APRENDERAPROGRAMAR.COM

TIPOS DE DATOS EN JAVASCRIPT. TIPOS PRIMITIVOS Y OBJETO. SIGNIFICADO DE UNDEFINED, NULL, NAN (CU01112E)

Sección: Cursos
Categoría: Tutorial básico del programador web: JavaScript desde cero

Fecha revisión: 2029

Resumen: Entrega nº12 del Tutorial básico “JavaScript desde cero”.

Autor: César Krall

TIPOS DE VARIABLES EN JAVASCRIPT

JavaScript permite trabajar con variables, como es habitual en los lenguajes de programación. Además permite trabajar con objetos, un tipo de información más compleja que una variable simple. De forma general podríamos referirnos a tipos de datos en JavaScript, y dentro de esos tipos de datos tenemos dos grupos fundamentales: los tipos primitivos y los tipos objeto.

TIPOS DE DATOS EN JAVASCRIPT

Los tipos de datos JavaScript se dividen en dos grupos: tipos primitivos y tipos objeto.

Los tipos primitivos incluyen: cadenas de texto (**String**), variables booleanas cuyo valor puede ser true o false (**Boolean**) y números (**Number**). Además hay dos tipos primitivos especiales que son **Null** y **Undefined**. Es en los tipos primitivos donde vamos a centrarnos por el momento.

Los tipos objeto son datos interrelacionados, no ordenados, donde existe un nombre de objeto y un conjunto de propiedades que tienen un valor. Un objeto puede ser creado específicamente por el programador. No obstante, se dice que todo aquello que no es un tipo primitivo es un objeto y en este sentido también es un objeto, por ejemplo, una función. Lo estudiaremos más adelante.

Existen algunos **objetos especiales** con un comportamiento que estudiaremos por separado más adelante: el "objeto global", el "objeto prototipo", los arrays, las funciones, las clases definidas por el programador, las clases predefinidas JavaScript (como la clase Date para manejo de fechas, la clase RegExp para manejo de expresiones regulares y búsqueda de patrones en texto, y la clase Error para almacenar información relacionada con los errores) y otros.

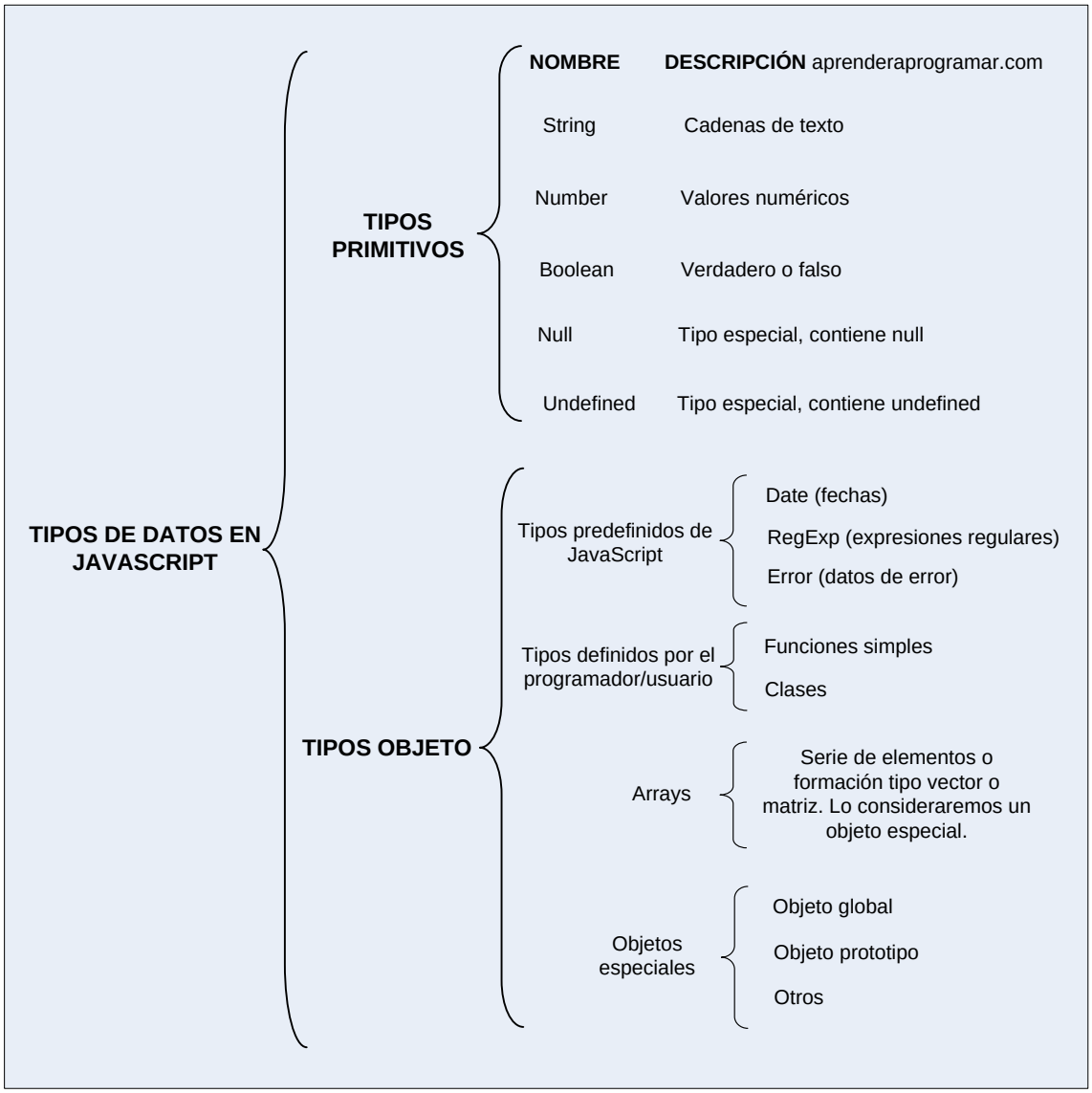
Los primeros lenguajes de programación no usaban objetos, solo variables. Una variable podríamos decir que es **un espacio de la memoria** del ordenador a la que asignamos un contenido que puede ser básicamente un valor numérico (sólo números, con su valor de cálculo) o de tipo carácter o cadena de caracteres (valor alfanumérico que constará sólo de texto o de texto mezclado con números).

