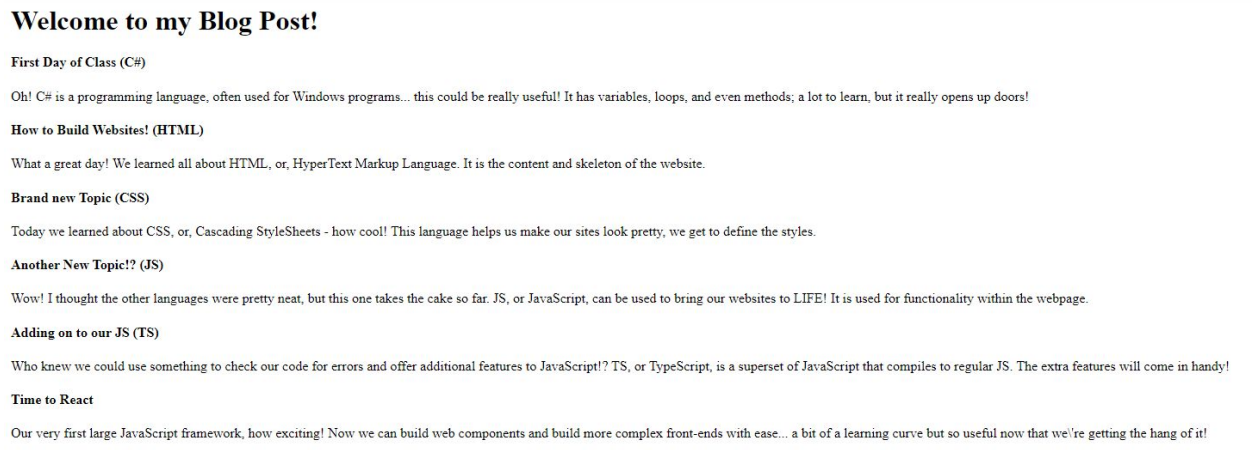
JavaScript Assignment - Blog Post

Deadline: Thursday, JUL 23 2020, 09:00 AM

GitHub Classroom Link REMEMBER TO SET REPO TO PRIVATE UNTIL AFTER DUE DATE



Welcome to my Blog Post!

First Day of Class (C#)

Oh! C# is a programming language, often used for Windows programs... this could be really useful! It has variables, loops, and even methods; a lot to learn, but it really opens up doors!

How to Build Websites! (HTML)

What a great day! We learned all about HTML, or, HyperText Markup Language. It is the content and skeleton of the website.

Brand new Topic (CSS)

Today we learned about CSS, or, Cascading StyleSheets - how cool! This language helps us make our sites look pretty, we get to define the styles.

Another New Topic!? (JS)

Wow! I thought the other languages were pretty neat, but this one takes the cake so far. JS, or JavaScript, can be used to bring our websites to LIFE! It is used for functionality within the webpage.

Adding on to our JS (TS)

Who knew we could use something to check our code for errors and offer additional features to JavaScript!? TS, or TypeScript, is a superset of JavaScript that compiles to regular JS. The extra features will come in handy!

Time to React

Our very first large JavaScript framework, how exciting! Now we can build web components and build more complex front-ends with ease... a bit of a learning curve but so useful now that we\'re getting the hang of it!

Introduction

This assignment is meant to challenge your understanding of DOM manipulation, understanding of a template literal, the use of a for ... of loop, instantiation of an object or object(s) of a class, adding elements to an array, loop through (iterate through) an array, and use of ES6 standards.

Requirements

- ❏ Create a class named “Article” which contains a constructor and output method:
  - ❏ The constructor should contain the fields “title” and “content” to store the title of a blog post and the content of a blog post respectively
  - ❏ Use a for ... of loop along with template literals to output each blog post stored in an array
- ❏ Create an array which stores the “Article” objects
- ❏ Recreate the blog posts in the above example - each post should be its own “Article” object

