

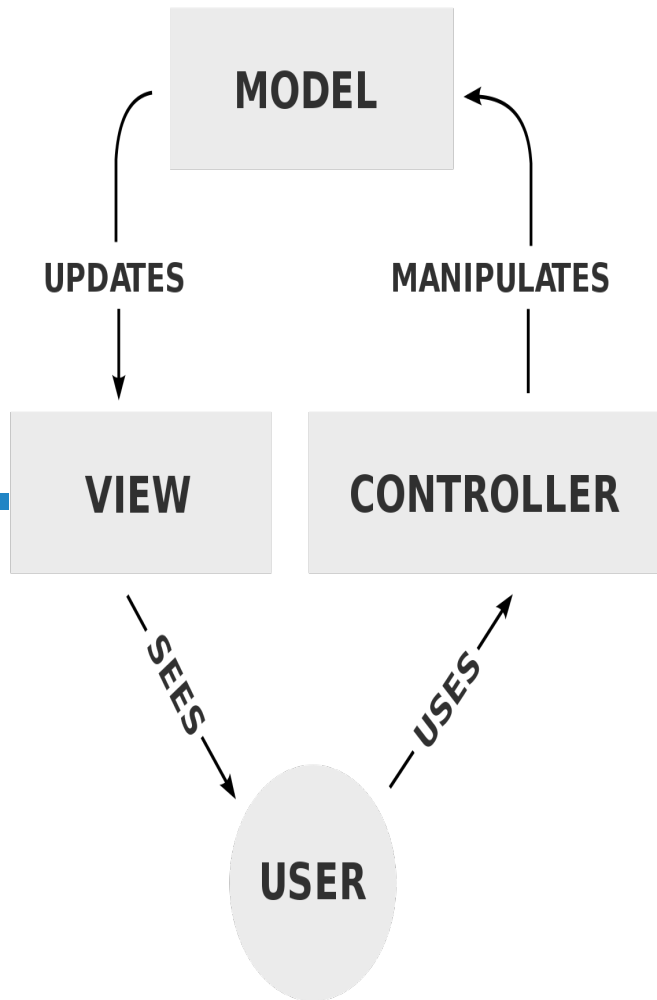


UNIVERSIDAD DE CÓRDOBA

PROGRAMACIÓN WEB – BLOQUE II

Fundamentos del desarrollo Web

Dr. José Raúl Romero Salguero
jrromero@uco.es



Contenidos del Bloque

1. Marcos tecnológicos
2. Lenguajes para la web
3. Principios de diseño y arquitectura

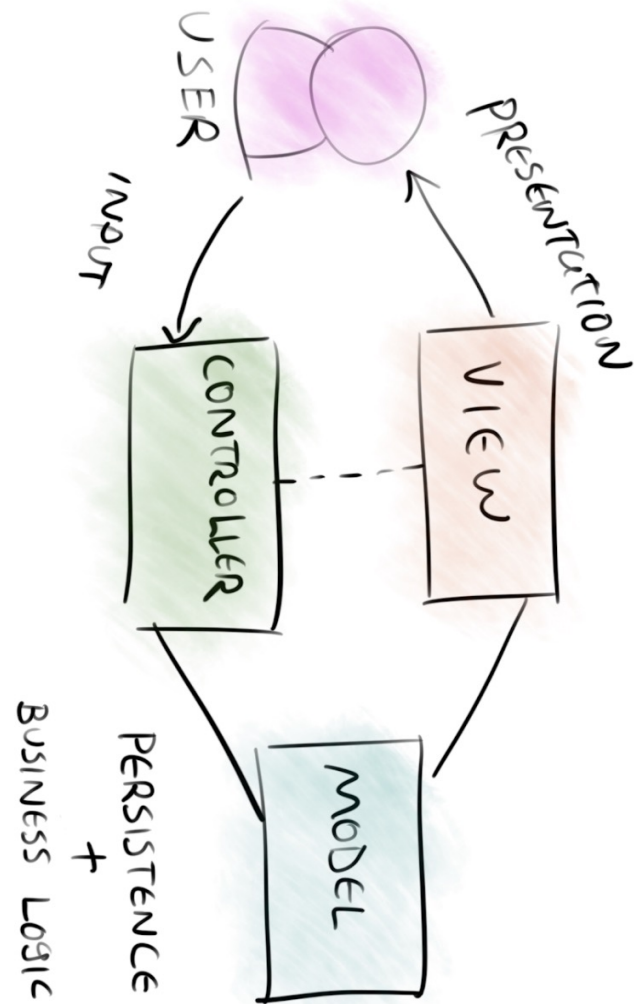


UNIVERSIDAD DE CÓRDOBA

PROGRAMACIÓN WEB – TEMA II-2

Lenguajes para la web

Dr. José Raúl Romero Salguero
jrromero@uco.es



Contenidos

1. Lenguajes de marcado
2. Lenguaje de diseño CSS
3. Lenguaje de programación Javascript
4. Otros lenguajes de serialización

1.

Lenguajes de marcado

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
 - ❑ Al desarrollador le **exige ser riguroso** para su correcto procesamiento (ej. apertura y cierre de etiquetas)

- **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
- Es la **base del desarrollo de APIs** y, en general, ha propiciado el desarrollo de la web (ej. SOAP) y de aplicaciones con intercambio de datos (ej. RSS, Atom, SVG)
- Utilizado por servidores (ej. almacenar ficheros), bases de datos (ej. devolver resultados de consultas) y servicios web
- XML es un **esqueleto para crear otros lenguajes** de marcas
- 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
- ❑ Un procesador (***XML parser***) analiza las marcas y estructura el contenido para pasarlo a la aplicación
- ❑ 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>
```

- El documento empieza con **<?xml .. ?>**
- El elemento message es el **elemento raíz**
- Otros lenguajes, como HTML, siguen una estructura igual para sus etiquetas, atributos y comentarios

- ▷ La estructura de un documento XML puede ser definida externamente en un **documento DTD** (*Document Type Definition*)
 - ❑ DTD es un mecanismo **heredado de SGML** y **actualmente en desuso**
 - ❑ La **definición explícita y externa** del documento XML es necesaria para su correcta validación
- ▷ Actualmente, W3C definió el **XML Schema** (XSD – *XML Schema Definition*) como un nuevo lenguaje de descripción del documento XML
