

Unidad 02. Los lenguajes de marcas en el entorno web (I)

Introducción

El principal objetivo de la web es la distribución de documentos entre personas, con independencia de la plataforma o del sistema operativo que utilicen. La web es, junto con el correo electrónico, uno de los servicios que han hecho de internet la red global que es. Otro ejemplo de servicio de internet es la transferencia de archivos.

2.1. Introducción, evolución y estado actual de HTML

HTML es el lenguaje de marcas más conocido y utilizado. Sus siglas significan “lenguaje de marcado de hipertexto” y con él están escritas la mayoría de las páginas de la WWW o web. El lenguaje HTML y la web son términos que van de la mano, pues de manera independiente no habrían tenido tanta repercusión como conjuntamente.

HTML está gestionado por W3C, consorcio dedicado a la estandarización de la web para que esta sea accesible, fiable, fácil de usar y de uso libre. HTML es, por lo tanto, gratuito.

Existen dos hechos fundamentales que han marcado el destino de HTML:

- Sus destinatarios han sido siempre las personas.
- La información que proporciona es estética y no semántica.

Estas dos características tienen algunas consecuencias. Por ejemplo, si un navegador al procesar el código HTML encuentra algún error de sintaxis, puede intentar solucionarlo con resultado más o menos estéticamente adecuado según el navegador utilizado. Con respecto a la semántica, la aporta el usuario en función de lo que ve y de su buen criterio.

2.2. Estructura y sintaxis del lenguaje HTML.

HTML define el contenido de una página web. Los navegadores interpretan las marcas contenidas en los documentos HTML y representan la información para las personas. Trabajando junto con las hojas de estilo CSS, HTML se ha convertido en un increíble lenguaje de representación de información.

2.2.1. Estructura y sintaxis

Los elementos que forman las páginas HTML están identificados por marcas o etiquetas. Las etiquetas están compuestas por un nombre rodeado de los símbolos “<” y “>”: <etiqueta>

El nombre de la etiqueta puede estar escrito en mayúsculas o minúsculas, aunque por costumbre se emplean letras minúsculas.

Existen dos tipos principales de etiquetas: las de apertura y las de cierre. Un contenido debe estar rodeado por ambas etiquetas. La etiqueta de cierre incluye el símbolo “/” justo antes del nombre:

<etiqueta>Esto es el contenido</etiqueta> elemento típico HTML

Algunas etiquetas no presentan cierre, como por ejemplo la que provoca un salto de línea en el texto:

 elemento vacío

Los elementos que sólo están formados por una etiqueta de apertura se llaman elementos vacíos.

Dentro de un elemento se pueden incluir atributos. Permiten caracterizarlo de cara a identificarlo, representarlo o tratarlo de manera específica. Se incluyen en la etiqueta de apertura y están compuesto por un nombre, el símbolo “=” y el valor del atributo entrecomillados (las comillas pueden ser

simples o dobles, pero se recomienda que en todo el documentos se emplee el mismo tipo):

```
<etiqueta atributo1="valor1" atributo2="valor2">contenido</etiqueta>
```

Los elementos se pueden anidar, es decir, un elemento puede contener a uno o más elementos en su interior:

```
<etiquetanivel1>
  <etiquetanivel2>
    <etiquetanivel3>
      contenido
    </etiquetanivel3>
  </etiquetanivel2>
</etiquetanivel1>
```

Una página HTML está compuesta por un conjunto de elementos y alguna información adicional que se indicará más adelante, que deben respetar una estructura definida, basada en la siguiente:

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

  </head>
  <body>

  </body>
</html>
```

El primer elemento indica el tipo de documento, HTML.

El siguiente elemento <html> </html> es el elemento raíz y dentro de él se encuentra todo el contenido de la página.

El elemento `<head> </head>` incluye información que no aparece en la visualización de la página, pero afecta a cómo va a ser representada.

El elemento `<body> </body>` alberga el contenido visible de la página: texto, tablas, listas, imágenes, elementos multimedia, etc...

Como ejemplo, crea (puedes utilizar cualquier editor de texto como el bloc de notas) el siguiente documento “ejercicio-resuelto-2-1.html” y ábrelo con un navegador.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta charset="utf-8">
    <title>Primer ejemplo HTML</title>
  </head>
  <body>
    Este es el documento básico.
  </body>
</html>
```

Observa como la página abierta por el navegador tiene como título “Primer ejemplo” y como contenido solo la frase “Este es el documento básico”.

Modifica el documento HTML para que muestre otro título y otro texto y ábrelo de nuevo con el navegador.

2.2.2. Entidades (+)

Algunos caracteres tienen un significado especial en HTML, son invisibles o no existen en el lenguaje en el que está escrita la página.

Una entidad en HTML es un conjunto de caracteres que indican al navegador que represente caracteres que por diversas razones, no se

representan correctamente. Comienzan por el símbolo “&” y terminan con el símbolo “;”.

Algunas son:

&	&
<	<
>	>
“	"
’	'
€	€
espacio en blanco	

2.2.3. Comentarios

HTML admite incluir comentarios, no tienen efecto sobre la página ya que su objetivo es proporcionar información a los diseñadores cuando están consultando el código de esta.

El contenido del comentario se delimita con los caracteres “<!--” y “-->”

```
<etiqueta>
.....
<!-- ESTO ES UN COMENTARIO -->
.....
</etiqueta>
```

Ejercicio (+) 2.5. Crea el documento “25comentarios.html” que muestre un título y texto en una web, con algún comentario dentro del código. ¿Qué sucede si ponemos un comentario dentro del contenido de la etiqueta “title”?

Ejercicio (+) 2.6. Visualiza el código de la página web anterior mostrada por el navegador. ¿Aparece exactamente tal como lo hemos escrito? Comenta si notas alguna diferencia.

2.3. Elementos HTML

Un documento HTML está compuesto por elementos organizados de manera jerárquica. El elemento `<html>` es la raíz y todos los demás descienden de él.

Cada elemento se identifica por una etiqueta y se caracteriza por un conjunto de atributos, muchos de los cuales son específicos del elemento concreto.

No obstante, existen atributos comunes o globales a todos los elementos HTML que son determinantes para el correcto funcionamiento de las hojas de estilo CSS o el lenguaje de programación JavaScript.

Algunos atributos globales son:

- `class`: permite agrupar elementos.
- `contenteditable`: el contenido del elemento es editable.
- `hidden`: el elemento no se representa en el navegador.
- `id`: identifica al elemento de manera única.
- `lang`: determina el idioma del contenido del elemento.
- `spellcheck`: determina si el elemento será analizado por el navegador.
- `style`: asigna un estilo al componente.
- `tabindex`: determina el orden de selección al usar el tabulador.
- `title`: permite especificar información extra al elemento.

- translate: determina si el elemento debe ser traducido o no.

Ejemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h2 id="encabezado" class="rojo">Encabezado</h2>
  <p class="rojo" style="background-color: aquamarine;">
    Este párrafo será visible en el navegador
  </p>
  <p id="explicacion" hidden>
    Este párrafo no será visible en el navegador
  </p>
  <p contenteditable spellcheck="true" tabindex="2">
    Este parrafo tiene faltas de ortografia y es editable
  </p>
  <p lang="en" translate="no">
    This paragraph is written in English
  </p>
  <input type="text" tabindex="1">
  <input type="text" tabindex="3">
</body>
</html>
```

Ejercicio (+) 2.7. Crea un documento HTML con el ejemplo anterior "27atributos01.html" y ábrelo con un navegador. Si hay algo que se te ocurra comentar, escríbelo.

2.3.1. Elemento html

Es el elemento principal y raíz del documento HTML. Todo el contenido del documento HTML está incluido dentro de él.

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

```
.....
contenido del documento HTML
.....
</html>
```

2.3.2 Elemento head

Primera de las secciones de una página web. No será visible en la representación por el navegador, con alguna excepción, pero sí tendrá efecto en dicha representación.

El elemento más habitual que se incluye es <title>. Contiene el título de la página que se muestra en la pestaña del navegador.

```
<title>Título del documento</title>
```

El título es también el nombre que se asigna cuando la dirección web se almacena como favorito en el navegador. Además, lo tienen en cuenta los buscadores para clasificar la página.

No es obligatorio, mostrándose el nombre del fichero html en su defecto, información confusa para el usuario final.

En esta sección head se especifican igualmente los metadatos, datos descriptivos que afectan a cómo se interpretan los datos contenidos en el documento.

Se identifican con la etiqueta <meta> seguida de atributos y valores con información que se le quiere proporcionar al navegador. Algunos habituales son:

- charset: codificación del juego de caracteres. El valor estándar es UTF-8, anteriormente era ISO-8859-1.

- http-equiv: especifica una directiva, son nombres de encabezados HTTP.
- name: nombre del metadato, se debe utilizar junto a content.
- content: valor asociado a atributos name y http-equiv.

Ejemplos de utilización:

- Especifica que el tipo de contenido es codificación UTF-8. Debe utilizarse en todas los documentos HTML.

`<meta charset=UTF-8">`

- Indica al navegador que no almacene la página en caché, para que cada vez que se vuelva a consultar se vuelva a descargar del servidor.

`<meta http-equiv="cache-control" content="no-store">`

- Indica al navegador que pasados 3 segundos cargue la página indicada en url.

`<meta http-equiv="refresh" content="3; url=https://www.soymotor.es/">`

- Indica a los robots de los motores de búsqueda que no se indexe la página.

`<meta name="robots" content="noindex"/>`

- Indica palabras clave para que los indexadores de los motores de búsqueda clasifiquen de manera más eficiente la página.

`<meta name="keywords" content="motor, alonso, sainz"/>`

- Indica el nombre del autor de la página.

`<meta name="author" content="Fangio"/>`

- Indica el propietario de los derechos de la página.

```
<meta name="copyright" content="Ediciones Lobato"/>
```

- Determina el ancho de la zona de representación de la página, la escala inicial o las posibilidades de escalado (+).

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

2.3.3 Elemento body

Delimita la sección que contiene los elementos que forman la parte visible de la página web.

```
<body>
.....
    contenido que mostrará la web
.....
</body>
```

El número de etiquetas que se pueden incluir es ingente y se estudiarán las más importantes y usadas. Los elementos que pueden aparecer en esta sección body se clasifican en elementos de bloque y de línea.

2.3.4 Elementos de bloque

Tienen un común características relacionadas con el espacio que ocupan y su distribución en la página. Permiten estructurar y organizar la página para facilitar su correcta interpretación por los programas informáticos.

<address>

Representa información de contacto (correo electrónico, número de teléfono, etc., para el elemento <article> más cercano o para el elemento <body> en el que se encuentra. El texto se muestra directamente en cursiva. Tiene significado semántico. Suele aparecer en <footer>.

`<address>`

Puede contactar con la editorial a través del sitio web `www.soymotor.es`
`</address>`

`<article>`

Identifica los artículos principales del contenido de la página Web, apareciendo normalmente dentro de etiquetas `<section>`. Agrupa información como una unidad independiente, como si fueran las partes de un capítulo de un libro. No genera representación gráfica específica, ya que sirve de orientación semántica. Suele aparecer en `<main>`.

VIDEO TUTORIALES JAVA

Tutoriales de java para principiantes o iniciación. Aquí aprenderemos desde cero a programar en java. Algunos de los conceptos que veremos son genéricos de programación pero los veremos siempre en el entorno de java.

`</article>`

`<article>`

VIDEO TUTORIALES SQL

Tutoriales de sql para principiantes. Aquí aprenderemos como utilizar el lenguaje de consulta SQL.

`</article>`

`<section>`

Indica una parte en general de la página. Esta etiqueta es de alguna forma análoga a dividir un libro en capítulos. Dentro de `<section>` suelen aparecer uno o más elementos `<article>`. Tiene significado semántico. Suele aparecer en `<main>`.

