



Universidad de Jaén

Escuela Politécnica Superior de Linares

Generación de contenidos con HTML5

Diseño básico de páginas web con HTML5

Juan Carlos Cuevas Martínez

30/01/2025

Servicios y aplicaciones telemáticas



Seminario. Generación de contenidos con HTML5

Diseño básico de paginas web con HTML5

Asignatura: Servicios y Aplicaciones Telemáticas

Grado en Ingeniería Telemática

Grado en Ingeniería de Tecnologías de Telecomunicación

Autor:

Juan Carlos Cuevas Martínez



Competencias

Código	Denominación de la competencia
C.7	Conocimiento y utilización de los fundamentos de la programación en redes, sistemas y servicios de telecomunicación.
CB.4	Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado.
CG.3	Conocimiento de materias básicas y tecnologías, que le capacite para el aprendizaje de nuevos métodos y tecnologías, así como que le dote de una gran versatilidad para adaptarse a nuevas situaciones.
CG.9	Capacidad de trabajar en un grupo multidisciplinar y en un entorno multilingüe y de comunicar, tanto por escrito como de forma oral, conocimientos, procedimientos, resultados e ideas relacionadas con las telecomunicaciones y la electrónica.

Resultados del aprendizaje

Código	Descripción
5	Fomentar los trabajos grupales, así como la transmisión de procedimientos, resultados e ideas en el ámbito de las telecomunicaciones.
7	Se adquiere la noción de organización en capas para resolver un problema de comunicación entre dos partes, presentando los principales paradigmas de la arquitectura de red.
10	El alumno distinguirá y relacionará los conceptos de servicio y aplicación en el ámbito de las comunicaciones; además asimilará los detalles vinculados con los servicios de telecomunicación más populares y las aplicaciones y protocolos que los soportan. Describir los principales servicios y aplicaciones de Internet.
20	Adquirir facilidad para el manejo de especificaciones, reglamentos y normas de obligado cumplimiento.

Resultados del aprendizaje

Código	Descripción
21	Conocer materias básicas y tecnológicas que capaciten para el aprendizaje de nuevos métodos y tecnologías.
22	Dotar de una gran versatilidad para adaptarse a nuevas situaciones.
23	Analizar y valorar el impacto social y medioambiental de las soluciones técnicas.
24	Trabajar en un grupo multidisciplinar y en un entorno multilingüe.
25	Comunicar, tanto por escrito como de forma oral, conocimientos, procedimientos, resultados e ideas relacionadas con las telecomunicaciones y la electrónica.

Contenido

1. Introducción
2. Estructura del lenguaje HTML
3. Elementos comunes de HTML
 1. Párrafos.
 2. Cabeceras.
 3. Listas numeradas.
 4. Listas no numeradas.
 5. Lista de definiciones.
 6. Enfatizar texto.
 7. Caracteres especiales.
 8. Enlaces.
 9. Imágenes.
 10. Tablas.
 11. Efectos especiales.
4. Formularios.
5. Cuadros de diálogo.
6. Multimedia.



Bibliografía

Bibliografía básica

- Cravens, J., “**HTML5 hacks**”. O'Reilly Media
- Gauchat, J.D., “**El gran libro de HTML5, CSS3 y JavaScript**”. Marcombo

Bibliografía

Bibliografía complementaria

- w3schools.com, “**HTML Tutorial**”. URL:
<https://www.w3schools.com/html/default.asp>
- Mozilla Developer Network, “**HTML**”. URL:
<https://developer.mozilla.org/es/docs/Web/HTML>



1. Introducción

HTML (*Hypertext Markup Language*)

- Es un lenguaje de marcas o etiquetas (*tags*) cuyo fin es almacenar información de una manera estructurada.
- Es el estándar para las páginas web.
- La versión actual es la 5, conocida como HTML5.
- Su especificación completa, en formato de “estándar vivo” se puede encontrar en **<https://html.spec.whatwg.org>**.
 - El estándar actual se mantiene fruto del acuerdo entre WHATWG y W3C, firmado en 2019.



Versión	Año
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

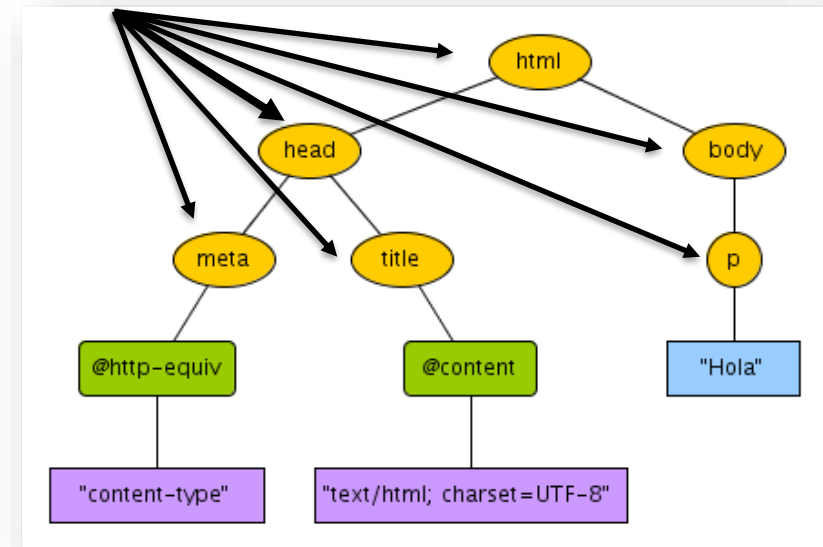
1. Introducción

HTML (*Hypertext Markup Language*)

Estructura

- La estructura de un documento es un árbol de elementos HTML.
- Cada **elemento** de ese árbol está definido por etiquetas.
- Las etiquetas pueden ser de apertura o cierre (<p>...</p>) o etiquetas aisladas (
).
- Existe una jerarquía de elementos
 - Elemento raíz: <html>
 - Elementos hijos : <head> (indicaciones) y <body> (contenidos)
- Cuando el navegador procesa un documento HTML crea una representación virtual en árbol conocida como DOM¹.

Elementos



1: DOM son las siglas de *Document Object Model*.

1. Introducción

HTML (*Hypertext Markup Language*)

Estructura

Ejemplo

<!DOCTYPE html>

TIPO DE DOCUMENTO

<html>

<head>

<title>Título</title>

</head>

CABECERA

<body>

<p>Hola mundo</p>

</body>

CUERPO

</html>

2. Estructura del lenguaje HTML

```
<!DOCTYPE html>
```

- La declaración `<!DOCTYPE>`:
 - Debe ser la primera línea en un documento HTML, justo antes de la etiqueta `<html>`.
 - No es una etiqueta HTML, es una instrucción para el navegador web para saber en qué versión de HTML está escrito el documento.
 - En HTML 4.01, la declaración `<!DOCTYPE>` refería a una DTD dado que HTML 4.01 estaba basado en SGML.
 - La DTD especifica la reglas del lenguaje de marcas para que el navegador pudiera representar la información correctamente.
- HTML5 no está basado en SGML y por lo tanto no hace referencia a una DTD.
- **Nota:** siempre se debe añadir la declaración `<!DOCTYPE>` en los documentos HTML para que el navegador sepa qué tipo de documento es.
 - Más información en: <https://developer.mozilla.org/es/docs/Glossary/Doctype>



Sí



Sí



Sí



Sí

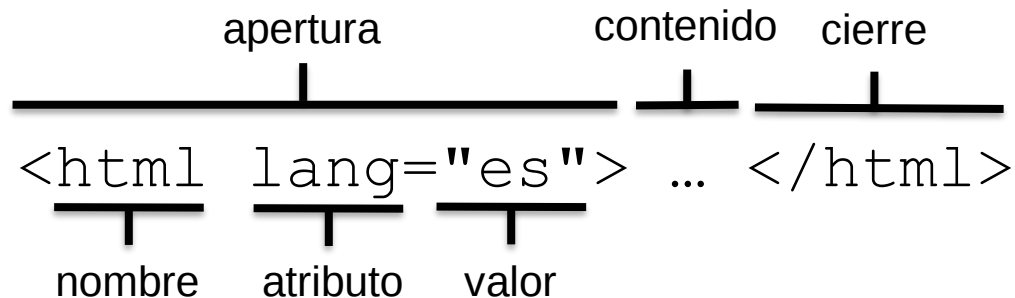


Sí

2. Estructura del lenguaje HTML

Etiquetas

- Las etiquetas deben seguir un orden jerárquico.
- Se cierran en el orden inverso al que se abrieron.
- Además pueden incorporar atributos.



2. Estructura del lenguaje HTML

Etiquetas

Atributos

- Los atributos añaden información adicional a los elementos HTML.
- En muchos casos son imprescindibles para que el elemento HTML tenga alguna función, como el atributo *href* en el elemento `<a>`.
- Los atributos disponibles para los elementos de HTML5 varían según estos.
 - Algunos atributos han dejado de ser válidos con respecto a HTML4, tales como *align*, *color* o *border*.

2. Estructura del lenguaje HTML

Etiquetas

Atributos

- Los atributos en HTML5 pueden escribirse en minúsculas o mayúsculas, aunque se recomienda escribirlos en minúsculas.
- HTML5 no obliga a delimitar el valor de un atributo entre comillas, pero el no hacerlo puede llevar a errores de interpretación.
 - Es obligatorio entrecomillar el valor en aquellos atributos que contengan espacios en blanco o los caracteres " ' ` = < o >.
- Si se necesita escribir unas comillas dentro del valor de un atributo, se pueden alternar las comillas dobles con las simples:

```
<button onclick="myFunction('10')">Click me</button>
```

2. Estructura del lenguaje HTML

Etiquetas

Atributos booleanos

- En los atributos booleanos en HTML son un caso algo particular, si el atributo aparece, se toma como valor verdadero (true), aunque se le asigne por ejemplo una cadena vacía, y si no aparece, como falso (“false”).
 - Realmente los valores “true” o “false” no se permiten en atributos booleanos.

```
<input  
  type="checkbox"  
  name="blue"  
  checked="false"  
  value="blue" />
```

Aunque **checked** se ponga a valor “false” seguirá siendo “true” por el mero hecho de estar presente.

