

# Introducción a JavaScript



Francisco Javier Arce Anguiano

<b>Introducción a JavaScript</b>	<b>1</b>
6. Manejo de cadenas	2
6.1. Concatenación de cadenas y conversión de datos	2
6.2. Métodos para convertir las cadenas a mayúsculas y minúsculas	3
6.3. Buscar subcadenas con el método indexOf	4
6.4. Manejo de subcadenas con subString(), subStr() y slice()	5
6.5. Leer los caracteres de una cadena	6

## 6. Manejo de cadenas

**Objetivo:** El alumno modificará las cadenas y subcadenas, así como leerá los caracteres de una subcadena.

### 6.1. Concatenación de cadenas y conversión de datos

El objeto String ofrece distintas formas de manejar cadenas de texto. Siempre que se asigne un String una variable o propiedad, se crea un objeto de tipo String.

En este caso, al definir una variable con una cadena de texto ya estamos utilizando un objeto String, es decir, no será necesaria su declaración.

```
var cadenaTexto;  
cadenaTexto="Aquí esta la cadena de texto";
```

Los objetos de tipo String disponen de una serie de métodos que permiten modificar y devolver el valor de la cadena de texto.

Los métodos disponibles para este objeto son:

- `anchor(nombre)` Crea un enlace
- `big()` Muestra la cadena de caracteres con una fuente grande.
- `blink()` La cadena se representa intermitentemente.
- `bold()` Muestra la cadena en negrita.
- `charAt()` Muestra el carácter situado en la posición n de la cadena.
- `fixed()` Muestra la cadena con una fuente no proporcional.
- `fontcolor(color)` Determina el color de la cadena.
- `fontsize(n)` Muestra la cadena con una tamaño n. Éste puede oscilar entre 1 y 7.
- `indexOf(smallString,start)` Facilita la posición de un fragmento de la cadena a partir de una posición determinada.
- `italics()` Muestra la cadena en cursiva.

- `lastIndexOf()` Da como resultado la última posición de un carácter.
- `link(URL)` Convierte la cadena en un vínculo.
- `small()` Muestra la cadena con una fuente pequeña.
- `strike()` Muestra la cadena subrayada.
- `sub()` Muestra la cadena en formato subíndice.
- `substring(x,y)` Muestra un fragmento de cadena que empieza en posición x y termina en la posición y.
- `sup()` Muestra la cadena en formato superíndice.
- `toLowerCase()` Muestra toda la cadena en minúsculas.
- `toUpperCase()` Muestra toda la cadena en mayúsculas.

Ejemplo en que se utilizan algunos de estos métodos:

```
var cadena;

cadena="Bienvenidos al apasionante mundo de JavaScript";
document.write(cadena+"<br>");
document.write(cadena.bold()+"<br>");
document.write(cadena.big()+"<br>");
document.write(cadena.toUpperCase()+"<br>");
document.write(cadena.small()+"<br>");
document.write(cadena.italics()+"<br>");
```

Los objetos **String()** disponen de las propiedades `length`, que determina el número de caracteres de la cadena, y `prototype`, que permite añadir nuevas propiedades y métodos (más adelante serán tratadas).

## 6.2. Métodos para convertir las cadenas a mayúsculas y minúsculas

La propiedad `"length"` te indica el número de caracteres o letras.

`toUpperCase()` o `toLowerCase()` convierten una cadena a mayúsculas o minúsculas.

```
var correo1 = "pacoarce@gmail.com.mx";
```

```

var correo2 = "PacoArce@gmail.com.mx";
document.write("La longitud de la clave "+correo1+" es de "+correo1.length+" caracteres<br>");
if (correo1.length<20) {
    document.write("La clave de acceso es muy corta<br>");
} else {
    if (correo1.toUpperCase()==correo2.toUpperCase()) {
        document.write("Las claves coinciden<br>");
    } else {
        document.write("Las claves NO coinciden<br>");
    }
}
}

```

### 6.3. Buscar subcadenas con el método indexOf

El método `indexOf()` busca una subcadena en una cadena. Inicia en y si no encuentra la subcadena, regresa un -1. El método **`indexOf()`** es sensible a mayúsculas y minúsculas. Su sintaxis es:

```
cadena.indexOf(valorBusqueda[, indiceDesde])
```

