



FUNDAMENTOS DE HTML Y JAVASCRIPT

FUNDAMENTOS DE HTML Y JAVASCRIPT

Hojas de estilo CSS

1. Introducción

Aunque el estándar XHTML proporciona un gran número de etiquetas para dar formato a la información de una página, su uso presenta algunos inconvenientes. Entre ellos podríamos destacar los siguientes:

- Opciones de estilo limitadas. El juego de etiquetas HTML de formato con sus respectivos atributos resulta bastante limitados para los requerimientos de las aplicaciones actuales.
- Dificultad en el mantenimiento de las páginas. La anidación de diferentes etiquetas hasta conseguir aplicar el formato adecuado hace que cualquier modificación posterior resulte compleja y susceptible de errores.
- Repetición constante de grupos etiquetas. Si queremos aplicar un determinado en distintas partes de la página a través de la anidación de una serie de etiquetas, no tendremos más remedio que incluir el bloque de etiquetas en todas las partes del documento donde se requiera.

Como solución al problema del estilo, el W3C desarrolló un estándar bajo el nombre de Cascading Style Sheet (CSS), que tiene como objetivo poder definir hojas de estilo para ser aplicadas sobre un documento HTML, permitiendo separar los datos de la información de formato y estilo de los mismos.

Las hojas de estilo CSS se basan en la definición de una **serie de propiedades de estilo asociadas a determinados tipos de etiquetas e independientes de su contenido**, de forma que al ser utilizadas estas etiqueta en el documento se apliquen automáticamente las diferentes propiedades y opciones de estilo definidas para las mismas.

El juego de propiedades de estilo proporcionado por el estándar CSS es bastante numeroso, ofreciendo a los diseñadores grandes posibilidades para el formato de páginas.



FUNDAMENTOS DE HTML Y JAVASCRIPT

2. Creación de una hoja de estilo

En la figura 1 se puede ver el aspecto de una hoja de estilo básica.

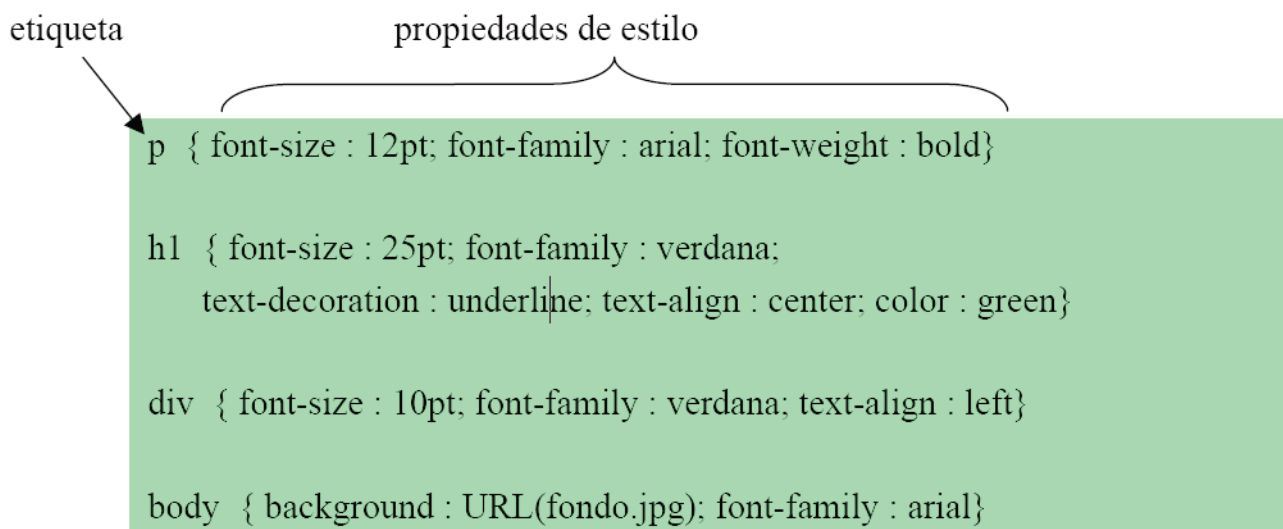


Figura. 1.