2. Estructura del lenguaje HTML

Etiquetas

Atributos globales id y class

- Existen atributos globales que pueden ser empleados en cualquier elemento HTML, estos son, por ejemplo, el atributo *id* y *class*.
 - Atributo *id*: permite asignar un **identificador único** a un elemento.
 - Atributo *class*: está pensado para asignar el mismo valor a grupos de elementos.
 - El valor de este atributo puede incluir varias clases separándolas por un espacio: `<p class="cuerpo resaltado">...</p>`
- Los valores asignados a estos atributos son arbitrarios.
- El valor asignado a ambos es empleado por otros elementos HTML, además de por reglas CSS y código JavaScript.

2. Estructura del lenguaje HTML

Etiqueta HTML <html>

- Su función es indicar al navegador que éste es un documento HTML.
- Es el elemento raíz de todo documento HTML.
- Todos los elementos HTML están contenidos dentro de la marca <html>, salvo la <!DOCTYPE>.
- Se recomienda declarar el lenguaje primario de la página web dentro de la marca html (la lista de abreviaturas se especifica en la ISO 639-1) con el atributo lang:

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

```
...
```

```
</html>
```



Sí



Sí



Sí



Sí



Sí

2. Estructura del lenguaje HTML

Etiqueta HTML <html>

Ejemplo

```
<!DOCTYPE HTML>
<html lang="es">
  <head>
    <title>Título del documento</title>
  </head>
  <body>
    El contenido del documento...
  </body>
</html>
```

2. Estructura del lenguaje HTML

Etiqueta HTML <html>

Omitir las etiquetas <html> <head> y <body>

- El estándar HTML5 permite que se omitan las marcas <html>, <body> y <head>.
 - Omitir <html> y <body> no se recomienda dado que puede ocasionar problemas con el procesado DOM o XML y errores en navegadores antiguos (IE9).
 - Últimamente sí es más común ver páginas que omiten la marca <head>.

```
<!DOCTYPE html>
<title>Page Title</title>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>
```

Ejemplo válido en HTML5 de omisión de las marcas <html>, <body> y <head>

2. Estructura del lenguaje HTML

Comentarios etiqueta <!--...-->

- Un comentario es contenido que no representará el navegador pero que está incluido en el fichero de la página web.
 - Su función es hacer más comprensible el código de la página.
 - No hay diferencia entre HTML 4.01 y HTML5.

```
<!-- Esto es un comentario -->  
<p>Esto es un párrafo.</p>  
<!-- otro  
comentario -->
```



Sí



Sí



Sí



Sí



Sí

2. Estructura del lenguaje HTML

Entidades HTML

- En HTML existen caracteres reservados que, si aparecieran en el texto de una página web, podrían ocasionar problemas en su interpretación.
 - Los más evidentes son los símbolos ‘<’ y ‘>’
- La manera de representar estos caracteres es a través de **entidades** HTML.
 - Comienzan con el carácter ‘&’ o ‘&#’ y terminan con ‘;’ y su formato es el siguiente:
 - `&entity_name;`
 - `&#entity_number;`
 - Ejemplo: ‘<’ sería `<`; del inglés “*less than*” y ‘>’ `>`; del inglés “*greater than*”
 - Las entidades con nombre (*entity_name*) puede que no estén disponibles en todos los navegadores pero las numéricas siempre estarán (*entity_number*).

2. Estructura del lenguaje HTML

Entidades HTML

Espacio sin ruptura de línea (Non-breaking Space)

- El espacio sin ruptura de línea permite separar caracteres sin que el navegador pueda fraccionar la línea por dicho espacio, o eliminar varios consecutivos.
 - Dos palabras separadas por este espacio se tratan como una sola.
 - Se emplea cuando la separación de estas palabras podría llevar a errores de interpretación.
- Corresponde con la entidad: ** **

Ejemplo.

Para conseguir “10 km/h” y evitar que se separe el código sería: 10 km/h

El guion sin ruptura (‑) consigue el mismo efecto pero con el carácter ‘-’.

2. Estructura del lenguaje HTML

Entidades HTML

Entidades HTML útiles

Resultado	Descripción	Nombre de la entidad	Número de la entidad
	non-breaking space	 	
<	less than	<	<
>	greater than	>	>
&	ampersand	&	&
"	double quotation mark	"	"
'	single quotation mark (apostrophe)	'	'
€	euro	€	€
©	copyright	©	©
®	registered trademark	®	®

2. Estructura del lenguaje HTML

Entidades HTML

Símbolos HTML

- Son entidades HTML que permiten representar gran cantidad de símbolos matemáticos y de moneda (entre otros) que no están disponibles directamente en los teclados.
- Si no tienen un nombre de entidad, tendrán disponible un número de entidad o una referencia en decimal o hexadecimal.

Carácter	Número	Entidad	Descripción
∀	∀	∀	FOR ALL
∂	∂	∂	PARTIAL DIFFERENTIAL
∏	∏	∏	N-ARY PRODUCT
Σ	∑	∑	N-ARY SUMMATION
Α	Α	Α	GREEK CAPITAL LETTER ALPHA
Β	Β	Β	GREEK CAPITAL LETTER BETA
μ	μ	μ	GREEK SMALL LETTER MU
€	€	€	EURO SIGN

Acceso a las listas completas de símbolos:
<https://html.spec.whatwg.org/multipage/named-characters.html#named-character-references>

2. Estructura del lenguaje HTML

Cabecera – etiqueta <head>

- La cabecera se delimita por la etiqueta <head>
 - Contiene etiquetas que informan al navegador o que sirven a aplicaciones de terceros, pero su contenido no se muestra (salvo el título en la ventana o pestaña de la página web).
 - Elementos típicos:
 - <title>: Define el título del documento (deber ser único).
 - <base>: Define la dirección de base (*href*) o el destino (*target*) por defecto para todos los enlaces de una página. Solamente puede haber un elemento <base>.
 - <link>: define la relación entre el documento y una fuente externa.
 - <meta>: define los metadatos de un documento HTML.
 - <script>: define un script del lado de cliente.
 - <style>: define la información de estilo de un documento. Pueden existir varios elementos <style>.



Sí



Sí



Sí



Sí



Sí

2. Estructura del lenguaje HTML

Cabecera – etiqueta <head>

```
<head>
  <title>Ejemplo HTML</title>
  <meta charset="UTF-8">
  <base href="https://www.ujaen.es/" target="_blank">
  <link rel="stylesheet" type="text/css" href="theme.css">
  <style>
    h1 {color:red;}
  </style>
  <script>
    function myFunction {
      document.getElementById("demo").innerHTML = "Hola";
    }
  </script>
</head>
...
```

2. Estructura del lenguaje HTML

Cabecera – etiqueta <head>

Conjunto de caracteres

- La codificación por defecto para HTML5 es UTF-8.
 - Esto no ha sido siempre así, en sus inicios fue ASCII, en HTML 2.0 hasta 4.01 fue ISO-8859-1.
 - El uso de UTF-8 (en HTML5 y XML) soluciona muchos problemas de representación que se daban en la web.
 - Se puede definir juegos de caracteres diferentes a UTF-8 en la etiqueta meta.

`<meta charset="ISO-8859-1">`

- El consorcio Unicode desarrolló UTF-8 y UTF-16 para solventar las limitaciones de la ISO-8859 en cuanto al soporte a un entorno multilingüe.
- El estándar Unicode cubre casi todos los caracteres, puntuaciones y símbolos del mundo.
- Todos los procesadores HTML5 y XML soportan **UTF-8, UTF-16, Windows-1252, and ISO-8859.**

2. Estructura del lenguaje HTML

Cabecera – etiqueta <head>

Conjunto de caracteres – Unicode y UTF-8

- Unicode es el conjunto de caracteres (caracteres a número) mientras que UTF-8 es la codificación (números a binario).

UNICODE: A = 65 (decimal)

UTF-8: A = 0100 0001 (binario)

- UTF-8 tiene códigos de uno a cuatro bytes para representar los caracteres UNICODE.
- Los primeros 127 símbolos son los mismos que ASCII (UTF-8 C0 Controls and Basic Latin).



2. Estructura del lenguaje HTML

Cabecera – etiqueta <head>

Conjunto de caracteres – Tabla Unicode, UTF-16 y UTF-8 (fuente Wikipedia)

Rango de puntos UNICODE	Valor escalar	UTF-16	UTF-8	Notas
000000-00007F	00000000 0xxxxxxx	00000000 0xxxxxxx	0xxxxxxx	Rango equivalente a US-ASCII. Símbolos de un único byte donde el bit más significativo es 0
000080-0007FF	00000yyy yyxxxxxx	00000yyy yyxxxxxx	110yyyyy 10xxxxxx	Símbolos de dos bytes. El primer byte comienza con 110, el segundo byte comienza con 10
000800-00FFFF	zzzzyyyy yyxxxxxx	zzzzyyyy yyxxxxxx	1110zzzz 10yyyyyy 10xxxxxx	Símbolos de tres bytes. El primer byte comienza con 1110, los bytes siguientes comienzan con 10
010000-10FFFF	000uuuuu zzzzyyyy yyxxxxxx	110110ww wwzzzzyy 110111yy yyxxxxxx (www = uuuu - 1)	11110uuu 10uuzzzz 0yyyyy 10xxxxxx	Símbolos de cuatro bytes. El primer byte comienza con 11110, los bytes siguientes comienzan con 10

2. Estructura del lenguaje HTML

Cabecera – etiqueta <head>

Configurar la ventana de visualización (viewport) con la etiqueta <meta>

- Con HTML5 se ha introducido un método para que el diseñador web pueda configurar la ventana de visualización (*viewport*) a través de la etiqueta <meta>.
 - La *viewport* es el área visible de una página web.
 - Varía entre diferentes dispositivos, como móviles, tabletas o pantallas de PC.
- Para establecer la ventana de visualización se debe incluir una etiqueta <meta> en todas las páginas webs tal que así:

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

- La etiqueta *viewport* en el elemento <meta> da al navegador las instrucciones para controlar las dimensiones de la página y su escalado.
 - **width=device-width** establece la anchura de la ventana para que se adapte a la anchura del dispositivo que la está mostrando (la cual puede variar considerablemente).
 - **initial-scale=1.0** establece el nivel de zoom inicial cuando la página se carga por primera vez en el navegador. Su uso fundamental es para ajustar el tamaño de las páginas en navegadores en terminales móviles. No tiene efecto en navegadores de escritorio.

