

UNIDAD DE TRABAJO 5

HOJAS DE ESTILO CSS

1	Introducción	2
1.1	Hoja de estilo	2
1.2	Utilidad de las hojas de estilo	2
1.3	Tipos de hojas de estilo	3
2	Creación de reglas de estilo	3
3	Propiedades de estilo	3
3.1	Fuente	3
3.1.1	Color	3
3.1.2	Tamaño de fuente	4
3.1.3	Tipo de fuente	4
3.1.4	Peso de fuente	4
3.1.5	Estilo de fuente	4
3.1.6	Decoración del texto	4
3.1.7	Transformaciones del texto	4
3.2	Párrafo	5
3.2.1	Interlínea	5
3.2.2	Alineamiento del texto	5
3.2.3	Indentación de texto	5
3.3	Bordes	5
3.3.1	Márgenes	5
3.3.2	Separadores	5
3.3.3	Anchura del los bordes	6
3.3.4	Estilo del borde	6
3.3.5	Color del borde	6
3.3.6	Anchura	6
3.3.7	Alineamiento	6
3.3.8	Clear	6
3.4	Sombreado	7
3.4.1	Imagen de fondo	7
3.4.2	Color de fondo	7
3.5	Numeración y viñetas	7
3.5.1	Estilo de lista	7
3.5.2	Imagen de lista	7
3.6	Otros	7
3.6.1	Display	7
3.6.2	Espacios en blanco	8
3.6.3	Cursor	8
4	Unidades	8
4.1	Unidades de longitud	8
4.2	Unidades de color	8
5	Introducción a CSS 2	9
5.1	Nuevos selectores	9
6	Manual de referencia	10
6.1	Comentarios en las hojas de estilo	10
6.2	Nuevas etiquetas HTML	10
6.2.1	La etiqueta <STYLE>	10
6.2.2	La etiqueta <LINK>	10
6.2.3	La etiqueta 	11
6.3	Nuevos atributos para las etiquetas HTML	11
6.3.1	El atributo STYLE	11
6.3.2	El atributo CLASS	11
6.3.3	El atributo ID	12
6.4	La etiqueta <DIV>	12
7	Localización de las definiciones de los estilos	12

7.1	Definición de hojas de estilo en la cabecera	13
7.2	Hojas de estilo en ficheros externos	13
7.3	Especificación de estilos en elementos individuales.....	14
8	Estilos de clase.....	14
9	Estilos individuales	16
10	Criterios de selección contextual.....	17
11	Combinación de hojas de estilo	18
12	Propiedades de elementos de bloque.....	19
12.1	Introducción	19
12.2	Márgenes.....	21
12.3	Bordes	21
12.4	Separadores	21
12.5	Herencia de propiedades	22

1 INTRODUCCIÓN

1.1 Hoja de estilo

Una hoja de estilo es un conjunto de instrucciones que definen la apariencia de diversos elementos de una página Web. En otras palabras una hoja de estilo nos permite indicar por ejemplo el tamaño de la fuente, color y estilo de cierto párrafo que nosotros indiquemos, mediante la definición de estilos y aplicación de los mismos.

1.2 Utilidad de las hojas de estilo

Las hojas de estilo se usan porque tienen muchas ventajas sobre las etiquetas tradicionales, ya que por ejemplo es posible crear una sola hoja de estilo que compartan muchos documentos, y al hacer un cambio en la hoja de estilo todos los documentos que la usan tendrán la apariencia deseada.

Podemos decir que las hojas de estilo eliminan las etiquetas de apariencia del lenguaje HTML. La última especificación del HTML desaprueba el uso de las principales etiquetas para formatear texto y sus atributos, como por ejemplo la etiqueta ``, `<CENTER>`, entre otros. El hecho de que esta especificación desaprobe esas etiquetas indica que posiblemente en la próxima especificación de HTML estas etiquetas quedarán obsoletas.

Las hojas de estilo son la innovación más importante al HTML (también se usan en otros lenguajes como XML y SGML), ya que le dan capacidades que nunca tuvo. Entre otras mencionaremos:

- Definir los márgenes de un documento o párrafo, o definir el espacio entre caracteres.
- El uso de diversas unidades de medición, pixeles, puntos, picas, mm, en los principales elementos del HTML, como son tablas, fuentes, bordes y en general los elementos que tenían atributos como `size`, `height` o `width`.
- El posicionamiento de bloques de texto en cualquier parte del documento HTML, ya que es posible definir en diversas unidades la posición de un bloque de texto.
- Mejor control sobre las fuentes de texto.
- El poder cambiar las características de una hoja de estilo mediante el uso de lenguajes de programación web como JavaScript.
- Las hojas de estilo son la base de la implementación estándar del HTML dinámico o DHTML.

1.3 Tipos de hojas de estilo

Existen actualmente dos tipos de hojas de estilo, las hojas de estilo tipo **CSS (cascading style sheet)** y las hojas de estilo en JavaScript. En este documento solo vamos a tratar las hojas de estilo CSS.

2 CREACIÓN DE REGLAS DE ESTILO

La creación de una regla de estilo sigue el siguiente formato:

```
estilo {  
    propiedad: valor;  
    ...  
    propiedad: valor;  
}
```

donde:

- **estilo**: es el nombre que daremos al estilo. Corresponderá con el nombre de una etiqueta.
- **propiedad**: el nombre de la propiedad que vamos a definir o modificar. Esta definido en el lenguaje CSS. Debe escribirse en minúsculas.
- **::** separa la propiedad del valor.
- **valor**: el valor que asignaremos a la propiedad.
- **;** indica el fin de la definición de la propiedad.
- Cuando se usen varias propiedades, estas deben separarse con espacios o ponerse en otro renglón.

