PHP To-Dos Assignment

Deadline: Thursday, AUG 13 2020, 09:00 AM
GitHub Classroom Link:

Introduction

Create a basic "To-Dos" app using PHP to demonstrate knowledge of form submission handling, array manipulation, and sessions.

Requirements

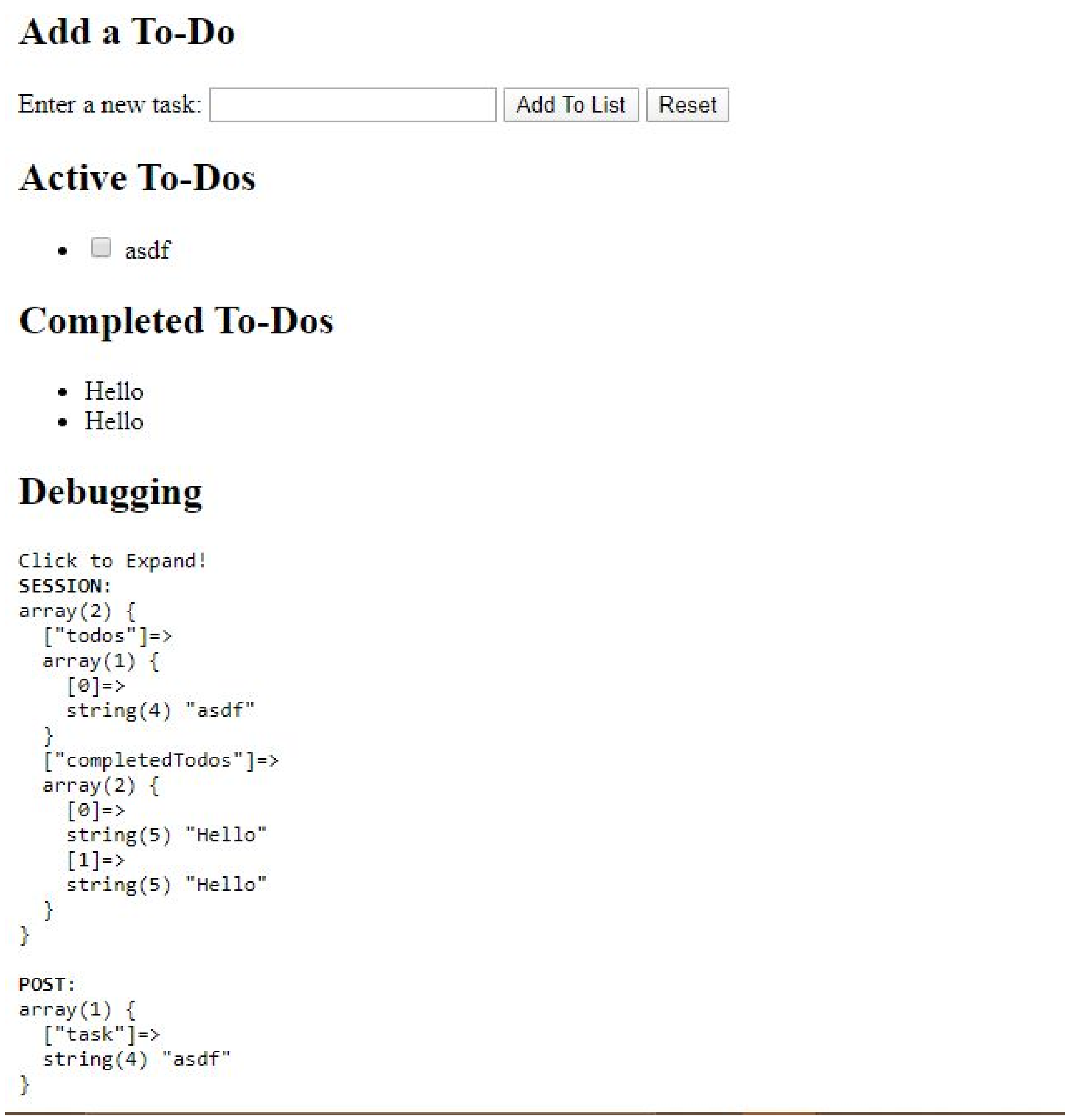
- ❏ PHP Variables
  - ❏ Declare variables
  - ❏ Assign values to variables
- ❏ Error-free PHP
  - ❏ No errors in developer tools JavaScript console while using the app
  - ❏ No errors in the PHP server terminal while using the app
  - ❏ Passes validator (https://phpcodechecker.com/)
- ❏ Use includes to help organize your code and application
- ❏ PHP Arrays
  - ❏ Create an array
  - ❏ Manipulate the array
  - ❏ Loop through an array and display values on screen
- ❏ <FORM> submission handling in PHP
  - ❏ Appropriate use of the "action" attribute
  - ❏ Appropriate use of the "method" attribute
- ❏ PHP Global Variables
  - ❏ Capture form submission details using $_POST
  - ❏ Store data in $_SESSION
  - ❏ Retrieve data from $_SESSION

Challenges

- ❏ Make it look nice with some CSS!
- ❏ Validate the form field(s)
- ❏ Split the list into two arrays (**the basic To-Do app could just use "Active"**):
  - ❏ Active
  - ❏ Completed
- ❏ Reset Button (Clears All To-Dos / Session)

Sample Screenshot

This is an image of what we want the to-do list to look like. If you can make it work and get all of the requirements then you can play with the css and make it pretty. But only after all requirements are done properly.



