

## **Unidad III: Lenguaje de presentación**

### **3.1 Introducción**

Un lenguaje de presentación define un conjunto de etiquetas y atributos válidos y que ofrecen un significado visual para cada elemento del lenguaje, sino que ofrece un número de reglas sintácticas para poder crear documentos. Uno de los lenguajes más comunes para presentar información web es HTML.

El lenguaje de presentación es aquel que indica el formato del texto. Este tipo de marcado es útil para maquetar la presentación de un documento para su lectura, pero resulta insuficiente para el procesamiento automático de la información.

El marcado de presentación resulta más fácil de elaborar, sobre todo para cantidades pequeñas de información. Sin embargo resulta complicado de mantener o modificar, por lo que su uso se ha ido reduciendo en proyectos grandes en favor de otros tipos de marcado más estructurados.

Se puede tratar de averiguar la estructura de un documento de esta clase buscando pistas en el texto. Por ejemplo, el título puede ir precedido de varios saltos de línea, y estar ubicado centrado en la página. Varios programas pueden deducir la estructura del texto basándose en esta clase de datos, aunque el resultado suele ser bastante imperfecto.

### **3.2 Sintaxis**

El HTML es un lenguaje de marcas que basa su sintaxis en un elemento de base al que llamamos etiqueta. A través de las etiquetas vamos definiendo los elementos del documento, como enlaces, párrafos, imágenes, etc. Así pues, un documento HTML estará constituido por texto y un conjunto de etiquetas para definir la forma con la que se tendrá que presentar el texto y otros elementos en la página.

La etiqueta presenta frecuentemente dos partes: Una apertura de forma general <etiqueta> Un cierre de tipo </etiqueta>

### **3.3 Selectores**

Los selectores identifican a un elemento dentro de la página Web para luego poder definir sus propiedades. Los distintos tipos de selectores abarcan desde el simple nombre de las etiquetas usadas en HTML (BODY, P, CODE, TABLE, UL, etc.) hasta complejas combinaciones que permiten un juego muy amplio de selecciones dentro de la página.

El conocimiento en profundidad de los distintos selectores es uno de los aspectos más complejos del lenguaje de las Hojas de Estilo y también el que nos permite sacar el máximo provecho de las CSS.

Existen muchos editores de páginas Web que facilitan la labor de crear y aplicar Hojas de Estilo, pero los selectores que se pueden definir con estos programas son sólo los más elementales. La complejidad que pueden alcanzar los selectores contextuales escapan a las posibilidades de cualquier herramienta y es ahí donde los diseñadores deberán apelar a sus propios recursos.

En esta sección se explican los llamados selectores simples.

### **3.4 Tipos de medios**

HTML permite a los autores diseñar documentos que saquen partido de las características de los medios en los que el documento vaya a ser representado (p.ej., pantallas gráficas, pantallas de televisión, dispositivos de mano, navegadores basados en voz, dispositivos táctiles Braille, etc.). Al especificar el atributo media, los autores permiten a los agentes de usuario cargar y aplicar las hojas de estilo de manera selectiva.

Las siguientes declaraciones de ejemplo se aplican a elementos H1. Cuando se proyecten en una reunión de negocios, todos ellos aparecerán en azul. Cuando se impriman, aparecerán centrados.

```
<HEAD>
<STYLE type="text/css" media="projection">
  H1 { color: blue }
</STYLE>
```

```
<STYLE type="text/css" media="print">
  H1 { text-align: center }
</STYLE>
```

Este ejemplo añade efectos de sonido a los vínculos para su uso en salidas por voz:

```
<STYLE type="text/css" media="aural">
  A { cue-before: uri(bell.aiff); cue-after: uri(dong.wav)}
</STYLE>
</HEAD>
```

El control de medios es particularmente interesante cuando se aplica a hojas de estilo externas, ya que los agentes de usuario pueden ahorrar tiempo si obtienen de la red únicamente aquellas hojas de estilo que se apliquen el dispositivo actual. Por ejemplo, los navegadores basados en voz pueden evitar la descarga de hojas de estilo diseñadas para la representación visual.

