



CPSC1520

Client Side Scripting With JavaScript

Due Date: End of Class

Weight: 5%

Introduction

In this lab you are going to use the skills developed in the last couple of weeks. You have to get the user input from the first name and last name fields inside the submit event handler function. Once you have the first and last names join the two strings and update the text of the output element.

Below you will find the markup showing you where in the HTML document to place the full name string.

```
<ul class="py-2 text-sm ">
  <li>
    <span class="text-xs text-zinc-400 ">Full Name: </span>
    <!-- PLACE THE FULL NAME STRING HERE -->
    <output class=" inline-block font-medium text-zinc-500">Jane
      Doe</output>
  </li>
</ul>
```

index.html

Check

Check to make sure that the `<script>` tag is in the head of the document with the `defer` attribute and that the `src` path points to your script inside the `js` folder.

```
<script src="js/your-file-name.js" defer></script>
```

index.html



Async And Defer What's The Difference

In years past we used to place the script at the bottom of the page next to the closing body tag. It is now considered to be a best practice to place your linked scripts in the head of the document and mark them with the defer attribute if you want faster page load times.

<script>(blocking)

When the browser encounters a script tag it immediately stops parsing the HTML document and goes and fetches the document. This is called render blocking (parsing paused) Once the script is finished downloading the script is executed and the browser continues rendering the HTML.



<script async>(blocking)

Async is similar to just a plain old script tag but with one major exception. Async downloads the javascript file but the browser can continue parsing the HTML while the document is being fetched. Once downloaded the script will then execute.



<script defer>(non-blocking)

Defer downloads the javascript and like async doesn't block the browser from parsing the HTML. The script will not execute but will wait till the browser has finished parsing the HTML then the javascript will run.



Further Reading: <https://flaviocopes.com/javascript-async-defer/>



Task

Give the output, form and the two input elements semantic id's. Use snake case Google's Style Guide for HTML suggests that this is best practice. So let's follow it.

Task

Along with the id attribute for the form inputs type text, add to placeholder attributes. One for each field. You can use the placeholder text Jane and for the other use Doe. You can use any names you want. Just make sure whatever you use appears in the output element for the initial state of the user card.

Task

Each of the input elements should have a required attribute and a name attribute. The name attribute can be the same as the id .

```
<input
  id=""
  name=""
  placeholder=""
  required
  type="text"
  autocomplete="given-name"
  class="block w-full rounded-md border-0 py-2 px-4 text-gray-900 shadow-sm
        ring-1 ring-inset ring-gray-300 placeholder:text-gray-400 focus:ring-2
        focus:ring-inset focus:ring-indigo-600"
/>
```

index.js



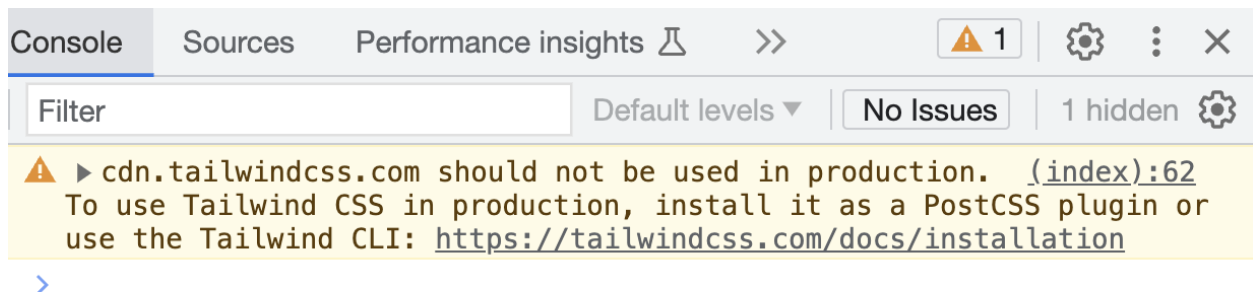
CPSC1520

Client Side Scripting With JavaScript

Submission

Once you have finished your lab deploy it to [Netlify](#). Once your site is deployed check that there are no errors in the console and everything works as intended. You will see a warning similar to the following error (chrome). Depending on your browser you may not see it.

Do not worry about this warning. I used tailwind css to style the UI. The warning recommends that you don't use Tailwind CSS from a CDN (content delivery network). You need a build tool to use Tailwind CSS. If there is interest I will show you how to use and build your JavaScript app for production using Tailwind CSS as the styling solution.



Submit To Moodle

Netlify URL make sure to add it as a link not just text please.

NOTE:

Late submission will not be graded. Sites that produce a 404 error page not found will also receive a grade of zero.



CPSC1520

Client Side Scripting With JavaScript

Marking Key

| Task | Mark | Task Value |
|--|------|------------|
| Form Submit Event. Submit event add to the form element.. | | 2 |
| Input Element Attributes. Both input elements have all the required attributes.. | | 2 |
| String Concatenation. The input from both fields is concatenated into a single string.. | | 2 |
| Output Element Updates The output element updates the users full name. | | 2 |
| Deployment Deployed to netlify with no errors. "Ignore the tailwind css warning" | | 2 |
| | | 10 |



CPSC1520

Client Side Scripting With JavaScript

Marking Rubric

| Marks | 5 Marks Criteria |
|---------------|---|
| 5 [0] | Task was completed with the highest proficiency, adhering to best practices. The tasks demonstrate a high level of proficiency implementing the subject matter and the presentation of the content is to a professional standard. |
| 4 [-1] | Task was completed well, with some minor mistakes. Well above average work, shows good understanding of the implementation details of subject matter. Tasks show a high degree of competence. |
| 3 [-2] | Task was completed satisfactorily. Some features are missing or incorrectly implemented. Shows a moderate level of understanding in the task with room for improvement. |
| 2 [-3] | Task completion is below average, the task was poorly completed. Shows understanding of the task and the requirements to implement, but implementation was poorly executed. |
| 1 [-4] | Some of the tasks were completed. Shows a lack of understanding in the subject matter and very poor execution. |
| 0 [-5] | Not completed. |

| Marks | 3 Marks Criteria [minus] |
|---------------|---|
| 3 [0] | Task was completed well, adhering to best practices, and shows a high degree of proficiency with the subject matter guidelines. |
| 2 [-1] | Task was completed satisfactorily. Some features are missing or incorrectly implemented. Shows a moderate level of understanding in the task with room for improvement. |



CPSC1520

Client Side Scripting With JavaScript

| | |
|---------------|--|
| 1 [-2] | Some of the tasks were completed. Shows a lack of understanding in the subject matter and very poor execution. |
| 0 [-3] | Shows a little to no degree of competence in completing the task; not completed. |

| Marks | 1 Marks Criteria |
|--------------|-------------------------------|
| 1 | Task completed satisfactorily |
| 0 | Task was not completed |