

JOHNATHAN SANTIAGO

Implement code for the following JavaScript functions, and be sure to use "use strict";

1. Define a function max() that takes two numbers as arguments and returns the largest of them. Use the if-then-else construct available in Javascript.

```
function max(n1,n2){  
  if(n1 > n2) console.log(n1);  
  else console.log(n2);  
}
```

2. Define a function maxOfThree() that takes three numbers as arguments and returns the largest of them.

```
function maxOfThree(n1,n2, n3){  
  if(n1 > n2 && n1 > n3) console.log(n1);  
  else if(n2 > n1 && n2 > n3) console.log(n2);  
  else console.log(n3);  
}
```

3. Write a function isVowel() that takes a character (i.e. a string of length 1) and returns true if it is a vowel, false otherwise.

```
function isVowel(x){  
  var listOfVowels = "aAeEiIoOuU";  
  console.log(x.length ==1 && listOfVowels.indexOf(x) > -1);  
}
```

4. Define a function sum() and a function multiply() that sums and multiplies (respectively) all the numbers in an input array of numbers. For example, sum([1,2,3,4]) should return 10, and multiply([1,2,3,4]) should return 24. Note/Hint: Do these using Imperative programming approach (i.e. for...loop or while...loop)

```
function sum(array) {  
  var result = 0;  
  for (let index = 0; index < array.length; index++) {  
    result += array[index];  
  }  
  console.log(result);  
}  
  
function multiply(array) {  
  var result = null;  
  if(array.length >= 1){  
    result = array[0];  
    for (let index = 1; index < array.length; index++) {
```

```

        result *= array[index];
    }
}
console.log(result);
}

```

- Define a function `reverse()` that computes the reversal of a string. For example, `reverse("jag testar")` should return the string "ratset gaj".

```

function reverse(string) {
    var result = '';
    for (let index = string.length - 1; index >= 0; index--) {
        result += string[index];
    }
    console.log(result);
}

```

- Write a function `findLongestWord()` that takes an array of words and returns the length of the longest one.

```

function findLongestWord(array) {
    var longest = 0;
    for (let index = 0; index < array.length; index++) {
        if (longest < array[index].length) longest = array[index].length;
    }
    console.log(longest);
}

```

- Write a function `filterLongWords()` that takes an array of words and an integer `i` and returns a new array containing only those words that were longer than `i` characters.

```

function filterLongWords(array, wordSize) {
    var newArray = [];
    for (let index = 0; index < array.length; index++) {
        if (array[index].length >= wordSize) newArray[newArray.length++] = array[index];
    }
    console.log(newArray);
}

```

- Write a function named, `computeSumOfSquares`, that takes as input, an array of numbers and calculates and returns the sum of the squares of each number in the input array. E.g. `computeSumOfSquares([1,2,3])` should be computed as $1^2 + 2^2 + 3^2 = 14$. Note: Write your Javascript code without using Imperative programming. i.e. Do NOT use any explicit looping construct; instead use functional programming style/approach.

```

function computeSumOfSquares(array) {
    var result = array.reduce((x, y) => x + Math.pow(y, 2));
    console.log(result);
}

```

9. Write a function named, `printOddNumbersOnly`, that takes as input, an array of integral numbers and it finds and prints only the numbers which are odd.

```
function printOddNumbersOnly(array) {  
  console.log(array.filter(function (element, index, array) {  
    return element % 2 !== 0;  
  }));  
}
```

10. Write a function named, `computeSumOfSquaresOfEvensOnly`, that takes as input, an array of integral numbers and calculates and returns the sum of the squares of only the even numbers in the input array. E.g. `computeSumOfSquaresOfEvensOnly ([1,2,3,4,5])` should be computed as $2^2 + 4^2 = 20$.

```
function computeSumOfSquaresOfEvensOnly(array) {  
  console.log(  
    array.filter(element=>element % 2 === 0).map(e=>e*e).reduce((x,y) => x + y)  
  );  
}
```

11. Using the `Array.reduce(...)` function, re-implement your functions, `sum(...)` and `multiply(...)` (defined in Problem 4 above) without using Imperative programming. i.e. Do NOT use any explicit looping construct; instead use functional programming style/approach.

```
function sumFunctionalWay(array) {  
  var result = array.reduce((x,y) => x + y);  
  console.log(result);  
}  
  
function multiplyFunctionalWay(array) {  
  var result = array.reduce((x,y) => x * y);  
  console.log(result);  
}
```

12. Implement Javascript code for a function named, **findSecondBiggest**, which takes as input, an array of numbers and finds and returns the second biggest of the numbers. For example, `findSecondBiggest([1,2,3,4,5])` should return 4. And `findSecondBiggest([19,9,11,0,12])` should return 12. (Note: Do not use sorting!)

```

function findSecondBiggest(array) {
    var biggest, secondBiggest;
    if(array.length < 2){
        console.log('invalid size!!!');
        return;
    }

    biggest = array[0];
    secondBiggest = array[1];

    if(biggest < secondBiggest) {
        biggest = secondBiggest;
        secondBiggest = biggest;
    }

    for (let index = 2; index < array.length; index++) {
        if(array[index] > biggest){
            secondBiggest = biggest;
            biggest = array[index];
        }else if(array[index] > secondBiggest){
            secondBiggest = array[index];
        }
    }

    console.log(secondBiggest);
}

```

13. Write a function named `printFibo`, that takes as input, a given length, `n`, and any two starting numbers `a` and `b`, and it prints-out the Fibonacci sequence, e.g. (0, 1, 1, 2, 3, 5, 8, 13, 21, 34,...) of the given length, beginning with `a` and `b`. (e.g. `printFibo(n=1, a=0, b=1)`, prints-out: "0", as output; `printFibo(n=2, a=0, b=1)`, prints-out: "0, 1", as output; `printFibo(n=3, a=0, b=1)`, prints-out: "0, 1, 1", as output; `printFibo(n=6, a=0, b=1)`, prints-out: "0, 1, 1, 2, 3, 5", as output; and `printFibo(n=10, a=0, b=1)`, prints-out: "0, 1, 1, 2, 3, 5, 8, 13, 21, 34", as output).

```

function printFibo(n,a,b){
    let counter =0; let out = [a,b];
    while(counter < n-2){
        out.push(out[counter]+out[counter+1]);
        counter++;
    }
    console.log(out);
}

```

14. Refer to your work on Lab Assignment 4. Add Javascript code to work with your 2 HTML forms as follows:

- a. Login Form: Add code such that when the Submit button is clicked, the values entered in the input fields are printed to the Console.

See link: Lab 5 on <https://johnsales.github.io/MIU/WAP/w1d4/webform2.html>

- b. New Product Form: Add code such that when the Submit button is clicked, the values entered in the input fields are displayed in a pop-up window.

See link: Lab 5 on <https://johnsales.github.io/MIU/WAP/w1d4/webform2.html>

15. Using JavaScript and HTML and CSS, implement a webpage that displays a working, ticking counter Clock, that counts/displays the current Date and time of the browser host, in the format: 2019-11-4 12:16:01

See link: Lab 5 on <https://johnsales.github.io/MIU/WAP/lab5/index.html>

Please submit your code as a single zip file attachment to Sakai and also push it to your github repository.

//-- Enjoy! --//