CALCULATOR

CLI

Student

Mohammed Riyaan

Index

About Project Calculator CLI

Command Line Interface (CLI)

A command-line interface (CLI) is a means of interacting with a computer program or operating system through text-based commands entered into a command-line interpreter or shell. It provides a direct and efficient way to communicate with the computer by typing commands and receiving textual output as a response.

Calculator CLI

A simple command-line interface (CLI) calculator that allows you to perform basic mathematical calculations without the need for a graphical user interface. This calculator CLI provides a convenient way to perform calculations directly from your command prompt or terminal.

Features:-

- Supports addition, subtraction, multiplication, and division operations.
- Handles both integer and floating-point numbers.
- Accepts multiple inputs in a single command.
- Provides a clear and concise output of the calculated result.
- Supports parentheses for grouping expressions and controlling the order of operations.

This project ask user, two operands and the operation such as addition , subtraction, multiplication, division, remainder and the result will be as:

```
Enter First Number : 10
Enter Second Number : 2
Enter The Operator : +

The Sum Will be : 10+2=12
```

Tech stack used in Calculator CLI

Backend - Server

1. Programming Languages - JavaScript
2. Runtime Environment - Node JS

NPM Packages

1. readline-sync:- Synchronous Readline for interactively running to have a conversation with the user via a console.

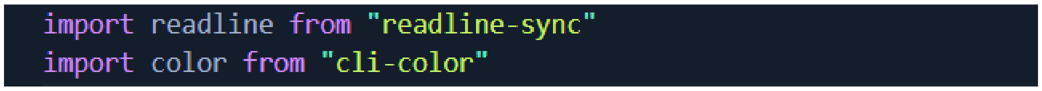
2. cli-color:- Colors, formatting and other goodies for the console. This package won't mess with built-ins and provides neat way to predefine formatting patterns.

3. Proposed Solution

Procedure

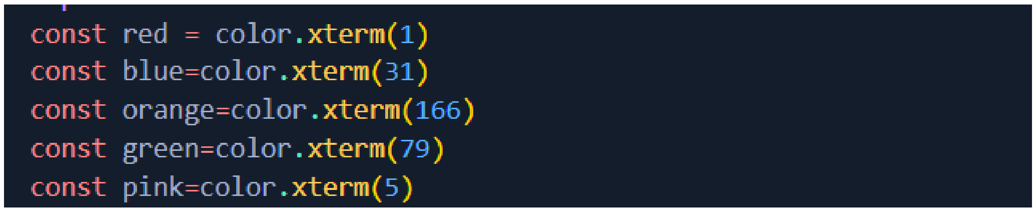
Installing readline-sync , cli-color packages

Importing packages



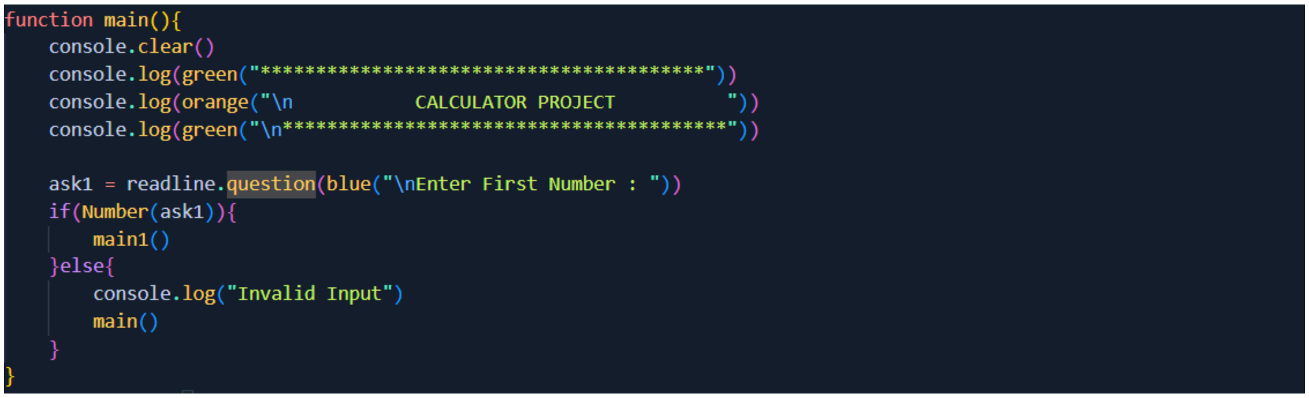
```
import readline from "readline-sync"
import color from "cli-color"
```

