

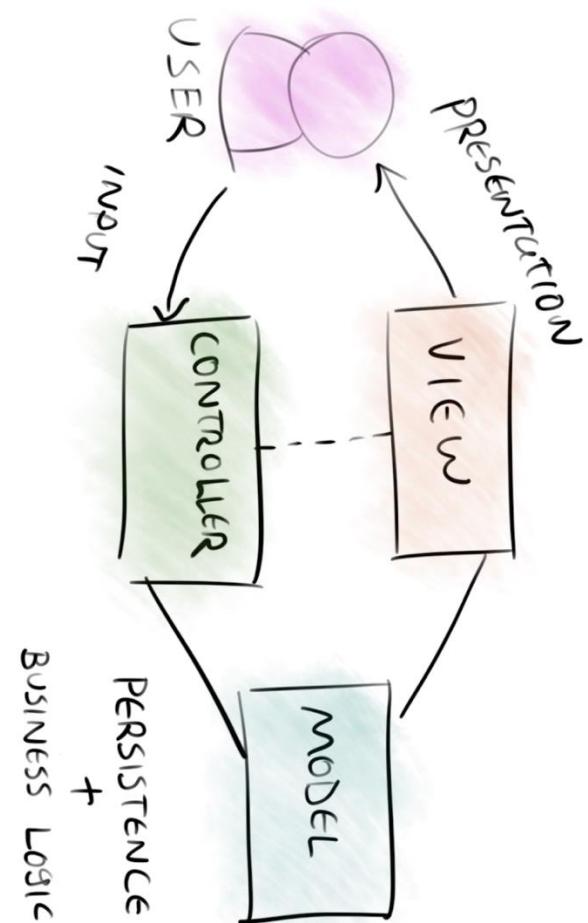


UNIVERSIDAD DE CÓRDOBA

PROGRAMACIÓN WEB – SEMINARIO 4

Lenguajes para la web

Dr. José Raúl Romero Salguero
jrromero@uco.es



S4-1

Lenguajes de mercado

Lenguajes de marcado: XML

- **XML** (*eXtensible Markup Language*) es un formato sencillo y flexible utilizado fundamentalmente para el **intercambio de datos** en Web
- Derivado de **SGML** (*Standard Generalized Markup Language*), ISO Std. 8879:1986 “Tratamiento de la información - Sistemas de texto y de oficina - Lenguaje de marcado generalizado estándar (SGML)”, que **-sin embargo-** resultó **muy complejo** para su interpretación y procesamiento por clientes web y aplicaciones
 - **Fijó las bases** de los lenguajes de marcado, basado en etiquetas
 - **Especificación declarativa**, describiendo estructura y otros atributos del documento

- XML es un **estándar libre y abierto** de la W3C, que define las reglas para la codificación de documentos en un formato basado en texto, legible tanto por la máquina como por el humano
- XML es un **esqueleto para crear otros lenguajes** de marcas
- Se valida con **XML Schemas**
- La **RFC 7303** recomienda el **tipo de medios “+xml”** para los formatos basados en XML (ej. “image/svg+xml”)
 - Especificación en <https://www.w3.org/XML/>
 - Tutorial: <https://www.w3schools.com/xml/default.asp>

➤ **Características** del documento XML:

- Un documento XML está basado en **marcas** (entre “<” y “>”) **y contenido**, todo ello como cadenas de caracteres
- Hay **tres tipos de etiquetas**: **inicio** (“<tag>”), **final** (“</tag>”) y etiquetas de **elemento vacío** (“<tag />”)
- Se denomina **elemento** al componente lógico que comienza y acaba con las correspondientes etiquetas (ej. el elemento DIV en <div>contenido</div>)
- Se denomina **atributo** a un par nombre-valor que puede contenerse dentro de las etiquetas de inicio o de elemento vacío (ej. el atributo CLASS en el elemento DIV: <div class="miclase">...</div>)
- Todo documento se inicia con la declaración XML, que incluye metainformación sobre el propio documento: <?xml version="1.0" encoding="UTF-8"?>

