JS Session I : functions / parameters / return

Write functions matching the following requirements.
Test their behaviours in your browser's console or in a node REPL shell (easier with nodemon)
Don't miss the last iteration ! (^_^)

Hint >>>

Don't forget these concepts when dealing with functions :

1. Function déclarations
2. Function parameter(s)
3. Function return statement
4. Function execution

Iteration 1 - write a function sayHelloWorld

This function displays a "hello world" **String** in the console.

This function doesn't take any **parameter**.

Do not use **return** statement for this function.

When executing, what are the displayed value ?

```
console.log(sayHelloWorld())
```

Can you explain why ?

Iteration 2 - Write a function foo

foo does not take a **parameter**.
It **returns** the **String** "bar" when executed.

Test it with the statement:

```
console.log(foo())
```

Test it also with a variable assignment:

```
var x = foo()
```

What is the value of x and why ?

Iteration 3 - write a function dummyReturn

This function takes a parameter p.
It **returns** p, unmodified.

Test with the statement console.log(dummyReturn()).
Test also with a variable assignment : var x1 = dummyReturn().
Check the value of x1 in your console.

Iteration 4 - write a function sayHelloTo

This function takes a parameter **name**.
It throws an Error if **name** is not a **String**.
It **returns** concatenated "hello" **String** with **name**.

Iteration 5 - write a function makeOpposite

It takes a boolean **b** as parameter.
If **b** is not a **Boolean**, it throws a new error.
Otherwise it **returns** the boolean opposite of b.

Test it with a variable assignment :

```
var x2 = makeOpposite(aValueHere).
```

Check the value of x2 in your console.
Test also with a if statement : if ((makeOpposite(aValueHere)) { console.log("yes !!!") }
Can you see the last log ?

Iteration 6 - write a function calculate

Calculate takes 3 parameters :

1. a **String operator**
2. a **Number operande1**
3. a **Number operande2**

The function **returns** the result of a simple math operation,
according to the operator and 2 operandes.

Use these call examples to get started:

```
calculate("+", 1, 1);
calculate("/", 10, 3);
calculate("*", calculate("%", 999, 10), calculate("-", 10, 3.333));
```

If the operator is not one of these `["+", "-", "*", "/", "%"]`, throw an explicit error.
If the operation result is NaN, throw an explicit error.
Else, calculate **returns** a **Number**, result of the operation.

WARNING: do **NOT** use the JS built-in eval function.

Iteration 7 - Write functions documentation

Write a detailed documentation for each function.
Get started by consulting the site usejsdoc.org.
Generate an HTML file with the command-line utility provided by jsdoc.