

CSS

1 - ¿Qué son las hojas de estilo (CSS)?

CSS son las siglas de Cascade Style Sheet que traducido significa hojas de estilo en cascada.

Las hojas de estilo es una tecnología que nos permite controlar la apariencia de una página web. En un principio, los sitios web se concentraban más en su contenido que en su presentación.

HTML no pone mucha atención en la apariencia del documento. CSS describe cómo los elementos dispuestos en la página son presentados al usuario. CSS es un gran avance que complementa el HTML y la Web en general.

Con CSS podemos especificar estilos como el tamaño, fuentes, color, espaciado entre textos y recuadros así como el lugar donde disponer texto e imágenes en la página.

El lenguaje de las Hojas de Estilo está definido en la Especificaciones CSS1, CSS2 y CSS3 del World Wide Web Consortium (W3C), es un estándar aceptado por toda la industria relacionada con la Web, o por lo menos, gran parte de navegadores (es verdad el IExplorer de Microsoft nos puede dar un dolor de cabeza con los navegadores antiguos). Podemos visitar W3C

Veremos que podemos asociar las reglas de estilo a las marcas HTML de tres maneras: directamente a la marca, en el head de la página o agrupar las reglas de estilo en un archivo independiente con extensión *.css

En este curso veremos las tres metodologías, pero pondremos énfasis en la tercera forma, que es la más adecuada para separar el contenido de la página y la forma como se debe representar la misma por medio de la hoja de estilo.

El curso brinda un concepto teórico corto, luego un problema resuelto que invito a ejecutar, modificar y jugar con el mismo. Por último, y lo más importante, un ejercicio propuesto que nos permitirá saber si podemos aplicar el concepto.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
<p style="color:#000000;background-color:yellow;font-family:vardana; font-size:25px;text-align:center">Esto es un ejemplo</p>
</body>
</html>
```

2 - Definición de estilos a nivel de marca HTML.

Es la forma más fácil pero menos recomendada para aplicación de un estilo a una marca HTML. Se define en la propiedad style los estilos a definir para dicha marca.

Es común definir estilos directamente en las marcas HTML cuando estamos diseñando la página y posteriormente trasladar el estilo creado a una hoja de estilos.

La sintaxis para definir un estilo a una marca HTML es la siguiente:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
</head>
<body>
<h1 style="color:#ff0000;background-color:#ffff00">
  Este mensaje es de color rojo sobre fondo amarillo.
</h1>
</body>
</html>
```

Por defecto, todo navegador tiene un estilo definido para cada marca HTML, lo que hacemos con la propiedad style es redefinir el estilo por defecto. En este problema definimos que la marca h1 defina como color de texto el rojo y como color de fondo el amarillo:

```
<h1 style="color:#ff0000;background-color:#ffff00">
  Este mensaje es de color rojo sobre fondo amarillo.
</h1>
```

Veremos más adelante que hay muchas propiedades en CSS. En este primer ejemplo inicializamos las propiedades color (define el color del texto) y background-color (define el color de fondo del texto).

Cada vez que inicializamos una propiedad debemos separarla de la siguiente por punto y coma.

Cuando definimos estilos directamente en las marcas HTML, tenemos que tener en cuenta que el estilo se aplica únicamente a la marca donde inicializamos la propiedad style, es decir, si tenemos dos secciones h1, deberemos definir la propiedad style para cada marca:

```
<h1 style="color:#ff0000;background-color:#ffff00">
  Primer título
</h1>
<h1 style="color:#ff0000;background-color:#ffff00">
  Segundo título
</h1>
```

Como podemos observar, más allá que los dos estilos son exactamente iguales, estamos obligados a definirlos para cada marca HTML.

3 - Definición de estilos a nivel de página.

Es la definición de estilos en una sección de la cabecera de la página HTML donde podemos definir estilos que se aplican a las distintas marcas HTML de la página. El problema del concepto anterior donde debíamos crear un estilo similar para la marca

h1 se puede resolver en forma más adecuada empleando la definición de estilos a nivel de página.

