

Array

Objeto Array:

El objeto **Array** permite crear matrices para guardar valores de diferentes tipos en una sola variable.

Consta de las siguientes propiedades:

Consta de las siguientes propiedades:

Propiedad	Descripción
constructor	Devuelve la función que ha creado el objeto
prototype	Permite añadir propiedades y métodos al objeto creado
length	Devuelve la longitud del objeto

Consta a su vez de los siguientes métodos:

Método	Descripción
<code>concat()</code>	Une dos o más matrices y devuelve el resultado
<code>indexOf()</code>	Busca en la matriz un elemento y devuelve su posición
<code>join()</code>	Une todos los elementos de la matriz y devuelve una cadena
<code>lastIndexOf()</code>	Busca en la matriz el último elemento y devuelve su posición
<code>pop()</code>	Borra el último elemento de la matriz y lo devuelve
<code>push()</code>	Añade un elemento al final de la matriz y devuelve su longitud
<code>reverse()</code>	Invierte el orden de los elementos de una matriz

Método	Descripción
<code>shift()</code>	Borra el primer elemento de la matriz y lo devuelve
<code>slice()</code>	Selecciona una parte de la matriz y la devuelve como otra matriz
<code>sort()</code>	Ordena los elementos de la matriz
<code>splice()</code>	Añade/borra elementos de la matriz
<code>toString()</code>	Convierte un matriz en una cadena
<code>unshift()</code>	Añade elementos al principio de la matriz y devuelve su resultado
<code>valueOf()</code>	Devuelve el valor original del objeto

String

STRING:

El objeto **String** se emplea para manipular cadenas de caracteres principalmente.

Existen varias formas de declarar cadenas:

```
var cadena1 = new String("Creando una cadena");
```

```
var cadena2 = "Creando otra cadena";
```

Dispone de las siguientes propiedades:

Propiedad	Descripción
constructor	Devuelve la función que ha creado el objeto
prototype	Permite añadir propiedades y métodos al objeto creado
length	Devuelve la longitud de la cadena

Dispone de los siguientes métodos:

Método	Descripción
<code>charAt(indice)</code>	Devuelve el caracter especificado por <i>indice</i>
<code>charCodeAt(indice)</code>	Devuelve el valor Unicode del carácter especificado por <i>indice</i>
<code>concat()</code>	Une dos o más cadenas, y devuelve el resultado
<code>fromCharCode()</code>	Convierte valores Unicode a caracteres
<code>indexOf()</code>	Devuelve la primera posición de la cadena especificada
<code>lastIndexOf()</code>	Devuelve la última posición de la cadena especificada
<code>localeCompare()</code>	Compara dos cadenas en la localización actual
<code>match()</code>	Busca coincidencias entre una expresión regular y una cadena
<code>replace()</code>	Reemplaza una cadena (o expresión regular) buscada por otra
<code>search()</code>	Busca coincidencias entre una expresión regular y una cadena
<code>slice()</code>	Extrae una parte de una cadena y devuelve el resultado
<code>split()</code>	Divide una cadena en una matriz de cadenas

<code>substr()</code>	Extrae caracteres de una cadena, desde un inicio [y longitud]
<code>substring()</code>	Extrae caracteres entre dos índices especificados
<code>toLocaleLowerCase()</code>	Convierte a minúsculas, según la configuración local del host
<code>toLocaleUpperCase()</code>	Convierte a mayúsculas, según la configuración local del host
<code>toLowerCase()</code>	Convierte a minúsculas
<code>toString()</code>	Devuelve el valor de un objeto String
<code>toUpperCase()</code>	Convierte a mayúsculas
<code>trim()</code>	Elimina espacios sobrantes delante y detrás de la cadena
<code>valueOf()</code>	Devuelve el valor original de un objeto String

Ejemplos:

```
var cadena1 = "¡Hola Mundo!";
```

```
var matrizCadenas = cadena1.split(' ');
```

```
var cadena2 = cadena.replace("Hola", "Hello");
```

```
var posicion = cadena1.search("Mundo");
```

```
var cadena3 = "   Hello World!   ";
```

```
var cadena4 = cadena3.trim(); // Devuelve "Hellow World!"
```

Math

MATH:

El objeto **Math** permite realizar operaciones aritméticas.

