

PCW

PROGRAMACIÓN DEL CLIENTE WEB

Tema 02 - HTML



Dept. de Ciència de la Computació i Intel·ligència *a*rtificial
Dpto. de Ciencia de la Computación e Inteligencia *a*rtificial



Universitat d'Alacant
Universidad de Alicante

HTML

HTML

1. Introducción
2. Sintaxis HTML
3. Estructura de un documento HTML
4. Elementos de texto
5. Enlaces
6. Listas
7. Tablas
8. Imágenes
9. Formularios

Introducción

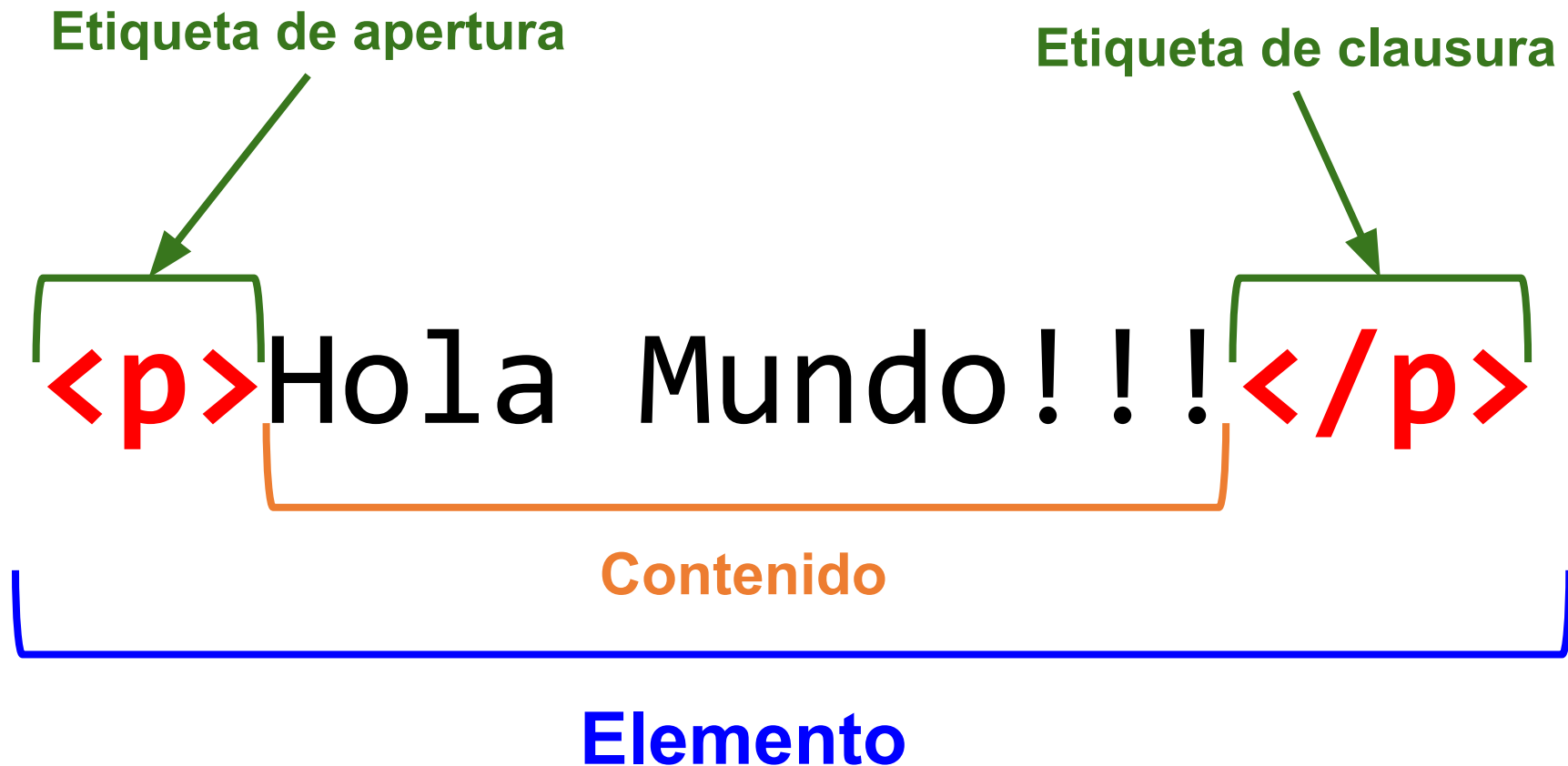
Introducción

¿Qué es HTML?

- **HTML: HyperText Markup Language**
- **Lenguaje de marcas** utilizado para crear la estructura de una página web
- **Se compone de una serie de elementos** que se utilizan para delimitar distintas partes del contenido.
- **Los elementos HTML aportan semántica** al contenido que delimitan.

Introducción

Anatomía de un elemento HTML



Introducción

Anidación de elementos HTML

Los elementos HTML se pueden anidar, colocando uno dentro de otro de forma adecuada.

```
<p>Hola <strong>Mundo!!!</strong></p>
```

```
<section>  
  <h2>  
    Hola <strong>Mundo!!!</strong>  
  </h2>  
</section>
```

Introducción

Elementos de bloque vs elementos de línea

Los elementos HTML pueden ser de dos tipos:

- Elemento de bloque: **block**. Permite delimitar partes grandes o importantes de contenido y será el único elemento que se vea en pantalla en la misma *línea* del documento.
- Elemento de línea: **inline**. Permite la existencia de otros elementos `inline` en la misma *línea* del documento.

Nota: Todo elemento HTML, independientemente del tipo que sea, puede hacerse de tipo bloque o de tipo línea mediante CSS

Introducción

Elementos HTML *vacíos* (*void elements*)

- Son elementos HTML que no tienen etiqueta de clausura, por lo que no tienen contenido, sólo tienen etiqueta de apertura.
- Su configuración se indica mediante **atributos** dentro del propio elemento de apertura.

Ejemplo:

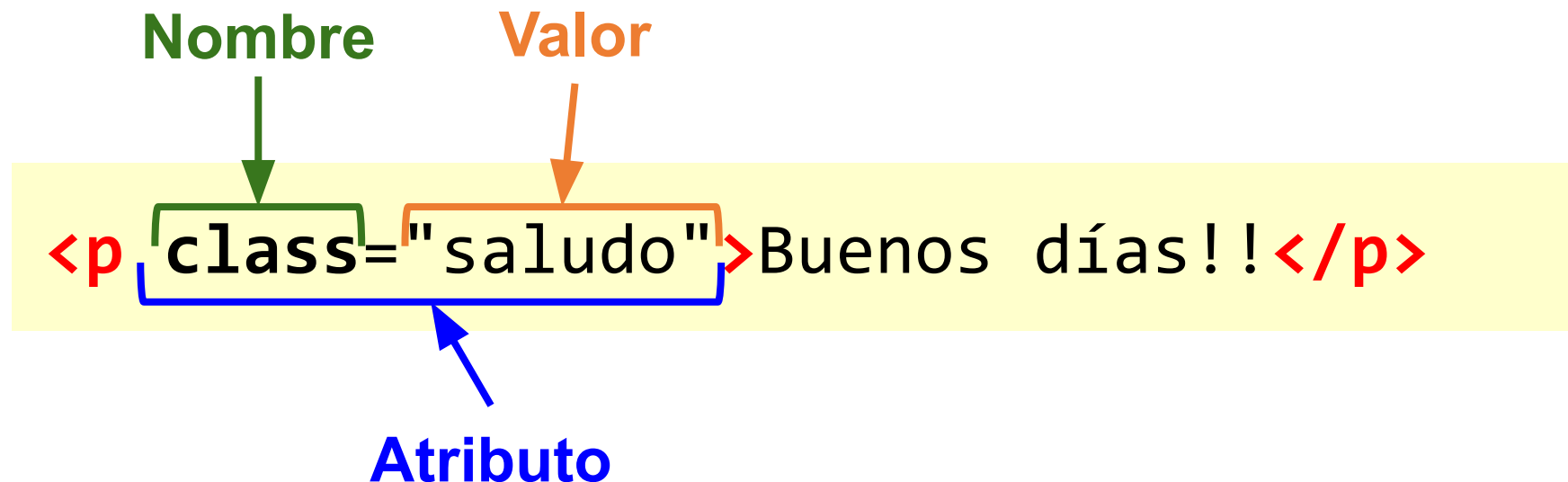
```

```

Introducción

Atributos

- Los elementos HTML pueden tener atributos.
- Los atributos **aportan información extra** sobre el elemento. Permiten, entre otras cosas, ajustar su configuración y aplicar estilo CSS.



