ESMAD | TSIW | POO
Exercise Sheet Resolution nº3
Functions

Create a function to solve each of these points. For coding it is advisable to use Visual Studio Code:

1. **Function declaration**

    a. Create a function that displays the phrase "HELLO WORLD!" In an alert box. Give the function a good name. Run the function 3 times.

```
showMessage();
showMessage();
showMessage();

function showMessage() {
   alert("OLÁ MUNDO!");
}
```

2. **Parameters**

    a. Create a function that prints "**HELLO [name]!**", Where **name** is a variable passed as a parameter. The **name** variable must be initialized "hard-coded".

```
const NAME = "Rui";

showName(NAME);

function showName(name) {
  console.log(`HELLO ${name}!`);
}
```

    b. Create a function that prints the area of a rectangle when passed as parameters the length of the two sides.
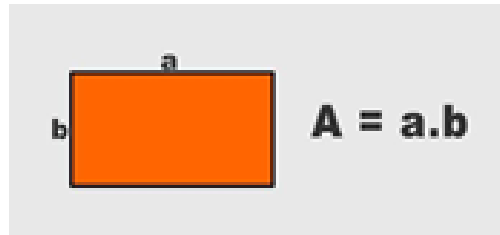
```
calcRectArea(3, 4);

function calcRectArea(a, b) {
   console.log(`A área do retângulo é ${a*b}`);
}
```

c. Create a function that displays the result of arithmetic operations in an alert box. The function must receive two integer values and an operator (+, -, *, /). All values must be obtained using **prompt** functions. Safeguard the division by 0 by presenting the user with a message stating that it is not possible to divide.


```
const num1 = +prompt('Digite o primeiro número?');
const num2 = +prompt('Digite o segundo número?');
const operator = prompt('Digite o operador?');

calculator(num1, num2, operator);

function calculator(num1, num2, operator) {
  switch (operator) {
   case '+':
    alert(`O resultado é ${num1 + num2}`);
    break;
   case '+':
    alert(`O resultado é ${num1 - num2}`);
    break;
   case '*':
    alert(`O resultado é ${num1 * num2}`);
    break;
   case '/':
```
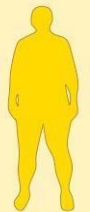
```
   if (num2 === 0) {
     alert('Não é possível efetuar a divisão de um número por 0');
   } else {
     alert(`O resultado é ${num1 / num2}`);
   }
   break;
  default:
   alert('Operador inválido!')
   break;
 }
}
```

d. Create a function that displays the classification of a person's BMI (Body Mass Index) in an alert box using the respective weight and height as parameters. Both parameters must be collected using **prompt** functions.

BMI = Weight(kg) / [Height(m)]²

Under weight	Normal weight	Over weight	Obese (Class I)	Obese (Class II)	Obese (Class III)
<18.5	18.5 – 24.9	25.0 – 29.9	30.0 – 34.9	35.0 – 39.9	>40.0

```
const weight = +prompt("Qual o seu peso?");
const height = +prompt("Qual a sua altura?");

calcImc(weight, height);

function calcImc(weight, height) {
  const imc = weight / (height * height);
  if (imc < 18.5) {
    alert("Abaixo do peso");
  } else if (imc >= 18.5 && imc <= 24.9) {
    alert("Peso normal");
  } else if (imc >= 25 && imc <= 29.9) {
    alert("Sobrepeso");
  } else if (imc >= 30 && imc <= 34.9) {
    alert("Obesidade grau I");
  } else if (imc >= 35 && imc <= 39.9) {
    alert("Obesidade grau II");
  } else
    alert("Obesidade grau III");
}
```

e. Create a function that simulates an echo. Given a string **s** and a number **n** it should print **s** written **n** times.

```
echo('ESMAD', 3);

function echo(s, n) {
  let result = '';
  for(let i = 0; i < n; i++) {
    result += s;
  }
  console.log(result);
}
```

f. Create a function that starts by reading two integer values **a** and **b** given by the user, and that writes all integer values belonging to the range **[a, b[**

```
const a = +prompt("Digite o primeiro número?");
const b = +prompt("Digite o segundo número?");

showNumbers(a, b);

function showNumbers(a, b) {
        let result = '';
        for(let i = a; i < b; i++) {
        result += i;
  }
  console.log(result);
}
```

g. Create a function that calculates the sum of multiples of 3 existing in an interval **[a, b]**, where **a** and **b** are passed as parameters.

```
const a = +prompt("Digite o primeiro número?");
const b = +prompt("Digite o segundo número?");

showMultiples(a, b);

function showMultiples(a, b) {
        let result = 0;
        for(let i = a; i <= b; i++) {
        if(i % 3 === 0)
                result += i;
  }
  console.log(result);
}
```