Como ejemplo podemos definir una variable a que contenga 32 y esto lo escribimos como a = 32. Posteriormente podemos cambiar el valor de a y hacer a = 78. O hacer "a" equivalente al valor de otra variable "b" así: a = b.

Dado que antes hemos dicho que un objeto también ocupa un espacio de memoria: ¿en qué se parecen y en qué se diferencia un objeto de una variable? Consideraremos que las variables son entidades elementales: un número, un carácter, un valor verdadero o falso… mientras que los objetos son entidades complejas que pueden estar formadas por mucha información. Pero ambas cosas ocupan lo mismo: un espacio de memoria (que puede ser más o menos grande).

© aprenderaprogramar.com, 2006-2029

El siguiente esquema es un resumen sobre tipos de datos en JavaScript.



No te preocupes ahora por conocer todos los posibles tipos de datos de JavaScript ni su significado. Los iremos estudiando y trabajando con ellos poco a poco.

Frente a otros lenguajes fuertemente tipados (las variables tienen un tipo declarado y no pueden cambiar el tipo de contenido que almacenan) como Java, JavaScript es un un lenguaje **débilmente tipado**: las variables pueden no ser de un tipo específico y cambiar el tipo de contenido que almacenan.

© aprenderaprogramar.com, 2006-2029

CONCEPTOS BÁSICOS SOBRE VARIABLES

Una variable se declara con la palabra clave **var**. Por ejemplo **var precio;** constituye la declaración de una variable denominada precio. En la declaración no figura qué tipo de variable es (por ejemplo si es texto tipo String o si es numérica tipo Number). Entonces, ¿cómo se sabe de qué tipo es una variable? JavaScript decide el tipo de la variable **por inferencia**. Si detecta que contiene un valor numérico, la considerará tipo Number. Si contiene un valor de tipo texto la considerará String. Si contiene true ó false la considerará booleana.

El nombre de una variable deberá estar formado por letras, números, guiones bajos o símbolos dólar ($), **no siendo admitidos otros símbolos**. El nombre de la variable no puede empezar por un número: obligatoriamente ha de empezar con una letra, un signo dólar o un guión bajo. Por tanto son nombres de variables válidos precio, $precio, _precio_ , _$dato1, precio_articulo, etc. y no son nombres válidos 12precio ni precio# ó pre!dato1.

Una variable se inicializa cuando se establece su contenido o valor por primera vez. Por ejemplo **precio = 22.55;** puede ser una forma de inicializar una variable.

Una variable se puede declarar e inicializar al mismo tiempo. Por ejemplo podríamos escribir **var precio = 22.55;** con lo que la variable ha sido declarada e inicializada en una misma línea.

JavaScript **no requiere declaración del tipo** de las variables, e incluso permite que una variable almacene contenido de distintos tipos en distintos momentos. Por ejemplo podríamos usar precio = 22.55; y en un lugar posterior escribir precio = 'muy caro';. Esto, que en otros lenguajes generaría un error, es aceptado por JavaScript.

JavaScript **distingue entre mayúsculas y minúsculas** (no sólo para las variables): por tanto no es lo mismo precio = 22.55 que Precio = 22.55. Precio es una variable y precio otra.

JavaScript permite hacer uso de una variable **sin que haya sido declarada**. En muchos lenguajes de programación es necesario declarar una variable antes de poder hacer uso de ella, pero JavaScript no obliga a ello. Cuando JavaScript se encuentra una variable no declarada, crea automáticamente una variable y permite su uso. Hablaremos de ello más adelante, de momento usaremos variables declaradas.