Como vemos, en una hoja de estilo se distinguen dos partes bien diferenciadas:

- Etiqueta. Indica el nombre de la etiqueta sobre la que se aplicará las propiedades de estilo definidas entre los símbolos “{” y “}”.
- Propiedades de estilo. Las propiedades de estilo están formadas por parejas nombre:valor, separadas por un “;”, donde *nombre* representa el nombre de la propiedad de estilo que se quiere aplicar y *valor* el valor asignado a la propiedad. Los valores asignados a las propiedades de estilo pueden ser valores numéricos con sus unidades de medida correspondientes (centímetros -cm-, puntos -pt-, pixels -px- o pulgadas -in-), o también pueden ser valores de texto predefinidos.

Una vez definida la hoja de estilo, cada vez que se haga uso de una de estas etiquetas en el documento se aplicarán las propiedades definidas, si bien el ámbito de aplicación de estas hojas de estilo dependerán de donde se hayan definido.

3. Tipos de hojas de estilo

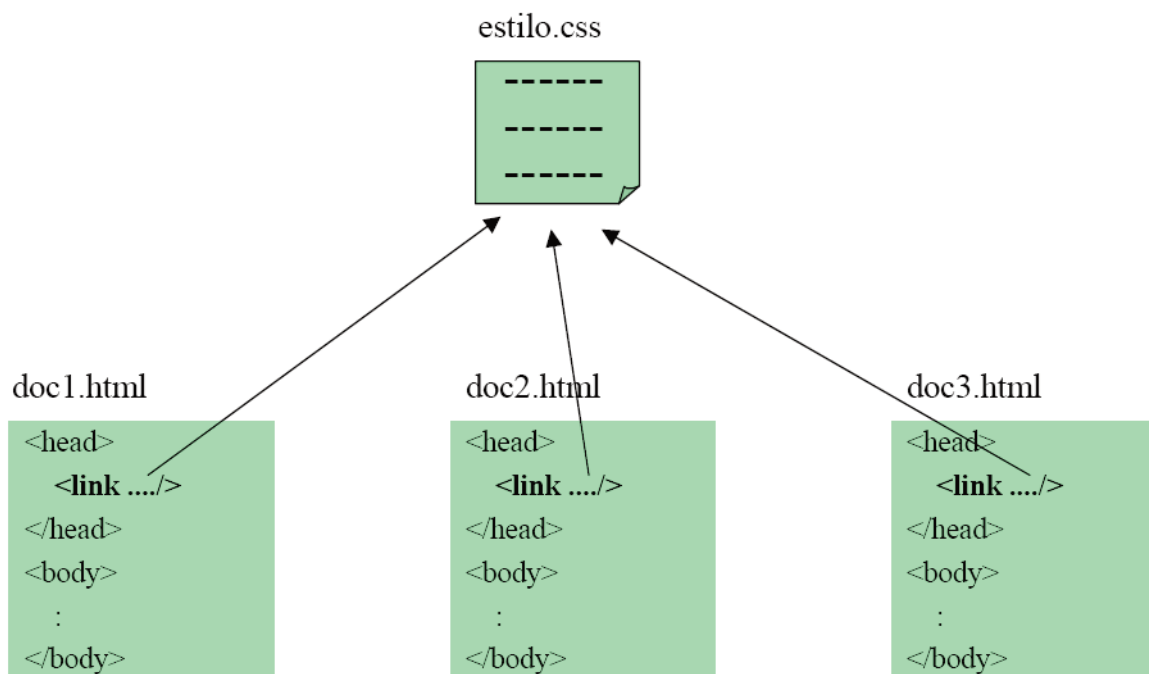
Según el lugar en que se defina una hoja de estilo y, por tanto, el ámbito de utilización de la misma, distinguimos tres tipos de hojas de estilo:

- Externas
- Internas
- En línea

3.1 Hojas de estilo externas

Una hoja de estilo externa se define en un archivo de texto con extensión .css, tal cual se indica en la figura 1. Esto permite que la hoja de estilo pueda ser utilizada por múltiples páginas XHTML o HTML.

