Práctica 1

Sesiones 1 y 2

Objetivos

Comprender los fundamentos básicos de HTML.

Entender cómo se estructura un documento y los conceptos de elemento, marca, atributo y valor.

Conocer los elementos principales del lenguaje HTML y su funcionalidad.

Ser capaz de combinar todos los elementos estudiados para diseñar y construir documentos HTML.

Introducción a HTML5

Internet es esencialmente una red de ordenadores. Los navegadores son programas que muestran páginas web que están escritas en HTML, que es el lenguaje de la Web, y que están alojadas en dichos ordenadores.

HTML es el acrónimo de *HyperText Markup Language*: hipertexto (texto que, mediante enlaces, nos permite "saltar" a otros textos localizados en él mismo o en otros textos dentro del mismo servidor, o en otros servidores remotos) y un lenguaje de marcado (que posibilita tomar un texto plano y, mediante etiquetas, estructurarlo en un texto comprensible para los dispositivos electrónicos).

El World Wide Web Consortium, la organización responsable de crear las especificaciones HTML, ha desarrollado varias versiones de este lenguaje de marcas. Concretamente en 1999 concluyó la recomendación de la versión 4 de HTML. Con la idea de mejorar las carencias de esta versión y de dotarla de nuevas prestaciones, en 2005 comienza a gestarse la versión 5 de HTML, más adaptada al concepto actual de la Web y a lo que será ésta en el futuro. Su definición concluyó en 2014 aunque está oficialmente considerado como un *Living Standard*, evolucionando con nuevas características.

HTML define documentos que pueden ser visualizados en un navegador. Estos documentos quedan estructurados en elementos, que son sus componentes, organizados en forma de árbol, donde el primer nodo es el elemento raíz de un documento y los nodos hojas contienen texto. El resto de nodos pueden tener texto y otros elementos. Los elementos representan un contenido semántico.

Ejercicio:

¿Cómo quedaría la representación en forma de árbol de este código HTML?

<!doctype html> esto es una marca, y es la que tenemos que poner primero. Le decimos al navegador que el documento que tiene que rnderizar es HTML.

<html> todo lo que se abre en html se cierra.

```
<head>
        <title>Just a Test</title>
</head>
<body>
        <h1>Just a test</h1>
        <p>Nothing much to see <a href="test.html">here</a>. </p>
        <hr>
        <p>Move <a href="nextpage.html">along</a>. </p>
</body>
</html>
```

se mete metainformación sobre el documento.
P.ej, un metadato es el título (aparece en la parte de arriba del navegador)

cuerpo del documento

Cada elemento, cada nodo interno de este árbol, tiene asociada una etiqueta en HTML (*tag*), con otros nodos como descendientes que incluyen su contenido, que puede ser otros nodos representando etiquetas o textos. Las etiquetas se componen del nombre del elemento más los paréntesis angulares. Las etiquetas de cierre también tienen el carácter '/' justo después del paréntesis angular de apertura:

```
<p> texto </p>
```

párrafo

Algunos elementos no requieren la etiqueta de cierre (como es el caso del elemento *br* o *hr* (
 -nueva línea-, <hr> -línea horizontal-, respectivamente). Todos los elementos que tengan contenido deben ser cerrados mediante la etiqueta correspondiente.

Los elementos pueden tener asociados atributos, que pueden verse como propiedades de los primeros, y pueden tener (o no) asociados un valor, normalmente entre comillas dobles.

```
<input type="checkbox" checked>  ó <a href="URL" title="Titulo del enlace">Texto del enlace</a>
```

con esto, aparece por defecto con el tick

Esta imagen representa gráficamente la estructura de un elemento (http://en.wikipedia.org/wiki/HTML_element):

Los elementos tienen asociados unas etiquetas, que van entre paréntesis angulares. Suelen ser etiquetas de apertura y cierre. Los nombres de las etiquetas (tag) y atributos pueden ir en mayúsculas o minúsculas.

Otro concepto importante es el de entidad, que es un conjunto de caracteres reservados en HTML que se asocian a un símbolo o carácter. Por ejemplo, si ponemos directamente en un código los caracteres '<' ó '>', HTML puede entender que vamos a abrir una etiqueta o cerrarla. Por tanto, se utilizan cadenas especiales, que van delimitadas por '*&*' y '*;*', que se sustituirán por el símbolo que representan a la hora de mostrar el documento en el navegador. Así, "*<*" representa '<' y "*>*" *es* '>'. Cada entidad tiene asociado un identificador que también puede emplearse (*<*). Las tildes en español serían "*á*", "*é*", "*í*", etc. La 'ñ' se representa mediante "*&ntilde*;" El espacio en blanco es "* *", lo que permite añadir varios espacios en blanco seguidos (de otra forma, HTML interpreta varios espacios en blanco como uno solo). En http://dev.w3.org/html5/html-author/charref existe una tabla completa con todas las entidades existentes en HTML5.

Finalmente, hablaremos de los comentarios en un documento HTML, los cuales son texto que no se muestra en el navegador y que tienen como finalidad asistir al programador, ya que nos permite introducir aclaraciones, recordatorios, y comentarios, como su propio nombre indica, en el código, y, por tanto, nos permitan explicarlo. En el código siguiente vemos cómo incluirlos:

```
<p> Esto es un párrafo del texto </p>
<!-- Esto es un comentario que no se mostrará en el navegador. -->
```

En este práctica, haremos un repaso de HTML5, aunque no pretende ser exhaustivo, sino ofrecer una visión general de este lenguaje y sentar las bases para un estudio más profundo de forma autónoma.