Using Cli-color package, naming the cli package codes to the new color variables



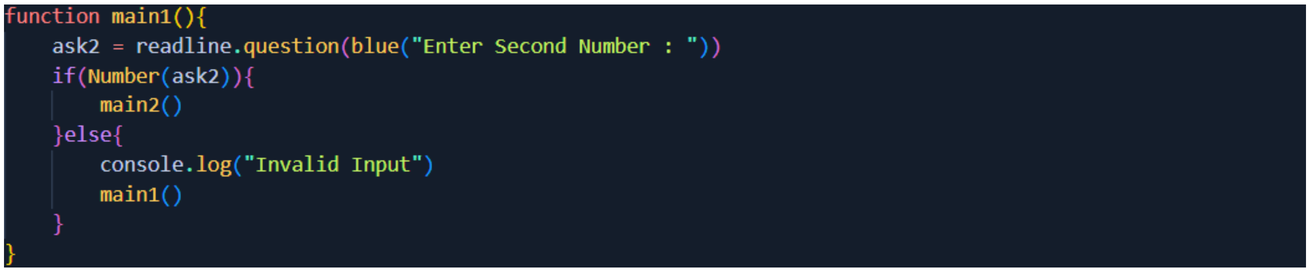
```
const red = color.xterm(1)
const blue=color.xterm(31)
const orange=color.xterm(166)
const green=color.xterm(79)
const pink=color.xterm(5)
```

Then the First Main Function consists of providing the first input from the user using readline-sync package and verifying that the input should be only Number using if-else statement. If the condition is true then the second function is initiated else it will print invalid input.



```
function main(){
    console.clear()
    console.log(green("****************************************"))
    console.log(orange("\n           CALCULATOR PROJECT          "))
    console.log(green("\n****************************************"))

    ask1 = readline.question(blue("\nEnter First Number : "))
    if(Number(ask1)){
        main1()
    }else{
        console.log("Invalid Input")
        main()
    }
}
```

The Second Function consists of providing the second input from the user using readline-sync package and verifying that the input should be only Number using if-else statement. If the condition is true then the third function is initiated else it will print invalid input.



```
function main1(){
    ask2 = readline.question(blue("Enter Second Number : "))
    if(Number(ask2)){
        main2()
    }else{
        console.log("Invalid Input")
        main1()
    }
}
```

The Third Function consists of providing the operation such as + , - , * , / , % from the user and declaring result as a variable. Then we are using switch-case statement to verify the operations .

1. If the operation input is + then the first case is used. In this case the first number and second number will be added and its result is stored in result variable and then breaking the switch-case.