Ejemplo de sintaxis CSS

```
h4 {  
    text-transform: uppercase;  
    color: blue;  
    font-style: italic;  
}
```

Uso de la hoja de estilo:

```
<h4>Una cabecera en mayúsculas, azul y cursiva</h4>
```

Resultado del ejemplo

UNA CABECERA EN MAYÚSCULAS, AZUL Y CURSIVA

3 PROPIEDADES DE ESTILO

3.1 Fuente

3.1.1 Color

Sintaxis CSS: **color**

- Posibles valores: color
- Valor inicial: black
- Se aplica a: todos los elementos

Esta propiedad especifica el color del texto.

3.1.2 Tamaño de fuente

Sintaxis CSS: `font-size`

- Posibles valores: tamaño absoluto, tamaño relativo, longitud, porcentaje.
 - Tamaño absoluto: xx-small, s-small, small, medium, large, x-large, xx-large.
 - Tamaño relativo: larger, smaller.
 - Longitud: número + unidad.
 - Porcentaje: tamaño de fuente relativo al tamaño de fuente del padre.
- Valor inicial: médium.
- Se aplica a: todos los elementos.

3.1.3 Tipo de fuente

Sintaxis CSS: `font-family`

- Posibles valores: un nombre de fuente: arial, sans-serif, cursive, monospace, ...
- Valor inicial: la fuente por defecto
- Se aplica a: todos los elementos

3.1.4 Peso de fuente

Sintaxis CSS: `font-weight`

- Posibles valores: `normal`, `bold`, `bolder`, `lighter`, 100 - 900
- Valor inicial: normal
- Se aplica a: todos los elementos

3.1.5 Estilo de fuente

Sintaxis CSS: `font-style`

- Posibles valores: `normal`, `italic`
- Valor inicial: normal
- Se aplica a: todos los elementos

3.1.6 Decoración del texto

Sintaxis CSS: `text-decoration`

- Posibles valores: none, underline, line-through, blink
- Valor inicial: none
- Se aplica a: todos los elementos

3.1.7 Transformaciones del texto

Sintaxis CSS: `text-transform`

- Posibles valores: capitaliza, uppercase, lowercase, none
- Valor inicial: none
- Se aplica a: todos los elementos

3.2 *Párrafo*

3.2.1 Interlínea

Sintaxis CSS: `line-height`

- Posibles valores: número, longitud, porcentaje, `normal`
- Valor inicial: tamaño por defecto para la fuente
- Se aplica a: elementos de bloque
- Valores porcentuales: relativos al tamaño de la fuente

3.2.2 Alineamiento del texto

Sintaxis CSS: `text-align`

- Posibles valores: `left`, `right`, `center`, `justify`
- Valor inicial: `left`
- Se aplica a: elementos de bloque

3.2.3 Indentación de texto

Sintaxis CSS: `text-indent`

- Posibles valores: longitud, porcentaje.
 - Longitud: número + unidad
 - Porcentaje: longitud de indentación relativo a la anchura del padre.
- Valor inicial: `0`
- Se aplica a: elementos de bloque

3.3 *Bordes*

3.3.1 Márgenes

Sintaxis CSS: `margin-left`, `margin-right`, `margin-top`, `margin-bottom`, `margin`

- Posibles valores: longitud, porcentaje, `auto`
- Valor inicial: `0`
- Se aplica a: todos los elementos
- Valores porcentuales: relativos a la anchura del padre

3.3.2 Separadores

Sintaxis CSS: `padding-left`, `padding-right`, `padding-top`, `padding-bottom`, `padding`

- Posibles valores: longitud, porcentaje
- Valor inicial: `0`
- Se aplica a: todos los elementos
- Valores porcentuales: relativos a la anchura del padre

3.3.3 Anchura del los bordes

Sintaxis CSS: `border-left-width`, `border-right-width`, `border-top-width`, `border-bottom-width`, `border-width`

- Posibles valores: longitud
- Valor inicial: none
- Se aplica a: todos los elementos

3.3.4 Estilo del borde

Sintaxis CSS: `border-style`

- Posibles valores: none, solid, double, inset, outset, groove, ridge
- Valor inicial: none
- Se aplica a: todos los elementos

3.3.5 Color del borde

Sintaxis CSS: `border-color`

- Posibles valores: color, `none`
- Valor inicial: none
- Se aplica a: todos los elementos
- Color: nombre de un color o 6 dígitos hexadecimales indicando su valor rgb.

3.3.6 Anchura

Sintaxis CSS: `width`

- Posibles valores: longitud, porcentaje, `auto`
- Valor inicial: auto
- Se aplica a: elementos de bloque
- Valores porcentuales: relativos a la anchura del padre

3.3.7 Alineamiento

Sintaxis CSS: `float`

- Posibles valores: `left`, `right`, `none`
- Valor inicial: none
- Se aplica a: todos los elementos
- Valores porcentuales: relativos a la anchura del padre

3.3.8 Clear

Sintaxis CSS: `clear`

- Posibles valores: `none`, `left`, `right`, `both`
- Valor inicial: none
- Se aplica a: todos los elementos

Esta propiedad especifica cuando un elemento permite la existencia de elementos flotando a sus lados. Exactamente, el valor de esta propiedad son los lados en que no se aceptan elementos flotantes.

