via an external js file. Make sure you do not delete this line.
5. Create a branch called M3-Challenge-HW if you haven't yet
6. Add this template to that branch (git add/git commit)
7. Make a pull request for this branch once you push it
8. You may manually deploy the HW branch to dev to get the evidence for the below prompts
9. Once done, generate the output/submission file
10. Add, commit, and push the submission file
11. Close the pull request by merging it to dev (double-check all looks good on dev)
12. Manually create a new pull request from dev to prod (i.e., base: prod <- compare: dev)
13. Complete the merge to deploy to production
14. Upload the same submission file to Canvas
15. Checkout dev and pull latest changes to prepare for future work

Branch name: M3-Challenge-HW

Tasks: 15 Points: 10.00

Screenshots (4 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: 5 Screenshots based on checklist items

Checklist

*The checkboxes are for your own tracking

#	Points	Details
#1	1	A screenshot showing the Primary page with the checklist items completed (the view that initially loads)
#2	1	A screenshot showing the page after the login link is clicked with URL shown
#3	1	A screenshot showing the page after the register link is clicked with URL shown
#4	1	A screenshot showing the page after the profile link is clicked with URL shown
#5	1	A screenshot showing the page after the logout link is clicked with URL shown
#6	1	Screenshots should be from heroku dev

Task Screenshots:

Large Gallery

Checklist Items (1)

#1 A screenshot showing the Primary page with the checklist items completed (the view that initially loads)

Initial Screenshot

Checklist Items (2)

#2 A screenshot showing the page after the login link is clicked with URL shown

#6 Screenshots should be from heroku dev

Login Screenshot

Checklist Items (2)

#3 A screenshot showing the page after the register link is clicked with URL shown

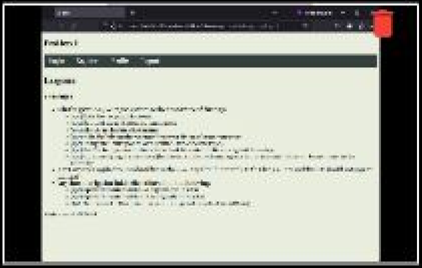
#6 Screenshots should be from heroku dev

Checklist Items (2)

#4 A screenshot showing the page after the profile link is clicked with URL shown

#6 Screenshots should be from heroku dev

Register screenshot | Profile screenshot



Checklist Items (2)

#5 A screenshot showing the page after the logout link is clicked with URL shown

#6 Screenshots should be from heroku dev

Logout screenshot

● COLLAPSE

Explanations (4 pts.)

● COLLAPSE

Task #1 - Points: 1

Text: Briefly explain how you made the navigation horizontal

Details:
Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

Utilizing CSS, I would navigate the nav tag to control all the elements that pertain to the challenge. I would use space, looking like "nav ul", as the CSS selector and do a few things like setting the list style to none and removing any padding and margins. Afterward, I would take all the children's anchors of nav ul by using "nav ul a" as a selector and setting the float to left with a padding of 10 pixels. The result is the nav bar being horizontal without any item markers and slightly spaced.

● COLLAPSE

Task #2 - Points: 1

Text: Briefly explain how you remove the navigation list item markers

Details:
Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

Using the nav ul, which is the unordered list parent of all the navigatable locations, as the selector, it is possible to change its list style to none which would remove the list marker. The way I did this was using a space to separate nav and ul, this would tell CSS to address the children of the nav. As there is only one nav tag, it would address the elements needed.

●
^ COLLAPSE ^

Task #3 - Points: 1

Text: Briefly explain how you gave the navigation a background

Details:
Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

By selecting the nav and the children that are ul, using "nav ul" as a selector, I would set the background color to a color I thought looked good using a hex code.

●
^ COLLAPSE ^

Task #4 - Points: 1

Text: Briefly explain how you made the links (or their surrounding area) change color on mouseover/hover

Details:
Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

By using the :hover modifier for the anchor selection in CSS, it would allow the CSS to change when the user hovers over the value. The entire selector in this case will be "nav a:hover" and by adding a background color to the selected elements would create the change color on the mouseover feature.