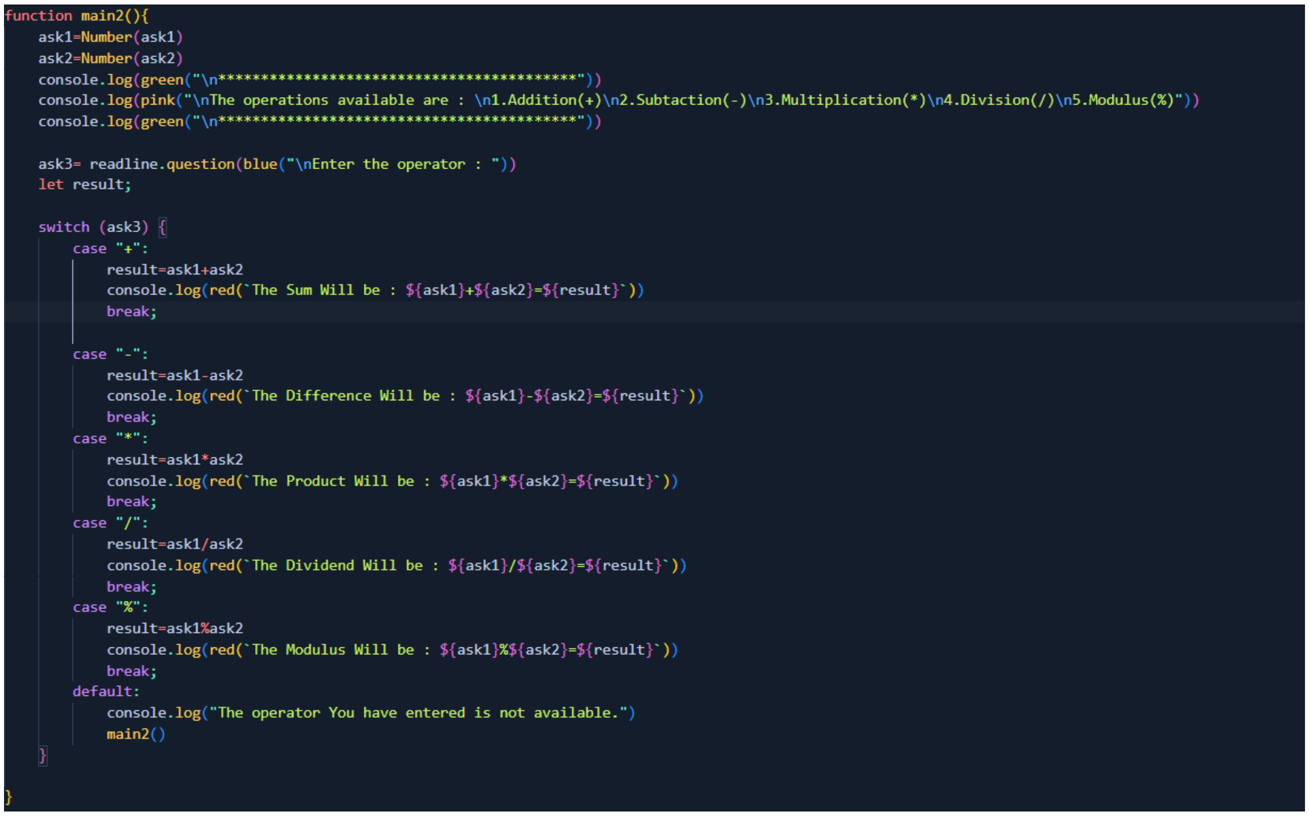
2. If the operation input is - then the second case is used. In this case the first number and second number will be subtracted and its result is stored in result variable and then breaking the switch-case.

3. If the operation input is * then the third case is used. In this case the first number and second number will be multiplied and its result is stored in result variable and then breaking the switch-case.

4. If the operation input is / then the fourth case is used. In this case the first number and second number will be divided and its result is stored in result variable and then breaking the switch-case.

5. If the operation input is % then the fifth case is used. In this case the first number and second number will be modulus and its result is stored in result variable and then breaking the switch-case.

6. If the operation input is anything other than the above cases then the default case is used. This case prints that the operator you have entered is not available. After this the 2nd function is called again .

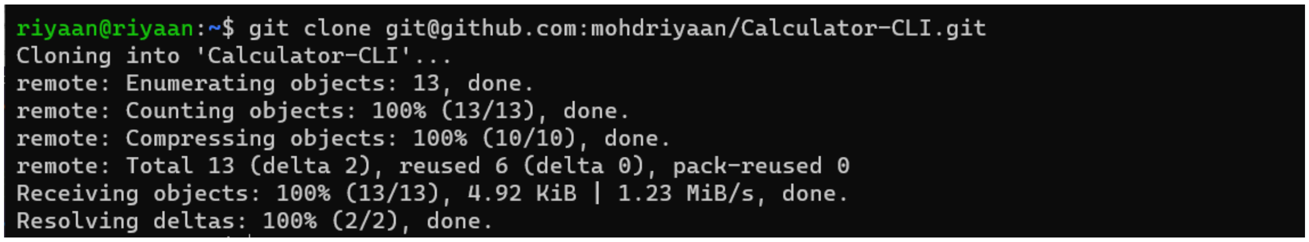


```javascript
function main2(){
    ask1=Number(ask1)
    ask2=Number(ask2)
    console.log(green("\n******************************************"))
    console.log(pink("\nThe operations available are : \n1.Addition(+)\n2.Subtaction(-)\n3.Multiplication(*)\n4.Division(/)\n5.Modulus(%)"))
    console.log(green("\n******************************************"))

    ask3= readline.question(blue("\nEnter the operator : "))
    let result;

    switch (ask3) {
        case "+":
            result=ask1+ask2
            console.log(red(`The Sum Will be : ${ask1}+${ask2}=${result}`))
            break;

        case "-":
            result=ask1-ask2
            console.log(red(`The Difference Will be : ${ask1}-${ask2}=${result}`))
            break;
        case "*":
            result=ask1*ask2
            console.log(red(`The Product Will be : ${ask1}*${ask2}=${result}`))
            break;
        case "/":
            result=ask1/ask2
            console.log(red(`The Dividend Will be : ${ask1}/${ask2}=${result}`))
            break;
        case "%":
            result=ask1%ask2
            console.log(red(`The Modulus Will be : ${ask1}%${ask2}=${result}`))
            break;
        default:
            console.log("The operator You have entered is not available.")
            main2()
    }

}
```

