Write a JavaScript program to find all the index positions of a given word within a given string.

```
function findAllIndexes(str, word) { let indexes = [];

let index = -1;

while ((index = str.indexOf(word, index + 1)) !== -1) { indexes.push(index);

}

return indexes;

}

let string = "hello world hello"; let word = "world";

let positions = findAllIndexes(string, word);

console.log("The word '" + word + "' appears at positions: " + positions.join(", "));
```

Write a JavaScript program to find the first index of a given element in an array using the linear search algorithm.

```
function linearSearch(arr, element) { for (let i = 0; i < arr.length; i++) {

if (arr[i] === element) {

return i;

}

}

return -1;

}
```

Write a JavaScript program to sort a list of elements using Quick sort.

```
function quickSort(arr) {

if (arr.length <= 1) { return arr;

}

const pivot = arr[Math.floor(arr.length / 2)]; const left = [];
```

```javascript
const right = [];

for (let i = 0; i < arr.length; i++) {

if (i === Math.floor(arr.length / 2)) { continue;

}

if (arr[i] < pivot) { left.push(arr[i]);

} else { right.push(arr[i]);

}

}

return [...quickSort(left), pivot, ...quickSort(right)];

}

// Example usage:

const arr = [5, 3, 7, 2, 8, 4, 1]; const sortedArr = quickSort(arr);

console.log("Sorted array:", sortedArr);
```

Write a JavaScript program to sort a list of elements using Merge sort.

```javascript
function merge_Arrays(left_sub_array, right_sub_array) { let array = []

while (left_sub_array.length && right_sub_array.length) { if (left_sub_array[0] < right_sub_array[0]) {
array.push(left_sub_array.shift())

} else { array.push(right_sub_array.shift())

}

}

return [ ...array, ...left_sub_array, ...right_sub_array ]

}

function merge_sort(unsorted_Array) {

const middle_index = unsorted_Array.length / 2 if(unsorted_Array.length < 2) {

return unsorted_Array

}

const left_sub_array = unsorted_Array.splice(0, middle_index)

return merge_Arrays(merge_sort(left_sub_array),merge_sort(unsorted_Array))

}

unsorted_Array = [39, 28, 44, 4, 10, 83, 11];
```

```
console.log("The sorted array will be: ",merge_sort(unsorted_Array));
```

Write a JavaScript program to sort a list of elements using Heap sort.

```
function customSort(arr) { var N = arr.length;

for (var i = Math.floor(N / 2) - 1; i >= 0; i--) customHeapify(arr, N, i);

for (var i = N - 1; i > 0; i--) { var temp = arr[0];

arr[0] = arr[i]; arr[i] = temp;

customHeapify(arr, i, 0);

}

}

function customHeapify(arr, N, i) { var largest = i;

var l = 2 * i + 1; var r = 2 * i + 2;

if (l < N && arr[l] > arr[largest]) largest = l;

if (r < N && arr[r] > arr[largest]) largest = r;

if (largest != i) { var swap = arr[i]; arr[i] = arr[largest];

arr[largest] = swap; customHeapify(arr, N, largest);

}

}

function customPrintArray(arr) { var N = arr.length;

for (var i = 0; i < N; ++i) console.log(arr[i]);

}

var originalArr = [12, 11, 13, 5, 6, 7]; customSort(originalArr); console.log("Sorted array:" + originalArr);
```

Write a JavaScript program to sort a list of elements using Insertion sort.

```
function insertionSort(arr) { const n = arr.length;

for (let i = 1; i < n; i++) { let key = arr[i];

let j = i - 1;

while (j >= 0 && arr[j] > key) { arr[j + 1] = arr[j];

j--;

}
```

```
arr[j + 1] = key;

}

return arr;

}

// Example usage:

const originalArr = [5, 3, 7, 2, 8, 4, 1]; const sortedArr = insertionSort(originalArr); console.log("Sorted array:", sortedArr);
```

Write a JavaScript program to sort a list of elements using Bubble sort

```
function bubbleSort(arr) { const n = arr.length;

for (let i = 0; i < n - 1; i++) { for (let j = 0; j < n - i - 1; j++) { if (arr[j] > arr[j + 1]) {

// Swap arr[j] and arr[j+1] let temp = arr[j];

arr[j] = arr[j + 1]; arr[j + 1] = temp;

}

}

}

return arr;

}

// Example usage:

const arr = [5, 3, 7, 2, 8, 4, 1]; const sortedArr = bubbleSort(arr);

console.log("Sorted array:", sortedArr);
```

