

<hr>

Otra forma posible de separar el texto es utilizar una línea visible horizontal.

Aparición: sin etiqueta de cierre (V).

Atributos: %attrs, align, noshade, size y width.

 <i> <big> <small> <s> <strike> <tt> <u>

Definición: son los llamados estilos de fuente.

b: negrita, i: itálica o cursiva, big: tamaño grande, small: tamaño pequeño,

s y strike: texto tachado mediante una línea a media altura,

tt: texto con apariencia de teletipo, u: subrayado con una línea por debajo.

Aparición: las etiquetas de apertura y de cierre son obligatorias.

Atributos: %attrs.

Se trata de elementos de línea, por tanto, no hacen salto de línea y se pueden introducir dentro de un párrafo <p>. También pueden contener a otros combinándose entre sí.

Definición: sirve para establecer el tamaño, el color y la fuente para el texto, está desaprobadado.

Aparición: apertura y cierre obligatorios.

Atributos: %i18, %coreattrs, size, color, face.

Ejemplo:

```

<body>
  <h1>Estilos de Fuente</h1> <hr>
  <ul>
    <li><h2>Estilos simples:</h2>
      <b>Negrita</b><i>Cursiva</i><big>Tamaño Grande</big>
      <small>Tamaño pequeño</small><tt>Teletipo</tt>
      <strike>Tachada</strike><u>Subrayado</u>
    </li>
    <li><h2>Tipos de letra:</h2>
      <font face="Verdana" size="4">Font face Verdana</font>
      <font face="Courier New">Font face Courier</font>
    </li>
  </ul>
</body>

```

Las posibilidades usando hojas de estilos son mucho mayores, por ello, todos los estilos de fuente están en desuso.

<pre> Texto con formato previo

Definición: sirve para que el navegador visualice el texto tal y como aparece en el contenido de esta etiqueta, respetando tabuladores, espacios y saltos de línea.

La única excepción es que introduzcamos en el contenido, otros elementos HTML, los cuales se ejecutarán también.

Aparición: etiquetas inicial y final obligatorias.

Atributos: %attrs y width.

- width: indica el ancho del bloque con formato previo.

Ejemplo:

```
<body>
  <h1>Programación en Java</h1>
  <h2>
    <pre>
      class HolaMundo{
        public static void main (String arg[]){
          System.out.println(<&lt;Hola Mundo&&gt;>)
        }
      }
    </pre>
  </h2>
</body>
```

Para evitar que los caracteres "<" y ">" se interpreten como etiqueta debemos utilizar:

< = "<" (*less than*)

> = ">" (*greater than*)

Programación en Java

```
class HolaMundo{
  public static void main (String arg[]){
    System.out.println(<<Hola Mundo>>)
  }
}
```

Figura 2.6: Texto formateado con <pre>

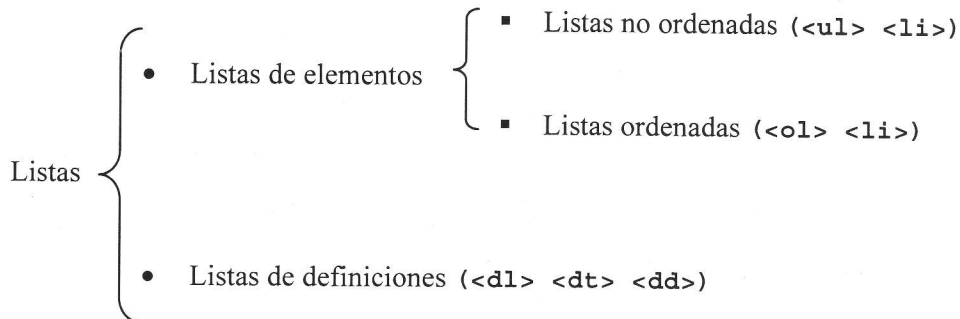
Si no utilizamos <pre> se visualizaría todo el texto en una sola línea.