3.4 *Sombreado*

3.4.1 Imagen de fondo

Sintaxis CSS: `background-image`

- Posibles valores: url
- Se aplica a: todos los elementos

3.4.2 Color de fondo

Sintaxis CSS: `background-color`

- Posibles valores: color
- Valor inicial: vacío
- Se aplica a: todos los elementos

3.5 *Numeración y viñetas*

3.5.1 Estilo de lista

Sintaxis CSS: `list-style-type`

- Posibles valores: `none`, `disk`, `circle`, `square`, `decimal`, `lower-roman`, `upper-roman`, `lower-alpha`, `upper-alpha`
- Valor inicial: `disk`
- Se aplica a: elementos cuya propiedad `display` es `list-item`

3.5.2 Imagen de lista

Sintaxis CSS: `list-style-image`

- Posibles valores: url.
- Se aplica a: elementos cuya propiedad `display` es `list-item`

3.6 *Otros*

3.6.1 Display

Sintaxis CSS: `display`

- Posibles valores: `none`, `block`, `inline`, `list-item`
 - `block`: los elementos se separan unos de otros generalmente por un salto de línea.
 - `inline`: el elemento forma parte de la línea.
 - `list-item`: el elemento es parte de una lista.
- Valor inicial: según HTML
- Se aplica a: todos los elementos

Esta propiedad indica cuando un elemento es en línea, como ``, de bloque, como `<h1>` o de lista, como ``.

3.6.2 Espacios en blanco

Sintaxis CSS: `white-space`

- Posibles valores: `normal`, `pre`
- Valor inicial: según HTML
- Se aplica a: elementos de bloque.

3.6.3 Cursor

Sintaxis CSS: `cursor`

- Posibles valores: `auto`, `crosshair`, `default`, `pointer`, `move`, `e-resize`, `ne-resize`, `nw-resize`, `n-resize`, `se-resize`, `sw-resize`, `s-resize`, `w-resize`, `text`, `wait`, `help`, `url`.
- Valor inicial: `auto`
- Se aplica a: todos los elementos

4 UNIDADES

4.1 Unidades de longitud

El formato de un valor de longitud es un signo opcional ('+' o '-', '+' por defecto), un número y una unidad de medida. Ejemplos: 12pt, 2em, 3mm.

Hay tres tipos de unidades: absolutas, relativas y pixel.

Tipos de unidades absolutas:

- **pt**: puntos (1/72 de pulgada).
- **pc**: picas (12 puntos).
- **in**: pulgadas (2,54 cm).
- **mm**: milímetros.
- **cm**: centímetros.

Tipos de unidades relativas:

- **em**: el tamaño de la fuente base (la especificada por el usuario en el navegador).
- **ex**: la mitad de la altura la fuente base.
- **px**: pixels, depende de la resolución establecida en el ordenador.

Los elementos hijo heredan los valores calculados y no los valores relativos. Por ejemplo:

```
body {font-size: 12; text-indent: 3em}
h1 {font-size: 15pt;}
```

En este ejemplo la indentación del texto en <h1> será 36pt y no 45pt.

4.2 Unidades de color

Un valor de color es o un nombre de color o una descripción RGB.

La lista de colores sugerida es: `aqua`, `black`, `blue`, `fuchsia`, `gray`, `green`, `maroon`, `navy`, `olive`, `purple`, `red`, `silver`, `teal`, `white`, `yellow`.

```
body {color: black;}
```

Un color RGB puede especificarse mediante 6 dígitos hexadecimales. Los dos primeros indican rojo, los dos segundos verde y los dos últimos azul. Ejemplos:


```
body {color: #ff0000;} /* rojo */
```

También podemos utilizar la función `rgb()`. Usa tres argumentos: rojo, verde y azul. Cada color puede ser un entero entre 0 y 255, o un porcentaje. Ejemplo:

```
p {color: rgb(200, 20, 240);} /* púrpura brillante */
blockquote {background-color: rgb(100%, 100%, 20%);} /* amarillo brillante */
```

5 INTRODUCCIÓN A CSS 2

CSS2 incorpora herramientas para los desarrolladores WEB y para los diseñadores y no sólo añade funcionalidad sino que también redefine algunas de las ya especificadas en CSS1. Estas nuevas características permiten al usuario seleccionar y formatear elementos de forma precisa:

- Se han añadido características que permiten dar formato a objetos seleccionados, objetos apuntados, etc.
- Se han añadido los tipos de medios. Esta nueva característica permite crear formatos diferentes dependiendo del tipo de medio que se vaya a utilizar para mostrarlo, video, TV, impresoras, etc.

5.1 Nuevos selectores

Utilizando CSS2 se pueden seleccionar elementos basándose en un patrón. El navegador busca ese patrón y después aplica las reglas especificadas para él.

En la siguiente tabla se resumen todos los selectores y su forma de trabajar.

Patrón	Significado
*	Selector universal.
E > F	Selecciona cualquier elemento F hijo del elemento E.
E:first-child	Selecciona todos los elementos E que son el primer hijo del elemento padre de que se trate.
E:link	Selecciona todos los elementos E de un enlace que no ha sido todavía visitado.
E:visited	Selecciona todos los elementos E enlace que han sido visitados
E:active	Selecciona todos los elementos E que están seleccionados.
E:hover	Selecciona los elementos E apuntados por el ratón.
E:focus	Selecciona los elementos E que tienen el foco.
E[atributo]	Selecciona todos los elementos E con el <i>atributo</i> especificado.
E[atributo="mi_valor"]	Selecciona todos los elementos E con el <i>atributo</i> especificado e inicializado a <i>mi_valor</i> .
E[atributo~="mi_valor"]	Selecciona todos los elementos E con el <i>atributo</i> inicializado a una lista de valores entre el que se incluye <i>mi_valor</i>

Ejemplo:

Sintaxis CSS

```
/* Evitar que los enlaces estén subrayados y cambiar el aspecto de los
enlaces*/
<STYLE TYPE= "TEXT/CSS">
    a { text-decoration: none }
    a:link { color: blue }
    a:active { color: red; font-weight: bold}
    a:visited { color: green }
</STYLE>
```

6 MANUAL DE REFERENCIA

6.1 Comentarios en las hojas de estilo

En las hojas de estilo en cascada se pueden usar comentarios. Estos se colocan entre los caracteres `/*` y `*/`. Por ejemplo:

```
em {color:red;} /* el texto enfatizado será rojo */
```

Los comentarios no pueden anidarse.

