

{JS}

JS

Clase 01

CONQUER BLOCKS

<índice>

Introducción a JS

¿Qué es JS?

¿Cómo hacer funcionar JavaScript?

Conceptos básicos del lenguaje

¿Qué hace diferente a JavaScript?

**Anatomía de una variable y convención
de nombres**

¿Qué es JS?

¿Qué es JS?

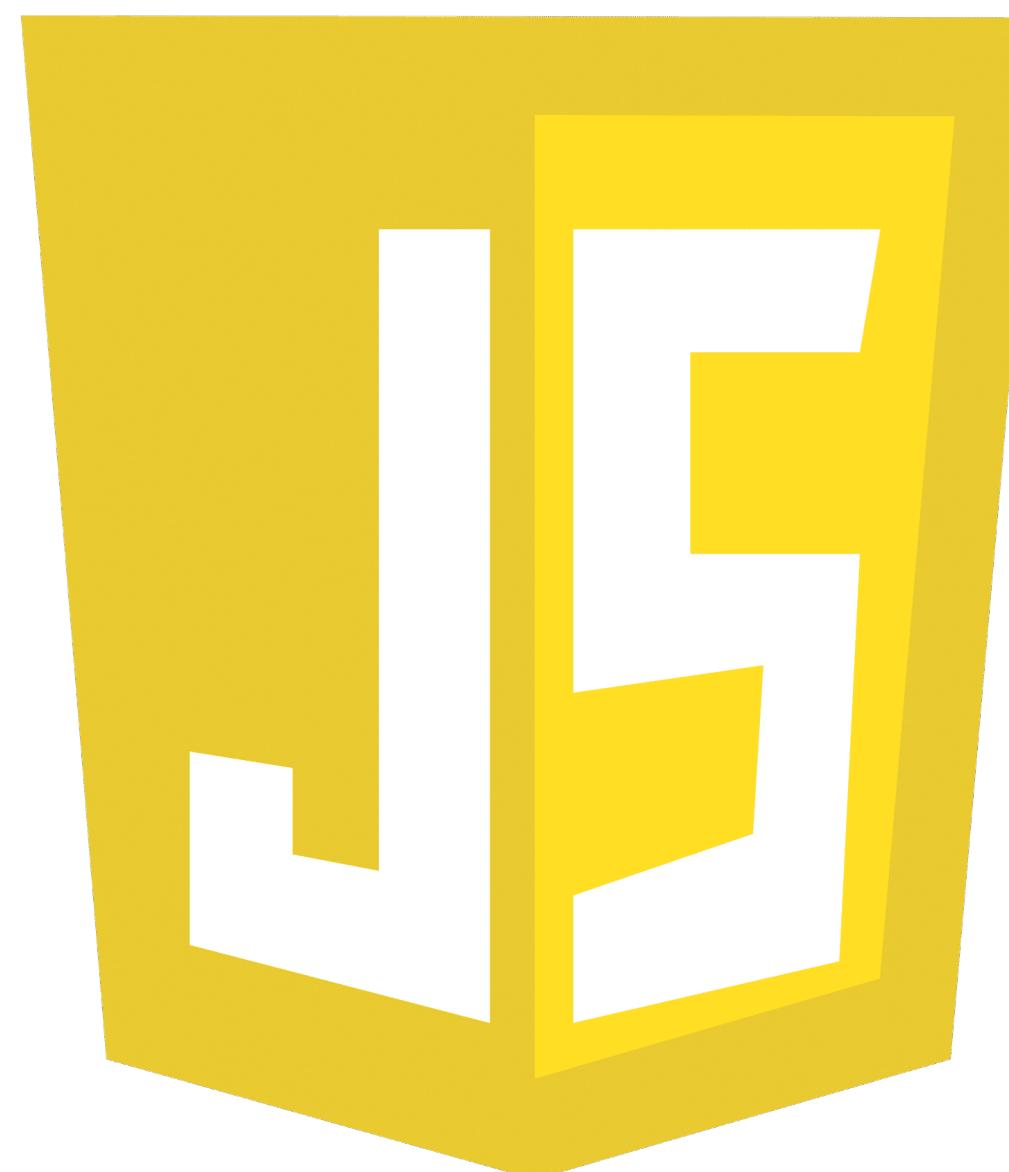
HTML



CSS

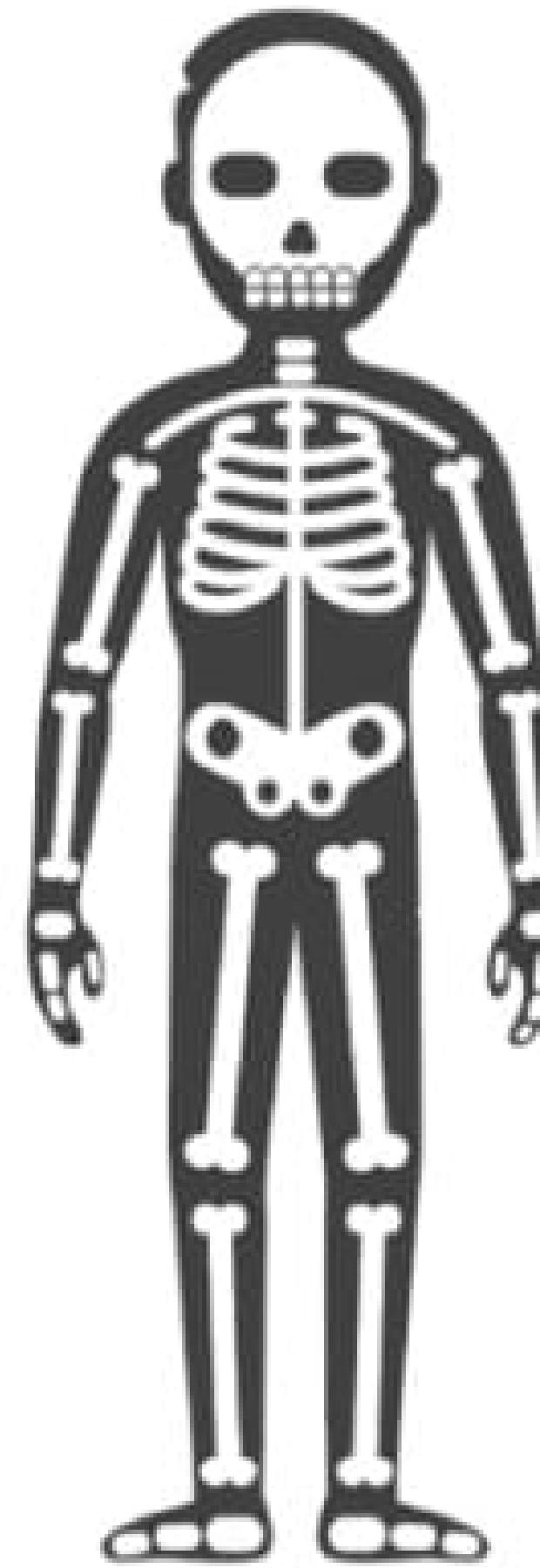


JavaScript



¿Qué es JS?

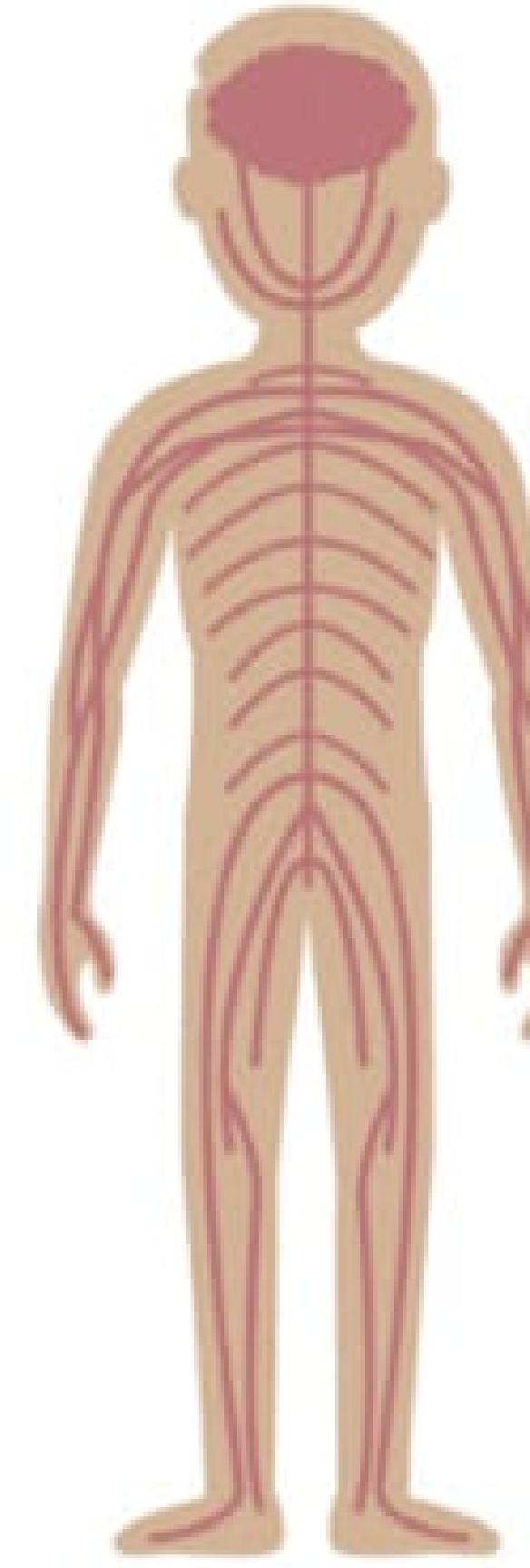
HTML the Skeleton



CSS the Skin



Javascript the Brain



¿Qué es JS?

What's the Difference?



HTML
Hypertext Markup Language

Create the structure

- Controls the layout of the content
- Provides structure for the web page design
- The fundamental building block of any web page



CSS
Cascading Style Sheet

Stylize the website

- Applies style to the web page elements
- Targets various screen sizes to make web pages responsive
- Primarily handles the "look and feel" of a web page



Javascript

Increase interactivity

- Adds interactivity to a web page
- Handles complex functions and features
- Programmatic code which enhances functionality

¿Qué es JS?