2.5.2. Listas

Tenemos 2 tipos fundamentales de listas: listas de elementos y listas de definiciones.

A su vez las listas de elementos pueden estar ordenadas o no.

Las listas son un recurso muy común en los textos, como se puede apreciar en este esquema, en el que se han utilizado listas no ordenadas.



• Listas de elementos.

**** Listas de elementos no ordenados.

Definición: se construyen con el elemento **** y su contenido está formado exclusivamente por elementos ****.

Estos elementos **** son los que contienen a su vez el texto.

Aparición: las etiquetas de apertura y cierre son obligatorias.

Atributos: **%attrs**, **type** y **compact**.

- **type:** indica el tipo de símbolo utilizado y puede tomar los valores “disc” (disco), “square” (cuadrado) y “circle” (círculo). Desaprobado.
- **compact:** si aparece indica al navegador que la lista debe compactarse.

**** Listas de elementos ordenados.

Definición: se construyen con el elemento **** y su contenido está formado exclusivamente por elementos ****.

Aparición: las etiquetas de apertura y cierre son obligatorias.

Atributos: **%attrs**, **type** y **start**.

- **type:** indica el tipo de secuencia de lista y puede tomar los valores 1 (números decimales), a (letras minúsculas), A (letras mayúsculas), i (números romanos minúsculas), I (números romanos mayúsculas). Desaprobado.
- **start:** indica el número del primer elemento de la lista. Desaprobado.

**** Elemento de lista.

Definición: objeto de lista, que puede contener otros elementos de bloque como párrafos o directamente el texto.

Aparición: El cierre es opcional.

Atributos: %attrs y value.

- **value:** establece el número del objeto de lista actual. Desaprobado.

Ejemplo:

```
<body>
  <h1>Lenguajes de Marcas</h1> <hr>
  <h2>Lista ordenada</h2>
  <ol>
    <li>HTML</li>
    <li>XHTML</li>
    <li>XML</li>
  </ol>
  <h2>Lista no ordenada</h2>
  <ul>
    <li>HTML</li>
    <li>XHTML</li>
    <li>XML</li>
  </ul>
</body>
```