6.2 Nuevas etiquetas HTML

En esta sección se verán las nuevas etiquetas que han sido añadidas para trabajar con estilos.

6.2.1 La etiqueta `<STYLE>`

Las etiquetas `<STYLE>` y `</STYLE>` se usan para crear una hoja de estilo. En su interior podemos especificar estilos para elementos, definir clases e identificadores y en general establecer los estilos que se utilizarán en todo el documento.

Para especificar qué tipo de sintaxis se empleará utilizaremos el atributo `TYPE`. Su valor por defecto es `"text/css"` y selecciona la sintaxis CSS. Ejemplo:

```
<style type="text/css">
    body {margin-left: 10%; margin-right: 10%;}
</style>
```

6.2.2 La etiqueta `<LINK>`

La etiqueta `<LINK>` sirve para poder utilizar en un documento una hoja de estilo que está almacenada en otro fichero.

```
<html>
  <head>
    <title>Titulo</title>
    <link rel=stylesheet type="text/css"
        href="http://estilos/mi_estilo.css">
  </head>
  <body>
    ...
```

6.2.3 La etiqueta

Las etiquetas y se utilizan para delimitar el principio y el fin de una trozo de código al que se le va aplicar un estilo. En el siguiente ejemplo se aplica un estilo a una parte del texto:

```
Este texto es normal. <span style="font-weight: bold; color: green;">Este texto es diferente gracias a <span>.</span> De nuevo el texto es normal.
```

El resultado del ejemplo es:

Este texto es normal. **Este texto es diferente gracias a .** De nuevo el texto es normal.

Veamos otro ejemplo. Este hace diferentes las letras iniciales:

```
<style type="text/css">
  all.letron {font-family: times; font-size: 200%; font-weight: bold;}
</style>
<p><span class="letron">E</span>n un lugar de la Mancha...</p>
```

Resultado:

En un lugar de la Mancha...

6.3 Nuevos atributos para las etiquetas HTML

En esta sección se enumeran los nuevos atributos que podemos utilizar con las etiquetas HTML y que son útiles para trabajar con estilos. Estos atributos pueden ser usados con cualquier etiqueta para especificar su estilo.

6.3.1 El atributo STYLE

El atributo **STYLE** determina el estilo del elemento al que se aplique. Por ejemplo:

Sintaxis CSS

```
<h4 style="font-weight: bold; color: red;">Cabecera h4 en rojo</h4>
```

Resultado del ejemplo:

Cabecera h4 en rojo

6.3.2 El atributo CLASS

El atributo **CLASS** permite aplicar una clase de estilo a un elemento. Ejemplo:

Sintaxis CSS

```
<style type="text/css">
  h3.cursiva {font-style: italic;}
</style>
```

Uso de la hoja de estilo:

```
<h3 class="cursiva">Cabecera h3 cursiva</h3>
```

Se debe tener en cuenta que para los nombres de clase se distingue entre mayúsculas y minúsculas. Cada elemento HTML sólo puede utilizar una clase de estilo.

Para especificar que una clase puede aplicarse a todos los elementos se utiliza el selector **all** cuando se definen las propiedades de la clase. En el siguiente ejemplo todos los elementos de clase limón serán amarillos.

Sintaxis CSS

```
<style type="text/css">
  all.limon {color: yellow;}
</style>
```

Uso de la hoja de estilo:

```
<h4 class="limon">Una cabecera amarilla</h4>
<p class="limon">Un párrafo amarillo</p>
```

Resultado del ejemplo

Una cabecera amarilla
Un párrafo amarillo

6.3.3 El atributo ID

Cuando se definen hojas de estilo, se pueden crear estilos individuales con nombre. Un elemento puede usar clases de estilos y además usar estos estilos individuales con nombre. Con ellos podemos crear excepciones a las clases de estilos para definir un estilo individual con nombre, en sintaxis CSS se utiliza el signo #. En ambos casos se utiliza el atributo **ID** para especificar el estilo de un elemento HTML. En los nombres de los ID's se distingue entre mayúsculas y minúsculas. En el siguiente ejemplo se define la clase calor, que hace a los párrafos ser de color rojo y estar en negrita. También se crea un nombre de estilo, frío, que es de color azul. Se muestra cómo utilizar frío para crear una excepción a calor.

Sintaxis CSS

```
<style type="text/css">
  p.calor {color: red; font-weight: bold;}
  #frio {color: blue;}
</style>
```

Uso de la hoja de estilo:

```
<p class="calor">Un párrafo caliente...</p>
<p class="calor" id="frio">Un párrafo frío...</p>
```

Resultado del ejemplo:

Un párrafo caliente...
Un párrafo frío...