Problema: Mostrar dos títulos con texto de color rojo sobre fondo amarillo.

tres1.html

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Problema</title>
<style type="text/css">
h1 {color:#ff0000;background-color:#ffff00}
</style>
</head>
<body>
<h1>Primer título</h1>
<h1>Segundo título</h1>
</body>
</html>
```

Podemos observar que en la cabecera de la página definimos la sección:

```
<style type="text/css">
h1 {color:#ff0000;background-color:#ffff00}
</style>
```

Debe estar encerrada por la marca style. En este ejemplo indicamos al navegador que en todos los lugares de esta página donde se utilice la marca h1 debe aplicar como estilo de color de texto el rojo y fondo el amarillo. Podemos observar que es mucho más eficiente que definir los estilos directamente sobre las marcas HTML.

Podemos definir estilos para muchas marcas en la sección style:

tres2.html

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Problema</title>
<style type="text/css">>
h1 {color:#ff0000}
h2 {color:#00ff00}
h3 {color:#0000ff}
</style>
</head>
<body>
<h1>rojo</h1>
<h2>verde</h2>
<h3>azul</h3>
</body>
</html>
```

4 - Propiedades relacionadas a fuentes.

Contamos con una serie de propiedades relacionadas a fuentes:

```
font-family  
font-size  
font-style  
font-weight  
font-variant
```

Podemos inicializar algunas o todas las propiedades relacionadas a fuentes, un ejemplo:

```
<!DOCTYPE html>  
<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
<title>Problema</title>  
<style type="text/css">  
h1 {  
    font-family:times new roman;  
    font-size:30px;  
    font-style:italic;  
    font-weight:bold;  
}  
h2 {  
    font-family:verdana;  
    font-size:20px;  
}  
</style>  
</head>  
<body>  
<h1>Titulo de nivel 1</h1>  
<h2>Titulo de nivel 2</h2>  
</body>  
</html>
```

Como podemos observar, hemos definido dos reglas de estilos para las marcas h1 y h2, eso significa que el navegador utilizará esas reglas de fuentes para todas las partes de la página que se utilicen dichas marcas HTML.

La primera regla definida para la marca h1 es:

```
h1 {  
    font-family:times new roman;  
    font-size:30px;  
    font-style:italic;  
    font-weight:bold;  
}
```

Recordemos que para definir la regla de estilo debemos indicar el nombre de la marca HTML a la que definiremos el estilo y luego, entre paréntesis, todas las propiedades y sus valores separados por punto y coma.

La primera propiedad inicializada es font-family, algunas de las fuentes más comunes que puede acceder el navegador son:

Arial
 Arial Black
 Arial Narrow
 Courier New
 Georgia
 Impact
 Tahoma
 Times New Roman
 Verdana

En caso que la fuente no esté disponible el navegador selecciona el estilo por defecto para esa marca HTML.

Podemos definir varias fuentes por si acaso alguna no se encuentra disponible para el navegador, esto lo hacemos separando por coma todas las fuentes (se eligen de izquierda a derecha):

```
font-family: verdana, arial, georgia;
```

La segunda propiedad inicializada es font-size, hay varias medidas para indicar el tamaño de la fuente (veremos más adelante otros sistemas de medida), en nuestro caso indicamos en píxeles:

```
font-size:30px;
```

La tercera propiedad es font-style, que puede tener los siguientes valores :

normal
 italic
 oblique

La última propiedad es font-weight, pudiendo tomar los siguientes valores:

normal
 bold
 bolder
 lighter
 100
 200
 300
 400
 500
 600
 700
 800
 900

Esta propiedad indica el peso de la fuente (mientras tenga un valor mayor los caracteres serán más rellenos).

La segunda regla define sólo dos propiedades relacionadas a la fuente:

```
h2 {
  font-family:verdana;
  font-size:20px;
}
```

Las propiedades que no se inicializan quedan como responsabilidad al navegador, quien elegirá los valores correspondientes.

Existe una última propiedad no utilizada en este ejemplo que es font-variant que puede asumir estos dos valores:

```
small-caps
normal
```

Define si la fuente se muestra en mayúsculas tipo normal o pequeñas.

