



Curso  
Desarrollo de Software

# FRONT END

NIVEL INTERMEDIO



## Tipos de Datos (strings, numbers, boolean)

### Strings

El objeto `string` se utiliza para representar y manipular una secuencia de caracteres.

Las [cadenas](#) son útiles para almacenar datos que se pueden representar en forma de texto. Algunas de las operaciones más utilizadas en cadenas son verificar su `length`, para construirlas y concatenarlas usando operadores de cadena `+` y `+=`, verificando la existencia o ubicación de subcadenas con `indexOf()` o extraer subcadenas con el método `substring()`

### String Concatenation

El [método](#) `concat()` combina dos o más cadenas de texto y devuelve una cadena de texto nueva.

**Actividad:** Introduce los siguientes códigos en tu página HTML y visualízalos en tu navegador.

```
let hello = 'Hello, '

console.log(hello.concat('Kevin', '. Have a nice day.')) //
Hello, Kevin. Have a nice day.

let greetList = ['Hello', ' ', 'Venkat', '!']
"".concat(...greetList) // "Hello Venkat!"

"".concat({}) // [object Object]

"".concat([]) //

"" "".concat(null) // "null"

"".concat(true) // "true"

"".concat(4, 5) // "45"
```

### String length y slicing

La propiedad `length` de un objeto `string` representa la longitud de una cadena, en unidades de código UTF-16.

ECMAScript 2016 (ed. 7) estableció una longitud máxima de  $2^{53} - 1$  elementos. Anteriormente, ninguna longitud máxima era especificada.

Para una cadena vacía, `length` es 0.

La propiedad `static String.length` devuelve 1.

Ejemplo:

```
var myString = "bluebells";

// Attempting to assign a value to a string's .length property
has no observable effect.

myString.length = 4;

console.log(myString);

/* "bluebells" */
```

Por su parte, el método `slice()` extrae una sección de una cadena y devuelve una cadena nueva. Los cambios en el texto de una cadena no afectan a la otra cadena.

Ejemplo: El siguiente ejemplo usa `slice()` para crear una nueva cadena.

```
var cadena1 = "La mañana se nos echa encima.";

var cadena2 = cadena1.slice(3, -2);

console.log(cadena2);
```

## Boolean

El objeto `Boolean` es un objeto contenedor para un valor booleano.

El valor pasado como primer parámetro se convierte en un valor booleano, si es necesario. Si el valor se omite o es 0, -0, `null`, `false`, `NaN`, `undefined`, o la cadena vacía (`""`), el objeto tiene un valor inicial de `false`. Todos los demás valores, incluido cualquier objeto, un arreglo vacío (`[]`) o la cadena `"false"`, crean un objeto con un valor inicial de `true`.

No confundas los valores del `Boolean` primitivo, `true` y `false` con los valores `true` y `false` del objeto `Boolean`.