Hay dos maneras de especificar las hojas de estilos dependientes de los medios:

- Desde una hoja de estilo con las reglas: @import o @media

@import url("archivo.css" ) tipo de medio;

@media tipo de medio{/\* la hoja de estilo para el tipo de medio va aquí \*/}

- • Especificando el medio de destino dentro del lenguaje del documento

```
<html>
<head>
<title>Vínculo a un tipo de medio</title>
<link rel="stylesheet" type="text/css" media= "tipo_medio1, tipo_medio2"
href="archivo.css">
</head>

<body>
<p>Texto, texto, texto</p>
</body>
</html>
```

Los posibles valores para tipos de medios:

all | brille | embossed | handheld | print | projection | screen | speech | tty | tv

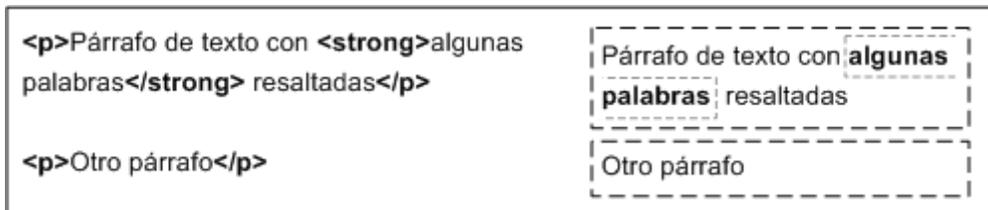
Una regla @media especifica los tipos de medios. Veamos unos ejemplos del uso de la regla @media:

- • @media print{body { font-size: 10pt }}
- • @media screen{body { font-size: 12px }}

### 3.5 Modelo de caja

El modelo de cajas o "*box model*" es seguramente la característica más importante del lenguaje de hojas de estilos CSS, ya que condiciona el diseño de todas las páginas web. El modelo de cajas es el comportamiento de CSS que hace que todos los elementos de las páginas se representen mediante cajas rectangulares.

Las cajas de una página se crean automáticamente. Cada vez que se inserta una etiqueta HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento. La siguiente imagen muestra las tres cajas rectangulares que crean las tres etiquetas HTML que incluye la página:



**Figura 4.1** Las cajas se crean automáticamente al definir cada elemento HTML

Las cajas de las páginas no son visibles a simple vista porque inicialmente no muestran ningún color de fondo ni ningún borde. La siguiente imagen muestra las cajas que forman la página web de <http://www.alistapart.com/> después de forzar a que todas las cajas muestren su borde:

### 3.6 Aplicación en documento web

La implementación de un lenguaje es la que provee una manera de que se ejecute un programa para una determinada combinación de software y hardware. Existen básicamente dos maneras de implementar un lenguaje: Compilación e interpretación. Compilación es la traducción a un código que pueda utilizar la máquina. Los programas traductores que pueden realizar esta operación se llaman compiladores. Éstos, como los programas ensambladores avanzados, pueden generar muchas líneas de código de máquina por cada proposición del programa fuente.

Se puede también utilizar una alternativa para traducir lenguajes de alto nivel. En lugar de traducir el programa fuente y grabar en forma permanente el código objeto que se produce durante la compilación para utilizarlo en una ejecución futura, el programador sólo carga el programa fuente en la computadora junto con los datos que se van a procesar. A continuación, un programa intérprete, almacenado en el sistema operativo del disco, o incluido de manera permanente dentro de la máquina, convierte cada proposición del programa fuente en lenguaje

de máquina conforme vaya siendo necesario durante el procesamiento de los datos. El código objeto no se graba para utilizarlo posteriormente.

La siguiente vez que se utilice una instrucción, se la deberá interpretar otra vez y traducir a lenguaje máquina. Por ejemplo, durante el procesamiento repetitivo de los pasos de un ciclo o bucle, cada instrucción del bucle tendrá que volver a ser interpretada en cada ejecución repetida del ciclo, lo cual hace que el programa sea más lento en tiempo de ejecución (porque se va revisando el código en tiempo de ejecución) pero más rápido en tiempo de diseño (porque no se tiene que estar compilando a cada momento el código completo). El intérprete elimina la necesidad de realizar una compilación después de cada modificación del programa cuando se quiere agregar funciones o corregir errores; pero es obvio que un programa objeto compilado con antelación deberá ejecutarse con mucha mayor rapidez que uno que se debe interpretar a cada paso durante una ejecución del código.

La mayoría de lenguajes de alto nivel, permiten la programación multipropósito, sin embargo, muchos de ellos fueron diseñados para permitir programación dedicada, como lo fue PASCAL con las matemáticas en su comienzo. También, se han implementado lenguajes educativos infantiles como LOGO que mediante una serie de simples instrucciones, permitía mover una tortuga entre otras cosas. En el ámbito de infraestructura de internet, cabe destacar a Perl con un poderoso sistema de procesamiento de texto y una enorme colección de módulos.