

CSS. Hojas de estilo

CONTENIDO

Introducción
Sintaxis
Selectores
Modelo de cajas
Texto
Listas
Tablas
Formularios
Layout
Prioridad
Miscelánea
Ejercicios propuestos

3.1. Introducción

En primer lugar debe quedar claro que CSS (Cascading Style Sheets - Hojas de Estilo en Cascada) no es un lenguaje de marcas sino un lenguaje de presentación, por tanto, su estructura y sintaxis distan mucho de HTML.

Lo estudiamos porque CSS está íntimamente ligado a los lenguajes de marcas, hasta el punto de que resulta imprescindible para crear sitios web con calidad.

El objetivo principal de CSS es manejar el aspecto y formato de los documentos, liberando de esta forma a HTML de las tareas de presentación.

Esta separación entre contenidos y presentación supone muchas ventajas, la más evidente es que evita la repetición innecesaria de código, como podemos ver en el siguiente ejemplo:

```
<body>
<h1><font face="Arial" size="5">UNIDAD 2. HTML y XHTML</font></h1>
<h2><font face="Verdana" size="4">2.5. Contenido del cuerpo</font>
</h2>
<h3><font face="Garamond" size="3">2.5.1. Manejo del texto</font>
</h3>
<h3><font face="Garamond" size="3">2.5.2. Listas</font></h3>
<h2><font face="Verdana" size="4">2.7. HTML 5</font></h2>
<h3><font face="Garamond" size="3">2.7.1. Elementos de sección
</font></h3>
</body>
```

Que se puede mejorar con este otro:

```
<html>
<head>
<style type="text/css">
h1 {font-family: Arial;}
h2 {font-family: Verdana;}
h3 {font-family: Garamond;}
</style>
</head>
<body>
<h1>UNIDAD 2. HTML y XHTML</h1>
<h2>2.5. Contenido del cuerpo</h2>
<h3>2.5.1. Manejo del texto</h3>
<h3>2.5.2. Enlaces</h3>
<h2>2.7. HTML 5</h2>
<h3>2.7.1. Elementos de sección</h3>
</body>
</html>
```

Con este último código, cada vez que incluyamos un elemento de cabecera **<h1>** al **<h3>**, el tipo de fuente ya estará establecido de antemano por las reglas css, contenidas en **<style>**.

Así pues, podemos citar como ventajas en el uso de CSS las siguientes:

- Disminuye el código a escribir.
- Facilita la generación de código y su mantenimiento.
- Mejora la legibilidad de los documentos.

En cuanto a su evolución histórica, el inicio de las hojas de estilo tiene lugar en torno a 1970, pero el uso generalizado se produjo más tarde con el auge de Internet.

La ausencia de un estándar oficial provocó diferentes formas de establecer reglas de estilo y como consecuencia, las habituales diferencias de visualización entre los navegadores. Como en el caso de HTML, fue el consorcio W3C el encargado de encontrar una especificación oficial que unificara la forma de definir los estilos.

Fue en 1996 cuando surgió la primera versión de CSS llamada CSS nivel 1 partiendo de los trabajos de Bert Bos.

Pero esta versión contenía bastantes problemas que fueron solucionándose con el tiempo, lo que dio lugar a la aparición de la segunda versión en el año 1997.

Entonces, el consorcio W3C decidió crear grupos específicos de trabajo para HTML, CSS y DOM. El estándar oficial CSS nivel 2 se publica en 1998.

Este estándar ha sufrido una revisión llamada **CSS 2.1** que es la especificación actual (consultar en <http://www.w3.org/TR/CSS2/>).

Desde 1998 hasta hoy el W3C sigue trabajando para establecer la versión CSS 3.

En cuanto al comportamiento de los navegadores respecto las reglas de CSS, podemos decir que todos los navegadores tienen un soporte muy amplio para CSS 2.1, pero no tan completo en su mayoría para la versión CSS 3, aunque progresivamente van incorporando las nuevas propiedades y reglas que contiene.

Para comprobar la compatibilidad de nuestro navegador con las versiones CSS consultar en <http://www.css3.info/selectors-test/>

En la siguiente tabla, se puede ver que las primeras versiones de Internet Explorer tienen problemas de compatibilidad con la versión CSS 2.1 y casi nula con CSS 3. Los demás navegadores se han adaptado con mayor rapidez a la evolución de CSS.