Introducción

Atributos

Atributos booleanos

- Solo pueden tener un valor que, por lo general, es el nombre del atributo.
- Se pueden escribir sin valor.
- Aportan una característica preestablecida.

Ejemplo:

```
<input type="text" disabled="disabled">
```

Es lo mismo que:

```
<input type="text" disabled>
```

Atributo booleano

Introducción

Atributos

Atributos enumerados

- La lista de valores que pueden tomar es finita.
- Se pueden *enumerar* los valores que puede tomar.
- Hay que escribirlos con valor.

Ejemplo:

```
<input type="text" disabled="disabled">
```

Atributo enumerado

Introducción

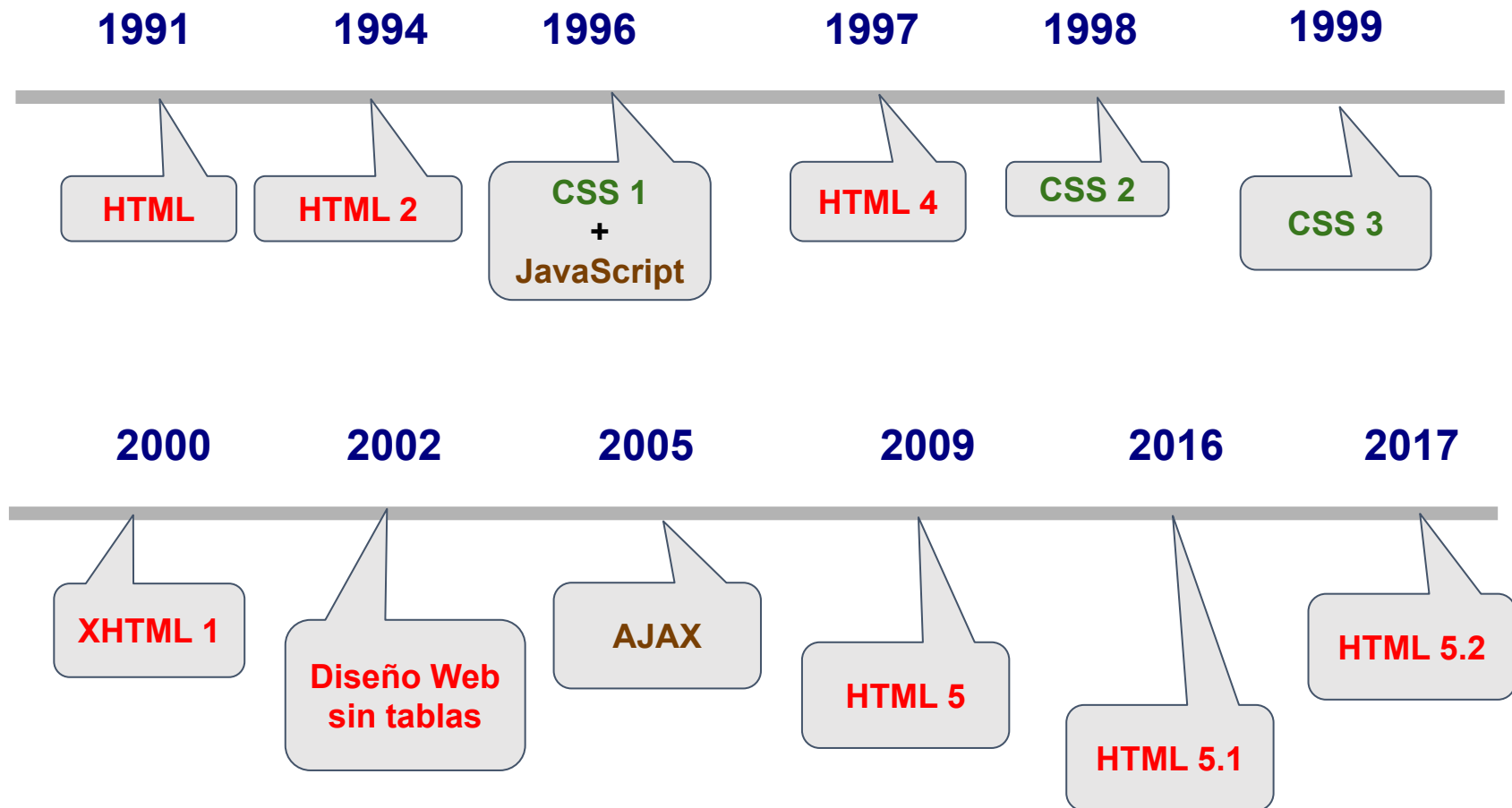
Atributos

Atributos globales

- Se pueden utilizar en todos los elementos HTML.
- Algunos de los más importantes son:
 - **id**. Permite asignar un identificador único al elemento que lo diferencie de todos los demás del documento HTML.
 - **class**. Permite asignar una clase al elemento. Se puede asignar la misma clase a distintos elementos para aplicarles el mismo estilo mediante reglas CSS.
 - **title**. Permite asignar un texto informativo al elemento, que se mostrará cuando el usuario sitúe el ratón sobre él.
 - **data-***. Permite asignar un atributo de datos personalizado para guardar información. El * representa cualquier cadena de texto que se utilice para completar el nombre final del atributo.

