**Interpolación, Operador Ternario y Destructuring en Javascript (ES Modules) – Maria Monchot - Katerina**

Para Complementar la lectura del siguiente documento, recomiendo los siguientes videos:

* <https://www.youtube.com/watch?v=qJSf7YrbDfE&ab_channel=Bluuweb>

Para seguir esta guía, se utilizará un archivo index.html, un archivo index.js y…

¿QUÉ ES INTERPOLACIÓN?

Primero veamos algo de Template Strings:

Las cadenas de texto de JavaScript han sido desde siempre muy limitadas, sobre todo comparadas con las cadenas de lenguajes como PHP, Python o Ruby. Las [Template Strings](https://www.chromestatus.com/feature/4743002513735680) del estándar EcmaScript 6 (que están disponibles a partir de Google Chrome 41) eliminan todas estas limitaciones y permiten incluso definir DSLs (*domain-specific languages*) en tus aplicaciones JavaScript.

Las principales funcionalidades que aportan las *Template Strings* son:

* Interpolación de cadenas.
* Posibilidad de incluir (y evaluar) expresiones dentro de cadenas.
* Definición de cadenas de texto en varias líneas sin tener que usar *hacks*.
* Formatear cadenas de manera avanzada.
* Cadenas etiquetadas.

En vez de añadir todas estas funcionalidades en las cadenas que conocemos y usamos a diario en JavaScript, las *Template Strings* añaden sus funcionalidades de una manera completamente nueva.

**Sintaxis**

Las *Template Strings* utilizan las comillas invertidas o *backticks* para delimitar sus contenidos, en vez de las tradicionales comillas simples o dobles de las cadenas de texto normales. Así que este es un ejemplo de cómo definir una *Template String*:

*// esto es una Template String*

var saludo = `¡Hola Mundo!`;

*// esto es una cadena normal con comillas simples*

var saludo = '¡Hola Mundo!';

*// esto es una cadena normal con comillas dobles*

var saludo = "¡Hola Mundo!";

Aunque de momento las *Template Strings* no parecen gran cosa, sigue leyendo para descubrir todas sus funcionalidades.

**Interpolación de cadenas**

Una de las mejores características de las *Template Strings* es la interpolación de cadenas. En pocas palabras, la interpolación permite utilizar cualquier expresión válida de JavaScript (como por ejemplo la suma de dos variables) dentro de una cadena y obtener como resultado la cadena completa con la expresión evaluada.

Las partes variables de una *Template String* se denominan *placeholders* y utilizan la sintaxis

${ } para diferenciarse del resto de la cadena. Ejemplo:

*// Sustitución simple de cadenas*

var nombre = "Juan";

console.log(`¡Hola ${nombre}!`);

*// resultado => "¡Hola Juan!"*

Como dentro de las partes variables de la cadena se puede incluir cualquier expresión válida de JavaScript, en la práctica sirven para mucho más que mostrar el contenido de una variable. En los siguientes ejemplos se muestran cómo interpolar algunas operaciones matemáticas sencillas:

var a = 10;

var b = 10;

console.log(`¡JavaScript se publicó hace ${a+b} años!`);

*// resultado => ¡JavaScript se publicó hace 20 años!*

console.log(`Existen ${2 \* (a + b)} frameworks JavaScript y no ${10 \* (a + b)}.`);

*// resultado => Existen 40 frameworks JavaScript y no 2000.*

Dentro de un valor interpolado también se puede utilizar cualquier función:

function fn() { return "Este es el resultado de la función"; }

console.log(`Hola "${fn()}" Mundo`);

*// resultado => Hola "Este es el resultado de la función" Mundo*

La sintaxis ${} también funciona con expresiones que invocan métodos y acceden a propiedades:

var usuario = { nombre: 'Juan Perez' };

console.log(`Estás conectado como ${usuario.nombre.toUpperCase()}.`);

*// resultado => "Estás conectado como JUAN PEREZ.";*

var divisa = 'Euro';

console.log(`Los precios se indican en ${divisa}. Utiliza nuestro conversor para convertir ${divisa} en tu moneda local.`);

*// resultado => Los precios se indican en Euro. Utiliza nuestro conversor para convertir Euro en tu moneda local.*

**Cadenas en varias líneas**

Definir una cadena de texto en varias líneas con JavaScript es sencillo, pero requiere de algunos *hacks* que ensucian el código de la aplicación. La solución más utilizada actualmente consiste en añadir una barra invertida antes de cada salto de línea. Ejemplo:

var saludo = "Hola \

Mundo";

Aunque este código funciona perfectamente en cualquier *engine* moderno de JavaScript, no deja de ser un *hack*. Otra forma de definir cadenas en varias líneas consiste en utilizar la concatenación de cadenas, aunque el resultado no es mucho mejor que el anterior:

var saludo = "Hola " +

"Mundo";

Las *Template Strings* simplifican drásticamente la definición de cadenas en varias líneas. De hecho, es tan sencillo que no hay que hacer absolutamente nada: pulsa <Enter> cada vez que quieras incluir un salto de línea en la cadena y ya está. Ejemplo:

var cadena = `Línea número 1 de la cadena

Línea número 2 de la cadena`;

Este código funciona bien porque cualquier espacio en blanco (incluyendo los saltos de línea) que se incluyen dentro de las comillas invertidas se considera que es parte de la cadena.

¿QUÉ ES EL OPERADOR TERNARIO?

¿QUÉ ES DESTRUCTURING?

BIBLIOGRAFÍA:

* <https://uniwebsidad.com/tutoriales/las-nuevas-cadenas-de-texto-de-javascript-6#:~:text=En%20pocas%20palabras%2C%20la%20interpolaci%C3%B3n,completa%20con%20la%20expresi%C3%B3n%20evaluada>.