Navegador: CSS:	IE6	IE7	IE8	IE9	FF3	FF3.6	FF4	Chrome
CSS 1	Parcial	Total	Total	Total	Total	Total	Total	Total
CSS 2.1	Muy Pobre	Parcial	Total	Total	Buena	Total	Total	Total
CSS 3	Nula	Pobre	Parcial	Total	Parcial	Buena	Total	Total

Tabla 3.1: Compatibilidad de navegadores con CSS

3.2. Sintaxis

3.2.1. Cómo incluir CSS en el documento

Vamos a ver 3 formas distintas de integrar hojas de estilo en documentos HTML.

La primera ya se utilizó en el capítulo anterior para el color de fondo y consiste en utilizar el atributo `style` en la propia línea. La segunda consiste en incluir el código css en la cabecera del documento, dentro de la etiqueta `<style>`. La tercera consiste en crear un archivo css externo y vincularlo al documento mediante un enlace `<link>` o `@import`.

- CSS en línea.

```
<body>
  <p style="font-family: Verdana; font-size: medium;">HOLA MUNDO</p>
</body>
```

Se utiliza para indicar el estilo particular de la línea (no utiliza llaves).

- CSS interno.

```
<html>
  <head>
    <style type="text/css">
      body {font-family: Courier New;}
      h1 {font-family: Arial; font-size: x-large;}
      p {font-family: Verdana; font-size: medium;}
    </style>
  </head>
  <body>
    <h1>Tipo de fuente Arial y tamaño grande</h1>
    <p>Tipo de fuente Verdana y tamaño medio</p>
  </body>
</html>
```

Se utiliza para indicar los estilos propios de esta página, que se complementan con los estilos globales del sitio web, y suelen definirse en un archivo css externo.

Se coloca dentro de la etiqueta `<style>`.

<p>Tipo de fuente Arial y tamaño grande</p> <p>Tipo de fuente Verdana y tamaño medio</p>

Figura 3.1: CSS interno

- CSS externo.

Creamos el archivo estilos.css en una carpeta como /css:

```
/* Estilos globales */
body {margin: 0px;}
td    {color: #000000;
       font-size: 12px;}
a      {color: #FF6600;
       font-weight: bold;}
a:hover {color: #3366CC;}
```

Después insertamos en cada documento el enlace mediante <link>:

```
<head>
  <link rel="stylesheet" type="text/css" href="/css/estilos.css">
</head>
```

Hay otra forma de enlazar el archivo .css, mediante la regla @import:

```
<style type="text/css">
  @import "/css/estilos.css";
</style>
```

Se utiliza para indicar los estilos globales que van a compartir todas las páginas web del sitio.

La prioridad de las reglas en caso de colisión, aumenta cuanto más particular sea la regla, es decir:

```
+  ↑ 1º css en línea
   | 2º css interno
-  | 3º css externo
```

Dentro de cada una de estas formas, en caso de repetición es la última regla la que se aplica, por ejemplo en este caso, se aplicaría tipo de fuente Verdana:

```
p {font-family: Arial;
   font-family: Verdana;}
```

Podemos aumentar la prioridad utilizando la palabra reservada **!important**, en el ejemplo anterior permitiría aplicar la fuente Arial, a pesar de no ser la última regla.

3.2.2. Cómo construir las reglas CSS

Cada regla CSS sirve para definir un estilo y consiste en la unión de un selector más una declaración entre llaves. A su vez, cada declaración se compone de una o más parejas de propiedades con su valor, terminadas en punto y coma.

Regla CSS

selector { propiedad: valor; propiedad: valor; ...; }



p {font-family: Verdana; font-size: medium;}

El selector especifica los elementos de HTML sobre los que se define un estilo.

La propiedad es una característica del selector, en el ejemplo anterior serían la fuente y el tamaño del texto. El listado de propiedades y valores posibles, se encuentra en la especificación oficial del W3C (consultar <http://www.w3.org/TR/CSS2/propidx.html>).

Las reglas se pueden agrupar en caso de compartir la declaración o el selector:

```
h1 {font-family: Verdana; color: red;}
h2 {font-family: Verdana; color: red;}
```

Se pueden resumir en: **h1, h2 {font-family: Verdana; color: red;}**

Igualmente:

```
h1 {font-family: Verdana;}
h1 {color: red;}
```

Se pueden resumir en: **h1 {font-family: Verdana; color: red;}**

Para los comentarios utilizaremos los símbolos /*,..., */ procedentes del lenguaje C:

```
/* Comentarios */
```

3.3. Selectores

En CSS 2.1 existen los siguientes tipos de selectores:

Selector universal, Selector de tipo, Selector descendiente, Selector hijo, Selector adyacente, Selector de atributos, Selector de clase, Selector de id, Pseudo-clase y Pseudo-elemento.

