

JavaScript Avanzado

Sesión 3



Universidad
Tecnológica
del Perú

¿Tienen alguna consulta o duda sobre la clase previa?



Logro de la sesión

Al finalizar la sesión el estudiante aplica el manejo de cadenas en el lenguaje JavaScript mediante la construcción de aplicativos web.

¿Qué son las cadenas o strings? ¿Qué usos tienen?





```
// ways to make strings

const message = 'Hello'
const greet = message + ' World!' // Hello World!

// String constructor
const str = String('constructor')

// more on coercion in another article
const coercion = '' + 5 // '5' as a string

// when using ' inside a ' wrapped string
// you need to escape it using \
const escapeCharacters = 'This could\'ve been better'
// This could've been better

// or use different wrappers
const noEscape = "This could've been better"

// more on `` template literals later
const evenLessEscape = `This could've been "worse"`
// This could've been "worse"
```

*¿Cuál es la
importancia del
manejo de
cadenas en la
programación?*



Universidad
Tecnológica
del Perú

Contenido

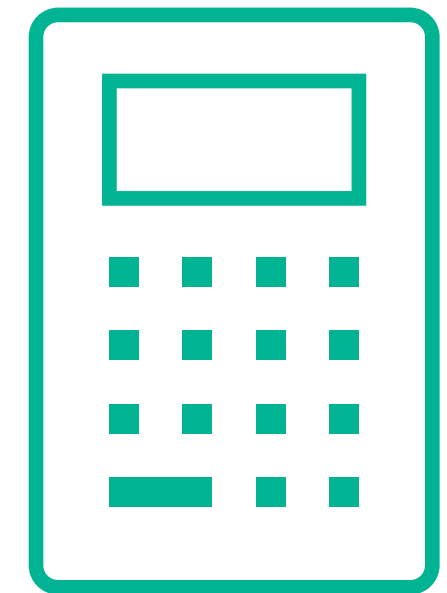
Manejo de cadenas

- Métodos de concatenación
- Extensión de cadenas
- Extraer contenido
- Búsqueda
- Sintaxis con cadenas (comillas, comillas invertidas, multilínea, etc.)
- Interpolación
- Expresiones regulares.



String en JavaScript

- Es una secuencia de caracteres utilizada para representar texto.
- Se puede crear utilizando comillas simples, dobles o acentos graves para encerrar los caracteres.
- Tienen métodos y propiedades que permiten realizar operaciones como concatenación, transformación y búsqueda de subcadenas.



Métodos de concatenación en JavaScript

Operador +

- Es el método más simple para concatenar cadenas.

Método `concat()`

- Este método une dos o más cadenas y devuelve una nueva cadena.

Extensión de cadenas en JavaScript

- La propiedad **length** de un objeto **String** representa la longitud de una cadena, en unidades de código UTF-16.
- UTF-16, el formato usado por JavaScript, usa 16-bit para representar los caracteres más comunes, pero necesita usar dos caracteres para otros menos usados, así que es posible que el valor devuelto por **length** no corresponda al número de caracteres de la cadena.

Extraer contenido

- 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.

```
var cad = "La mañana se nos echa encima.";
cad.slice(-3); // retorna 'ma.'
cad.slice(-3, -1); // retorna 'ma'
cad.slice(0, -1); // retorna 'La mañana se nos echa encima'
```

Búsqueda en cadenas

- **indexOf():** Devuelve el índice de la primera aparición de una subcadena.
- **lastIndexOf():** Devuelve el índice de la última aparición de una subcadena.
- **search():** Busca una subcadena utilizando una expresión regular.
- **includes():** Determina si un string contiene una subcadena especificada.
- **startsWith():** Determina si un string comienza con una subcadena especificada.
- **endsWith():** Determina si un string termina con una subcadena especificada.

Sintaxis con cadenas

- En JavaScript, las cadenas (**strings**) son una secuencia de caracteres utilizados para representar texto.

Es factible definir una cadena usando:

- **Comillas simples** ('...')
- **Comillas dobles** ("...")
- **Comillas invertidas** (`...`) para soportar interpolación y múltiples líneas.