5 - Agrupación de varias marcas HTML con una misma regla de estilo.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Problema</title>
<style type="text/css">
h1,h2,h3 {
  font-family:verdana;
  color:#0000ff;
}
</style>
</head>
<body>
<h1>Título de nivel 1</h1>
<h2>Título de nivel 2</h2>
<h3>Título de nivel 3</h3>
</body>
</html>
```

Es decir, separamos por coma todas las marcas a las que se aplicará la misma regla de estilo:

```
h1,h2,h3 {
  font-family:verdana;
  color:#0000ff;
}
```

6 - Definición de varias reglas para una misma marca HTML.

En algunas circunstancias, es necesario desglosar la inicialización de propiedades en distintas reglas.

Supongamos que queremos todas las cabeceras con la misma fuente pero tamaños de fuente distinta, luego podemos implementarlo de la siguiente manera:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Problema</title>
<style type="text/css">
```

```

h1,h2,h3,h4,h5,h6 {
  font-family:Verdana;
}
h1 {
  font-size:40px;
}
h2 {
  font-size:30px;
}
h3 {
  font-size:25px;
}
h4 {
  font-size:20px;
}
h5 {
  font-size:15px;
}
h6 {
  font-size:10px;
}
</style>
</head>
<body>
<h1>Título de nivel 1</h1>
<h2>Título de nivel 2</h2>
<h3>Título de nivel 3</h3>
<h4>Título de nivel 4</h4>
<h5>Título de nivel 5</h5>
<h6>Título de nivel 6</h6>
</body>
</html>

```

Es decir, podemos inicializar un conjunto de marcas HTML con una serie de propiedades de estilo comunes. Luego agregamos en forma individual las propiedades no comunes a dichas marcas:

```

h1,h2,h3,h4,h5,h6 {
  font-family:Verdana;
}
h1 {
  font-size:40px;
}

```

Es decir, a la marca h1 realmente le hemos definido 2 propiedades: font-family y font-size. Lo mismo para las otras marcas HTML.

7 - Propiedades relacionadas al texto.

A una de estas propiedades ya la vimos cuando comenzamos con los primeros conceptos: color, nos permite definir el color del texto, lo podemos indicar por medio de tres valores hexadecimales que indican la mezcla de rojo, verde y azul. Por ejemplo si queremos rojo puro debemos indicar:

```
color:#ff0000;
Si queremos verde puro:
color:#00ff00
Si queremos azul puro:
color:#0000ff
```

Luego si queremos amarillo debemos mezclar el rojo y el verde:

```
color:#ffff00
```

Hay muchos programas que nos permiten seleccionar un color y nos descomponen dicho valor en las proporciones de rojo, verde y azul.

Otra forma de indicar el color, si tenemos los valores en decimal, es por medio de la siguiente sintaxis:

```
color:rgb(255,0,0);
```

Es decir, por medio de la función rgb (red,green,blue), indicamos la cantidad de rojo, verde y azul en formato decimal.

La segunda propiedad relacionada al texto es text-align, que puede tomar alguno de estos cuatro valores:

```
left
right
center
justify
```

Si especificamos:

```
text-align:center;
```

El texto aparecerá centrado. Si queremos justificar a derecha, emplearemos el valor right y si queremos a la izquierda, el valor será left.

La tercera propiedad relacionada al texto que podemos emplear es text-decoration que nos permite entre otras cosas que aparezca subrayado el texto, tachado o una línea en la parte superior, los valores posibles de esta propiedad son:

```
none
underline
overline
line-through
```

Como ejemplo, definiremos estilos relacionados al texto para las marcas de cabecera h1, h2 y h3:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Problema</title>
<style type="text/css">
h1 {
  color:#ff0000;
```



```

text-align:left;
text-decoration:underline;
}
h2 {
color:#dd0000;
text-align:center;
text-decoration:underline;
}
h3 {
color:#aa0000;
text-align:right;
text-decoration:underline;
}
</style>
</head>
<body>
<h1>Título de nivel 1.</h1>
<h2>Título de nivel 2.</h2>
<h3>Título de nivel 3.</h3>
</body>
</html>

```

Es decir, para los títulos de nivel 1 tenemos la regla:

```

h1 {
color:#ff0000;
text-align:left;
text-decoration:underline;
}

```

Es decir, color de texto rojo intenso, el texto debe aparecer de izquierda a derecha y subrayado.

Luego para la marca h2 tenemos:

```

h2 {
color:#dd0000;
text-align:center;
text-decoration:underline;
}

```

El color sigue siendo rojo pero un poco más oscuro, el texto debe aparecer centrado y subrayado.

Y por último:

