

JS | Data Types

number & string

Objetivos:

- ✓ Entender cuáles son los dos principales tipos de datos en JavaScript en función de sus valores.
- ✓ Poder utilizar el tipo de dato *number*.
- ✓ Poder utilizar el tipo de dato *string*.
- ✓ Familiarizarse con algunos *string methods*.

Dos principales tipos de datos en JavaScript:

1. primitivos o valores primitivos
(cualquier dato que no sea un objeto y no tenga métodos).
2. objetos o valores no primitivos

1. Hay 6 tipos de datos primitivos:

- a) number
- b) string
- c) boolean
- d) null
- e) undefined
- f) symbol (nuevo en ES6)

a) Un número como tipo de datos

- Integer
- Floating-point

```
const age = 34;  
const price = 12.99;
```

NaN significa Not a Number es un número y representa un error computacional.

```
const name = "Sandra"; // <== string data type  
const whatIsThis = name/2;  
console.log(whatIsThis); // ==> NaN
```

Operaciones:

+ suma

- resta

* multiplicación

/ division

** potenciación/exponenciación

% modulo

Operadores de Asignación

Name	Operator	Equivalent
Assignment	$x = y$	N / A
Addition assignment	$x += y$	$x = x + y$
Subtraction assignment	$x -= y$	$x = x - y$
Multiplication assignment	$x *= y$	$x = x * y$
Division assignment	$x /= y$	$x = x / y$
Remainder assignment	$x \% = y$	$x = x \% y$
Exponentiation assignment	$x ** = y$	$x = x ** y$

El orden de las operaciones es una colección de reglas que definen qué procedimientos realizar primero (igual que en matemática).



b) String como tipo de dato

Un string es simplemente una secuencia de caracteres. Un carácter puede ser una letra, número, puntuación, etc.

Crear un string, utilizando:

- “ ” comillas dobles
- ‘ ’ comillas simples
- ` ` backticks (podemos agregar variables y expresiones dentro de los strings).

backticks

```
let name = "Ana";  
console.log(`Hello there, ${name}!`);  
// ==> Hello there, Ana!  
  
console.log(`${name} walks every day at least ${1+2} km`);  
// ==> Ana walks every day at least 3km
```


Caracteres especiales

Algunos strings contienen caracteres especiales, por lo que tenemos que usar secuencias de escape para que todo funcione.

Por ejemplo:

```
const favBook = "My favorite book is \"Anna Karenina\".";
console.log(favBook); // <== error: Unexpected token

const favBook = "My favorite book is 'Anna Karenina'.";
console.log(favBook); // <== My favorite book is 'Anna Karenina'.
```

Backslash (\)

cuando es necesario escapar de un carácter especial en una cadena.

Por ejemplo:

```
const mood = 'I\'m OK.';  
console.log(mood); // <== I'm OK.
```


Multiline strings

\n

` backticks y enter en el string.

```
console.log("Web Dev \n UX/UI");  
console.log(`Web Dev  
UX/UI`);
```

```
// The same:  
// Web Dev  
// UX/UI
```

Longitud del string (.length)

```
const name = "Ana";  
console.log(name.length); // <== 3
```


Métodos para manipular strings

JavaScript incluye una biblioteca de métodos para simplificar algunas de las tareas más comunes:

-concatenar o agregar al string “+” o “+ =”

```
let emptyContainer = "Hello there, student!";  
emptyContainer += " How are you?";  
console.log(emptyContainer);  
// Hello there, student! How are you?
```

Acceder a los caracteres dentro de un string: `charAt(n)`

```
const greeting = "Hello there!";  
console.log(`"${greeting}" is a string and it's length is ${greeting.length}.`);  
// "Hello there!" is a string and it's length is 12.  
console.log(greeting.charAt(0)); // <== H  
console.log(greeting.charAt(2)); // <== l
```

el primer carácter está indexado con cero (0).

Encontrar una subcadena: .indexOf(substr)

Devuelve el índice (posición) donde se encuentra un carácter/subcadena particular.

Si no la encuentra, devuelve -1.

```
const message = "Don't be sad, be happy!";  
console.log(message.indexOf("Don't")); //<== 0  
console.log(message.indexOf("t")); // <== 4  
console.log(message.indexOf("Be")); // <== -1
```

(capitalized Be ≠ lowercased be)

str.lastIndexOf(substr) - en el orden reverso

```
const message = "Don't be sad, be happy!";  
console.log(message.lastIndexOf("be"));  
//The index of the first "be" from the end is 14
```


Práctica

Escribir código que encuentre el índice de la letra "j" en:

"My favorite dessert is jello."

.repeat()

Si llamamos a `.repeat()` en un string y le pasamos un argumento de las veces que se repetirá.

```
console.log("$".repeat(3));  
// $$$
```

```
console.log("la".repeat(10));  
//lalalalalalalalalala
```


Obteniendo una subcadena

.substring()
.substr()
.slice()

```
const message = "JavaScript is cool!";

let withString = message.substring(4,6);
console.log(withString); // <== Sc

let withSubstr = message.substr(4,6);
console.log(withSubstr); // <== Script

let withSlice = message.slice(0,3);
console.log(withSlice); // <== Jav
```

Resumiendo: Aprendimos...

- ✓ A declarar, usar y manipular números y cadenas
- ✓ Operaciones
- ✓ Dos tipos de datos primitivos (string y numbers)