educación IT
CODE YOUR FUTURE

Node.js Backend

Módulo 2

educación IT
CODE YOUR FUTURE

ES7: ECMAScript 2016

Includes()

Antes, cuando necesitábamos saber si un elemento **existía dentro de un array en JavaScript,** usábamos métodos que no estaban pensados precisamente para esta función.

Con la nueva propuesta ECMAScript 2016 **(ES7)**, ahora tenemos un método para ello: `includes()`.

Este método **determina si un array incluye un elemento determinado.** Según corresponda, **regresará `true` o `false`.**

Sintaxis

```
Array.prototype.includes( searchElement[, fromIndex ])
```

- `SearchElement` es el **elemento a buscar.**
- `FromIndex` es un parámetro opcional que **marca la posición en la matriz a partir de la cual se comienza a buscar dicho elemento.**

```
var myArr = [ 'la', 'donna', 'e', 'mobile', 'cual', 'piuma', 'al', 'vento' ];

console.log( myArr.includes( 'donna' ) ); // true
console.log( myArr.includes( 'pensiero' ) ); // false

console.info( myArr.includes( 'cual', 5 ) ); // false
console.info( myArr.includes( 'donna', 1 ) ); // true
console.info( myArr.includes( 'vento', -1 ) ); // true
```

Como curiosidad, podemos ver que, en el **último ejemplo**, es posible usar un **índice negativo para que el conteo inicie desde la derecha.**

Actuando sobre cadenas

Además de trabajar sobre el objeto `array`, `includes()` se añadió como un método para el **objeto `String`**. Esto nos permite realizar **comprobaciones sobre cadenas de texto.** En este caso, podemos buscar tanto palabras y oraciones como secuencias de caracteres:

```
var str = 'En un lugar de la Mancha, de cuyo nombre no quiero acordarme,
no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero,
adarga antigua, rocín flaco y galgo corredor.';

console.info( str.includes( 'lugar' ) ); // true
console.info( str.includes( 'de cuyo' ) ); // true
console.info( str.includes( 'ero aco' ) ); // true -> quiERO ACOrdarme
console.info( str.includes( 'Mancha', 18 ) ); // true
console.info( str.includes( 'Mancha', 19 ) ); // false
```

Operador exponencial (**)

Antes, en JavaScript, teníamos que hacer `Math.pow` (base,exponente) para calcular la potencia de una base elevada a un exponente.

Ahora con el nuevo operador **`**`** será muy fácil.

El operador de exponenciación es **asociativo a la derecha:**

`a ** b ** c` es igual a **`a ** (b ** c)`**

```
let miResultado

miResultado = 2 ** 3;       // base ** exponente
console.log(miResultado)    // 8

miResultado = 3 ** 2;       // base ** exponente
console.log(miResultado)    // 9

miResultado = 2 ** 4;       // base ** exponente
console.log(miResultado)    // 16

miResultado = 3 ** 3;       // base ** exponente
console.log(miResultado)    // 27
```

educación IT
CODE YOUR FUTURE

¡Sigamos trabajando!