<elemento atributo="valor">contenido</elemento>

```
<?xml version="1.0" encoding="UTF-8"?>                                <!-- XML UTF-8 -->
<message private="true">                                         <!-- root element -->
    <from id="alba@miempresa.es">Alba Lopez</from>
    <to id="Roberto@miempresa.es">Roberto Manuel</to>
    <subject>Mañana en mi fiesta de cumpleaños!</subject>
    <content language="spanish">
        Oye, Roberto, no te olvides de llamarme para organizar la fiesta.
    </content>
</message>
```

- ▷ **Ventajas** de XML:
 - ❑ Formato **abierto y estándar** (¡no reinventes la rueda!)
 - ❑ Permite representar **casi cualquier tipo general de datos** (árboles, listas, registros, tipos básicos, etc.)
 - ❑ Fácil de leer por la máquina y por humanos
 - ❑ Multitud de **herramientas** para su procesamiento

- ▷ **Desventajas** de XML:
 - ❑ Formato **muy voluminoso**, que genera grandes ficheros – puede influir negativamente en el rendimiento
 - ❑ Algunos **datos pueden ser difíciles de encajar** en un formato XML válido
 - ❑ El **código Javascript** para navegar por el árbol de etiquetas de XML (XML DOM) es **voluminoso y tedioso**

Lenguajes de marcado: HTML

- **HTML** (*Hypertext Markup Language*) es el lenguaje de marcado utilizado en el **lado de cliente**, que sirve para **desarrollar la estructura de una página web** permitiendo:
 - ❑ Publicar cabeceras, texto, tablas, imágenes, listados, etc.
 - ❑ Acceso a documentos en línea a través de hipervínculos
 - ❑ Diseñar formularios con el fin de realizar transacciones con servicios remotos
 - ❑ Incluir hojas de cálculo, videos, audios y elementos externos de otras aplicaciones directamente en el documento
- Heredado de XML (**XHTML**), **HTML 5** es la recomendación actual
- El **validador de HTML** W3C permite asegurar que un documento `<!DOCTYPE html>` sigue las reglas del esquema de HTML

- En el **W3C Workshop 2004** se establecen las bases de **HTML5**, al que se unen Opera, Apple y Mozilla (que forman **WHATWG**) – su desarrollo se inició junto con W3C en 2007
- Aparece la recomendación de **HTML5 en 2012 por la W3C**, incluyendo información semántica en los documentos
- Aparece la especificación **HTML de WHATWG** (estado: *Living Standard*) en 2016
- En mayo de 2019, WHATWG y W3C **firman un acuerdo** para llevar una única especificación de HTML y DOM
- Especificación de *Living Standard* - **HTML5** (last update: 13 oct. 2025):
<https://html.spec.whatwg.org/multipage/>
- **Tutorial de HTML5:** <https://www.w3schools.com/html/>

HTML



S4_ej1

S4-2

Lenguaje de diseño CSS

Lenguaje de diseño: CSS

- **CSS** (*Cascading Style Sheets*) es el lenguaje de hoja de estilos utilizado para la **presentación del diseño gráfico** de un documento escrito en el lenguaje **HTML**
 - ❑ Describe la estructura del diseño (*layout*), colores, fuentes y reglas de presentación
 - ❑ Permite describir diferentes presentaciones para distintos dispositivos
- HTML, CSS y Javascript forman la **base del desarrollo web** (**lado de cliente**)
 - ❑ Esta separación permite mejorar la **accesibilidad**, **flexibilidad**, **control** y **escalabilidad** del sitio web
 - ❑ Múltiples documentos web pueden compartir el mismo diseño
 - ❑ Tipo de medios **text/css**

➤ Tres posibles métodos de aplicación de estilos:

- En línea, utilizando el atributo **style** de HTML

```
<body style="background-color: #FF0000;">
```

- Interno, incrustando CSS como contenido de la etiqueta **<style ...>** en la cabecera (**<head>...</head>**) del documento HTML

```
<style type="text/css">
    body {background-color: #FF0000; }
</style>
```