2. Estructura del lenguaje HTML

Cabecera – etiqueta <head>

Otras capacidades de la etiqueta <meta>

- La etiqueta <meta> a través del atributo *http-equiv* permite comportamientos especiales:
 - *content-type*: permite especificar la codificación del documento:
`<meta http-equiv="content-type" content="text/html; charset=UTF-8">`
 - *default-style*: especifica la hoja de estilos preferida para emplear.
`<meta http-equiv="default-style" content="título de la hoja de estilos">`
 - *refresh*: hace que la página se recargue automáticamente
 - **Ejemplo: ir a una URL en 2 segundos:**
`<meta http-equiv="refresh" content="2;URL=http://www.ujaen.es"/>`
 - **Ejemplo: recargar página actual periódicamente (15 segundos):**
`<meta http-equiv="refresh" content="15">`

2. Estructura del lenguaje HTML

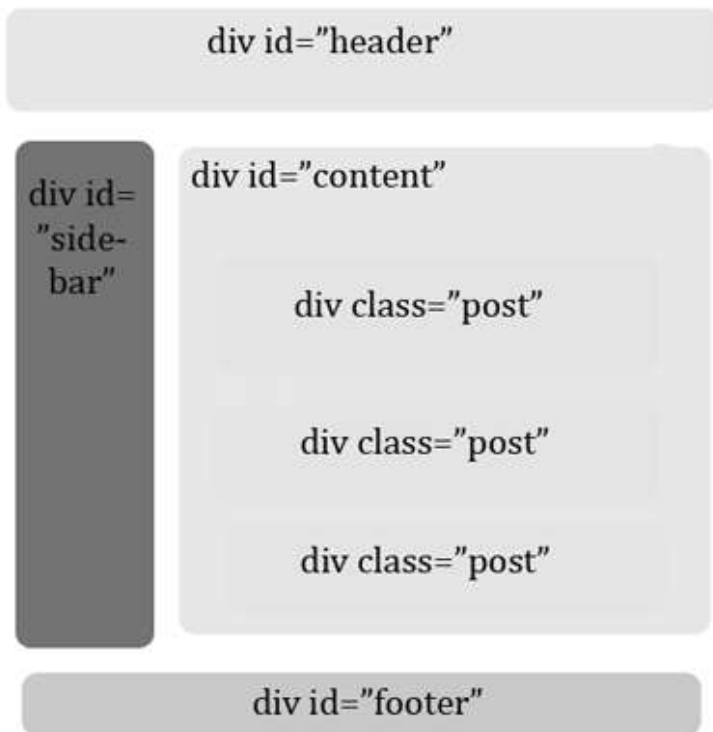
El cuerpo de una página web <body>

- El cuerpo es donde se encuentra el contenido del documento.
- Dentro del cuerpo pueden aparecer un número indeterminado de marcas o elementos diferentes.
- En la actualidad la estructura del cuerpo de las páginas web es muy diversa, pero en general suelen contar con una serie de elementos comunes, aunque su aspecto y colocación en la página puede variar enormemente.
- En HTML5 se han introducido **nuevos elementos** para ayudar al diseño e interpretación de las páginas web.

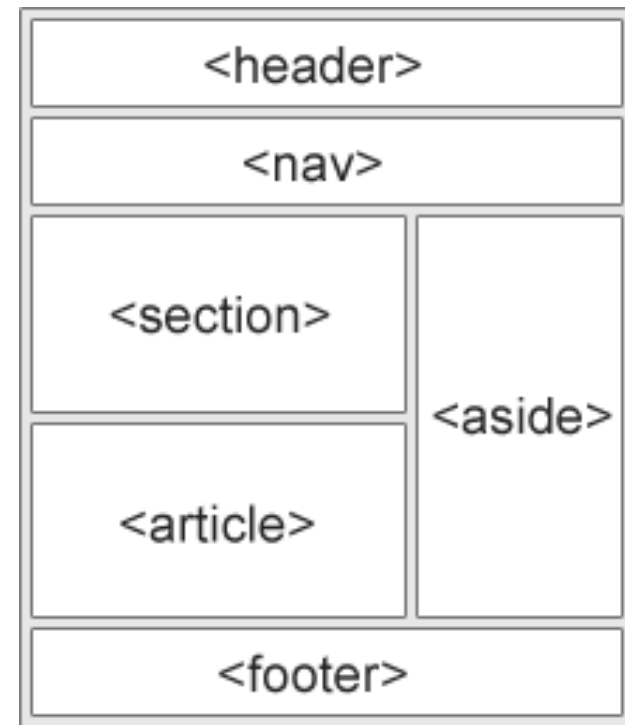
2. Estructura del lenguaje HTML

El cuerpo de una página web <body>

Empleando <div>



Empleando elementos HTML5



2. Estructura del lenguaje HTML

El cuerpo de una página web <body>

Nuevos elementos en HTML5

- **Elementos semánticos:** describen claramente su contenido y función, tanto para un navegador, como para un desarrollador (<form>, <table>,...)
 - Nuevos elementos semánticos en HTML5: <article>, <aside>, <details>, <figcaption>, <figure>, <footer>, <header>, <main>, <mark>, <nav>, <section>, <summary>, <time>.
- **Elementos no semánticos:** no informan acerca de lo que significa o representa su contenido (<div>,).
- Sin embargo, antes de HTML5 (e incluso con HTML5), se ha empleado la etiqueta <div> para crear el diseño de la página y es muy habitual aún.
 - La marca <div> no tiene atributos obligatorios pero es muy común que incluya *style*, *class* e *id*.
 - Ejemplos:

```
<div id="nav">...</div>
<div class="header">...</div>
<div id="footer">...    </div>
```

2. Estructura del lenguaje HTML

El cuerpo de una página web <body>

Nuevos elementos en HTML5

- A continuación se comentan las funciones más importantes de algunos de los elementos semánticos de HTML5:
 - <main>: define una división que debe albergar el contenido dominante de la página (tema central de la página).
 - <header>: se emplea para definir cabeceras del cuerpo de la página o de secciones dentro de ella.
 - <nav>: esta división se emplea para contener una ayuda a la navegación, el menú principal de la página o agrupaciones de enlaces necesarias para navegar en el sitio web.
 - <section>: define una sección genérica, frecuentemente se usa para separar contenido temático, crear columnas o bloques que ayudan a organizar el contenido de la página.
 - <article>: representa un artículo independiente, que tiene sentido por sí solo, como un mensaje de foro, un comentario, una entrada de blog, etc.
 - <aside>: se emplea para mostrar información, no necesariamente relacionada con el contenido principal, pero que no es parte del mismo, como referencias a artículos que apuntan a publicaciones anteriores.
 - <footer>: este elemento define el pie de la página o de una sección.

2. Estructura del lenguaje HTML

El cuerpo de una página web <body>

Colocación de elementos: en bloque y elementos en línea

- En HTML, por defecto, solamente hay dos tipos de colocación de elementos:
 - **Elementos en bloque (*block*):** este tipo de elementos siempre comienzan en línea nueva y ocupan todo el ancho disponible.
 - Ejemplos: <address>, <article>, <aside>, <blockquote>, <canvas>, <dd>, <div>, <dl>, <dt>, <fieldset>, <figcaption>, <figure>, <footer>, <form>, <h1>-<h6>, <header>, <hr>, , <main>, <nav>, <noscript>, , <p>, <pre>, <section>, <table>, <tfoot>, , <video>.
 - **Elementos en línea (*inline*):** este tipo de elementos no comienzan en nueva línea y solamente ocupan el espacio necesario para mostrar la información que contienen.
 - Ejemplos: <a>, <abbr>, <acronym>, , <bdo>, <big>,
, <button>, <cite>, <code>, <dfn>, , <i>, , <input>, <kbd>, <label>, <map>, <object>, <output>, <q>, <samp>, <script>, <select>, <small>, , , <sub>, <sup>, <textarea>, <time>, <var>.
- La colocación de los elementos puede ser cambiada por estilos (*display*).

2. Estructura del lenguaje HTML

El cuerpo de una página web <body>

Elementos HTML4 eliminados en HTML5

Elemento eliminado	Emplear
<acronym>	<abbr>
<applet>	<object>
<basefont>	CSS
<big>	CSS
<center>	CSS
<dir>	

Elemento eliminado	Emplear
	CSS
<frame>	
<frameset>	
<noframes>	
<strike>	CSS, <s>, o
<tt>	CSS

La lista completa de elementos y características obsoletas se puede encontrar aquí:

<https://html.spec.whatwg.org/#toc-obsolete>

3. Elementos comunes de HTML

- Dentro de cuerpo de una página web se encuentran todos los elementos que estructuran la información que se quiere transmitir.
- Se estudiarán:
 - Encabezados.
 - Párrafos.
 - Listas.
 - Enfatizar texto.
 - Enlaces.
 - Imágenes.
 - Tablas.
 - Formularios.
 - Cuadros de diálogo.
 - Elementos multimedia.