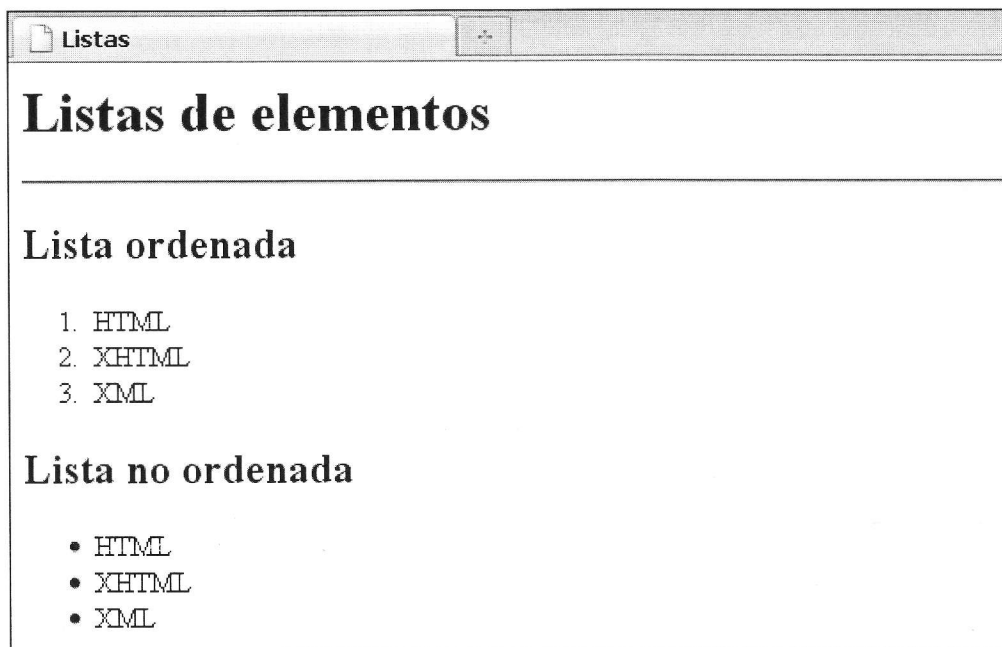


Figura 2.7: Listas con y con

- **Listas anidadas.**

La mayor utilidad de las listas resulta cuando se combinan, es decir, cuando cada elemento de una lista es a su vez otra lista. En estos casos los niveles de anidación deben ser de tipos distintos, para evitar la confusión del lector.

Ejemplo:

```
<body>
<h1>Listas Anidadas</h1> <hr>
<h2>Lenguajes de Marcas</h2>
<ol>
<li><b>Páginas Web</b> <ul><li>HTML</li> <li>XHTML</li> </ul> </li>
<li><b>Dispositivos móviles</b>
  <ul> <li>WML</li> <li>XHTML MP</li> </ul> </li>
<li><b>Sindicación</b> <ul> <li>RSS</li> <li>Atom</li> </ul></li>
<li><b>Documentos</b>
  <ul> <li>RTF</li> <li>TeX</li> <li>MathML</li> </ul> </li>
<li><b>Gráficos</b> <ul> <li>SVG</li> <li>STEP</li> </ul></li>
<li><b>Específicos</b> <ul> <li>SMIL</li> <li>WSDL</li> </ul></li>
</ol>
</body>
```

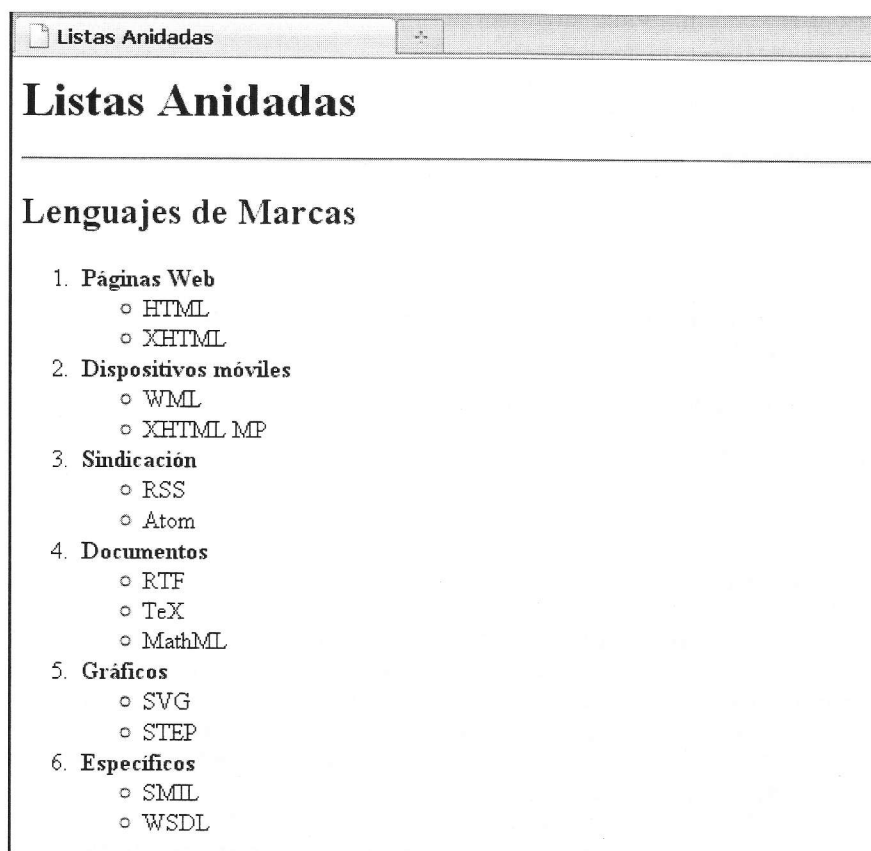


Figura 2.8: Listas anidadas

Actividad 2.15:

Construye una lista anidada, que refleje los módulos que estás estudiando ordenados numéricamente y los contenidos de cada módulo sin ordenar.