- Externo, vinculando el documento HTML a un fichero de estilos externo con extensión .css

```
<link rel="stylesheet" type="text/css" href="style/style.css" />
```

- CSS cubre tres aspectos diferentes de estilo y maquetación:
 - ❑ **Estilo del texto**, incluyendo fuentes, colores de fuente, tipos de ítems de las enumeraciones, etc.
 - ❑ **Estilo de las cajas**, considerando que todos los elementos de HTML son considerados como cajas con las siguientes propiedades:



Permite modificar el estilo y tamaño del contenido, relleno, borde, margen y caja

- ❑ **Distribución de las cajas**, permitiendo modificar su posición absoluta y relativa de una respecto a otras

HTML



CSS



CSS3 no solo da estilo: también permite crear *layouts* adaptables y modernos de forma sencilla

Lenguaje de diseño: CSS_

Fundamentos del lenguaje

- CSS define reglas que se aplican a un documento, y modifican el modo en que este es presentado
- Cada regla CSS se compone de:
 - Conjunto de propiedades: valores establecidos para la presentación del documento HTML
 - Cada par propiedad-valor recibe el nombre de declaración
 - Selector: “Seleccionador” de los elementos HTML afectados por el nuevo valor de la propiedad

Selector → h1 {
Declaración → color: blue;
background-color: yellow;
border: 1px solid black; }

Lenguaje de diseño: CSS

Principales selectores

Selector	Ejemplo	Qué selecciona
Elemento	p { ... }	Todas las etiquetas <p>
Clase	.destacado { ... }	Elementos con class="destacado"
ID	#principal { ... }	El elemento con id="principal"
Descendiente	div p { ... }	<p> dentro de un <div>
Múltiple	h1, h2 { ... }	Varios elementos a la vez
Universal	* { ... }	Todos los elementos
Atributo	input[type="text"] { ... }	Elementos con un atributo concreto
Pseudoclase	a:hover { ... }	Estado o interacción (ej.: al pasar el ratón)

HTML



CSS



s4_ej1

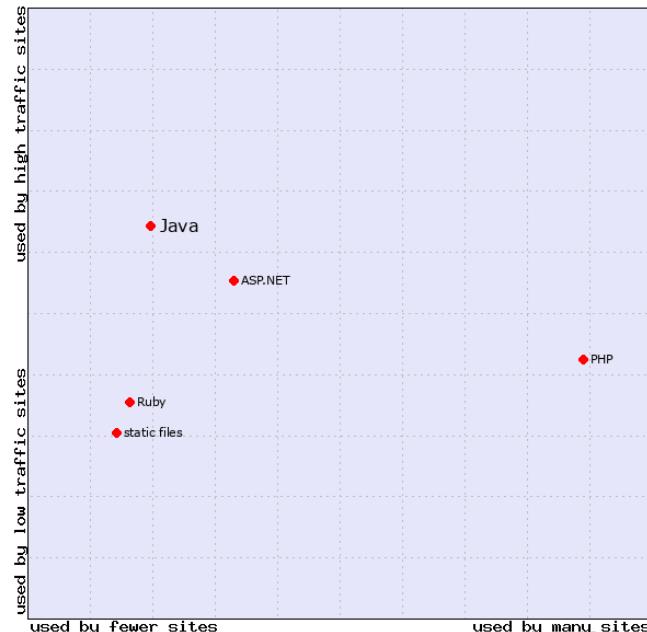
Asociemos el CSS

S4-3

Lenguaje de programación JS

Lenguajes de programación

- Deben diferenciarse lenguajes de lado de cliente y lenguajes de lado de servidor
- Javascript es el lenguaje de programación lado de cliente predominante
- Como en cualquier otra aplicación, en programación web, cada lenguaje de programación del lado de servidor ofrece características propias que vendrán mejor o peor a un proyecto determinado