Incluye una serie de métodos y valores constantes:

```
// Tomar el valor de la constante Pi
```

```
var x = Math.PI;
```

```
// Calcular la raíz cuadrada de 16
```

```
var y = Math.sqrt(16);
```

```
// Obtener el seno del ángulo Pi
```

```
var z = Math.sin(x)
```

Consta de las siguientes propiedades:

Propiedad	Descripción
E	Devuelve el número de Euler
LN2	Devuelve el logaritmo neperiano de 2
LN10	Devuelve el logaritmo neperiano de 10
LOG2E	Devuelve el logaritmo en base 2 del número E
LOG10E	Devuelve el logaritmo en base 10 del número E
PI	Devuelve el número Pi
SQRT1_2	Devuelve la raíz cuadrada de 1 entre 2
SQRT2	Devuelve la raíz cuadrada de 2

Consta de los siguientes métodos:

Método	Descripción
<code>abs(x)</code>	Devuelve el valor absoluto de x
<code>acos(x)</code>	Devuelve el arcocoseno de x en radianes
<code>asin(x)</code>	Devuelve el arcoseno de x en radianes
<code>atan(x)</code>	Devuelve el arcotangente de x en radianes
<code>atan2(y,x)</code>	Devuelve el arcotangente del cociente de dos números
<code>ceil(x)</code>	Redondea al número entero más próximo por exceso
<code>cos(x)</code>	Devuelve el coseno de x en radianes
<code>exp(x)</code>	Devuelve el número E elevado a x
<code>log(x)</code>	Devuelve el logaritmo neperiano de x
<code>max(x,y,z,...,n)</code>	Devuelve el valor máximo de un conjunto de valores
<code>min(x,y,z,...,n)</code>	Devuelve el valor mínimo de un conjunto de valores
<code>pow(x,y)</code>	Devuelve el valor de x elevado a y
<code>random()</code>	Devuelve un número aleatorio entre 0 y 1
<code>round(x)</code>	Redondea al número entero más próximo
<code>sin(x)</code>	Devuelve el seno de x en radianes
<code>sqrt(x)</code>	Devuelve la raíz cuadrada de x

Ejemplos:

```
var e = Math.E; // Copie el valor del número E
```

```
document.writeln(Math.round(4.7)); // Escribe 5
```

```
document.writeln(Math.random()); // Escribe un número en 0 y 1
```

```
document.writeln(Math.floor(Math.random() * 11));
```

```
document.writeln((1 - 0.9) * 10); // Escribe 0.9999999999999998
```


Number

NUMBER:

El objeto **Number** se usa para almacenar valores numéricos simples.

La asignación de un número a una variable se puede hacer de dos formas diferentes:

```
var n = new Number(100);
```

```
var m = 300;
```

```
var x = new Number(1.122334);
```

```
var y = 2.132;
```

Consta de las siguientes propiedades:

Propiedad	Descripción
constructor	Devuelve la función que ha creado el prototipo de objeto Number
MAX_VALUE	Devuelve el mayor número que puede manipular el JavaScript
MIN_VALUE	Devuelve el menor número que puede manipular el JavaScript
NEGATIVE_INFINITY	Representa el infinito negativo
NaN	Representa el valor "Not-a-Number" (no es un número)
POSITIVE_INFINITY	Representa el infinito positivo
prototype	Permite añadir propiedades y métodos a un objeto

Consta de los siguientes métodos:

Método	Descripción
<code>toExponential(x)</code>	Convierte un número a su notación exponencial
<code>toFixed(x)</code>	Formatea un número con un número x de decimales.
<code>toPrecision(x)</code>	Formatea un número a x dígitos
<code>toString()</code>	Convierte un objeto Number a cadena de caracteres
<code>valueOf()</code>	Devuelve el valor original de un objeto Number

Ejemplos:

```
document.writeln(n.toFixed(5)); // Escribe con 5 decimales
```

```
var s = m.toString(); // Convierte el valor de m en cadena
```