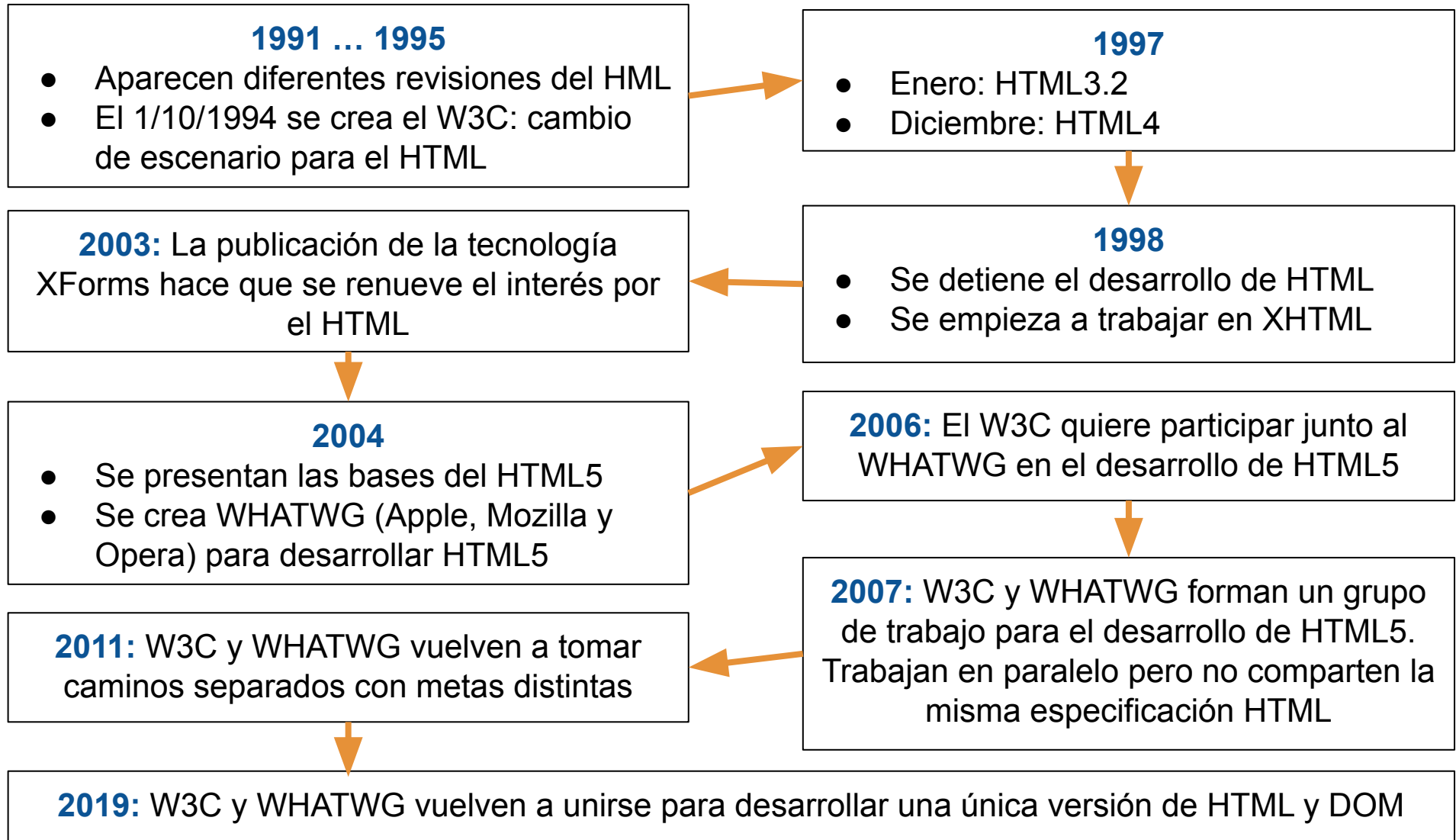
Introducción

Evolución de las Tecnologías Web



Introducción

Evolución del lenguaje HTML



Introducción

Organismos desarrolladores de HTML

WHATWG

**WEB HYPERTEXT
APPLICATION
TECHNOLOGY
WORKING GROUP**

<https://whatwg.org/>

W3C

**WORLD WIDE WEB
CONSORTIUM**

<https://www.w3.org/>

ESPECIFICACIÓN OFICIAL DE HTML

<https://html.spec.whatwg.org/multipage/>

Introducción

¿HTML5 es HTML?

- Sí. HTML5 fue la última gran versión antes de decidir llamarlo simplemente HTML
- HTML5 introdujo un conjunto de nuevas tecnologías para la web (*Video, audio, canvas, drag and drop, ...*), que pretendía proporcionar soporte para el desarrollo de aplicaciones web.
- HTML5 apareció como recomendación W3C (28/10/2014)
- HTML 5.1 => Recomendación W3C (01/11/2016).
- HTML 5.2 => Recomendación W3C (14/12/2017).
- **HTML** Standard => W3C y WHATWG (28/05/2019).

HTML



Introducción

Tecnologías web introducidas por HTML5



Semántica

- Aporta significado a la estructura
- Conjunto más rico en etiquetas: nuevas etiquetas semánticas



Sin conexión y almacenamiento

- Las aplicaciones web funcionan más rápido e, incluso, sin conexión gracias a *Service Workers*, *Local Storage*, *Indexed DB* y *File API*



Acceso al dispositivo

- *Geolocation API*
- Captura de audio/video
- Datos locales: contactos, eventos, ...



Conectividad

- Mejor comunicación y más rápida
- Chats en tiempo real
- *Web Sockets*
- *Server-Sent Events (SSE)*



Multimedia

- Elementos *audio* y *video* compatibles con aplicaciones y sitios web



Gráficos y efectos 2D y 3D

- *SVG*, *Canvas*, *Web GL*, *CSS3 3D*



Rendimiento e integración

- Funcionamiento más rápido de aplicaciones web y contenido web dinámico
- *Web Workers*
- *XMLHttpRequest 2*



CSS3

- Gran diversidad de estilos y efectos, mejorando la aplicación web sin sacrificar la estructura o el rendimiento
- Tipografía flexible: *Web Open Font Format (WOFF)*

Sintaxis HTML

Sintaxis HTML

- **Los nombres de los elementos** (o etiquetas) HTML y sus atributos son invariantes a mayúsculas/minúsculas. Por ejemplo: `<div>`, `<DIV>`, `<dIV>` o `<Div>` es el mismo elemento HTML.
- Los valores de los **atributos** se pueden escribir entre comillas, simples o dobles; o también sin ellas, siempre y cuando no contengan un espacio en blanco ni los caracteres `"`, `'`, ```, `=`, `<`, `>`. Incluso, se puede dejar únicamente el nombre del atributo si su valor es una cadena vacía.
- Los elementos *void* pueden llevar (`</>`) o no (`<>`) la barra final.
- Algunos elementos HTML se pueden utilizar sin elemento de cierre en determinadas circunstancias.

Sintaxis HTML

Ejemplo (aunque no a seguir) de uso válido de sintaxis HTML:

```
<!DoCtYpe hTm1>
<Html lAng='es'>
  <head>
    <tITle>Sintaxis poco cuidada, pero
      v&aacute;lida</title>
  </HEAD>
  <boDY>
    <p>Ah&iacute; va!.
    <P><bR/><BR>Los elementos <I>void</i> pueden tener
      barra final o no.</p>
    <ol>
      <LI class=green>Butter
      <li CLASs="red">Bread</LI>
      <lI claSS=blue>Milk
    </OL>
  </BodY>
</htML>
```

Sintaxis HTML

Otro ejemplo de sintaxis HTML válida, aunque no recomendable:

```
...  
<table>  
  <thead>  
    <tr><th>ID<th>Measurement<th>Average<th>Maximum  
  <tbody>  
    <tr><td><th scope=rowgroup>Cats<td><td>  
    <tr><td>93<th scope=row>Legs<td>3.5<td>4  
    <tr><td>10<th scope=row>Tails<td>1<td>1  
  <tbody>  
    <tr><td><th scope=rowgroup>English  
      speakers<td><td>  
    <tr><td>32<th scope=row>Legs<td>2.67<td>4  
    <tr><td>35<th scope=row>Tails<td>0.33<td>1  
</table>  
...
```

Sintaxis HTML

- **Caracteres especiales** (á, ü, ñ, ¿, ...). Es conveniente escribirlos mediante su *referencia numérica* o su *nombre de entidad*.

Carácter español	Nombre de entidad	Referencia numérica	Carácter español	Nombre de entidad	Referencia numérica
Á	Á	Á	Ú	Ú	Ú
á	á	á	ú	ú	ú
É	É	É	Ü	Ü	Ü
é	é	é	ü	ü	ü
Í	Í	Í	«	«	«
í	í	í	»	»	»
Ñ	Ñ	Ñ	¿	¿	¿
ñ	ñ	ñ	¡	¡	¡
Ó	Ó	Ó	€	€	€
ó	ó	ó	Pts		₧

Sintaxis HTML

Recomendaciones:

- Escribir los nombres de los elementos y sus atributos en minúsculas.
- Escribir los valores de los atributos de los elementos HTML entre comillas dobles.
- Escribir los elementos de cierre del correspondiente elemento de apertura, aunque no sea necesario.
- No utilizar barra final en los elementos *void* de HTML.
- Escribir los caracteres especiales mediante su código de referencia o nombre de entidad.
- Usar un servicio de validación de documentos HTML:
<http://validator.w3.org/>
- Usar un servicio de validación de las reglas CSS:
<https://jigsaw.w3.org/css-validator/>