Lenguaje de programación: Javascript

- **Javascript** permite desarrollar páginas web interactivas/reactivas en los que hay que integrar algún tipo de programación (ej. para validación de datos)
 - HTML y CSS son **suficiente para sitios web estáticos**, ya que no se prevé que cambien frecuentemente
- Fundamentalmente, Javascript es un lenguaje del lado de cliente:
 - Código **escrito como scripts** separados del contenido/estructura y diseño del documento web
 - Código puede ser **incrustado** en marcas HTML o **vinculado** desde un fichero separado del documento, que debe ser accesible al cliente
 - El código **se ejecuta en el cliente*** (navegador web) o en un entorno externo en el dispositivo del usuario

Un *script* es un programa que realiza operaciones sencillas y/o de control:

- ❑ Código **ejecutado en tiempo de ejecución**, sin requerir compilación ni conversión a *bytecode* (algunos lenguajes permiten representación de bajo nivel –previa– como modo de optimización)
 - Por seguridad, **Javascript tiene limitado el acceso a recursos** del cliente
- ❑ Se requiere un **entorno de ejecución** adecuado que interprete y ejecute las instrucciones del *script*
- ❑ Puede hacer uso de un **conjunto amplio de librerías**
- ❑ Generalmente, los lenguajes de *script* **son lentos** (interpretados @*runtime* VS los compilados y *bytecodes*)
 - En Javascript, diferentes motores de Javascript **pueden producir distintos resultados** o efectos
 - El rendimiento de **Javascript depende de la eficiencia del motor Javascript** del cliente y de la capacidad de cómputo del cliente

Lenguaje de programación: Javascript

Historia

- Originalmente desarrollado (nombre del proyecto: *Mocha*) y liberado por Netscape en septiembre de 1995 bajo el nombre de **LiveScript**
- Nombrado “Javascript” en diciembre de 1995, tras un acuerdo de **Netscape y Sun Microsystem** (empresa creadora de Java y adquirida por Oracle en 2009)
 - Javascript **no tiene relación con Java**, más allá de alguna influencia – mezclada con otros lenguajes, fundamentalmente de la familia del C
- Javascript es un **dialecto de ECMAScript**, un lenguaje de *script* estandarizado como especificación **ECMA-262** y **ISO/IEC Std.16262:1997**
- Desarrollados **otros dialectos de ECMAScript**, como JScript (Microsoft) y ActionScript (Adobe)
- Desde 2012, todos los navegadores soportan ECMAScript 5.1. Desde 2015, **ECMAScript lanza ciclos anuales** (ECMAScript 2025)

Lenguaje de programación: Javascript

El lenguaje

- Javascript es un lenguaje multiparadigma, basado en prototipos y dinámico
 - Multiparadigma: Permite programación funcional (con funciones de primera clase), orientada a objetos e imperativa
 - Funciones de primera clase: Paradigma en el que las funciones son los elementos de primer nivel, permitiendo pasar funciones como argumentos de otras, funciones como valores de retorno y asignar funciones a variables
 - Basado en prototipos (programación classless): Los objetos no se instancian, sino que se clonian (a partir de prototipos) o son programados explícitamente por el programador
- Javascript puede incrustarse en el documento HTML (etiqueta "`<script>...</script>`") y vincularse a un fichero externo en el HEAD:

```
<script src="miscript.js"></script>
```

HTML



CSS



JS



S4_ej1

Enlacemos el JS

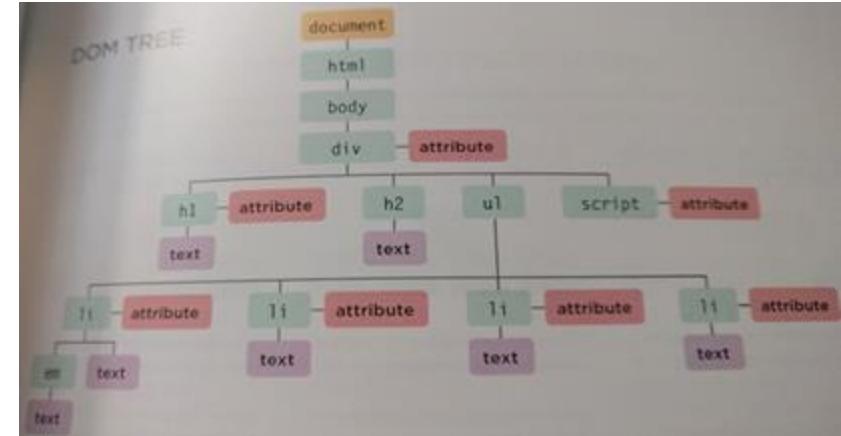
Lenguaje de programación: Javascript *Document Object Model (DOM)*