```

<section>
  <article>
    .....
  </article>
  <article>
    .....
  </article>
</section>

```

<aside>

Indica que el contenido dentro de ella está relacionado con el contenido principal de la página pero que no es parte de ella. Es análogo a usar paréntesis para hacer un comentario en un cuerpo de texto (como éste). El contenido entre paréntesis proporciona información adicional sobre el elemento que lo contiene. Es un elemento de orientación semántica, no tiene representación visual. Se usa principalmente para mostrar información complementaria al contenido principal del documento. Suele aparecer en <main>.

```
<h1>El blog de astrofotografía de Gary</h1>
```

```

<p>Hola a todos, y bienvenidos a mi blog...</p>
<aside>
  <nav>
    <h1>Sitios relacionados</h1>
    <ul>
      <li><a href="http://science.nasa.gov/astrophysics/">Astrofísica - NASA
        Science</a></li>
      <li><a href="http://www-astro.physics.ox.ac.uk/">Astrofísica
        Universidad de Oxford</a></li>
      <li><a href="http://aston.blogejemplo.com/">El blog de astrofísica de
        Aston</a></li>
    </ul>
  </nav>
</aside>

```

<blockquote>

Contiene una cita que ocupa todo un párrafo, cita de bloque. Tiene significado semántico. Atributo cite: contiene la referencia a la fuente de la cita, no es visualizada, solo como información en el código. Suele aparecer en <main>.

```
<h1>El poder de la web</h1>
```

```
<blockquote cite="Yoda">
```

El poder de la web está en su universalidad. Acceso para todos. Independientemente de la discapacidad es un aspecto esencial. <cite>Un sabio informático</cite>

```
</blockquote>
```

<details>

Almacena un contenido textual que se hace visible cuando se despliega el componente. Puede incluir el elemento <summary> para representar una etiqueta que identifique el contenido. Tiene significado semántico.

Algunos navegadores puede que no lo interpreten correctamente.

```
<details>
```

```
<summary>Nota para el usuario</summary>
```

El contenido de este manual está actualizado a la versión 10.8 del producto.

```
</details>
```

Ejercicio 2.12. Escribe un documento html llamado "212metabloque01.html" que incluya los siguientes elementos:

- *Lenguaje español.*
- *Codificación de caracteres "UTF-8"*
- *Título de la página web "Mis primeros metas y bloques".*

- Comentario “Estoy empezando con HTML, paciencia”.
- La página no debe almacenarse en caché.
- Indicar a los robots que los motores de búsqueda no indexen la página.
- Autor de la página “Luke Skywalker”.
- Información de contacto “Llame al 66 para nueva orden imperial”.
- Cita sin referencia “(1) Ejecuten la orden 66”.
- Contenido de la página web “Esta página web es un ejemplo parcial de metadatos y elementos de bloque en star wars (1)”.
- Si es necesario, modifica el orden de los elementos para que tengan sentido semántico para el usuario final.

Ejercicio 2.13. Escribe un documento html llamado “213metabloque02.html” que contenga los siguientes elementos:

- Lenguaje español.
- Codificación de caracteres “UTF-8”.
- Contenido del documento html es texto HTML.
- Indica al navegador que pasados 10 segundos cargue la página “https://www.starwars.com/”
- Palabras claves para indexar la página: “r2d2, c3po”.
- Propietario de la página “Imperio galáctico”.
- Noticia sobre los rebeldes “El último ataque de los terroristas rebeldes ha sido rechazado con gran éxito, 20 x-wing destruidos y 15 pilotos capturados.”
- Contenido textual desplegable sin etiqueta que identifique el contenido “Año 22 de la caída de la odiosa república”.
- Contenido de la página web “Esta página web es otro ejemplo parcial de metadatos y elementos de bloque”.
- Si es necesario modifica el orden de los elementos para que tengan sentido semántico para el usuario final.

Ejercicio 2.14. Escribe un documento html llamado “214metabloque03.html” que contenga los siguientes elementos:

- *Comentarios en al menos dos partes del documento.*
- *Título “Vamos a poner casi todo”.*
- *Codificación “UTF-8”.*
- *Indicar que la página no se almacene en caché.*
- *Indicar que pasados 60 segundos se cargue la página web del instituto.*
- *Indicar que los motores de búsqueda no indexen la página.*
- *Indicar como palabras claves “raro, prueba, curioso”.*
- *Autor de la página el nombre del alumno o alumna.*
- *Propietario de los derechos de la página “Alberti S.A.”.*
- *Información de contacto “Acuda de 9 a 10 a secretaría para dudas.”*
- *Comentario de foro “El instituto huele muy bien y es luminoso, un 10.”*
- *Contenido de la página web patrocinado por “Zona Franca de Cádiz”.*
- *Cita “El poder del amor lo encontramos en la poesía de Alberti”. Referencia de la cita “Admirador de Alberti”.*
- *Texto que se despliega “Esto es un ejemplo bastante completo de lo que hemos aprendido hasta ahora”. Etiqueta que identifica el contenido “Nota para los interesados”.*
- *Contenido de la página web “Esta página web es un ejemplo muy completo de metadatos y elementos de bloque”.*
- *Si es necesario modifica el orden de los elementos para que tengan sentido semántico para el usuario final.*

<p>

Delimita un párrafo de texto. Tiene significado semántico.

<pre>

Delimita un párrafo de texto representándolo tal y como está en el documento HTML.

Ejemplo:

```
<p>
    Texto dentro de un párrafo.
    Se ignoran    saltos de línea    y demás.
</p>
<pre>
    Texto dentro de un pre.
    Se respetan    saltos de línea    y demás.
</pre>
```

Resultado:

Texto dentro de un párrafo. Se ignoran saltos de línea y demás.

Texto dentro de un pre.

Se respetan saltos de línea y demás.

<h1>, <h2>, <h3>, <h4>, <h5>, <h6>

Representan elementos de encabezado. El elemento <h1> es el más importante (tamaño más grande) y el <h6>, el menos importante (tamaño más pequeño).

Utilizar estos elementos mejora la visibilidad de la página en los buscadores y el posicionamiento (SOE). Tiene significado semántico.

Ejemplo:

```
<h1>Encabezado de nivel 1</h1>
```

Resultado:

Encabezado de nivel 1

Ejemplo:

```
<h6>Encabezado de nivel 6</h6>
```

Resultado:

Encabezado de nivel 6

<hr>

Representa un cambio de párrafo mediante una línea. Tiene significado semántico.

Ejemplo:

```
<p>Este texto está en un párrafo.</p>  
<hr>  
<p>Este texto está en otro párrafo.</p>
```

Resultado:

Este texto está en un párrafo.

Este texto está en otro párrafo.

<dialog>

Muestra una caja de diálogo. Tiene que aparecer el atributo open para que se muestre. Su uso va muy unido a CSS, teniendo un comportamiento

extraño sin usar hojas de estilo, por ejemplo no suele aparecer donde se desea. Tiene significado semántico.

Ejemplo:

```
<dialog open>  
  <p>¡La matrícula se ha realizado correctamente!</p>  
</dialog>
```

Resultado:

¡La matrícula se ha realizado
correctamente!

Ejercicio 2.15. Escribe un documento HTML “215parrafos01.html” según los siguientes puntos:

- *Deben aparecer los 6 tipos de encabezado.*
- *Después de cada encabezado un texto de no más de 3 líneas que explique el tema del encabezado.*
- *En los encabezados impares se utilizará <p>.*
- *En los encabezados pares se utilizará <pre>.*
- *Una caja de diálogo con el texto ¡Por fin puedo organizar la web!.*
- *Antes de la caja de diálogo, una línea que la separe claramente del resto del texto.*

Ejercicio 2.16. Escribe un documento HTML “216parrafos02.html” según los siguientes puntos:

- *La página web debe aparecer lo mejor organizada posible.*

- *Lenguaje html español.*
- *Codificación “UTF-8”.*
- *Comentarios en al menos 2 partes del documento.*
- *Al menos 1 entidad.*
- *Título “Los párrafos”*
- *Indicar que los motores de búsqueda no indexen la página.*
- *Autor de la página el nombre del alumno o alumna.*
- *Información de contacto “Llame al teléfono 077 y no se lo cogerán nunca”. Debe ser la última información de la web.*
- *Página web patrocinada por “Sal gorda de Cádiz”. Debe aparecer como primera información en la web.*
- *Cita con el texto “Un gran camino empieza con un sencillo paso”.*
- *Texto desplegable “Los párrafos y encabezados permiten organizar mucho mejor la información de las páginas web (1)”. Etiqueta que identifica el contenido “Para saber más”. Justo después de los tres encabezados con texto.*
- *Tres tipos de encabezados con texto que le acompañe. No más de tres líneas de texto por párrafo. Justo después del patrocinio de la web.*
- *Una línea debe separar el texto normal de la web de las secciones:*
 - *Cita.*
 - *Información de contacto.*

<div>

Permite crear divisiones y agrupar contenidos. Se utiliza, junto con las hojas de estilo CSS, para maquetar y estructurar páginas. No tiene representación visual. A partir de HTML 5 se recomienda no utilizar salvo que sea imprescindible.

<p>El idioma por defecto de esta web es español, pero deseamos

insertar un texto en francés y queremos que no tenga problemas de visualización</p>

```
<div lang="fr">
```

```
<p>Dans le numéro de novembre de Marie Claire, une erreur s'est  
glissée dans la rubrique Mode Actualités.</p>
```

```
</div>
```

```
<p>Fuera de la etiqueta div el idioma vuelve a ser el español. </p>
```

<dl>, <dt> y <dd>

Permiten definir una lista de descripciones. <dl> marca la lista, <dt> cada uno de los términos definidos y <dd> la definición de cada término. Tienen significado semántico.

```
<h2>Metalenguajes</h2>
```

```
<dl>
```

```
<dt>SGML</dt>
```

```
<dd>Metalenguaje para la definición de otros lenguajes de  
marcado</dd>
```

```
<dt>XML</dt>
```

```
<dd>Lenguaje basado en SGML y que se emplea para describir  
datos</dd>
```

```
<dt>XUL</dt>
```

```
<dd>Lenguajes derivados de XML para determinadas  
aplicaciones</dd>
```

```
</dl>
```

Resultado:

Metalenguajes

SGML

Metalenguaje para la definición de otros lenguajes de marcado

XML

Lenguaje basado en SGML y que se emplea para describir datos

XUL

Lenguajes derivados de XML para determinadas aplicaciones

, ,

Representan una lista y sus elementos. La etiqueta identifica cada uno de los elementos de la lista. Las etiquetas y identifican una lista ordenada (mediante números) y una desordenada (mediante viñetas). Las listas se pueden anidar. Tienen significado semántico.