```

h3 {
color:#aa0000;
text-align:right;
text-decoration:underline;
}

```

Para los títulos de nivel tres, el texto es rojo más oscuro, alineado a derecha y subrayado.

8 - Más propiedades relacionadas al texto.

Vimos en el concepto anterior las propiedades:

color
text-align
text-decoration

Ahora veremos el objetivo de las siguientes propiedades que nos faltan:

letter-spacing
word-spacing
text-indent
text-transform

La propiedad letter-spacing y word-spacing permiten indicar el espacio que debe haber entre los caracteres y entre las palabras.

La propiedad text-indent, indenta la primera línea de un texto. A partir de la segunda línea, el texto aparece sin indentación. Podemos indicar un valor negativo con lo que la indentación es hacia la izquierda.

Por último, la propiedad text-transform puede inicializarse con alguno de los siguientes valores:

none - capitalize - lowercase - uppercase

Cada uno de estos valores transforman el texto como sigue:

capitalize: Dispone en mayúsculas el primer carácter de cada palabra.

lowercase: Convierte a minúsculas todas las letras del texto.

uppercase: Convierte a mayúsculas todas las letras del texto.

none: No provoca cambios en el texto.

El siguiente ejemplo define reglas para las marcas h1 y p:

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Problema</title>
<style type="text/css">
h1 {
  letter-spacing:10px;
  word-spacing:30px;
  text-transform:capitalize;
}
p {
  text-indent:20px;
}
</style>
</head>
<body>
<h1>Este es un título de nivel 1</h1>
<p>Todo el texto siguiente se encuentra encerrada en la marca de párrafo
y con el objetivo de ver el efecto de la propiedad text-indent.
La propiedad text-indent podemos inicializarla con un valor positivo,
```

```

como es el caso actual o podemos inicializarla con un valor negativo lo que
provocará que el texto de la primera línea del párrafo
comience en una columna inferior al de todo el bloque. </p>
</body>
</html>

```

La cabecera de nivel uno, tiene la siguiente regla:

```

h1 {
  letter-spacing:10px;
  word-spacing:30px;
  text-transform:capitalize;
}

```

Es decir que las letras deben tener una separación de 10 píxeles, las palabras deben estar separadas por 30 píxeles y por último deben disponerse en mayúsculas la primera letra de cada palabra.

Para la marca p tenemos la siguiente regla:

```

p {
  text-indent:20px;
}

```

Es decir, la primera línea del párrafo deberá imprimirse 20 píxeles a la derecha donde normalmente debería aparecer.

9 - Herencia de propiedades de estilo.

Ahora veremos que el conjunto de marcas HTML forman en sí un árbol que en su raíz podemos identificar la marca body del cual se desprenden otras marcas contenidas en esta sección, como podrían ser las marcas h1,h2,h3,h4,h5,h6,p,div luego estas en su interior contienen otras marcas HTML como podrían ser em,b,i,pre etc.

Ahora veremos con ejemplos que hay muchos estilos que se heredan (todos los vistos hasta el momento se heredan), es decir si definimos la propiedad color para la marca h1, luego si dicha marca incorpora un texto con la marca b (bold) en su interior, la propiedad color de la marca b tendrá el mismo valor que la propiedad h1 (es decir la marca b hereda las propiedades de la marca h1)

Con un ejemplo veremos el resultado de la herencia de propiedades entre las marcas HTML:

nueve1.html

```

<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Problema</title>
<style type="text/css">
body {
  color:#0000ff;
  font-family:verdana;
}
</style>
</head>

```

```
<body>
<h1>Este es un título de nivel 1 y con la marca 'em' la palabra:
<em>Hola</em></h1>
<p>Todo este párrafo debe ser de color azul ya que lo hereda de la
marca body.</p>
</body>
</html>
```

En este ejemplo hemos definido la siguiente regla para la marca body:

```
body {
  color:#0000ff;
  font-family:verdana;
}
```

Inicializamos la propiedad color con el valor de azul y la fuente de tipo verdana. Con esto todas las marcas contenidas en el body que no redefinan estas dos propiedades recibirán los valores aquí definidos. En este ejemplo la cabecera de primer nivel es decir h1, el párrafo y el hipervínculo tienen como color el azul y la fuente es de tipo verdana.

