

JS | JavaScript Intro

Objetivos

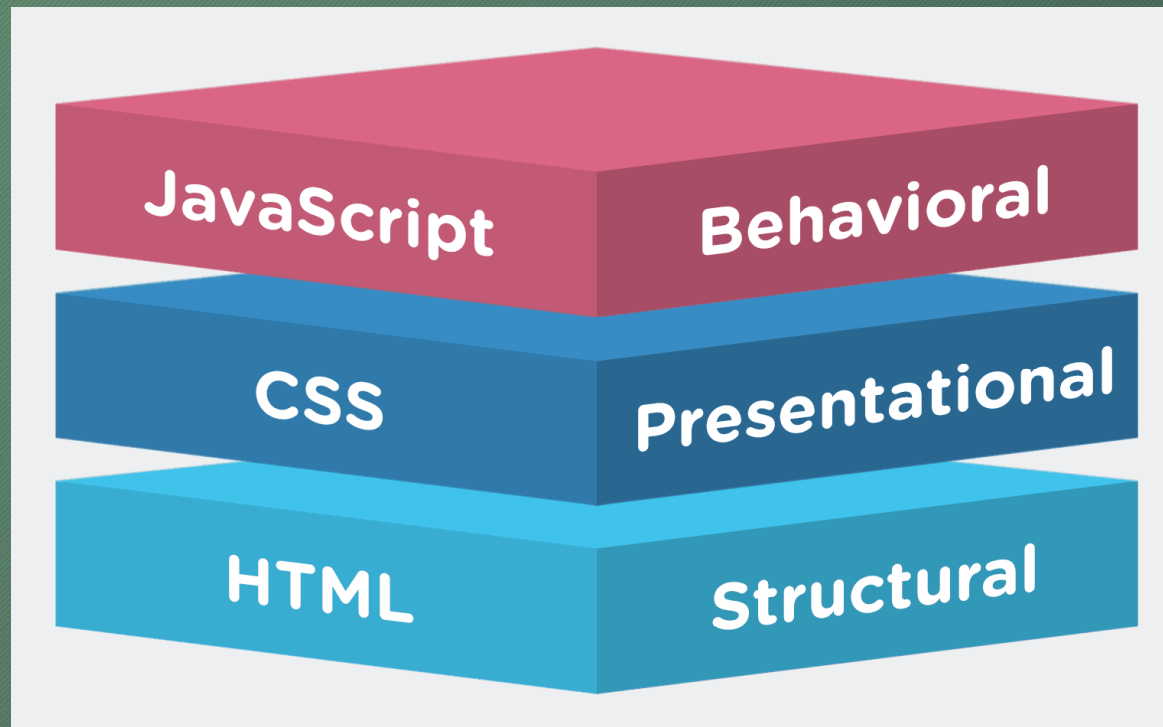
- ✓ Entender qué es JavaScript
- ✓ Comprender qué es ES6 y cómo evoluciona JavaScript
- ✓ Comprender y aplicar la sintaxis básica de JavaScript
- ✓ Declarar variables en JavaScript
- ✓ Nombrar correctamente a las variables

Qué es JavaScript

JavaScript es un lenguaje de programación que te permite realizar actividades complejas en una página web, y crear contenido dinámico.

HTML → CSS → JS

Las 3 capas se complementan una con la otra



JavaScript es un lenguaje basado en prototipos, multi-paradigma, de tipado dinámico, orientado a objetos, imperativo y estructurado, y declarativo (por ejemplo, programación funcional).



Imperative and Declarative Styles

El código **imperativo** es donde escribimos explícitamente cada paso de cómo queremos que se haga algo.

Mientras que con el código **declarativo** simplemente dices qué es lo que quieres que se haga.

JavaScript fue creado por Brendan Eich en 1995.
El nombre original de JavaScript fue LiveScript, y
pasó por muchas actualizaciones y cambios.



ECMAScript

ECMAScript es el estándar y JavaScript es su implementación más popular.

Las versiones de ECMAScript que todavía están en uso son ES5 (desde 2009), ES6 (desde 2015), ES7 (2016), ES8 (2017), ES9 (2018), ES10 (2019). Utilizaremos principalmente ES6.

La sintaxis de JavaScript

Es el conjunto de reglas que definen una estructura correcta.

Los paréntesis y las llaves son obligatorios y causarán un error si se omiten.

El punto y coma (;) no es obligatorio en la mayoría de los casos.

Usaremos la función `console.log()` para crear un output

```
console.log("Hello, I'm writing JavaScript");
```


Comentarios

Escribir comentarios es extremadamente importante para múltiples propósitos.