Web con los ejemplos del seminario: <http://labtelema.ujaen.es/ssaatt/ejemplos/index.html#titulo1>

3. Elementos comunes de HTML

Encabezados

- Los encabezados son cualquier texto entre una marca `<h1>` a `<h6>`.
- Se emplean como títulos de sección para facilitar la lectura de un documento y completar su estructura.
- Tipos de encabezado:
 - Cabecera 1: `<h1>Cabecera 1</h1>`
 - Cabecera 2: `<h2>Cabecera 2</h2>`
 - Cabecera 3: `<h3>Cabecera 3</h3>`
 - Cabecera 4: `<h4>Cabecera 4</h4>`
 - Cabecera 5: `<h5>Cabecera 5</h5>`
 - Cabecera 6: `<h6>Cabecera 6</h6>`
- Como división del contenido también se emplea la barra horizontal: `<hr>`

Cabecera 1

Cabecera 2

Cabecera 3

Cabecera 4

Cabecera 5

Cabecera 6

Encima de la línea

Debajo de la línea

Ejemplo de visualización
en Mozilla Firefox



Sí



Sí



Sí



Sí



Sí

3. Elementos comunes de HTML

Párrafos

- La etiqueta `<p>` define párrafos, de comportamiento similar a los que se emplean en los documentos escritos con procesadores de texto.
 - Un párrafo se define entre la marca de apertura `<p>` y la de cierre `</p>`.
 - Habitualmente, los navegadores interpretarán correctamente los párrafos que no tengan marca de cierre (en la norma está previsto para ciertos casos).
 - **NOTA:** se recomienda no omitirla ya que a veces puede ocasionar resultados inesperados.
- El atributo *title* permite que se muestre una *tooltip* (pequeño mensaje emergente) al pasar el cursor sobre el párrafo (*title* se puede aplicar a más elementos HTML).



Sí



Sí



Sí



Sí



Sí

3. Elementos comunes de HTML

Párrafos

- Los navegadores añaden un poco de espacio antes y después de cada párrafo (márgenes – *margin*).
 - Estos márgenes pueden ser modificados con CSS a través de las propiedades de márgenes (*margin*).
- Diferencias entre HTML 4.01 y HTML5: el atributo *align* no está soportado en HTML5, para cambiar la alineación de un párrafo se deben emplear estilos CSS (*text-align*).



Sí



Sí



Sí



Sí



Sí

3. Elementos comunes de HTML

Párrafos

*Salto de línea
 y párrafos preformateados <pre>*

- Para introducir un salto de línea sin comenzar un nuevo párrafo se emplea la marca
 .
 -
 es una etiqueta vacía y no tiene marca de cierre.
 - No se debe emplear para forzar saltos de línea de párrafos, pero sí en casos por direcciones o poemas, **ni tampoco para separar elementos entre sí, como un formulario de un párrafo, para esto se debe emplear CSS.**
- Los párrafos encerrados entre las marcas <pre>...</pre> se representan en una fuente mono espaciada (normalmente Courier) y mantienen los espacios y saltos de línea.
 - Se suele emplear para representar código fuente (también con <code> que es en línea) o textos similares como las RFC (Request for Comment).



Sí



Sí



Sí



Sí



Sí

3. Elementos comunes de HTML

Párrafos

Ejemplo

```
<!DOCTYPE html>
<html lang="en" dir="ltr">
<head><meta charset="utf-8"><title>Ejemplo de párrafos</title></head>
<body>
  Esta primera línea es sin formato
  <p title="Párrafo 1">Párrafo 1. Ahora se separa de la anterior</p>
  <p>Y a continuación párrafo 2 </p>
  <p>Párrafo con salto de <br>línea</p>
  <pre>
+ + + + + + + +
+               +
+               +
+      pre      +
+               +
+               +
+ + + + + + + +
  </pre>
</body></html>
```

Esta primera línea es sin formato

Párrafo 1. Ahora se separa de la anterior

Y a continuación párrafo 2

Párrafo con salto de
línea

```
+ + + + + + + +
+               +
+               +
+      pre      +
+               +
+               +
+ + + + + + + +
```

Ejemplo de visualización
en Mozilla Firefox

3. Elementos comunes de HTML

Listas

- Podemos encontrar tres tipos de listas:
 - **Numeradas:** los elementos van precedidos de un número o carácter que indica su orden dentro de la lista, normalmente creciente.
 - **No numeradas:** los elementos van precedidos de un símbolo que los distingue como elemento de una lista, pero sin indicación de orden.
 - **Definiciones:** muestran en orden un término y su definición.
- Las listas pueden anidarse.
 - Los elementos de las listas () pueden incluir otros elementos HTML, como imágenes, enlaces,...



Sí



Sí



Sí



Sí



Sí

3. Elementos comunes de HTML

Listas

Listas numeradas

- Se delimitan entre las marcas `<ol>` y `</ol>`.
- Cada elemento se delimita con la marca `<li>`.
- El atributo *type* establece el marcador de elemento.
- El atributo *start* establece el valor de inicio de la lista, que por defecto es 1.
- El atributo *reversed* hace que la lista se ordene de manera descendente.

type	Descripción
type="1"	Ordenados por número (por defecto)
type="A"	Numerado con letras mayúsculas
type="a"	Numerado con letras minúsculas
type="I"	Numerado con números romanos en mayúscula.
type="i"	Numerado con números romanos en minúscula



Sí



Sí



Sí



Sí



Sí

3. Elementos comunes de HTML

Listas

Listas numeradas

```
<ol type="I">
  <li>Contabilidad</li>
  <li>Ventas</li>
  <li>Recursos humanos</li>
</ol>
<ol start="5">
  <li>Contabilidad</li>
  <li>Ventas</li>
  <li>Recursos humanos</li>
</ol>
<ol reversed>
  <li>Contabilidad</li>
  <li>Ventas</li>
  <li>Recursos humanos</li>
</ol>
```

Lista numerada

I. Contabilidad
II. Ventas
III. Recursos humanos

5. Contabilidad
6. Ventas
7. Recursos humanos

3. Contabilidad
2. Ventas
1. Recursos humanos

Ejemplo de visualización
en Mozilla Firefox

Ejemplo

3. Elementos comunes de HTML

Listas

Listas no numeradas

- Se delimitan entre las marcas `<ul>` y `</ul>`.
- Cada elemento se delimita con la marca `<li>`.
- El símbolo que marca cada elemento de la lista se cambia con estilos CSS en HTML5: `<ul style="list-style-type:disc;">`

```
<ul>
  <li>Contabilidad</li>
  <li>Ventas</li>
  <li>Recursos humanos</li>
</ul>
<ul style="list-style-
type:circle;">
  <li>Contabilidad</li>
  <li>Ventas</li>
  <li>Recursos humanos</li>
</ul>
```

Ejemplo de
visualización en
Mozilla Firefox

Listas no numeradas

- Contabilidad
 - Ventas
 - Recursos humanos
-
- Contabilidad
 - Ventas
 - Recursos humanos



Sí



Sí



Sí



Sí



Sí

3. Elementos comunes de HTML

Listas

Listas de definiciones

- Se delimitan entre las marcas `<dl>` y `</dl>`.
- Entre la marca `<dl>` se encierran elementos `<dt>` para los términos y `<dd>` para la definición.

```
<dl>  
  <dt>UNO:</dt>  
  <dd>DEFINICION 1</dd>  
  <dt>DOS: </dt>  
  <dd>DEFINICION 2</dd>  
  <dt>TRES:</dt>  
  <dd>DEFINICION 3</dd>  
</dl>
```

Lista de definiciones

UNO:
DEFINICION 1
DOS:
DEFINICION 2
TRES:
DEFINICION 3

Ejemplo de visualización
en Mozilla Firefox



Sí



Sí



Sí



Sí



Sí

3. Elementos comunes de HTML

Enfatizar texto

- Estos elementos se emplean para dar un significado especial al texto.

- Ejemplo:

```
texto normal<br>
<em>texto resaltado</em><br>
<strong>texto muy resaltado</strong><br>
<code>código fuente</code><br>
<samp>texto como ejemplo</samp><br>
<kbd>texto a teclear</kbd><br>
<var>nombre de variable</var><br>
<dfn>definicion</dfn><br>
<cite>una cita</cite><br>
<q>comillas</q>
```

Énfasis de texto

```
texto normal
texto resaltado
texto muy resaltado
código fuente
texto como ejemplo
texto a teclear
nombre de variable
definicion
una cita
“comillas”
```

Ejemplo de visualización
en Mozilla Firefox



Sí



Sí



Sí



Sí



Sí

3. Elementos comunes de HTML

Enfatizar texto

Formatos de texto

- Ejemplo

`<p>Este <b> texto en negrita</b> le sigue <i>texto en cursiva</i> y un <mark>texto marcado</mark><br>`

`A un <small>texto pequeño</small> le sigue un <del>texto eliminado</del> y otro <ins>insertado</ins><br>`

`Este <sub>texto es un subíndice</sub> y este otro <sup>un superíndice</sup>.<br></p>`

Este **texto en negrita** le sigue *texto en cursiva* y un **texto marcado**
A un texto pequeño le sigue un ~~texto eliminado~~ y otro insertado
Este _{texto es un subíndice} Y este otro ^{un superíndice}.

Ejemplo de visualización en Mozilla Firefox



Sí



Sí



Sí



Sí



Sí

3. Elementos comunes de HTML

Enfatizar texto

*La marca *

- Realmente la marca `<span>` no aporta ningún énfasis o estilo por defecto al texto, pero se emplea para definir otras funcionalidades a una porción de texto en línea.
- `<p>Mi gato tiene ojos <span style="color:green">verdes</span>.</p>`
- Otro ejemplo de su uso es con cualquiera de los atributos globales o de eventos.



Sí



Sí



Sí



Sí



Sí

3. Elementos comunes de HTML

Enlaces

- Los enlaces son la base del origen del lenguaje HTML y el protocolo HTTP.
- Los enlaces se definen con la marca `<a>` que proviene del inglés “anchor” o ancla.

```
<a href="https://ujaen.es/">UJA</a>
```
- El texto/elemento que se mostrará será el que se incluya entre las marcas `<a>` y `</a>`.
- El atributo *href* especifica la dirección destino del enlace.
 - Al pulsar en un enlace el navegador intentará mostrar el documento que se especifica en la dirección destino.
 - También se puede especificar enlaces locales (no se incluye el esquema y la autoridad, y es relativo a la ubicación del archivo que tiene el enlace):

```
<a href="noticias.html">Noticias</a>
```
 - Enlace local desde la raíz del sitio web:

```
<a href="/noticias.html">Noticias</a>
```
 - La barra final es importante, ya que permite diferenciar si el contenido solicitado es un directorio (con `/`) o un fichero (sin `/`). Si hay ambigüedad, el servidor puede generar una respuesta de redirección y necesitarse una nueva petición.