Ahora bien en muchas situaciones podemos redefinir propiedades para marcas contenidas, veamos cómo podemos hacer que el texto contenido en las marcas em y p aparezcan de color distinto:

nueve2.html

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Problema</title>
<style type="text/css">
body {
  color:#0000ff;
  font-family:verdana;
}
em {
  color:#008800;
}
p {
  color:#999999;
}
</style>
</head>
<body>
</body>
<h1>Este es un título de nivel 1 y con la marca 'em' la palabra:
<em>Hola</em></h1>
<p>Todo este párrafo debe ser de color gris ya que lo redefine la
marca p y no lo hereda de la marca body.</p>
</html>
```

Ahora hemos definido tres reglas, la primera igual que el problema anterior, define la propiedad color en azul y la fuente de tipo verdana para la marca body:

```
body {
  color:#0000ff;
  font-family:verdana;
}
```

La segunda regla define la propiedad color en verde para la marca em, con esto no heredará el color azul de la marca body (que es la marca padre):

```
em {
  color:#008800;
}
```

Algo similar hacemos con la marca p para indicar que sea de color gris:

```
p {
  color:#999999;
}
```

Pero podemos ver que todas las marcas heredan la fuente verdana ya que ninguna marca la sobrescribe.

10 - Definición de un estilo en función del contexto.

Este otro recurso que provee las css es la definición de un estilo para una marca HTML siempre y cuando la misma esté contenida por otra marca determinada. Sólo en ese caso el estilo para dicha marca se activará.

Supongamos que queremos que cuando disponemos un texto encerrado por la marca b (bold) dentro de un párrafo el color de la misma sea verde. Pero si la marca b está contenida por la marca h1 el color debe ser rojo, luego la sintaxis del problema es:

diez1.html

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Problema</title>
<style type="text/css">
p b{
  color:#0000ff;
}
h1 b{
  color:#ff0000;
}
</style>
</head>
<body>
<h1>Acá tenemos un título de nivel uno, luego un bloque con
la marca bold debe aparecer <b>así</b></h1>
<p>
Luego si escribimos un párrafo y encerramos un bloque con la marca bold
debe aparecer <b>así</b>
</p>
</body>
</html>
```

Es importante analizar la sintaxis para la definición de estilos en función del contexto. Tenemos:

```
p b{
  color:#0000ff;
}
```

Estamos diciendo que todas los bloques de la página que contengan la marca b (bold) y que estén contenidas por la marca p (párrafo) se pinten de azul. Si bien hay dos bloques en la página que están encerrados por la marca b, solo uno se pinta de color azul, el otro bloque se pinta de color verde, ya que tenemos:

```
h1 b{
  color:#ff0000;
}
```

Es decir, activar el color rojo para el contenido encerrado por la marca b, siempre y cuando se encuentre contenido por la marca h1.

No confundir con la sintaxis para definir reglas de estilo a dos marcas que se hace separando por coma las marcas HTML:

```
h1,b{
  color:#ff0000;
}
```

Es decir que el texto contenido por las marcas h1 y b deben aparecer de color rojo.

Se pueden definir estilos en función del contexto, con mayor precisión, como por ejemplo:

```
div h1 em {
  color:#ff0000;
}
```

Con esto estamos diciendo que se debe pintar de color rojo el texto contenido por la marca em siempre y cuando esté contenida en un bloque con la marca h1 y esta a su vez dentro de un div.

La página completa queda codificada de la siguiente forma:

diez2.html

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Problema</title>
<style type="text/css">
div h1 em {
  color:#ff0000;
}
</style>
</head>
<body>
<div>
<h1>Este es un titulo de nivel 1 dentro de un <em>div</em></h1>
</div>
```

```
<h1>Este es un titulo de nivel 1 fuera de un <em>div</em></h1>
</body>
</html>
```

11 - Definición de hojas de estilo en un archivo externo.

Hasta ahora hemos visto la definición de estilos a nivel de marca HTML y la definición de estilos a nivel de página. Dijimos que la primera forma es muy poco recomendada y la segunda es utilizada cuando queremos definir estilos que serán empleados sólo por esa página.

Ahora veremos que la metodología más empleada es la definición de una hoja de estilo en un archivo separado que deberá tener la extensión css.

Este archivo contendrá las reglas de estilo (igual como las hemos visto hasta ahora) pero estarán separadas del archivo HTML.

La ventaja fundamental es que con esto podemos aplicar las mismas reglas de estilo a parte o a todas las páginas del sitio web. Veremos que esto será muy provechoso cuando necesitemos hacer cambios de estilo (cambiando las reglas de estilo de este archivo estaremos cambiando la apariencia de múltiples páginas del sitio).

También tiene como ventaja que al programador le resulta más ordenado tener lo referente a HTML en un archivo y las reglas de estilo en un archivo aparte.

Otra ventaja es que cuando un navegador solicita una página, se le envía el archivo HTML y el archivo CSS, quedando guardado este último archivo en la caché de la computadora, con lo cual, en las sucesivas páginas que requieran el mismo archivo de estilos, ese mismo archivo se rescata de la caché y no requiere que el servidor web se lo reenvíe (ahorrando tiempo de transferencia).

Ahora veremos la primera página HTML que tiene asociada una hoja de estilo en un archivo externo. El archivo HTML es (pagina.html):

once.html