- Es el responsable de controlar toda la interactividad del usuario en la web

Razones por las que aprender JS

- Trabajes en lo que trabajes en web debes conocer JS si o sí
- Toda la web está hecha HTML + CSS + JS
- Son los cimientos de toda web

¿Qué es JS?

JavaScript



Características

1. Texto plano - Cualquier editor
2. Lenguaje nativo de los navegadores
3. No necesita ser compilado ni transpilado
4. SI ES UN LENGUAJE DE PROGRAMACIÓN

¿Qué es JS?

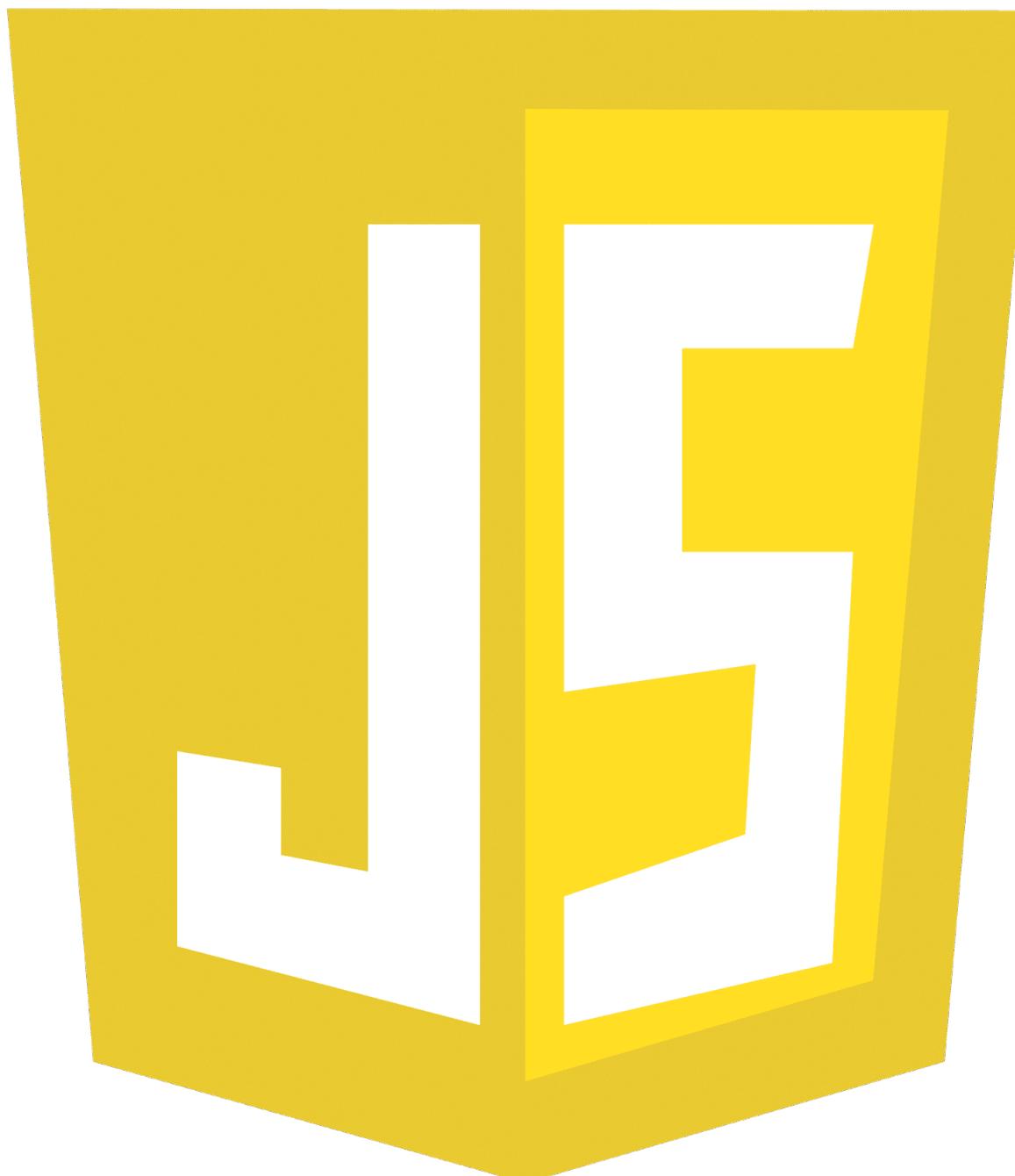
JavaScript



JavaScript fue desarrollado originalmente por Brendan Eich de Netscape con el nombre de Mocha, el cual fue renombrado posteriormente a LiveScript, para finalmente quedar como JavaScript.

¿Qué es JS?

