

Tema 2: Elementos de contenido

Encabezados, párrafos y listas

Encabezados

Los elementos de encabezado también se usan para mostrar texto en la página, pero lo hacen de forma destacada, ya que lo resaltan visualmente. Su uso principal es organizar el contenido dividiéndolo en secciones.

```
<h1>Esto es un elemento de encabezado</h1>
```

Existen seis niveles de encabezados, desde `<h1>` hasta `<h6>`. El nivel `<h1>` representa el encabezado más importante, mientras que `<h6>` es el de menor relevancia.

Cuando se utiliza un `<h1>` en una página, el navegador lo interpreta como el título o sección principal, y por eso lo muestra con un tamaño de fuente mayor. Los encabezados de nivel `<h2>` aparecen un poco más pequeños, y así van disminuyendo progresivamente hasta llegar al `<h6>`, el cual se muestra con el tamaño más reducido de todos los encabezados [18].

```
<h1>Este es el más grande</h1>
<h2>Este es un poco más pequeño</h2>
<h3>Este es aún más pequeño</h3>
<h4>Cada vez son más pequeños</h4>
<h5>Este ni siquiera es tan grande</h5>
<h6>Bastante pequeño ahora</h6>
```

Párrafos

Los párrafos son la forma más común de dividir un texto extenso en secciones más pequeñas y con sentido. Al hacerlo, el contenido se separa visualmente con espacios (interlineado), lo que mejora su presentación y facilita la lectura, evitando que el texto resulte abrumador.

Este tipo de estructura suele utilizarse en textos largos, como artículos, lecciones o explicaciones, donde es importante que la información esté organizada de manera clara. Por ejemplo, en plataformas como Hexlet o Code Basics, los contenidos se estructuran en párrafos para que el lector pueda entenderlos más fácilmente.

Cada párrafo representa una idea o unidad de significado, lo que también ayuda a localizar información específica dentro de un texto largo [19].

Por ejemplo.

```
<p>Primer párrafo</p>  
<p>Segundo párrafo</p>
```

Listas

En HTML5 se definen tres tipos de listas:

- **Listas numeradas:** se delimitan con las etiquetas ` ... ` y sirven para crear listas con números.
- **Listas con viñetas:** se delimitan con las etiquetas ` ... ` y sirven para crear listas que utilizan símbolos o viñetas.
- **Listas de definiciones:** se delimitan con las etiquetas `<dl> ... </dl>` y sirven para crear definiciones de términos.

Para delimitar cada uno de los elementos de las dos primeras listas se usan las etiquetas ` ... `.

En cambio, las definiciones usan dos tipos de etiquetas distintas:

- `<dt> ... </dt>` delimitan los términos.
- `<dd> ... </dd>` delimitan las definiciones.

A modo de ejemplo, probemos las siguientes listas [20]:

```
<p>Los contenidos de este libro son:</p>
<ol>
<li>Introducción al Desarrollo Web</li>
<li>HTML5</li>
<li>CSS3</li>
<li>...</li>
</ol>
<p>Las tecnologías más utilizadas en el front-end son:</p> <ul>
<li>HTML5</li>
<li>CSS</li>
<li>JavaScript</li>
<li>jQuery</li>
</ul>
<p>Las partes de un documento html son:</p>
<dl>
<dt>Cabecera</dt>
<dd>Incluye información sobre el documento.</dd>
<dt>Cuerpo</dt>
<dd>Incluye el contenido del documento</dd>
</dl>
Las listas se pueden anidar unas dentro de otras:
<ol>
<li>Primer elemento </li>
<li>Segundo elemento </li>
<ul>
<li>Elemento de lista desordenada anidada</li>
</ul>
<li>Definiciones: </li>
<dl>
<dt>Término</dt>
<dd>Definición del término 1</dd>
</dl>
</ol>
```

Vínculos e imágenes

Imágenes

En una página web podemos insertar imágenes en formato fotográfico, también conocido como mapa de bits. Existen distintos tipos de formatos para guardar las imágenes, aparte del archiconocido jpg. El problema de este formato es que perderemos calidad dependiendo del nivel de compresión utilizado para que el fichero ocupe menos. Por eso existen otros formatos que quizás sean más convenientes:

- .png : es un formato de compresión sin pérdida que además soporta transparencias. Eso quiere decir que no perderemos calidad al comprimirlo (aunque ocupará más que un jpg) y que podemos representar iconos o imágenes con zonas transparentes. Es muy buena alternativa para cualquier tipo de imagen que no sea una fotografía. Para eso utilizaremos jpg.
- .jpg: como ya he comentado, es un formato de compresión con pérdida que no soporta transparencias. Se usa normalmente para fotografías.
- .gif: es un formato gráfico sin pérdida, pero limitado a 256 colores. Por ello ocupa muy poco y es idóneo para imágenes básicas, como los iconos. Además, soporta transparencias y animaciones.

