Wade Moulton

09/15/2020

CS 4720 W01

Assignment 2 Report

Weblink: <http://studentweb.kennesaw.edu/~jmoulto2/>

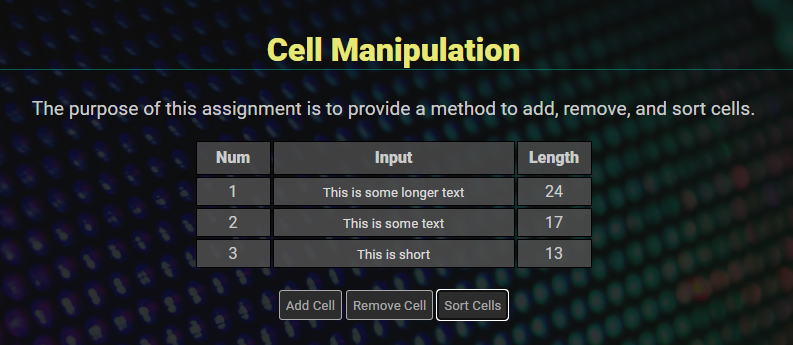
**Assignment Description**

Assignment 2 has two parts that I have completed utilizing my student webpage. These parts consist of a cell manipulation section wherein a user may add/remove cells as well as input text. The cell manipulation section also allows the sorting of the cells by the text length in decreasing order. The second part of the Assignment 2 page consists of the 3 scripts that were required to add: squareNumber, fixStart, and notBad. I have decided to implement these as interactive sections of the webpage using JavaScript.

**Section 1**

*Script file: Assignment2.js*

Section 1 is the Cell manipulation portion of the assignment 2 webpage. To achieve this I decided to make use of table rows and table data. Each row has 3 data cells with the following layout: cell 1 is used to number each row, cell 2 is used to house the input box for inputting text strings, and cell 3 is used to display the length of the string in its corresponding row. There are 3 buttons in this section as well. The first button adds a cell by adding a new table row. The delete cell button removes a row from the table until it gets to the first row. The sort cells button sorts the cells based off the cell 3 data cells which is where the text lengths are stored. For the CSS I used the base CSS from assignment 1 and added a new *Assignment2.css* file to alter some things specific to this page, such as the table styling.



**Section 2**

*Script files: squarenumber.js, fixstart.js, notbad.js*

Section 2 contains the 3 additional scripts needed for the assignment. I decided to implement these on the page to provide a more interesting way to interact with each. The squareNumber function of *squarenumber.js* takes one argument *num* and returns the square of that number. Before this function is called, the input is checked to validate that it is a number and, if not, a message is displayed on the error. Otherwise, the message of the squared value is displayed.

The fixStart function in *fixstart.js* first takes the text input and assigns the first letter to a variable *first*. The remaining substring from the second letter through the end of the string then uses the replaceAll function to replace all occurrences of the first letter with ‘\*’. Both the lower and uppercase variations are checked.

The last function notBad in the *notbad.js* file first searches for the index of the word ‘not’ and assigns it to a variable. It then looks for the word ‘bad’ and assigns its index to a variable. Next, if the word not appears before the word bad, a string is returned from the beginning until the beginning of ‘not’ concatenated with the word good and the substring from ‘bad’ until the end of the string.



At the bottom of this report I have added collapsible sections for my source code for this assignment and will also submit the source code within the assignment submission folder.

# Assignment2.html

*<!--*

*Name: Wade Moulton*

*Date: 09/15/2020*

*Class: CS 4720*

*Section: W01*

*Instructor: Dr. Sarah North*

*-->*

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <link rel="stylesheet" href="../styles.css">

    <link rel="stylesheet" href="../css/Assigment2.css">

    <title>Assignment 2</title>

</head>

<body>

    <header class="main-header">

        <nav class="nav main-nav">

            <ul>

                <li><a href="../index.html">Home</a></li>

                <li><a href="./Assignment2.html">Reset Page</a></li>

            </ul>

        </nav>

    </header>

    <h1 class="my-name">Assignment 2</h1>

    <div class="container">

        <section class="content-section">

            <h2 class="section-header">Cell Manipulation</h2>

            <p class="inner-content"> The purpose of this assignment is to provide a method

                to add, remove, and sort cells.

            </p>

            <div id="cell-div">

                <table id="input-table">

                    <thead>

                        <tr>

                            <th>Num</th>

                            <th>Input</th>

                            <th>Length</th>

                        </tr>

                    </thead>

                    <tr>

                        <td id="td-first-num"></td>

                        <td><input type="text" id="td-first-input"></td>

                        <td id="td-first-count"></td>

                    </tr>

                </table>

            </div>

            <div class="btn-container">

                <button id="add-cell-btn">Add Cell</button>

                <button id="remove-cell-btn">Remove Cell</button>

                <button id="sort-cell-btn">Sort Cells</button>

            </div>

        </section>

        <section class="content-section">

            <h2 class="section-header">squareNumber() Function</h2>

            <p class="inner-content">Enter a number and press the button to square!</p>

            <p id="square-number-value"></p>

            <input type="text" class="functionInputs" id="square-number-input">

            <button id="square-number-submit-btn">Square Me!</button>

            <button id="square-number-reset-btn">Reset</button>

        </section>

        <section class="content-section">

            <h2 class="section-header">fixStart() Function</h2>

            <p class="inner-content">Enter a string to replace all occurences of the first letter with \*'s!</p>