Sí



Sí



Sí



Sí



Sí

3. Elementos comunes de HTML

Enlaces

Atributo target

- El atributo *target* especifica la manera en la que se abrirá el nuevo documento:
 - **_blank**: abren el documento enlazado en una nueva ventana o pestaña.
 - **_self**: es la opción por defecto, abre el documento enlazado en la misma ventana/pestaña en HTML5. En HTML4 sería el mismo marco*.
 - **_parent**: abre el documento enlazado en el marco* padre.
 - **_top**: Abre el documento enlazado en el cuerpo de la ventana saltándose la jerarquía de marcos*.
 - **Frame-name***: abre el documento enlazado en el marco que se nombra.

*: Los marcos (*frames*) **no están soportados en HTML5**. Se empleaban para organizar el diseño de las páginas web y aportar cierto dinamismo. Se empleaban las etiquetas `<frameset>` y `<frame>`. Esto hace que en HTML5, el valor de *target*: `_self`, `_parent` y `_top`, tengan comportamientos equivalentes.



Sí



Sí



Sí



Sí



Sí

3. Elementos comunes de HTML

Enlaces

Puntos de anclaje

- Punto de anclaje HTML4:
`<h1 name="chapter1">Capítulo 1</h1>`
- Punto de anclaje **HTML5**:
`<h1 id="chapter1">Capítulo 1</h1>`
- Saltar a un punto de anclaje:
`<a href="#chapter1">sitio</a>`
- Dentro de una URL
`<a href="http://.../book.html#chapter1">sitio</a>`



Sí



Sí



Sí



Sí



Sí

3. Elementos comunes de HTML

Enlaces

Ejemplo

```
<!DOCTYPE html >
<html>
  <head>
    <title>Enlace a una página web</title>
  </head>
  <body>
    <h1 id="titulo1">Enlaces a una página web</h1>
    <a href="http://www.ujaen.es/">Pulsa aquí para la web de la Universidad de Jaén.
    </a><br>
    <a href="http://www.ujaen.es/" target="_blank">
      Ahora la web de la Universidad de Jaén en una pestaña nueva.
    </a>
    ...
    <a href="#titulo1">Volver arriba (esto es un enlace a un ancla)</a>
  </body>
</html>
```

3. Elementos comunes de HTML

Imágenes

- Las imágenes son fundamentales para aportar información gráfica y decoración en la web actual.
- Se incluyen a través de la marca aislada `<img>`:
 - El atributo *src* indica la URL absoluta o relativa a la imagen.
 - El atributo *alt*, obligatorio para la adecuada validación de la página, debe incluir una descripción de lo que se aprecia en la imagen ya que es el reemplazo cuando no está disponible la imagen.
 - El atributo de alineación de imágenes (*align*) no está permitido en HTML5.

```

```



Sí



Sí



Sí



Sí



Sí

3. Elementos comunes de HTML

Imágenes

Formatos de imágenes

- Las imágenes en Internet son unos de los recursos más intercambiados hoy en día.
 - Su tamaño es mucho mayor comparado con el código HTML o el del código como JavaScript.
 - El tipo de fichero gráfico es determinante en este tamaño.
 - Hay formatos sin compresión y con compresión, normalmente con pérdidas.
- Formatos más habituales en páginas web:
 - **GIF**: 256 colores sin perdidas. Adecuado para gráficas con datos. Puede incluir animaciones.
 - **JPEG**: imágenes en color verdadero y con perdidas. Adecuado para fotos. Tienen diversos niveles de compresión.
 - **PNG**: Diseñado para la web. Tiene compresión y permite máscaras de transparencias.
 - **SVG**: Formato vectorial. Especialmente recomendado para logotipos, formas geométricas y decoración. Se definen entre la marca `<svg>...</svg>`

svg



4.0



9.0



3.0



10.1



3.2

3. Elementos comunes de HTML

Imágenes

Nuevos formatos de imágenes

- Su soporte será gradual pero ya son comunes.
 - **WebP**: lanzado en 2010 por Google (pero Firefox no lo incluyó hasta 2019), es un formato con o sin pérdidas a partir del codificador VP8 empleado para la codificación de cuadros de video. También soporta transparencia y en formato sin pérdidas consigue un 26% menos tamaño que PNG y con pérdidas un 25-34% menos que JPEG.
 - **AVIF**: también es un formato basado en un códec de vídeo (estándar ISO HEIF y el códec de vídeo AV1). Es de libre uso y consigue mejores resultados incluso que WebP (50% con respecto a un JPEG con la misma calidad percibida).

3. Elementos comunes de HTML

Imágenes

Tamaño de las imágenes

- Siempre se debe especificar el tamaño de una imagen.
 - Se detallan en píxeles.
 - Se evitan fluctuaciones en el navegador al cargar las imágenes.
- Se pueden emplear estilos, o los atributos *width* y *height*, en este último caso no se especifican las unidades.

```

```

```

```

3. Elementos comunes de HTML

Imágenes

Colocación de las imágenes

- En HTML5 las imágenes se deben ubicar a través de estilos o insertándolas en otros elementos que ya tengan una ubicación.
 - Ubicación libre a la derecha o izquierda: **style="float:right | left"**
 - Ubicación con respecto al texto que la rodea: **style="vertical-align:bottom | middle | top"**
 - Recuerda: el atributo ***align*** no está permitido en HTML5 (sí en HTML4).

3. Elementos comunes de HTML

Tablas

- Las tablas tienen la misión de estructurar datos de manera tabulada.
- **No se deben emplear para el diseño de la página web.**
- Las tablas se delimitan por la etiqueta `<table>` y `</table>`.
- Los elementos que se pueden añadir a una tabla son muy diversos, y dependerá de qué información se desee mostrar y cómo.
 - Se pueden poner títulos, encabezados a filas y columnas, fusionar celdas, etc.



Sí



Sí



Sí



Sí



Sí

3. Elementos comunes de HTML

- Tablas

Elementos que se pueden incluir a una tabla

- `<caption>...</caption>`: especifica el título de la tabla. Debe ir justo a continuación de la etiqueta `<table>`.
- `<thead>...</thead>`: agrupa los encabezados de las columnas de una tabla.
- `<tbody>...</tbody>`: es el elemento que contendría las filas de la tabla. Se puede omitir si simplemente se incluyen elementos `<tr>`.
- `<tfoot>...</tfoot>`: resultados/pie de las columnas.
- `<tr>...</tr>`: define una fila de la tabla y puede contener celdas y encabezados.
- `<td>...</td>`: define una celda dentro de una fila. Son el elemento contenedor de información y puede contener cualquier elemento de HTML (texto, imágenes, enlaces, otras tablas,...).
- `<th>...</th>`: definen encabezados de fila o columna, tanto en el encabezado de la tabla como en el cuerpo.

3. Elementos comunes de HTML

Tablas

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

ID	Measurement	Average	Maximum
	Cats		
93	Legs	3.5	4
10	Tails	1	1
	English speakers		
32	Legs	2.67	4
35	Tails	0.33	1

Ejemplo en:

<https://html.spec.whatwg.org/multipage/tables.html#the-th-element>

En este ejemplo, como puede observarse se omiten muchas de las marcas de cierre, pero esto está permitido en la norma para estos casos particulares.

3. Elementos comunes de HTML

Tablas

Estilos para las tablas

- En HTML 5 el estilo de las tablas, bordes, fondo, etc., debe ser añadido a través de CSS.
 - Los atributos de HTML4.01 *align*, *bgcolor*, *border*, *cellpadding*, *cellspacing*, *frame*, *rules*, *summary* y *width* no están soportados en HTML5.

```
<style>
  table, th, td {
    border: 1px solid black;
    border-collapse: collapse;
  }

  th, td {
    padding: 5px;
    text-align: left;
  }
</style>
```

3. Elementos comunes de HTML

Tablas

Ejemplo

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="utf-8">
  <title>Imágenes</title>
  <style>
    table, th, td {
      border: 1px solid black;
      border-collapse: collapse;
    }
    th, td {
      padding: 5px;
      text-align: left;
    }
  </style>
</head>
<body>
<h1>Tablas</h1>

  <table>
    <caption>TÍTULO DE LA TABLA</caption>
    <tr>
      <th></th>
      <th>Columna 1</th>
      <th>Columna 2</th>
      <th>Columna 3</th>
    </tr>
    <tr>
      <th>Fila 1</th>
      <td>Fila 1; Columna 1</td>
      <td>Fila 1; Columna 2</td>
      <td>Fila 1; Columna 3</td>
    </tr>
    <tr>
      <th>Fila 2</th>
      <td>Fila 2; Columna 1</td>
      <td>Fila 2; Columna 2</td>
      <td>Fila 2; Columna 3</td>
    </tr>
  </table>
</body>
</html>
```

Tablas

TÍTULO DE LA TABLA

	Columna 1	Columna 2	Columna 3
Fila 1	Fila 1; Columna 1	Fila 1; Columna 2	Fila 1; Columna 3
Fila 2	Fila 2; Columna 1	Fila 2; Columna 2	Fila 2; Columna 3

Ejemplo de visualización
en Mozilla Firefox

3. Elementos comunes de HTML

Tablas

Unión de celdas

- Las celdas adyacentes, en horizontal o vertical se pueden unir. Se consigue añadiendo un atributo al elemento `<td>`:
 - En horizontal: se emplea el atributo *colspan* indicando el número de celdas a agrupar.
 - En vertical: se emplea el atributo *rowspan* indicando el número de filas a agrupar.

3. Elementos comunes de HTML

Tablas

Ejemplo

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

  <style>
    table, th, td {
      border: 1px solid black;
      border-collapse: collapse;
    }

    th, td {
      padding: 5px;
      text-align: left;
    }
  </style>
</head>
<body>
<h1>Tablas</h1>

<table>
  <caption>Unión de celdas</caption>
  <tr>
```

```
    <th>Tipo</th>
    <th>Nombre</th>
    <th colspan="2">Apellidos</th>
  </tr>
  <tr>
    <td rowspan="2">Admin</td>
    <td>Juan</td>
    <td>García</td>
    <td>Pérez</td>
  </tr>
  <tr>
    <td>Marta</td>
    <td>Fernández</td>
    <td>López</td>
  </tr>
</table>
</body>
</html>
```