Al igual que ocurría con el editor de HTML, tienes varias opciones de software disponibles para manipular las imágenes de tu sitio web. Dejando de lado las típicas herramientas comerciales, como Photoshop, puedes utilizar el editor GIMP, que es muy similar y de código libre. Además, para convertir entre diferentes formatos puedes utilizar ImageMagick (<http://www.imagemagick.org/script/index.php>).

Una vez que ya tenemos la imagen editada, la insertamos en nuestro HTML a través del elemento img. Esta etiqueta tiene un atributo src donde indicamos la ruta a la imagen. Normalmente es una ruta relativa al directorio donde se encuentra el archivo HTML, pero también puede ser una URL de una en otra página web.

Supongamos que en el directorio donde estamos sirviendo los archivos con Apache creamos una carpeta llamada img donde guardaremos todas las imágenes. En ese caso, el enlace sería ``. La otra opción es enlazar directamente con una imagen que esté en otro servidor, por ejemplo:

```

```

En este caso hemos añadido el atributo alt, que contiene una descripción textual que se usa para los navegadores utilizados personas con deficiencias visuales que no pueden ver correctamente la imagen. En estos casos, el navegador “lee” el contenido del HTML y utiliza esta etiqueta alternativa para describir la imagen.

Como ejercicio vamos a incluir una imagen en la barra lateral de nuestro ejemplo. Crea con tu editor favorito una imagen de unos 200 px de ancho y 500 de alto (o busca una cualquiera en Google). Añádela dentro del elemento `<aside>` de nuestra página de ejemplo:

```
<aside></aside>
```

Y modifica un poco la CSS, tal y como se muestra a continuación, para que se vea correctamente (ya explicaré los detalles en el próximo capítulo). El resultado debería ser similar al que aparece en la Figura.

```
article {  
width: 100%;  
background-color: lemonchiffon;  
min-height: 80%;  
}  
aside {
```

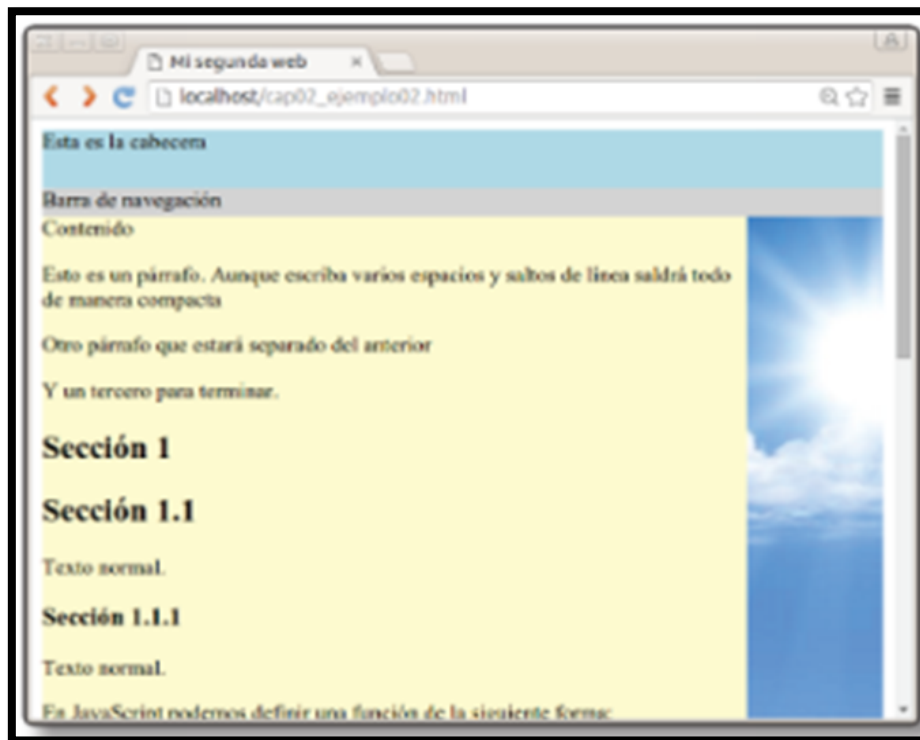


Figura 20 Resultado de añadir una imagen

Vínculos

La principal característica de la Web es que se basa en el hipertexto. Es decir, documentos enlazados unos con otros que nos permiten explorar de manera no secuencial. Estos enlaces, también conocidos como hipervínculos o hiperenlaces, nos van a permitir movernos a otros documentos externos o a distintas partes de un documento.