Estructura básica de un documento HTML5

La estructura básica de un documento HTML es la siguiente:

```
<!doctype html>
<html lang="es">   el idioma de este doc es español. Pero no sirve más que para que quien vea el documento sepa que el documento está en español.
        <head>
            <meta charset="utf-8">   meta es uno de los elementos que solo va a funcionar con etiqueta de apertura y atributo.
            <title>Ejemplo de página web</title>
        </head>
        <body>
            <p>¡Hola, Mundo!</p>
        </body>
</html>
```

Podemos observar los siguientes elementos principales en este código:

<!doctype html>: debe especificarse como primera línea del documento HTML. Sirve para informar al navegador que está trabajando con un documento HTML.

<html lang="es"> … *</html>*: *html* es el elemento raíz del documento y mediante el atributo *lang*, especifica el idioma del documento ("en", "fr", etc.). Contiene dos elementos: *head*, seguido del elemento *body*.

<head>... *</head>*: contendrá el conjunto de metadatos para el documento (título, información sobre estilos y guiones, etc.). Su contenido no se muestra en el navegador. Este elemento contiene a su vez el elemento *meta* que, mediante el atributo *charset,* indica que se emplea UTF-8 como formato de codificación de caracteres. En el Ejemplo 2 podemos ver un elemento *head* más complejo. El elemento *title* sirve para poner el título del documento, el cual será visualizado por el navegador en la parte superior de la ventana que muestra el documento.

Veamos un segundo ejemplo, en donde podemos observar una cabecera de documento más compleja:

```
(http://www.w3.org/TR/html5/document-metadata.html#the-head-element)
1       <!DOCTYPE HTML>
2       <HTML lang="es">
3           <HEAD>
4             <META CHARSET="UTF-8">
5             <BASE HREF="http://www.ejemplo.es/">
6             <TITLE>Un documento con una cabecera grande</TITLE>
```

```
7        <LINK REL="STYLESHEET" HREF="defecto.css">
8        <SCRIPT SRC="support.js"></SCRIPT>
9        <META NAME="APPLICATION-NAME" CONTENT="CABECERA GRANDE">
10       <META NAME="AUTHOR" CONTENT="Juan Manuel">
11       <META NAME="DESCRIPTION" CONTENT="DOCUMENTO DE PRUEBA">
12     </HEAD>
13     <BODY>
         ...
```

enlazar un fichero externo donde tenemos las reglas de visualización de mi HTML, y las aplique

es la forma que tengo de que se ejecute código javascript

El elemento *base,* mediante el atributo *href*, indicará un URL que actuará de ruta absoluta en el servidor para las rutas relativas que se indiquen a lo largo del documento.

Seguidamente, en la línea 7 aparece un elemento *link*, donde se nos indica que el fichero de estilos (valor *stylesheet* en el atributo *rel*) que se debe aplicar en este documento es defecto.css (*href* es el atributo que indicará un recurso externo). Obsérvese que no se ha especificado la ruta donde se encuentra este fichero de estilos, ya que se ha indicado en el previamente en el elemento base. Se asume, por tanto, que el fichero .css estará situado en *http://www.ejemplo.es/*, siendo la ruta absoluta *http://www.ejemplo.es/defecto.css*.

En definitiva, mediante el elemento *link* podemos identificar recursos externos, referenciados mediante el atributo *href*, cuyo tipo se indica mediante *rel*, y que nos permite cargar el contenido del documento o fichero en href en el documento donde se ubique el elemento link.

En la línea 8, mediante el elemento *script* se especifica un programa en JavaScript (por defecto, se asume que está en este lenguaje) que se ejecutará en el cliente.

Por último, tenemos varias ocurrencias del elemento *meta*, el cual se emplea para indicar metadatos de la página (información general sobre ella, como el nombre del autor, el nombre de la aplicación, la descripción, etc.). Mediante el atributo *name* indicamos el tipo de metadato (*application-name*, *description*, *author*) y con *content* establecemos el valor correspondiente.

<*body*> … </*body*>: representa el contenido principal del documento y es visible en el navegador.

Ejercicio:

Crea un fichero que se llame holamundo.html con el contenido del Ejemplo 1, súbelo al servidor y visualízalo en el navegador.

Atributos globales de los elementos

Antes de entrar en la descripción de algunos de los principales elementos de HTML, es conveniente indicar varios atributos que se pueden emplear en todos los elementos, razón por la que se denominan atributos globales. Algunos de ellos son los siguientes:

id: identifica un elemento de forma única. ** … **

accesskey: toma como valor una tecla y permite llegar a ese elemento pulsando la tecla o número correspondiente.

class: se indica la hoja de estilo que se aplicaría al elemento.

hidden: no toma valor ninguno, de tal forma, que cuando está presente, el elemento que lo contiene no se muestra. **<p hidden > no muestra el párrafo**

lang: permite especificar el idioma del contenido del elemento.

style: posibilita la aplicación de atributos de hojas de estilo.

tabindex: establece un orden a la hora de desplazarse por el documento pulsando la tecla tabulador.

title: título del elemento. Aparecería al pasar el ratón por elemento.

Elementos generales de HTML5

Imágenes

El elemento *img* es el encargado de posibilitar la inclusión de imágenes (jpg, png y gif, normalmente) en un documento HTML. Su uso más básico es el siguiente:

```
<img src = "fotografia.png" alt = "Fotografía del equipo web" />
```

El fichero que contiene la imagen se especifica asignándole su nombre (y, en su caso, su ruta, al atributo src. En caso de que el fichero no se encuentre o no exista, se emplea el atributo *alt* al que se le asigna un texto describiendo la imagen y que será mostrado en ese caso.

Ejercicio:

Busca y bájate algunas imágenes e introduce en el fichero holamundo.html varios elementos img. ¿Cómo especificarías el tamaño de la imagen?