JavaScript

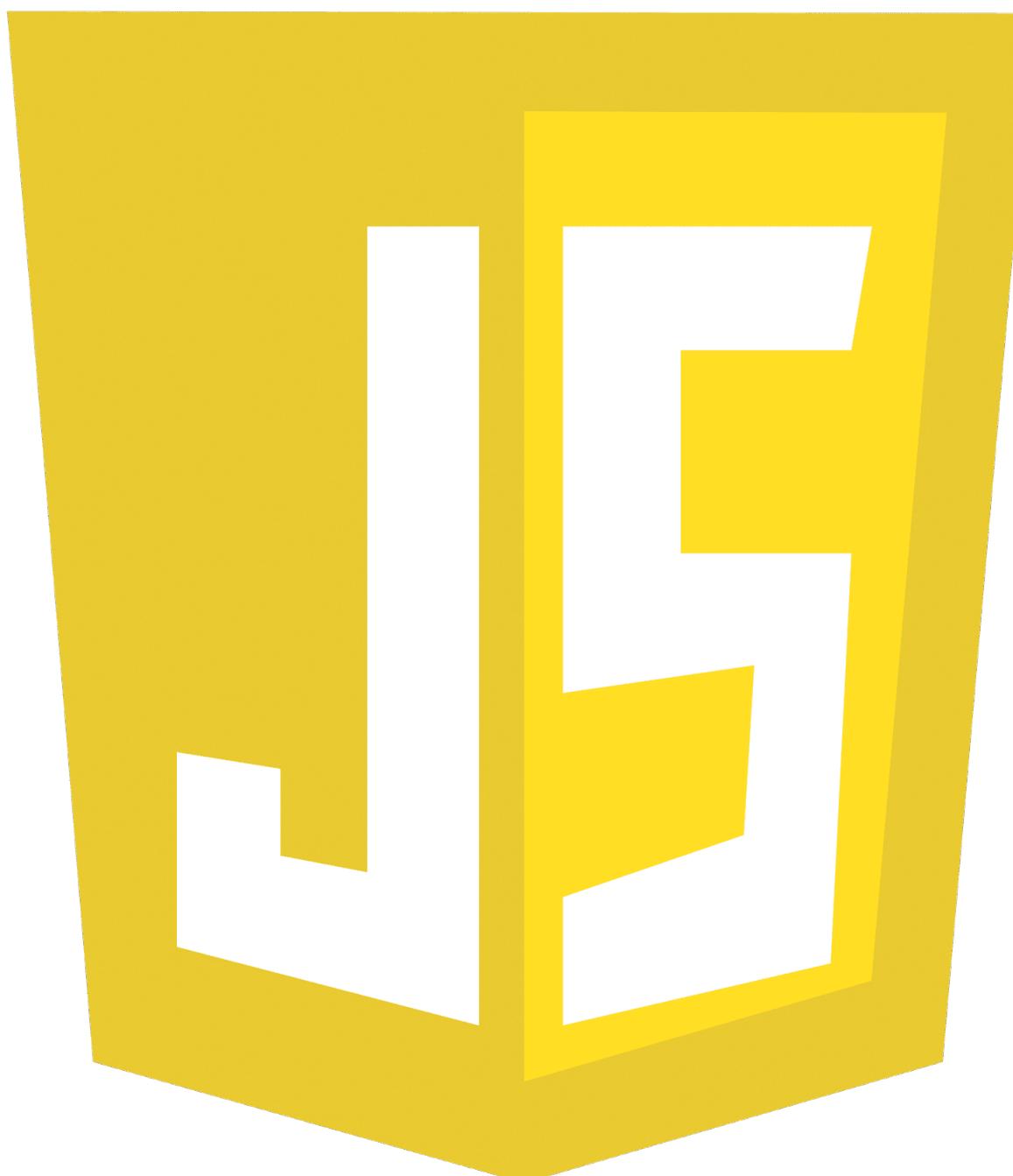


Imperativo y estructurado

- JavaScript es compatible con gran parte de la estructura de programación de C (por ejemplo, sentencias if, bucles for, sentencias switch, etc.).
- JavaScript difiere sobre todo en el ámbito o scope de las variables

¿Qué es JS?

