

## String

- En JavaScript cualquier cadena de caracteres es un String.
- Con *length* obtenemos el número de caracteres que componen la cadena

```
var texto = 'pepe';  
alert (texto.length) // 4
```

- Podemos acceder a cada carácter de un string con su posición (índice), el primer carácter se encuentra en la posición 0.

```
alert (texto[2]) // p
```

Los métodos más interesantes del objeto String

- **concat(*cadena1*,...,*cadenaN*)**
  - Concatena *cadena1*, ..., *cadenaN* a continuación de la cadena pero la cadena no se queda modificada

```
var otro = 'luis';  
alert (texto.concat(otro)); // pepeluis  
alert (texto); // pepe
```

- **indexOf(*aguja*[,*inicio*])**
  - Devuelve la primera posición desde la izquierda en la que se encuentra la cadena *aguja* dentro de la cadena empezando a buscar por la posición *inicio*.
  - Si no se encuentra devuelve -1.

```
var texto='pepe';  
alert (texto.indexOf('e')); // 1  
alert (texto.indexOf('e',2)); // 3
```

- **lastIndexOf(*aguja*[,*fin*])**
  - Como el anterior pero buscando desde la derecha y comenzando por *fin*

```
alert (texto.lastIndexOf('e')); // 3
```

- **slice(*inicio*[,*fin*]), substring (*inicio*[,*fin*])**
  - Devuelve los caracteres comprendidos entre *inicio* (incluido) y *fin* (excluido), o hasta el final de la cadena si se omite *fin*.

```
var texto = 'Hola Pepe, qué tal?';  
alert(texto.substring(5,8)); // Pep  
alert (texto. slice(5,8)); // Pep
```

- **substr (*inicio*[,*cantidad*])**
  - Devuelve *cantidad* número de caracteres a partir de *inicio* (incluido). Si se omite *cantidad*, devuelve hasta el final de la cadena.

```
var texto = 'Hola Pepe, qué tal?';
```

```
alert(texto.substr(5,4)); // Pepe
```

- **toLowerCase()**

- Devuelve la cadena convertida en minúsculas.

```
alert(texto.toLowerCase()); // pepe
```

- **toUpperCase()**

- Devuelve la cadena convertida a mayúsculas.

```
alert(texto.toUpperCase()); // PEPE
```

- **trim() , trimLeft() , trimRight()**

- Eliminan los espacios al principio y al final de la cadena, al principio solamente o al final solamente, respectivamente.

```
var texto=" hola ";  
alert(texto.trim()); // "hola"
```

- **split (separador)**

- Devuelve un array cuyos elementos son los fragmentos de la cadena comprendidos entre el carácter o cadena *separador*.

```
var nombres = 'Pepe, Luis, Juan';  
var resultado=nombres.split(',');  
alert (resultado[0]); // Pepe  
alert(resultado[1]); //Luis  
alert(resultado[2]); //Juan
```

- **replace(cadena1, cadena2)**

- Busca una coincidencia en una cadena y si existe la reemplaza por otra cadena pasada por parámetro.

```
var texto = 'hola manola';  
alert (texto.replace('hola','adios')); // adios manola
```

## Date

[http://msdn.microsoft.com/es-es/library/cd9w2te4\(v=vs.94\).aspx](http://msdn.microsoft.com/es-es/library/cd9w2te4(v=vs.94).aspx)

- Sirve para manejar fechas
- En JavaScript las fechas se expresan como el número de milisegundos transcurridos desde el 1 de enero de 1970 a las 00:00:00 UTC (Tiempo Universal Coordinado) Esta fecha se conoce como origen o época UNIX .
- Para crear un objeto de este tipo :

```
var fecha = new Date();  
  
fecha contendrá la fecha y hora del sistema
```

```
var fecha = new Date(milisegundosDesdeEpocaUNIX);
```

construye la fecha correspondiente, se usa poco

```
var fecha = new Date(año,mes,día[,hora,minuto,segundo,milisegundo]);
```

Donde todos los parámetros son números enteros.

año debe expresarse con 4 cifras.

mes es un entero entre 0 (enero) y 11 (diciembre).

día es un entero entre 1 y 31.

hora, minuto, segundo, milisegundo no son obligatorios.

```
var fechaHoy = new Date();  
var fechaNac = new Date(1999,9,7);  
alert (fechaHoy);  
alert (fechaNac);
```

Para saber si una fecha se ha construido correctamente podemos preguntar por `isNaN(Fecha)` de devolverá `false` o `true` dependiendo de que lo haya sido o no.