 - ❑ Ofrece un **amplio sistema de tipos**
 - ❑ Permite definir **restricciones** sobre la estructura de los documentos
 - ❑ XSD es un formato **basado en XML** (favorece su procesamiento)

```
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"></xs:schema>
```

- ▷ El **validador de HTML** W3C permite asegurar que un documento **<!DOCTYPE html>** sigue las reglas del esquema de HTML

▷ **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 – basado en texto– más 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 de otras aplicaciones directamente en el documento
- Heredado de XML (**XHTML**) hasta la extinción de HTML4 (agosto'18)
- **HTML 5** es la recomendación actual

Lenguajes de marcado: HTML__

Historia

- HTML se desarrollo en [CERN](#), pasando posteriormente a la IETF durante los 5 primeros años (1990 – 1995)
- Con la creación del [W3C](#), adoptó HTML3 en 1995, avanzando hasta un modelo más pragmático – HTML 3.2 – en 1998
- La W3C vio la necesidad de evolucionar HTML hacia un lenguaje más estructurado basado en XML, obteniendo [HTML4](#)
- Surge [XHTML 1.0](#) (HTML4) en 2000
- Sucesivas evoluciones de XML+HTML derivan en [XHTML DOM Level 1](#) (1998), [DOM Level 2](#) (2000-2003) y [DOM Level 3](#) (2004)
- [XForms](#) surge en 2003, para el desarrollo de formularios sin necesidad de recodificar los motores de renderizado de los navegadores

- El desarrollo de XML permite la especificación de nuevos formatos como **RSS** (después, **Atom**)
- 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 HTML5 (last update: 13 oct. 2021):
<https://html.spec.whatwg.org/multipage/>
- Tutorial de HTML5: <https://www.w3schools.com/html/>

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 MIME **text/css**

➤ Tres posibles **métodos de aplicación de estilos**:

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

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

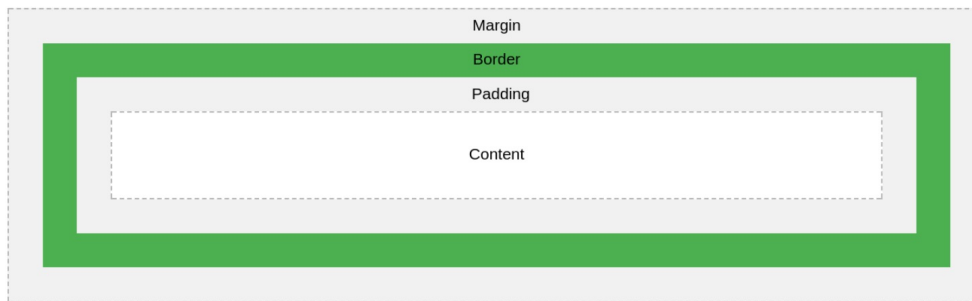
- ❑ Interno, incrustando CSS como contenido de la etiqueta **<style ...>** en la cabecera (**<head>...</head>**) del documento HTML