JavaScript

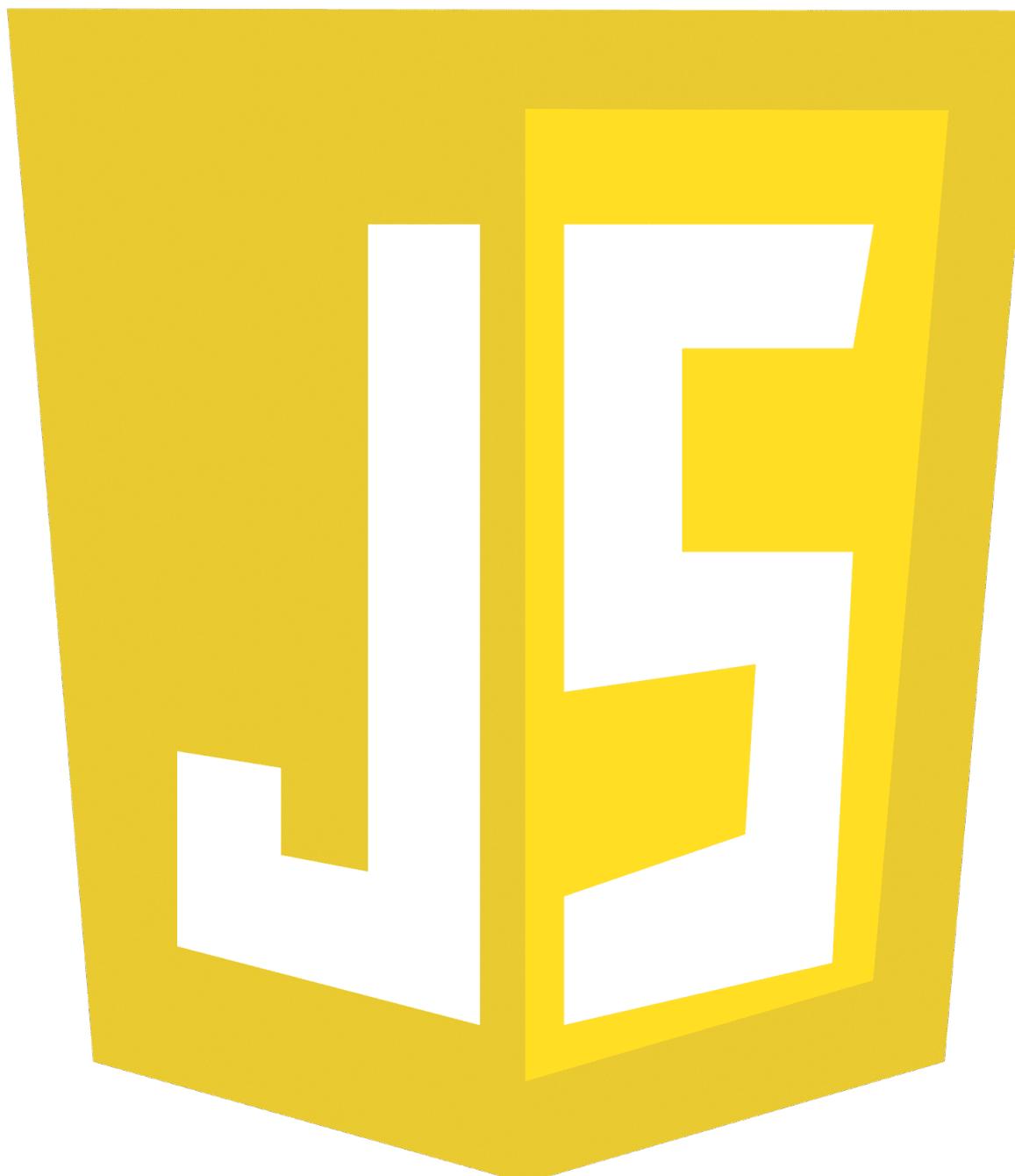


Tipado dinámico

- Como en la mayoría de lenguajes de scripting, el tipo está asociado al valor, no a la variable.
- Por ejemplo, una variable x en un momento dado puede estar ligada a un número y más adelante, religada a una cadena.

¿Qué es JS?

