Aiden Shih 10/3/17

Asynchronous Programming

We traditionally think of code as being executed from top to bottom such that lower code waits for code higher up to complete before being executed.

Many programming languages operate this way. But this is not always useful. For example, if you're fetching data through an API call, it might take a while for that request to go all the way to the server, be processed, and make its way back to you. Meanwhile, your app could be doing other things, like loading the rest of the page.

For this reason, JavaScript -- a language designed for the web -- heavily relies on asynchronous processes. From StackOverflow: "When you execute something synchronously, you wait for it to finish before moving on to another task. When you execute something asynchronously, you can move on to another task before it finishes."

function sayHello(){

console.log('hello')

}

function sayHello2(){

while (true){

}

}

setInterval(sayHello, 100)

setInterval(sayHello2, 200)

var readline = require('readline')

console.log(readline)

var rl = readline.createInterface({

input: process.stdin,

output: process.stdout

})

rl.on('line', function(input) {

return typeof input

})

Exercise 1

Write a counter program that reads the user input one line at a time. The count would be initially zero. The app will start by displaying the count and prompting the user for input. valid input strings are: 'inc', 'dec', 'res'. The user input may be in uppercase, lowercase, or any combination of the two. On input of 'inc' and 'dec', the counter will be incremented / decremented by 1. On input of 'res'  the counter will be reset. Following each of these, the new count will be displayed, followed by a relevant message (e.g. 'the counter has been incremented') followed by a prompt to the user for more input.

var readline = require('readline')

var rl = readline.createInterface({

input: process.stdin,

output: process.stdout

})

function clear() {

process.stdout.write('\u001B[2J\u001B[0;0f')

}

var validInput = '[Valid inputs: inc, dec, res]';

console.log('Count: 0 ' + validInput)

var lowerCase;

var count = 0;

rl.on('line', function (input) {

clear()

lowerCase = input.toLowerCase();

if (lowerCase === 'inc'){

count++;

console.log('Count: ' + count + ' The counter has been increased. \nPlease enter another input. ' + validInput)

} else if (lowerCase === 'dec'){

count--;

console.log('Count: ' + count + ' The counter has been decreased. \nPlease enter another input. ' + validInput)

} else if (lowerCase === 'res'){

count = 0;

console.log('Count: ' + count + ' The counter has been reset. \nPlease enter another input. ' + validInput)

} else {

console.log('Invalid entry. Please enter another input. ' + validInput)

}

})

Question 2

Write a program that reads the user input one line at a time. It would expect the user input to start with one of four mathematical symbols: + - \* / followed by two numbers, separated by spaces. For example: + 4 2 . The program will perform the desired mathematical operation (addition, subtraction, multiplication or division) and log the result. The program will display an explanatory message if:

1. the operation is invalid.

2. The operation is not followed by the expected two numbers (with spaces)

For example, if the user enters: \* 2 3 the program will log 6.

var add = function (num1, num2) {

return num1 + num2

}

var sub = function (num1, num2) {

return num1 - num2;

}

var mult = function (num1, num2) {

return num1 \* num2;

}

var div = function (num1, num2) {

return num1 / num2;

}

console.log('[Please start with a symbol. + - \* / Then choose two numbers.')

rl.on('line', function (input) {

clear();

console.log(input);

var newArr;

newArr = input.split(' ');

for (var i = 0; i < 1; i++) {

if (newArr[i] === '+') {

return console.log(add(Number(newArr[i + 1]), Number(newArr[i + 2])))

} else if (newArr[i] === '-') {

return console.log(sub(Number(newArr[i + 1]), Number(newArr[i + 2])))

} else if (newArr[i] === '\*') {

return console.log(mult(Number(newArr[i + 1]), Number(newArr[i + 2])))

} else if (newArr[i] === '/') {

return console.log(div(Number(newArr[i + 1]), Number(newArr[i + 2])))

} else {

console.log('Incorrect format, please check and try again.')

}

}

})

Question 3

Write a program that reads the user input one line at a time. It would expect the user input to start with one of four mathematical symbols: + - \* / followed by numbers, separated by spaces. For example: + 2 4 2 . The program will perform the desired mathematical operation (addition, subtraction, multiplication or division) on all the numbers, from left to right, and log the result . The program will display an explanatory message if:

1. the operation is invalid.

2. The operation is not followed by at least two numbers (with spaces)

For example, if the user enters:- 6 5 1 2 the program will log -2 (= 6 - 5 - 1 - 2).

var readline = require('readline')

var rl = readline.createInterface({

input: process.stdin,

output: process.stdout

})

function clear() {

process.stdout.write('\u001B[2J\u001B[0;0f')

}

rl.on('line', function (input) {

clear();

var a = input.split(' ');

var ans = 0;

var operators = '+-\*/';

if (!operators.includes(a[0]))

console.log('invalid operator');

else {

for (var i = 1; i < a.length; i++) {

if (!isNaN(a[i])) {

if (i === 1)

ans = a[i];

else

switch (a[0]) {

case '+':

ans += Number(a[i]);

break;

case '-':

ans -= Number(a[i]);

break;

case '\*':

ans \*= Number(a[i]);

break;

case '/':

ans /= Number(a[i])

break;

}

} else {

console.log('invalid number(s)');

break;

}

}

}

if (i === a.length)

console.log(ans);

})