Para añadir un enlace usamos la etiqueta `<a>` (anchor, ancla):

```
<a href="destino">Título del enlace</a>
```

Donde el campo destino puede ser:

- Una dirección absoluta: `http://www.ra-ma.es`
- Una dirección relativa: `foros/general.php`
- Un correo electrónico: `mailto:jareciog@fdi.ucm.es`

Un enlace puede representarse como una imagen. Para ello incluiremos el elemento `` dentro de la etiqueta `<a>...`:

```
<a href="http://www.ra-ma.es/">  

```

El destino del enlace se da comienzo del documento. Pero también podemos usar enlaces para saltar a distintos puntos de una página web, tanto la actual como cualquier otra página externa. Para eso lo primero es definir los marcadores de destino, es decir, los puntos exactos del documento que serán el destino del enlace. En este caso utilizamos la etiqueta `<a>` pero sin contenido ni atributo href. En su lugar emplearemos el atributo name donde especificamos una etiqueta que identifica el marcador de destino:

```
<a name="marcador"/>
```

Para definir un enlace a esos marcadores destino, indicamos su nombre precedido por el símbolo #:

```
<a href="#marcador">Ir al marcador de esta misma página</a>  
<a href="pagina.html#marcador">Ir al marcador del documento pagina.html</a>
```

Para practicar los enlaces vamos a incluir uno en la barra de navegación de nuestra página. Si has ido copiando los distintos fragmentos de código de este capítulo, la parte central (elemento article) será bastante larga y al final del todo estarán los ejemplos sobre listas y descripciones. Lo que haremos será añadir un marcador de destino en el código con descripciones, y luego un enlace a este desde la barra de navegación. Para permitir volver fácilmente al principio de la

página añadiremos otro marcador en la cabecera y un enlace a este punto en el pie de página. El código quedaría así:

```
...  
<header><a name="arriba"/>Esta es la cabecera</header>  
<nav><a href="#definiciones">Ir a definiciones</a></nav>  
<div id="principal">  
<article>  
...  
<p>Las partes de un documento html son:</p>  
<a name="definiciones"/>  
<dl>  
<dt>Cabecera</dt>  
<dd>Incluye información sobre el documento.</dd>  
<dt>Cuerpo</dt>  
<dd>Incluye el contenido del documento</dd> </dl>  
...  
</article>  
<aside>  
</aside>  
</div>  
<footer><a href="#arriba">Volver arriba</a></footer>  
...
```

Las Figuras muestran el resultado de añadir el enlace tanto a la cabecera como al pie de la página.