Para vincular un documento HTML con una hoja de estilo externa será necesario incluir en la cabecera del documento (interior del elemento <head>), una etiqueta de tipo <link> mediante la que se indicará la dirección de la hoja de estilo que se quiere utilizar (figura 2).



FUNDAMENTOS DE HTML Y JAVASCRIPT

El formato de la etiqueta `<link>` es el que se indica a continuación:

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

Los atributos utilizados por esta etiqueta son:

- **rel.** Tipo de enlace. En el caso de hojas de estilo el valor de este atributo será siempre "stylesheet".
- **href.** Dirección de la hoja de estilo. Se indicará la dirección relativa del archivo .css respecto al documento
- **type.** Tipo de hoja de estilo. Al tratarse de hojas de estilo CSS su valor será text/css.

En la figura 3 se muestra el conjunto completo de hoja de estilo, documento XHTML de utilización y página resultante.

estilo.css

```
p { font-size : 12pt; font-family : arial; font-weight : bold}

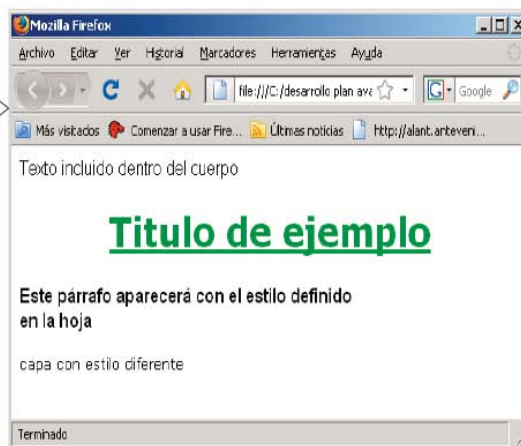
h1 { font-size : 25pt; font-family : verdana;
    text-decoration : underline; text-align : center; color : green}

div { font-size : 10pt; font-family : verdana; text-align : left}

body { background : URL(fondo.jpg); font-family : arial}
```

ejemplo.html

```
<html>
<head>
    <link rel="stylesheet" type="text/css"
          href="estilo.css"/>
</head>
<body>
    Texto incluido dentro del cuerpo
    <h1>Titulo de ejemplo</h1>
    <p>Este párrafo aparecerá con el estilo definido<br>
    en la hoja</p>
    <div>
    </div>
    capa con estilo diferente
    <div>
    </div>
</body>
</html>
```



3.2 Hojas de estilo internas

Las hojas de estilo internas se definen en el interior del documento donde van a ser utilizadas, concretamente, en la cabecera del mismo en el interior de la etiqueta `<style>`:

```
<head>
  <style>
    //definición de propiedades de
    //estilo
  </style>
</head>
```

El formato utilizado en la definición de las propiedades de estilo para cada etiqueta es exactamente igual que en el caso de las hojas de estilo externas.

En el documento de ejemplo anterior podemos conseguir el mismo efecto de estilos definiéndolo como se indica en el siguiente listado:

```
<html>
  <head>
    <style>
      p { font-size : 12pt; font-family : arial;
font-weight : bold}

      h1 { font-size : 25pt; font-family : verdana;
          text-decoration : underline;
text-align : center; color : green}

      div { font-size : 10pt; font-family : verdana;
text-align : left}

      body { background : URL(fondo.jpg);
font-family : arial}
    </style>
  </head>
  <body>
    Texto incluido dentro del cuerpo
```



FUNDAMENTOS DE HTML Y JAVASCRIPT

```
<h1>Titulo de ejemplo</h1>
<p>Este párrafo aparecerá con el estilo definido<br/>
    en la hoja</p>
<div>
    capa con estilo diferente
</div>

</body>
</html>
```

3.3 Hojas de estilo en línea

Hay circunstancias en las que puede interesar aplicar sobre una determinada etiqueta de la página una serie de propiedades de estilo, sin que esto afecte al resto de etiquetas de su mismo tipo que estén definidas en el documento.

