# Homework: JavaScript Syntax

This document defines the homework assignments from the [“JavaScript Basics“ Course @ Software University](http://softuni.bg/courses/javascript-basics/). Please submit as homework a single zip / rar / 7z archive holding the solutions (source code) of all below described problems.

## Powerful Cars

Write a JavaScript function **convertKWtoHP(value)** to convert car’s **kW** into **hp** (horse power). Write a JS program **powerfulCars.js** that converts a few sample values to **hp** (see the examples below). The result should be printed on the console, rounded up to the second sign after the decimal point. Run the program through **Node.js**. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| 75 | 100.58 |
| 150 | 201.15 |
| 1000 | 1341.02 |

***Hint:*** Use the link: <http://www.rapidtables.com/convert/power/how-hp-to-kw.htm>.

## Simple Calculations

Write a JavaScript function **roundNumber(value)** that rounds floating-point number using **Math.round()**, **Math.floor()**. Write a JS program **roundingNumbers.js** that rounds a few sample values. Run the program through **Node.js**. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| 22.7 | 22  23 |
| 12.3 | 12  12 |
| 58.7 | 58  59 |

## Check if Number is Even

Write a JavaScript function **evenNumber(value)** that checks if an **integer number** is **even.** Write JS program **evenChecker.js** to check if a few numbers are even. The result should be printed on the console (**true or false**). Run the program through **Node.js**. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| 3 | false |
| 127 | false |
| 588 | true |

## Check if Number is Prime

Write a JavaScript function **isPrime(value)** that checks if an **integer number** is **prime.** Write JS program **primeChecker.js** that checks if a few numbers are prime. The result should be printed on the console (**true** or **false**) on the console. Run the program through **Node.js**. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| 7 | true |
| 254 | false |
| 587 | true |

## Division by 3

Write a JavaScript function **divisionBy3(value)** that finds the **sum of digits of integer number n (n > 9)** and **checks if the sum is divided by 3 without remainder.** Write JS program **divisionChecker.js** to check a few numbers. The result should be printed on the console **(“the number is divided by 3 without remainder”** or **“the number is not divided by 3 without remainder”**). Run the program through **Node.js**. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| 12 | the number is divided by 3 without remainder |
| 188 | the number is not divided by 3 without remainder |
| 591 | the number is divided by 3 without remainder |

## Bit Checker

Write a JavaScript function **bitChecker(value)** that finds if the bit 3 an **integer number** (counting from 0) is 1. Write JS program **checkingBits.js** to check a few numbers. The result (**true** or **false**) should be printed on the console. Run the program through **Node.js**. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| 333 | true |
| 425 | true |
| 2567564754 | false |

## The Lifetime Supply Calculator

Write a JavaScript function **calcSupply(value)** that accepts the following parameters: **your age** (years), **your maximum age** (years), your **favorite food name** (e.g. "chocolate"), **estimate amount of your favorite food per day** (a number).The functioncalculates how many of the food you will eat for the rest of your life. Write JS program **lifetimeSupplyCalc.js** that calculates the amount of a few foods that you will eat. The result should be printed on the console. Run the program through **Node.js**. *Note: we assume that there are no leap years.* Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| 38  118  chocolate  0.5 | 14600kg of chocolate would be enough until I am 118 years old. |
| 20  87  fruits  2 | 48910kg of fruits would be enough until I am 87 years old. |
| 16  102  nuts  1.1 | 34529kg of nuts would be enough until I am 102 years old. |

## Cylinder Volume

Write a JavaScript function **calcCylinderVol(value)** that accepts the following parameters: **radius** and the **height** of a **straight circular cylinder**. The function calculates the volume of the cylinder. Write JS program **cylinderVol.js** that calculates the volume of a few cylinders. The result should be printed on the console. Run the program through **Node.js**. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| 2  4 | 50.265 |
| 5  8 | 628.319 |
| 12  3 | 1357.168 |

## Tree or House

Write a JavaScript function **treeHouseCompare(value)** that accepts the following parameters: integers **a** and **b**. The function **compares the area of the house and the area of the tree** (Figure 1) and **returns the name** of the figure with bigger area (**house** or **tree**) and the **value of the area**. Write JS program **treehouse.js** that compares a few houses and trees. The result should be printed on the console. Run the program through **Node.js**. Examples:

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Figure 1** |
| 3  2 | house/12.00 |
| 3  3 | tree/15.57 |
| 4  5 | tree/43.24 |

## Digit Checker

Write a JavaScript function **checkDigit(value)** that **finds if the third digit** (right-to-left) of an **integer number n (n>1000)** is 3. Write JS program **digitChecker.js** that checks a few numbers. The result (**true** or **false**) should be printed on the console. Run the program through **Node.js**. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1235 | false |
| 25368 | true |
| 123456 | false |

## Digit as Word

Write a JavaScript function **convertDigitToWord(value)** that **prints the name** of an **integer number n (0<n<10)** in English using switch statement. Write JS program **digitAsWord.**js that prints a few digits on the console. Run the program through **Node.js**. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| 8 | eight |
| 3 | three |
| 5 | five |

## Variables

Write a JavaScript function **variablesTypes(value)** that accepts the following parameters: **name**, **age**, **isMale** (**true** or **false**), array of your favorite foods. The function must return the values of the variables and their types. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| variablesTypes(['Pesho', 22, true, ['fries', 'banana', 'cake']]) | "My name: Pesho //type is string  My age: 22 //type is number  I am male: true //type is boolean  My favorite foods are: fries,banana,cake //type is object" |

***Constrains:*** Use **typeof()** to find the type of the variables.

# Problems for Champions

The next few problems are not mandatory. Implement them to challenge your skills.

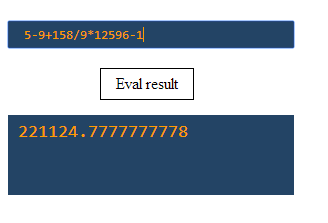
## \*Digital Soothsayer

Write a JavaScript function **soothsayer(value)** that accepts the following parameters (source arrays): **array of numbers**, **array of programming languages**, **array of cities**, **array of cars**. Each array must consist of five elements. The function must return an array **result[]** that consists of one random item from each source array. Write a JS program that prints on the console the following output: “You will work result[0] years on result[1]. You will live in result[2] and drive result[3].”. Run the program through **Node.js**. Examples:

|  |  |
| --- | --- |
| **Input** | **Output** |
| soothsayer([[3, 5, 2, 7, 9], [‘Java’, ‘Python’, ‘C#’, ‘JavaScript’, ‘Ruby’], [‘Silicon Valley’, ‘London’, ‘Las Vegas’, ‘Paris’, ‘Sofia’], [‘BMW’, ‘Audi’, ‘Lada’, ‘Skoda’, ‘Opel’]]) -->  result [5, JavaScript, Sofia, Opel] | You will work 5 years on JavaScript. You will live in Sofia and drive Opel. |

## \* Calculate Expression

Write a **HTML page** (with **text field**, **button**, and **paragraph**). Write JS program **calcExpression.js** that **calculates** any expression put in the text field and **prints** it in the paragraph. Link the JS file to the HTML file. (100% accuracy is not required). Example:



***Hint:*** Use **eval()** for calculating the result, but first remove the potentially dangerous characters.