Estructura de un documento HTML

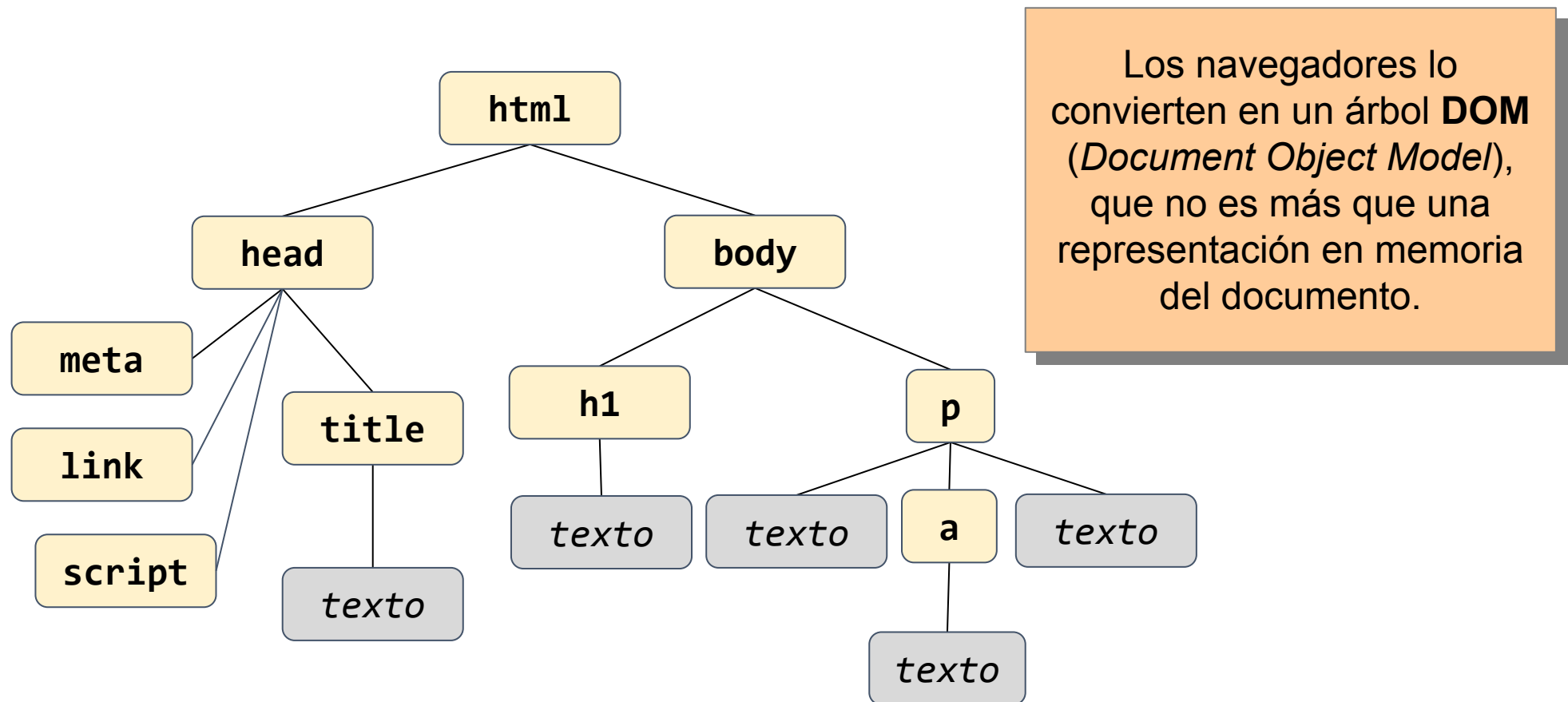
Estructura de un documento HTML

Estructura básica

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <link rel="stylesheet" href="estilo.css">
    <script src="archivo.js"></script>
    <title>P&acute;gina de ejemplo</title>
  </head>
  <body>
    <h1>P&acute;gina de ejemplo</h1>
    <p>Esto es una <a href="demo.html">simple</a>
      muestra.</p>
    <!-- esto es un comentario -->
  </body>
</html>
```

Estructura de un documento HTML

Un documento HTML es un árbol de elementos y texto. Cada elemento se corresponde con una etiqueta HTML de apertura y una de cierre, aunque en algunos casos esto puede no cumplirse.



Estructura de un documento HTML

- Requiere la especificación del **tipo de documento** mediante una única declaración *DOCTYPE*.

`<!DOCTYPE html>`

La declaración se ha simplificado con respecto a versiones anteriores de html. Es invariante a mayúsculas o minúsculas, aunque todo lo que no sea escribirla en mayúsculas se considera una *declaración obsoleta*. Su única función es indicar al navegador que el documento se ha escrito utilizando la sintaxis HTML.

Ejemplo:

```
<!DOCTYPE html>
<html>
  <head>
    ...
```

Estructura de un documento HTML

- **Título de la página.** Es obligatorio especificar un título para la página web mediante el elemento `<title>` dentro del elemento `<head>`. Además, es importante para el posicionamiento en los buscadores seguir las siguientes recomendaciones:
 - Utilizar **entre 30 y 60 caracteres**, como regla general.
 - Utilizar **palabras clave**, pero sin abusar, para no ser penalizado.
 - Evitar poner todo en mayúsculas para no ser penalizado. **Utilizar mayúsculas y minúsculas** como en un texto normal.
 - **Evitar repetir el mismo título** para todas las páginas del sitio.
 - Debe **resumir el contenido de la página** de forma adecuada.

Ejemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Escribir un buen título para una página web</title>
    ...
```

Estructura de un documento HTML

- Aunque no es obligatorio, se recomienda especificar **la codificación de caracteres**. En el HTML estándar se ha simplificado al máximo esta tarea y se realiza mediante un elemento `<meta>`:

`<meta charset="utf-8">`

Debe estar dentro de los primeros 1024 bytes del documento. Normalmente es la primera línea tras el elemento `<head>`.

Ejemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Escribir un buen título para una página web</title>
    ...
```

Estructura de un documento HTML

- Elementos `<meta>`

Proporcionan información en forma de **metadatos** sobre el documento html. Se declaran en el `<head>` del documento.

Algunos de los elementos `<meta>` más importantes son:

- Para especificar la codificación de caracteres:
`<meta charset="utf-8">`
- Nombre del autor del documento HTML:
`<meta name="author" content="Nombre del autor">`
- Breve resumen del contenido de la página, entre 70 y 130 caracteres:
`<meta name="description" content="Texto resumen">`
- Controlar cómo indexan y recorren el sitio web los motores de búsqueda:
`<meta name="robots" content="noindex, nofollow">`
- Evitar que el navegador redimensione el contenido en dispositivos móviles:
`<meta name="viewport" content="width=device-width, initial-scale=1">`

Estructura de un documento HTML

- **Especificación del idioma.** Se realiza mediante el atributo `lang` en el elemento `<html>`:

```
<html lang="es">
```

El valor del atributo `lang` es el código ISO 639-1 del idioma en el que se encuentra el contenido del elemento `html` al que se aplica el atributo.

Ejemplo:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    ...
```


Estructura de un documento HTML

- **Enlazando con archivos externos**

- Etiqueta `<script>`. Permite enlazar con un fichero del lenguaje de script *JavaScript*.
- Etiqueta `<link>`. Permite enlazar con un fichero de hojas de estilo en cascada *CSS*.