```
<ol>
```

```
<li>La libertad de usar el programa, con cualquier propósito.</li>
```

```
<li>La libertad de estudiar cómo funciona el programa y adaptarlo a  
tus necesidades.</li>
```

```
<li>La libertad de distribuir copias del programa.</li>
```

```
<li>La libertad de mejorar el programa y hacer públicas las mejoras a  
los demás.</li>
```

```
</ol>
```

Resultado lista ordenada

1. La libertad de usar el programa, con cualquier propósito.
2. La libertad de estudiar cómo funciona el programa y adaptarlo a tus necesidades.
3. La libertad de distribuir copias del programa.
4. La libertad de mejorar el programa y hacer públicas las mejoras a los demás.

```
<ul>
```

```
<li>La libertad de usar el programa, con cualquier propósito.</li>
```

```
<li>La libertad de estudiar cómo funciona el programa y adaptarlo a  
tus necesidades.</li>
```

```
<li>La libertad de distribuir copias del programa.</li>
```

```
<li>La libertad de mejorar el programa y hacer públicas las mejoras a
    los demás.</li>
</ul>
```

Resultado lista no ordenada:

- La libertad de usar el programa, con cualquier propósito.
- La libertad de estudiar cómo funciona el programa y adaptarlo a tus necesidades.
- La libertad de distribuir copias del programa.
- La libertad de mejorar el programa y hacer públicas las mejoras a los demás.

La utilización del atributo align no funciona óptimamente, alinea el texto pero ignora la numeración y el símbolo de las no numeradas. Se soluciona con CSS.

<figure> y <figcaption>

Representan una ilustración, figura, diagrama, etc., y su leyenda. Tienen significado semántico.

```
<figure>
  
  <figcaption>El logotipo del World Wide Web Consortium (W3C)
</figcaption>
</figure>
```

<nav>

Este contenedor alberga enlaces de navegación. Habitualmente se muestran tablas de contenidos, índices, menús o sistemas de navegación alternativos. No tiene visualización, sólo objetivos semánticos en el documento HTML. Suele aparecer en <header> y <main>.

```
<nav>
  <ul>
    <li>Tutoriales Web-Html</li>
    <li>Tutoriales Juegos Android</li>
    <li>Tutoriales Java para principiantes</li>
    <li>Tutoriales Sql</li>
    <li>Contacto</li>
  </ul>
</nav>
```

<footer>

Representa un pie de documento o sección. Suele contener información sobre la propiedad, autoría, firmas, información acerca del autor, licencias, documentos relacionados, derechos del documento, etc. Dentro de <footer> deben aparecer las etiquetas que hacen referencia a esa información. Tiene significado semántico.

```
<footer>
  <address>
    argust email: argust@star.com
  </address>
</footer>
```

<header>

Contiene el encabezado de un documento o de una sección. En este elemento, de naturaleza semántica, se colocan habitualmente los encabezados, los vínculos de navegación (menú de navegación de la página web, enlaces a partes de la propia página web), los formularios de búsqueda, los logos, las tablas de contenidos, las introducciones, etc. Tiene significado semántico.

```
<header>
```

```

<h1>El nombre de mi web</h1>

<nav>
  <ul>
    <li>Enlace menu 1</li>
    <li>Enlace menu 2</li>
  </ul>
</nav>
</header>

```

<main>

Representa el contenido principal del documento. Este contenido central puede interpretarse como todo lo que resta luego de quitar anuncios, encabezados (header), pies (footer), navegación (nav), vínculos relacionados y otros elementos secundarios.

No tiene visualización, sólo para uso semántico, pues identifica de manera inequívoca el bloque de información principal de la página.

Normalmente <main> está dividido en varias <section>.

Debe incluirse un único contenedor <main> dentro de un documento HTML y no debe figurar dentro de <article>, <aside>, <footer>, <header>, <section> ni <nav>, entre otras etiquetas.

```

<html>
  <head>
    <title>Artículo: Software libre</title>
  </head>
  <body>
    <header>
      <p>¡Bienvenido a Artículos Locos!</p>
    <nav>

```



```

        <a href="/">Inicio</a>
        <a href="/articulos.html">Artículos</a>
        <a href="/contacto.php">Contáctanos</a>
    </nav>
</header>
<main>
    <h1>Software libre</h1>
    <p>El término software libre refiere el conjunto de software
que por elección
    manifiesta de su autor, puede ser copiado, estudiado,
modificado, utilizado
    libremente con cualquier fin y redistribuido con o sin
cambios o mejoras.</p>
</main>
<footer>
    <p>&copy; Todos los derechos reservados</p>
</footer>
</body>
</html>

```

Ejercicio 2.17. Crea un documento html llamado “217bloques01.html” con el ejemplo anterior y ábrelo con un navegador. Añade contenido que hemos considerado como convenientes aunque no sean obligatorias.

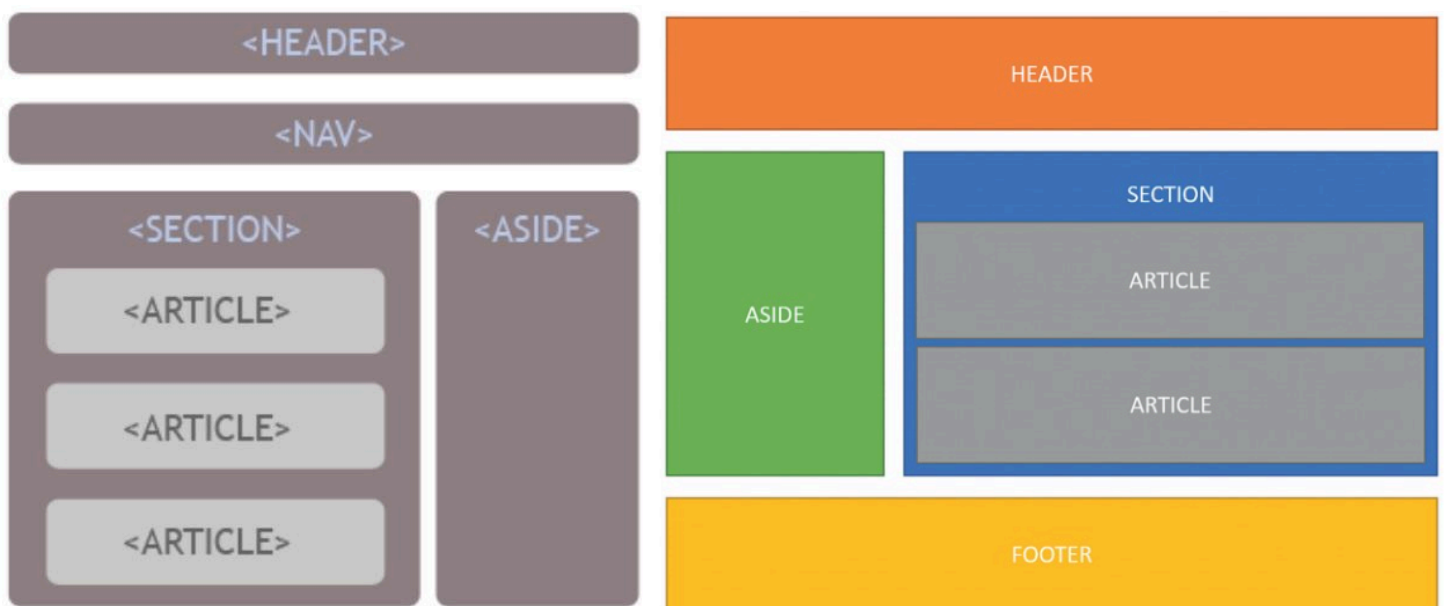
Ejercicio (+) 2.18. W3C define una serie de pautas para garantizar que una página es accesible a personas con discapacidad. Busca información sobre ellas y escribe las principales que toda web debería cumplir en tu opinión.

Ejercicio 2.19. Crea un documento llamado “219bloques02.html” que contenga los siguientes elementos, utilizados correctamente según su sentido semántico.

- Título “Web no semántica”.

- *Autor: nombre de cada uno.*
- *Un contenedor <div> para indicar que el texto Star Wars está en inglés.*
- *Una lista ordenada.*
- *Una lista desordenada.*
- *Una foto de Palpatine y su leyenda “¿Clon del emperador?”.*
- *Una <section> y en su interior un <article> y una lista de descripciones.*

Alguno de los esquemas más utilizados para el diseño de páginas web, que debe reflejarse en el documento HTML, serían los siguientes:



De esta forma, el documento HTML reflejará también la información semántica que se visualiza en la página web.

```
<!DOCTYPE html>
<html lang="es">
  <head>
    contenido .....
  </head>
  <body>
    <header>
      contenido que deseamos incluir en la cabecera de la web
```

```
</header>
<nav>
    contenido de menú, enlace ....
<aside>
    contenido que deseamos incluir en la zona lateral de la web
</aside>
<main>
    contenido principal con la información sobre el tema de la web
    <section>
        <article>
            contenido .....
        </article>
        <article>
            contenido .....
        </article>
        ....
    </section>
    <section>
        <article>
            contenido .....
        </article>
        <article>
            contenido .....
        </article>
        ....
    </section>
</main>
<footer>
    contenido que deseamos incluir en el pie de la web
</footer>
</body>
</html>
```

Ejercicio (+) 2.20. Teniendo en cuenta todas las etiquetas vistas hasta ahora, indica su o sus lugares semánticos correctos (algunas etiquetas es correcto que se puedan usar en varios sitios) según deban aparecer en <header></header>, <main></main>, <footer></footer> o fuera de ellos. Intercambia tu opinión con tus compañeros cercanos para decidir lo más conveniente según HTML 5. Puedes pedir ayuda a la web si lo consideras necesario, pero sin abusar, intenta deducirlo según la costumbre humana.

Ejercicio 2.22. Escribe el documento “222atributos01.html” en el que uses las etiquetas y atributos del ejercicio anterior.

Ejercicio 2.23. Escribe el documento “223completo01.html” donde aparezcan las etiquetas contenedoras <header>, <main> y <footer>. Dentro de ellas debes poner una etiqueta que les corresponda. Diseña la web para que su contenido semántico sea correcto, tema de la web libre.

Ejercicio 2.24. Escribe el documento “224completo02.html” que incluya prácticamente todo lo estudiado hasta el momento (salvo lo no recomendado ya para HTML 5). Diseña la web para que su contenido semántico sea correcto, tema de la web libre.

2.3.5. Elementos de línea(sin revisar)

Se integran entre los elementos anterior y posterior sin introducir saltos de línea.

<a>

Representa un enlace a una URL. Al pulsar sobre el contenido marcado con esta etiqueta, el navegador se dirigirá a la url indicada (página web) o realizará la acción correspondiente (por ejemplo, si es una dirección de correo electrónico, abrirá el gestor de correos del ordenador). Tiene significado semántico.

Salvo que se indique lo contrario, y siempre que tenga el atributo href, el enlace aparecerá subrayado y en un color predeterminado.

Los enlaces se pueden clasificar en:

- Externos: referencian a un recurso ubicado en un servidor distinto del alojamiento de la página HTML que contiene la etiqueta <a>.
- Internos: referencian a un recurso ubicado en el mismo servidor de alojamiento de la página HTML que contiene la etiqueta <a>.
- Internos y externos a elementos concretos: referencian al identificador de un elemento dentro del propio documento HTML (interno) o de un documento HTML distinto (externo). Utiliza el símbolo # para identificar al elemento referenciado .

Atributos significativos:

- download: descarga el recurso al que apunta la URL en lugar de abrirlo en el navegador. Depende su funcionamiento de cada navegador.
- href: contiene la URL a la que apunta el enlace. Imprescindible si se desea que el enlace se muestre como tal. Puede ser una página HTML, una imagen, un PDF, un fichero de audio, etc... Puede hacer referencia a un correo electrónico, añadiendo mailto: o un número de teléfono, añadiendo tel:.
- hreflang: indica el idioma de la página destino del enlace.
- ping: se suele utilizar para registrar la actividad del usuario.
- rel: determina la relación entre el enlace y el recurso al que hace referencia.

- target: permite indicar dónde abrir lo referenciado.
 - _self: en la misma pestaña.
 - _blank: en una nueva pestaña del navegador.
 - _parent: en el contexto de navegación padre del actual.
 - _top: en el contexto más alto de navegación.

