Batch #12 / Backend Class

Remote Learning Assignment - Week 2

Assignment 1: Function and Array

Complete the following functions

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

Complete the following functions

1. count: return an object which shows the count of each characters.
2. groupByKey: return an object which shows the summed up value of each keys.

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 same key but different values, the output should have each key only once.

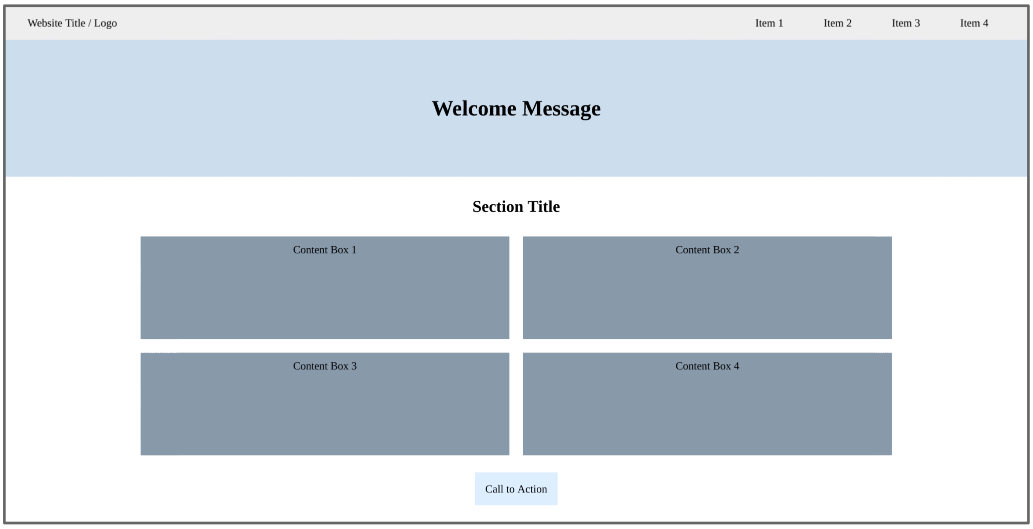
```
function count(input) {
        // your code here
}

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}
```

Assignment 4: HTML DOM and Event Handling

Following 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.

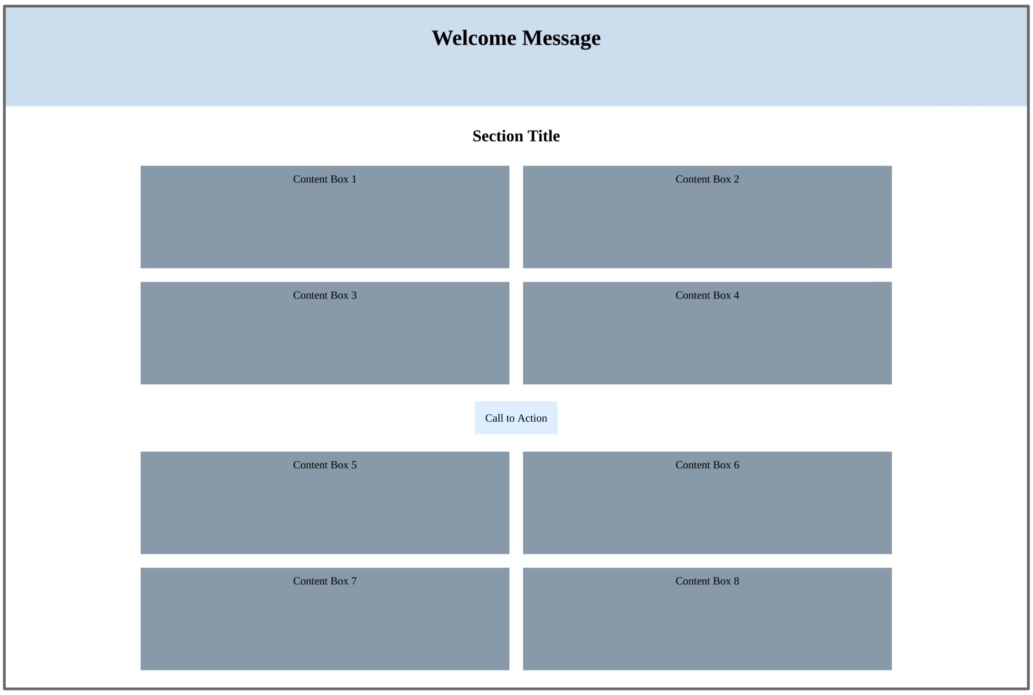
When user click on "Welcome Message" block, change text to "Have a Good Time!".



Request 2: Click to Show More Content Boxes.

There are some more content boxes waiting to show. When 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 which 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 lookup these keywords and learn the concept behind this algorithm. If you still have time, you can try to implement it by 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
```