

Entrega 1 - JavaScript básico

Versión: 12 de Febrero de 2019

Objetivo

Practicar con las sentencias (variables, asignaciones, if/else, bucles,...), los tipos (number, string, boolean) y los objetos básicos (console, Date, ..) de JavaScript.

Descripción de la práctica

Realizar un programa **mod1_types_sentences.js** para node.js con los siguientes requisitos. Incluir el código en el fichero vacío **mod1_types_sentences.js** que se incluye con el proyecto de prueba: https://github.com/practicas-ging/mooc_node-mod1_types_sentences (ver sección "Realización y prueba de la práctica"). El programa debe mostrar por consola:

- 1) Primera línea en blanco
- 2) Línea con el siguiente saludo:

"Good morning/afternoon/night, it's xxx o'clock"

Utilizar if...else y el método getHour() del objeto Date para obtener el saludo:

- getHour() devuelve la hora del día: 0-23. El saludo debe ser: Buenas noches (22h-6h), Buenos días (7h-12h) y Buenas tardes (13h-22h).

- 3) Una línea en blanco
- 4) En la siguiente línea: Number PI with 6 decimals: (Pi con 6 decimales)
- 5) Una línea en blanco
- 6) La tabla con los números enteros de 0 a 22 en decimal, hexadecimal, octal y binario:
0 dec = 0 hex = 0 oct = 0 bin
1 dec = 1 hex = 1 oct = 1 bin

.....

21 dec = 15 hex = 25 oct = 10101 bin

22 dec = 16 hex = 26 oct = 10110 bin

Cada línea de la tabla debe generarse con una función que recibe solo el número decimal como parámetro y muestra por consola la línea con el valor representado en decimal, hexadecimal, octal y binario. Utilizar una plantilla de string ES6 para generar la línea.

- 7) Una línea en blanco
- 8) Una tabla similar de 1 a 21, pero solo con los impares no incluidos en el intervalo de 10 a 20:
1 dec = 1 hex = 1 oct = 1 bin
3 dec = 3 hex = 3 oct = 11 bin
.....
9 dec = 9 hex = 11 oct = 10101 bin
21 dec = 15 hex = 25 oct = 10101 bin

Realizar el programa añadiendo una sentencia if...else en el bucle anterior, de forma que solo se muestren por consola los números que cumplan el nuevo requisito.

- 9) Una línea en blanco

10) La siguiente frase utilizando caracteres escapados para representar hola en chino:

Hi in Chinese is written as: 嗨, 你好吗.

嗨, 你好吗 se corresponde con los siguientes códigos UNICODE escapados:

\u55e8\u54c4\u4f60\u597d\u5417

11) Una línea en blanco

12) Finalizar con la siguiente línea:

"The program has finished"

El programa puede ejecutarse con:

```
$  
$ node mod1_types_sentences.js    ## Ejecuta el programa node*  
$
```

*Nota: En algunas pruebas en Windows los caracteres chinos no se veían correctamente por consola porque no estaba configurado UNICODE como código del S.O.

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/practicass-ging/mooc_node-mod1_types_sentences

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/practicass-ging/mooc_node-mod1_types_sentences  
$  
$ cd mooc_node-mod1_types_sentences    ## 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 raíz **mod1_types_sentences**. El fichero **mod1_types_sentences.js**, esta incluido (vacío) en dicho directorio raíz del proyecto descargado. Este fichero 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-mod1_types_sentences_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-mod1_types_sentences_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.

RÚBRICA: La nota se calcula solo con los puntos 2, 4, 6, 8 y 10, que darán un 20% de la nota total cada uno, si son correctos.

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.