h. Create a function that returns the multiplication table for a given number passed as a parameter. If no number is passed, the function must print the multiplication table of 1.

```
1 X  1 =  1      2 X  1 =  2      3 X  1 =  3      4 X  1 =  4      5 X  1 =  5
1 X  2 =  2      2 X  2 =  4      3 X  2 =  6      4 X  2 =  8      5 X  2 = 10
1 X  3 =  3      2 X  3 =  6      3 X  3 =  9      4 X  3 = 12      5 X  3 = 15
1 X  4 =  4      2 X  4 =  8      3 X  4 = 12      4 X  4 = 16      5 X  4 = 20
1 X  5 =  5      2 X  5 = 10      3 X  5 = 15      4 X  5 = 20      5 X  5 = 25
1 X  6 =  6      2 X  6 = 12      3 X  6 = 18      4 X  6 = 24      5 X  6 = 30
1 X  7 =  7      2 X  7 = 14      3 X  7 = 21      4 X  7 = 28      5 X  7 = 35
1 X  8 =  8      2 X  8 = 16      3 X  8 = 24      4 X  8 = 32      5 X  8 = 40
1 X  9 =  9      2 X  9 = 18      3 X  9 = 27      4 X  9 = 36      5 X  9 = 45
1 X 10 = 10      2 X 10 = 20      3 X 10 = 30      4 X 10 = 40      5 X 10 = 50

6 X  1 =  6      7 X  1 =  7      8 X  1 =  8      9 X  1 =  9     10 X  1 = 10
6 X  2 = 12      7 X  2 = 14      8 X  2 = 16      9 X  2 = 18     10 X  2 = 20
6 X  3 = 18      7 X  3 = 21      8 X  3 = 24      9 X  3 = 27     10 X  3 = 30
6 X  4 = 24      7 X  4 = 28      8 X  4 = 32      9 X  4 = 36     10 X  4 = 40
6 X  5 = 30      7 X  5 = 35      8 X  5 = 40      9 X  5 = 45     10 X  5 = 50
6 X  6 = 36      7 X  6 = 42      8 X  6 = 48      9 X  6 = 54     10 X  6 = 60
6 X  7 = 42      7 X  7 = 49      8 X  7 = 56      9 X  7 = 63     10 X  7 = 70
6 X  8 = 48      7 X  8 = 56      8 X  8 = 64      9 X  8 = 72     10 X  8 = 80
6 X  9 = 54      7 X  9 = 63      8 X  9 = 72      9 X  9 = 81     10 X  9 = 90
6 X 10 = 60      7 X 10 = 70      8 X 10 = 80      9 X 10 = 90     10 X10 =100
```

```javascript
const num = +prompt("Digite um número?");

showMultiplicationNumber(num);

function showMultiplicationNumber(num = 1) {
        let result = '';
        for(let i = 1; i <= 10; i++) {
        result += `${num} x ${i} = ${num*i}\n`;
    }
  }
  alert(result);
}
```

i. Create a function that adds **N** numbers passed by parameters, Use the **arguments** object to solve this problem.

```javascript
sumNums(3, 4);
sumNums(7, 12, 9, 1);

function sumNums() {
    let result = 0;
    for(let i = 0; i < arguments.length; i++) {
        result += arguments[i];
    }
    alert(result);
}
```

j. Create a function that receives a child's first and last name and a set of strings that represent the name of each of your friends. The function must present the following sentence: "**The [firstName] [lastName] has [nFriends] friends!**". Use the **Rest** parameter to resolve this issue.

```
showFriends("Maria", "Silva");
showFriends("Maria", "Silva", "Inês");
showFriends("Maria", "Silva", "Joana", "Rute", "Inês");

function showFriends(firstName, lastName, ...friends) {
    console.log(`A ${firstName} ${lastName} tem ${friends.length} amigas!`);
}
```

3. **Function return**
    a. Create a **min(a, b)** function that returns the minimum of two numbers **a** and **b**.
        Test cases:

```
min(2, 5) == 2
min(3, -1) == -1
min(1, 1) == 1
```

```
console.log(min(2, 5));
console.log(min(3, -1));
console.log(min(1, 1));

function min(num1, num2) {
        if(num1<num2) {
        return num1;
  } else {
        return num2;
  }
}
```

    b. Create a **pow(x, n)** function that returns **x** at the power **n**. Or, in other words, multiply **x** by itself **n** times and return the result.
        Test cases:

```
pow(3, 2) = 3 * 3 = 9
pow(3, 3) = 3 * 3 * 3 = 27
pow(1, 100) = 1 * 1 * ...* 1 = 1
```