6.4 La etiqueta <DIV>

La etiqueta <DIV> no es nueva en HTML. Esta etiqueta también funciona con los atributos **id**, **class** y **style**. La etiqueta <DIV> se puede usar como alternativa a la etiqueta . La única diferencia entre estas etiquetas es que <DIV> provoca un salto de línea al terminar de definir un párrafo, y deja el texto que va después de su uso inmediatamente después que este termina.

7 LOCALIZACIÓN DE LAS DEFINICIONES DE LOS ESTILOS

Disponemos de tres posibilidades de distintas características para situar las definiciones de los estilos en una página Web:

- Definir los estilos en la cabecera.
- Definir los estilos en ficheros externos.

- Especificar estilos en elementos individuales.

7.1 Definición de hojas de estilo en la cabecera

Para definir una hoja de estilo directamente dentro de un documento se utiliza la etiqueta `<style>` dentro de la sección `<head>`. La etiqueta `<style>` abre la hoja de estilo, y la etiqueta `</style>` la cierra. Asegúrate de utilizar `<style>` antes de `<body>`.

Cuando use `<style>` podrá especificar el atributo `type` para indicar que tipo de sintaxis se va a emplear. Los dos posibles son `"text/css"` y `"text/javascript"`. EL valor por defecto es `"text/css"`. El siguiente ejemplo define una hoja de estilo que especifica que todos los títulos de nivel 4 serán en mayúsculas y azules, y todos los bloques en cursiva y rojos:

Sintaxis CSS

```
<head>
  <style type="text/css">
    h4 {text-transform: uppercase; color: blue;}
    blockquote {font-style: italic; color: red;}
  </style>
</head>
<body>
  ...
```

Uso de la hoja de estilo:

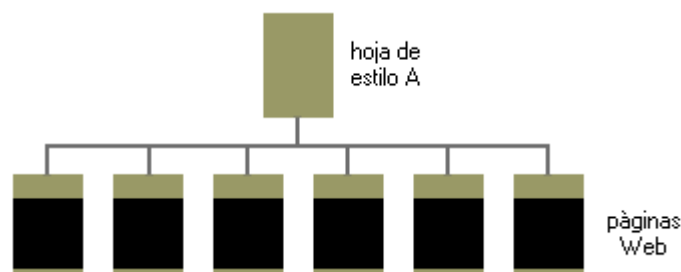
```
<h4>Este titulo es azul y esta en mayúsculas.</h4>
<blockquote>Este bloque en cursiva es rojo.</blockquote>
```

Resultado del ejemplo:

```
ESTE TITULO ES AZUL Y ESTA EN MAYÚSCULAS.
Este bloque en cursiva es rojo.
```

7.2 Hojas de estilo en ficheros externos

Se puede definir una hoja de estilo en un fichero distinto del que contiene la página y después enlazarlos. Las ventajas de este método son que podremos utilizar la hoja de estilo desde cualquier documento HTML. Se podría pensar en una hoja de estilo así definida como en un patrón que pudiera aplicarse a cualquier documento. De esta forma, se puede aplicar un estilo a todas las páginas servidas desde un sitio Web sin más que incluir un enlace al fichero con el estilo en cada página.



La sintaxis para definir estilos en ficheros es idéntica a la que se usa para definirlos en el propio documento, excepto que no es necesario incluir la etiqueta `<style>`. Ejemplo:

Sintaxis CSS

```
/* hoja de estilo externa estilo.css */
```

```
all.BOLDBLUE {color: blue; font-weight: bold;}
h1 {line-height: 18pt;}
p {color: yellow;}
/* fin de fichero */
```

Para utilizar esta hoja de estilo, se usa la etiqueta `<link>` como se muestra en el siguiente ejemplo:

Sintaxis CSS

```
<head>
  <title>El titulo</title>
  <link rel = stylesheet type = "text/css" href = "estilo.css">
</head>
```

7.3 Especificación de estilos en elementos individuales

De igual forma que se pueden definir hojas de estilo, podemos utilizar el atributo `style` de cualquier etiqueta HTML para definir un estilo que le será aplicado solamente a ella. Esta aproximación puede ser útil en situaciones en que necesitemos utilizar un estilo en un sitio y no sea necesario volver a utilizarlo. Sin embargo, en general, es mejor definir todos los estilos usados en un documento en un único lugar. Así es más fácil realizar modificaciones en su estilo sin tener que recorrerlo. Si se necesita hacer algún cambio sólo es necesario hacerlos una vez y el cambio automáticamente se aplica a todo el documento. A veces, sin embargo, se necesita especificar el estilo de un elemento y la forma más fácil de hacerlo es mediante el atributo `style`. En el ejemplo se especifica un estilo para el elemento `<p>`. También se muestra el uso de `` para aplicar un estilo a varios elementos.

Sintaxis CSS

```
<p style="color: green; font-weight: bold; margin-right: 20%;
margin-left: 20%; border-width: 2pt; border-color: blue;">
Este párrafo, y sólo este párrafo, es verde, esta en negrita y tiene
un borde azul.
</p>
<p>
Este párrafo es del color habitual, pero esta <span
style="color: red; font-style: italic;">palabra</span> es diferente
al resto.
</p>
```

Resultado del ejemplo

Este párrafo, y sólo este párrafo, es verde, esta en negrita, tiene unos grandes márgenes y un borde azul.

Este párrafo es del color habitual, pero esta *palabra* es diferente al resto.

8 ESTILOS DE CLASE

Si un documento incluye o se enlaza con una hoja de estilo, todos los estilos definidos en dicha hoja podrán utilizarse en cualquier punto del documento. Si la hoja de estilo especifica el estilo de una etiqueta HTML, entonces todas las etiquetas de ese tipo en el documento utilizarán dicho estilo.