Add a To-Do

Enter a new task: [   ] Add To List Reset

Active To-Dos

- ☐ asdf

Completed To-Dos

- Hello
- Hello

Debugging

```
Click to Expand!
SESSION:
array(2) {
  ["todos"]=>
  array(1) {
    [0]=>
    string(4) "asdf"
  }
  ["completedTodos"]=>
  array(2) {
    [0]=>
    string(5) "Hello"
    [1]=>
    string(5) "Hello"
  }
}

POST:
array(1) {
  ["task"]=>
  string(4) "asdf"
}
```

Rubric

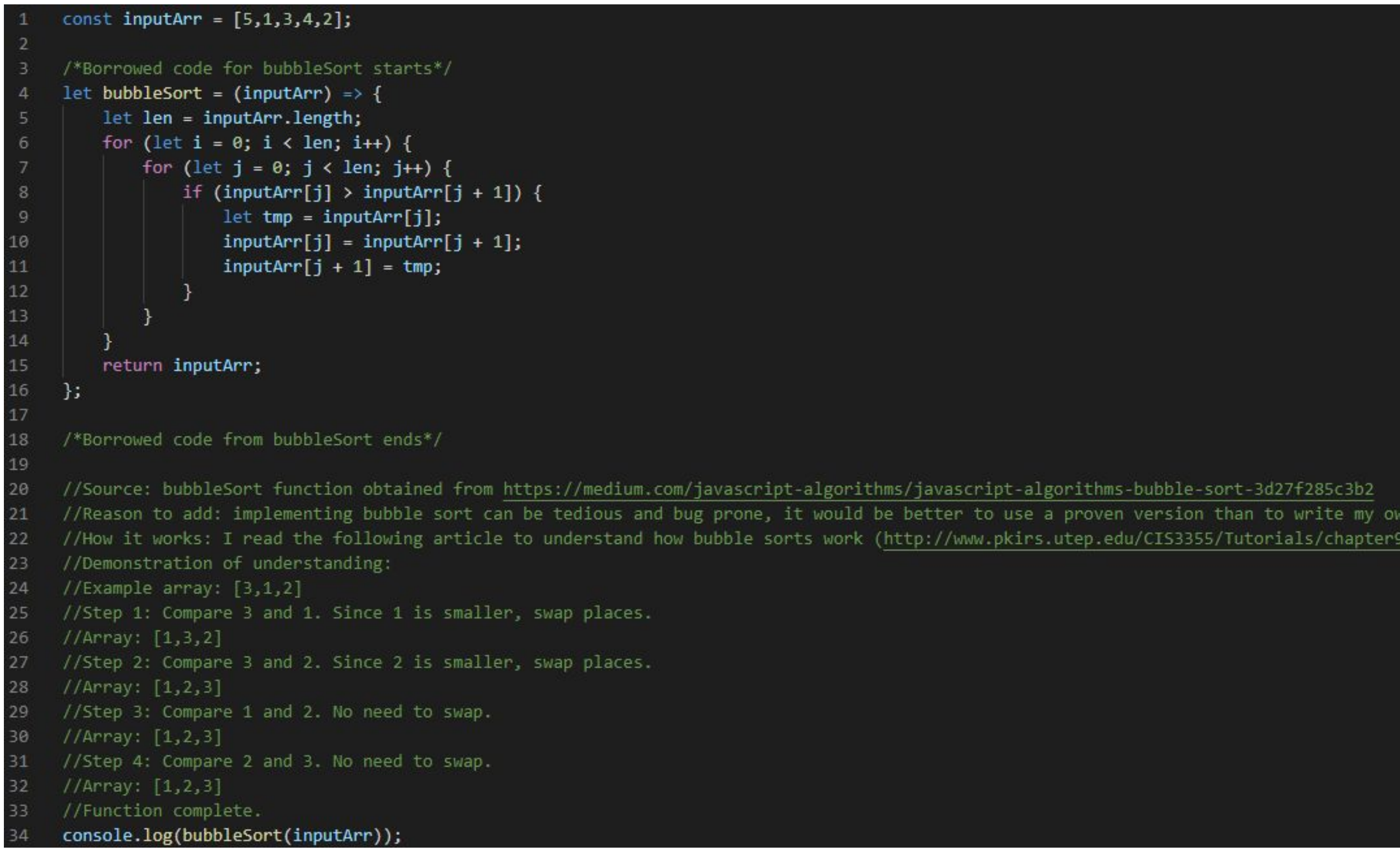
You will be evaluated on the following points. You must get all 6 of the Mandatory points to pass:

Requirement	**Points**
MANDATORY: ● PHP Variables o Declare variables o Assign values to variables ● Error-free PHP o No errors in developer tools JavaScript console while using the app o No errors in the PHP server terminal while using the app o Passes validator (https://phpcodechecker.com/) ● Use includes to help organize your code and application ● PHP Arrays o Create an array o Manipulate the array o Loop through an array and display values on screen ● <FORM> submission handling in PHP o Appropriate use of the “action” attribute o Appropriate use of the “method” attribute ● PHP Global Variables o Capture form submission details using $_POST o Store data in $_SESSION o Retrieve data from $_SESSION	6
CHALLENGE: ● Make it look nice with some CSS! ● Validate the form field(s) ● Split the list into two arrays (the basic To-Do app could just use “Active”): o Active o Completed ● Reset Button (Clears All To-Dos / Session)	4
Total:	10

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));
```