Enlaces

El uso básico y original de este tipo de elemento de HTML es poder navegar de una página a otra, aunque también se pueden enlazar muchos otros objetos, como imágenes, documentos pdf, vídeos, etc.). Para ello se utiliza el elemento *anchor*, *a*:

```
<a href="siguiente.html"> Página siguiente </a>
```

href es el atributo que indica el destino del enlace, y puede ser absoluto, indicando toda la ruta completa (http://servidor/directorio/fichero) o relativo (sólo el nombre del fichero como en el ejemplo). También podríamos clasificarlos en internos al servidor (local), cuando se enlaza con otro documento dentro del mismo servidor, o incluso a una parte concreta de la misma página, y externos, cuando se enlaza con un documento en un servidor remoto. Como contenido propiamente dicho del elemento *a* se deberá indicar el texto del enlace. También se pueden crear enlaces utilizando una imagen en lugar del texto anteriormente indicado:

```
<a href="www.ejemplo.es/html5-tutorial/siguiente.html">
        <img src = "fsiguente.jpg" alt = "Flecha siguiente" />
</a>
```

En cuanto a los enlaces internos dentro del mismo documento, cada elemento en HTML puede llevar un atributo *id* que lo identificaría de forma unívoca. Se puede crear un enlace que "salte" directamente al elemento deseado: **# = dentro del propio documento --> #id x = elemento del documento con id x**

```
<a href="#titulo133 "> Solución comentada del problema. </a>
<!-- O también: <a href="problema1.html#titulo133 "> Solución comentada del problema. </a>
     indicando que saltaremos al elemento con id titulo133 del documento problema1.html. -->
…
<h1 id="titulo133"> Solución comentada del problema </h1>
```

También se puede enlazar a un correo electrónico, de tal forma, que al seleccionar el enlace nos aparezca un editor de mensajes de correo electrónico:

```
<a href="mailto:jmfluna@decsai.ugr.es">Contacto </a>
```

O a un teléfono, con objeto de lanzar una aplicación para realizar llamadas:

```
<a href="tel:6666666666">Teléfono</a>
```

Ejercicio:

Introduce los siguientes enlaces:

- De texto, a la página principal de la Universidad de Granada.
- Mediante una imagen, al buscador Google.
- Crea varios párrafos en tu documento con diferentes identificadores y enlaza uno de ellos.
- Crea un segundo documento HTML, prueba.html, y enlaza a dicho fichero.

Tablas

Las tablas en HTML5 nos permiten estructurar datos de forma sencilla. Veamos el siguiente ejemplo para conocer las características básicas de las mismas:

```
<!DOCTYPE html>
<html>
  <body>
    <table summary="Esta tabla nos ofrece un ejemplo de cómo construirlas en HTML">
    <caption>Título de la tabla – caption</caption>
    <thead>
      <tr>
        <th rowspan="2">Encabezamiento multifila - thead/tr/th - Ejemplo de rowspan</th>
        <th colspan="2">Encabezamiento multicolumna - thead/tr/th Ejemplo de colspan</th>
      </tr>
      <tr>
        <th>Encabezamiento columna 1 – tr/th</th>
        <th>Encabezamiento columna 2 - tr/th </th>
      </tr>
    </thead>
  <tfoot>
    <tr>
      <td colspan="3">Esto es un ejemplo de un pié de tabla - Ejemplo de tfoot y colspan</td>
    </tr>
  </tfoot>
```

```
  <tbody>
    <tr>
      <th>Fila 1 – tr/th</th>
      <td>Fila 1 - Columna 1 – tr/td</td>
      <td>Fila 1 - Columna 2 - tr/td </td>
    </tr>
    <tr>
      <th>Fila 2 - tr/th </th>
      <td>Fila 2 - Columna 1 – tr/td</td>
    <td>Fila 2 - Columna 2 – tr/td</td>
    </tr>
  </tbody>
 </table>
</body>
</html>
```

En este código se crea una tabla (mediante el elemento *table*), el cual tiene dos atributos: *border*, que nos indica si la tabla tiene (1) o no (0) bordes en el exterior o en el interior.

Seguidamente, ya en el interior del elemento *table*, nos encontramos con *caption*, que nos permite indicar el título de la tabla. A partir de este momento, se disponen las celdas, organizadas en filas y columnas. Podríamos decir que hay tres tipos de celdas: las que pertenecen al encabezamiento de la tabla (contienen los títulos de las columnas y filas), las que pertenecen al pie de la tabla, y las que componen el cuerpo. El encabezamiento se representa mediante *thead*, el pie de página mediante *tfoot* y el cuerpo, por medio de *tbody*. Los navegadores pueden usar estos elementos para permitir hacer scroll del cuerpo de la tabla de manera independiente del encabezamiento o pie. Además, al imprimir una tabla grande que cubra varias páginas, estos elementos pueden permitir que el encabezamiento y pie de la tabla aparezcan en cada página. En el interior de estos elementos se disponen las filas, que agrupan las celdas que contienen, mediante el elemento *tr*, el cual puede tener elementos *th*, para indicar celdas de encabezamiento, y elementos *td* para indicar una celda que contiene datos.

Las celdas se pueden expandir hacia las filas (atributo *rowspan* de *th* y *td*) y hacia las columnas (atributo *colspan* de ambos elementos). En los dos casos hay que indicarle el número de filas o columnas en que se expande la celda correspondiente.

Ejercicio:

Escribe código en HTML que represente una tabla para tu horario de clase. Utiliza todos los elementos descritos anteriormente.

Listas

Existen en HTML dos elementos que se emplean para nombrar una lista de ítems, y que se relacionan con una secuencia numerada de ítems y con una relación no ordenada. Las primeras se identifican por medio del elemento *ol* y las segundas por *ul*. Cada componente de la lista se especifica mediante el elemento *li* (list item). Veamos un ejemplo:

```
    <ul>
```

```
        <li> un ítem </li>
        <li> otro ítem </li>
        <li> y uno final</li>
    </ul>
    <ol>
        <li> ítem 1</li>
        <li> ítem 2</li>
        <li value="33"> ítem 32</li>
    </ol>
```

En el ejemplo de lista ordenada, mediante el atributo value del elemento *li* se puede especificar el número que se pondrá en la numeración y desde el que se continuará por los sucesivos ítems.

