



Programación con Javascript

Unidad 2: Variables y operadores



Indice

Unidad 2: Variables y operadores

- Tipos de variables





Objetivos

Que el alumno logre:

- Conocer los tipos de variables y los operadores



Tipos de variables

Antes de comenzar a desarrollar programas y utilidades con JavaScript, es necesario conocer los elementos básicos con los que se construyen las aplicaciones.

Variables

Una variable es un elemento que se emplea para almacenar y hacer referencia a otro valor. Gracias a las variables es posible crear "programas genéricos", es decir, programas que funcionan siempre igual independientemente de los valores concretos utilizados.

De la misma forma que si en Matemáticas no existieran las variables no se podrían definir las ecuaciones y fórmulas, en programación no se podrían hacer programas realmente útiles sin las variables.

Sin el uso de datos variables, un programa que suma dos números podría definirse como:

```
resultado = 4 + 2
```

El programa anterior sólo sirve para el caso en el que el primer número de la suma sea el 4 y el segundo número sea el 2. En cualquier otro caso, el programa obtiene un resultado incorrecto.

Sin embargo, el programa se puede rehacer de la siguiente manera utilizando variables para almacenar y referirse a cada número:

```
primer_numero = 4  
segundo_numero = 2  
resultado = primer_numero + segundo_numero
```

Los elementos `primer_numero` y `segundo_numero` son variables que almacenan los valores que utiliza el programa.

El resultado se calcula siempre en función del valor almacenado por las variables, por lo que este programa funciona correctamente para cualquier par de números indicado. Si se modifica el valor de las variables `primer_numero` y `segundo_numero`, el programa sigue funcionando correctamente.

Las variables en JavaScript se crean mediante la palabra reservada `var`. De esta forma, el ejemplo anterior se puede realizar en JavaScript de la siguiente manera:



```
var primer_numero = 4  
var segundo_numero = 2  
var resultado = primer_numero + segundo_numero
```

El nombre de una variable también se conoce como identificador y debe cumplir las siguientes normas:

- Sólo puede estar formado por letras, números y los símbolos \$ (dólar) y _ (guión bajo).
- El primer carácter no puede ser un número.

La palabra reservada `var` solamente se debe indicar al definir por primera vez la variable, lo que se denomina declarar una variable. Cuando se utilizan las variables en el resto de instrucciones del script, solamente es necesario indicar su nombre. En otras palabras, en el ejemplo anterior sería un error indicar lo siguiente:

```
var primer_numero = 4  
var segundo_numero = 2  
var resultado = var primer_numero + var segundo_numero
```

Si cuando se declara una variable se le asigna también un valor, se dice que la variable ha sido inicializada.

En JavaScript no es obligatorio inicializar las variables, ya que se pueden declarar por una parte y asignarles un valor posteriormente. Por tanto, el ejemplo anterior se puede rehacer de la siguiente manera:

```
var primer_numero; var segundo_numero; var resultado  
primer_numero = 4; segundo_numero = 2  
resultado = primer_numero + segundo_numero
```



Tipos de variables

Cuando declaramos una variable en Javascript no es necesario determinar el tipo de dato a almacenar, sin embargo, de acuerdo al contenido, podemos identificar distintos tipos de variables:

Numéricas

Se utilizan para almacenar valores numéricos enteros (llamados integer en inglés) o decimales (llamados float en inglés).

En este caso, el valor se asigna indicando directamente el número entero o decimal. Los números decimales utilizan el carácter . (punto) en vez de , (coma) para separar la parte entera y la parte decimal:

```
var descuento = 65      // variable tipo entero
var total = 652.53      // variable tipo decimal
```

Cadenas de texto

Se utilizan para almacenar caracteres, palabras y/o frases de texto.

Para asignar el valor a la variable, se encierra el valor entre comillas dobles o simples, para delimitar su comienzo y su final:

```
var mensaje = "Bienvenido a nuestro sitio web"
var nombreProducto = 'Producto ABC'
var letraSeleccionada = 'c'
```

En ocasiones, el texto que se almacena en las variables no es tan sencillo. Si por ejemplo el propio texto contiene comillas simples o dobles, la estrategia que se sigue es la de encerrar el texto con las comillas (simples o dobles) que no utilice el texto:

```
/* El contenido de texto1 tiene comillas simples, por lo que se encierra con comillas dobles
*/
var texto1 = "Una frase con 'comillas simples' dentro"
/* El contenido de texto2 tiene comillas dobles, por lo que se encierra con comillas simples
*/
var texto2 = 'Una frase con "comillas dobles" dentro'
```

No obstante, a veces las cadenas de texto contienen tanto comillas simples como dobles. Además, existen otros caracteres que son difíciles de incluir en una variable de texto (tabulador, ENTER, etc.)



Para resolver estos problemas, JavaScript define un mecanismo para incluir de forma sencilla caracteres especiales y problemáticos dentro de una cadena de texto.

El mecanismo consiste en sustituir el carácter problemático por una combinación simple de caracteres.

A continuación se muestra la tabla de conversión que se debe utilizar:

Si se quiere incluir...	Se debe incluir...
Una nueva línea	\n
Un tabulador	\t
Una comilla simple	\'
Una comilla doble	\"
Una barra inclinada	\\

De esta forma, el ejemplo anterior que contenía comillas simples y dobles dentro del texto se puede rehacer de la siguiente forma:

```
var texto1 = 'Una frase con \'comillas simples\' dentro';  
var texto2 = "Una frase con \"comillas dobles\" dentro";
```

Este mecanismo de JavaScript se denomina "mecanismo de escape" de los caracteres problemáticos, y es habitual referirse a que los caracteres han sido "escapados".

Booleanos

Las variables de tipo boolean o booleano también se conocen con el nombre de variables de tipo lógico.

Una variable de tipo boolean almacena un tipo especial de valor que solamente puede tomar dos valores: true (verdadero) o false (falso). No se puede utilizar para almacenar números y tampoco permite guardar cadenas de texto.

Los únicos valores que pueden almacenar estas variables son true y false, por lo que no pueden utilizarse los valores verdadero y falso. Ejemplo:

```
var clienteRegistrado = false;  
var ivaIncluido = true;
```



Resumen

En esta Unidad...

Trabajamos con variables y operadores

En la próxima Unidad...

Trabajaremos con vectores y ciclos