- **Listas de Definiciones.**

<dl>

Se trata de listas que no tienen índices ni símbolos de ninguna clase como las anteriores, sino que se componen de dos elementos: el término y la definición, a modo de diccionario. Todo el texto aparece con sangría.

Definición: se construyen con el elemento `<dl>` y contienen solamente elementos de tipo `<dt>` y `<dd>` en número variable.

Aparición: etiquetas de apertura y cierre obligatorias.

Atributos: `%attrs`.

<dt>

Definición: es el término que se quiere definir, normalmente una palabra o un texto breve y se trata de un elemento de línea.

Aparición: etiqueta de cierre opcional.

Atributos: `%attrs`.

<dd>

Definición: es la definición del término anterior, normalmente un texto largo y se trata de un elemento de bloque.

Aparición: etiqueta de cierre opcional.

Atributos: `%attrs`

Ejemplo:

```
<body>
  <h1>Lenguajes de Marcas</h1>
  <dl>
    <dt>HTML</dt> <dd>HyperText Markup Language </dd>
    <dt>XHTML</dt> <dd>Extensible HyperText Markup Language </dd>
    <dt>WML</dt> <dd>Wireless Markup Language </dd>
    <dt>XHTML MP</dt> <dd>Extensible HyperText Markup Language
      Mobile Policy</dd>
    <dt>RSS</dt> <dd>Rich Site Summary</dd>
    <dt>RSS</dt> <dd>Really Simple Syndication</dd>
  </dl>
</body>
```

```

<dt>RTF</dt> <dd>Rich Text Format</dd>
<dt>SVG</dt> <dd>Scalable Vector Graphics </dd>
<dt>SMIL</dt> <dd>Sichronized Multimedia Integration Language
</dd>
<dt>WSDL</dt> <dd>Web Services Description Language </dd>
</dl>
</body>

