

# Desarrollo web entorno cliente

Javascript como lenguaje:

- Historia
- Propósito del lenguaje
- Principios básicos

# JavaScript. Historia

- 1995 encargo de Netscape a Brendan Eich para dotar de dinamismo/interactividad a HTML (10 días).
- Mocha → LiveScript → JavaScript
- No tiene nada que ver con Java (Marketing).
- 96-97 Estandarización por ECMA. Navegadores deben cumplir el estándar. Soporte a navegadores antiguos!.
- Muchas críticas de comunidades que no tienen estándar.
- Comparativa navegadores, ECMA y paradigmas

# JavaScript. Propósito

- Ver enlace de paradigmas. ¿Funcional, OO, etc.?
- Funcional → Base matemática. LISP, SCALA...
- OO. Herencia, Polimorfismo → Abstracción

# JavaScript. Principios básicos

- Concurrencia: Varios agentes acceden a un recurso a la vez. Asincronía, a efectos prácticos. Simulado.
- Single-thread. WebWorkers. Main Thread
- Gestión de memoria. Recursos HW, comunicaciones, etc.
- Tipado débil. Asociado al valor más que a las variables.

# JavaScript. Funcionamiento

- Todo programa acaba ejecutándose sobre un hardware. SSOO. Navegadores. Varias capas.
- Runtimes. Parsean, optimizan y ejecutan.
- Event loop. Contextos de ejecución.
- Recolector de basura → Marca elementos “perdidos”

# JavaScript. Primitivas

- **string, number, boolean, null, undefined, symbol y object.**
- Si typeof da alguno de los anteriores → primitiva → inmutable. Si da object, se ha usado el constructor. Probar añadir propiedad a string y usar métodos.
- NaN, Infinity y - Infinity
- null vs undefined. null → Recolector basura.  
undefined → comprobar tipos.
- valueOf() → Devuelve el valor primitivo.

# JavaScript. Sintaxis. Variables

- **primitivas.** Ejemplo
- **scope.** Ejemplo
- **var vs let vs const.** Ejemplo
- Paso por valor - referencia. Método slice().  
Ejemplo
- Destructuring assignment
- Hoisting. En un contexto de ejecución 2 fases: creación (declaracion variables y funciones) y ejecución.
- Enlazar JS. HTML, script, etc.

# JavaScript. Sintaxis

- E/S de datos. console, alert y prompt.
- Operaciones con números
- Operaciones con cadenas. String literals.
- Operadores lógicos. [Falsy values](#)
- Expresiones regulares.
- Control del flujo de los programas.
- Estructuras condicionales.
- Estructuras iterativas.



# JavaScript. Sintaxis

- Funciones. Funciones flecha.
- Arrays
- Clases y objetos.