Help to complete the tasks of this exercise can be found at least from the following book sources: From the first half of the chapter 10 “Functions” of the book “ Professional JavaScript for Web Developers” (5th edition) by Matt Frisbie; and from the chapters 3 ”Functions” and chapter 4 “Data Structures: Objects and Arrays” of the book “Eloquent JavaScript” (3rd edition) by Marijin Haverbeke.

The aims of the exercise are to learn the basics of working with functions, arrays, and objects in JavaScript.

Embed your theory answers, drawings, codes, and screenshots directly into this document. Always immediately after the relevant question. Return the document into your return box in itsLearning by the deadline.

Remember to give your own assessment when returning this document.

It’s also recommendable to use Internet sources to supplement the information provided by the course book.

The maximum number of points you can earn from this exercise is 10 + 1 = 11.

**Tasks:**

**1. Program a function by using a function declaration. (2 points)**

Write the function isLeapYear according to ES5 standard. **Use a function declaration**.

The function takes a year to be checked as a parameter.

The function returns true if the given parameter value is a leap year and false if it is not a leap year.

It should be noted that leap years must meet the following two conditions at the same time: 1) when the year is divided by four, the remainder is zero, and 2) (when the year is divided by one hundred, the remainder is not zero) or (when the year is divided by four hundred, the remainder is zero).

Fill a year into a textbox on a web page. Call the function by clicking the button Check year and display (by utilizing another function) "Year xxxx is a leap year" or "Year xxxx is not a leap year" in a div below. The xxx is the filled in year.

WE WILL PROGRAM THIS TOGETHER.

**2. Program a function by using a function expression. (1 point)**

Implement the function containsNumber according to ES5 standard. However, **use this time a function expression**.

The function takes two arguments. The first argument is numbers, which is an array of numbers. The second argument is aNumber, which is the number to search.

An example of one possible elements array:

[6,4,2,5,9,7,5,7,2]

The function returns true, if the array contains the number given in aNumber. Otherwise, the function returns false.

Display the result on the web page like in the task 1. However, this time the result text should be like “Array contains the number x” or “Array doesn’t contain the number x”.

**3. Program a function by using an arrow function. (2 points)**

Write the function convertToMinutesFormat. Use this time an arrow function introduced in ES6.

The function takes a hoursInHundredths as a parameter. The function should able to handle a parameter value that is given in one of the following formats: x.xx, xx.xx, x,xx or xx,xx.

Example parameter values: 3,40, 03.20, 0.15, 14.80.

The function returns hours and minutes in one of the following formats: h:mm or hh:mm.

From the above example parameter values the function returns: 3:24, 3:12, 0:09, 14:48.

Fill an hour time to convert into a textbox on a web page. Call the function by clicking the button “Convert to Minutes” and display the result like (by utilizing another function) "3,20 hours are in hours and minutes equal to 3:12"

Take care of the necessary rounding of minutes.

**4. Use Net to learn JavaScript API. (1 point)**

Search 3 good JavaScript references from the Net. Answer the following questions.

1. What are the names and the url of the references you found? (0,5 points)

W3Schools: <https://www.w3schools.com/js/default.asp>

MDN Web Docs – Javacript: <https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide>

JavaScript Tutorial: https://www.javascripttutorial.net/

1. What kind of information the references give to you? (0,5 points)

All the references are great resource for learning JavaScript. MDN Web Docs has extensive JavaScript documentation. All the references cover basic concepts of JavaScript and some of the references has information about modern JavaScript features or web API’s for JavaScript.

**5. Give short code examples. (1 point)**

a. How do you give default values for the function parameters?

A computer screen shot of text

Description automatically generated

b. How do you use rest parameters?

A computer screen with text and numbers

Description automatically generated

**6. Basics of object literals. (2 point)**

This time we concentrate on object literals. Write code clips to

1. Create an object book containing the following properties: isbn, name, authors, publicationDate. (0,5 points)

A screen shot of a computer

Description automatically generated

1. Add the following methods with the following names to the book object: getAuthors, setAuthors, getIsbn, setIsbn. (0,5 points)

A screen shot of a computer code

Description automatically generated

1. Create two book objects. Compare if they model the same book. You can use the value of the field isbn as comparison criteria in defining equality: Same isbn value, same book. (0,5 points)

A screen shot of a computer

Description automatically generated

1. Create two book objects with exactly the same values in all the features. Do they have the same identity? (0,5 points)

Even though the objects have identical properties, they are distinct objects in memory. Therefore, when comparing them using the strict equality operator (===), the result is false, indicating that they do not have the same identity.

**7. Working with objects. (2 points)**

Write the function convertOuncesToGrams. You can use ES6 standard features.

The function takes measurements as a parameter.

The measurements parameter is an array containing objects.

An example of the value of the measurements parameter:

[{ batchid: 434, unit: “ounce”, weight: 12.21 }, {batchid: 414, unit: “gram”, weight: 199.54 },{ batchid: 522, unit: “ounce”, weight: 18.88 }]

The function returns an array where all the measurements are in grams. Like the following:

[{ batchid: 434, unit: “gram”, weight: 346.15 }, {batchid: 414, unit: “gram”, weight: 199.54 },{ batchid: 522, unit: “gram”, weight: 535.24 }]

Please, give the results with two digits.