Batch #19 / Back-End Class

Remote Learning Assignment - Week 2

Assignment 1: Function and Array

1. max: find the max value of an array of numbers
2. findPosition: find the first position of the target number inside an array of numbers. The position should be counted starting from 0, if you can't find the target, please return -1

Reminder: you cannot use those built-in functions like Math.max() and Array.prototype.findIndex() to complete this assignment, please implement it by yourself.

```
function max(numbers) {
 // your code here, for-loop method preferred
}

function findPosition(numbers, target) {
 // your code here, for-loop method preferred
}
console.log(max([1, 2, 4, 5])); // should print 5
console.log(max([5, 2, 7, 1, 6])); // should print 7

console.log(findPosition([5, 2, 7, 1, 6], 5)); // should print 0
console.log(findPosition([5, 2, 7, 1, 6], 7)); // should print 2
console.log(findPosition([5, 2, 7, 7, 7, 1, 6], 7)); // should print
2 (the first position)
console.log(findPosition([5, 2, 7, 1, 6], 8)); // should print -1
```

Assignment 2: Function, Array, and Object

Complete the function below to calculate the average price of all the products.

```
function avg(data) {
 // your code here
}

console.log(
 avg({
   size: 3,
   products: [
     {
       name: 'Product 1',
       price: 100,
     },
     {
       name: 'Product 2',
       price: 700,
     },
     {
       name: 'Product 3',
       price: 250,
     },
   ],
 })
); // should print the average price of all products
```

Assignment 3: Data Manipulation

1. count: return an object which shows the count of each character.
2. groupByKey: return an object which shows the summed-up value of each key.

This time, you may get letters from ‘a’ to ‘z’, try to avoid using ‘if’ or ‘switch’ to split each letter into different cases *(e.g. if(letter == ‘a’) {…} else if (letter == ‘b’) {…})*, otherwise, your code will be very long.

Note:

1. The input format is different for these two functions.
2. In the second function, the input may have the same key but different values, the output should have each key only once.

```
function count(input) {
 // your code here
}
let input1 = ["a", "b", "c", "a", "c", "a", "x"];
console.log(count(input1));
// should print {a:3, b:1, c:2, x:1}

function groupByKey(input) {
 // your code here
}

let input2 = [
 { key: "a", value: 3 },
 { key: "b", value: 1 },
 { key: "c", value: 2 },
 { key: "a", value: 3 },
 { key: "c", value: 5 },
];

console.log(groupByKey(input2));
// should print {a:6, b:1, c:7}
```