Puede haber casos en que interese aplicar un estilo selectivamente. Por ejemplo, se puede querer que los párrafos de un documento sean unas veces rojos y otras veces azules. En este caso definir un estilo que se aplique a todos los párrafos no será la solución correcta. Podemos obtener el efecto deseado definiendo una clase de estilo y especificando cuando queremos que sea utilizada. Para aplicar una clase de estilo a un elemento HTML, primero se debe definir la clase en la hoja de estilo, y después se utilizará empleando el atributo `class` en cualquier elemento.

Una clase de estilo se define usando un punto e inmediatamente después el nombre que tendrá la clase de estilo:

```
.mi_clase {propiedades}
```

Ejemplo de sintaxis CSS

```
<style type="text/css">
  all.GREENBOLD {color: green; font-weight: bold;}
</style>
```

Uso de la hoja de estilo

```
<h3 class=GREENBOLD>Este titulo es muy verde</h3>
<p class=GREENBOLD>
  Este párrafo usa la clase de estilo GREENBOLD. Se puede utilizar el
  atributo class para especificar una clase de estilo para cualquier
  elemento HTML.
</p>
<blockquote class=GREENBOLD>
  Este bloque usa la clase de estilo GREENBOLD. En consecuencia, es
  verde y está en negrita. Puede ser útil para hacer que los bloques
  resalten del resto de la página.
</blockquote>
```

Resultado del ejemplo

Este titulo es muy verde

Este párrafo usa la clase de estilo GREENBOLD. Se puede utilizar el atributo class para especificar una clase de estilo para cualquier elemento HTML.

Este bloque usa la clase de estilo GREENBOLD. En consecuencia, es verde y está en negrita. Puede ser útil para hacer que los bloques resalten del resto de la página.

Los nombres de clases no pueden contener operadores como: -, +, *, /, %, . Cuando se definen clases de estilo se puede especificar a qué elementos se podrán aplicar dicha clase, o usaremos la palabra clave `all` para indicar que todos los elementos podrán utilizarla. En el siguiente ejemplo se crea una clase de estilo amarillo que podrá utilizar cualquier elemento HTML. También se crea una clase rojo que sólo podrán utilizar párrafos y bloques.

Sintaxis CSS

```
<style type="text/css">
  all.amarillo {color: yellow; font-weight: bold;}
  p.rojo {color: red; font-weight: bold;}
  blockquote.rojo {color: red; font-weight: bold;}
</style>
```

Uso de la hoja de estilo

```

<p class="rojo">Este párrafo es rojo.</p>
<p>Este párrafo es del color por defecto, porque no utiliza la clase
rojo</p>
<blockquote class="rojo">Este bloque usa la clase rojo.</blockquote>
<h4 class="rojo">
    Este título intenta usar la clase rojo, pero no le está permitido
</h4>
<p class="amarillo">Este párrafo es amarillo</p>
<h4 class="amarillo">
    Este título es amarillo porque usa la clase amarillo
</h4>

```

Resultado del ejemplo

Este párrafo es rojo.

Este párrafo es del color por defecto, porque no utiliza la clase rojo

Este bloque usa la clase rojo.

Este título intenta usar la clase rojo, pero no le está permitido

Este párrafo es amarillo

Este título es amarillo porque usa la clase amarillo

Un elemento HTML sólo puede utilizar una clase de estilo. Si se especifican dos o más clases, se utilizará la primera. Por ejemplo, en el siguiente código un párrafo intenta usar las clases rojo y amarillo. Como resultado final se acaba empleando la clase rojo que es la primera que se especifica.

Ejemplo:

```

<p class="rojo" class="amarillo">Otro párrafo rojo.</p>

```

Resultado:

Otro párrafo rojo.

9 ESTILOS INDIVIDUALES

Se pueden crear estilos individuales con nombre. Los elementos HTML pueden utilizar una clase de estilo y un estilo individual con nombre. Normalmente estos se utilizan para expresar excepciones de estilo. Por ejemplo, si un párrafo utiliza la clase de estilo PRINCIPAL, podemos usar el estilo con nombre AZUL1 para expresar alguna diferencia respecto a la clase PRINCIPAL.

También son útiles para definir capas de contenidos HTML posicionadas de forma precisa. Para definir estilos con nombre, en sintaxis CSS, se precede el nombre con el signo #. Para aplicar el estilo a un elemento, se utiliza el nombre de estilo individual como valor del atributo ID.

```

#mi_individual {propiedades}

```

En el siguiente código se define una clase de estilo PRINCIPAL. Esta clase especifica una fuente de 15 puntos, negrita y de color rojo, y una interlinea de 20 puntos. También se define un estilo con nombre llamado AZUL1 cuyo color es azul.

Ejemplo de sintaxis CSS

```

<style type="text/css">
    all.PRINCIPAL {

```



```

    line-height: 20pt;
    font-size: 15pt;
    font-weight: bold;
    color: red;
}
#AZUL1 {color: blue;}
</style>

```

Uso de la hoja de estilo

```

<p class="PRINCIPAL">
    Aquí se puede ver un texto rojo y en negrita. En este párrafo la
    interlinea y el tamaño de la fuente son mayores de lo habitual.
</p>
<p class="PRINCIPAL" id="AZUL1">
    Este párrafo es casi igual al anterior. Está en negrita y su fuente
    es mayor de lo habitual. Aunque usa la clase
    PRINCIPAL es azul porque se utiliza el estilo con nombre AZUL1.
</p>

```

Resultado del ejemplo

Aquí se puede ver un texto rojo y en negrita. En este párrafo la interlinea y el tamaño de la fuente son mayores de lo habitual.