Elementos semánticos

HTML5 nos da la posibilidad de estructurar semánticamente nuestros documentos mediante el uso elementos que nos permitan dotarlos de una organización interna. Estos son los siguientes:

Títulos (headlines)

Se emplean para enfatizar semánticamente texto que servirá como títulos de partes de los documentos. Hay seis niveles de cabeceras, desde h1 a h6 (<h1> Capítulo 1 </h1>). Se suelen utilizar de forma jerárquica, por lo que h1 representa el nivel más alto y h6 el más bajo, y no se pueden anidar.

Ejercicio:

Introduce varios títulos en holamundo.html para probar su uso.

Cabecera (header)

La etiqueta *header* se va a usar para una cabecera, generalmente la cabecera del documento, que suele contener contenido introductorio (títulos, logos, menús, etc.). Aunque también podemos emplearla como la cabecera de un artículo o de un apartado, por ejemplo. Su uso es el siguiente:

```
<header> contenido de la cabecera </header>
```

Nótese que este elemento es totalmente diferente, en función y concepción, al elemento *head*.

Pié de página (footer)

Análogamente al elemento *header*, existe el elemento *footer* que se refiere a un pié, generalmente asociado al documento:

```
<footer> contenido del pié </footer>
```

Áreas de navegación (nav)

Conceptualmente, este elemento se ha añadido a HTML5 para incluir el menú de navegación del sitio web. Típicamente suele incluirse en un elemento *header* y suele contener listas de enlaces.

```
<header>
        <img src="imagenes/logo.jpg" />
```

```
        <nav>
            <ul>
                <li><a href="home.html">Principal</a></li>
                <li><a href="productos.html">Productos</a></li>
                <li><a href="servicios.html">Servicios</a></li>
                <li><a href="sobre.htm">Sobre nosotros</a></li>
            </ul>
        </nav>
    </header>
```

Principal (main)

Especifica el contenido principal de un documento. Debe ser una etiqueta única al documento y no debería contener ningún contenido que se repita a lo largo de los documentos del sitio, como por ejemplo, cabeceras y pies de páginas, secciones laterales, enlaces de navegación, logos, etc. Y no debe ser descendiente de un article, aside, footer, header or nav.

```
<!DOCTYPE html>
<html>
    <head>
        <title>Ejemplo de uso de main </title>
    </head>
    <body>
        <header>Cabecera:
          <nav>
            <ul>
              <li><a href="enlace1.html">Enlace1</a></li>
              <li><a href="enlace2.html">Enlace2</a></li>
            </ul>
          </nav>
        </header>
      <main>
        <h1>Partes del contenido principal</h1>
        <nav>
            <ul>
                <li><a href="#parte1">Parte 1</a></li>
                <li><a href="#parte2">Parte 2</a></li>
            </ul>
```

```
            </nav>

            <h2 id="parte1">Parte1</h2>
            <p>Aquí va el contenido de la parte 1</p>
            <!-- … -->
            <h2 id="parte2">Parte2</h2>
            <p>Aquí va el contenido de la parte 2</p>
            <!-- … -->
            <h2 id="parte3">Parte 3</h2>
        </main>
        <footer> Copyright 2025 jmfluna</footer>
    </body>
</html>
```

Su especificación y ejemplos de uso podéis encontrarlos en https://www.w3.org/TR/html-main-element/ y https://www.w3schools.com/tags/tag_main.asp.

Secciones (sections)

Se emplean para agrupar contenido con una semántica similar, y la marca para indicarlas es *section* (piénsese en los capítulos de un libro). Podríamos organizar un documento de un portal de comercio electrónico en varias secciones, una para cada categoría de producto con el que comerciar (la sección de ordenadores de sobremesa, la de portátiles, la de monitores, etc.). O la página principal de un periódico, en que las secciones serían las categorías de noticias (nacional, internacional, deportes,...). Dentro de estas secciones encontraríamos los productos o las noticias concretas.

```
    <section>
        <h1>Visión general de HTML5</h1>
        <p>HTML5 se emplea para estructurar contenido en la web.</p>
    </section>
    <section>
        <h1>Visión general de CSS3</h1>
        <p>CSS3 tiene como objeto la parte de diseño de la web.</p>
    </section>
```

Artículos (article)

Los artículos se refieren a elementos textuales con contenido independiente unos de otros. Siguiendo con la analogía anterior, los productos de cada sección, o las noticias, irían incluidas en artículos separados.

```
    <section>
        <h1>Nacional</h1>
        <article>
```

```
            <h2> La prima de riesgo vuelve a rozar mínimos históricos </h2>
            <p> El mercado de deuda ha premiado la evolución económica española...</p>
        </article>
        <article>
            <h2> El paro vuelve a decrecer por quinto mes consecutivo</h2>
            ...
        </article>
    </section>
    <section>
        <h1>Internacional</h1>
        <article>
            <h2> España recupera el representante en el Banco Central Europeo</h2>
            <p> Tras varios meses de ...</p>
        </article>
    </section>
```

Secciones "laterales" (aside)

Son partes de un documento, artículo o sección que contienen material relacionado, pero un tanto lateral y no crítico al contenido de la parte al que se asocian. Por ejemplo, podrían ser enlaces a documentos relacionados, o material adicional:

```
<aside> Enlaces de interés... </aside>
```

En en enlace indicado a continuación se puede encontrar una explicación detallada de cómo usar estos elementos semánticos para estructurar un documento html:

https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction_to_HTML/Document_and_website_structure#Enter_HTML5_structural_elements

Bloques de texto

El texto en HTML5 suele enmarcarse en elementos párrafo, *p*: * Este es un texto. *. Separan un bloque de texto de otros elementos, incluyendo otros párrafos. Puede contener más de una frase. Aún estando dispuesto el texto en varias líneas de una marca * *, todo el texto que lo contiene aparecerá continuo. Para romper la linealidad del contenido, se utiliza la marca *
* (nueva línea).