Ejemplo:

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <script src="archivo.js"></script>
    <link rel="stylesheet" href="estilo.css">
  ...
```

Estructura de un documento HTML

Elementos estructurales

Estructura de un documento HTML

Elementos estructurales

<header>	Permite añadir contenido de encabezamiento a un documento, artículo, sección, ...
<footer>	Representa el pie de un documento, artículo, sección, ...
<nav>	Encierra enlaces de navegación o cualquier otro contenido que permita desplazarse, únicamente, dentro del mismo sitio web. Se trata de una sección con enlaces de navegación.
<main>	Representa el contenido principal del documento que no va a repetirse en otros documentos relacionados.
<section>	Se utiliza para separar distintas áreas en base a funcionalidad o contenido. Es algo más genérica que el resto de etiquetas.
<article>	Sirve para encerrar fragmentos individuales de contenido relacionado, tales como entradas individuales de un blog.
<aside>	Alberga información relacionada con el flujo principal del documento, pero no tiene cabida en él.

Estructura de un documento HTML

Elementos estructurales

<header>

Permite añadir contenido de encabezamiento, no sólo a una página web, sino también a cualquier otro elemento estructural de HTML.

Ejemplo:

```
<header>
  <h1>Título principal</h1>
  <h2>Descripción</h2>
  <p>Última modificación: <time datetime="2014-02-11">el
    martes</time></p>
</header>
```

Estructura de un documento HTML

Elementos estructurales

<nav>

Encierra enlaces de navegación o cualquier otro contenido (por ejemplo, un formulario de búsqueda) que permita desplazarse, únicamente, dentro del mismo sitio web, ya sea a otras páginas, o a otras zonas dentro de la misma página.

Ejemplo:

```
<header>
  <h1>Título principal</h1>
  <h2>Descripción</h2>
  <p>Última modificación: <time datetime="2014-02-11">el martes</time></p>
  <nav>
    <h2>Navegación</h2>
    <ul>
      <li><a href="home.html">Home</a></li>
      <li><a href="about.html">About</a></li>
      <li><a href="contact.html">Contact</a></li>
    </ul>
  </nav>
</header>
```

Estructura de un documento HTML

Elementos estructurales

`<main>`

Se utiliza para encapsular el contenido principal del documento que no se repite en otros documentos relacionados. Un ejemplo sería el contenido de una página web, propio de esa página y que no se repetirá en el resto de páginas, tal y como ocurre con barras de navegación, secciones de enlaces, barras de búsqueda, etc.

No puede haber más de un elemento `<main>` por documento

Ejemplo:

```
<main>
  <!-- Aquí va el contenido principal y propio del documento -->
</main>
```

Estructura de un documento HTML

Elementos estructurales

<section>

Se utiliza para separar distintas áreas en base a funcionalidad o contenido, o dividir un artículo o una historia en diferentes secciones. Es algo más genérica que el resto de etiquetas pero tiene más significado semántico que <div>. Si lo que se desea es simplemente aplicar estilo, se recomienda utilizar <div>.

Debe llevar un título o un encabezado (h1-h6) que identifica la temática del contenido.

Ejemplo:

```
<section>
  <!-- Aquí va el contenido -->
</section>
```

Estructura de un documento HTML

Elementos estructurales

<article>

Está relacionado con <section>, pero tiene mayor significado semántico. El elemento <article> sirve para encerrar fragmentos individuales de contenido relacionado, tales como entradas individuales de un blog, vídeos, imágenes o noticias.

Ejemplo:

```
<section>
  <!-- Sección de noticias de deportes -->
  ...
  <article>
    <!-- Primera noticia -->
  </article>
  <article>
    <!-- Segunda noticia -->
  </article>
  ...
</section>
```


Estructura de un documento HTML

Elementos estructurales

La versatilidad de las etiquetas `<section>` y `<article>` es todavía mayor, ya que se pueden anidar secciones dentro de artículos y viceversa.

Ejemplo:

```
<article>
  <!-- Noticia de deportes con una estructura determinada -->
  <section class="introduccion">
    ...
  </section>
  <section class="contenido">
    ...
  </section>
  <section class="resumen">
    ...
  </section>
</article>
```

Estructura de un documento HTML

Elementos estructurales

<aside>

Alberga información relacionada con el contenido principal del documento, pero que no es necesaria para que éste tenga sentido por sí mismo. Suele tratarse de información complementaria.

Ejemplo:

```
<section>
  <h2>Comentarios</h2>
  <article>
    <!-- Primer comentario sobre la noticia -->
  </article>
  <article>
    <!-- Segundo comentario sobre la noticia -->
  </article>
</section>
<aside>
  <h2>Categorías</h2>
  <span><a href="/futbol.html">Fútbol</a></span>
</aside>
```

Estructura de un documento HTML

Elementos estructurales

<footer>

Al igual que ocurría en el caso de <header>, el propósito del elemento <footer> está bastante claro y no es otro que encerrar el contenido del pie del documento, o bien del elemento estructural que lo contiene.

Ejemplo:

```
<article>
  <header>
  </header>
  ...
  <footer>
    <a href="./">Volver a index</a>
  </footer>
</article>
```

Elementos de texto

Elementos de texto

- HTML tiene diferentes elementos que permiten añadir texto a un documento.
- Cada elemento aporta una **semántica** propia que indica la importancia del texto que encapsula.
- Esta semántica es tenida en cuenta por parte de los buscadores a la hora de indexar las páginas web. Está directamente relacionado con el **S.E.O.** (Search Engine Optimization), que es la forma que tienen los motores de búsqueda de clasificar y ordenar los sitios web.

Elementos de texto

Elemento **<p>**

- Es el elemento más básico para añadir texto en un documento.
- Se utiliza para encapsular un párrafo de texto normal.
- Se pueden poner todos los que se quieran, pero no se pueden anidar.
- Visualmente, no cambian el texto, pero añaden un margen por defecto arriba y abajo del párrafo.

Ejemplo:

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vitae orci id mauris finibus pellentesque ac non orci.</p>
```

```
<p>Nam rutrum nisi vitae feugiat venenatis. Nunc sodales sagittis placerat. Suspendisse luctus orci dui, eu tempus odio tincidunt vitae.</p>
```

Elementos de texto

Elementos de encabezado `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`

- Son elementos que aportan mayor semántica al texto que encapsulan.
- Se utilizan generalmente para poner títulos o encabezados de secciones.
- El encabezado de primer nivel, `<h1>`, es el que más importancia tiene, siendo `<h6>` el que menos tiene.
- Visualmente, por defecto muestran el texto en negrita y en diferentes tamaños según el encabezado empleado, siendo `<h1>` el que muestra el texto más grande y `<h6>` el que lo muestra más pequeño.

Ejemplo:

```
<h1>Lorem ipsum</h1>
```

```
<h2>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed vitae orci id  
mauris finibus pellentesque ac non orci.</h2>
```

Elementos de texto

Elementos a nivel de texto

Este tipo de elementos pueden usarse dentro del elemento `<p>` o cualquiera de los encabezados `<h1>` ... `<h6>`. Algunos aportan semántica extra y/o cambian la forma de visualizar el texto.

■ Elemento ``

- Representa una sección del texto sobre la que se quiere atraer la atención del lector.
- No aporta importancia (significado semántico) adicional.
- Visualmente muestra el texto encapsulado en negrita.

Ejemplo: `<p>Lorem ipsum dolor sit amet consectetur</p>`

■ Elemento ``

- Representa una importancia fuerte del texto que contiene.
- Añade significado semántico al texto que contiene.
- Visualmente muestra el texto encapsulado en negrita.

Ejemplo: `<p>Lorem ipsum dolor sit amet consectetur</p>`

Elementos de texto

Elementos a nivel de texto

■ Elemento **<i>**

- Representa una sección de texto que se sale de la prosa normal del párrafo. Por ejemplo, un término técnico, una palabra o frase en otro idioma, etc.
- No añade significado semántico al texto que contiene.
- Visualmente muestra el texto encapsulado en cursiva.

Ejemplo: `<p>Lorem ipsum <i>dolor</i> sit amet consectetur</p>`

■ Elemento ****

- Representa un énfasis en la voz sobre el texto que contiene.
- Aporta significado semántico al texto.
- Visualmente muestra el texto encapsulado en cursiva.

Ejemplo: `<p>Lorem ipsum dolor sit amet consectetur</p>`

Elementos de texto

Elementos semánticos a nivel de texto

■ Elemento semántico ****

- Se suele utilizar para aplicar estilo a partes de un texto o agrupar elementos *inline*.
- No añade significado semántico al texto que contiene.
- Visualmente no modifica el texto que contiene.