```
<p>
Una olla de algo más vaca que <a href="https://dle.rae.es/carnero" target="_
blank" type="text/html">carnero</a>, salpicón las más noches...
</p>
<p>
Escribir un email a <a href="mailto:cervantes@paraninfo.es">Miguel de
Cervantes</a>
</p>
```

Una olla de algo más vaca que [carnero](https://dle.rae.es/carnero), salpicón las más noches...

Escribir un email a [Miguel de Cervantes](mailto:cervantes@paraninfo.es)

```
<ul>
  <li><a href="#capitulo1">Sección 1</a></li>
  <li><a href="#capitulo2">Sección 2</a></li>
  <li><a href="#capitulo3">Sección 31</a></li>
</ul>
<article id="capitulo1">
  <!-- CONTENIDO DEL CAPÍTULO 1-->
</article>
<article id="capitulo2">
  <!-- CONTENIDO DEL CAPÍTULO 2-->
</article>
<article id="capitulo3">
  <!-- CONTENIDO DEL CAPÍTULO 3-->
</article>
```

Ejercicio 2.25. Crea un documento “225enlaces01.html” que contenga los siguientes enlaces:

- *2 enlaces a páginas de tu interés, el primer enlace debe abrirse en la misma pestaña, el segundo en una pestaña nueva.*
- *3 enlaces a otras partes del documento (no olvides el atributo id).*
- *1 enlace a una cuenta de correo.*

<abbr>

Indica que el texto marcado es abreviatura o acrónimo. Se representa subrayado con una línea de puntos. Tiene significado semántico.

Atributo title: contiene la descripción completa de la abreviatura, al pasar el ratón sobre la abreviatura se muestra el contenido de title.

De todos los videojuegos, el que más me gusta es el <abbr title="Call of Duty">COD</abbr>

De todos los videojuegos, el que más me gusta es el COD

Indica que el texto se mostrará en negrita. Debe utilizarse para llamar la atención de un texto sin mayor importancia. No tiene importancia semántica.

<p>Pulsa Ctrl+C para copiar al portapapeles.</p>

Pulsa **Ctrl+C** para copiar al portapapeles.

**
**

```

<p>
  "Sí, repite la hormiga,<br>
  he visto las estrellas.<br>
  subí al árbol más alto<br>
  que tiene la alameda<br>
  y vi miles de ojos<br>
  dentro de mis tinieblas."<br>
</p>
<p>
  Extracto del poema 'Los encuentros de un caracol aventurero', de Federico
  García Lorca.
</p>

```

"Sí, repite la hormiga,
he visto las estrellas.
subí al árbol más alto
que tiene la alameda
y vi miles de ojos
dentro de mis tinieblas."

Produce un salto de línea. Tiene significado semántico.

<cite>

Indica una referencia a un autor o fuente de información. El texto se muestra en cursiva.

<p>Según el genial pintor <cite>Picasso </cite>"La inspiración existe, pero tiene que encontrarte trabajando".</p>

Resultado:

Según el genial pintor *Picasso* "La inspiración existe, pero tiene que encontrarte trabajando".

<code>

Indica que el texto contenido es código fuente de un programa informático, utilizando un tipo de letra de ancho fijo (monoespaciada). Tiene significado semántico.

<p>En Java con <code>Math.random()</code>, podemos generar un número aleatorio.</p>

Resultado:

En Java, con `Math.random()`, podemos generar un número aleatorio.

Permite indicar que el texto ha sido borrado, mostrándose tachado. Tiene significado semántico.

Presenta los atributos:

- cite: contiene la URL donde se explica la razón del cambio.
- datetime: fecha u hora del cambio.

<p>El lenguaje de programación más utilizado es
 <del datetime="2020-03-04 10:00:00"
 cite="estudioLenguajes.html">Java Python
 </p>

Resultado:

El lenguaje de programación más utilizado es ~~Java~~ Python

<dfn>

Dentro de elementos <p>, <section> o <dl> que definen un término, <dfn> identifica dicho término en letra cursiva para señalarlo sobre el texto de la definición. Tiene significado semántico.

<p>Los <dfn>videojuegos de plataformas</dfn> son muy dinámicos para conseguir

completar el objetivo del juego.</p>

Resultado:

Los *videojuegos de plataformas* son muy dinámicos para conseguir completar el objetivo del juego.

2.26. Crear un documento HTML llamado “226breve01.html” en el que aparezcan correctamente utilizados los elementos <abbr>, ,
, <cite>, <code>, , <dfn>.

Permite enfatizar un texto, mostrándolo en cursiva. Tiene como significado semántico que el autor está poniendo énfasis en lo marcado por .

<p>Te dije que la cita era mañana, no hoy.</p>

Resultado:

Te dije que la cita era *mañana*, no hoy.

<i>

El texto delimitado aparece en cursiva, pues es diferente por alguna razón indeterminada. No tiene significado semántico.

<p>En el Parque Nacional anidan varios ejemplares de <i>Aegyptius monachus</i>,
el ave de mayor envergadura.</p>

Resultado:

En el Parque Nacional anidan varios ejemplares de *Aegypius monachus*, el ave de mayor envergadura.

<ins>

Indica que el contenido ha sido añadido a una versión anterior del documento HTML. Aparece subrayado. Tiene significado semántico.

Atributos:

- cite: contiene la URL donde se explica la razón del cambio.
- datetime: fecha u hora del cambio.

<p>Los lenguajes de programación mejores son Java, Python y
<ins>Kotlin</ins>.</p>

Resultado:

Los lenguajes de programación mejores son Java, Python y Kotlin.

<kbd>

Indica que el texto contenido debe introducirlo el usuario. La representación depende del navegador.

<p>Por favor, pulse <kbd>Ctrl+F1</kbd> para reiniciar el simulador.</p>

Resultado:

Por favor, pulse Ctrl+F1 para reiniciar el simulador.

<mark>

Resalta un texto en color amarillo. Tiene significado semántico.

Ejemplo:

<p>Rafael tiene alergia a los <mark>tomates</mark>.</p>

Resultado:

Rafael tiene alergia a los tomates.

<q>

Indica una cita literal. El contenido suele aparecer entrecomillado. Tiene significado semántico.

El atributo cite contiene la referencia a la fuente de la cita, no es visualizada, solo como información en el código.

<p>Como indica la Wikipedia, <q cite="https://es.wikipedia.org/wiki/HTML">HTML es un lenguaje de marcado.</q></p>

Resultado:

Como indica la Wikipedia "HTML es un lenguaje de marcado."

2.27. Crear un documento HTML llamado "227breve02.html" en el que aparezcan correctamente utilizados los elementos , <i>, <ins>, <kbd>, <mark> y <q>.

<s>

Muestra el texto tachado con una línea horizontal, indicando que ya no está vigente, disponible o es incorrecto.

<p><s>Nintendo Game Boy Color por 12.990 pesetas</s></p>

~~Nintendo Game Boy Color por 12.990 pesetas~~

<samp>

Indica que el texto marcado es un mensaje proporcionado por el ordenador. Los navegadores suelen utilizar una fuente monoespaciada, para diferenciarlo del texto normal. Se suele utilizar con el elemento `<pre>` para que se

```
<p>
Si introduce la dirección de manera incorrecta, el sistema mostrará la
siguiente información:
<pre><samp>
    Ha introducido una dirección incorrecta.
    Pulse F10 para reiniciar el proceso.
</samp></pre>
</p>
```

Si introduce la dirección de manera incorrecta, el sistema mostrará la siguiente información:

```
Ha introducido una dirección incorrecta.
Pulse F10 para reiniciar el proceso.
```

<small>

Representa un comentario al margen, y reduce el tamaño del texto etiquetado. Se suele utilizar para mostrar avisos, atribuciones, derechos de propiedad, textos legales, etcétera. Tiene significado semántico.

Título: Cloud Computing, tecnología y negocio

`<hr>`

`<small>`Este contenido es propiedad de ediciones Palote`</small>`

Resultado:

Título: Cloud Computing, tecnología y negocio

Este contenido es propiedad de ediciones Palote

Muestra el texto en negrita. Debe utilizarse cuando el texto tiene una fuerte importancia o un sentido de urgencia o seriedad. Tiene importancia semántica.

<p>Para prevenir una explosión es importante humedecer la pólvora</p>

Resultado:

Para prevenir una **explosión** es importante humedecer la pólvora

<sub> y sup>

Permiten mostrar subíndices y superíndices con importancia semántica.

```
<p>Para igualar la velocidad a  $x^{2}$ , la aceleración debe ser constante entre  $T_0$  y  $T_1$ .</p>
```

Resultado:

Para igualar la velocidad a x^2 , la aceleración debe ser constante entre T_1 y T_2 .

<var>

Indica que el texto marcado es el nombre de una variable de una expresión matemática o un programa informático. Aparece en cursiva.

Ejemplo:

<p>En este punto hay que modificar el valor de la variable <var>Sumandos</var> para obtener el resultado correcto</p>

Resultado:

En este punto hay que modificar la variable *Sumandos* para obtener el resultado correcto

2.28. Crear un documento HTML llamado “228breve03.html” en el que aparezcan correctamente utilizados los elementos `<samp>`, `<small>`, ``, `<sub>`, `<sup>` y `<var>`.

2.29. Crea un documento HTML a tu elección “229completo02.html” en el que aparezcan todos los elementos de línea que aparecen en el punto 2.3.5.

2.30. Elabora una lista de todas las etiquetas estudiadas indicando si tienen significado semántico o no.

2.31. Crea un documento HTML a tu elección “231todas01.html” en el que uses todas, todas, todas las etiquetas vistas desde principio de curso, tanto para `<head>` como para `<body>`. Utiliza también tantos atributos como puedas. Ten en cuenta siempre el significado semántico de las etiquetas. Debe ser un documento HTML totalmente nuevo, sin reutilizar los anteriores.

2.4. Tablas en HTML: estructura y elementos

Permite mostrar información organizada en filas y columnas. En el pasado se utilizaban, además, para la maquetación de las páginas, pero es una práctica absolutamente desaconsejada.

2.4.1. Elementos de las tablas

`<table>`

Delimita una tabla, todo el contenido de la tabla debe ir entre `<table>` y `</table>`. Tiene significado semántico.

Excepcionalmente usaremos el atributo `border="número de borde"`, aunque HTML 5 no aconseja utilizarlo, pues el estilo de las tablas se define con las hojas de estilo, todavía no estudiadas.