```
<!DOCTYPE html>
<html>
<head>
<title>Problema</title>
<link rel="StyleSheet" href="estilos.css" type="text/css">
</head>
<body>
<h1>Definición de hojas de estilo en un archivo externo.</h1>
<p>
Hasta ahora hemos visto la definición de estilos a nivel de marca HTML
y la definición de estilos a nivel de página. Dijimos que la primera
forma es muy poco recomendada y la segunda es utilizada cuando queremos definir
estilos que serán empleados solo por esa página.
Ahora veremos que la metodología más empleada es la definición
de una hoja de estilo en un archivo separado que deberá tener la extensión
css.
</p>
</body>
</html>
```

El archivo que tiene las reglas de estilo es (estilos.css):

estilos.css

```
body {
  background-color:#eafadd;
}
h1 {
  color:#0000cc;
  font-family:times new roman;
  font-size:25px;
  text-align:center;
  text-decoration:underline;
}
p {
  color:#555555;
  font-family:verdana;
  text-align:justify;
}
```

Como podemos observar, para indicar el archivo de estilos externo, debemos agregar en la cabecera (head) del archivo HTML la siguiente marca:

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

La propiedad href hace referencia al archivo externo que afectará la presentación de esta página. En la propiedad type, indica al navegador cual es el formato de archivo. El atributo rel se usa para definir la relación entre el archivo enlazado y el documento HTML.

Sólo queda probar esta página funcionando.

De ahora en adelante nos acostumbraremos a trabajar con hojas de estilo definidas en un archivo externo, que es en definitiva la forma más común de desarrollar un sitio web aplicando CSS.

12 - Definición de estilos por medio de clases.

En muchas situaciones una regla de estilo puede ser igual para un conjunto de marcas HTML, en esos casos conviene plantear una regla de estilo con un nombre genérico que posteriormente se puede aplicar a varias marcas de HTML.

Para conocer la sintaxis para la definición de clases veamos un ejemplo, la pagina.html es:

doce.html

```
<!DOCTYPE html>
<html>
<head>
<title>Problema</title>
<link rel="StyleSheet" href="estilos.css" type="text/css">
</head>
<body>
<h1 class="resaltado">Titulo de nivel 1</h1>
<p>
Este párrafo muestra el resultado de aplicar la clase .resaltado en la última
```



```
<span class="resaltado">palabra.</span>
</p>
</body>
</html>
```

La hoja de estilo externa (estilos.css) es:

```
body {
    background-color:#eeeeee;
}
.resaltado{
    color:#000000;
    background-color:#ffff00;
    font-style:italic;
}
```

La sintaxis para definir una clase aplicable a cualquier marca HTML es:

```
.resaltado{
    color:#000000;
    background-color:#ffff00;
    font-style:italic;
}
```

Es decir, la inicializamos con el carácter punto y seguidamente un nombre de clase. Dentro definimos las reglas de estilo como hemos venido trabajando normalmente. El nombre de la clase no puede comenzar con un número. Luego, para indicar que una marca sea afectada por esta regla:

```
<h1 class="resaltado">Titulo de nivel 1</h1>
```

Es decir, agregamos la propiedad class y le asignamos el nombre de la clase (sin el punto).

Podemos inicializar tantas marcas HTML con esta regla como necesitemos:

```
<p>
Este parrafo muestra el resultado de aplicar la clase .resaltado en la última
<span class="resaltado">palabra.</span>
</p>
```

Aquí definimos la propiedad class a la marca span y le asignamos la misma clase aplicada anteriormente a la marca h1.

13 - Definición de estilos por medio de Id.