```
<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

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

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 (**sitios dinámicos**) 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 legible y modificable por el usuario, que realiza operaciones sencillas y/o de control sobre la operación de otros programas:
 - ❑ 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 motivos de 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

- Además, los lenguajes de script tienen como **características**:
 - ❑ Variables, funciones y métodos suelen ser **no tipados** (conversión automática de tipos)
 - ❑ Ofrecen **estructuras de datos pre-construidas**
 - ❑ Permiten generar, cargar e **interpretar código fuente @runtime** (ej. función **eval ()** de Javascript)

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** (ES12 / ECMAScript 2021)

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 clonan (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="miscrypt.js"></script>
```

```
var genericAnimal = Object.create(null);
genericAnimal.name = 'Animal';
genericAnimal.gender = 'female';
genericAnimal.description = function() {
    return 'Gender: ' + this.gender + '; Name: ' + this.name;};

var cat = Object.create(genericAnimal);
cat.purr = function() {
    return 'Purrrrr!'; };

var colonel = Object.create(cat);
colonel.name = 'Colonel Meow';

var puff = Object.create(cat);
puff.name = 'Puffy';
```

```
// Esta función añade un texto cada vez que se pulsa un botón
// Código en el <script> del HEAD
function createParagraph() {
    var para = document.createElement('p');
    para.textContent = '¡Presionaste el Botón!';
    document.body.appendChild(para);
}

// En documento HTML, al definir el botón del formulario
<button onclick="createParagraph()">¡Presioname!</button>
```

Fuente: Mozilla.org

Lenguaje de programación: Javascript_

Document Object Model (DOM)

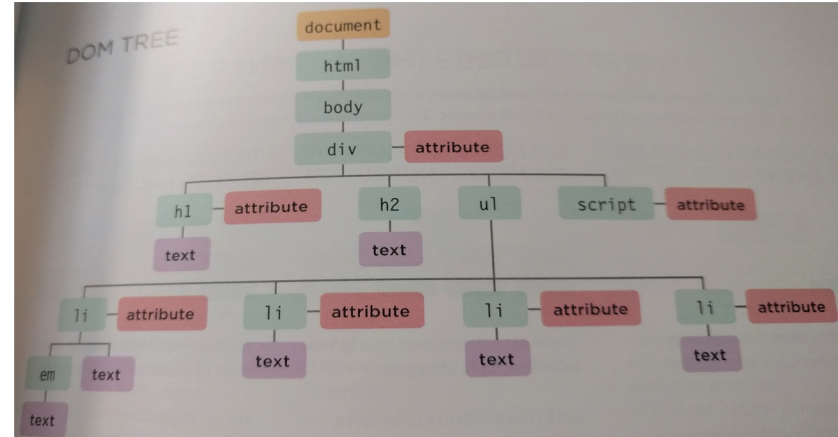
- **DOM** es una **interfaz independiente del lenguaje y de la plataforma**, que permite el acceso y actualización dinámica al contenido, estructura y estilo de los documentos a los lenguajes de *script*
 - Estándar de W3C: <https://www.w3.org/DOM/>
- En Javascript, DOM es una **API para documentos HTML y XML**, que proporciona una representación estructural del documento
- 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" );
```