```
<table border="1">
```

<caption>

Proporciona un título descriptivo a la tabla. Declarado una sola vez al comienzo de la tabla, se muestra, por defecto, en la parte superior de la tabla.

<thead>

Contiene el conjunto de columnas que forman la fila del encabezado de una tabla. Solo debe incluirse una vez y tiene que ubicarse inmediatamente antes del cuerpo (elemento <tbody>).

<tbody>

Delimita el cuerpo de una tabla, las filas que contienen los datos. Una tabla puede contener varios <tbody> que formen diferentes grupos de datos.

<tfoot>

Delimita el conjunto de filas que forman el pie de una tabla. Se suelen mostrar los acumulados de las columnas o resúmenes de los datos mostrados en el elemento <tbody>.

Ejemplo:

```
<table>
  <caption>Gastos anuales</caption>
  <thead>
    .....
  </thead>
  <tbody>
    .....
  </tbody>
  <tfoot>
    .....
```


`</tfoot>`
`</table>`

`<tr>`

Representa una fila de una tabla. Es el elemento contenedor de celdas de la tabla, ya sean de encabezado, cuerpo o pie.

`<th>`

Delimita una celda del encabezado de la tabla, suelen ubicarse en elementos `<tr>`. Los textos se muestran en negrita.

`<td>`

Representa una celda del cuerpo de la tabla, suelen ubicarse en elementos `<tr>`.

```

<table>
  <caption>Gastos anuales</caption>
  <thead>
    <tr>
      <th>Concepto</th>
      <th>Primer semestre</th>
      <th>Segundo semestre</th>
      <th>Total</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <th>Total</th>
      <th>2650</th>
      <th>2080</th>
      <th>4730</th>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>Alimentación</td>
      <td>1500</td>
      <td>1750</td>
      <td>3250</td>
    </tr>
    <tr>
      <td>Tecnología</td>
      <td>850</td>
      <td>30</td>
      <td>880</td>
    </tr>
    <tr>
      <td>Deporte</td>
      <td>300</td>
      <td>300</td>
      <td>600</td>
    </tr>
  </tbody>
</table>

```

Resultado:

Gastos anuales			
Concepto	Primer semestre	Segundo semestre	Total
Alimentación	1500	1750	3250
Tecnología	850	30	880
Deporte	300	300	600
Total	2650	2080	4730

Ejercicio resuelto:

Crear una página HTML que contenga una tabla que muestre, para las consolas PS1, PS2, PS3, PS4 Y PS5, el año de lanzamiento y el año de retirada.

Solución

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <title>Consolas</title>
</head>
<body>
  <table>
    <thead>
      <tr>
        <th>Consola</th>
        <th>Año de lanzamiento</th>
        <th>Año de retirada</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>PlayStation 1</td>
        <td>1994</td>
```

```
        <td>2000</td>
      </tr>
      <tr>
        <td>PlayStation 2</td>
        <td>2000</td>
        <td>2004</td>
      </tr>
      <tr>
        <td>PlayStation 3</td>
        <td>2006</td>
        <td>2010</td>
      </tr>
      <tr>
        <td>PS4</td>
        <td>2013</td>
```

```
        <td>2017</td>
      </tr>
      <tr>
        <td>PS5</td>
        <td>2020</td>
        <td>En venta</td>
      </tr>
    </tbody>
  </table>
</body>
</html>
```

2.32. Crea un documento HTML “232tabla01.html” con el ejemplo anterior y prueba su funcionamiento. Añade un título descriptivo a la tabla. Si es necesario utiliza el atributo border.

2.33. Crea un documento HTML “233tabla02.html” con la misma información que el ejercicio anterior añadiendo los datos de la familia XBOX. Ponle un título descriptivo a la tabla. Si es necesario utiliza el atributo border.

2.34. Crea un documento HTML “234tabla03.html” que contenga una tabla compuesta por una fila de encabezado, 3 filas de datos y 6 columnas, por último un pie de tabla. La primera columna especifica el año. Cada columna de datos se corresponde con un trimestre del año, inventa valores para las 3 filas de datos, la última columna mostrará el sumario del año completo. El encabezado debe indicar textos que indiquen el contenido de cada columna. El pie de tabla debe contener la suma de las 4 columnas trimestrales y del total de los tres años. Si es necesario utiliza el atributo border. Tiene que ser un ejercicio original, no basado en alguno anterior.

Atributos:

- colspan: número de celdas consecutivas hacia la derecha que debe agrupar.
- rowspan: número de celdas consecutivas hacia abajo que debe agrupar.

Ejemplo de tabla con colspan y rowspan:

```
<table>
  <tr>
    <td>Columna 1</td>
    <td>Columna 2</td>
    <td>Columna 3</td>
    <td>Columna 4</td>
  </tr>
  <tr>
    <td colspan="2">F1 C1, C2</td>
    <td rowspan="2">F1, F2 C3</td>
    <td>F1 C4</td>
  </tr>
  <tr>
    <td>F2 C1</td>
    <td>F2 C2</td>
    <td>F2 C4</td>
  </tr>
</table>
```

Resultado:

Columna 1	Columna 2	Columna 3	Columna 4
F1 C1, C2		F1, F2 C3	F1 C4
F2 C1	F2 C2		F2 C4

2.35. Crea un documento HTML “235tabla04.html” que haga uso del atributo `colspan` de forma distinta en dos tablas separadas. Si es necesario utiliza el atributo `border`. Tiene que ser un ejercicio original, no basado en alguno anterior.

2.36. Crea un documento HTML “236tabla05.html” que haga uso del atributo `rowspan` de forma distinta en dos tablas separadas. Si es necesario utiliza el atributo `border`. Tiene que ser un ejercicio original, no basado en alguno anterior.

2.37. Crea un documento HTML “237tabla06.html” que haga uso de los atributos `colspan` y `rowspan` de forma distinta en dos tablas separadas. Si es necesario utiliza el atributo `border`. Tiene que ser un ejercicio original, no basado en alguno anterior.

<colgroup>

Permite crear grupos de una o más columnas. También permite asignar atributos comunes a todas las columnas agrupadas. Debe ser puesto al comienzo de la definición de la tabla.

La agrupación se realiza de izquierda a derecha de manera consecutiva utilizando el atributo span o agregando elementos <col> -o span o <col>, no ambos-.

Tiene significado semántico, si solo se desea un objetivo de presentación visual, debemos utilizar <col> para visualizar las agrupaciones.

Atributo significativo:

- span: determina el número de columnas agrupadas.

Ejemplo:

Si añadimos el siguiente código al ejemplo de la tabla de “Gastos Anuales”

```
<table>
    <colgroup          span="2"          style="background-color:
bisque;"></colgroup>
    <colgroup          span="2"          style="background-color:
greenyellow;"></colgroup>
    <caption>Gastos anuales</caption>
    .....
    .....
```

Resultado:

Gastos anuales			
Concepto	Primer semestre	Segundo semestre	Total
Alimentación	1500	1750	3250
Tecnología	850	30	880
Deporte	300	300	600
Total	2650	2080	4730

Nota: debemos recordar que el uso de CSS permitirá modificar el aspecto de las tablas.

2.38. Crea un documento HTML “238tabla07.html” con el ejemplo anterior y comprueba que funciona correctamente.

2.39. Modifica el ejercicio 2.34 y llámalo “239tabla08.html” para que muestre la tabla de la siguiente forma:

Año	1.Trimestre	2.Trimestre	3.Trimestre	4.Trimestre	Total año
2010	500	600	400	-100	1400
2011	700	800	-200	100	1200
2012	-300	-400	100	200	-400
Totales	300	1000	300	200	2200

<col>

Permite indicar algunas propiedades a las columnas o grupos de columnas de una tabla. Desde HTML 5, los elementos <col> deben estar dentro de un elemento <colgroup>.

Atributo significativo:

- span: determina el número de columnas agrupadas visualmente.

El objetivo de <col> es destacar visualmente, sin ningún tipo de significado semántico.

Ejemplo:

Si añadimos el siguiente código al ejemplo de la tabla de “Gastos Anuales”

```
<table>
  <colgroup>
    <col>
    <col span="2" style="background-color:khaki;">
  </colgroup>
  <caption>Gastos anuales</caption>
```

Resultado:

2.40. Modifica el documento HTML “238tabla07.html” con el ejemplo anterior y llámalo “240tabla09.html”. Prueba que funciona bien.

2.41. Crea un documento HTML “241tabla10.html” que contenga tres tablas, en la primera usa solo <colgroup> con sus atributos, en la segunda solo <col> con sus atributos, dentro de <colgroup> y en la tercera <colgroup> con sus atributos y <col> con sus atributos, dentro de <colgroup>. Cada tabla debe tener como mínimo 3 filas y 5 columnas. Tiene que ser un ejercicio original, no basado en alguno anterior.

Concepto	Primer semestre	Segundo semestre	Total
Alimentación	1500	1250	2750
Tecnología	850	30	880
Deporte	300	300	600
Total	2650	2080	4730

2.42. Crea un documento HTML “242tabla11.html” que contenga una tabla dentro de una celda. Si es necesario utiliza el atributo border. Pista: define

una nueva tabla con todos sus elementos dentro de la celda que prefieras. Tiene que ser un ejercicio original, no basado en alguno anterior.

2.43. Ejercicio libre de tablas. Crea un documento HTML llamado "243tablas12.html", que contenga varias tablas que muestren todo lo estudiado para tablas, puedes utilizar en una misma tabla varios de los elementos estudiados. Incluye también fuera de las tablas y en celdas de las tablas etiquetas vistas desde el inicio de curso, cuantas más mejor. Tiene que ser un ejercicio original, no basado en alguno anterior.

2.44. ¿Qué atributos de las tablas no deben ser utilizados a partir de HTML 5?

2.5. Formularios en HTML: estructura y elementos

Un formulario es un bloque compuesto por un conjunto de elementos de entrada de información. Permite al usuario introducir datos con el objetivo de enviarlos a un servidor para realizar algún tipo de proceso con ellos.

Es uno de los componentes más importantes en la interacción usuario y sitio web. Por ejemplo, acceso mediante usuario y contraseña, búsqueda de artículos en tienda web, reserva de billetes o habitaciones, etc...

2.5.1. Elementos de los formularios

Los elementos de los formularios también se pueden utilizar fuera del contexto de los formularios, ya sea para presentar información o para realizar algún tipo de programación mediante JavaScript.

<form>

Delimita el formulario. Tiene significado semántico. Debe cumplir las siguientes reglas:

- Hay que definir el destino (atributo action) y el modo de envío de los datos (atributo method).
- Hay que indicar de qué componentes queremos enviar los valores introducidos por el usuario (atributo name).
- Hay que especificar cuándo se quiere hacer el envío (elemento <input> de tipo submit).
- Opcionalmente, se pueden incluir reglas para validar los datos de entradas de forma automática antes del envío, cancelándose si no se cumplen.

Por lo tanto , el ciclo de funcionamiento de un formulario es:

- El usuario introduce los datos en los campos de entrada.

- El usuario pulsa el botón de envío (submit).
- El navegador valida las reglas asignadas a los campos de entrada.
- Si se cumplen, los datos se envían al destino indicado según el método seleccionado.