Ejercicio:

Introduce varios párrafos en holamundo.html.

La marca *blockquote* permite indicar que el texto, organizado en párrafos, está citado de otra fuente:

```
<blockquote> En un lugar de la Mancha cuyo nombre no quiero recordar... </blockquote>
```

Para indicar el origen de la referencia se emplea la marca *cite:*

```
<p> Frase original de <cite> Cervantes </cite> </p>
<!-- ó -->
```

```
<blockquote> En un lugar de la Mancha cuyo nombre no quiero recordar... Frase original de
<cite> Cervantes </cite> </blockquote>
<!-- ó también se puede meter como atributo de blockquote -->
<blockquote cite="Frase original de Miguel de Cervantes"> En un lugar de la Mancha cuyo
 nombre no quiero recordar... </blockquote>
<blockquote cite="www.ugr.es"> Descubierto medicamento infalible para Covid19 </blockquote>
```

Para indicar al navegador que el texto se muestre tal cual ha sido incluido en el documento HTML, respetando los espacios en blanco, tabuladores, nuevas líneas, etc. se emplea el elemento *pre*, especialmente útil cuando se va a mostrar código fuente de algún lenguaje de programación (en combinación con la marca *code*.

```
<pre>
  <code>
      % Código en Matlab.
       for i=1:3:100
            m(i,:)= 5* m(i,:);
       end
  </code>
</pre>
```

Otros elementos semánticos

Diálogo (dialog)

Define un cuadro de diálogo o modal nativo.

```
<dialog open>Hola, soy un cuadro de diálogo</dialog>
```

Dirección (address)

Este elemento tiene como objetivo almacenar información de contacto para el autor de un documento, sección o artículo:

```
<address> C/ Periodista Daniel Saucedo Aranda, s/n, 18071, Granada </address>
```

Definición (dfn)

Este elemento representa un término que se va a definir:

```
<p>   <dfn>pájaro</dfn>
      Ave, especialmente si es pequeña. <br/>
      -- RAE
</p>
```

Listas de definiciones

Para realizar una lista con definiciones de términos se emplea el elemento *dl*, en combinación con *dt*, para indicar el término definido, y *dd*, para la definición propiamente dicha:

```
<dl>
    <dt>Perro</dt>
    <dd>Mamífero doméstico de la familia de los Cánidos,...</dd>
    <dt>Gato</dt>
    <dd>Mamífero carnívoro de la familia de los Félidos,...</dd>
</dl>
```

Detalles (details)

Soportado por pocos navegadores, el elemento *details* permite mostrar un texto a demanda del usuario (este podrá ocultarlo o visualizarlo según su interés). Suele emplearse con elemento *summary*, el cual representa un resumen del contenido en *details*:

```
<details>
    <summary>Copyright 2013 DECSAI.</summary>
    <p> - Departamento de Ciencias de la Computación e Inteligencia Artificial</p>
    <p> Todo el contenido de este sitio web es  propiedad del Departamento CCIA</p>
</details>
```

Abreviaturas

Elemento que se emplea para indicar que un texto es una abreviatura o acrónimo. Se emplea en combinación con el atributo global *title*:

```
<p> Mi computadora tiene 6 <abbr title="Gigabytes">GB</abbr>
    de memoria <abbr title="Random Access Memory">RAM</abbr>.</p>
```

Importancia (strong)

Cuando se quiere remarcar que algún texto es crítico o muy relevante, se emplea el elemento *strong*:

```
<p> Lo más destacable de la aplicación es su capacidad de
<strong> aprender de los datos </strong> .</p>
```

El elemento ** solo se debe utilizar para remarcar la importancia del texto, pero no para estilizarlo, es decir, cuando queramos que un texto aparezca en negrita. Para esto último, se deberá utilizar CSS.

Énfasis (em)

Se emplea cuando se desea remarcar enfáticamente algún texto (resaltar palabras que se pronunciarán con énfasis al leer):

```
<p> Las <em> personas </em> también son animales.</p>
```

El elemento ** solo se debe utilizar para poner énfasis en el texto, pero no para estilizarlo, es

decir, cuando queramos que un texto aparezca en cursiva. Para esto último, se deberá utilizar CSS.

Resaltado (mark)

Se emplea cuando se desea destacar texto relacionado con una búsqueda o contexto específico:

```
<p>Resultados de búsqueda: encontrado <mark>HTML5</mark> en la página.</p>
```

El elemento *<mark>* solo se debe utilizar para destacar texto, pero no para estilizarlo, es decir, cuando queramos que un texto aparezca en cursiva. Para esto último, se deberá utilizar CSS.

Citas (cite)

Se emplea, como ya se ha indicado, para ofrecer una fuente de una referencia:

```
<p> <cite>El Capitán Alatriste</cite> de Arturo Pérez Reverte</p>.
```

Elementos multimedia

Figuras (figure)

Elemento semántico que describe una o más imágenes con un título adicional. Este elemento no muestra imágenes sino que actúa como contenedor de cualquier tipo de ellas. Cada imagen puede tener un título mediante la inclusión del elemento *figcaption*.

```
<figure>
        <img src = "ejemplo.jpg" alt = "figura de ejemplo" />
        <figcaption>Texto explicativo de la figura.</figcaption>
</figure>
```

Aunque un *figure* puede contener más de una imagen, solo debe contener un único elemento *figcaption* para el conjunto entero de imágenes. En caso de querer agrupar varias imágenes, cada una con un caption diferente, se pueden anidar elementos *figure*. Adicionalmente, un *figure* puede contener elementos distintos a imágenes, como fragmentos de código para los cuales queremos asignarle un título (*figcaption*).