Interpolación

- Es una técnica que permite insertar cadenas o valores en una cadena existente para crear cadenas dinámicas y potentes.
- Por ejemplo, insertar variables y expresiones dentro de cadenas de texto, o para usar un operador ternario dentro de una cadena para verificar el valor de una variable y mostrar diferentes cosas dependiendo de ese valor.

```
let nombre = 'Jhon';  
let apellido = 'Doe';  
let saludo = `Hola, ${nombre} ${apellido}!`; // "Hola, Jhon Doe!"
```



Universidad
Tecnológica
del Perú

Expresiones regulares

- Las expresiones regulares son patrones que se utilizan para hacer coincidir combinaciones de caracteres en cadenas.
- En JavaScript, las expresiones regulares también son objetos.
- Estos patrones se utilizan con los métodos **exec()** y **test()** de **RegExp**, y con **match()**, **matchAll()**, **replace()**, **replaceAll()**, **search()** y **split()** métodos de **String**.

```
<p>
```

```
  Ingresa tu número de teléfono (con el código de área) y luego haz clic en  
  "Comprobar".
```

```
  <br />
```

```
  El formato esperado es como ###-###-####.
```

```
</p>
```

```
<form action="#">
```

```
  <input id="phone" />
```

```
  <button onclick="testInfo(document.getElementById('phone'));">
```

```
    Comprobar
```

```
  </button>
```

```
</form>
```

```
var re = /(?:\d{3}|\(\d{3}\))([-\./])\d{3}\1\d{4}/;
```

```
function testInfo(phoneInput) {
```

```
  var OK = re.exec(phoneInput.value);
```

```
  if (!OK) {
```

```
    console.error(phoneInput.value + " isn't a phone number with area code!");
```

```
  } else {
```

```
    console.log("Gracias, tu número de teléfono es " + OK[0]);
```

```
  }
```

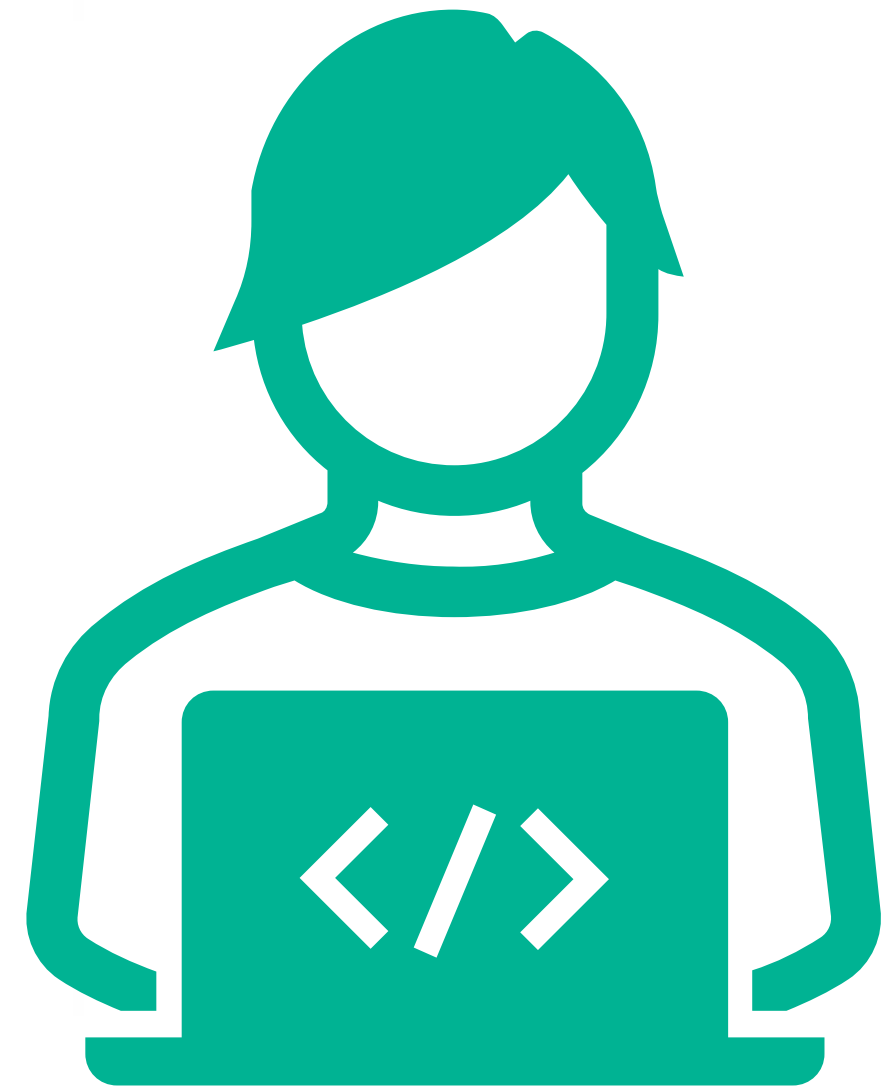
```
}
```



Universidad
Tecnológica
del Perú

Programando

- Construir algoritmos empleando manejos de cadenas en JavaScript.



¿Tienen alguna consulta o duda?



Actividad



Resolver la actividad planteada en la plataforma.

Cierre

¿Qué hemos aprendido hoy?



Elaboramos nuestras conclusiones sobre el tema tratado



**Universidad
Tecnológica
del Perú**