Unión de celdas

Tipo	Nombre	Apellidos	
Admin	Juan	García	Pérez
	Marta	Fernández	López

Ejemplo de visualización
en Mozilla Firefox

4. Formularios

- Los formularios aportan herramientas para interactuar con un documento HTML.
- Permiten enviar información a un servidor web desde el navegador (o a un script).
- Se crean con la etiqueta **<form>** y pueden contener una gran diversidad de elementos de entrada de datos **<input>**, además de otros elementos HTML.
 - Un elemento **<form>** no es visible normalmente.
 - Los elementos en un formulario, por defecto, siempre se muestran en línea.
 - No se pueden anidar formularios, pero sí incluir varios en la misma página.

```
<form action="/login.php" method="post">
  <label>Usuario:
    <input type="text" name="user">
  </label><br>
  <label>Contraseña:
    <input type="password" name="pass">
  </label><br>
  <input type="submit" value="Entrar">
</form>
```



Sí



Sí



Sí



Sí



Sí

4. Formularios

Introducción

Funcionamiento

1. El formulario se rellena con datos en el navegador.
2. Los datos se envían al servidor a través de una petición HTTP, empleando un método POST o GET.
3. El servidor recibe la petición y pasa los datos a un recurso en el cual debe formar parte de una aplicación.
4. La salida de la ejecución del programa implementado en ese recurso se reenvía al navegador en una respuesta HTTP.

4. Formularios

Características

Atributos de los formularios

Atributo	Descripción
<i>accept-charset</i>	Especifica el juego de caracteres que se empleará al enviar el formulario (por defecto se emplea el mismo que el de la página).
<i>action</i>	Especifica la dirección (URL) donde enviar el formulario (por defecto la misma página donde está el formulario).
<i>autocomplete</i>	Define si el navegador debe emplear el auto-completado en el formulario (por defecto está activado "on").
<i>enctype</i>	Especifica la codificación en la información enviada (por defecto es application/x-www-form-urlencoded).
<i>method</i>	Especifica el método HTTP a emplear al enviar los datos del formulario (por defecto GET, más habitual POST).
<i>name</i>	Especifica el nombre por el que se identifica el formulario en el DOM (A través del objeto <code>document.forms.name</code>).
<i>novalidate</i>	Especifica que el navegador no debe validar el formulario.
<i>target</i>	Especifica el objetivo de la dirección especificada en el atributo <i>action</i> (por defecto <code>_self</code>). Es equivalente al mismo atributo en los enlaces.

4. Formularios

Características

Atributos de los formularios – action y method

- El atributo *action* contiene la URL del recurso que deberá procesar los datos.
 - Normalmente está implementado con alguna tecnología de servidor.
- El atributo *method* marca el método HTTP que se empleará para enviar:
 - GET: los datos se envían como *queries* en la URL, por lo que son visibles.
 - POST: los datos se envían en la parte de entidad de la petición, no son visibles y no se puede guardar un marcador.
 - Aunque no sean visibles a simple vista en el navegador, no incorporan ningún otro mecanismo de protección.

```
<form action="/action_page.php" method="get" >  
  Nombre:<br>  
  <input type="text" name="nombre"><br>  
  Apellidos:<br>  
  <input type="text" name="apellidos" ><br><br>  
  <input type="submit" value="Enviar">  
</form>
```

Notas acerca del uso de GET:

- Añade la Información del formulario dentro de la URL en forma de pares nombre/valor.
- La longitud de una URL está limitada a 2048 caracteres.
- Nunca se debe emplear GET para enviar datos sensibles ya que serán visibles en la URL.
- Es útil cuando el usuario puede estar interesado en guardar el marcador a esa URL.

4. Formularios

Características

Atributos de los formularios – enctype

- El atributo *enctype* puede tomar tres valores:
 - **application/x-www-form-urlencoded**: Es el valor por defecto. Todos los caracteres son codificados antes de enviar (los espacios son convertidos a “+” y los caracteres especiales a los valores hexadecimales ASCII).
 - Los valores de los formularios se codifican nombre=valor.
 - Por ejemplo, un campo de texto *name* con contenido “Juan Manuel” sería:
name=Juan+Manuel
 - Si hay más de un dato a enviar se separan por el símbolo “&”.
 - Por ejemplo, si al ejemplo anterior añadimos la edad: name=juan+manuel&age=20
 - **multipart/form-data**: Los caracteres no son codificados. Este valor es requerido cuando se emplea el formulario para enviar archivos.
 - **text/plain**: Los espacios son convertidos a “+”, pero los caracteres especiales no.

4. Formularios

Entrada de información

Elementos de entrada de información <input>

- Fundamentalmente los elementos para introducir información en un formulario son los elementos **<input>**.
 - **<input>** es un elemento aislado.
 - Su representación y tipo de dato a introducir se especifica en el atributo *type* (si se omite el tipo por defecto sería "text").
 - Cada elemento **<input>** debe tener un atributo ***name*** que identificará al valor cuando se envíen los datos. Si no se especifica el atributo *name* el valor de ese campo no se enviará.
 - Si se quiere que el elemento input muestre un valor de inicio se puede incluir el atributo ***value***.

4. Formularios

Entrada de información

Elementos de entrada de información

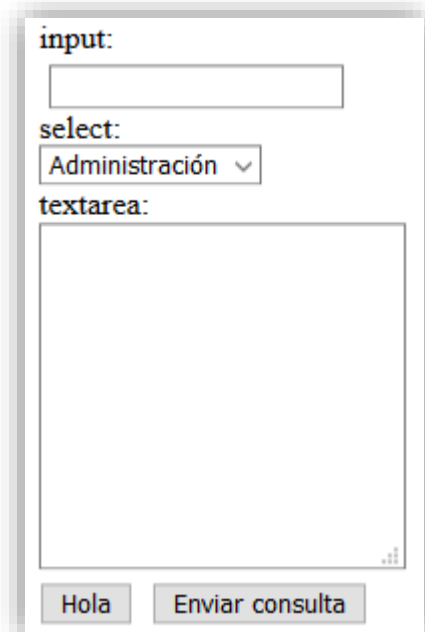
- **<select>... </select>** para seleccionar uno o varios valores entre varios disponibles. Si se le añade el atributo *multiple* permite seleccionar varios. Con *size* define el número de opciones visibles en la lista desplegable.
 - Cada elemento se especifica entre las marcas **<option>...</option>** y el valor que se envía al servidor se añade en el atributo *value*.
- **<textarea>... </textarea>** para introducir textos con múltiples líneas. Sus atributos *rows* y *cols* definen las filas y columnas que se mostrarán.
- **<button>** para representar botones y disparar efectos programados con un script a través de atributos de evento como ***onclick***. Se debe especificar siempre *type="button"* para evitar inconsistencias entre navegadores.

4. Formularios

Entrada de información

Elementos de entrada de información

```
<form action="/action_page.php" method="get">
  input:<br><input type="text" name="nombre"><br>
  select:<br><select name="departamento">
    <option value="com">Comercial</option>
    <option value="admon">Administración</option>
  </select><br>
  textarea:<br><textarea name="comentario"
rows="10" cols="20"></textarea><br>
  <button type="button"
onclick="alert('Hola')">Hola</button>
  <input type="submit" value="Enviar">
</form>
```



input:

select:
Administración ▾

textarea:

Hola Enviar consulta

Ejemplo de visualización
en Mozilla Firefox

Ejemplo

4. Formularios

Entrada de información

Elementos de entrada de información

- HTML5 añadió también:
 - `<datalist>` permite predefinir una lista de valores para un elemento `<input>` y se mostrará una lista desplegable conforme se escribe.
 - El atributo *list* del elemento `<input>` debe coincidir con el *id* de la `<datalist>`.
 - `<output>` está diseñado para representar el resultado de un cálculo, como lo que podría realizar un *script*.

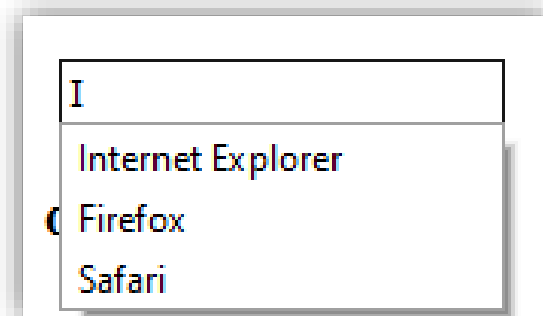
4. Formularios

Entrada de información

Elementos de entrada de información

Ejemplo <datalist>

```
<form action="/action_page.php">
  <input list="browsers">
  <datalist id="browsers">
    <option value="Internet Explorer">
    <option value="Firefox">
    <option value="Chrome">
    <option value="Opera">
    <option value="Safari">
  </datalist>
</form>
```



