1. Write a functional expression that duplicates each element of a given array.  
   **Input:** [2, 4, 7, 11, -2, 1]  
   **Output:** [2, 2, 4, 4, 7, 7, 11, 11, -2, -2, 1, 1]

|  |
| --- |
| 'use strict';  function duplicateElements(*inputArray*) {  var outputArray = [];  *//* Using forEach built in array method to  *//* iterate over array  inputArray.forEach(function (*elem*) {  outputArray.push(elem, elem);  });  return outputArray;  }  var input = [2, 4, 7, 11, -2, 1];  var output = duplicateElements(input);  console.log(output); |

1. Write a functional expression that removes all duplicates in a given array.  
   **Input:** [8, 13, 8, 9, 12, 8, 1, 1, 4, 13]  
   **Output:** [1, 4, 8, 9, 12, 13]

|  |
| --- |
| 'use strict';  function removeDuplicates(*inputArray*) {  var resultArray = [];  *//* We create a new array and then we sort  *//* it because sort() method is modifying  *//* existing array  var myArray = inputArray.slice().sort();  *//* We can use forEach to iterate array also  myArray.forEach(function (*element*, *index*, *arr*) {  var nextElem = arr[index + 1];  if (element !== nextElem) {  resultArray.push(element);  }  });  *//* Sort function sorts array in place  *//* modifying existing array  resultArray.sort(function (*a*, *b*) {  return a - b;  })  return resultArray;  }  var input = [8, 13, 8, 9, 12, 8, 1, 1, 4, 13];  var output = removeDuplicates(input);  console.log(output); |

* 1. Write a function that checks if a given array has odd number of elements.  
     **Input:** [1, 2, 9, 2, 1]  
     **Output:** true

|  |
| --- |
| 'use strict';  function isOddNumOfElements(*inputArray*) {  return !!(inputArray.length % 2);  }  var input = [1, 2, 9, 2, 3, 1];  var output = isOddNumOfElements(input);  console.log(output); |

* 1. Write a function that counts the number of elements less than the middle element. If the given array has an even number of elements, print out an error message.   
     **Input:** [-1, 8.1, 3, 6, 2.3, 44, 2.11]  
     **Output:** 4

|  |
| --- |
| 'use strict';  function numOfElementsLessThanMiddle(*inputArray*) {  if (inputArray.length % 2 === 0) {  return "ERROR";  }  var counter = 0;  var middleElementIndex = (inputArray.length - 1) / 2;  var middleElement = inputArray[middleElementIndex];  inputArray.forEach(function (*element*) {  if (element < middleElement) {  counter++;  }  });  return counter;  }  var input = [-1, 8.1, 3, 6, 2.3, 44, 2.11];  var output = numOfElementsLessThanMiddle(input);  console.log(output); |

1. Write a function that finds the smallest element of a given array. The function should return an object that contains the smallest value and its last position in the array.

|  |
| --- |
| 'use strict';  function findLowest(*inputArray*) {  var sortedArray = inputArray.slice().sort();  var minElem = sortedArray[0];  var minElemIndex = inputArray.lastIndexOf(minElem);  var minElemValue = inputArray[minElemIndex];  return {  minIndex: minElemIndex,  minValue: minElemValue  }  }  var input = [2, 2, 4, 4, 7, 7, 11, 11, -2, -2, 1, 1]  var output = findLowest(input)  console.log(output); |

* 1. Write a function that finds all the elements in a given array less than a given element.

|  |
| --- |
| 'use strict';  function elementsLessThanE(*inputArray*, *element*) {  var elementsLessThanE = [];  for (var i = 0; i < inputArray.length; i++) {  var arrayElement = inputArray[i];  if (arrayElement < element) {  elementsLessThanE.push(arrayElement);  }  }  return elementsLessThanE;  }  var input = [3, 4, 7, 9, 2, 6, 0, 1, 9]  var output = elementsLessThanE(input, 5)  console.log(output); |

* 1. Write a function that finds all the elements in a given array that start with the “pro” substring. The function should be case insensitive.

|  |
| --- |
| *//* Function that receives array and callback function  *//* Using a callback function we give our function  *//* flexibility to filter based on flexible criteria  'use strict';  function filterArray(*inputArray*, *callback*) {  var resultArray = [];  for (var i = 0; i < inputArray.length; i++) {  var elem = inputArray[i];  if (callback(elem)) {  resultArray.push(elem)  }  }  return resultArray;  }  var input = ["projection", "javascript", "promise"];  var output1 = filterArray(input, startsWithPro);  var output2 = filterArray(input, function (*element*) {  *//* Contains word "java"  return element.indexOf("java") !== -1  })  console.log(output1);  console.log(output2); |

* 1. Write a function that expects an array and a callback function that filters out some of the elements. Use functions defined in a) or b) to test it.

|  |
| --- |
| function eleBeginningWithPro(arr) {  var newArr = [];  for (var i = 0; i < arr.length; i++) {  var elem = arr[i];  var prefix = elem.toLowerCase().slice(0, 3);  if (prefix === "pro") {  newArr.push(elem);  }  }  function filterElements(arr, f) {  return f(arr);  }  console.log(filterElements(["projection", "javascript", "promise"], eleBeginningWithPro)); |

* 1. Write a list (array) of products you usually buy in the supermarket. Write a price and name for each product.  
     [

{name: ‘apples’, price: 100},

{name: ‘milk’, price: 80},

{name:’bananas’, price: 150}

]

* 1. Write a function that calculates the total price of your shopping list.