Audio (audio)

Permite incorporar a un documento HTML un fichero de audio.

```
<audio src="NewYork.mp3" controls>
        <a href = "NewYork.mp3"> Canción New York, New York</a>
</audio>
```

El texto dentro del elemento *audio* solo se mostrará en navegadores que no soporten dicho elemento. En este ejemplo dentro del elemento audio se ofrece un enlace al fichero NewYork.mp3, con objeto de permitir la descarga de este fichero como alternativa al caso de que el navegador no muestre el reproductor de audio y reproduzca el fichero.

Algunos atributos interesantes de este elemento son:

autoplay: en el momento en que el navegador cargue el documento, se reproducirá el audio.

controls: presenta un reproductor de audio simple.

preload: el fichero de audio se comenzará a cargar en memoria en el momento en que la página se cargue en el navegador, pero no comenzará su reproducción.

src: indica la dirección del fichero.

Alternativamente al atributo src se emplea el elemento *source*, que puede usarse con los elementos *audio* y *video*. En el caso de especificar varios elementos *source*, el navegador reproducirá el primer archivo que soporte.

```
<audio>
        <source src= "Cancion1.mp3" >
        <source src= "Cancion2.mp3">
</audio>
```

Vídeo (video)

Permite incluir vídeos en el documento directamente, sin emplear ningún plugin:

```
<video src="NewYork.mp4" controls>
        Tu navegador no soporta el elemento vídeo.
</video>
```

Igualmente se suele emplear el elemento source.

Canvas (canvas)

Habilita una zona de la pantalla para gráficos que se controlan mediante programas (por ejemplo en JavaScript). Por sí sólo, este elemento hace poco.

```
<canvas id = "canvas1" width = "300" height = "200">
        Esto es un ejemplo del elemento canvas.
</canvas>
```

iframe

El elemento <iframe> se usa para incrustar documentos o contenido externo dentro de una página web. Es muy útil para incluir vídeos, mapas, documentos PDF, otras páginas web, etc.

```
<iframe width="560" height="315"
        src="https://www.youtube.com/embed/dQw4w9WgXcQ"
        title="Reproductor de vídeo de YouTube" allowfullscreen
        referrerpolicy="no-referrer"
        sandbox
</iframe>
```

El *allowfullscreen* permite visualizar el vídeo a pantalla completa. Al incluir contenido de

terceros, es recomendable limitar los permisos con *sandbox* y referrerpolicy. El primer atributo se usa para limitar las acciones que puede realizar el contenido embebido, aumentando la seguridad, ya que se pueden ejecutar scripts maliciosos, se roben datos del usuario (cookies, sesiones) o se redirija a los usuarios a sitios fraudulentos. Si aparece el atributo sandbox sin valores ninguno, ofrece la máxima seguridad. El segundo atributo, *referrerpolicy*, evita que el sitio embebido obtenga información del usuario.

Ejercicio:

Escribe un documento HTML que muestre el plan de estudios de un grado, organizado por cursos. En cada uno de ellos aparecerán las asignaturas e información básica sobre ellas. Utiliza todos los elementos semánticos vistos anteriormente para estructurar internamente el documento.

Formularios en HTML5

Los formularios nos permiten captar entradas del usuario y pasar los datos leídos al servidor para su procesamiento posterior. Se especifican con el elemento *form* (*<form> … </form>*). Los dos atributos obligatorios son los siguientes:

action: sirve para especificar dónde enviar los datos del formulario cuando se han captado todas las entradas y se haya hecho click en el botón de envío. Por ejemplo, *<form action="procesar.php" </form>*, indica que, una vez que se tengan todos los datos, se envíen al programa en PHP procesar.*php* para su procesamiento. Finalmente, y una vez ejecutado éste, se envía la salida al cliente (navegador) y la página se recarga. También podríamos hacer que los datos del formulario se enviaran por correo mediante

```
<form action="mailto:direccion@dominio.com?&subject=Asunto" enctype="text/plain">
    ...
</form>
```

method: indica el tipo de método HTTP que se empleará para enviar los datos del formulario. Puede tomar los valores *get* y *post:*

```
<form action="procesar.php" method="get"> ... </form>
```

Si es *get* el que se usa, los datos del formulario se añadirán al final del URL que contiene el programa que procesará los datos y mostrará la salida:

http://www.ejemplo.es/procesar.php?mi_nombre=Juan

Con *post*, los datos del formulario se envían igualmente, pero no se muestran.

Otros atributos interesantes pueden ser:

name: nombre del formulario. También se puede identificar mediante el atributo global *id*.

autocomplete: que toma los valores *ON* u *OFF* y sirve para indicar la activación o no del autocompletado del formulario.

Los formularios se componen de elementos como campos de texto, botones, listas, etc., pero antes de verlos, presentaremos un par de elementos que nos permiten organizar y dichos componentes y describir ese agrupamiento: el primero de ellos es *fieldset*, que nos permite agrupar un conjunto de

componentes gráficos mediante un recuadro:

```
<fieldset> Conjunto de elementos que componen el formulario </fieldset>.
```

El segundo es el elemento *legend*, permite añadir una leyenda al elemento *fieldset:*

```
<fieldset>
  <legend> Datos personales </legend> Conjunto de componentes del formulario agrupados
</fieldset>.
```

Información adicional sobre el elemento *form* se puede encontrar en la web http://www.w3.org/TR/html-markup/form.html.