Write a JavaScript program to sort the characters in a string alphabetically.

```
function sortStringAlphabetically(str) { return str.split('').sort().join('');

}

const inputString = "hello world";

const sortedString = sortStringAlphabetically(inputString); console.log("Original string:", inputString); console.log("Sorted string:", sortedString);
```

Write a JavaScript program to check if a numeric array is sorted or not.

```
function isArraySorted(arr) {

for (let i = 0; i < arr.length - 1; i++) { if (arr[i] > arr[i + 1]) {

return false;

}

}

return true;

}

// Example usage:

const sortedArray = [1, 2, 3, 4, 5];

const unsortedArray = [5, 3, 7, 2, 8];

console.log("Is sortedArray sorted?", isArraySorted(sortedArray)); console.log("Is unsortedArray sorted?",
isArraySorted(unsortedArray));
```

Write a JavaScript function to validate whether a given value type is null or not.

```
function isNull(value) { return value === null;

}

// Example usage: console.log(isNull(null)); // true console.log(isNull(5)); // false
```

Write a JavaScript function to validate whether a given value is a number or not.

```
function isNumber(value) {

return typeof value === 'number' && !isNaN(value);

}

// Example usage:

console.log(isNumber(5)); // true console.log(isNumber("hello")); // false
```

Write a JavaScript function to validate whether a given value is RegExp or not.

```
function isRegExp(value) {

return Object.prototype.toString.call(value) === '[object RegExp]';

}
```

```
// Example usage:
console.log(isRegExp(/test/)); // true console.log(isRegExp("hello")); // false
```

Write a JavaScript program to delete the rollno property from the following object. Also print the object before or after deleting the property.

Sample object:

var student = {

name : "David Rayy", sclass : "VI",

rollno : 12 };

```
var student = {

name: "David Rayy", sclass: "VI",

rollno: 12

};

console.log("Object before deleting rollno property:", student);

delete student.rollno;

console.log("Object after deleting rollno property:", student);
```

Write a JavaScript program to display the reading status (i.e. display book name, author name and reading status) of the following books.

```
var library = [

{

author: 'Bill Gates', title: 'The Road Ahead', readingStatus: true

},

{

author: 'Steve Jobs', title: 'Walter Isaacson', readingStatus: true

},

{

author: 'Suzanne Collins',

title: 'Mockingjay: The Final Book of The Hunger Games', readingStatus: false
```

```javascript
}];

var library = [

{

author: 'Bill Gates', title: 'The Road Ahead', readingStatus: true

},

{

author: 'Steve Jobs', title: 'Walter Isaacson', readingStatus: true

},

{

author: 'Suzanne Collins',

title: 'Mockingjay: The Final Book of The Hunger Games', readingStatus: false

}

];

for (var i = 0; i < library.length; i++) { var book = library[i];

var bookInfo = "" + book.title + " by " + book.author;

if (book.readingStatus) {

console.log("You have already read " + bookInfo + ".");

} else {

console.log("You haven't read " + bookInfo + " yet.");

}

}
```

-
  1. **Write a JavaScript program to create a clock. Note: The output will come every second. Expected Console Output :**

"14:37:42"

"14:37:43"

"14:37:44"

"14:37:45"

"14:37:46"

"14:37:47"

```javascript
function displayTime() {

var date = new Date();
```

```javascript
var hours = formatTime(date.getHours());

var minutes = formatTime(date.getMinutes()); var seconds = formatTime(date.getSeconds());

console.log(hours + ":" + minutes + ":" + seconds);

}

function formatTime(time) {

return time < 10 ? "0" + time : time;

}

setInterval(displayTime, 1000);
```

Write a JavaScript function to parse an URL.

```javascript
function parseURL(url) {

var parser = document.createElement('a'); parser.href = url;

return {

protocol: parser.protocol, hostname: parser.hostname, port: parser.port,

pathname: parser.pathname, search: parser.search,

hash: parser.hash, origin: parser.origin

};

}
// Example usage:

var url = "https://www.example.com:8080/path/to/page?key1=value1&key2=value2#section1"; var parsedURL = parseURL(url);

console.log(parsedURL);
```

