

Intro to JavaScript Week 5 Coding Assignment

Points possible: 70

Category	Criteria	% of Grade
Functionality	Does the code work?	25
Organization	Is the code clean and organized? Proper use of white space, syntax, and consistency are utilized. Names and comments are concise and clear.	25
Creativity	Student solved the problems presented in the assignment using creativity and out of the box thinking.	25
Completeness	All requirements of the assignment are complete.	25

Instructions: In VS Code, or an IDE of your choice, write the code that accomplishes the objectives listed below. Ensure that the code compiles and runs as directed. Take screenshots of the code and of the running program (make sure to get screenshots of all required functionality) and paste them in this document where instructed below. Create a new repository on GitHub for this week's assignments and push this document, with your JavaScript project code, to the repository. Add the URL for this week's repository to this document where instructed and submit this document to your instructor when complete.

Coding Steps:

- 1. Create a menu app as seen in this week's video. What you create is up to you as long as it meets the following requirements.
 - **a.** Use at least one array.
 - **b.** Use at least two classes.
 - **c.** Your menu should have the options to create, view, and delete elements.

Screenshots of Code:

</>>

```
constructor(flavor, filling) {
    this.flavor = flavor;
2
3
4
5
6
7
8
9
               this.filling = filling;
          describe() {
    return `The cupcake you chose is ${this.flavor} with a ${this.filling} filling.`;
10
11
      class DessertTable {
12
           constructor(name) {
13
              this.name = name;
14
               this.cupcakes = [];
15
16
17
           addCupcake(cupcake) {
18
               if (cupcake instanceof Cupcake) {
19
20
                  this.cupcakes.push(cupcake);
               } else {
21
                   throw new Error (`You can only add an instance of Cupcake. Argument is not a cupcake: ${cupcake}.`);
22
23
24
          describe() {
    return `The ${this.name} dessert table has ${this.cupcakes.length} kinds of cupcakes.`;
25
26
27
28
29
```

```
class Menu {
30
31
         constructor () {
32
             this.dessertTable = [];
             this.selectedDessertTable = null;
33
34
35
36
         start() {
37
              let selection = this.showMainMenuOptions();
38
39
             while (selection !=0) {
                  switch (selection) {
40
41
                      case '1':
42
                          this.createDessertTable();
43
                          break;
44
                      case '2':
                          this.viewDessertTable();
45
46
                          break:
47
                      case '3':
48
                          this.displayDessertTable();
49
                      case '4':
50
51
                          this.overturnDessertTable();
52
                          break;
53
                      default:
54
                          selection = 0;
55
56
                  selection = this.showMainMenuOptions();
57
              alert ('Your selection is invalid. Try again.')
58
59
60
```

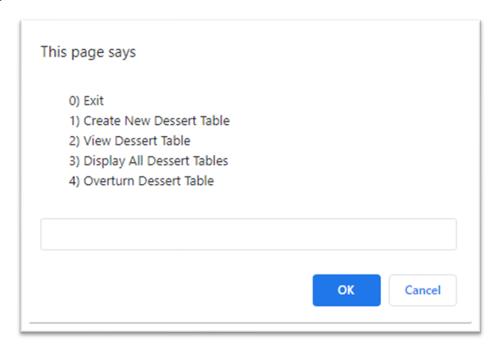
```
showMainMenuOptions() {
61
62
              return prompt(`
63
              0) Exit
64
              1) Create New Dessert Table
65
              2) View Dessert Table
              3) Display All Dessert Tables
66
67
              4) Overturn Dessert Table
68
              `);
69
70
71
          showDessertTableMenuOptions(dessertTableInfo) {
72
              return prompt(`
73
              0) back
74
              1) Create cupcake
75
              2) Eat a cupcake
76
77
              ${dessertTableInfo}
78
              `);
79
80
81
         displayDessertTable() {
82
              let dessertTableString = '';
83
              for (let i = 0; i < this.dessertTable.length; i++) {</pre>
84
                  dessertTableString += i + ')' + this.dessertTable[i].name + '\n';
85
86
              alert(dessertTableString);
87
88
          createDessertTable() {
89
              let name = prompt('Give your dessert table the name of a color.');
90
              this.dessertTable.push(new DessertTable(name));
91
92
93
```

```
122
123
124
          createCupcake() {
125
               let flavor = prompt ('What flavor of cupcake would you like to make?');
126
              let filling= prompt ('What type of filling would you like for your cupcake?');
              this.selectedDessertTable.cupcakes.push(new Cupcake(flavor, filling));
127
128
129
130
          eatCupcake() {
131
              let index = prompt ('What cupcake would you like to devour with your pie-hole?')
132
              if (index > -1 && index < this.selectedDessertTable.cupcakes.length) {</pre>
133
                   this.selectedDessertTable.cupcakes.splice(index, 1);
134
135
136
137
138
       let menu = new Menu();
139
      menu.start();
```