- DOM es una interfaz independiente del lenguaje y de la plataforma, que permite el acceso y actualización dinámica al contenido, estructura y estilo de los documentos a los lenguajes de script
 - Estándar de W3C: <https://www.w3.org/DOM/>
- Permite manipular, crear, eliminar y cambiar el código de los elementos escritos en HTML y CSS aplicando valores dinámicos

Lenguaje de programación: Javascript

Document Object Model (DOM)

- En DOM, los **elementos de HTML** son **objetos**
 - Tiene las **propiedades** de todos los elementos HTML
 - Tiene **métodos** para acceder a todos los elementos
 - Recoge los **eventos** para todos los elementos HTML
- El árbol DOM tiene **4 tipos de nodos**:
 - Nodo **documento** (página HTML completa)
 - Nodo **elemento**
 - Nodo **atributo** (en las etiquetas de inicio de HTML)
 - Nodo **texto** (dentro de los elementos)



Lenguaje de programación: Javascript *Document Object Model (DOM)*

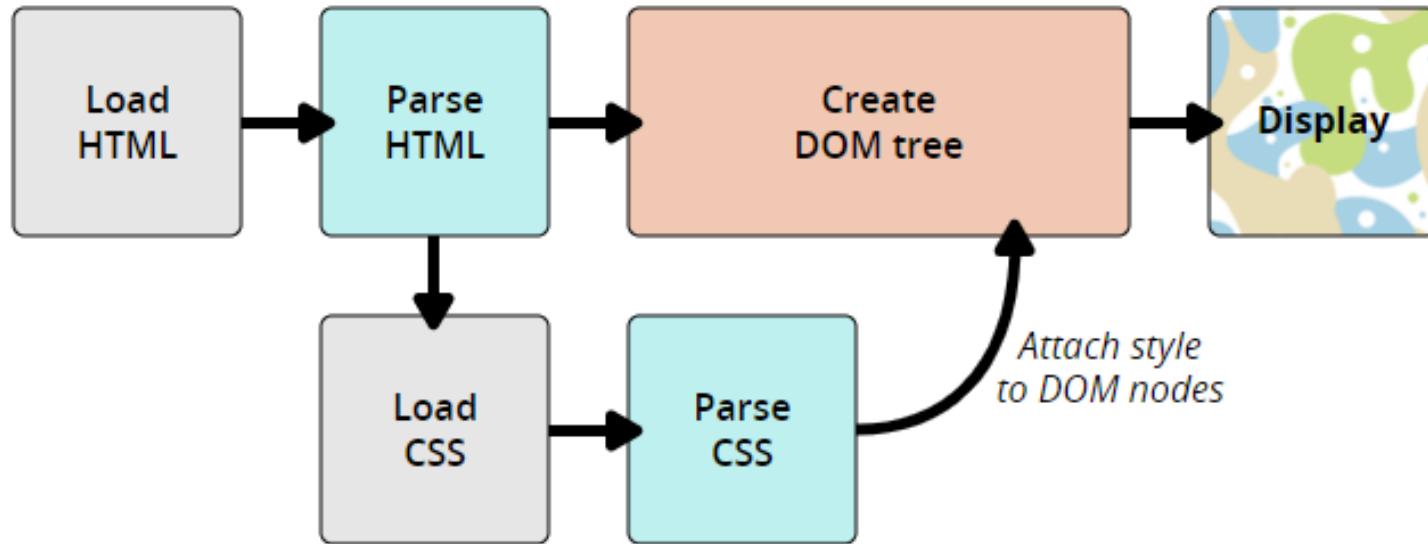
- Javascript DOM permite **consultas en los elementos** HTML

```
const elem = document.getElementById("one");
```

- Recientemente, se han añadido las **consultas de selectores CSS**, lo que permite mayor flexibilidad, aunque no se garantiza que todos los navegadores sean compatibles

```
const first_elem = document.querySelector ("li.one");
```

Resumiendo: ¿Cómo funciona el navegador web?