Ejemplo de visualización
en Mozilla Firefox

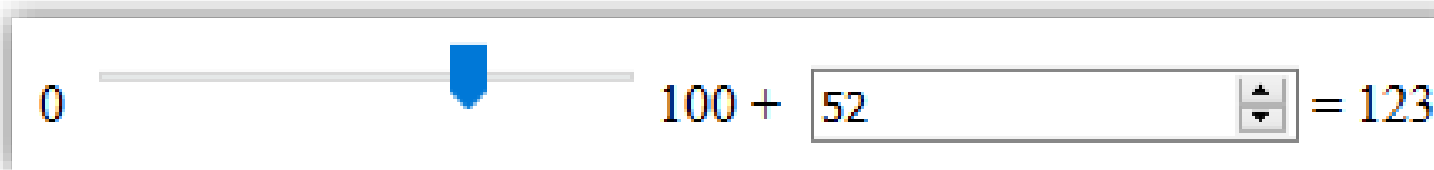
4. Formularios

Entrada de información

Elementos de entrada de información

```
<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">  
  0 <input type="range" id="a" name="a" value="50">  
  100 + <input type="number" id="b" name="b" value="50">  
  = <output name="x" for="a b"></output><br><br>  
  <input type="submit">  
</form>
```

Esto es
JavaScript



0 100 + 52 = 123

Ejemplo de visualización
en Mozilla Firefox

Ejemplo <output>

4. Formularios

Entrada de información

Tipos de elementos de entrada de información <input> más comunes

- **checkbox:** Casilla de selección. Se debe usar el atributo *value* para definir el valor que se enviará por este elemento.
 - Se usa el atributo booleano *checked* para indicar si el elemento está seleccionado.
- **radio:** Botón radio. Se debe usar el atributo *value* para definir el valor que se enviará por este elemento.
 - Se usa el atributo *checked* para indicar si el elemento está seleccionado de forma predeterminada.
 - Los botones radio que tengan el mismo valor para su atributo *name* están dentro del mismo "grupo de botones radio".
 - Solo un botón radio dentro de un grupo puede ser seleccionado a la vez.
- **submit:** Botón que envía el formulario. En el atributo *value* se puede cambiar el texto a visualizar.
- **text:** Campo de texto de línea simple. Los saltos de línea son eliminados automáticamente del valor introducido.

4. Formularios

Entrada de información

Tipos de elementos de entrada de información <input> más comunes

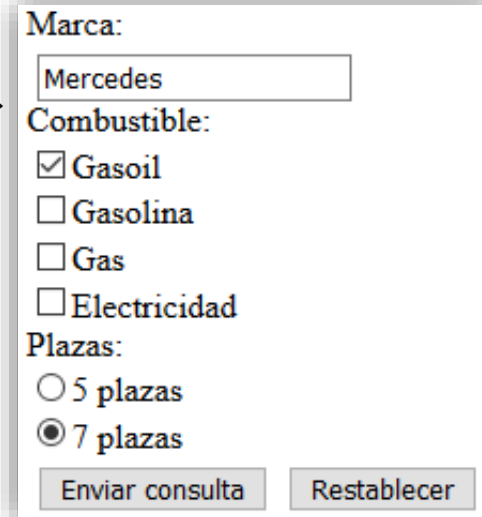
- **password**: oculta los caracteres introducidos.
- **button**: Botón sin un comportamiento específico.
 - Se suelen emplear cuando se quieren emplear estilos especiales para el control de la página web o un formulario.
 - Dentro de los formularios es mejor usar <input> por problemas de interpretación de algunos navegadores.
- **file**: Control que permite al usuario seleccionar un archivo.
 - Se puede usar el atributo *accept* para definir los tipos de archivo que el control podrá seleccionar.
- **hidden**: Control que no es mostrado en pantalla, pero cuyo valor es enviado al servidor.
 - Se suelen emplear para guardar estados y/o la historia para procesar el formulario.
- **reset**: Botón que restaura los contenidos de un formulario a sus valores predeterminados.

4. Formularios

Entrada de información

Tipos de elementos de entrada de información <input> más comunes

```
<form>
  Marca:<br><input type="text" name="brand" value="Mercedes"><br>
  Combustible:<br>
  <input type="checkbox" name="fuel" value="Gasoil" checked>Gasoil<br>
  <input type="checkbox" name="fuel" value="Gasolina">Gasolina<br>
  <input type="checkbox" name="fuel" value="Gas">Gas<br>
  <input type="checkbox" name="fuel" value="Elec">Eléctrico<br>
  Plazas:<br>
  <input type="radio" name="seats" value="5">5 plazas<br>
  <input type="radio" name="seats" value="7">7 plazas<br>
  <input type="hidden" value="offer0223AD">
  <input type="submit" value="Enviar">
  <input type="reset">
</form>
```



Ejemplo de visualización
en Mozilla Firefox

Ejemplo de tipos de input

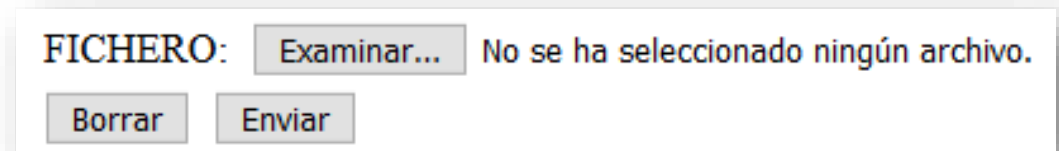
4. Formularios

Entrada de información

Tipos de elementos de entrada de información <input> más comunes

- Seleccionar y enviar archivos, tipo *file*:

```
<form action="http://direccion/file.jsp" method="post"
enctype="multipart/form-data">
  FICHERO:<br>
  <input type="file" name="fichero"/><br/>
  <input type="reset" value="Borrar"/>
  <input type="submit" value="Enviar"/>
</form>
```



FICHERO: No se ha seleccionado ningún archivo.

Ejemplo de visualización en Mozilla Firefox

4. Formularios

Entrada de información

Tipos de elementos de entrada de información <input> más comunes - tipo image

- Emplear imágenes para enviar un formulario: tipo *image*
 - Sustituyen al componente de tipo submit.
 - El atributo *src* especifica la imagen.
 - Esto permite hacer imágenes con las que interactuar como, por ejemplo, un mapa mudo para hacer preguntas.
 - Se pueden emplear otros atributos para imágenes.
 - Si no se especifica el atributo *name* se envía: *x=posicionX* y *y=posicionY*
 - Con el atributo *name* se envía *name_x* y *name_y*.

```
<form action="/form.php" method="post">  
  <input type="image" src="UJA.jpg" alt="enviar" name="logo">  
</form>
```

4. Formularios

Entrada de información

Tipos de elementos de entrada de información <input> más comunes

Nuevos tipos de <input> de HTML5

color	Control para especificar un color.
date	Control para introducir una fecha (año, mes y día, sin tiempo).
datetime-local	Control para introducir fecha y hora, sin zona horaria específica.
email	Campo para introducir una dirección de correo electrónico.
month	Control para introducir un mes y año, sin zona horaria específica.
number	Control para introducir un número de punto flotante. La propiedad step , que por defecto vale uno, es la que permite la introducción de valores decimales si se pone, por ejemplo en 0.1 o en any .
range	Control para introducir un número cuyo valor exacto no es importante.
search	Cuadro de texto de línea simple para introducir textos de búsqueda. Los saltos de línea son eliminados automáticamente del valor introducido.
tel	Control para introducir un número telefónico.
time	Control para introducir un valor de tiempo sin zona horaria específica.
url	Campo para editar una URL.
week	Control para introducir una fecha que consiste en número de semana del año y número de semana sin zona horaria específica.

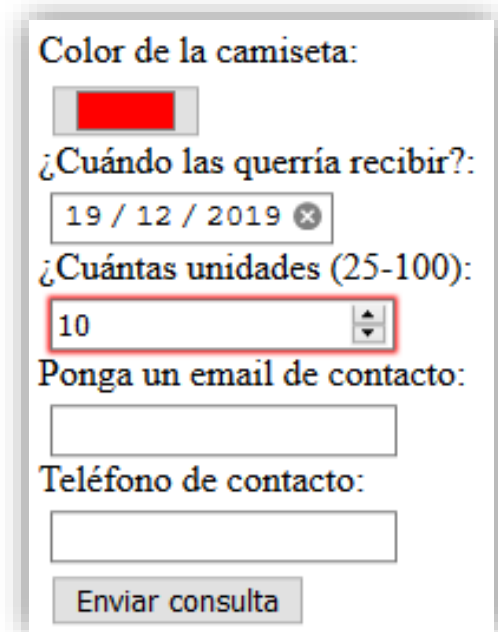
4. Formularios

Entrada de información

Tipos de elementos de entrada de información <input> más comunes

Ejemplo de tipos nuevos en HTML5

```
<form>
  Color de la camiseta:<br>
  <input type="color" name="color" value="#ff0000"><br>
  ¿Cuándo las querría recibir?:<br>
  <input type="date" name="date" ><br>
  ¿Cuántas unidades (25-100):<br>
  <input type="number" name="uds" min="25" max="100"
    step="any"><br>
  Ponga un email de contacto:<br>
  <input type="email" name="email"><br>
  Teléfono de contacto:<br>
  <input type="tel" name="phone"><br>
  <input type="submit" value="Enviar">
</form>
```



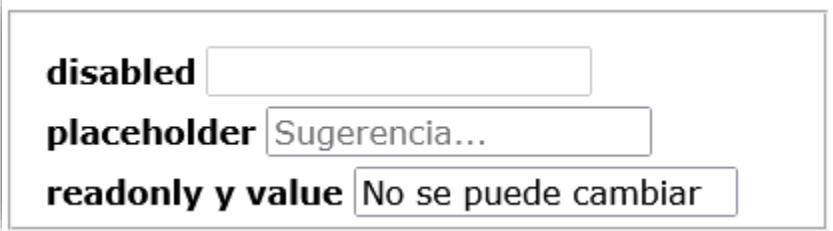
Ejemplo de visualización
en Mozilla Firefox

4. Formularios

Entrada de información

Atributos de información y ayuda

- A los campos **input** de formulario se le pueden añadir otros atributos para adaptarlos a las necesidades del formulario:
 - **disabled**: el campo está deshabilitado y no se puede interactuar con él.
 - **placeholder**: muestra un texto cuando el campo está vacío.
 - **readonly**: define el campo como de solo lectura y el valor no es editable.
 - **value**: el valor inicial del campo.



disabled

placeholder

readonly y value

Ejemplo de visualización
en Mozilla Firefox

```
<form action="/form.php" method="post">
  <fieldset>
    <label>disabled<input type="text" name="name" disabled></label>
    <label>placeholder<input type="text" name="p" placeholder="Sugerencia..."></label>
    <label>readonly y value<input type="text" name="ro" value="No se puede cambiar" readonly></label>
  </fieldset>
</form>
```

4. Formularios

Validación de formularios

- El objetivo fundamental de un formulario es la recopilación de datos introducidos por el usuario para ser enviados a un servidor (normalmente a una aplicación).
 - Este proceso debe estar acompañado de una **validación en el lado del cliente**: detectar valores no válidos, que no sigan un patrón determinado o falten. Este proceso se denomina **validación del formulario**.
 - Para conseguir que ciertos campos de un formulario puedan validar los datos, se necesita de atributos especiales como **required**. Además, algunos elementos tienen incorporada cierta validación como los de tipo **email**.
 - Si alguno de los campos de un formulario no es válido según su diseño o los datos introducidos no cumplen con las restricciones impuestas en el HTML, el navegador nos mostrará un error y no enviará ningún dato.
 - La validación en el lado del cliente siempre debe estar complementada por la **validación en el lado del servidor**, ya que es relativamente sencillo enviar datos no validados al servidor desde un cliente (normalmente un navegador).