Before calling the function, you must ask the user for **x** and **n**. The function must support only natural values of **n**: integers above 1.

```
console.log(pow(3, 2));
console.log(pow(3, 3));
console.log(pow(4, 'x'));
console.log(pow(1, 100));

function pow(base, exponent) {
        if(isNaN(+base) || base <= 1) {
         return "Base deve ser um número inteiro superior a 1";
  } else if(isNaN(+exponent) || exponent <= 1) {
         return "Expoente deve ser um número inteiro superior a 1";
  } else {
        return base ** exponent;
  }
}
```

c. Create a function that checks whether a number passed as a parameter is prime or not. Remember that a prime number **N** is a natural number greater than 1 that has no divisors other than 1 and itself. Return true if it is prime and false otherwise. The function must be invoked within an **IF** structure and if it is true it must be presented "The number **[N]** is prime". Otherwise, "The number **[N]** is not prime".

```
const NUM = 8;

if(checkPrime(NUM)) {
  console.log(`O número ${NUM} é primo!`)
} else {
  console.log(`O número ${NUM} não é primo!`)
}


function checkPrime(num) {
        let isPrime = true;
  for(let i = 2; i < num; i++)        {
        if(num % i === 0) {
        isPrime = false;
        break;
    }
  }
  return isPrime;
}
```

d. Create a function that returns the factorial of a positive integer value passed as a parameter. Examples: 0!=1; 1!=1; 5!=5x4x3x2x1= 120; 6!=6x5x4x3x2x1= 720;

```
console.log(calcFactorial(0));
console.log(calcFactorial(1));
console.log(calcFactorial(5));
console.log(calcFactorial(6));

function calcFactorial(num) {
        let sum = 1;
  for(let i = 1; i <= num; i++)        {
        sum *= i
  }
  return sum;
}
```

4. **Function expressions**

    a. Create a function expression called **isLeapYear** that should return **true** if a year passed by the user is a leap or **false** otherwise.


```
let checkLeapYear = function(year) {
  return ((year % 4 == 0) && (year % 100 != 0)) || (year % 400 == 0);
}

console.log(checkLeapYear(2000));
console.log(checkLeapYear(2019));
console.log(checkLeapYear(2016));
```

    b. Create a function expression called **isPerfect** that checks if a number is perfect. A perfect number is a natural number for which the sum of all its own natural divisors (excluding itself) is equal to the number itself. For example, the number 28 is: 28=1+2+4+7+14. If the number is perfect return **true**. Otherwise, it must return **false**. Use the function in a loop that should continue to ask the user for a number until the number entered is perfect.

```
let checkPerfect = function(num) {
        let temp = 0;
  for(let i =1;i <= num/2; i++) {
        if(num % i === 0) {
        temp += i;
  }
 }
 if(temp === num && temp !== 0) {
        return true;
 } else {
        return false;
 }
}

let result;
do {
 const num = +prompt("Numero: ");
 result = checkPerfect(num);
}while(!result);
```

c. Create an abbreviated function expression (arrow function) that given a number from 100 to 999 check if it is a palindrome number. If so, it must return **true**. Otherwise, it must return **false**. Choose a good name for the function.

```
let checkPalindrome = num => Math.floor(num/100) === num % 10

console.log(checkPalindrome(121));
console.log(checkPalindrome(283));
console.log(checkPalindrome(350));
```

5. Create a guessing game. Start by generating a random number between 1 and 100. Then ask the user to guess the number. If the user enters a higher number, it should display the following text "**DOWN**". Otherwise, it must indicate: "**UP**". If the user gets it right, he should see the message **"CONGRATULATIONS, YOU GUESSED IT!"**.

In this game, you must have a function that takes two parameters: the number to guess and the user's attempt. The function must return:

-1 if the attempt is LOWER than the number initially generated

1 if the attempt is HIGHER than the number initially generated

0 if the attempt is EQUAL to the number initially generated

Make the game more interesting, and give the player just 5 tries. If the user reaches the limit, he should see: "PATIENCE, PLEASE PLAY AGAIN!".

```javascript
let attempts = 0;
let result;
const limitAttempts = 5;
const randomNumber = Math.floor((Math.random() * 100) + 1);

do {
        attempts++;
        const attemptNumber = +prompt("Digite um número?");
  result = checkNumber(randomNumber, attemptNumber);
        if(result === 0) {
    break;
  } else if(result > 0) {
        alert("PARA BAIXO!");
  } else {
        alert("PARA CIMA!");
  }
} while(attempts < limitAttempts)

if(result === 0) {
        alert("ADIVINHOU, PARABÉNS!");
} else {
        alert("PACIÊNCIA, JOGUE OUTRA VEZ");
}

function checkNumber(randomNumber, attemptNumber) {
        if(randomNumber === attemptNumber) {
        return 0;
  } else if(randomNumber > attemptNumber) {
        return -1;
  } else {
        return 1;
  }
}
```