Ejemplo:

```

var correo1 = "pacoarce@gmail.com.mx";
var correo2 = "PacoArce@gmail.com.mx";
//PacoArce@gmail.com.mx
//012345678
var pos = correo2.indexOf("@");
console.log(pos);
var punto = correo2.indexOf(".",pos);
console.log(punto);
//Primera validación: longitud
if (correo1.length<10) {
    document.write("La clave de acceso es muy corta<br>");
} else if(pos==-1){
    document.write("El correo no tiene la arroba<br>");
}

```

```

} else if(punto== -1){
    document.write("El correo debe de tener al menos un
punto<br>");
} else if (!(correo1.toUpperCase()==correo2.toUpperCase())) {
    document.write("Las claves NO coinciden<br>");
} else {
    document.write("Las claves coinciden<br>");
}

```

#### 6.4. Manejo de subcadenas con subString(), subStr() y slice()

Partimos una cadena de una posición con slice()

```
email.slice(pos+1);
```

substr() divide una cadena de un punto determinado cierto número de caracteres:

```
email.substr(0,pos);
```

Ejemplo:

```

var correo1 = "pacoarce@gmail.com.mx";
var correo2 = "PacoArce@gmail.com.mx";
//PacoArce@gmail.com.mx
//012345678
var pos = correo2.indexOf("@");
console.log(pos);
var punto = correo2.indexOf(".",pos);
console.log(punto);
//Primera validación: longitud
if (correo1.length<10) {
    document.write("La clave de acceso es muy corta<br>");
} else if(pos== -1){
    document.write("El correo no tiene la arroba<br>");
}

```

```

} else if(punto== -1){
    document.write("El correo debe de tener al menos un
punto<br>");
} else if (!(correo1.toUpperCase()==correo2.toUpperCase())) {
    document.write("Las claves NO coinciden<br>");

} else {
    servicio = correo2.slice(pos+1).toLowerCase();
    usuario = correo2.substr(0,pos).toLowerCase();
    document.write("Tu servicio de correo es:
<b>"+servicio+"</b><br>");
    document.write("Tu usuatrio de correo es:
<b>"+usuario+"</b><br>");
    document.write("Las claves coinciden<br>");
}

```

## 6.5. Leer los caracteres de una cadena

La función **charAt** regresa un caracter (i) de una cadena

```
var c = invalidos.charAt(i);
```

En un correo electrónico todos los caracteres son válidos a excepción de:

- las letras acentuadas
- los caracteres de control (CTRL + tecla del teclado)
- los espacios
- los signos especiales como ()<>@,;:[]"{}%&

```

function caracteresInvalidos(cadena){
    var invalidos = "áéíóúÁÍÓÚÑñ ()<>,;:[]"{}%&";
    var bandera = false;
    for(i=0; i<invalidos.length; i++){
        var c = invalidos.charAt(i);
        if(cadena.indexOf(c)!= -1){
            bandera = true;
        }
    }
}

```

```

        break;
    }
}
return bandera;
}
//Variables
var correo1 = "pacoarce@gmail.com.mx";
var correo2 = "Pacoarce@gmail.com.mx";
//PacoArce@gmail.com.mx
//012345678
var pos = correo2.indexOf("@");
console.log(pos);
var punto = correo2.indexOf(".",pos);
console.log(punto);
//Primera validación: longitud
if (correo1.length<10) {
    document.write("La clave de acceso es muy corta<br>");
} else if(pos==-1){
    document.write("El correo no tiene la arroba<br>");
} else if(punto==-1){
    document.write("El correo debe de tener al menos un
punto<br>");
} else if(caracteresInvalidos(correo2)){
    document.write("El correo tiene caracteres inválidos<br>");
} else if (!(correo1.toUpperCase()==correo2.toUpperCase())) {
    document.write("Las claves NO coinciden<br>");

} else {
    servicio = correo2.slice(pos+1).toLowerCase();
    usuario = correo2.substr(0,pos).toLowerCase();
    document.write("Tu servicio de correo es:
<b>"+servicio+"</b><br>");
    document.write("Tu usuatrio de correo es:
<b>"+usuario+"</b><br>");
    document.write("Las claves coinciden<br>");
}

```