Challenges

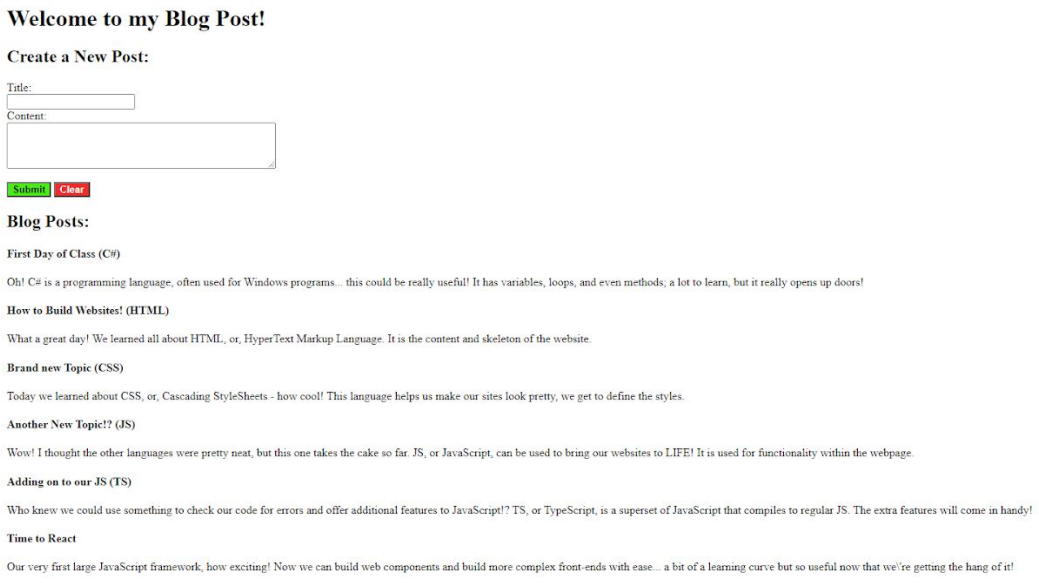
- ❏ Add an option for users to create new blog posts without having to tamper with the code directly
- ❏ Add an option for users to delete existing blog posts
- ❏ Add some CSS to improve user experience

- ❑ Add a blog post date every time a new blog post is made
- ❑ Add a blog post edit feature for each existing blog post

Hints

- Look over what was done in class! It should contain all the tools necessary to get started on this assignment
- Remember that you need an index.html and js/scripts.js file in your project
- Focus on the requirements before moving on to the challenges! Challenges are extra
- Comment your code and commit/push your work regularly

Additional Sample Screenshots



Rubric

You will be evaluated on the following points. You must get all 4 of the Mandatory points to pass:

Requirement	**Points**
MANDATORY: ● User can see all basic blog posts on browser via DOM ● Article class has a constructor with title and content fields ● Article class has an output method ● An array is used to store the Article classes ● Blog posts are recreated as seen in the screenshot	5
CHALLENGE: ● Student has added an add feature ● Student has added a delete feature ● Student has nice CSS ● Student has added timestamps ● Student has added an edit feature	5
Total:	5

Citation Guide

Whenever you borrow code, the following information must be included:

- ❏ Comments to indicate both where the borrowed code begins and ends.
- ❏ A source linking to where you found the code.
- ❏ Your reason for adding the code to your assignment/project instead of writing it out yourself
- ❏ How it works. Explain to us how the code is supposed to work, include links to documentation/articles you read to help you understand.
- ❏ A small demonstration to prove you understand how the code works.

```
const inputArr = [5,1,3,4,2];

/*Borrowed code for bubbleSort starts*/
let bubbleSort = (inputArr) => {
    let len = inputArr.length;
    for (let i = 0; i < len; i++) {
        for (let j = 0; j < len; j++) {
            if (inputArr[j] > inputArr[j + 1]) {
                let tmp = inputArr[j];
                inputArr[j] = inputArr[j + 1];
                inputArr[j + 1] = tmp;
            }
        }
    }
    return inputArr;
};

/*Borrowed code from bubbleSort ends*/

//Source: bubbleSort function obtained from https://medium.com/javascript-algorithms/javascript-algorithms-bubble-sort-3d27f285c3b2
//Reason to add: implementing bubble sort can be tedious and bug prone, it would be better to use a proven version than to write my ow
//How it works: I read the following article to understand how bubble sorts work (http://www.pkirs.utep.edu/CIS3355/Tutorials/chapter9
//Demonstration of understanding:
//Example array: [3,1,2]
//Step 1: Compare 3 and 1. Since 1 is smaller, swap places.
//Array: [1,3,2]
//Step 2: Compare 3 and 2. Since 2 is smaller, swap places.
//Array: [1,2,3]
//Step 3: Compare 1 and 2. No need to swap.
//Array: [1,2,3]
//Step 4: Compare 2 and 3. No need to swap.
//Array: [1,2,3]
//Function complete.
console.log(bubbleSort(inputArr));
```