Formulario de ejemplo, se indica que el destinatario es un programa escrito en PHP ubicado en un servidor local, que el método de envío es GET y que los datos que se van a enviar son los que se introduzcan en los elementos <input> que tienen atributo name. El atributo required obliga a introducir valores en los campos de entrada, no se pueden dejar vacíos.

```
<form          action="http://localhost/calculadoraimpuestos.php"
method="GET">
    <input type="text" placeholder="Concepto" name="concepto"
required>
    <input type="number" placeholder="Importe" name="importe"
required>
    <input type="submit" value="Calcular">
</form>
```

Resultado:

Concepto	Importe	Calcular
----------	---------	----------

Los atributos más importantes que acompañan a <form> son los siguientes:

- action: URL de destino. Determina la URL (dirección de un programa en un servidor web) que va a procesar el formulario.
- method: GET, POST. GET, los datos se envían como anexo a la URL, serán visibles en el navegador. POST, los datos se envían en el cuerpo de la petición, no son visibles en una navegación convencional. Para datos privados siempre usar POST.

- target: `_self` (por defecto), `_blank`: indica dónde mostrar el contenido devuelto por el servidor. `_self` lo muestra en la misma pestaña. `_blank` en una pestaña nueva.

<input>

Proporciona un componente interactivo para que el usuario introduzca información.

Hay muchos tipos de entrada que reconoce `<input>`. Los tipos de entrada aportan cierta seguridad, ya que contienen restricciones para certificar que el tipo de dato que se está introduciendo es acorde al tipo de entrada diseñada. Por ejemplo, el tipo *email* incluye una validación que comprueba, antes de enviar los datos, que la dirección de correo electrónico es correcta.

Ejemplo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="urf-8">
  <title>Ejemplo entradas</title>
</head>
<body>
  <header></header>
  <main>
    <form>
      Botón <input type="button" value="No sirvo para mucho"><br>
      Fichero <input type="file"><br>
      Número <input type="number"><br>
      Clave <input type="password"><br>
    <p>
```

```

Color de pelo:<br>
<input type="radio" name="colorpelo" value="rubio"> Rubio<br>
    <input type="radio" name="colorpelo" value="castaño">
Castaño<br>
    <input type="radio" name="colorpelo" value="gris"> Gris<br>
</p>
    Imagen que envía formulario <input type="image" src="flecha.jpg"
width="25"><br>
    Botón que envía formulario <input type="submit"><br>
    Reset formulario <input type="reset"><br>
</form>
</main>
<footer></footer>
</body>
</html>

```

2.45. Crea un documento HTML con el ejemplo anterior llamado "245form01.html". Prueba su visualización. Ejemplo de visualización:

Botón

Fichero No se ha seleccionado ningún archivo

Número


Clave

Color de pelo:

☐ Rubio

☐ Castaño

☐ Gris

Imagen que envía formulario 

Botón que envía formulario

Reset formulario

Principales valores del atributo *type* de `<input>`:

- button: botón de acción.

- checkbox: casilla para la selección de un elemento.
- color: selector de color.
- date: fecha.
- datetime-local: fecha y hora.
- email: correo electrónico.
- file: fichero del ordenador.
- hidden: no visualiza lo introducido.
- image: mismo comportamiento de *submit* pero muestra una imagen.
- month: mes y año.
- number: valor numérico.
- password: contraseña, sustituye los caracteres introducidos por asteriscos o puntos.
- radio: selección de un único elemento dentro de un conjunto.
- range: selección de un valor numérico dentro de un rango.
- reset: restaura el estado inicial de los campos del formulario.
- search: texto para realizar una búsqueda.
- submit: verifica y envía los datos del formulario al destino en el atributo *action* de este.
- tel: número de teléfono.
- text: texto cualquiera.
- time: hora.
- url: URL
- week: año y ordinal de la semana de ese año.

2.46. Crea un documento HTML llamado "246form02.html" que incluya al menos un ejemplo de cada uno de los tipos de entrada (atributo type) que reconoce <input>. Ejemplo de visualización:

Botón

Idiomas:

☐ Español

☐ Catalán

☐ Vasco

☐ Gallego

Color

Fecha

Fecha y hora

Correo

Fichero Ninguno archivo selec.

Oculto

Mes

Número

Rango

Clave

Color de pelo:

☐ Rubio

☐ Castaño

☐ Gris

Búsqueda

Teléfono

Texto

Hora

Dirección web

Semana

Imagen que envía formulario

Botón que envía formulario

Reset formulario

Atributos de <input>:

- autofocus: el componente obtiene el foco cuando se carga la página, sólo se utiliza en un elemento por página.
- disabled: desactiva el componente.
- list: permite asociar el componente con una lista de opciones definidas en un elemento <datalist>.
- name: nombre del elemento de cara a su envío al servidor. Si el elemento no tiene este atributo, no será enviado al servidor, aunque aparezca en el formulario.
- placeholder: recibe como valor un texto y este se muestra en el espacio destinado a la entrada de datos. Desaparece cuando el usuario introduce el valor real.

- readonly: hace que el valor sea visible pero no editable.
- required: convierte un campo en obligatorio.
- type: determina la apariencia del componente, el tipo de datos que se va a introducir y la validación. Si se omite, el tipo por defecto es text.
- value: valor por defecto del componente. Este atributo es opcional, excepto cuando el atributo type es radio o checkbox, para indicar el valor devuelto por el formulario.

2.47. Crea un documento HTML llamado “247form03.html” que incluya los tipos de datos que te parezcan adecuados. Usa en el documento, como mínimo una vez, todos los atributos que pueden acompañar a <input>-menos list-. Tiene que ser un ejercicio original, no basado en alguno anterior. Ejemplo de visualización:

Identificación, por favor:

Correo electrónico	<input type="text"/>
Contraseña	<input type="password"/>
<input type="button" value="Enviar"/>	

Instituto	<input type="text" value="IES Rafael Alberti"/>
Apellidos y nombre	<input type="text"/>
Fecha de ingreso	<input type="text" value="dd/mm/aaaa"/> <input type="button" value="📅"/>
<input type="button" value="Mandar datos"/>	

Dirección de contacto	<input type="text" value="Plaza del pollo S/N"/>
Teléfono	<input type="text"/>

<button>

Permite crear un botón, que no tiene una utilidad predefinida ya que debe programarse, por ejemplo utilizando JavaScript, invocado por onclick

dentro de la definición de <button>. No confundir con el valor button del atributo type de input.

Puede incluirse dentro de <button> cualquier contenido HTML que se desee. Puede tener distinto funcionamiento dependiendo del navegador utilizado, por eso es muy conveniente indicar siempre type="button".

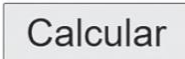
Presenta como atributos básicamente los mismos que la etiqueta <input> de <form>.

Aunque puede usarse dentro de la etiqueta <form>, lo normal es que se utilice fuera de ella, como código HTML independiente.

Ejemplo 1:

```
<button type="button">Calcular</button>
```

Resultado:



Ejemplo 2:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <name charset="utf-8">
  <title>Ejemplo button</title>
</head>
<body>
  <header>
  </header>
  <main>
    <br>
    <br>
```

```

<button type="button">
    <p><strong>El contenido HTML</strong> está permitido en este
botón</p>
    <p>También estudia al
    <a href="https://es.wikipedia.org/wiki/HTML" target="_blank">Wiki
HTML.</a>
    </p>
</button>
<hr>
<form>
    Versión HTML <input type="number" name="VER" placeholder="5">
    Año <input type="date" name="anio">
    <input type="submit" value="Mandar datos">
</form>
<hr>
<br>
<br>
</main>
<footer>
</footer>
</body>
</html>

```

Resultado:

El contenido HTML está permitido en este botón

También estudia al [Wiki HTML](https://es.wikipedia.org/wiki/HTML).

Versión HTML

Año

2.48. Crea un documento HTML llamado “248form04.html” que incluya tres botones dentro de una tabla con textos indicativos del uso de cada botón.

Al pulsar el primero enlazará a una página web en la misma pestaña. Al pulsar el segundo saltará el inicio de nuestra página web. Al pulsar el tercero saltará al final de nuestra página web. Tiene que ser un ejercicio original, no basado en alguno anterior. Ejemplo de visualización:

Tabla de botones

Botón a una web	Botón a inicio	Botón al final
Pulsame y verás	Me voy al inicio	Me voy al final

Se acabó

<select>

Permite al usuario elegir una opción de un conjunto de valores como una lista de opciones. Las opciones se establecen con el componente <option> y se pueden agrupar con el componente <optgroup>.

Atributos significativos:

- multiple: indica si permite selección múltiple. Para asegurarnos que todas las opciones seleccionadas se mandarán al servidor, agregaremos un par de corchetes ("[]") al final del valor en el atributo name.
- size: indica el número de líneas que se mostrarán simultáneamente.
- Además presenta básicamente los mismos que la etiqueta <input> de <form>.

<option>

Define una opción seleccionable dentro de un elemento <select> o una sugerencia de un elemento <datalist>.

Atributos significativos:

- disabled: indica si la opción está deshabilitada.
- selected: indica que la opción debe estar seleccionada al cargar el documento.
- value: dato que se envía al servidor cuando se envía el formulario. No hay que olvidar que el contenido del elemento <option> tiene como único objetivo ser presentado en el navegador.

Ejemplo 1: Al pulsar sobre el primer elemento se despliega la lista para elegir la opción.

```
<form>
.....
.....
<select name="deporte">
    <option>Baloncesto</option>
    <option>Fútbol</option>
    <option selected>Tenis</option>
    <option>Golf</option>
    <option>Esquí</option>
    <option>Remo</option>
</select>
.....
.....
</form>
```

Resultado:

Lista sin desplegar

Baloncesto ▾
Baloncesto
Fútbol
Tenis
Golf
Esquí
Remo

Lista desplegada

Ejemplo 2: como el atributo size tiene valor "5", aparece la lista con cinco elementos visibles. El atributo múltiple permite selección múltiple pulsando la tecla Ctrl y el botón izquierdo del ratón para selecciones/deselecciones alternas, o pulsando la tecla Shift y el botón izquierdo del ratón para hacer selecciones/deselecciones por bloques.

<form>

.....

.....

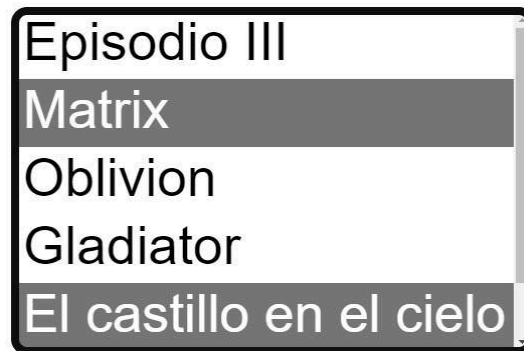
```
<select name="pelifav[]" multiple size="5">
  <option>Episodio III</option>
  <option>Matrix</option>
  <option>Oblivion</option>
  <option>Gladiator</option>
  <option>El castillo en el cielo</option>
  <option>Ratatouille</option>
</select>
```

.....

.....

</form>

Resultado:



Ejemplo 3:

<form>

.....

.....

```
<select name="personajeaño" size="7">
  <option value="opcion1">Sauron</option>
  <option value="opcion2">Frodo</option>
  <option value="opcion3">Gandalf</option>
  <option value="opcion4" disabled>Gollum</option>
  <option value="opcion5">Galadriel</option>
  <option value="opcion6">Éowyn</option>
  <option value="opcion7" selected>Yo mismo</option>
</select>
```

.....

.....