            <p id="fix-start-value"></p>

            <input type="text" class="functionInputs" id="fix-start-input">

            <button id="fix-start-submit">Fix Me!</button>

            <button id="fix-start-reset">Reset</button>

        </section>

        <section class="content-section">

            <h2 class="section-header">notBad() Function</h2>

            <p class="inner-content">Finds text between not...bad and replaces with good!</p>

            <p id="not-bad-value"></p>

            <input type="text" class="functionInputs" id="not-bad-input">

            <button id="not-bad-submit">Not Bad!</button>

            <button id="not-bad-reset">Reset</button>

        </section>

    </div>

<script src="../scripts/Assignment2.js"></script>

<script src="../scripts/squarenumber.js"></script>

<script src="../scripts/fixstart.js"></script>

<script src="../scripts/notbad.js"></script>

</body>

</html>

# Assignment2.css

*/\* Name: Wade Moulton*

*Date: 09/15/2020*

*Class: CS 4720*

*Section: W01*

*Instructor: Dr. Sarah North \*/*

\* {

    text-align: center;

}

#cell-div {

    text-align: center;

    margin: 0 auto;

    margin-bottom: 20px;

}

.btn-container {

    text-align: center;

    margin: 0 auto;

}

input {

    width: 238px;

    background-color: rgba(0, 0, 0, 0);

    border: none;

    color: rgba(255, 255, 255, 0.8);

}

input:focus {

    border: 1px solid white;

    outline: none;

    color: #0f4;

}

#input-table {

    margin: 0 auto;

}

.functionInputs{

    background-color: rgba(75, 75, 75, 0.9);

    line-height: 25px;

    border: none;

}

.functionInputs:focus {

    border: 1px solid white;

}

p {

    text-align: center;

}

th {

    line-height: 30px;

    background-color: rgba(70, 70, 70, 0.9);

    border: 1px solid black;

}

td {

    min-width: 75px;

    border: 1px solid black;

    background-color: rgba(75, 75, 75, 0.9);

    line-height: 25px;

}

button {

    border: 1px solid rgba(200, 200, 200, 0.8);

    background: #2c2c2c;

    color: rgba(200, 200, 200, 0.8);

    height: 30px;

    border-radius: 3px;

}

button:hover {

    background: #3c3c3c;

    color: white;

    cursor: pointer;

}

# Squarenumber.js

*// Name: Wade Moulton*

*// Date: 09/15/2020*

*// Class: CS 4720*

*// Section: W01*

*// Instructor: Dr. Sarah North*

const squareSubmit = document.getElementById('square-number-submit-btn');

const squareReset = document.getElementById('square-number-reset-btn');

const squareInput = document.getElementById('square-number-input');

const squareTextBox = document.getElementById('square-number-value');

squareSubmit.addEventListener('click', () => {

    try {

        const flag = Number.isInteger(parseInt(squareInput.value));

        if (flag) {

            squareTextBox.innerHTML = `The result of squaring <strong>${squareInput.value}</strong> is ${squareNumber(squareInput.value)}!`;

        }

        else{

            throw new Error();

        }

    } catch(err) {

        squareTextBox.innerHTML = `Could not square <span class="error">${squareInput.value}</span>. Please make sure to input a number and try again.`

    }

})

squareReset.addEventListener('click', () => {

    squareTextBox.innerHTML = "";

    squareInput.value = "";

})

function squareNumber(num) {

    return Math.pow(num, 2);

}

# Fixstart.js

*// Name: Wade Moulton*

*// Date: 09/15/2020*

*// Class: CS 4720*

*// Section: W01*

*// Instructor: Dr. Sarah North*

const fixSubmit = document.getElementById('fix-start-submit');

const fixInput = document.getElementById('fix-start-input');

const fixReset = document.getElementById('fix-start-reset');

const fixTextBox = document.getElementById('fix-start-value');

fixSubmit.addEventListener('click', () => {

    try {

        fixTextBox.innerHTML = `Your new string is: <strong>${fixStart(fixInput.value)}`;

    } catch(err) {

        fixTextBox.innerHTML = `Oops something went wrong! Try again!`;

    }

})

fixReset.addEventListener('click', () => {

    fixTextBox.innerHTML = "";

    fixInput.value = "";

})

function fixStart(text){

    const first = text[0];

    const end = text.substring(1).replaceAll(first.toUpperCase(), '\*').replaceAll(first.toLowerCase(), '\*');

    return `${first}${end}`;

}

# Notbad.js

*// Name: Wade Moulton*

*// Date: 09/15/2020*

*// Class: CS 4720*

*// Section: W01*

*// Instructor: Dr. Sarah North*

const notBadInput = document.getElementById('not-bad-input');

const notBadSubmit = document.getElementById('not-bad-submit');

const notBadReset = document.getElementById('not-bad-reset');

const notBadTextBox = document.getElementById('not-bad-value');

notBadSubmit.addEventListener('click', () => {

    try {

        notBadTextBox.innerHTML = notBad(notBadInput.value);

    } catch (err) {

        notBadTextBox.innerHTML = `Looks like something went wrong! Try again!`;

    }

})

notBadReset.addEventListener('click', () => {

    notBadInput.value = "";

    notBadTextBox.innerHTML = "";

})

function notBad(text) {

    const notIndex = text.search('not');

    const badIndex = text.search('bad');

    if (-1 < notIndex && notIndex < badIndex) {

        return `${text.substring(0, notIndex)}good${text.substring(badIndex + 3)}`;

    }

    else {

        return text;

    }

}