Javascript Lab Seminar

MELL-JAVASCRIPT-04

Bistromatik

v1.61

Bistromatik

repository name: javascript_lab

branch name: bistro

Your repository must contain the totality of your source files.

You must have one file per Task. my_compute_factorial_it is the task one so you need to have a file named `my_compute_factorial_it.js`.

You are only allow to use **var let const if while** and **for**
Do not use any function of any kind that is not your. If you need to use concact() method then create it.

If one of your files prevents you from compiling and if we are not able to correct your work you will receive a 0.

All of the day's functions must produce an answer in under 2 seconds. Overflows must be handled (as errors).

Here's a complete list of the packages we'll use specifically for developing on the command line:

- chalk — colorizes the output
- clear — clears the terminal screen
- clui — draws command-line tables, gauges and spinners
- figlet — creates ASCII art from text
- inquirer — creates interactive command-line user interface
- minimist — parses argument options
- configstore — easily loads and saves config without you having to think about where and how.

Task 01

The goal of this project is to write a program that can evaluate an arithmetic expression composed of integers of infinite size

It takes a string as parameter, which represents a mathematical expression, evaluates this expression and returns the result as an integer.

The string received as parameter will always be **valid** (no syntax errors, no divisions by zero,...).

The following five operators must be supported:

- ’**+**’ for addition,
- **’-’** for subtraction,
- ’/’ for division,
- ’*’ for multiplication,
- ’**%**’ for modulo.

The function must also handle any number of parenthesis.

Here is how the function will be tested:

```
$> node bistro.js “(3+2)*5”


$> node bistro.js “(30000+2)*99000”
```