Este párrafo es casi igual al anterior. Esta en negrita y su fuente es mayor de lo habitual. Aunque usa la clase PRINCIPAL es azul porque se utiliza el estilo con nombre AZUL1.

10 CRITERIOS DE SELECCIÓN CONTEXTUAL

Se pueden definir estilos para utilizarse con todos los elementos HTML de una clase particular. Si se necesita un mayor control sobre el uso de estilos podemos hacer que estos se apliquen selectivamente. Podríamos, por ejemplo, querer que el texto enfatizado sea de color verde, pero sólo si el texto enfatizado está en el interior de un título de tamaño 4. Se puede conseguir este nivel de control sobre la aplicación de estilos usando los criterios de selección contextual. Estos, en general, permiten especificar que un estilo se aplicará sólo si un elemento se encuentra anidado dentro de un elemento de otro cierto tipo. Mediante la sintaxis, CSS esto se consigue listando ordenadamente los elementos HTML antes de las llaves.

Sintaxis CSS

```

<style type="text/css">
    h4 em {color: green;}
</style>

```

Uso de la hoja de estilo

```

<h4>El <em>texto enfatizado</em> de este titulo es verde.</h4>
<p>En cambio este <em>texto enfatizado</em> no es verde.</p>

```

Resultado del ejemplo

El **texto enfatizado** de este titulo es verde.

En cambio *este texto enfatizado* no es verde.

Ahora se verá otro ejemplo que hace que las marcas de los elementos de la lista que hereden de al menos dos listas desordenadas sean de color azul.

Sintaxis CSS

```
ul ul li {color: blue;}
```

Se pueden utilizar los criterios de selección contextual para buscar etiquetas, clases, ids o combinaciones de estos. En el siguiente ejemplo se crea la clase MAGENTA que lo colorea todo de magenta. Todos los párrafos MAGENTA que estén dentro de un `<div>` estarán en cursiva. Además los textos dentro de `` anidados dentro de párrafos dentro de un `<div>` en MAGENTA usarán una fuente grande.

Sintaxis CSS

```
<style type="text/css">
  all.MAGENTA {color: magenta;}
  div p.MAGENTA {font-style: italic;}
  div p.MAGENTA b {font-size: large;}
</style>
```

Uso de la hoja de estilo

```
<div class=MAGENTA>
  <h4> Titulo h4 en MAGENTA</h4>
  <p>Este párrafo debería ser magenta y cursivo. Ahora viene un
  <b>texto grande</b>. Conseguimos este efecto con selección
  contextual</p>
</div>

<p class="MAGENTA">Este párrafo todavía es MAGENTA, pero como no está
dentro de un bloque <div>, no es cursivo.</p>
```

Resultado del ejemplo

Titulo h4 en MAGENTA

*Este párrafo debería ser magenta y cursivo. Ahora viene un **texto grande**. Conseguimos este efecto con selección contextual*

Este párrafo todavía es MAGENTA, pero como no esta dentro de un bloque <div>, no es cursivo.

11 COMBINACIÓN DE HOJAS DE ESTILO

Se puede utilizar más de una hoja de estilo para fijar las características de un documento. Esto es deseable si se tienen varias hojas de estilos parciales, de forma que cada una de ellas define diferentes estilos. Supongamos, por ejemplo, que estamos escribiendo un informe sobre los beneficios de un producto de red de una compañía llamada RAD's. Puede que se necesite usar tres hojas de estilo: una definiendo el estilo habitual de los informes, otra que defina el estilo de los productos de red, y otra que defina el estilo de la compañía RAD's. El siguiente ejemplo muestra el uso de varias hojas de estilo en un mismo documento.

```
<style type="text/css" src="http://www.rads.com/estilo/empresa.css"
</style>

<style type="text/css" src="estilos/informe.css"
</style>

<style type="text/css"
  h1 {color: red;} /* tiene preferencia sobre las hojas externas */
```

```
</style>
```

Entre las hojas de estilo externas, la última tiene precedencia sobre las demás. Así, en caso de conflicto, se escoge el estilo de la última hoja de estilo especificada. Los estilos definidos para elementos individuales tienen precedencia sobre los definidos en el elemento `<style>` y sobre los definidos en las hojas externas. En general, los estilos locales se superponen a los generales, como se muestra en el siguiente ejemplo.

Sintaxis CSS

```
<style type="text/css">
  p {color: white;}
  b {color: green;}
</style>
```

Uso de la hoja de estilo

```
<p>Tal y como debería ser este párrafo es de color blanco, <b>y esta
parte en negrita es de color verde</b></p>

<p style="color: yellow;">Este otro párrafo no es de color blanco, <b
style="color: red;"> ni esta parte en negrita es de color verde</b>,
porque se ha usado style para cambiarles el color.</p>
```

Resultado del ejemplo

Tal y como debería ser este párrafo es de color blanco, **y esta parte en negrita es de color verde**

Este otro párrafo no es de color blanco, **ni esta parte en negrita es de color verde**, porque se ha usado style para cambiarles el color.

12 PROPIEDADES DE ELEMENTOS DE BLOQUE

En esta sección se exponen las opciones de formato de los elementos de bloque. Los elementos de bloque comienzan en una nueva línea. Por ejemplo, `<h1>` y `<p>` son elementos de bloque, pero `` no lo es.