Ejemplo:

```
<p>Lorem ipsum <span class="ident">dolor</span> sit amet  
consectetur</p>
```

■ Elemento semántico **<u>**

- Se suele utilizar para representar una sección del texto que no expresa bien algo, por ejemplo un un texto mal escrito.
- No añade significado semántico al texto que contiene.
- Visualmente muestra subrayado el texto que contiene.

Ejemplo:

```
<p>Lorem ipsum <u>ddolor</u> sit amet consectetur</p>
```

Elementos de texto

Elementos semánticos a nivel de texto

■ Elemento semántico **
**

- Representa un salto de línea en el texto.
- Es un elemento void.

Ejemplo:

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.<br>Sed vitae orci id mauris finibus pellentesque ac non orci.</p>
```

■ Elemento semántico **<wbr>**

- Representa un lugar en el texto por el que se puede partir la línea de texto y seguir en la siguiente línea.
- Es un elemento void.

Ejemplo:

```
<p>Lorem ipsum dolor sit amet, con<wbr>sectetur adipiscing elit.<br>Sed vitae orci id mauris finibus pellen<wbr>tesque ac non orci.</p>
```

Elementos de texto

Elementos semánticos a nivel de texto

■ Elemento semántico **<time>**

- Representa tiempo (años, meses, fechas, horas, ...) con respecto al texto que contiene.
- Por lo general se utiliza con su atributo **datetime**, que permite especificar el valor del contenido en un formato estándar comprensible para los motores de búsqueda.

Ejemplo: `<p>Inicio de las clases: <time datetime="2023-02-01 15:00">1 de febrero, a las 3 de la tarde</time></p>`

El formato del atributo **datetime** es el siguiente: **AAAA-MM-DDThh:mm:ssDZH**

AAAA: Año, con cuatro cifras

hh: Hora, de 00 a 23

MM: Mes, con dos cifras

mm: Minutos, de 00 a 59

DD: Día del mes, con dos cifras

ss: Segundos, de 00.0 a 59.99999

DH: Designador de Zona Horaria. Posibles valores: Z (hora Zulu ó GMT); +02:00 (GMT+02:00); -05:00 (GMT-05:00)

T: Carácter separador. Puede ser sustituido por un (y sólo uno) espacio.

Enlaces

Enlaces

Para incluir enlaces a otras páginas web, documentos, etc, en un documento HTML, se utiliza el elemento `<a>`. Permite encapsular tanto texto como imágenes y cualquier otro elemento html que no sea interactivo.

Los atributos más utilizados que admite son los siguientes:

- **href**. Sirve para indicar la dirección del recurso (página web, documento, etc)
- **target**. Permite indicar si el recurso se quiere abrir en una nueva pestaña/ventana del navegador utilizándolo con el valor `_blank`. Por defecto se abre en la misma pestaña/ventana de navegador.
- **download**. Permite indicar que lo que queremos es descargar el recurso y no abrirlo.

Ejemplo:

```
<p>Para descargar el enunciado de la práctica puedes pinchar <a  
href="practica1.pdf" download>aquí</a>. Si lo quieres abrir en el  
navegador pincha <a href="practica1.pdf" target="_blank">aquí</a>.</p>
```

Listas

Listas

HTML permite crear tres tipos de listas mediante los siguientes elementos:

- ****. Lista desordenada. Los elementos de la lista aparecen por defecto con el típico círculo negro como marcador (*bullet*).
- ****. Lista ordenada. Los elementos de la lista aparecen enumerados por defecto como 1, 2, ... Se puede utilizar el atributo **type** para cambiar el tipo de numeración utilizada.
- **<dl>**. Lista de definición. Permite ordenar los elementos de la lista como si se tratara de un diccionario: término más definición.

Para los dos primeros tipos de lista, **** y ****, los elementos de la lista se deben añadir utilizando el elemento HTML ****.

Para el tercer tipo de lista, **<dl>**, los elementos se añaden utilizando los elementos HTML **<dt>** para el término y **<dd>** para su definición.

Listas

Los posibles valores para el atributo **type** para la lista ordenada, ``, son los siguientes:

- `<ol type = "1">` - Números 1,2,... Es la opción por defecto.
- `<ol type = "I">` - Números romanos en mayúsculas.
- `<ol type = "i">` - Números romanos en minúsculas.
- `<ol type = "A">` - Letras del abecedario en mayúsculas.
- `<ol type = "a">` - Letras del abecedario en minúsculas.

Con la lista ordenada también se puede utilizar el atributo **start**. Este atributo permite especificar el valor de inicio de los marcadores de los elementos de la lista. Por ejemplo, con `<ol type="a" start="4">`, el primer elemento de la lista tendrá como marcador la letra d.

Los elementos `` de la lista ordenada pueden establecer el número de marcador que le toca mediante el atributo **value**. Los siguientes elementos de la lista verán modificada su numeración y continuarán sus marcadores con los números siguientes al establecido.

Listas

Ejemplo:

```
<ul>
  <li>Elemento 1</li>
  <li>Elemento 2</li>
  <li>Elemento 3</li>
  <li>Elemento 4</li>
</ul>
```

- Elemento 1
- Elemento 2
- Elemento 3
- Elemento 4

```
<ol type="I">
  <li>Elemento 1</li>
  <li>Elemento 2</li>
  <li value="10">Elemento 3</li>
  <li>Elemento 4</li>
</ol>
```

- I. Elemento 1
- II. Elemento 2
- X. Elemento 3
- XI. Elemento 4

```
<dl>
  <dt>Palabra 1</dt>
  <dd>Definición de la palabra 1</dd>
  <dt>Palabra 2</dt>
  <dd>Definición de la palabra 2</dd>
  <dt>Palabra 3</dt>
  <dd>Definición de la palabra 3</dd>
  <dt>Palabra 4</dt>
  <dd>Definición de la palabra 4</dd>
</dl>
```

```
Palabra 1
  Definición de la palabra 1
Palabra 2
  Definición de la palabra 2
Palabra 3
  Definición de la palabra 3
Palabra 4
  Definición de la palabra 4
```

Tablas

Tablas

- HTML proporciona el elemento `<table>` para mostrar datos tabulados, es decir, organizados en filas y columnas.
- El elemento `<table>` no se debe utilizar para maquetar.

Elementos HTML de tabla

`<table>`

Es el elemento principal.

`<caption>`

Es un elemento opcional y permite establecer un título dentro de la tabla.

`<thead>`

Es un elemento opcional que contendrá las filas cabecera de la tabla.

Tablas

Elementos HTML de tabla

`<tbody>`

Es el elemento opcional que contendrá las filas del cuerpo de la tabla, es decir las filas que contienen los datos.

`<tfoot>`

Es un elemento opcional que contendrá las filas del pie de la tabla. Estas filas suelen ser las de resumen de los datos (totales, subtotales, promedio, ...).

`<tr>`

Elemento para definir una fila de la tabla.

`<th>`

Elemento para definir una celda cabecera de la tabla.

`<td>`

Elemento para definir una celda de datos de la tabla.

Tablas

Ejemplo:

```
<table>
  <caption>Notas de la asignatura</caption>
  <thead>
    <tr>
      <th>DNI</th><th>Práctica 1</th><th>Práctica 2</th><th>Práctica 3</th>
    </tr>
  </thead>
  <tbody>
    <tr><td>12345678A</td><td>8.5</td><td>6.5</td><td>8</td></tr>
    <tr><td>23452556S</td><td>7</td><td>6</td><td>7.75</td></tr>
    <tr><td>34904726H</td><td>9.25</td><td>8.5</td><td>9</td></tr>
  </tbody>
  <tfoot>
    <tr><th>Media:</th><td>8.25</td><td>7</td><td>8.25</td></tr>
  </tfoot>
</table>
```

Notas de la asignatura			
DNI	Práctica 1	Práctica 2	Práctica 3
12345678A	8.5	6.5	8
23452556S	7	6	7.75
34904726H	9.25	8.5	9
Media:	8.25	7	8.25

Imágenes

Imágenes

Para incluir imágenes en un documento HTML, existen diferentes formas de hacerlo, aunque los elementos más utilizados son:

- ``
- `<figure>`
- `<picture>`

Elemento ``

Es el elemento más utilizado para insertar una imagen. Como mínimo necesita los atributos `src` y `alt`.

- El atributo `src` sirve para indicar la ubicación (URL) del fichero de imagen a insertar.
- El atributo `alt` se utiliza para especificar un texto alternativo descriptivo de la imagen, cuyo objetivo es servir de reemplazo de la misma cuando ésta no es accesible y el usuario no la puede ver.

Imágenes

Elemento ``

Ejemplo:

```

```

Se pueden utilizar dos atributos más que permiten variar el tamaño (ancho y alto) con el que se va a mostrar la imagen: **width** y **height**.

Ejemplo:

```

```

Los valores de **width** y **height** son numéricos y representan el tamaño en píxeles.

Imágenes

Elemento ``

Además, este elemento admite dos atributos más para la inclusión de imágenes adaptativas en un documento html. Estos dos atributos se deben utilizar siempre juntos y son:

- **srcset**: permite indicar una o más imágenes candidatas (separadas por comas) a utilizar en función de las circunstancias (densidad de píxel del dispositivo, tamaño de la pantalla, etc). Cada imagen irá acompañada de su ancho real expresado en unidades “w” (en lugar de “px”), o de su densidad de pixel expresada en unidades “x”.
- **sizes**: permite especificar el tamaño a utilizar en función de una condición sobre el diseño de la página. En el último tamaño especificado no se pone condición.

Nota: Si el navegador no soporta el atributo srcset, utilizará el atributo src.

Imágenes

Elemento ``

Ejemplo: imagen adaptativa

```

```

Funcionamiento

Primero se evalúa el atributo `sizes`. A continuación se busca la imagen, de entre las indicadas por `srcset`, cuyo tamaño se ajuste mejor al hueco seleccionado por `sizes` en base del tamaño de la ventana.

El código anterior hará lo siguiente:

- para anchos de ventana de hasta 320px, se mostrará `img_pequenya.jpg`, ocupando un ancho máximo de 150px;
- para anchos de ventana por encima de 320px y hasta 640px, se mostrará `img_mediana.jpg`, ocupando un ancho máximo de 300px;
- para el resto de anchos mayores de 640px, se mostrará `img_grande.jpg`, ocupando un ancho máximo de 600px.

El atributo `src` es conveniente ponerlo porque se utiliza como *fallback* por si el navegador no soporta los otros dos atributos.

Imágenes

Elemento `<figure>`

- Se utiliza, no sólo para imágenes, sino también para diagramas, códigos de ejemplo, etc.
- Inserta contenido independiente del flujo principal del documento, aunque relacionado con éste, susceptible de ir en un anexo.
- Si en el flujo principal del documento se trata como una única unidad de contenido y autosuficiente, pudiéndose eliminarse sin afectar al significado de éste, debería llevar un título o una leyenda que se indicará mediante el elemento `<figcaption>`, aunque el uso de `<figcaption>` es optativo.

Ejemplo:

```
<figure>  
    
  <figcaption>Foto 12: Llegada al punto de destino.</figcaption>  
</figure>
```

Imágenes

Elemento `<picture>`

- El elemento `<picture>` permite especificar diferentes fuentes, mediante el elemento `<source>`, para el elemento `` que contiene.
- Si se especifican elementos `<source>`, éstos deben ir ordenados y antes del elemento ``, ya que se aplica el primer elemento `<source>` cuya condición asociada se cumpla. En caso de que no se cumpla ninguna condición de ningún `source`, se mostraría la imagen del elemento ``.
- Las condiciones asociadas a cada elemento `<source>` se especifican mediante sus atributos.

Imágenes

Elemento `<picture>`

Atributos del elemento `<source>`

- **type**: permite especificar el formato de la imagen. El navegador utilizará la primera imagen cuyo formato soporte. Valores posibles son `image/svg+xml`, `image/webp`, `image/jpeg`, `image/png`, `image/gif`.
- **srcset**: permite indicar una o más imágenes candidatas (separadas por comas) a utilizar en función de las características de la imagen (densidad de píxel del dispositivo, tamaño de la imagen, etc). Para cada imagen se especifica la ubicación mediante su URL, el ancho de la misma mediante un número entero positivo seguido por “w” (si no se pone, se supone infinito) y un descriptor de la densidad de píxel (número de píxeles por pulgada), que es un número decimal seguido por “x” (si no se pone, se supone un valor 1x).
- **media**: permite especificar reglas de tipo *media query* para utilizar o no el elemento `<source>`.
- **sizes**: permite especificar el tamaño a utilizar para cada imagen indicada en el atributo `srcset` en función del diseño de la página.

Imágenes

Elemento `<picture>`

Ejemplo:

```
<picture>
  <source media="(min-width: 45em)" srcset="img_grande.jpg">
  <source media="(min-width: 32em)" srcset="img_mediana.jpg">
  
</picture>
```

El código anterior:

- para anchos de ventana mayores de 45em, se mostrará `img_grande.jpg`;
- para anchos de ventana mayores de 32em y menores de 45em, se mostrará `img_mediana.jpg`;
- para el resto de anchos (menores de 32em), se mostrará `img_pequenya.jpg`.

Imágenes

Elemento `<picture>`

Ejemplo:

- utilizando *device-pixel-ratio*:

```
<picture>
  <source srcset="imagen_1_5x.jpg 1.5x, imagen_2x.jpg 2x">
  
</picture>
```

- utilizando el atributo *type*:

```
<picture>
  <source srcset="imagen.webp" type="imagen/webp">
  <source srcset="imagen.png" type="imagen/png">
  <source srcset="imagen.svg" type="imagen/svg+xml">
  
</picture>
```


Formularios

Formularios

- Se construyen mediante el elemento HTML `<form>`.
- Admite **numerosos controles de formulario**: cajas de texto, botones, *checkboxes*, campos para valores numéricos, etc.
- Mediante **atributos propios** se puede filtrar/validar la entrada de datos en los distintos campos del formulario.
- Permiten **enviar a un servidor** la información recogida en los controles de formulario que lo componen. Para ello, dichos controles deben tener un atributo **name** cuyo valor será el nombre del campo asociado al valor recogido en el control.

Formularios

Atributos del elemento <form>

Además de los atributos globales, admite otros atributos propios. Algunos de ellos se indican a continuación:

- **action**. URL del fichero/programa al que se enviará la información recogida para que sea tratada.
- **autocomplete**. Admite valores **on** y **off**. Permite indicar si se permite que se asignen valores a los controles del formulario automáticamente basados en valores anteriores de esos mismos controles. Esta configuración para todos los controles del formulario se puede modificar para un control concreto del formulario utilizando en él de nuevo el atributo **autocomplete**.
- **method**. Es el método que el navegador utilizará para enviar la información al programa indicado en el atributo **action**. Posibles valores GET y POST. Cuando se utiliza GET los datos del formulario se añaden a la URL del programa que tratará los datos mediante un carácter “?” y las parejas nombre=valor de cada control del formulario que tenga atributo **name**. Cuando se utiliza POST los datos son enviados al servidor en el cuerpo de la petición.

Formularios

Atributos del elemento <form>

- **enctype**. Se utiliza para indicar el tipo del contenido del formulario cuando el valor de `method` es `POST`. El valor `application/x-www-form-urlencoded` es el valor por defecto. Si hay un control de tipo fichero en el formulario, se utiliza el valor `multipart/form-data`.
- **target**. Se utiliza de la misma forma que en el elemento `<a>`. Por defecto, y si no utilizamos este atributo o su valor es `_self`, el formulario abrirá la respuesta del servidor en la misma ventana del navegador. El valor `_blank` hará que lo abra en una nueva ventana/pestaña.
- **novalidate**. Se utiliza para inhabilitar la validación del formulario.

Ejemplo:

```
<form method="POST" action="fichero.php" autocomplete="on">
  <label>Usuario: <input type="text" name="login"></label><br>
  <label>Contraseña: <input type="password" name="pwd"></label><br>
  <input type="reset"><input type="submit">
</form>
```

Formularios

Controles de formulario

Elemento `<input>`

La mayoría de controles de un formulario se representan mediante el elemento `<input>`. Su atributo **type** permite especificar el tipo de control deseado. Los valores para el atributo **type** son los siguientes:

- **button**. Representa un botón cuyo comportamiento por defecto es enviar el formulario.
- **checkbox**. Representa una casilla de selección cuyo valor se indica en el atributo **value**. Si se quiere activar por defecto, hay que utilizar el atributo booleano **checked**.
- **color**. Permite seleccionar un color de la paleta de colores.
- **date**. Permite introducir una fecha sin hora.
- **datetime-local**. Permite introducir una fecha con hora.
- **email**. Permite introducir una dirección de correo electrónico. El control valida automáticamente que la dirección de correo introducida tenga un formato correcto.

Formularios

Controles de formulario

Elemento `<input>`. Valores del atributo **type**

- **file**. Permite al usuario seleccionar un solo archivo. Si se quiere permitir seleccionar más de un archivo hay que utilizar el atributo **multiple**. Se puede filtrar qué archivos se permiten mediante el atributo **accept**.
- **hidden**. Permite añadir información al formulario para enviar al servidor, pero que no es visible para el usuario.
- **image**. Permite utilizar una imagen como un botón de envío de formulario. Los atributos que admite son los mismos que para un elemento ``.
- **month**. Permite introducir un mes y año.
- **number**. Permite introducir números con o sin decimales. Mediante los atributos **min**, **max** y **step** se puede establecer el rango de valores permitidos, así como el incremento (**step**) permitido entre un valor válido y otro. Se puede utilizar el atributo **value** para establecer un valor por defecto.
- **password**. Se utiliza para introducir un valor que no se pueda leer. Para ello se muestra un carácter no legible por cada carácter introducido.

Formularios

Controles de formulario

Elemento `<input>`. Valores del atributo `type`

- **radio**. Permite crear un grupo de opciones seleccionables mutuamente excluyentes. Para ello hay que utilizar el atributo `name` con el mismo valor en todos los controles de tipo radio del grupo de opciones. Mediante el atributo `value` de cada control se asigna el correspondiente valor y se puede seleccionar una opción por defecto mediante el atributo `checked`.
- **range**. Gráficamente se representa como un slider que permite introducir valores numéricos. Admite los mismos atributos que el `type="number"`.
- **reset**. Limpia el formulario restaurando los controles a sus valores por defecto.
- **search**. Permite introducir textos de búsqueda.
- **submit**. Botón que envía el formulario al servidor.
- **tel**. Permite introducir un número de teléfono.
- **text**. Permite introducir texto simple en una sola línea. Los saltos de línea no los tiene en cuenta.

Formularios

Controles de formulario

Elemento `<input>`. Valores del atributo **type**

- **time**. Permite introducir un dato de tiempo en forma de hora y minutos.
- **url**. Permite introducir una url. El control valida automáticamente que la url introducida tenga un formato correcto.
- **week**. Permite introducir una semana del año El dato guardado es el número de semana y el año.

Elemento `<textarea>`

Proporciona un caja de texto multilínea.

Elemento `<label>`

Permite etiquetar un control de formulario. El control puede ir dentro de la etiqueta `<label>` o bien, enlazado mediante su atributo **for** con el valor del atributo **id** del control que etiqueta.

Formularios

Controles de formulario

Elemento `<select>`

Se representa como un cuadro de lista desplegable y permite seleccionar entre un conjunto de opciones. Las distintas opciones se especifican con el elemento `<option>` que, a su vez, pueden ser agrupadas mediante el elemento `<optgroup>` y darle un título mediante su atributo **label**. Por defecto sólo permite seleccionar una de las opciones. Si se quisiera permitir la selección de múltiples opciones habría que añadirle el atributo **multiple**. Mediante el atributo **size** se puede especificar cuántas opciones hacer visibles y convertir el control en una lista vertical con *scroll*.

Elemento `<datalist>`

Contiene un conjunto de elementos `<option>`, que representan opciones predefinidas para otros controles de formulario, por ejemplo `<input>`. Este elemento no es visible por parte del usuario. Para que esté enlazado con un elemento `<input>`, éste debe tener el atributo **list** cuyo valor sea el valor del atributo **id** del control `<datalist>`.

Formularios

Controles de formulario

Elemento `<button>`

Representa un botón cuya etiqueta es su contenido. Mediante su atributo **type** se puede establecer su comportamiento: `submit` (valor por defecto), para enviar el formulario; `reset`, para limpiar los controles del formulario; y `button`, no hace nada por defecto, su comportamiento se indicaría mediante JavaScript.

Elemento `<fieldset>`

Permite agrupar varios controles dentro de un formulario. Se suele utilizar para agrupar distintos controles que recogen información desglosada como puede ser la de un nombre completo (nombre, apellido 1 y apellido 2), una dirección (calle, población, provincia, código postal), etc.

Elemento `<legend>`

Se utiliza dentro de `<fieldset>` y permite especificar un título para el grupo de controles contenidos en el elemento `<fieldset>`.

Formularios

Controles de formulario

Atributos comunes

Además de los atributos particulares a cada control, existen otros atributos comunes a los controles de formulario con los que el usuario puede interactuar:

- **autocomplete**. Funciona igual que para el elemento `<form>`.
- **autofocus**. Obtiene el foco automáticamente al cargar la página.
- **readonly**. Especifica que el control es sólo de lectura.
- **disabled**. Al añadirlo a un control, éste se deshabilita.
- **title**. Se utiliza para personalizar el mensaje de error de validación del control cuando ésta falle.
- **pattern**. Permite indicar una expresión regular que debe validar el valor introducido y que se comprueba al enviar el formulario.
- **placeholder**. Permite mostrar al usuario un pequeño mensaje informativo dentro del propio control.
- **required**. Si se indica, el control no puede quedar vacío.
- **form**. Permite especificar el valor del atributo `id` del formulario al que pertenece el control cuando éste está fuera del mismo.

Formularios

Controles de formulario que no son de entrada

Se trata de controles que no recogen información del usuario sino que muestran información complementaria a la interacción del usuario con el formulario.

Elemento `<output>`

Permite mostrar el resultado de un cálculo realizado, o el resultado de una acción de usuario.

Elemento `<progress>`

Se representa como una barra de progreso y se utiliza para mostrar la evolución de un valor que cambia con el tiempo hasta llegar a un valor máximo. Admite el atributo `max` para indicar el valor máximo a alcanzar (si no se indica, el valor por defecto es 1) y `value`, para indicar el valor de progreso a representar, que es un valor decimal entre 0 y `max`.

Formularios

Controles de formulario que no son de entrada

Elemento `<meter>`

Se representa como una barra en la que se muestra un valor fijo entre un máximo y un mínimo. El rango de posibles valores se divide en tres zonas que se identifican como la zona buena, la zona media, y la zona mala. Los atributos que admite son los siguientes:

- **min** y **max**. Permiten especificar los valores máximo y mínimo.
- **value**. Permite expresar el valor actual que se quiere mostrar.
- **low** y **high**. Permiten especificar los dos puntos de corte que dividen al rango total en tres partes.
- **optimum**. Es el atributo que indicará cuál es la zona buena y cuál la zona mala dependiendo del valor que se le asigne:
 - Si su valor está en la parte inferior, es decir **optimum** < **low**, entonces la zona inferior es considerada la buena y la superior la mala.
 - Si **high** < **optimum**, entonces la zona superior es la buena.
 - Si **low** < **optimum** < **high**, entonces tanto la parte inferior como la superior se consideran como un rango medio y la parte central se considera como la buena.