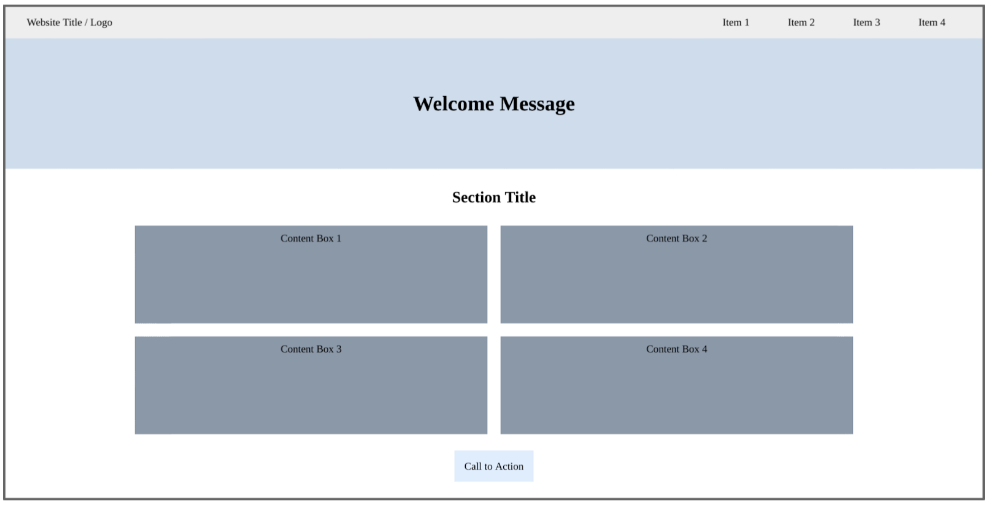
Batch #19 / Back-End Class

Remote Learning Assignment - Week 2

Assignment 4: HTML DOM and Event Handling

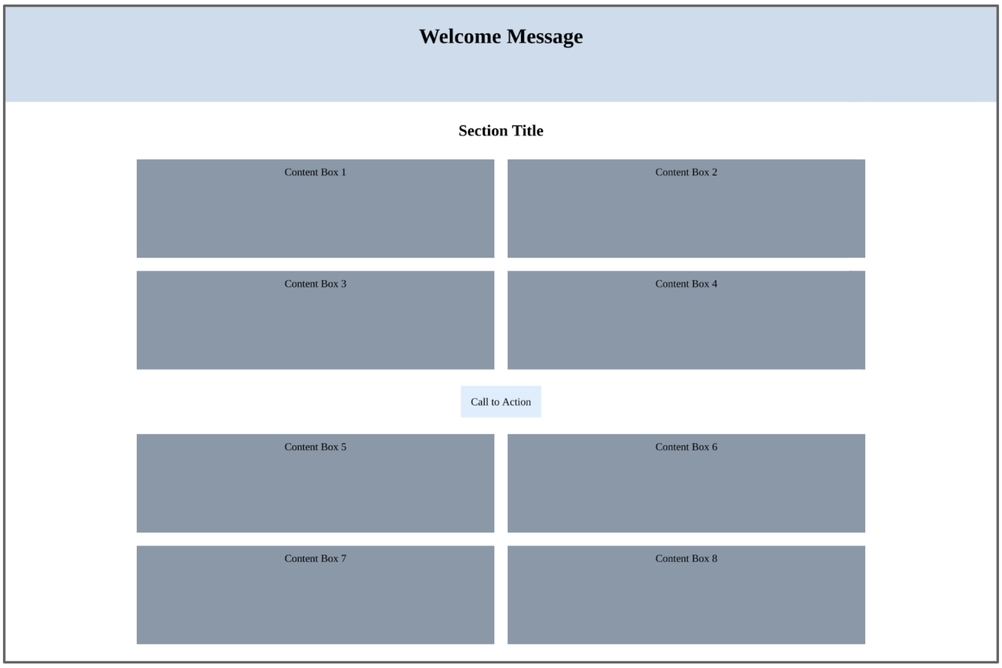
Following the assignment in week 1, let’s add some effects on it by only pure JavaScript without Bootstrap, jQuery, or any other libraries.

Request 1: Click to Change Text.
When the user clicks on the "Welcome Message" block, change the text to "Have a Good Time!".



Request 2: Click to Show More Content Boxes.
There are some more content boxes waiting to show. When the user clicks the Call-to-Action button, show those hidden content boxes.

Hint: all content boxes are already there, they are just set to display: none at the beginning.



Assignment 5: Algorithm Practice (Advanced Optional)

Remember what we did in Assignment 1? We created a function that can find the position of the target number inside an array of numbers. Actually, if the array was **Sorted** already, there is a beautiful algorithm called **Binary Search** which can do this job efficiently. You can try to look up these keywords and learn the concept behind this algorithm. If you still have time, you can try implementing it yourself.

For simplicity, you can assume that there are no duplicate numbers in the given array. It will be a challenge if you haven't learned any algorithm before.

```
function binarySearchPosition(numbers, target) {
 // your code here
}

console.log(binarySearchPosition([1, 2, 5, 6, 7], 1)); // should print 0
console.log(binarySearchPosition([1, 2, 5, 6, 7], 6)); // should print 3
```

Assignment 6: HTML DOM and Event Handling with jQuery (Advanced Optional)

Remember what we did in Assignment 4? For this assignment, try redoing assignment 4 again, but using jQuery.

Note: jQuery is a simple, easy-to-learn, and feature-rich JavaScript library. After you have a basic knowledge of HTML, CSS, and JavaScript, jQuery can help you “Write Less, Do More”. Even though jQuery is not as cool anymore, it’s still a great tool for back-end engineers to help us with our front-end development. So, it is recommended that you complete this assignment, it will be useful for your future projects. If you want to learn any other front-end framework or tool, please open your mind and discuss it with your instructor.