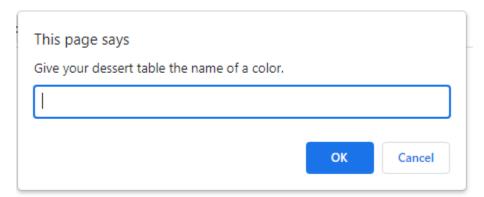


Screenshots of Running Application:

Menu page

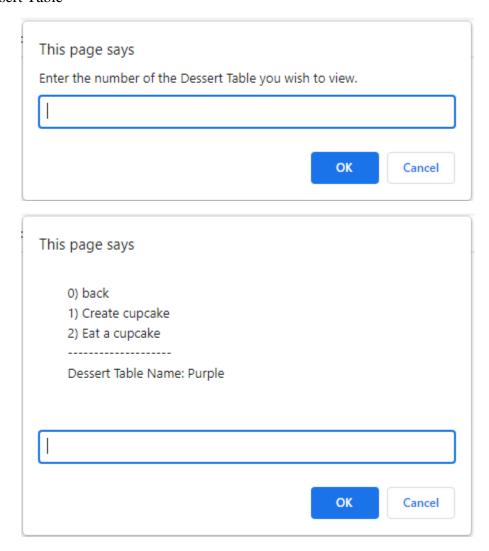


Create New Dessert Table

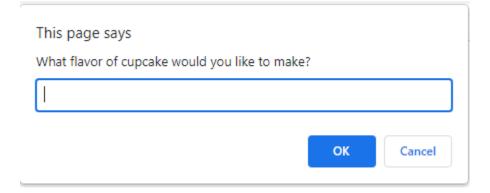




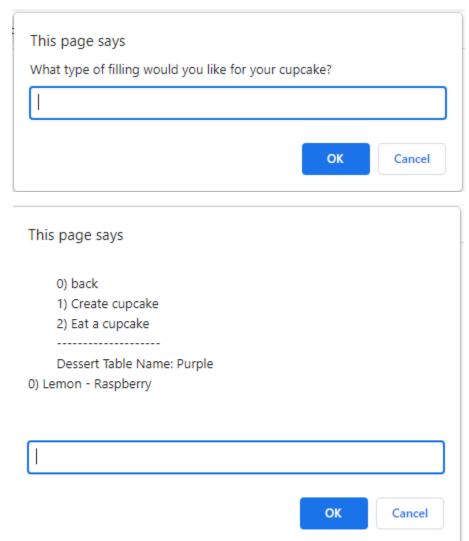
View Dessert Table



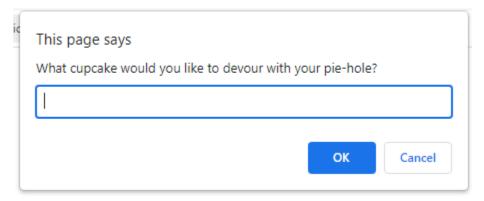
Create cupcake



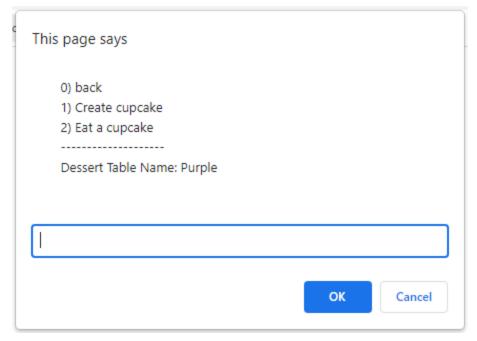




Eat a cupcake







Display All Dessert Tables





Overturn Dessert Table

This page says What dessert table do you want to overte	urn in rage?	
	ОК	Cancel
This page says 0)B 1)Red 2)Green 3)Blue		
		ОК

URL to GitHub Repository:

https://github.com/ramirezm19/Week5.git