JavaScript

Homework

**Variables**

Una variable es una forma de almacenar el valor de algo para usar más tarde.

Para crear una variable en JavaScript necesitamos la palabra clave**var**,seguida de un **espacio** y el **nombre** de la variable (con este nombre podremos hacer referencia a ella más adelante). Además de declarar una variable, podemos **asignarle un valor** usando el signo **=.**

**Las palabras claves o keywords** son palabras especiales que utiliza el lenguaje para indicar algo. Por lo que **NO** podremos utilizar las palabras claves del lenguaje cómo nombres de variables.

**var**

Es la forma declarar una variable en ES5 (ES5 es la versión de JS, hoy en día existe ES6 que es la nueva versión, pero que todavía no es la más usada). Esta es una ***palabra clave*** genérica para **"variable".**

Existen tres formas de declarar una variable, pero la que vamos a utilizar es var, las otras dos maneras también son validas pero las vamos a usar mas adelante en el curso.

**Strings**

Las "strings" son bloques de texto, siempre se definirán entre **comillas**, ya sea simple o doble. Cualquier texto entre comillas es una cadena o string.

**var nombrePerro = 'firulais';**

**Introducción a las Funciones**

Las funciones son una parte muy importante de todo lenguaje de programación y sobre todo en JavaScript. Son tipos particulares de objetos, llamados ***callable objects*** u objetos invocables.

Una función comenzará con la palabra clave **functión,** esto le dice a lo que sea que esté ejecutando tu programa que lo que sigue es una función y que debe tratarse como tal. Después de eso viene el **nombre de la función**, nos gusta dar nombres de funciones que describan lo que hacen. Luego viene un **paréntesis abierto y uno cercano**. Y finalmente, **abra y cierre los corchetes.** Entre estos corchetes es donde irá **todo** nuestro código a ejecutar.

**function logHola () {**

**console.log(¡’hola!’);**

**}**

**logHola ();**

En este ejemplo declaramos una función **logHola**y la configuramos en console.log('hola'). Entonces podemos ver que, para ejecutar esta función, necesitamos escribir el nombre y los paréntesis. Esta es la sintaxis para ejecutar una función. Una función **siempre** necesita **paréntesis** para ejecutarse.

**Argumentos**

Los argumentos son los***valores realmente suministrados a un procedimiento cuando éste es invocado o llamado.***

Ahora que podemos ejecutar una función básica, vamos a comenzar a pasarle argumentos.

**function logHola(nombre) {**

**console.log ('Hola, ' + nombre);**

**}**

**logHola('Tomas');**

Si agregamos una variable a los paréntesis cuando declaramos la función, podemos usar esta variable dentro de nuestra función. Iniciamos el valor de esta variable pasándola a la función cuando la llamamos. Entonces en este caso nombre = **'Tomas'.**

Podemos agregar múltiples argumentos colocando una coma entre ellos:

**function sumarDosNumeros (a, b) {**

**var suma = a + b;**

**return suma;**

**}**

**sumarDosNumeros (5, 4); // 9**

**return**

En el ejemplo anterior presentamos la declaración return. **NO** vamos a usar **console.log** con todo lo que salga de una función. Lo más probable es que queramos devolver algo. En este caso es la suma de los dos números. Piense en la declaración de retorno ("return") como la única forma en que los datos escapan de una función. No se puede acceder a nada más que a lo que se devuelve fuera de la función. Además, tome en cuenta que cuando una función golpea una declaración de retorno, la función detiene inmediatamente lo que está haciendo y "devuelve" lo especificado.

**function dividirDosNumeros (a, b) {**

**var producto = a / b;**

**return producto;**

**}**

**dividirDosNumeros (45, 9); // 5**

**declaraciones if**

El flujo de control ("control Flow") es una forma de que nuestra función verifique si algo es true, y ya sea ejecutando el código suministrado si es así o avanzando si no lo es. Para esto usaremos la palabra clave **if:**

**function puedebeber(edad) {**

**if (edad > 18) {**

**return true;**

**}**

**return false;**

**}**

**puedebeber (25); // true**

Aquí estamos tomando un número (edad) y verificando si la declaración es true (25>18), lo es, por lo que devolveremos **true**, y la función se detendrá. Si no es así, omitirá ese código y la función devolverá **false.**

**Valores Booleanos**

Los booleanos provienen de la [lógica de Boole](https://es.wikipedia.org/wiki/%C3%81lgebra_de_Boole). Es un concepto que alimenta el código binario y el núcleo de las computadoras.

Esencialmente significa que tiene dos opciones, activar o desactivar, 0 o 1, verdadero o falso. En JavaScript utilizamos booleanos para significar **verdadero o falso.**

Esto puede parecer simple al principio, pero puede complicarse más adelante.

**var meEncantalapasta = true;**

Los valores posibles de un dato booleando en JS son: **true o false.**