●
^ COLLAPSE ^

Task #5 - Points: 1

Text: Briefly explain how you changed the challenge list bullet points to checkmarks (✓)

Details:
Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

In this case, I would follow the HTML and figure out where each bullet point is. In this case, I found out it was the children li under a ul, li, and ul. Therefore, for the CSS I selected specifically those elements using "ul li ul li." With those items selected, I would change the list-style-type to the checkmark. The result is a checkmark on all the list bullet points.

● COLLAPSE

Task #6 - Points: 1

Text: Briefly explain how you made the first character of the h1 tags and anchor tags uppercased

Details:
Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

To select the first character of both, using the comma as a separator between h1 and a would select all the h1 and anchor tags. However, to get the first character I would add the modifier ::first-letter to both to only select the first letter of all the matching tags. Lastly, the style would require a text-transform to uppercase, which sets every first character to uppercase. The result is the following as a selector, "h1::first-letter, a::first-letter."

● COLLAPSE

Task #7 - Points: 1

Text: Briefly explain/describe your custom styling of your choice

Details:
Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

I sought to make things look a little less generic, so I would change the body background color utilizing just the body as a selector. In addition, I would change the text in the navbar to be a clearer color with the navbar color. In addition, I would change the sizes of the text to give it a hierarchy using font size with large and medium values. Using the same selector to change the list to checkmarks, I used it to change the font size to medium, and for all the other lists, it would be a large size. Lastly, I would center the text in anchors for the navbar by using the same selector for helping it make it horizontal.

● COLLAPSE

Task #8 - Points: 1

Text: Briefly explain how the styling for the challenge list doesn't impact the navigation list

Details:
Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

The functionality of the navigation list is independent of the style as its functionality comes from the HTML. As long as the elements are clickable or interactable, the styling would just change the aesthetics of the page.

● COLLAPSE

Task #9 - Points: 1

Text: Briefly explain how you updated the content of the h1 tag with the link text

Details:
Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

Using JavaScript to process the document, I would add the getCurrentSelection() function to the window load event listener. This allows the function to get running, but to get it run every time something is clicked, a window.onhashchange listener would be used. Within that function, to get the current page, it would get the page's hash using window.location.hash. However, this would return the hash as well, so using a string slice with the starting index of 1, it would slice from the first letter to the end. The value of the slice will be stored in a variable called currPage. Lastly, by calling updateCurrentPage(currPage) with the currPage string, it would update the h1 tag.

● COLLAPSE

Task #10 - Points: 1

Text: Briefly explain how you updated the content of the title tag with the link text

Details:
Explanation clearly describes solution and why it works (explain the code, don't use the code as an explanation)

Response:

Same as the last task, however, updateCurrentPage would also update the title tag. To be more generalized, in JavaScript, a hash change listener would look for a change and when there is a change the hash is gotten and sent to the updateCurrentPage function with the string.

● Misc (2 pts.)

^ COLLAPSE ^

●
^ COLLAPSE ^

Task #1 - Points: 1

Text: Comment briefly talking about what you learned and/or any difficulties you encountered and how you resolved them (or attempted to)

Details:
At least a few sentences

Response:

Throughout this challenge, it was interesting to see how such a stale default website can turn into something more modern with only a few lines of CSS and the same with functionality with JavaScript. The main challenge for this was navigating what the functions did. As I felt weak in JavaScript before starting, reading the code to see where to start was difficult, but after figuring out what listeners were, it greatly helped speed up the process.

●
^ COLLAPSE ^

Task #2 - Points: 1

Text: Add a link to your pull request (hw branch to dev only)

URL #1
https://github.com/WokFriedE/ekh3-it202-008/pull/9

●
^ COLLAPSE ^

Task #3 - Points: 1

Text: Add a link to your production file

URL #1
https://ekh3-it202-008-prod-e9bbcd10cf36.herokuapp.com/challenge.php

●
^ COLLAPSE ^

Task #4 - Points: 1

Text: Waka Time (or related) Screenshot

Checklist *The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Screenshot clearly shows what files/project were being worked on (the duration of time doesn't correlated with the grade for this item)

Task Screenshots:

Large Gallery

Checklist items (0)

WakaTime screenshot