Entrega 2 - Arrays e iteradores

Versión: 12 de Febrero de 2019

Objetivo

Practicar con: arrays y sus métodos, el entorno de ejecución de nodejs, programación funcional en JS6, iteradores, multi-asignación y los nuevos operadores spread/rest.

Descripción de la práctica

Realizar el programa node.js **mod2_cmd_iterators.js**, que pueda invocarse con un número variable de parámetros, procesándolos tal y como se indica a continuación.

Incluir el código en el fichero vacío **mod2_cmd_iterators.js** que se incluye con el proyecto de prueba: https://github.com/practicas-ging/mooc_node-mod2_cmd_iterators (ver sección "Realización y prueba de la práctica").

Este comando debe realizar lo siguiente:

1. Primero debe mostrar una línea en blanco seguida de las rutas al interprete de node.js y al fichero mod2_cmd_iterators.js (que contiene el programa) y de otra línea en blanco
2. A continuación debe mostrar los parámetros en orden alfabético, cada uno en una línea, seguido del string ": " y del número de veces que esta repetido. Si se añade la opción -r delante de un parámetro este se elimina y no aparece en el listado. Este ejemplo muestra que mostrar por pantalla
3. Por último debe finalizar y retornar a la shell.

Por ejemplo, si invocamos así programa, deberá dar el resultado que se indica a continuación:

```
$
$ node mod2_cmd_iterators.js  uno -r dos uno dos tres dos tres tres

Route to node.js: /usr/local/bin/node
Route to this file: /Users/jq/sol/mod2_cmd_iterators.js

tres: 3
uno: 2
$
$
```

Nota: Los parámetros de la invocación de un programa node.js están accesibles en el array process.argv del entorno (para mas detalles se puede ver el primer tema de node.js). process.argv contiene en las posiciones 0 y 1 las rutas absolutas al interprete de node.js y al fichero fuente. A continuación vienen los demás parámetros.

Realización del programa:

El programa debe comenzar asignando con multi-asignación los parámetros de la invocación a tres variables: a la primera el parámetro 0 (ruta al interprete de node), a la segunda el parámetro 1 (ruta al fichero mod2_cmd_iterators.js) y a la tercera un array con el resto de los parámetros.

A continuación se imprimen por consola los mensajes con la ruta a node.js y al fichero.

Después se busca si se ha incluido la opción -r en el array con el resto de parámetros y se eliminan todas las ocurrencias del parámetro que viene a continuación en dicho array.

El array debe ordenarse con la función sort() y reducir con el iterador reduce(..) para mostrar por consola los parámetros, tal y como se indica, en orden alfabético y con el número de ocurrencias:

El array ordenado tendrá todos los parámetros repetidos en posiciones consecutivas. El iterador reduce(..) permite recorrer el array, utilizando el acumulador para contar el número de parámetros iguales. En cada iteración de reduce se puede proceder así:

- Si parámetro igual al anterior, incrementa el acumulador en 1
- Si parámetro no es igual al anterior, muestra el parámetro anterior por consola con el número de repeticiones e inicializa el acumulador a 0 para empezar la cuenta del nuevo parámetro.

Realización y prueba de la práctica

Para comprobar que la práctica ha sido realizada correctamente hay que utilizar el validador de este repositorio

https://github.com/practicas-ging/mooc_node-mod2_cmd_iterators

Recuerde que para utilizar el validador se debe tener node.js (y npm) (https://nodejs.org/es/) y Git instalados. El proyecto se descarga, instala y ejecuta en el ordenador local con estos comandos:

```
$                                ## El proyecto debe clonarse en el ordenador local
$ git clone https://github.com/practicas-ging/mooc_node-mod2_cmd_iterators
$
$ cd mooc_node-mod2_cmd_iterators  ## Entrar en el directorio de trabajo
$
$ npm install                    ## Instala el programa de test
$
$                             ##  ->  Incluir la solución en el esqueleto clonado
$
$ npm run checks                           ## Pasa los tests al fichero solicitado
.....................................      ## en el directorio de trabajo
.....................................
... (resultado de los tests)
$
```

Una vez descargado el proyecto, esta entrega se debe realizar de la siguiente forma. Entrar en el directorio raiz **mod2_cmd_iterators**. El fichero **mod2_cmd_iterators.js**, esta incluido (vacío) en el directorio raíz del proyecto descargado. Este debe completarse con el editor o sustituirse por otro del mismo nombre que contenga la solución. A continuación deben pasarse los tests para ver si la solución es correcta.

Los tests pueden pasarse varias veces, incluso con el ejercicio incompleto para probar solo partes de la solución. El programa de test incluye además un comando para generar el fichero ZIP

```
$
$ npm run zip         ##  Comprime los ficheros del directorio en un fichero .zip
$
```

Este genera el fichero **mooc_node-mod2_cmd_iterators_entregable.zip** con el directorio de la practica comprimido. Este fichero ZIP debe subirse a la plataforma para su evaluación.

Instrucciones para la Entrega y Evaluación.

Se debe entregar el fichero **mooc_node-mod2_cmd_iterators_entregable.zip** con los ficheros comprimidos de la entrega.

El evaluador debe descargar el fichero entregado y comprobar que funciona correctamente. El fichero descargado es un paquete npm, que puede instalarse con todas sus dependencias con npm. Una vez instalado puede ejecutarse o pueden pasarse los test.

RUBRICA. Se puntuará el ejercicio a corregir sumando el % indicado a la nota total si la parte indicada es correcta:
- 20%: Extrae bien los parámetros de la matriz de argumentos con multi-asignación
- 10%: Muestra en consola rutas a fichero ejecutable de node y al programa
- 10%: Detecta correctamente si tiene opción -r
- 20%: Elimina correctamente todos los parámetros asociados a opción -r
- 40%: muestra correctamente el número de cada parámetro del resto.

Si pasa todos los tests se deberá dar la máxima puntuación.

El objetivo de este curso es sacar el máximo provecho al trabajo dedicado y para ello lo mejor es utilizar las evaluaciones para ayudar al evaluado, especialmente a los principiantes. Al evaluar se debe dar comentarios sobre la corrección del código, su claridad, legibilidad, estructuración y documentación, siempre que puedan ayudar al evaluado.

¡Cuidado! Una vez enviadas, tanto la entrega, como la evaluación, no se pueden cambiar. Esperar a tener completa y revisada, tanto la entrega, como la evaluación antes de enviarlas.