JavaScript

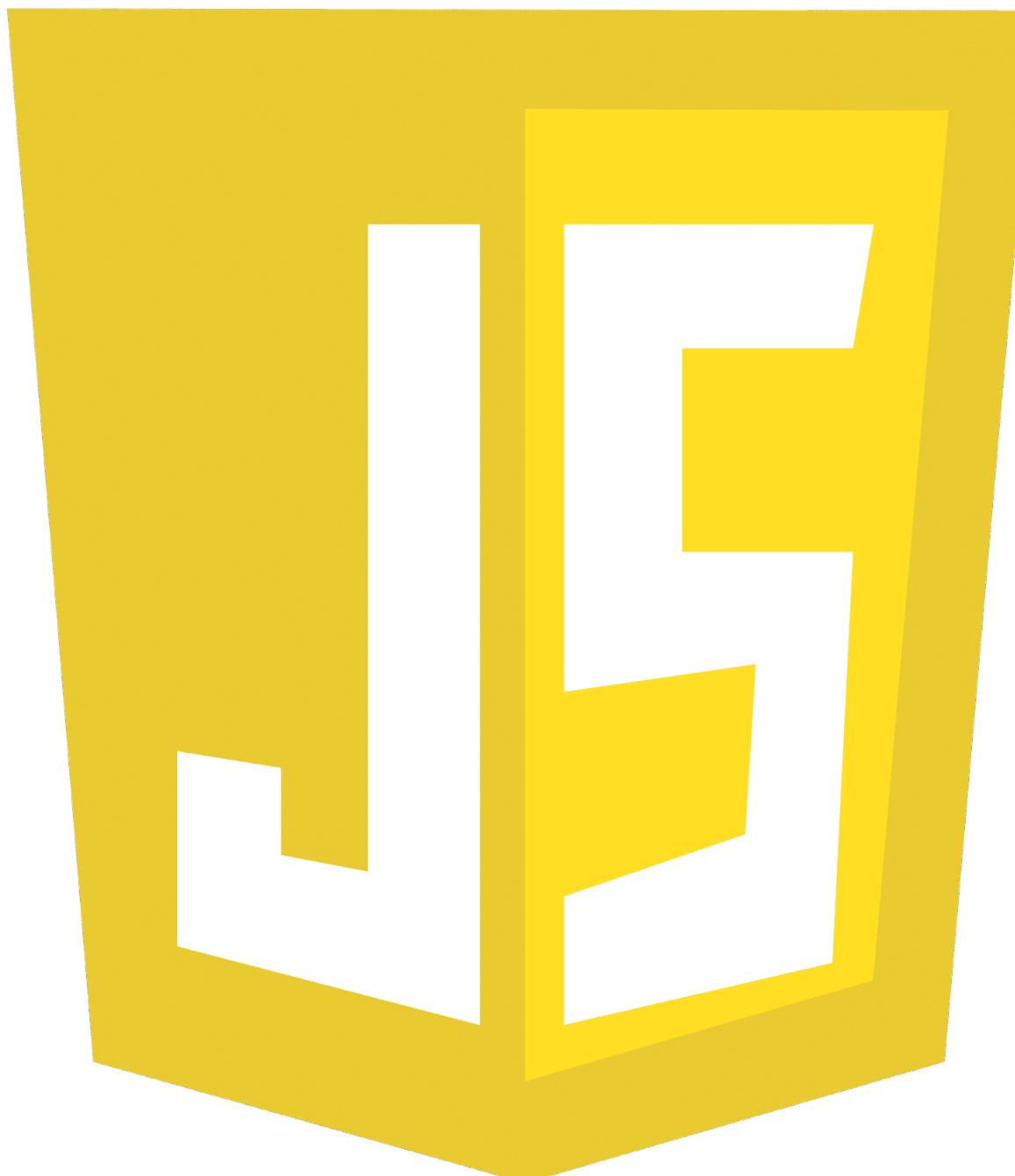


Objetual

- JavaScript está formado casi en su totalidad por objetos.
- Los objetos en JavaScript son arrays asociativos, mejorados con la inclusión de prototipos (ver más adelante). Los nombres de las propiedades de los objetos son claves de tipo cadena: `obj.x = 10` y `obj['x'] = 10` son equivalentes,

¿Qué es JS?

JavaScript

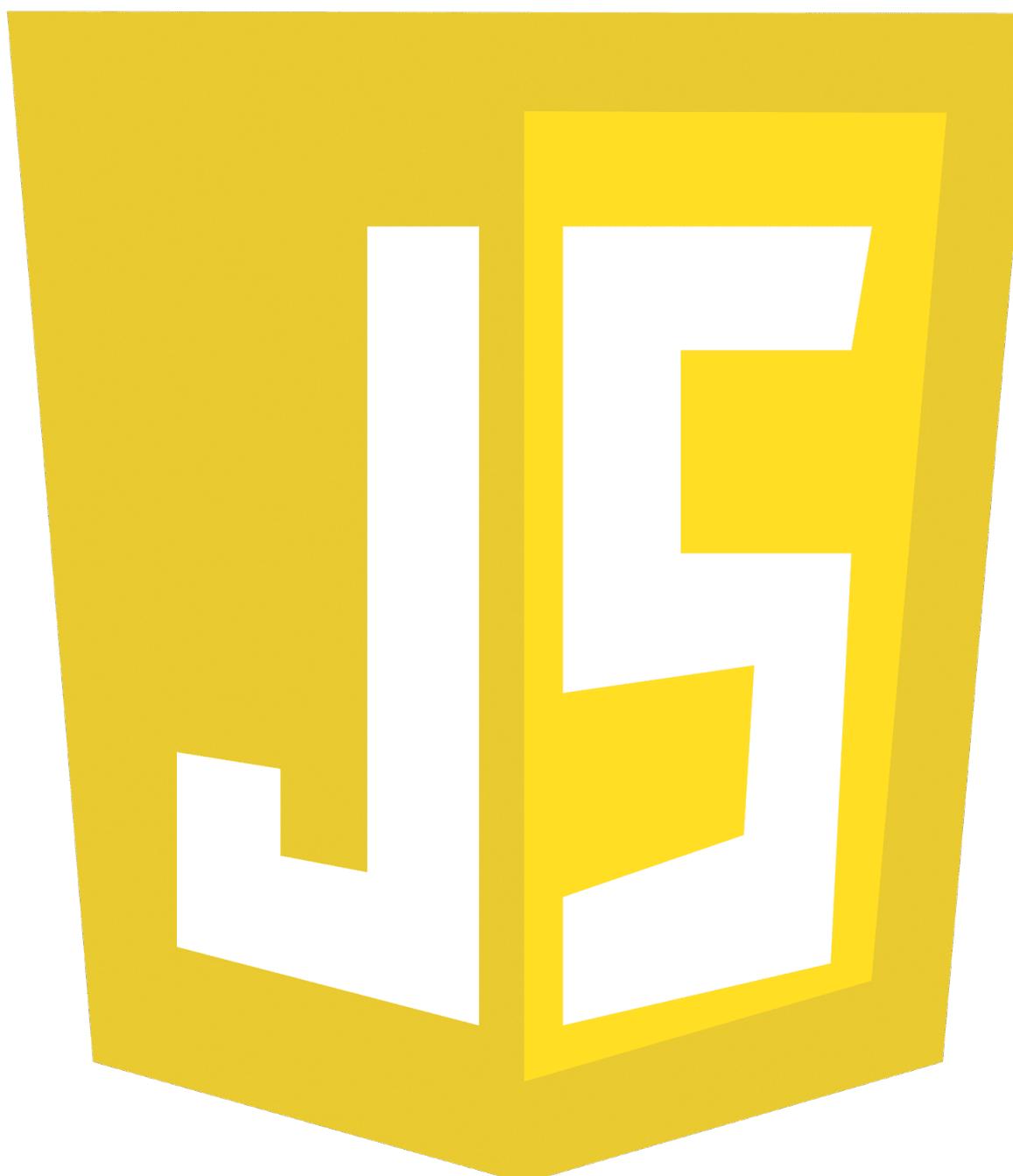


Evaluación en tiempo de ejecución

- JavaScript incluye la función eval que permite evaluar expresiones expresadas como cadenas en tiempo de ejecución.
- Por ello se recomienda que eval sea utilizado con precaución y que se opte por utilizar la función