La mayoría de los componentes de entrada de datos que podemos ver en un formulario se basan en el elemento *input*. Mediante los diferentes valores que tome el atributo *type* podremos determinar el tipo exacto de componente que queremos.

El más común de estos componentes es la caja de texto (text box):

```
<form action="procesar.php" method="post">
        <label for="campoDeTexto1">Nombre:</label>
        <input type = "text" id = "campoDeTexto1" name="campoDeTexto1" value = "Texto por defecto" />
</form>
```

Indicando *text* como valor del atributo *type*, se está creando un campo de texto. El atributo *id* nos permitirá identificarlo. Por último, si queremos que aparezca un texto por defecto en el interior del campo de entrada, deberemos asignarle un valor al atributo *value* y si ese texto no se desea que se modifique se pondrá la palabra *readonly*. El elemento *label* establece una etiqueta que se asociará al elemento input cuyo valor del atributo *id* sea el mismo que el contenido en el atributo *for* de *label*, y que se mostrará el navegador junto al componente.

Por otro lado, el elemento *input* tiene otro atributo, *name*, del que también disponen el resto de elementos que representan controles gráficos, que almacenará el nombre del componente y por el cual se podrá referir cualquier programa que procese el formulario a posteriori. De cualquier forma, se puede emplear para este fin tanto el atributo *id* como *name*.

Se puede limitar la longitud del campo de texto, en caracteres, mediante el atributo *size*, y limitar el número de caracteres a introducir empleando *maxlength*.

El atributo *placeholder* hace las veces de pista para el usuario en relación con la entrada que tiene que introducir. Aparece en el interior del campo, con una intensidad atenuada y en el momento en que el usuario pone el cursor en el campo de texto, esta sugerencia desaparece: *<input type="text" placeholder="1 letra + 4 números"/>*. Si se usa en combinación con el atributo *value*, no se aplica.

Si al atributo *type* se le asigna el valor *password*, entonces estaremos creando un campo de texto en el que los caracteres tecleados se sustituyen por asteriscos.

Cuando se desea introducir más de una línea, en lugar de un campo de texto, debemos emplear un área de texto (text area), para lo cual usamos el elemento *textarea*:

```
<form action="procesar.php" method="post">
        <label for="textArea1">Área de texto</label>
        <textarea  id = "textArea1" rows="5" cols="80">  Texto por defecto </textarea>
</form>
```

Para este elemento se especifica el número de líneas de texto que tendrá el recuadro (atributo

rows), así como el número de columnas (*cols*). Lo habitual es que sean 80 caracteres. Seguidamente aparecerá el texto que deseamos que aparezca por defecto.

Las listas desplegables son muy comunes en las páginas web. Permiten al usuario seleccionar una opción de una lista de ellas. En HTML5 se implementan mediante el elemento *select* en combinación con *option* para indicar cada una de las opciones a elegir:

```
<form action="procesar.php" method="post">
        <label for="lista1">Lista desplegable </label>
        <select id = "lista1" >
                <option value = "1">Uno</option>
                <option value = "2">Dos</option>
                <option value = "3">Tres</option>
                <option value = "4">Cuatro</option>
        </select>
</form>
```

El elemento lista vendrá identificado por el atributo *id* y en su interior aparecerán tantos elementos *option* como opciones queramos que aparezcan en el desplegable. Cada una de ellas tendrá un valor diferente, que será el que se envíe al programa encargado de su procesamiento cuando se elija una. Seguidamente se indica el texto de la opción.

Parecido a este componente anterior, tendríamos la combinación de los elementos *datalist* e *input* con *text* como tipo de entrada esperada. *datalist* contiene un conjunto de opciones que estarán disponibles en el momento en que el usuario introduzca un carácter en el campo de texto, a modo de sugerencias (autocomplete list).

```
<form action="procesar.php" method="post">
        <datalist id="listaSuperheroes">
                <option value="Superman">
                 <option value="Spiderman">
                <option value="Hulk">
        </datalist>
        <label for="campoDeTexto1">Lista de superherores </label>
        <input type="text" id="campoDeTexto1" list="listaSuperheroes">
</form>
```

En este ejemplo, creamos un elemento *datalist* identificado como *listaSuperheroes,* en el que introducimos la lista de sugerencias disponibles, mediante el elemento *option* (aunque no aparece en el ejemplo, los elementos *option* pueden estar agrupados por medio de *optgroup*). Seguidamente creamos un campo de texto, pero empleamos el atributo *list* para indicar que emplee dicha lista.

En ocasiones tendremos situaciones en las que algún tipo de información pueda ser verdadera o falsa (está o no seleccionada). El componente que nos permite seleccionar de forma exclusiva una alternativa es una casilla de verificación (*checkbox*), nombre que se asigna como valor al atributo *type* del elemento input:

```
<form action="procesar.php" method="post">
        <label for = "checkHuevos">Huevos: </label>
        <input type = "checkbox" id = "checkHuevos" value = "huevos" />
        <label for = "checkJamon">Jamon: </label>
        <input type = "checkbox" id = "checkJamon" value = "jamon" />
</form>
```

Hay que indicar que, a pesar de que puedan ir varias juntas, las casillas de verificación son independientes unas de otras. Al seleccionar cada una de ellas, el valor asignado al atributo *value* será el que se envíe al programa correspondiente para su proceso (este valor no será visto por el usuario). En este caso, es necesario introducir un elemento *label* por cada casilla, de tal forma, que se asocie a ella mediante el valor del atributo *for*, que debe coincidir con el del *id* de la correspondiente casilla a la que se asocia. El texto que verá el usuario se indica a continuación.