</form>

Resultado:



2.49. Crea un documento HTML llamado "249form05.html" que incluya 4 listas de opciones. Las dos primeras de selección única -libros y canciones-, las dos últimas de selección múltiple -colores y ciudades-. Utiliza en cada una de ellas un atributo distinto de los que se permiten. Tiene que ser un ejercicio original, no basado en alguno anterior. Ejemplo de visualización:

The screenshot shows a web form with the following elements:

- A label "Indica el libro que prefieras:" followed by a single-select dropdown menu with options: "Los renglones torcidos de Dios", "La legión perdida", "Fundación y tierra", and "No me gusta leer".
- A label "Selecciona la canción y canta:" followed by a single-select dropdown menu with the option "El Reloj".
- A label "Dime que colores prefieres:" followed by a multiple-select dropdown menu with options: "marrón", "rojo", "azul", and "amarillo".
- A label "¿Ciudad en la que vives?" followed by a multiple-select dropdown menu with options: "San Fernando", "Páris", and "Cádiz".
- A button labeled "Mandar selecciones" located to the right of the city dropdown.

<optgroup>

Agrupar un conjunto de elementos <option> de un elemento <select>.

Atributos significativos:

- disabled: impide seleccionar la opción.
- label: determina el nombre del grupo de opciones.

Ejemplo:

```
<form>
.....
<select name="ruina">
  <optgroup>
    <option value="opcion0" label="Coches"></option>
  </optgroup>
  <optgroup label="Disponibles">
    <option value="opcion1">Seat</option>
    <option value="opcion2">Kia</option>
    <option value="opcion3">Tesla</option>
```

```

</optgroup>
<optgroup label="Sin stock">
  <option value="opcion4">Ferrari</option>
  <option value="opcion5">Volvo</option>
  <option value="opcion6">Fiat</option>
  <option>Tata</option>
</optgroup>
</select>

```

.....

</form>

Resultado:



Coches ▾

- Coches
- Disponibles**
 - Seat
 - Kia
 - Tesla
- Sin stock**
 - Ferrari
 - Volvo
 - Fiat
 - Tata

2.50. Crea un documento HTML llamado “250form06.html” que incluya 3 listas de opciones agrupadas -planetas, comidas y bebidas-. La lista de bebidas debe ser de elección múltiple. Usa algunos atributos. Tiene que ser un ejercicio original, no basado en alguno anterior. Ejemplo de visualización:

Planetas Comidas Bebidas

Con gas
Coca cola
Kas naranja
Agua con gas
Con alcohol
Ron cubano
Vodka ruso
Sin alcohol sin gas
Té
Agua
Zumo
No beber

Planetas Comidas Bebidas

Con gas
Coca cola
Kas naranja
Agua con gas
Con alcohol
Ron cubano
Vodka ruso
Sin alcohol sin gas
Té
Agua
Zumo
No beber

Rocosos
Mercurio
Venus
Tierra
Gaseosos
Júpiter
Saturno
Urano
Neptuno

Planetas Comidas Bebidas

Con gas
Coca cola
Kas naranja
Agua con gas
Con alcohol
Ron cubano
Vodka ruso
Sin alcohol sin gas
Té
Agua
Zumo
No beber

Elige la mejor
Sanas
Lechuga
Mortales
Jamón serrano
Chorizo
Morcilla
Pescado con mercurio
Verdura con pesticida

<datalist>

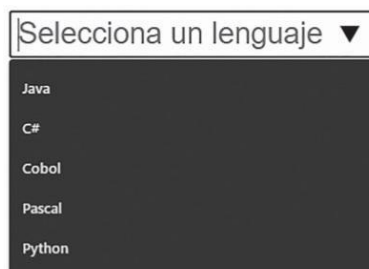
Define un conjunto de elementos `<option>` como una lista para su uso en otros elementos del formulario, permitiendo elegir de la lista el valor del elemento definido en `<input type="..." ...>`

El elemento `<datalist>` contiene un atributo `"id"` que es referenciado por el atributo `"list"` del componente `<input>`. Es necesario para relacionar un componente de entrada con la lista de datos.

Ejemplo 1: `<datalist>` con `type="text"`:

```
<form>
.....
.....
<input type="text" list="lenguajes" placeholder="Selecciona un
lenguaje">
  <datalist id="lenguajes">
    <option>Java</option>
    <option>C#</option>
    <option>Cobol</option>
    <option>Pascal</option>
    <option>Python</option>
  </datalist>
.....
.....
</form>
```

Resultado:



Ejemplo 2: usemos ahora <datalist> con type="time":

```
<form>
.....
.....
<input type="time" name="horelegida" list="horas">
  <datalist id="horas">
    <option value="10:00"></option>
    <option value="10:15"></option>
    <option value="10:30"></option>
    <option value="10:45"></option>
    <option value="11:00"></option>
  </datalist>
.....
.....
</form>
```

2.51. Crea un documento HTML llamado "251form07.html" que incluya 3 listas <datalist> con 1 type="text" y 2 type más distintos, no vistos en los ejemplos. Una de las listas debe ser obligatoria de contestar. Investiga un poco los atributos que pueden aparecer con <type> de <input> para mejorar la práctica. Tiene que ser un ejercicio original, no basado en alguno anterior. Ejemplo de visualización:

Elige la temperatura:

Pincha en un número par:

Selecciona idioma:

Enviar

Elige la temperatura:

Pincha en un número par:

Selecciona idioma:

Enviar

Elige la temperatura:

Pincha en un número par:

Selecciona idioma:

Enviar

<fieldset>

Permite agrupar de forma visual un conjunto de campos de un formulario, mediante un marco, y de forma funcional permitiendo realizar acciones sobre los elementos agrupados. Mediante el elemento <legend> se puede asignar un texto que haga de título del marco.

Atributo significativo:

- disabled: deshabilita todos los elementos agrupados.

<legend>

Asigna un título o leyenda a un conjunto de campos agrupados mediante <fieldset>.

Ejemplo:

<form>

.....

.....

```
<fieldset>
  <legend>Cliente particular:</legend>
  <p>Nombre: <input type="text" placeholder="Nombre"></p>
  <p>Dirección: <input type="text" placeholder="Dirección de envío"></p>
</fieldset>
<fieldset disabled>
<legend>Cliente empresa:</legend>
  <p>Nombre: <input type="text" placeholder="Nombre de la empresa"></p>
  <p>Dirección: <input type="text" placeholder="Dirección de envío"></p>
  <p>CIF: <input type="text" placeholder="Código de Identificación
Fiscal"></p>
</fieldset>
```

.....

.....

</form>

Resultado:

Cliente particular:

Nombre:

Dirección:

Cliente empresa:

Nombre:

Dirección:

CIF:

<label>

Representa un texto que se asocia a un dato de un formulario. Los navegadores pueden enlazar ambos elementos -el texto al dato- permitiendo que los usuarios establezcan el enfoque sobre el dato al hacer clic en su texto asociado por <label>.

La asociación se realiza conteniendo el dato entre las etiquetas de apertura y cierre de <label>.

Ejemplo:

<form>

.....

.....

<label>Apodo: <input type="email"></label>

<label>Dirección de correo: <input type="email"></label>

.....

.....

</form>

Resultado:

Apodo: Dirección de correo:

Al hacer clic sobre “Apodo” o “Dirección de correo” el foco se sitúa sobre el dato elegido.

<textarea>

Proporciona un área de múltiples líneas para la introducción de texto.

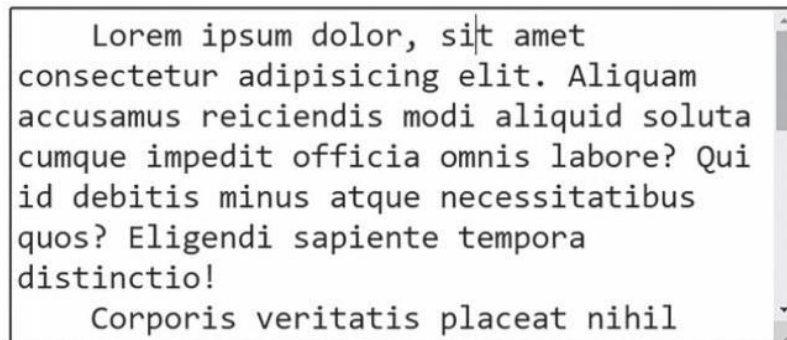
Atributos significativos:

- cols: ancho del área.
- maxlength: número máximo de caracteres.
- minlength: número mínimo de caracteres.
- name: nombre del elemento para su envío al servidor.
- placeholder: texto mostrado inicialmente, desaparece al escribir de nuevo.
- readonly: el componente es de solo lectura.
- required: obligatorio introducir el valor del componente.
- rows: número de líneas visibles.
- wrap: indica el método de ajuste del texto para el envío al servidor.
 - hard: envía los saltos de línea como parte del texto.
 - soft: envía el texto tal cual ha sido introducido.

Ejemplo:

```
<form>
.....
.....
<textarea name="hard" cols="40" rows="8" wrap="hard"></textarea>
.....
.....
</form>
```

Resultado:



Lorem ipsum dolor, sit amet
 consectetur adipiscing elit. Aliquam
 accusamus reiciendis modi aliquod soluta
 cumque impedit officia omnis labore? Qui
 id debitis minus atque necessitatibus
 quos? Eligendi sapiente tempora
 distinctio!
 Corporis veritatis placeat nihil

2.52. Crea un documento HTML llamado "252form08.html" que muestre 3 ejemplos distintos del elemento/etiqueta <fieldset> -uno en <header>, otro en <main> y otro en <footer>-, incluyendo los elementos/etiquetas <legend>, <label> y <textarea> según tu criterio. Usa algunos atributos ya estudiados desde el inicio de los formularios. Tiene que ser un ejercicio original, no basado en alguno anterior. Ejemplo de visualización:

Introduzca los datos

Iniciar sesión

Correo electrónico:

Contraseña:

Datos de contacto

Ahorros:

Color de coche:

Teléfono:

Fecha y hora de entrada:

Comentario sobre la empresa

Si me pinchas, lo mando

<meter> (+)

Representa un valor dentro de un rango. Para medir grados de avance es más apropiado utilizar el componente <progress>. Para su funcionamiento eficaz es necesario que obtenga sus valores del resultado de un programa. Puede aparecer dentro o fuera de un formulario.

En ausencia de los atributos “low”, “high” y “optimun”, las barras se representarán en color verde.


Atributos significativos:


- low: indica el valor para que la barra se represente en rojo.
- high: indica el valor para que la barra se represente en naranja.
- optimun: indica el valor para que la barra se represente en verde.
- value: indica el valor actual.
- min: indica el valor mínimo.
- max: indica el valor máximo.

Puede que los navegadores no representen bien los colores de low, high y optimum, teniendo que usar CSS para su correcta visualización.

Ejemplo 1:

```
<p>Avance del proceso: <meter min="0" max="100" value="20"></meter></p>
<p>Avance del proceso: <meter min="0" max="100" value="90"></meter></p>
```


Avance del proceso: 


Avance del proceso: 


Ejemplo 2:

```
<p>Calificación: <meter min="0" max="10" low="5" high="7" optimum="8" value="4.9"></meter></p>
<p>Calificación: <meter min="0" max="10" low="5" high="7" optimum="8" value="5"></meter></p>
<p>Calificación: <meter min="0" max="10" low="5" high="7" optimum="8" value="6.9"></meter></p>
<p>Calificación: <meter min="0" max="10" low="5" high="7" optimum="8" value="7"></meter></p>
<p>Calificación: <meter min="0" max="10" low="5" high="7" optimum="8" value="10"></meter></p>
```

Resultado:

Calificación: 

Calificación: 

Calificación: 

Calificación: 

Calificación: 

<progress> (+)

Muestra un indicador visual del grado de avance de un proceso. Si no se asigna valor a `value`, aparece en movimiento de izquierda a derecha y de derecha a izquierda, indicando que el progreso es desconocido. Para su funcionamiento eficaz es necesario que obtenga sus valores del resultado de un programa. Puede aparecer dentro o fuera de un formulario.