|  |
| --- |
| *//* Write a function that calculates the total price of your shopping list.  'use strict';  function calculateTotalPrice(*inputArray*) {  var totalPrice = 0;  for (var i = 0; i < inputArray.length; i++) {  totalPrice += inputArray[i].price;  }  return totalPrice;  }  var shoppingList = [  { name: 'apples', price: 100 },  { name: 'juice', price: 80 },  { name: 'bananas', price: 150 }  ]  var output = calculateTotalPrice(shoppingList);  console.log(output); |

* 1. Write a function that calculates the average product price of your shopping list. Print this value with the precision of three decimals.

|  |
| --- |
| *//* Write a function that calculates the average product price of your shopping list.  *//* Print this value with the precision of three decimals.  'use strict';  function calculateAveragePrice(*inputArray*) {  var sum = 0;  for (var i = 0; i < inputArray.length; i++) {  var obj = inputArray[i];  sum += obj.price;  }  var averagePrice = sum / inputArray.length;  return averagePrice.toFixed(3);  }  var shoppingList = [  { name: 'apples', price: 99 },  { name: 'juice', price: 83 },  { name: 'bananas', price: 150 }  ]  console.log(calculateAveragePrice(shoppingList)); |

* 1. Write a function that prints out the name of the most expensive product on your shopping list. Write it in uppercase.

|  |
| --- |
| 'use strict';  function mostExpensiveProduct(*inputArray*) {  var mostExpensive = inputArray[0];  for (var i = 0; i < inputArray.length; i++) {  if (inputArray[i].price > mostExpensive.price) {  mostExpensive = inputArray[i];  }  }  return mostExpensive.name.toUpperCase();  }  var shoppingList = [  { name: 'apples', price: 100 },  { name: 'juice', price: 80 },  { name: 'bananas', price: 150 }  ]  console.log(mostExpensiveProduct(shoppingList)); |

* 1. Write a function that checks if a given string is written in all capitals.

|  |
| --- |
| 'use strict';  function isAllCapitals(*str*) {  return str === str.toUpperCase();  }  var input = "HALLO";  var output = isAllCapitals(input);  console.log(output); |

* 1. Write a function that checks if a given string contains any digits.

|  |
| --- |
| 'use strict';  function isWithDigits(*str*) {  for (var i = 0; i < str.length; i++) {  var elem = str[i];  var parsedChar = parseInt(elem);  if (!isNaN(parsedChar) && typeof parsedChar === "number") {  return true;  }  }  return false;  }  var input = "H3llo";  var output = isWithDigits(input);  console.log(output); |

* 1. Write a function that checks if a given string is a valid hexadecimal color.

|  |
| --- |
| 'use strict';  function isHex(*string*) {  if (string.indexOf("#") === 0) {  string = string.slice(1);  }  if (string.length !== 6 && string.length !== 3) {  return false;  }  for (var i = 0; i < string.length; i++) {  var hexDigit = string.charAt(i);  var decimalHexDigit = parseInt(hexDigit, 16);  if (isNaN(decimalHexDigit)) {  return false;  }  }  return true;  }  var input = "#f9f9f9";  var output = isHex(input);  console.log(output); |

* 1. Write a function that checks if a given number belongs to the interval from 1900 to 2018.

|  |
| --- |
| 'use strict';  function isInInterval(*num*) {  return num >= 1900 && num <= 2018;  }  var input = 1995;  var output = isInInterval(input);  console.log(output); |

* 1. Write a function named *validator* that returns an object with properties *stringValidator*, *passwordValidator,* *colorValidator*, and *yearValidator* referencing the functions from a) to d).

|  |
| --- |
|  |

1. Write a function that calculates a number of days to your birthday.

**Input**: 25 February

**Output**: 5 days

|  |
| --- |
|  |

1. Write a function that for a given departure and arrival time calculates the time the trip takes.  
    **Input:** 8:22:13 11:43:22  
    **Output:** 3 hours 21 minutes 8 seconds

|  |
| --- |
|  |

1. 1. Write a constructor function that creates points in space. Each point in space has its own x, y, and z coordinate. For example, (3, 5, 1) can be a point in space.

|  |
| --- |
|  |

* 1. Write a function that calculates the distance between two points in the space.

|  |
| --- |
|  |

* 1. Write a function that generates a random integer value between 5 and 20.

|  |
| --- |
|  |

* 1. Write a function that generates a random integer value between 50 and 100.

|  |
| --- |
|  |

* 1. Write a function that expects a number and a callback generator function and returns an array of numbers produced by the generator function.

|  |
| --- |
|  |

1. Write a function that shuffles the elements of a given array.

**Input:** [3, 6, 11, 2, 9, 1]

**Output:** [6, 2, 9, 1, 3, 11] (it can be any random permutation of the given array)

|  |
| --- |
| 'use strict';  function shuffle(array) {  /\*  The idea is to pick a random index from 0 to array.length-1, read the element at that position and add it to  the resulting array. We should avoid random index repetition, therefore we need to store the information  about the used ones.  \*/  /\* resulting array \*/  var shuffledArray = [];  /\* used indexes \*/  var usedIndexes = [];  var randomIndex;  /\*  We should read all the elements from the given array and move them to the resulting array.  We can count the number of successful movings. The job is done as soon as we  move all the elements.  \*/  for (var counter = 0; counter < array.length;) {  /\* we generate a random index from 0 to array.length-1 \*/  randomIndex = Math.floor(Math.random() \* array.length);  /\* we check if the generated random index is used \*/  if (usedIndexes.indexOf(randomIndex) == -1) {  /\* if not, we add the element from that position to the resulting array \*/  shuffledArray.push(array[randomIndex]);  /\* we add the randomIndex to the array of used indexes \*/  usedIndexes.push(randomIndex);  /\* we can count this iteration as successful \*/  counter++;  }  /\* if the generated random index repeats, we proceed to the next iteration \*/  }  return shuffledArray;  }  console.log(shuffle([1, 2, 3, 4, 5, 6, 7])); |