3.3.1. Selector universal

Sirve para seleccionar todos los elementos de la página.

Se indica mediante el símbolo asterisco, como por ejemplo:

```
* { margin: 0px }
```

3.3.2. Selector de tipo

Sirve para seleccionar todos los elementos de la página cuya etiqueta coincida con el selector, por ejemplo:

```
p {font-family: Verdana; color: red;}
```

3.3.3. Selector descendiente

En primer lugar decimos que D es un elemento descendiente del elemento E, si D está contenido entre la apertura y cierre de E.

El selector descendiente se puede construir con 2 o más selectores simples separados por un espacio en blanco. Un selector del tipo E M {declaración}, sirve para seleccionar todos los elementos M que son descendientes de E a cualquier nivel, por ejemplo:

```
p a {font-size: 50px}
```

Aplicada al código:

```
<p>Texto1</p>
```

```
<p><a>Texto2</a></p>
```

```
<p><span class="nieto"><a>Texto3</a></span></p>
```

Visualizaría el Texto2 y el Texto3 a tamaño 50.

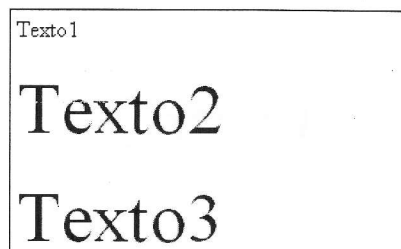


Figura 3.2: Selector descendiente

3.3.5. Selector adyacente

Se emplea para seleccionar elementos que son “hermanos” (tienen el mismo padre) y son adyacentes (consecutivos) en el código.

Se construye con el signo +, de forma que una regla como `E + M {declaración;}`, selecciona a todos los elementos de tipo M, tales que E y M tienen el mismo padre y E precede inmediatamente a M en el código (salvo comentarios).

```
h1 + h2 {font-size: 50px}
```

3.3.6. Selector de atributos

El selector de atributos puede seleccionar de 4 formas distintas:

- [atributo]: selecciona los elementos que tengan ese atributo independientemente del valor que tome (puede tener otros atributos).
- [atributo = valor]: selecciona los elementos que tengan el atributo con el valor especificado (puede tener otros atributos).
- [atributo ~ = valor]: selecciona los elementos que tengan el atributo con al menos uno de los valores como el especificado (puede tener otros atributos).
- [atributo |= valor]: selecciona los elementos que tengan el atributo y que la palabra comience con el valor especificado (solo sirve para el atributo lang).

Ejemplos:

```
span[class] {font-family: Arial;}

span[class = "azulon"] {color: blue;}

span[class ~ = "grandote"] {font-size: 50px}

span[lang |= "es"] {color: red;}
```

3.3.7. Selector de clase

Sirve para seleccionar una instancia en particular de un elemento.

Por ejemplo, en el siguiente código tenemos dos instancias del elemento `<p>`, para poder distinguir cada una utilizamos el atributo class.

```
<h3 class = "grandote">Texto1</h3>
<p class = "grandote">Texto2</p>
<p class = "peque">Texto3</p>
```

Se construye de la forma `selector_simple.clase`, como por ejemplo:

```
p.grandote {font-size: 50px}      Visualizaría solamente el Texto2 en grande.
```

También se puede omitir el selector `_simple`, por ejemplo:

```
.grandote {font-size: 50px}
```

En este caso selecciona todos los elementos con la clase “grandote”.

Visualizaría el Texto1 y el Texto2 en grande, ya que el elemento `<h3>` también tiene la clase “grandote”.

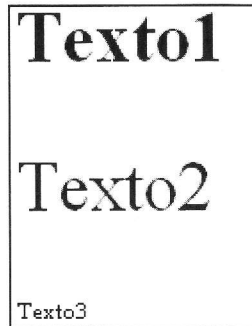


Figura 3.5: Selector de clase

3.3.8. Selector ID

Sirve para seleccionar a una sola línea de toda la página.

Se puede utilizar el selector de clase, pero resulta menos eficiente porque está concebido para seleccionar un subconjunto de instancias, para una sola línea es preferible el selector id.

En el código anterior sería más adecuado escribir:

```
<p id="peque">Texto4</p>
```

La regla con el selector id, se construye usando el carácter almohadilla #:

```
#peque {font-size: 8px}
```

Visualizaría solamente el Texto3 en pequeño.

Actividad 3.1:

Prueba todos los selectores vistos hasta ahora, en un nuevo documento html.

Usa CSS interno, es decir, pon las reglas dentro de la etiqueta `<style>`.