¿Qué es JS?

JavaScript



Funciones de primera clase

- En JavaScript a las funciones se le llaman ciudadanos de primera clase y es que hasta las funciones son objetos con propiedades y métodos

¿Qué es JS?

JavaScript



Entornos de ejecución

- JavaScript normalmente depende del entorno en el que se ejecute para ofrecer objetos y métodos por los que los scripts pueden interactuar con el "mundo exterior".
- De hecho, depende del entorno para ser capaz de proporcionar la capacidad de incluir o importar scripts (por ejemplo, en HTML por medio del tag <script>). (Esto no es una característica del lenguaje, pero es común en la mayoría de las implementaciones de JavaScript.)

¿Cómo hacer funcionar JS?

¿Cómo hacer funcionar JS?

Ejecuta JS

- Consola del navegador
- Etiqueta script dentro de html
- Script externo
- nodejs desde la terminal

¿Cómo hacer funcionar JS?

Consola del navegador

- El clásico primer ejemplo cuando se comienza a programar, es crear un programa que muestre por pantalla un texto, generalmente el texto «Hola Mundo».
- También podemos realizar, por ejemplo, operaciones numéricas. En la consola Javascript podemos hacer esto de forma muy sencilla:

¿Cómo hacer funcionar JS?

Consola del navegador

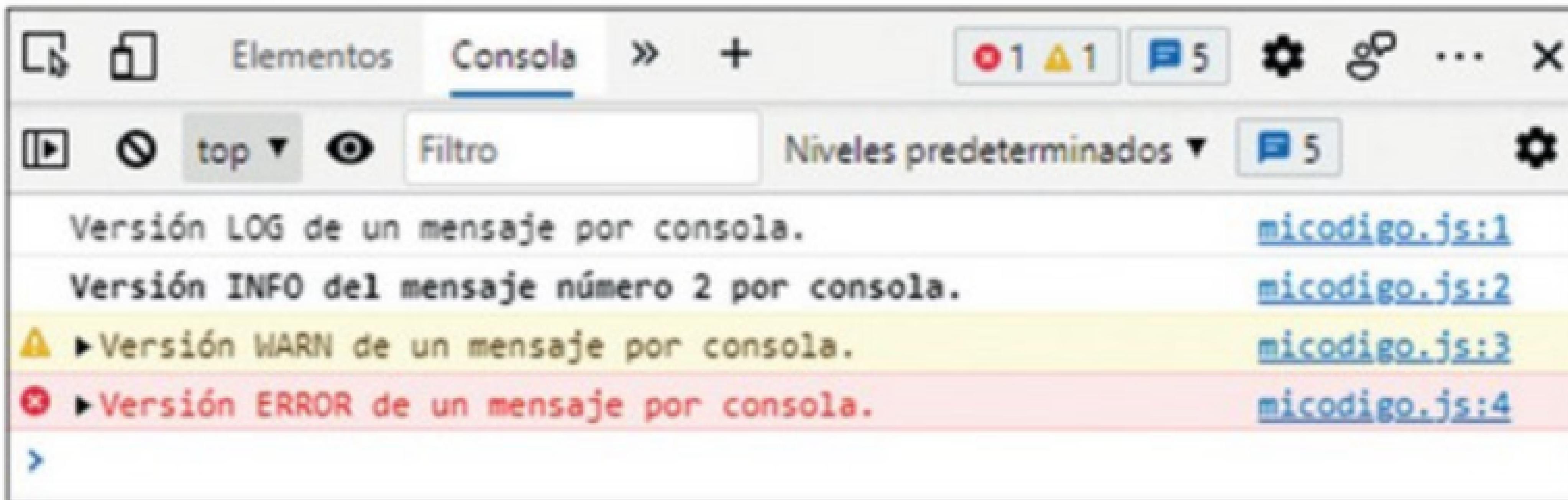
```
console.log("Hola Mundo");  
console.log(2 + 2);
```

¿Cómo hacer funcionar JS?

Consola del navegador

Función	Descripción
<code>console.log()</code>	Muestra la información proporcionada en la consola Javascript.
<code>console.info()</code>	Equivalente al anterior. Se utiliza para mensajes de información.
<code>console.warn()</code>	Muestra información de advertencia. Aparece en amarillo.
<code>console.error()</code>	Muestra información de error. Aparece en rojo.
<code>console.clear()</code>	Limpia la consola. Equivalente a pulsar CTRL + L o escribir <code>clear()</code> .