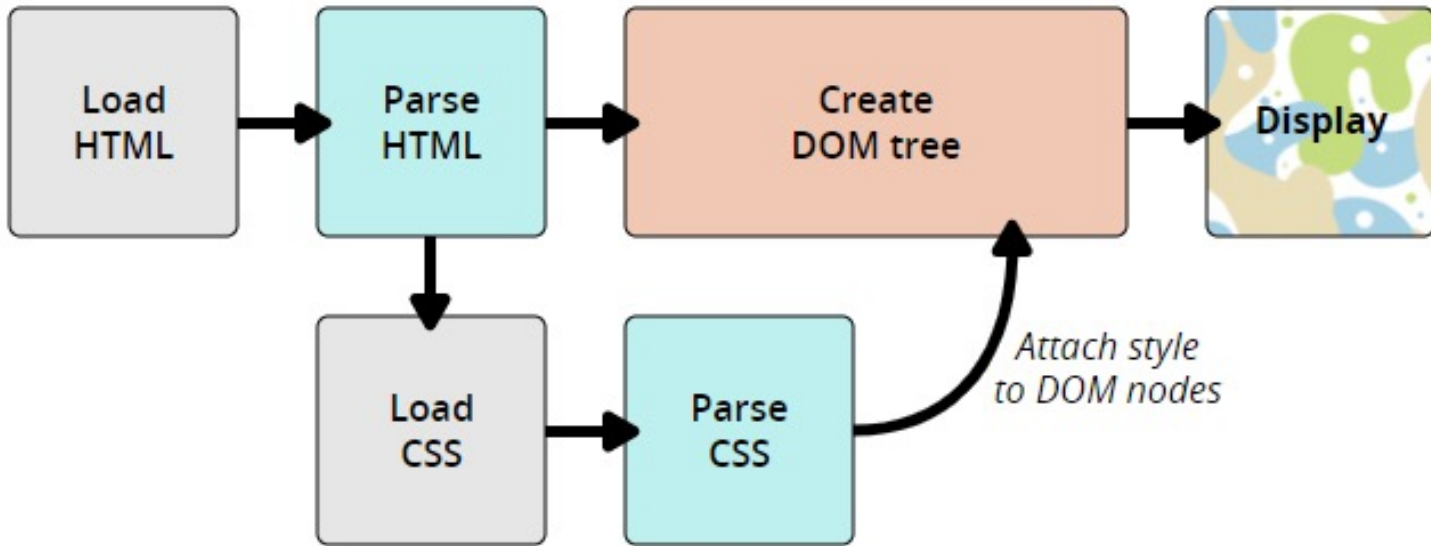
- Recientemente, se han añadido las **consultas de selectores CSS**, lo que permite mayor flexibilidad, aunque no se garantiza que todos los navegadores sean compatibles

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

- ¡La compatibilidad con distintos clientes ya no suele ser problema!

Resumiendo:

¿Cómo funciona el navegador web?



Fuente: Mozilla.org

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
 - ❑ Derivado de ECMAScript
 - ❑ Representa los datos como un conjunto de objetos de Javascript
 - ❑ Popularizado por su uso en APIs y servicios web
- JSON es un estándar IETF (última actualización: 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."
  }
}
```

JSON³⁵

Otros lenguajes de serialización: YAML

- **YAML** (“YAML Ain’t Markup Language”, 2006) es un lenguaje de serialización –legible por el humano– utilizado en general para ficheros de configuración:
 - ❑ Es un **lenguaje orientado a líneas** (*line-oriented*) que ofrece puramente una representación de datos (no necesariamente estructurados)
 - ❑ YAML **no ofrece comandos** (no es ejecutable)
 - ❑ YAML **no es un lenguaje de etiquetas** (como XML)
 - ❑ Utiliza una **sintaxis mínima** diferenciada de XML, más cercana al estilo de formato de Python
 - ❑ Serializado en **ficheros con extensión .yaml**
- Desde YAML 1.2, **YAML dice cumplir con JSON** como un subconjunto oficial del mismo
 - ❑ Especificación (no estandarizado): <https://yaml.org/spec/1.2/spec.html>



Programación Web

Fundamentos del desarrollo web__ Curso 2021/22