Para ello, debemos definir los valores de estas propiedades de estilo dentro del atributo *style* de la etiqueta en la que se quieren aplicar; es lo que se conoce como una **hoja de estilo en línea**. La sintaxis para la definición de las propiedades será la misma que utilizamos en los casos anteriores.

Otra situación en la que puede resultar útil la definición de una hoja de estilo en línea sobre una etiqueta se da cuando queremos aplicar sobre ésta un estilo diferente al definido en un estilo superior. Los estilos en línea tienen pues **prioridad** sobre los estilos internos y externos.

Por ejemplo, si quisiéramos añadir al documento de ejemplo anterior un segundo párrafo con un estilo distinto al definido para la etiqueta `<p>`, sería:

```
<body>
    Texto incluido dentro del cuerpo
    <h1>Titulo de ejemplo</h1>
    <p>Este párrafo aparecerá con el estilo definido<br/>
        en la hoja</p>
    <div>
        capa con estilo diferente
    </div>
    <p style="color:blue;font-weight:normal">
        segundo párrafo con estilo diferente al
global</p>
</body>
```



FUNDAMENTOS DE HTML Y JAVASCRIPT

4. Principales propiedades de estilo

Seguidamente, comentaremos algunas de las propiedades de estilo más comúnmente utilizadas del estándar CSS2, para lo cual realizaremos una subdivisión en grupos según el tipo de formato llevado a cabo por la propiedad.

4.1 Fuentes y colores

Definen estilos relativos al tipo de letra. Entre las más utilizadas están:

- **Color**. Define el color del texto. Se puede asignar el nombre del color en inglés o la combinación RGB correspondiente en hexadecimal (por ejemplo, color:#FF12AB. Esta propiedad la admiten casi todas las etiquetas de HTML.
- **font-size**. Indica el tamaño de la fuente. Se pueden utilizar valores absolutos, indicando el tipo de medida utilizada, o valores relativos como: xx-small, x-small, small, médium, large, x-large o xx-large.
- **font-family**. Define el nombre del tipo de letra que se quiere aplicar.
- **font-weight**. Establece el grosor de los caracteres. Sus posibles valores pueden ser números del 100 al 900 en pasos de 100 o valores predefinidos como bold, normal, bolder o lighter.
- **font-style**. Estilo que queremos aplicar al texto (italic, normal u oblique).

4.2 Texto

Definen diferentes características del texto. Entre las más utilizadas están:

- **text-align**. Indica la alineación del texto (left, right, center o justified).
- **text-decoration**. Decoración del texto, es decir, subrayado (underline) o normal (none).
- **text-transform**. Transforma el texto en mayúsculas (uppercase) o en minúsculas (lowercase).

FUNDAMENTOS DE HTML Y JAVASCRIPT

4.3 Bordos y contorno

Definen distintos aspectos sobre el contorno de la etiqueta. Destacamos:

- **border-style**. Estilo del borde de la etiqueta. Por defecto, una etiqueta no muestra su contorno (*none*), aunque se puede hacer que aparezca un rectángulo que define los bordes de la etiqueta indicando los valores *solid*, *double* (borde doble) o *dotted* (borde punteado) en esta propiedad.
- **border-width**. Ancho del borde de la etiqueta. Se definen valores numéricos seguidos de la unidad de medida (por ejemplo, `border-width:20px`).
- **border-color**. Color del borde de la etiqueta. Admite los mismos valores que el atributo `color`.
- **margin**. Margen entre los bordes de la etiqueta y sus alrededores. Se definen valores numéricos seguidos de la unidad de medida.
- **padding**. Indica la distancia entre los bordes de la etiqueta y el contenido de la misma. Se definen valores numéricos seguidos de la unidad de medida.