Otro componente de los formularios habituales son los botones circulares (*radio buttons*), los cuales pueden parecer similares a las casillas de verificación, pero difieren en varios aspectos: en primer lugar, estos botones circulares siempre aparecen en grupo y sólo se puede seleccionar uno del mismo; siempre debe haber seleccionado uno por defecto; cada *radio button* debe tener un atributo *name* con el mismo valor para indicar que están relacionados. Se crean mediante el valor *radio* en el atributo *type* del elemento *input*.

```
<form action="procesar.php" method="post">
        <input type = "radio" id = "radiobutton1" name="menu" value = "huevos"  checked/>
        <input type = "radio" id = "radiobutton2" name="menu" value = "habas" />
        <input type = "radio" id = "radiobutton3" name="menu" value = "jamon" />
</form>
```

El botón circular que se considere seleccionado por defecto se especificará por medio de *checked*.

Se puede incluir una barra de progreso con el elemento *progress*:

```
<form action="procesar.php" method="post">
        <progress value="70" max="100"></progress>
</form>
```

Y también para mostrar un indicador con *meter*:

```
<form action="procesar.php" method="post">
        <label for="disk_c">Uso unidad C:</label>
        <meter id="disk_c" value="2" min="0" max="10">2 de 10</meter><br>

        <label for="disk_d">Uso unidad D:</label>
        <meter id="disk_d" value="0.6">60%</meter>
</form>
```

El contenido del elemento se usa como texto alternativo si el navegador no lo soporta. No se muestra habitualmente porque el navegador renderiza la barra visualmente.

En HTML5 hay tres tipos de botones: los estándares (*button*) que, en conjunción con JavaScript,

lanzan la ejecución de alguna acción en el cliente; los de envío (*submit*), que se emplean en la programación con servidor, empaquetan todos los datos del formulario y se envían a algún programa situado en un servidor; y los de reinicio (*reset*), que hace que los datos del formulario se asignen a los valores por defecto. Se pueden crear mediante el elemento input:

```
<form action="procesar.php" method="post">
        <input type = "submit" id = "button1" value = "Procesar datos"/>
        <input type = "reset" id = "button2" value = "Reiniciar formulario" />
        <input type = "button" id = "button3" value = "Calcular trayectoria" />
</form>
```

Un atributo interesante para dirigir el foco a un componente del formulario es *autofocus*, el cual suele situarse en el primer componente del formulario, independientemente del tipo de éste.

required se utiliza en aquellos componentes que el usuario debe rellenar obligatoriamente antes de enviarse los datos.

Otros valores para el atributo *type* del elemento *input*, que determinan el tipo de entrada que se solicita al usuario son los siguientes:

color: permite al usuario seleccionar un color utilizando los formatos estándar para identificar colores (mediante nombres o valores hexadecimales). El navegador podría mostrar un paleta para seleccionar el color correspondiente: <*input type="color" id="color1" />*.

date: se indica que se requiere una fecha. Unos navegadores mostrarán un simple campo de texto (formato yyyy-mm-dd), mientras que otros un calendario, donde se puede seleccionar la fecha correspondiente (<*input type="date" id="fecha1" />*). De forma parecida funciona la variedad month, pero sólo espera el formato yyyy-mm-dd.

time: en este caso se espera una hora en el formato hh:mm.

datetime: en este caso se espera una entrada de una fecha y una hora a la vez. De nuevo dependerá del navegador el método de entrada que se emplee, pero siempre siguiendo el formato: yyyy-mm-ddThh:mm+ff:gg. (la T es simplemente un indicador de separación entre la fecha y la hora).

email: utilizado para incluir una dirección de correo electrónico.

number: el dato requerido debe ser un número. Este componente se suele apoyar de algún tipo selector, por ejemplo, los cursores ascendente y descendente. Se puede especificar el rango en que debe encontrarse el valor introducido (atributos *max* y *min*). Otro tipo de entrada relacionado es *range,* en el que se muestra una barra de desplazamiento que permite seleccionar un valor numérico en el rango (en vez de tener que ser introducido manualmente).

search: diseñado como campo de texto con el objetivo de que el usuario introduzca una consulta para un buscador (<*input type="search" id="busqueda1" />*). No se efectúa ninguna búsqueda.

tel: entrada que representa un número de teléfono.

url: para indicar una dirección web.

Más información sobre el elemento *input*, con todos los valores para el atributo *type* definidos en HTML5 se puede encontrar en http://www.w3.org/TR/html-markup/input.html. En https://oku.ozturkibrahim.com/WebProgramming/examples//ch03/fig03_01/NewFormInputTypes.html se tiene también una descripción de los diferentes tipos entradas, con ejemplos en línea.

Ejercicio:

Escribe un documento HTML que realice la inscripción para un congreso, con las siguientes

características:

- Contenga una cabecera donde incluya el título y fecha del congreso (*header*).

Incluya una sección de contacto, vía correo electrónico (*aside*).

- Tenga una sección con los organizadores del congreso (*footer*).

- Aparezca una sección con una tabla con los diferentes precios del congreso (*section*) (para estudiantes, y profesionales y registro temprano, tardío y durante el congreso.

- Albergue una sección donde se inserte el formulario de recogida de datos (*section*).

- Los datos que se deberán tomar del usuario son los siguientes: nombre, apellidos, correo electrónico, teléfono, fecha de nacimiento, número de acompañantes, ¿es tu primera asistencia al congreso? (radiobutton), ¿cómo conociste el evento? (input + datalist), botón para registrarse y para resetear el formulario.

Validadores de HTML5

Es conveniente verificar que nuestro documento no tiene ningún error que lo validemos. Existen herramientas en línea para este fin, como, por ejemplo, http://validator.w3.org.

Bibliografía

http://es.wikipedia.org/wiki/HTML

http://www.w3schools.com/html/default.asp

http://www.w3.org/TR/html5/

https://www.dummies.com/category/articles/general-programming-web-design-33610/