4. Formularios

Validación de formularios

¿Por qué se debe validar un formulario?

- **Se deben obtener los datos correctos en el formato correcto.** Las aplicaciones que se desarrollen no funcionarán correctamente si los datos introducidos por los usuarios se almacenan en el formato incorrecto, son incorrectos o se omiten por completo.
- **Debemos proteger los datos de los usuarios.** Obligar a los usuarios a introducir contraseñas seguras facilita proteger la información de su cuenta.
- **Debemos protegernos a nosotros mismos.** Hay muchas formas en que los usuarios maliciosos puedan usar mal los formularios desprotegidos y dañar la aplicación.



4. Formularios

Validación de formularios

¿Cómo podemos validar un formulario?

- La validación de formularios incorporada utiliza características de validación de formularios HTML5.
 - Esta validación generalmente no requiere mucho JavaScript.
 - La validación de formularios incorporada tiene un mejor rendimiento que JavaScript, pero no es tan personalizable como la validación con JavaScript.
- La validación con JavaScript se lleva a cabo utilizando scripts de este lenguaje.
 - Esta validación es completamente personalizable, pero se debe implementar todo (o usar una biblioteca).

4. Formularios

Validación de formularios

Usar la validación de formulario incorporada

- Una de las características más importantes de los controles de formulario de HTML5 es la capacidad de validar la mayoría de los datos de usuario sin depender de JavaScript.
- Esto se realiza mediante el uso de atributos de validación en los elementos del formulario:
 - **required**: Especifica si un campo de formulario debe completarse antes de que se pueda enviar el formulario.
 - **minlength** y **maxlength**: Especifican la longitud mínima y máxima de los datos de texto (cadenas).
 - **min** y **max**: Especifican los valores mínimo y máximo de los tipos de entrada numéricos.
 - **type**: Especifica si los datos deben ser un número, una dirección de correo electrónico o algún otro tipo de preajuste específico.
 - **pattern**: Especifica una expresión regular que define un patrón que los datos que se introduzcan deben seguir.

4. Formularios

Diseño de formularios

Agrupando la información de un formulario

- Se puede emplear `<fieldset>` para agrupar elementos en un formulario.
 - Dentro de `<fieldset>` se puede poner un elemento `<legend>` que define un título para el grupo.
- Los elementos de un formulario se les puede (*debe*) añadir etiquetas con `<label>`.
 - Emplear `<label>` mejora la legibilidad de la página y su interpretación por lectores de pantalla y aumenta el área en la que se señala el componente.
 - `<label>` se puede emplear insertando el elemento de entrada entre la marca de inicio y final de `<label>`, o vinculado el elemento `<label>` a través del valor de su atributo *for*, haciéndolo coincidir con el valor del atributo *id* del elemento de entrada.

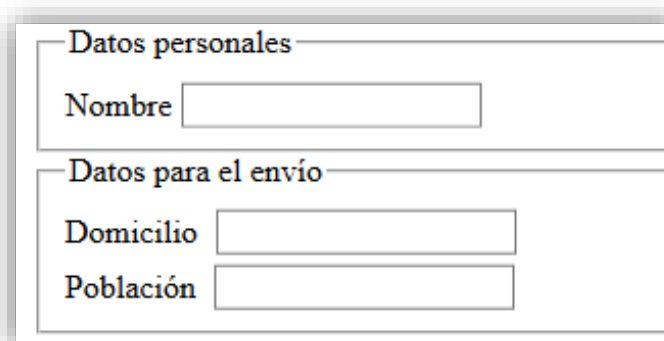
4. Formularios

Diseño de formularios

Agrupando la información de un formulario

Ejemplo <fieldset> y <label>

```
<form>
  <fieldset>
    <legend>Datos personales</legend>
    <label>Nombre<input type="text" name="name"></label>
  </fieldset>
  <fieldset>
    <legend>Datos para el envío</legend>
    <label for="domicilio">Domicilio</label>
    <input type="text" id="domicilio" name="domicilio">
    <br>
    <label for="poblacion">Población</label>
    <input type="text" id="poblacion" name="poblacion">
  </fieldset>
</form>
```



Ejemplo de visualización
en Mozilla Firefox

5. Cuadros de diálogo

- Los cuadros de diálogo, son “ventanas”, ya sean modales o no modales que pueden abrirse dentro de una página web para mostrar nueva información o llevar a cabo nuevas acciones.
 - Cuadro de diálogo modal: no permite la interacción con el resto de la página web.
 - Cuadro de diálogo no modal: permite que se interactúe con él y con el resto de la página web.
 - Su soporte en los navegadores se ha ido generalizando desde 2022.
 - Un cuadro de diálogo no modal se muestra si su atributo **open** tiene como valor **true**.
 - Se cierra pulsando el botón. Debe ser de tipo submit (por defecto).
 - Para poder volver a mostrarlo necesitamos JavaScript. El método más apropiado sería con `HTMLDialogElement.show()`.
 - Para una funcionalidad completa, el elemento `<dialog>` necesita de JavaScript.
 - Los diálogos modales emplean `HTMLDialogElement.showModal()` y `HTMLDialogElement.close()`.

```
<dialog open>
  <p>¡Hola!</p>
  <form method="dialog">
    <button>OK</button>
  </form>
</dialog>
```

Atributos booleanos:
Recuerde que simplemente por aparecer un atributo su valor es true.



37.0



79.0



98.0



24.0



15.4

6. Multimedia

Introducción

- En HTML5 el contenido multimedia se inserta en las páginas web a través de los elementos `<video>` y `<audio>`.
- **Formatos soportados.**
 - Para video por ejemplo MP4, WebM y Ogg están soportados por el estándar HTML5.
 - Para audio, formatos de audio MP3, WAV y Ogg están soportados por el estándar HTML5.

Formato de archivo	Tipo de contenido
Video MP4	video/mp4
Video WebM	video/webm
Video Ogg	video/ogg
Audio MP3	audio/mpeg
OGG	audio/ogg
WAV	audio/wav

6. Multimedia

Elementos multimedia

Elementos de vídeo

- Antes de HTML5 para reproducir vídeo en un navegador era siempre con un plug-in, como flash.
- Ahora el elemento `<video>` es la manera estándar de incluir un vídeo en una página web.
 - El atributo *controls* permite que el vídeo incluya los controles típicos de reproducción.
 - El atributo *autoplay* consigue que la reproducción comience en cuanto se cargue la página.
 - Es aconsejable que se incluyan los atributos *width* y *height* al igual que para las imágenes.
- Dentro de el elemento `<video>` se pueden incluir varios elementos `<source>`, permitiendo poner a disposición del navegador el mismo contenido (sería lo lógico) en diferentes formatos para facilitar su reproducción.
 - También se puede incluir un texto que se mostrará si el navegador no conoce el elemento `<video>`.

VIDEO



4.0



9.0



3.5



10.5



4.0

6. Multimedia

Elementos multimedia

Elementos de vídeo

```
<video controls>  
  <source src="movie.mp4" type="video/mp4">  
  <source src="movie.ogg" type="video/ogg">  
  Su navegador no soporta esta etiqueta de video.  
</video>
```

Ejemplo de visualización
en Mozilla Firefox



6. Multimedia

Elementos multimedia

Elementos de audio

- Al igual que con los vídeos, antes de HTML5, para reproducir vídeo en un navegador se empleaba un plug-in como flash.
- Ahora el elemento `<audio>` es la manera estándar de reproducir audio en una página web.
 - El atributo *controls* permite que el vídeo incluya los controles típicos de reproducción.
 - El atributo *autoplay* consigue que la reproducción comience en cuanto se cargue la página.
 - El atributo *Loop* especifica que el audio se reproducirá en bucle (comenzará cada vez que finalice).
 - El atributo *muted*, si está presente, especifica que la salida de audio debería estar silenciada.
- Dentro de el elemento `<audio>` se pueden incluir varios elementos `<source>` permitiendo poner a disposición del navegador el mismo contenido (sería lo lógico) en diferentes formatos para facilitar su preproducción.
 - También se puede incluir un texto que se mostrará si el navegador no conoce el elemento `<audio>`.

AUDIO



4.0



9.0



3.5



10.5



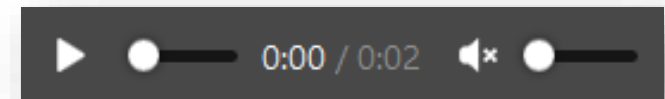
4.0

6. Multimedia

Elementos multimedia

Elementos de audio

```
<audio controls>  
  <source src="horse.ogg" type="audio/ogg">  
  <source src="horse.mp3" type="audio/mpeg">  
  Su navegador no soporta este elemento de audio.  
</audio>
```



Ejemplo de visualización
en Mozilla Firefox

Ejemplo