¿Cómo hacer funcionar JS?



¿Cómo hacer funcionar JS?

Eiqueta script en nuestro html

- Este método de escribir scripts se denomina Javascript en línea (inline), y significa que el Javascript está escrito directamente en el código HTML. Nos puede servir como ejemplo inicial, pero no es la forma recomendable de escribirlo, ya que lo ideal es separar el código HTML del código Javascript

¿Cómo hacer funcionar JS?

**Eiqueta script
en nuestro html**

```
<html>
  <head>
    <title>Título de la página</title>
    <script>
      console.log("¡Hola!");
    </script>
  </head>
  <body>
    <p>Ejemplo de texto.</p>
  </body>
</html>
```

¿Cómo hacer funcionar JS?

Script externo

- Esta otra forma de incluir Javascript en una página tiene la ventaja de, en el caso de necesitar incluir el código Javascript desde varios documentos HTML, no tendremos que volver a escribir dicho código, sino simplemente referenciar el nombre del mismo archivo Javascript a incluir en todas las páginas HTML.

¿Cómo hacer funcionar JS?

Script externo

```
<html>
  <head>
    <title>Título de la página</title>
    <script src="js/index.js"></script>
  </head>
  <body>
    <p>Ejemplo de texto.</p>
  </body>
</html>
```

¿Cómo hacer funcionar JS?

Script externo Dónde ubicar la etiqueta script

Ubicación	¿Cómo descarga el archivo Javascript?	Estado de la página
En <head>	ANTES de empezar a dibujar la página.	Página aún no dibujada.
En <body>	DURANTE el dibujado de la página.	Dibujada hasta donde está la etiqueta <script>.
Antes de </body>	DESPUÉS de dibujar la página.	Dibujada al 100%.

¿Cómo hacer funcionar JS?

Etiqueta script

¿Cómo hacer funcionar JS?

Ejecución con nodejs

¿Qué hace diferente a JS?

¿Qué hace diferente a JS?

- Se puede ejecutar en el front y back
- Sabe manipular el DOM (lista 50 elementos en html por ejemplo)
- No es el mejor lenguaje para aprender a programar
- Se partirá de todo lo visto en los módulos de programación

¿Qué hace diferente a JS?

ECMAScript

A lo largo de los años, Javascript ha ido sufriendo modificaciones que los navegadores han ido implementando para acomodarse a la última versión de ECMAScript cuanto antes. La lista de versiones de ECMAScript aparecidas hasta el momento son las siguientes, donde encontramos las versiones enmarcadas en lo que podemos considerar el pasado de Javascript

¿Qué hace diferente a JS?

Ed.	Fecha	Nombre formal / informal	Cambios significativos
1	JUN/1997	ECMAScript 1997 (ES1)	Primera edición
2	JUN/1998	ECMAScript 1998 (ES2)	Cambios leves
3	DIC/1999	ECMAScript 1999 (ES3)	RegExp, try/catch, etc...
4	AGO/2008	ECMAScript 2008 (ES4)	Versión abandonada.
5	DIC/2009	ECMAScript 2009 (ES5)	Strict mode, JSON, etc...
5.1	DIC/2011	ECMAScript 2011 (ES5.1)	Cambios leves

¿Qué hace diferente a JS?

A partir del año 2015, se marcó un antes y un después en el mundo de Javascript, estableciendo una serie de cambios que lo transformarían en un lenguaje moderno, partiendo desde la especificación de dicho año, hasta la actualidad:

Ed.	Fecha	ECMAScript	Cambios significativos
6	JUN/2015	ES2015	Clases, módulos, hashmaps, sets, for of, proxies...
7	JUN/2016	ES2016	Array includes(), Exponenciación **
8	JUN/2017	ES2017	Async/await
9	JUN/2018	ES2018	Rest/Spread operator, Promise.finally()...
10	JUN/2019	ES2019	flat, flatMap, trimStart(), optional error catch...
11	JUN/2020	ES2020	Dynamic imports, BigInt, Promise.allSettled

Anatomía de una variable

Anatomía de una variable

¿Qué es una variable?

Anatomía de una variable

- Declaración
- Asignación
- Nombres descriptivos
- Evitar ambigüedades
- Evitar palabras prohibidas
- Evitar símbolos de puntuación

Anatomía de una variable

- Ser descriptivos
- Usar camelcase
- Algunos consejos
- Más consejos

Anatomía de una variable

```
// Good  
const isActive = false
```

```
// Bad  
const is-active = false  
const is_active = false
```

```
// Good  
const article = {title: 'Some title'}  
const isAlreadyPaid = true
```

```
// Bad  
const a = {title: 'Some title'}  
const paid = true
```

```
// Good  
const isLoggedIn = true  
const shouldBeRendered = true
```

```
// Bad  
const logged = true  
const areDefined = true
```

Anatomía de una variable

Declaremos nuestras
primeras variables

Anatomía de una variable

¿let, var o const?

Anatomía de una variable

¿Qué podemos guardar en las variables?

<Despedida>

Email

bienvenidosaez@gmail.com

Instagram

@bienvenidosaez

Youtube

youtube.com/bienvenidosaez

CONQUERBLOCKS