Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT202-008-S2024/it202-js-and-css-challenge/grade/lm457

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

Submissions:

Submission Selection

1 Submission [active] 2/12/2024 11:15:03 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

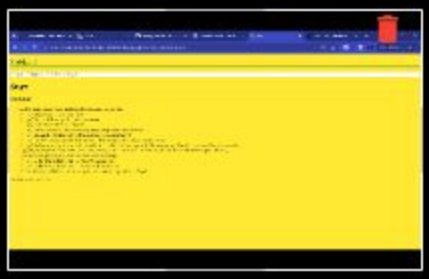
Checklist

*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

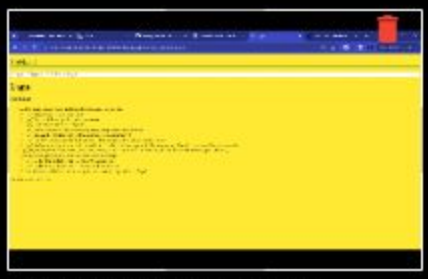
Task Screenshots:

Large Gallery



Checklist Items (1)

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

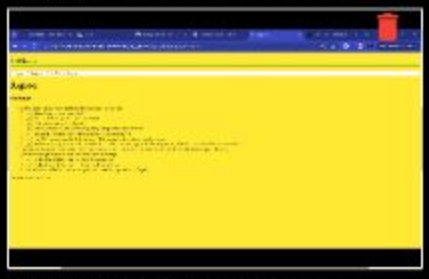


Checklist Items (1)

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

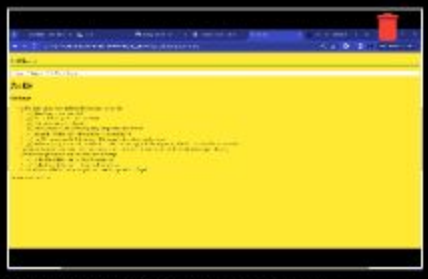
1st check.

2nd check.



Checklist Items (1)

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



Checklist Items (1)

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

3rd check.

4th check.

Checklist Items (1)

#5 A screenshot

showing the page after the logout link is clicked with URL shown

5th check.

Explanations (4 pts.)

COLLAPSE

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:

For every element in <ul> inside the <nav> tag, using the "flex" function for "display" meant all of its children are horizontally structured. Meaning, the navigation links inside the <ul> element now appear horizontally beside one another, instead of just creating one vertical stack.

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:

Since the list item markers appear before each list item <li>, we remove those inherent those markers by setting 'list-style-type' to none for <ul> elements in the <nav> tag, thus resulting in all of the navigation links appearing without any of those markers on the main menu.

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:

We can give the navigation a background by altering the <ul> element in the <nav> tag containing the navigation links. Using the background color property in CSS, we can choose from several predefined colors for the <ul> element.

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:

In order to give a different and distinct color on an area surrounding a link when our mouse hovers over it, we need to use the :hover class within a more specified region. To do so, we include the tags <nav ul li a> to specifically focus on the <a> tags within the navigation list items, which determine the color of the text of those specific navigation links. Using the color property, we can assign a color of our preference to highlight the region when a mouse hovers over it.

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:

Since we had previously removed the predefined markers, in order to insert checkpoints, we need to specify the region before the list items and their contents. Therefore, using the ::before class to specify the region before <li> tag which contains the list items would allow us to insert the checkmarks at our desired location. Using the content property, we can enter the checkmark symbol within double quotes to ensure each list item has a checkmark prior to their beginning. We can additionally elaborate on the styling of those checkmarks as we prefer.

Task #6 - Points: 1

COLLAPSE

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:

In order to specifically target the characters within the h1 and anchor tags, we need to narrow down our region to where we implement our capitalization changes. Using 'h1, a 'as the desired area, we can alter the contents of all text contained within those elements. Since we intend to capitalize only the first character rather than the entire textual content, we use 'capitalize' instead of 'uppercase' for the 'text-transform' property.

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 made several stylistic choices to customize the appearance of my webpage. I first spaced out my list items and assigned the default non-hovering color to the navigation links. I also spaced out the unordered lists and designated the border and text color of my header to make it pop out a bit more. I also made changes to the body of the page as a whole, including my preferred choice of font, the color for that font, as well as the background color of the page, which I chose to be a bright yellow, since yet again I intend to draw attention to the page.

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:

Since we use different tags/selectors to specify the particular region we intend to modify, we can ensure that our modifications don't overlap with one another. Meaning, when stylizing our challenge list, I particularly used the <ul> and

 tags to target the contents only within those specific selectors. Similarly, I included the <nav> tags when targeting the navigation list items and stylizing them corresponding to a different set of standards. Therefore, the stylistic choices made in the challenge list don't impact those made in the navigation list.

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:

In order to update the h1 tag content in correspondence to changes made when a click occurs, we need to both monitor when a click is made, as well as check to see if the click is made on a navigation link, at which point we can retrieve the link's textual content and update our h1 tag. To do this, we create an event listener, whose event is defined to be a click. Then, using an if statement, we check to see if the click has been made at a navigation link. If it has, we inhibit the default navigation behavior and instead copy the contents of the link. We then pick our h1 selector and modify its text to reflect the updated text obtained from the navigation link.

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:

In a similar vein to the previous changes, in order to update the content of the title tag, we need to monitor the "clicks" made on navigation links, using an event listener whose event is defined to be a click, inside the <a> tag where the navigation links are located. When a user clicks on one of these navigation links, we specify commands within the event listener function that extract the contents of the clicked link. The document's title is then updated to match the content retrieved from the clicked link. (We can notice the change in the URL as well).

COLLAPSE

Misc (2 pts.)

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:

This was the most difficult and time-intensive assignment I've encountered in this class so far. Since CSS and JavaScript are both relatively new to me, I had to scramble all over the place to find the necessary predefined functions and properties unique to both languages and then figure out how to make use of them. Not to mention, using the different selectors was also something that felt alien to me and I still think I haven't fully adapted or grasped the concept. I almost felt like this assignment was a little rushed given the relatively modest amount of time we spent actually working on these languages. Even then, I learned an incredible amount about using CSS to stylize and design webpages and how to effectively update text or any other form of content based on user input.

COLLAPSE

Task #2 - Points: 1

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

URL #1
https://github.com/Kith07/spring2024-IT202/pull/9

COLLAPSE

Task #3 - Points: 1

Text: Add a link to your production file

URL #1
https://it202-lm457-prod-cf97d0c34365.herokuapp.com/Module_3/challenge.html

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)

Waka Time report on this assignment.