```

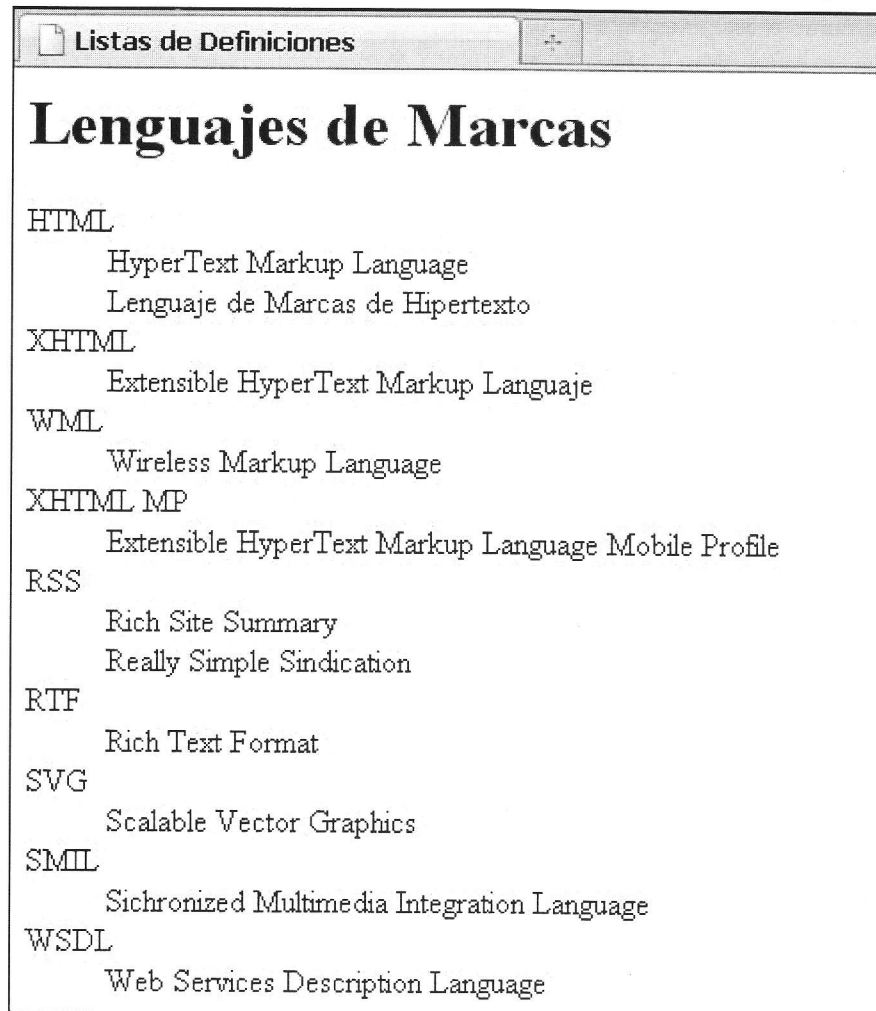


Figura 2.9: Listas de definiciones

Actividad 2.16:

Construye un mini diccionario de inglés-español mediante una lista de definiciones, con los términos: ace, dbms, geek, ide, nerd, nerf, owned, p2p, raid, uri. En caso de tener varios significados incluye el contexto al que pertenece cada uno.

2.5.3. Enlaces

Los enlaces o hipervínculos son los elementos más característicos del HTML, ya que fueron el motivo que inspiró su desarrollo, para vincular información procedente de muchas fuentes distintas. Un hipervínculo es en realidad una zona del documento, que al realizar alguna acción sobre ella (como hacer clic o colocar el ratón encima) nos redirecciona a otro punto del mismo documento o de otro distinto.

El hipervínculo en principio es invisible, lo que vemos son los elementos que insertamos en su contenido, como texto, imagen, objetos, etc.

<a>

Definición: este elemento sirve para establecer el origen o el destino del hipervínculo.

Aparición: etiquetas de apertura y cierre obligatorias.

Atributos: %**attrs**, **charset**, **type**, **name**, **href**, **hreflang**, **rel**, **rev**, **accesskey**, **shape**, **coords**, **tabindex**, **onfocus**, **onblur**.

- **name**: sirve para nombrar un ancla o marcador que será el destino del enlace.
 - **href**: sirve para especificar la URI destino del hipervínculo.
 - **accesskey**: asigna una tecla única de acceso al hipervínculo.
 - **shape**: establece la forma de la imagen que contiene el hipervínculo.
 - **coords**: establece la posición en pantalla de la imagen.
 - **target**: establece el marco de destino (por defecto se toma la ventana actual).
- Valores posibles: `_blank`, `_self`, `_parent`, `_top`.
- **tabindex**: establece un número de orden.

Se trata de un elemento en línea por lo que puede estar dentro de un párrafo, lista, etc., a su vez <a> puede contener otros elementos en línea excepto a sí mismo. Los atributos **name** y **href** son opuestos en el sentido de que, o bien creamos el origen con , o bien el destino con .