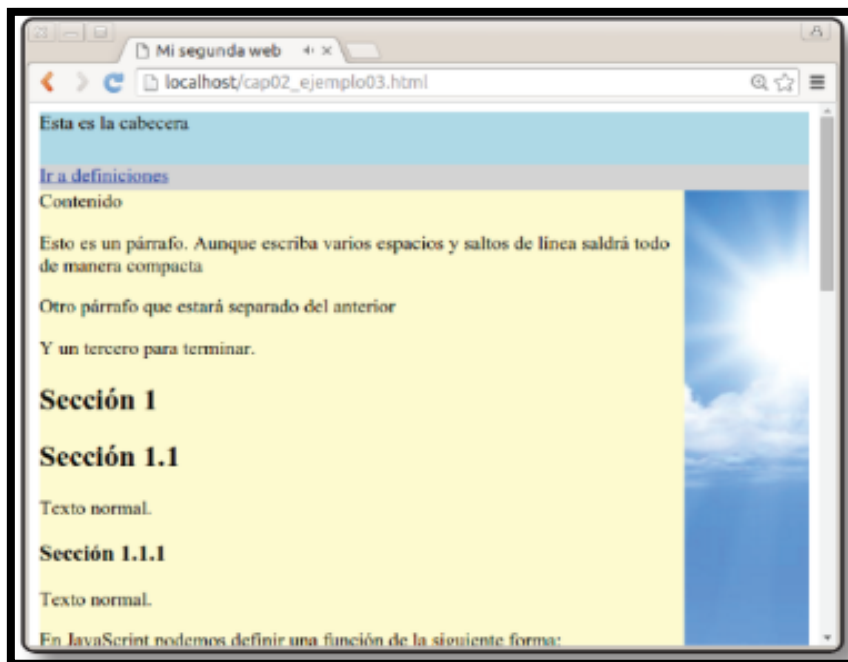


Figura 21 Resultado de añadir un enlace a la barra de navegación

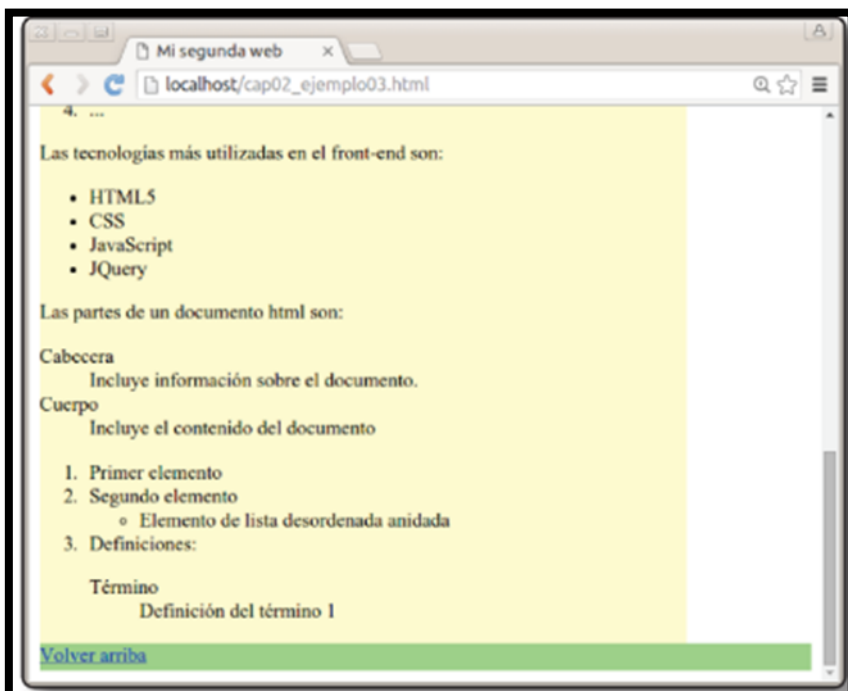


Figura 22 Resultado de añadir un enlace en el pie de página

Tablas y divisiones

Tablas

Las tablas permiten presentar información tabular, en filas y columnas, con cabeceras. Cada elemento de la tabla puede ser simple, o, a su vez, ser otra agrupación de filas y de columnas, cabeceras y pies de tabla, subdivisiones, cabeceras múltiples y otros elementos complejos.

Como las tablas permiten un control muy detallado, a veces se usan para organizar la estructura general de una página web. Sin embargo, esto es algo que no se recomienda en absoluto. Para estructurar las páginas debemos utilizar etiquetas div.

Una tabla se define mediante el elemento `<table>`. Dentro del elemento `<table>` se definen las filas con `<tr>` (table row). Además, para cada fila, se define cada celda de la tabla con los elementos `<td>` (table data). Dentro de este elemento se puede poner cualquier otro elemento HTML (incluida otra tabla, aunque no se recomienda). Veamos un ejemplo:

```
<table>

<tr>

<td><strong>Curso</strong></td>

<td><strong>Horas</strong></td>

<td><strong>Horario</strong></td>

</tr>


<tr>

<td>CSS</td>

<td>20</td>

<td>16:00 - 20:00</td>

</tr>


<tr>

<td>HTML</td>

<td>20</td>

<td>16:00 - 20:00</td>

</tr>
```

Normalmente necesitaremos combinar celdas (práctica llamada en inglés *spanning*) para hacer que varias celdas de una tabla se unan para formar una sola. Para agrupar o combinar varias celdas de una fila se usa el atributo `colspan`, indicando el número de celdas que se combinarán.

```
<table>

<tr>

<td colspan="3">Tecnologías del Front-End</td>

</tr>

<tr>

<td>HTML5</td>

<td>CSS</td>

<td>JavaScript</td>

</tr>

</table>
```