El objeto *Date* tiene una serie de métodos que pueden dividirse en tres subconjuntos:

- Métodos de lectura: empiezan con el prefijo *get*. Consultan las diferentes partes de una instancia del objeto Date
  - Métodos de escritura: empiezan con el prefijo *set*. Inician o modifican las diferentes partes de una instancia del objeto Date
  - Métodos de conversión: convierten objetos de tipo Date en cadenas de texto o milisegundos
- 
- **getDate()** Devuelve el día del mes entre 1 y 31.
  - **getDay()** Devuelve el día de la semana (0-6). El 0 corresponde al domingo y el 6 al sábado. Atención, el day no se puede establecer, sólo leer; en otras palabras, no existe `setDay()`
  - **getMonth()** Devuelve el mes (0-11).
  - **getFullYear()** Devuelve el año expresado con 4 dígitos.
  - **getHours()** Devuelve la hora (0-23).
  - **getMinutes()** Devuelve los minutos (0-59).

- **getSeconds()** Devuelve los segundos (0-59).
- **getMilliseconds()** Devuelve los milisegundos (0-999).

```
var fechaHoy = new Date();  
alert ('Hoy es '+fechaHoy.getDate()+'', son las '+fechaHoy.getHours()+  
' horas y '+fechaHoy.getMinutes()+' minutos');
```

- **getTime()** Devuelve los milisegundos transcurridos desde la época UNIX hasta la fecha que se aplica

```
alert(fechaHoy.getTime());
```

- **setDate()** Establece el día del mes.
- **setMonth()** Establece el mes (0-11).
- **setFullYear()** Establece el año expresado con 4 dígitos.
- **setHours()** Establece la hora (0-23).
- **setMinutes()** Establece los minutos (0-59).
- **setSeconds()** Establece los segundos (0-59).
- **setMilliseconds()** Establece los milisegundos (0-59).
- **setTime()** Establece la fecha como el número de milisegundos transcurrido desde la época UNIX

```
var fecha=new Date();  
fecha.setDate(8);  
fecha.setMonth(9);  
fecha.setFullYear(2013);  
alert (fecha);
```

- **toString()** Convierte la fecha del objeto Date en una cadena de caracteres
- **toISOString()** Convierte el tiempo del objeto Date en una cadena de caracteres
- **toLocaleDateString()** Devuelve una cadena de la fecha en el formato de idioma local que tenga configurado el cliente.
- **toLocaleTimeString()** Devuelve una cadena de la hora (hora, minuto, segundo) en el formato de idioma local que tenga configurado el cliente.

Ejemplo:

```
Fecha=new Date (2015,9,8,11,20,22);  
alert (Fecha.toString()) ; // Tue Oct 08 2015  
alert (Fecha.toTimeString()) ; // 11:20:22 GMT+0200  
alert (Fecha.toLocaleDateString());// 8/10/2015  
alert (Fecha.toLocaleTimeString());// 11:20:22
```

**VER EJEMPLOS FECHAS ejemplo 01, ejemplo 02, ejemplo 03**

- El objeto **Math** nos permite realizar operaciones matemáticas: raíces cuadradas, logaritmos, operaciones trigonométricas y, muy importante, obtener números pseudo-aleatorios.

[http://msdn.microsoft.com/es-es/library/b272f386\(v=vs.94\).aspx](http://msdn.microsoft.com/es-es/library/b272f386(v=vs.94).aspx)

### Métodos

- **ceil(valor)** Devuelve el entero inmediatamente mayor o igual que valor.
- **floor(valor)** Devuelve el entero inmediatamente menor o igual que valor.
- **max(val1, ..., valN)** Devuelve el máximo de los valores recibidos como argumentos
- **min(val1, ..., valN)** Devuelve el mínimo de los valores recibidos como argumentos
- **pow(base,exponente)** Devuelve el resultado de elevar base a exponente
- **random()** Devuelve un número aleatorio entre 0 y 1, 1 no incluido
- **round(valor)** Devuelve el resultado de redondear valor al entero más próximo
- **sqrt(valor)** Devuelve la raíz cuadrada de valor.
- **abs(valor)** Devuelve el valor absoluto de valor, es decir, su valor sin signo

```
alert (Math.random()*10); // Un número entre el 0 y el 9.9
alert(round(Math.random()*10)); // Un número entre el 0 y el 10
```

```
alert (Math.PI); // 3.1415...
alert (Math.ceil (33.3)); // 34
alert (Math.floor(33.3)); // 33
alert(Math.max(2,6,3,7)); // 7
alert(Math.min(2,6,3,7)); // 2
alert(Math.pow(2,4)); // 16
alert(Math.random());
alert(Math.round(33.3)); // 33
alert(Math.sqrt(25)); // 5
```

Devuelven **NaN** cuando no han podido efectuar el cálculo matemático

### Number

Se utiliza principalmente para indicar el máximo y mínimo valor posible que podemos representar con JavaScript.

### Propiedades

- MAX\_VALUE** Devuelve el mayor número posible en JavaScript
- MIN\_VALUE** Devuelve el menor número posible en JavaScript

```
alert (Number.MAX_VALUE);
alert (Number.MIN_VALUE);
```