4.4 Posición

Estas propiedades de estilo permiten colocar las etiquetas en una posición diferente a la que le corresponde originariamente. Entre las más importantes están:

- **position**. Indica el modo de colocación de la etiqueta. Sus posibles valores son:
 - **static**. La etiqueta es fija y no puede ser desplazada
 - **absolute**. Se puede desplazar a cualquier punto de la página
 - **relative**. El desplazamiento es relativo respecto a la posición original
 - **left**. Indica la posición horizontal de la esquina superior izquierda de la etiqueta. Se define mediante un valor numérico seguido de una unidad de medida.
 - **top**. Indica la posición vertical de la esquina superior izquierda de la etiqueta. Se define mediante un valor numérico seguido de una unidad de medida.
- **width**. Ancho de la etiqueta
- **height**. Alto de la etiqueta



FUNDAMENTOS DE HTML Y JAVASCRIPT

4.5 Visibilidad

Mediante la propiedad de estilo **visibility** se puede indicar si una etiqueta será o no visible. Sus posibles valores son *visible* y *hidden*. Esta propiedad de estilo puede ser alterada mediante código de JavaScript para ocultar/mostrar dinámicamente ciertas partes de la página.



5. Estilos genéricos

En los ejemplos que hemos visto para los tres tipos de hojas de estilo existentes, las propiedades de estilo se aplicaban sobre un determinado tipo de etiqueta, de modo que estas propiedades se aplicarán de manera automática cada vez que la etiqueta es utilizada.

También es posible definir un conjunto de propiedades de estilo con sus correspondientes valores, sin necesidad de asociarlas a una etiqueta concreta. A este conjunto de propiedades se le conoce como estilo genérico por no estar aplicadas sobre una etiqueta particular.

Los estilos genéricos se definen:

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

Donde “nombre_estilo” representa un nombre cualquiera que se asigna al estilo genérico.

Los estilos genéricos pueden estar definidos en una hoja de estilo externa o interna y podrán ser aplicados a cualquier etiqueta del documento en el que se utilice la hoja de estilos.

Para aplicar un estilo genérico sobre una determinada etiqueta se utilizará el atributo *class* de dicha etiqueta, asignando como valor de este atributo el nombre del estilo que queremos aplicar.

Por ejemplo, supongamos que tenemos los siguientes estilos genéricos definidos sobre una hoja de estilos:

- maximo{font-size:40pt;font-weight:900}
- medio{font-size:20pt;font-weight:500}
- maximo{font-size:10pt;font-weight:normal}

En un documento HTML aplicamos los estilos genéricos anteriores en tres párrafos del mismo, tal y como se indica en el siguiente listado:

```
<html>
  <head>
    <style>
      .maximo{font-size:40pt;font-weight:900}
      .medio{font-size:20pt;font-weight:500}
      .minimo{font-size:10pt;font-weight:normal}
    </style>
  </head>
```



FUNDAMENTOS DE HTML Y JAVASCRIPT

```
<body>
<p class="maximo">
    parrafo estilo máximo
</p>
<p class="medio">
    parrafo estilo medio
</p>
<p class="minimo">
    parrafo estilo minimo
</p>
</body>
</html>
```

Al cargar la página anterior en el navegador se mostrará una página con el aspecto de la figura 4