Fuente: Mozilla.org

S4-4

Otros lenguajes de serialización

Otros lenguajes de serialización: JSON

- **JSON (JavaScript Object Notation)** es un lenguaje de intercambio de datos ligero, basado en texto e independiente del lenguaje:
 - Define un conjunto de reglas de formato para la representación de datos estructurados
 - JSON está derivado directamente de la sintaxis literal de objetos del lenguaje JavaScript (ECMAScript)
 - Popularizado por su uso en APIs y servicios web
- JSON es un estándar IETF (última actualización: dic. 2017)
 - RFC 8259 “The Javascript Object Notation (JSON) Data Interchange Format”: <https://tools.ietf.org/html/rfc8259>

```

<?xml version="1.0" encoding="UTF-8"?>           <!-- XML prolog -->
<message private="true">                      <!-- root element -->
    <from id="alba@miempresa.es">Alba Lopez</from>
    <to id="Roberto@miempresa.es">Roberto Manuel</to>
    <subject>Mañana en mi fiesta de cumpleaños!</subject>
    <content language="spanish">
        Oye, Roberto, no te olvides de llamarme para organizar la fiesta.
    </content>
</message>

```

XML

```

{
    "private": "true",
    "from": {
        "id": "alba@miempresa.es",
        "text": "Alba Lopez"
    },
    "to": {
        "id": "Roberto@miempresa.es",
        "text": "Roberto Manuel"
    },
    "subject": "Mañana en mi fiesta de cumpleaños!",
    "content": {
        "language": "spanish",
        "text": "Oye, Roberto, no te olvides de llamarme para organizar la fiesta."
    }
}

```

JSON



Exploraremos cómo funciona JSON
en la invocación a APIs en Javascript

S4_ej3

Otros lenguajes de serialización: YAML

YAML (“*YAML Ain’t Markup Language*”, 2006) es un lenguaje de serialización de datos diseñado para ser fácil de leer y escribir por personas. Se usa en::

- **Ficheros de configuración** (p.ej. herramientas de CI/CD)
- Intercambio de datos entre aplicaciones
- Definición de infraestructura (*Infrastructure as Code*, como Kubernetes)

Características:

- YAML **no es un lenguaje de etiquetas** (como XML) ni ofrece comandos
- Compatible con JSON desde YAML 1.2
- Formato flexible con sintaxis sencilla basada en indentación
- Serializado en **ficheros con extensión .yaml**
 - ❑ Especificación (no estandarizado): <https://yaml.org/spec/1.2/spec.html>

Recursos y lecturas

- Especificación de XML (W3C)
<https://www.w3.org/TR/xml/>
- Ejemplo de XML Schema
https://www.w3schools.com/xml/schema_example.asp
- ECMAScript Versión 2025 Oficial
<https://tc39.es/ecma262/2025/>
- Novedades de ECMAScript 2025
https://www.w3schools.com/js/js_2025.asp
- Tutoriales de Javascript en Mozilla Dev
<https://developer.mozilla.org/es/docs/Web/JavaScript>
- "The HTML DOM" – *Document Object Model*
<https://www.geeksforgeeks.org/javascript/dom-document-object-model/>

Recursos y lecturas

- Guía de estilo de HTML
https://www.w3schools.com/html/html5_syntax.asp
- Especificación estándar de HTML5 (WHATWG)
<https://html.spec.whatwg.org/multipage/introduction.html>
- Especificación estándar de CSS3
<https://www.w3.org/Style/CSS/specs.en.html>
- CSS3 en Mozilla Dev
<https://developer.mozilla.org/en-US/docs/Web/CSS>