El uso inadecuado de la palabra clave var puede generar un error ante el que JavaScript deja de funcionar, es decir, la página web se visualizará correctamente pero **JavaScript no se ejecuta**. Un ejemplo de código erróneo podría ser este:

```
var precio1=44.20; alert ('La variable ahora es ' + var precio1);
```

var se usa únicamente para declarar una variable. No puede usarse para otra cosa. Una vez declarada la variable, ya se hará uso de ella sin precederla de la palabra clave var. Si se declara una variable estando ya declarada, JavaScript intentará continuar (y posiblemente lo consiga), pero esto puede considerarse una mala práctica excepto si se sabe muy bien lo que se está haciendo.

© aprenderaprogramar.com, 2006-2029

Veamos un ejemplo. Escribe este código y guárdalo en un archivo de extensión html (puedes cambiar la ruta de la imagen si quieres):

```
<html>
<head>
<title>Portal web - aprenderaprogramar.com</title> <meta charset="utf-8">
<script type="text/javascript">
function mostrarMensaje1() {
alert('La variable precio vale: ' + precio);
dato1 = null;
alert('La variable dato1 vale: ' + dato1);
var precio = 22.55;
precio = precio + 10;
alert('La variable precio vale: ' + precio);
alert('El doble de precio es: ' + (precio*2));
cantidad = 10;
alert('El importe resultante de multiplicar precio por cantidad es: ' + (precio*cantidad));
precio = 'muy caro';
alert('La variable precio vale ahora: ' + precio);
alert('El doble de precio es ahora: ' + (precio*2));
var precio = 99.55
alert('La variable precio ha sido declarada por segunda vez y ahora vale: ' + precio);
var $descuento_aplicado = 0.55;
alert('La variable $descuento_aplicado vale : ' + $descuento_aplicado);
}
</script>
</head>
<body>
<div>
<p>Aquí un párrafo de texto situado antes de la imagen, dentro de un div contenedor</p>
<img onclick="mostrarMensaje1()" src="http://i.imgur.com/afC0L.jpg" alt="Notepad++" title="Notepad++,
un útil editor de texto">
<p style="background-color:yellow;" onclick="mostrarMensaje2()">Aquí otro párrafo de texto. JavaScript es
un lenguaje utilizado para dotar de efectos dinámicos a las páginas web.
</p>
</div>
</body>
</html>
```

Visualiza el resultado y comprueba que la página web se muestra con normalidad y que JavaScript se ejecuta con normalidad cuando pulsas sobre la imagen.

El resultado esperado es que se muestre lo siguiente:

La variable precio vale: undefined (Aceptar)
La variable dato1 vale: null (Aceptar)
La variable precio vale: 32.55 (Aceptar)
El doble de precio es: 65.1 (Aceptar)
El importe resultante de multiplicar precio por cantidad es: 325.5 (Aceptar)
La variable precio vale ahora: muy caro (Aceptar)
El doble de precio es ahora: NaN (Aceptar)
La variable precio ha sido declarada por segunda vez y ahora vale: 99.55 (Aceptar)
La variable $descuento_aplicado vale: 0.55 (Aceptar)

© aprenderaprogramar.com, 2006-2029

SIGNIFICADO DE UNDEFINED, NULL Y NAN

De este ejemplo debemos destacar lo siguiente:

a) Se cumplen los **conceptos básicos** explicados.
b) El contenido de una variable no inicializada es **undefined**. En este caso decimos que la variable es de tipo Undefined.
c) El contenido de una variable puede ser **null** y en ese caso decimos que la variable es de tipo Null.
d) Si intentamos realizar una operación matemática con una variable cuyo contenido no es numérico sino texto, la variable toma el valor **NaN**. Para JavaScript NaN (abreviatura de "Not-a-Number") es un valor numérico especial, que representa "número ilegal o no representable". La asignación de NaN que realiza JavaScript automáticamente cuando se intentan realizar operaciones numéricas ilegales evita la aparición de errores explícitos o que el código JavaScript deje de ejecutarse.

EJERCICIO

Crea un código JavaScript donde se genere un error por un mal uso de la palabra clave var y comprueba la respuesta del navegador.

Crea una variable y asignale los contenidos true y false, e intenta multiplicarlas por un número (por ejemplo por 2). ¿Qué resultados obtienes? ¿A qué crees que se deben estos resultados?

Para comprobar si es correta tu solución puedes consultar en los foros aprenderaprogramar.com.

Próxima entrega: CU01113E

Acceso al curso completo en aprenderaprogramar.com -- > Cursos, o en la dirección siguiente: http://aprenderaprogramar.com/index.php?option=com_content&view=category&id=78&Itemid=206

© aprenderaprogramar.com, 2006-2029