12.1 Introducción

Las hojas de estilo tratan a cada elemento de bloque como si estuviera rodeado de una caja. Cada caja puede tener características de estilo propias tales como márgenes, bordes, separadores y una imagen o color de fondo. Los márgenes indican la separación entre el borde de la caja y el borde del documento. Estos bordes pueden tener apariencia plana o tridimensional. Los separadores ("padding") indican la separación entre el borde de los elementos y el contenido de los mismos. También se puede fijar la anchura de los elementos de bloque, bien mediante un valor específico, o bien mediante un porcentaje de la anchura total del documento.

En este caso es redundante fijar los márgenes derecho o izquierdo y la anchura. Si se especifican la anchura y los dos márgenes, el valor del margen izquierdo tiene prioridad sobre los demás valores en caso de conflicto. En este caso el valor del margen derecho especifica la distancia máxima desde el borde derecho de elemento que lo contiene. El valor de anchura es utilizado sólo si no sobrepasa los límites de anchura del elemento que lo contiene. El alineamiento horizontal puede ser a la izquierda, derecha o centrado. Esto se consigue usando la propiedad `float` en CSS. En los siguientes ejemplos se muestra el uso de márgenes, separadores, bordes, fondos y alineamiento.

Sintaxis CSS

```
<style type="text/css">
  p {
```

```

color: #ffffff; /* blanco */
/* márgenes */
margin-left: 20%; margin-right: 20%;
/* anchura del borde */
border-top-width: 10pt; border-bottom-width: 10pt;
border-right-width: 5pt; border-left-width: 5pt;
/* estilo y color del borde */
border-style: outset; border-color: blue;
/* separadores */
padding-top: 10pt; padding-bottom: 10pt;
padding-right: 20pt; padding-left: 20pt;
}
h3 {
/* tamaño y peso de la fuente*/
font-size: 14pt; font-weight: bold;
background-image: url("papel.jpg");
/* centra el titulo y le da una anchura del 90% */
width: 90%; float: center;
borde-color: green; borde-style: solid;
/* todas las partes del borde tienen la misma anchura */
borde-width: 10pt;
/* los separadores del borde son igual de anchos */
padding: 5%;
}
</style>

```

Uso de la hoja de estilo

```

<h3>Titulo h3 con borde sólido y fondo</h3>
<p>Los bordes se usan muy a menudo. Por ejemplo, si un bloque tiene
    borde resalta mucho más que si no lo tiene.
</p>
<p>Este es otro párrafo con borde. Ten cuidado con los bordes, no los
    hagas demasiado anchos, pues de lo contrario ocuparán demasiado
    espacio.
</p>

```

Resultado del ejemplo

Titulo h3 con borde sólido y fondo

Los bordes se usan muy a menudo. Por ejemplo, si un bloque tiene borde resalta mucho más que si no lo tiene.

Este es otro párrafo con borde. Ten cuidado con los bordes, no los hagas demasiado

anchos, pues de lo contrario ocuparán demasiado espacio.

12.2 Márgenes

Los márgenes indican la separación entre el borde del bloque y el borde del documento, o elemento padre. Se pueden fijar los márgenes derecho, izquierdo, superior e inferior. Para ello se deben utilizar los siguientes nombres de propiedad:

Sintaxis CSS

- margin-top
- margin-bottom
- margin-left
- margin-right
- margin

En vez de especificar los dos márgenes se puede utilizar la propiedad [width](#). Se pueden utilizar valores específicos, como 200 puntos, o valores relativos, como el 50% de la anchura del elemento padre.

Es redundante fijar los dos márgenes y la anchura, pues dos de estos tres valores implican el tercero. Para especificar los márgenes por defecto para un documento se deben especificar para la etiqueta `<body>`. En el siguiente ejemplo se fijan dichos márgenes en 20 puntos a derecha e izquierda.

Sintaxis CSS

```
<style type="text/css">
  body {margin-left: 20pt; margin-right: 20pt;}
</style>
```

12.3 Bordes

Se puede fijar la anchura del borde que rodea un elemento de bloque usando las siguientes propiedades.

Sintaxis CSS

- border-top-width
- border-bottom-width
- border-left-width
- border-right-width
- border-width

Se puede fijar el estilo del borde usando la propiedad [border-style](#) de CSS. Los valores que pueden tomar son [solid](#), [double](#), [groove](#), [ridge](#), [inset](#) y [outset](#).



12.4 Separadores

Los separadores indican la distancia entre el borde de un elemento y su contenido. El separador se muestra incluso si el borde del elemento no lo hace. Se puede fijar el tamaño de los separadores de un elemento de bloque utilizando las siguientes propiedades.

Sintaxis CSS

- padding-top
- padding-bottom
- padding-left
- padding-right

12.5 Herencia de propiedades

Las características de anchura, márgenes, bordes y separadores de los elementos padre no son heredadas por sus hijos. Sin embargo, a primera vista, a veces puede parecer que sí son heredados, pues los valores de los elementos padre afectan a sus elementos hijos. Supongamos que a un elemento `<div>` le fijamos un margen izquierdo de valor 10 puntos. De esta forma la caja que le rodea se encuentra desplazada 10 puntos hacia la derecha. Además supondremos que no tiene bordes ni separadores. Todos sus elementos hijo estarán pegados a su margen izquierdo, que como está 10 puntos desplazado hacia la derecha, causará un efecto parecido si ellos mismos también tuviesen borde izquierdo.

Pensemos en qué ocurriría si los hijos heredasen estas características de sus padres. El bloque `<div>` está indentado 10 puntos. Sus hijos a su vez estarán indentados otros 10 puntos con respecto a él, con lo cual el aspecto global es que los hijos estarían indentados 20 puntos.