Js Quest05

Subject 1 Solution Additional Resources (1)

# Js Quest05

Remember to git add && git commit && git push each exercise!

We will execute your function with our test(s), please DO NOT PROVIDE ANY TEST(S) in your file

For each exercise, you will have to create a folder and in this folder, you will have additional files that contain your work. Folder names are provided at the beginning of each exercise under `submit directory` and specific file names for each exercise are also provided at the beginning of each exercise under `submit file(s)`.

# Introduction

Let's continue our coding journey by writing some basic math functions.

Our first assignment will be focus on "dynamic" and we can pass arguments to any of our script from the terminal.

## Control Center

roup formation

n Progress

ubmitted

est review

Finished: approved

Go To DoCode

Access:

READ WRITE

Go To Gitea

Keep Working On This Soluti

## Looking for a group

| Js Quest05 | My First Script With Args |
|---|---|
| Submit directory | ex00 |
| Submit file | my_first_script_with_args.js |

## Description

Let's do our first loop statement!

Create a file `my_first_script_with_args.js`.

It will print any argument received to the script

**Example 00 (In Javascript)**

```
$>node my_first_script_with_args.js blah1 blah2 blah3
blah1
blah2
blah3
$>
```

**Example 01 (In Python)**

```
$>python my_first_script_with_args.py blah1
blah1
$>
```

**Example 02 (In Ruby)**

```
$>ruby my_first_script_with_args.rb "blah1 blah2 blah3"
blah1
blah2
blah3
$>
```

No body is looking for a partner at the moment

## Also working on the project

iron_j jamolidd_b toychibo_a xolxojay_f atoyev_u

## Just finished

aripova_b akhmeto_sa kozhemra_d vitale_d peralta_r bennett-_k champeno_s musharba_j thomason_b zuckerma_r hwang_d ji_j chun_k hanna_e

Type

Project

Group Size

1 Participant

*Tip*
Google the following: script in YOURCODINGLANGUAGE receiving arguments (argv)

| Js Quest05 | My Is Negative |
|---|---|
| Submit directory | ex01 |
| Submit file | my_is_negative.js |

### Description

Let's get starting with some if-else statement!

Create a `my_is_negative` function.
This function `my_is_negative` returns `1` or `0` depending on the integer's sign entered as a parameter.
If `n` is negative, return `0`. If `n` is positive or 0, return `1`.

### Function prototype (javascript)

```
/*
**
** QWASAR.IO -- my_is_negative
**
**
** @param {Integer} param_1
** @return {integer}

**
*/


function my_is_negative(param_1) {

};
```

*Tip*
(In Javascript)
Your script will look like something close to this:

Review system
Test Review (Gandalf)

Difficulty
Initiation

Average duration
1 Week

### Project's Metadata

Project

id: 61

name: js-quest05

visible: True

```
function my_is_negative(n) {
  if (XXXXX) {
    return XXX;
  }
  else {
    return XXX;
  }
}

console.log(my_is_negative(-1));
console.log(my_is_negative(1));
console.log(my_is_negative(0));

// console.log(my_is_negative(1337));

// REMEMBER WHEN YOU ARE FINISHED TO COMMENT ALL CALL TO YOUR
// FUNCTION my_is_negative function
// OTHERWISE IT WILL FAIL THE AUTOMATIC TEST SYSTEM
//
// <- yes this a way to comment your code
```

---

| Js Quest05 | My Abs |
|---|---|
| Submit directory | ex02 |
| Submit file | my_abs.js |

### Description

Create a `my_abs` function.

Reproduce behavior of an abs() function. It returns always the positive value of a number.

### Function prototype (javascript)

```
/*
**
** QWASAR.IO -- my_abs
**
**
** @param {Integer} param_1
** @return {integer}

**
*/


function my_abs(param_1) {

};
```

**Example 00**

```
Input: -30
Output:
Return Value: 30
```

**Example 01**

```
Input: 30
Output:
Return Value: 30
```

**Example 02**

```
Input: 0
Output:
Return Value: 0
```

---

| Js Quest05 | My Add |
|---|---|
| | |

| | |
|---|---|
| Submit directory | ex03 |
| Submit file | my_add.js |

### Description

Create a `my_add` function which takes `2` parameters (`nbr1` and `nbr2`) and returns a `value`.
This `value` is the result of the addition of `nbr1` and `nbr2` parameters.

### Function prototype (javascript)

```
/*
**
** QWASAR.IO -- my_add
**
**
** @param {Integer} param_1
** @param {Integer} param_2
** @return {integer}

**
*/


function my_add(param_1, param_2) {

};
```

**Example 00**

```
Input: 0 && 1
Output:
Return Value: 1
```

**Example 01**

```
Input: 10 && 10
Output:
Return Value: 20
```

**Example 02**

```
Input: -10 && 10
Output:
Return Value: 0
```

---

| Js Quest05 | My Sub |
|---|---|
| Submit directory | ex04 |
| Submit file | my_sub.js |

### Description

Create a `my_sub` function which takes `2` parameters ( `nbr1` and `nbr2` ) and returns a `value` .
This `value` is the result of the subtraction of `nbr1` and `nbr2` parameters.

### Function prototype (javascript)

```
/*
**
** QWASAR.IO -- my_sub
**
**
** @param {Integer} param_1
** @param {Integer} param_2
** @return {integer}

**
*/


function my_sub(param_1, param_2) {

};
```

**Example 00**

```
Input: 0 && 1
Output:
Return Value: -1
```

**Example 01**

```
Input: 10 && 10
Output:
Return Value: 0
```

**Example 02**

```
Input: -10 && 10
Output:
Return Value: -20
```

---

| Js Quest05 | My Mult |
|---|---|
| Submit directory | ex05 |
| Submit file | my_mult.js |

### Description

Create a `my_mult` function which takes `2` parameters ( `nbr1` and `nbr2` ) and returns a `value` .
This `value` is the result of the multiplication of `nbr1` and `nbr2` parameters.

### Function prototype (javascript)

```
/*
**
** QWASAR.IO -- my_mult
**
**
** @param {Integer} param_1
** @param {Integer} param_2
** @return {integer}

**
*/


function my_mult(param_1, param_2) {

};
```

**Example 00**

```
Input: 0 && 1
Output:
Return Value: 0
```

**Example 01**

```
Input: 10 && 10
Output:
Return Value: 100
```

**Example 02**

```
Input: -10 && 10
Output:
Return Value: -100
```

---

| Js Quest05 | My String Formatting |
|---|---|

| | |
|---|---|
| Submit directory | ex06 |
| Submit file | my_string_formatting.js |

### Description

Create a `my_string_formatting` function which takes `3` parameters ( `firstname` , `lastname` and `age` ) and prints a string composed `value` .

Formatting should be: "Hello, my name is FIRSTNAME LASTNAME, I'm AGE."

Make sure you are printing a newline.

### Function prototype (javascript)

```
/*
**
** QWASAR.IO -- my_string_formatting
**
**
** @param {String} param_1
** @param {String} param_2
** @param {Integer} param_3
**
*/


function my_string_formatting(param_1,
param_2, param_3) {

};
```

**Example 00**

```
Input: "john" && "doe" && 37
Output: Hello, my name is john doe, I'm
37.

Return Value: nil
```

**Example 01**

```
Input: "Baby" && "Yoda" && 50
Output: Hello, my name is Baby Yoda, I'm
50.

Return Value: nil
```

**Example 02**

```
Input: "Marie" && "Curie" && 26
Output: Hello, my name is Marie Curie,
I'm 26.

Return Value: nil
```

*Tip*
You should use Google to learn about String interpolation :-)