Puede que los navegadores no representen bien `<progress>`, teniendo que usar CSS para su correcta visualización.

Atributos significativos:

- `value`: valor actual.
- `max`: valor máximo.

Ejemplo:

```
<progress></progress>  
<progress max="10" value="3"></progress>
```



`<output>` (+)

Delimita el resultado de una operación realizada, normalmente, por un programa. Puede aparecer dentro o fuera de un formulario.

Atributos significativos:

- `for`: contiene la relación separada por espacios de los identificadores de los elementos que participan en el cálculo. Es información semántica.
- `form`: incluye el identificador del formulario con el que está asociado, aparece fuera de un formulario.
- `name`: nombre con el que el valor del campo se envía al servidor.

Ejemplo 1:

```

<form>
    .....
    .....
    <input type="number" id="operador1" value="8">
    +
    <input type="number" id="operador2" value="2">
    =
    <output for="operador1 operador2" name="resultado">10</output>
    .....
    .....
</form>

```

Resultado:

+ = 10

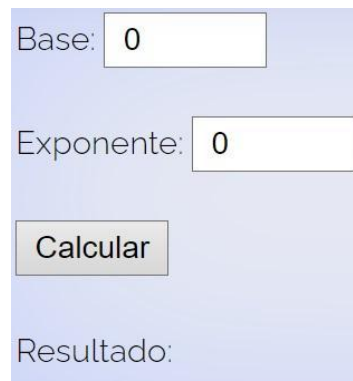
Ejemplo 2:

```

<form>
    .....
    <p>Base: <input type="number" id="base" value="0" min="0"
    max="30"></p>
    <p>Exponente: <input type="number" id="exponente" value="0"
    min="0" max="30"></p>
    <p><input type="button" value="Calcular"
    onclick="mostrarPotencia()"></p>
    <p>Resultado: <output id="potencia" for="base
    exponente"></output></p>
    .....
</form>

```

Resultado:



Base:

Exponente:

Resultado:

(+) 2.53. Crea un documento HTML llamado “253form09.html” que muestre algunos ejemplos simulados de <meter>, <progress> y <output>. Tiene que ser un ejercicio original, no basado en alguno anterior.

2.54. Crea un documento HTML “254form10.html” de estilo libre con lo estudiado para formularios web, cuanto más completo mejor. Tiene que ser un ejercicio original, no basado en alguno anterior.

2.5.2 Validaciones (+)

Antes de enviar un formulario al servidor es conveniente validar que los datos introducidos por el usuario cumplen con las restricciones que se hayan establecido.

Durante la definición de un formulario, se especifican las condiciones que deben cumplir dichos datos.

La validación de las reglas se realiza previamente al envío mediante la pulsación de un elemento <input> de tipo submit. Se puede programar la validación en JavaScript, pero requiere más esfuerzo y solo en caso de necesidad.

Algunos de los atributos de validación en formularios son:

- required: campo obligatorio, en tipo texto los espacios en blanco se consideran válidos.

- minlength y maxlength: longitudes mínimas y máximas en tipo texto.
- min y max: valores mínimos y máximos en tipos numéricos.
- type: en función del valor asignado a este parámetro, se fijan valores correctos de entrada. Por ejemplo: si el valor es email, se validará que la entrada sea una dirección de correo electrónico.

Una idea importante sobre las validaciones es que una aplicación web consta de una parte cliente (se ejecuta en el navegador) y otra parte cliente (se ejecuta en el servidor). Las validaciones que hemos estudiado corresponden a la parte cliente, que son relativamente sencillas de saltar.

En caso de desarrollar una aplicación web con un nivel de fiabilidad medio o medio-alto, es conveniente disponer de validación de datos en el servidor, ya que es mucho más difícil de evitar.

2.6. Multimedia en HTML

Los elementos multimedia han estado en la web desde sus inicios usando la etiqueta ``. HTML 5 incrementó la posibilidad de añadir multimedia mediante los elementos `<audio>` y `<video>`.

2.6.1. Elementos multimedia

`<source>` (+)

Permite indicar diferentes versiones de un archivo en los elementos `<audio>` y `<video>`, que pueden contener uno o más elementos `<source>`, cada uno de los cuales provee una versión diferente de audio o vídeo (por ejemplo, codificados de forma diferente). Es el navegador el encargado de elegir qué versión cargar, una decisión que tomará en base a los formatos que soporta.

Atributos significativos:

- `src`: determina la URI del recurso.
- `type`: indica el tipo de contenido del recurso.

`<audio>`

Permite insertar sonido en la web. Para añadir el recurso se puede asignar con el atributo `src` o con la etiqueta `<source>`.

Permite indicar un contenido distinto por si el recurso no fuera compatible con el navegador.

Atributos significativos:

- `src`: determina la URI del recurso.
- `autoplay`: comienza la reproducción automática del recurso.
- `loop`: indica que la reproducción será en bucle.

- muted: si existe, silencia el componente.
- controls: si existe, proporciona controles para controlar la reproducción.

Ejemplo 1:

```
<audio src="rugido.ogg" controls>
```

Este contenido no es compatible con el navegador.

```
</audio>
```

Resultado:



Ejemplo 2:

```
<audio controls>
```

```
<source src="horse.ogg" type="audio/ogg">
```

```
<source src="horse.mp3" type="audio/mpeg">
```

Tu navegador no soporta este elemento audio.

```
</audio>
```

Resultado:



<video>

Permite incluir vídeo en la página. Se puede asignar con el atributo `src` o con el elemento `source`. Proporciona contenido alternativo si no es compatible con el navegador.

Atributos significativos:

- `src`: indica la URI del recurso.
- `poster`: indica la URI para mostrar una imagen mientras se carga el vídeo.
- `autoplay`: comienza la reproducción automática.
- `loop`: reproducción en bucle.
- `muted`: si existe, silencia el componente.
- `controls`: si existe, proporcionar interfaz de control de reproducción.
- `width` y `height`: número de píxeles de ancho y alto. Es común indicar sólo uno de ellos, ya que el otro se adaptará automáticamente, evitando deformar el vídeo mostrado.

Ejemplo:

```
<video      src="trailerpelicula.mp4"      poster="posterpelicula.jpg"
height="300" controls>
  Contenido no soportado.
</video>
```

Resultado:



Permite incluir una imagen mediante el atributo src. La compatibilidad del tipo de imagen dependerá del navegador.

Atributos significativos:

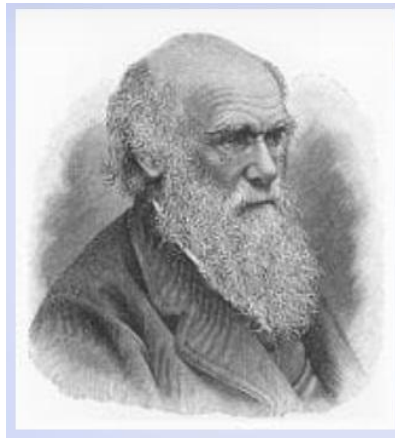
- alt: contiene texto que se mostrará cuando no se pueda ver la imagen.
- src: URI de la imagen a mostrar. Obligatorio.
- width y height: número de píxeles de ancho y alto. Es común indicar sólo uno de ellos, ya que el otro se adaptará automáticamente, evitando deformar la imagen mostrada.

Ejemplo:

```

```

Resultado:



<iframe>

Permite acceder a un recurso que se encuentra fuera de la ubicación de la página mostrada en el navegador. Básicamente incrusta un recurso dentro de una página HTML, mostrándolo en el navegador Web: anuncios, otras webs, vídeos de youtube, ventanas de google maps, etc.

Atributos significativos:

- src: URI del recurso incrustado.
- allowfullscreen: permite al contenido incrustado ocupar la pantalla completa.
- width y height: número de píxeles de ancho y alto. Es común indicar sólo uno de ellos, ya que el otro se adaptará automáticamente, evitando deformar el vídeo mostrado.

Ejemplo 1: incrustar una página HTML dentro de otra.

```
<iframe width="500" src="240tabla09.html"></iframe>
```

Ejemplo 2: el contenido de la etiqueta <iframe> se obtiene del propio youtube mediante “Compartir-Insertar”. Incluye atributos no estudiados que su conocimiento se deja a criterio del alumno/alumna”.

```
<iframe width="560" height="315"
src="https://www.youtube.com/embed/-eoVyqtASEE" title="YouTube video
player" frameborder="0" allow="accelerometer; autoplay; clipboard-write;
encrypted-media; gyroscope; picture-in-picture" allowfullscreen></iframe>
```

Resultado:

Gastos anuales		
Concepto	Primer semestre	Segundo semestre
Alimentación	1500	1750
Tecnología	850	30
Deporte	300	300



2.55. Crea un documento HTML “255media01.html” con dos ejemplos distintos de `<audio>`, `<video>`, `` e `<iframe>`. Tiene que ser un ejercicio original, no basado en alguno anterior.

2.56. Diseña una página web “256todo01.html” haciendo uso de los elementos estudiados durante éste primer trimestre -excepto los etiquetados como (+)-. De forma excepcional, utiliza tablas para crear una estructura de la página web lo más agradable y estéticamente atractiva. Se valorará el diseño de la página web (40%) y la cantidad de elementos/etiquetas distintas usadas (60%) -se da por supuesto que la

página web presentará un amplísimo significado semántico-. Tiene que ser un ejercicio original, no basado en alguno anterior.

2.7. XHTML: la versión XML de HTML (+)

Es más estricto, robusto y está mejor estructurado que HTML. Facilita el procesamiento automático y permite validaciones de manera más sencilla. Su objetivo era avanzar hacia la web semántica.

Pese a sus ventajas, no se ha consolidado y en la actualidad la combinación de HTML5 y CSS3 se ha convertido en el estándar para el diseño de páginas web.

Los documentos XHTML se almacenan en ficheros de texto con extensión .xhtml.

2.7.1. Diferencia entre XHTML y HTML (+)

- Todos los elementos/etiquetas deben cerrarse.
- Los elementos no vacíos que no tengan contenido no pueden cerrarse de forma abreviada.
- Los elementos deben estar correctamente anidados.
- Distingue entre mayúsculas y minúsculas.
- Todas las etiquetas deben estar en minúsculas.
- Los valores de los atributos deben ir obligatoriamente entre comillas.
- Todos los atributos deben tener asignado un valor.

En HTML muchas de estas obligaciones pueden saltarse por error de codificación y los navegadores sabrán interpretarlos y mostrar correctamente la web.

2.8. Complemento Emmet (+)

Proporciona atajos para la creación automática de código HTML.

Ejemplo: Utilizando Visual Studio Code, realizar la siguiente práctica.

- Crear un fichero con el nombre index.html.
- Escribir en la primera línea `html:5` y pulsar el tabulador o enter.
- Dentro del elemento `<body>` escribir `table>tr*3>td*2` y pulsar tabulador o enter.
- A continuación escribe `ol>li*3` y pulsar tabulador o enter.

(+) 2.57. Busca información sobre el complemento Emmet y su uso en Visual Studio Code.