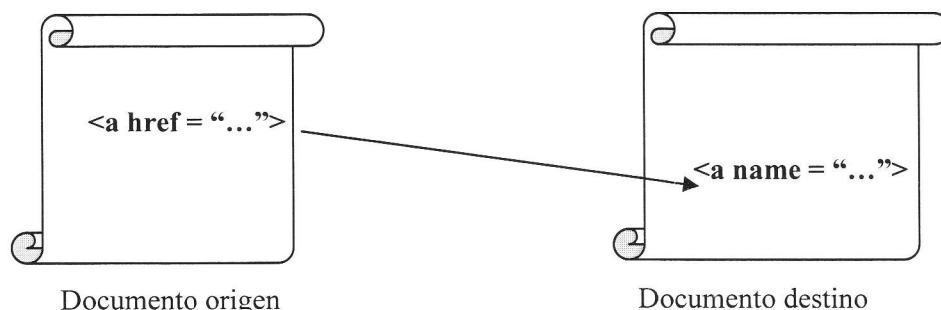


Figura 2.10: Hipervínculo

Vamos a ver cómo se utilizan los hipervínculos, mediante 3 situaciones posibles: destino en la misma página, destino en otra página del mismo servidor y destino en distinto servidor.

- Destino en la misma página.

Insertamos el marcador en el destino: ``

Insertar el hipervínculo en el origen: ``

Es preciso usar el carácter almohadilla “#” para indicar, que el destino es un marcador de posición interno al documento.

Ejemplo:

```
<body>
  <h1>Lenguajes de marcas </h1>
  <ul>
    <li> <a href="#H"> HTML </a> </li>
    <li> <a href="#X"> XHTML </a> </li>
  </ul>

  <a name="H"></a>
  <h2>HTML</h2>
  <p>HTML (HyperText Markup Language)
    Aparece a principios de los años 90.
    El objetivo principal era enlazar los documentos mediante
    enlaces de forma que al hacer clic sobre algún texto, se
    abriera otro documento relacionado con la información
    seleccionada.
  </p>
  <a name="X"></a>
  <h2>XHTML</h2>
  <p>XHTML (Extensible HyperText Markup Language)
    Es un dialecto de XML que contiene todos los elementos de
    HTML pero que se ajusta a las normas sintácticas de XML.
  </p>
  ...
</body>
```

También es posible utilizar como marcador de posición, un elemento ya existente en la página, en lugar de crear un nuevo marcador. En este ejemplo usamos el atributo `id`.

Por ejemplo: `<h2 id="X"> XHTML </h2>`

El resultado sería el mismo que con el marcador `<a name>`.

Actividad 2.17:

Inserta un enlace a la lista de módulos y otro a la lista del diccionario, creadas anteriormente.

- Destino en otra página del mismo servidor.

Supongamos ahora que tenemos una página `indice.html` que es el origen de todos los hipervínculos. También tenemos una segunda página llamada `definiciones.html` que contiene los marcadores de posición, que son el destino de los hipervínculos.

El marcador estará en la página destino: ``

El hipervínculo estará en el origen: ``

Simplemente hay que añadir el nombre de la página destino en el atributo `href`.

```
<ul>
  <li> <a href="Definiciones.html#H"> HTML </a> </li>
  <li> <a href="Definiciones.html#X"> XHTML </a> </li>
  <li> <a href="Definiciones.html#W"> WML </a> </li>
</ul>
```

Indice.html

```
<a name="X"></a>
<h2>XHTML</h2>
<p>XHTML (Extensible HyperText Markup Language)
  Es un lenguaje que contiene todos los elementos de HTML
  pero que se ajusta a las normas sintácticas de XML.
</p>
```

Definiciones.html

Actividad 2.18:

Crea una segunda página `.html` para tu empresa, e inserta un enlace en `index.html` que lleve al comienzo de la segunda página y otro al final.

- Destino en otro servidor (hipervínculo externo)

En este caso no necesitamos marcadores sino especificar la URI completa de la página destino.

Ejemplo:

```
<ul>
  <li> <a href="http://www.w3.org/TR/html401/"> HTML</a> </li>
  <li> <a href="http://www.w3.org/TR/xhtml1/"> XHTML</a> </li>
  <li> <a href="http://www.wapforum.org"> WML</a> </li>
</ul>
```