Sin embargo, para agrupar columnas utilizamos el atributo `rowspan`, indicando también en su valor el número de celdas que queremos combinar.

```
<table>

<tr>

<td rowspan="3">Tecnologías del Front-End</td>

<td>HTML5</td>

</tr>

<tr>

<td>CSS</td>

</tr>

<tr>

<td>JavaScript</td>

</tr>

</table>
```

Otros elementos de las tablas

Las tablas pueden tener un título o leyenda de la tabla, definido mediante el elemento `<caption>`. Es un texto opcional que se muestra fuera de la tabla (por defecto, arriba). Se suele poner justo después de `<table>`. También hay un tipo especial de fila que se crea con el elemento `<thead>` y que representa la fila que hace de cabecera de la tabla. En este caso, las celdas se definen con el elemento `<th>` (table header). En este caso también es necesario utilizar el elemento `<tbody>` para separar la cabecera del cuerpo de la tabla. Utilizar estos elementos nos permitirá dar un formato específico a la cabecera más fácilmente.

Por último, también se pueden acabar la tabla con un pie de tabla, definido con el elemento `<tfoot>`. Curiosamente, este elemento tiene que definirse antes del elemento `<tbody>` [21].

```
<table>
<thead>
<tr>
<th>Curso</th>
<th>Horas</th>
</tr>
</thead>
<tfoot>
<tr>
<td>Total</td>
<td>180 horas</td>
</tr>
</tfoot>
<tbody>
<tr>
<td>HTML5</td>
<td>100 horas</td>
</tr>
<tr>
<td>CSS</td>
<td>80 horas</td>
```

Divisiones

La etiqueta `<div>` es un elemento de tipo bloque en HTML5 que se emplea como contenedor para agrupar contenido dentro de una página web. Puede incluir una variedad de elementos HTML como texto, imágenes, enlaces, formularios, entre otros.

Su función principal es estructurar y organizar los elementos de la página para facilitar una presentación ordenada y visualmente atractiva. Además, permite aplicar estilos CSS que modifican su apariencia y comportamiento dentro del sitio.

Es relevante señalar que `<div>` no tiene un significado semántico propio; se trata de un contenedor genérico. Por esta razón, siempre que sea posible, se recomienda utilizar etiquetas semánticas más específicas como `<header>`, `<nav>`, `<article>`, `<footer>`, etc., según el tipo de contenido que se desee representar.

La estructura básica de esta etiqueta es la siguiente [22]:

```
<div>  
    Contenido a agrupar  
</div>
```

Ejemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo de uso de la etiqueta div</title>
    <style>
      .container {
        width: 80%;
        margin: 0 auto;
        text-align: center;
      }
      .header {
        background-color: lightgray;
        padding: 20px;
      }
      .main-content {
        background-color: white;
        padding: 40px;
      }
      .footer {
        background-color: lightgray;
        padding: 20px;
        text-align: center;
      }
    </style>
  </head>
  <body>
    <div class="container">
      <div class="header">
        <h1>Bienvenido a mi sitio web</h1>
      </div>
      <div class="main-content">
        <p>Este es el contenido principal de mi sitio web.</p>
        
        <p>Aquí puedes encontrar más información sobre mis servicios y productos.</p>
      </div>
      <div class="footer">
        <p>Copyright &copy; 2021 - Todos los derechos reservados</p>
      </div>
    </div>
  </body>
</html>
```

Ejercicios resueltos

Revisa con la ayuda de tu profesor las cápsulas de código usando el programa **Visual Studio Code**. Práctica y modifica el código. Resuelve los mini-retos.

```
<!--
2.2.1 Encabezados, párrafos y listas
Idea clave: La jerarquía visual y semántica se logra con <h1>...<h6>, <p>,
<ul>, <ol>.
-->
<!DOCTYPE html>
<html lang="es">
<head><meta charset="UTF-8"><title>Texto y listas</title></head>
<body>
  <h1>Tema principal (h1)</h1>
  <h2>Subtema (h2)</h2>
  <p>Este párrafo introduce el subtema con claridad y concisión.</p>

  <h3>Lista no ordenada</h3>
  <ul>
    <li>Concepto A</li>
    <li>Concepto B</li>
    <li>Concepto C</li>
  </ul>

  <h3>Lista ordenada</h3>
  <ol>
    <li>Paso 1</li>
    <li>Paso 2</li>
    <li>Paso 3</li>
  </ol>
</body>
</html>
<!--
Mini-reto: Anida una lista dentro de otra para detallar subpasos.
Error común: Saltar niveles de encabezados (de h1 a h4 sin h2/h3).
-->
```