- Para comentar algo que cabe en una línea, usamos `//`:

```
// a short, one-line
```

- más de una línea, usaremos `/* */`

```
/* this is a long,  
multi-line comment  
Hopefully one day I'll appreciate  
writing all these comments :) */
```

Variables

El objetivo principal de las variables es **almacenar** información.

Se pueden almacenar diferentes tipos de datos: string, number, array, object, etc.

Declarando una variable

Antes de poder usar una variable, debemos declararla. La declaración de una variable no significa que se le haya asignado algún valor; simplemente significa “reservar un lugar en una memoria” que luego se llenará con algún valor.

Declarando una variable

Para declarar o definir una variable podemos usar las keyword *var*, *let* o *const*. Luego veremos cada una.

Por ejemplo:

```
//variable declaration  
let name;  
  
//multiple variable declaration  
let name, age, email;
```


Declaración de una variable con *let*

Inicialización de la variable

Puede hacer la inicialización de la variable en el momento de la creación de la variable (cuando la declara) o en un momento posterior.

Si no asigna un valor a una variable cuando la declara, su valor predeterminado será *undefined*.

```
let name;  
name = "John";  
  
let name = "John";
```

Nombrar una variable

Las reglas para nombrar variables:

1. Los nombres pueden contener letras (mayúsculas y minúsculas), números y los símbolos _ y \$.
2. El primer carácter del nombre no puede ser un número.

```
let a;  
let color;  
let _private;  
let $button;  
let getTop10;  
let a_large_name;  
let thisWayIsCalledCamelCase;
```


Nombrar una variable

Al crear una variable con más de una palabra, usamos el estilo "camelCase".

```
let thisIsCalledCamelCase;
```

El propósito de esta práctica es mejorar la legibilidad.

JavaScript distingue entre mayúsculas y minúsculas, por lo que, por ejemplo, *color* y *Color* son nombres de variables diferentes.

Nombrar una variable

Puede usar cualquier palabra, pero hay algunas keyword reservadas que no se pueden usar, por ejemplo: *let*, *class*, *return*, *function*.

Aquí está la lista completa:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Lexical_grammar#Keywords

Valores cambiantes

Las variables declaradas con la keyword *let* se pueden manipular y sus valores se pueden cambiar:

```
let favoriteFood;  
  
favoriteFood = "Steak";  
console.log(favoriteFood);  
  
favoriteFood = "Pizza";  
console.log(favoriteFood);  
  
// Steak  
// Pizza <== the variable changed value from 'Steak' to 'Pizza'
```

Conversión de tipo

Puede reasignar valores y cambiar el tipo de datos de las variables en JavaScript.

Al cambiar el valor, podemos cambiar el tipo de variable.

```
let favoriteFood;  
  
favoriteFood = "Steak";  
console.log("Value: ", favoriteFood, " Type:", typeof favoriteFood);  
  
favoriteFood = 20;  
console.log("Value: ", favoriteFood, " Type:", typeof favoriteFood);  
  
// Value:  Steak   Type:  string  
// Value:  20      Type:  number
```


typeof

typeof es un operador de JavaScript que devuelve el tipo de la variable que le pasamos.

Declaración de una variable con *const*

const se usa al declarar una variable cuyo valor será constante.

Cuando se usa *const*, la variable debe tener asignado algún valor en el mismo momento en que se inicializa la variable.

```
let name; //<== we can do this  
const price; // <== error
```


Reglas importantes de nomenclatura:

- use expresiones comprensibles para humanos, como `firstName`, `hasColor`, `pricePreTax`, etc.
- nunca, use `x`, `j`, `a` para nombrar sus variables, a menos que sea muy claro para usted y para todos los demás miembros del equipo.
- nombrar variables tan descriptivas y concisas como sea posible.

Imperative

```
let arr = [1, 2, 3, 4, 5],  
    arr2 = [];  
  
for (var i=0; i<arr.length; i++) {  
    arr2[i] = arr[i]*2;  
}  
  
console.log(arr2);
```

Declarative

```
let arr = [1, 2, 3, 4, 5];  
  
arr2 = arr.map(function(v, i) {  
    return v * 2;  
});  
  
console.log(arr2);
```


Resumiendo: Aprendimos...

- ✓ un poco sobre historia y evolución de JavaScript.
- ✓ a declarar una variable, las reglas de nomenclatura.
- ✓ que JavaScript es un lenguaje de tipo dinámico y que es posible cambiar valores y tipos de variables en el tiempo de ejecución.