Write a JavaScript function to split a string and convert it into an array of words

```javascript
function splitStringIntoWords(str) {

// Use the split method to split the string into an array of words return str.split(/\s+/);

}
// Example usage:

var sentence = "This is a sample sentence.";

var wordsArray = splitStringIntoWords(sentence); console.log(wordsArray);
```

Write a JavaScript function that takes a string with both lowercase and upper case letters as a parameter. It converts upper case letters to lower case, and lower case letters to upper case.

```javascript
function swapCase(str) {

var swapped = "";

for (var i = 0; i < str.length; i++) { var char = str[i];

if (char === char.toUpperCase()) { swapped += char.toLowerCase();

} else {

swapped += char.toUpperCase();

}

}

return swapped;

}

// Example usage:

var inputString = "Hello World";

var swappedString = swapCase(inputString); console.log("Original string:", inputString); console.log("Swapped string:", swappedString);
```

Write a JavaScript function that returns the number of minutes in hours and minutes.

Input :

console.log(timeConvert(200)); Output :

"200 minutes = 3 hour(s) and 20 minute(s)."

```javascript
function timeConvert(minutes) {

var hours = Math.floor(minutes / 60); var remainingMinutes = minutes % 60;

return minutes + " minutes = " + hours + " hour(s) and " + remainingMinutes + " minute(s).";

}

// Example usage: console.log(timeConvert(200));
```

Write a JavaScript program to implement a stack that

checks if a given element is present or not in the stack.

```javascript
class Stack {

constructor() { this.items = [];

}

push(element) { this.items.push(element);

}

search(element) {

return this.items.includes(element);

}

}

// Example usage:

var stack = new Stack(); stack.push(5); stack.push(10); stack.push(15);

console.log("Is 10 present in the stack?", stack.search(10)); // true console.log("Is 20 present in the stack?", stack.search(20)); // false
```

Write a JavaScript program to check whether a single linked list is empty or not. Return true otherwise false.

```javascript
class Node {

constructor(data) { this.data = data; this.next = null;

}

}

class LinkedList {

constructor() { this.head = null;

}

isEmpty() {

return this.head === null;

}

}

// Example usage:

var linkedList = new LinkedList(); console.log(linkedList.isEmpty()); // true

linkedList.head = new Node(10); console.log(linkedList.isEmpty()); // false
```

Write a JavaScript program to create a class called 'Rectangle' with properties for width and height. Include two methods to calculate rectangle area and perimeter. Create an instance of the 'Rectangle' class and calculate its area and perimeter.

```javascript
class Rectangle {

constructor(width, height) { this.width = width; this.height = height;

}

calculateArea() {

return this.width * this.height;

}

calculatePerimeter() {

return 2 * (this.width + this.height);

}

}

var rectangle = new Rectangle(5, 10);

var area = rectangle.calculateArea();

var perimeter = rectangle.calculatePerimeter();

console.log("Area:", area); //50 console.log("Perimeter:", perimeter); //30
```

Write a JavaScript program to create a slideshow that changes the displayed image when a next or previous button is clicked.

```javascript
var images = ["image1.jpg", "image2.jpg", "image3.jpg"]; // Replace these with actual image URLs

var currentIndex = 0;

function showSlide(index) {

var image = document.getElementById("image"); if (index >= 0 && index < images.length) { image.src = images[index];

currentIndex = index;

}

}

function nextSlide() {
```

```javascript
currentIndex = (currentIndex + 1) % images.length; showSlide(currentIndex);

}

function prevSlide() {

currentIndex = (currentIndex - 1 + images.length) % images.length; showSlide(currentIndex);

}

showSlide(currentIndex);
```

Write a JavaScript program that uses a try-catch block to catch and handle a 'SyntaxError' when parsing an invalid JSON string.

```javascript
var invalidJSONString = '{"name": "Jason", "age": 24,}';

try {

var parsedObject = JSON.parse(invalidJSONString); console.log(parsedObject);

} catch (error) {

if (error instanceof SyntaxError) { console.log("Error: Invalid JSON string."); console.log(error.message);

} else { throw error;

}

}
```

Write a JavaScript program to redirect to a specified URL.

```javascript
// Specified URL to Redirect To

var redirectURL = "https://www.example.com";

// Redirection to the Specified URL window.location.href = redirectURL;
```