Installation of Calculator CLI

GitHub Repository: https://github.com/mohdriyaan/Calculator-CLI

To clone and Install packages:



```
riyaan@riyaan:~$ git clone git@github.com:mohdriyaan/Calculator-CLI.git
Cloning into 'Calculator-CLI'...
remote: Enumerating objects: 13, done.
remote: Counting objects: 100% (13/13), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 13 (delta 2), reused 6 (delta 0), pack-reused 0
Receiving objects: 100% (13/13), 4.92 KiB | 1.23 MiB/s, done.
Resolving deltas: 100% (2/2), done.
```

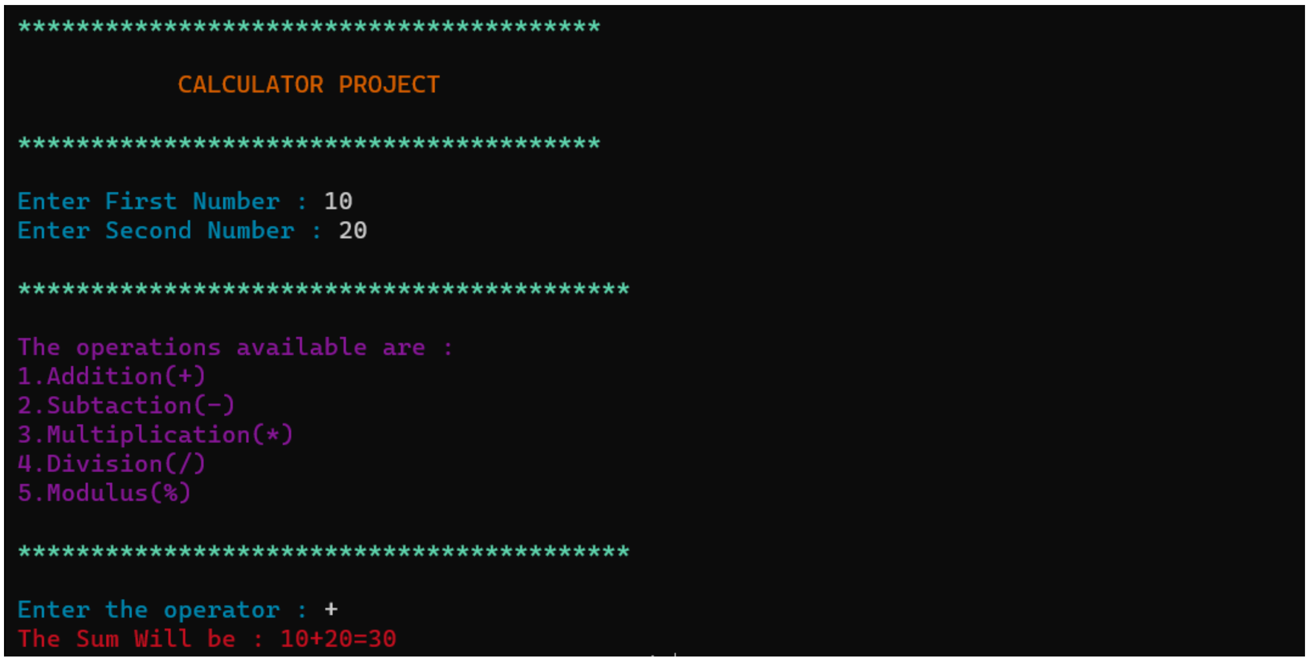
To Run Project

cd Calculator-CLI

npm i

npm start

Output :-



```
****************************************

           CALCULATOR PROJECT

****************************************

Enter First Number : 10
Enter Second Number : 20

******************************************

The operations available are :
1.Addition(+)
2.Subtraction(-)
3.Multiplication(*)
4.Division(/)
5.Modulus(%)

******************************************

Enter the operator : +
The Sum Will be : 10+20=30
```