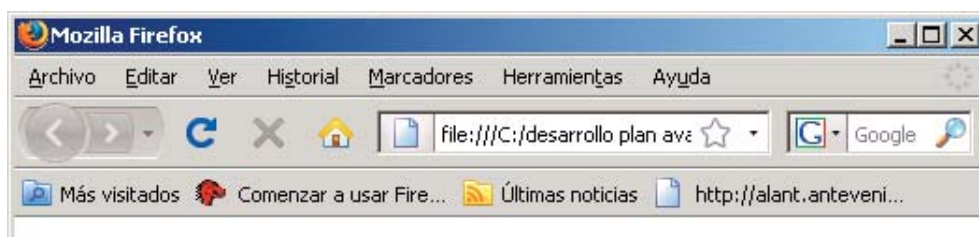


FIGURA 4

Aunque en el ejemplo anterior hemos aplicado los estilos genéricos a un mismo tipo de etiqueta, un estilo genérico puede ser aplicado a cualquier tipo de etiqueta, incluso un mismo estilo puede ser aplicado en distintas partes del documento a diferentes etiquetas.

Los estilos genéricos pues proporcionan una gran potencia y flexibilidad a la hora de crear hojas de estilos CSS para un sitio Web.

Se puede restringir el alcance de los estilos genéricos a determinados tipos de etiqueta. Para ello, a la hora de definir el estilo genérico se indicará delante del nombre del estilo el nombre de la etiqueta sobre la que se aplicará.

El siguiente ejemplo aplica la regla CSS solamente a los elementos de tipo <div> cuyo atributo class tenga el valor "box":

```
p.box {
    padding: 2px;
    border-style: solid;
    border-width: 1px;
}
```



6. Selectores id

Con CSS también permite es posible aplicar determinados estilos a las etiquetas en función de su atributo id. La sintaxis es similar a la utilizada con los estilos genéricos, salvo que en este caso se utiliza el símbolo de la almohadilla (#) en vez del símbolo del punto (.).

En el siguiente ejemplo se define un estilo de tipo selector id llamado “principal”, que aplicara las propiedades de estilo definidas en el mismo a la etiqueta H1 cuyo atributo id es “principal”:

```
<style>
#principal { font-size: 35pt; color: red; }
</style>
<body>
<h1 id="principal">Titulo principal</h1>
<p>Texto del párrafo</p>
<h1>Titulo normal</h1>
</body>
```

7. Herencia de estilos

Una de las propiedades menos conocidas de las hojas de estilos en cascada es la capacidad de heredar estilos de clases de jerarquía superior.

La herencia de estilos proporciona numerosas ventajas en el diseño de una Web; entre ellas cabe destacar:

- Simplifica la edición de las hojas de estilos, ya que la modificación de los padres es trasladada automáticamente (en cascada) a los hijos.
- Evita la redundancia y la cantidad de código necesaria, puesto que una regla sólo necesita ser declarada una vez para múltiples nodos del documento. Sólo las excepciones necesitan ser redeclaradas.
- Cuando la estructura de un documento puede ser ambigua o incluso desconocida (como al formatear un documento xml), la herencia facilita controlar su posible apariencia.

Esta jerarquía viene definida por el anidamiento de elementos dentro del documento, es decir por la posición de unos elementos dentro de otros y todos ellos contenidos por el elemento <body>.

No todas las propiedades son heredables, de hecho, solamente aquellas que no modifican el fondo del documento (tipo, tamaño y color de fuente, por ejemplo) son heredables; en cambio, otras como el color del fondo no lo son.

Por ejemplo, supongamos que tenemos definida la siguiente hoja de estilos:

```
<style>
.padre {
  border-style:solid;
  border-width:2px
  font-family: Arial;
  font-size: 20px;
  font-weight: bold;
}
.hijo {
  color: green;
}
</style>
```



FUNDAMENTOS DE HTML Y JAVASCRIPT

Por otro lado, supongamos que tenemos los siguientes elementos anidados:

```
<body class = "padre">  
  texto del padre  
  <div class = "hijo">  
    texto del hijo  
  </div>  
</body>
```

Como el elemento `<div>` se encuentra anidado en `<body>`, adquirirá automáticamente todas las propiedades de estilo definidas en el elemento superior, añadiendo las suyas propias a través del estilo definido en "hijo".

Mediante la herencia también pueden sobrescribirse alguna de las propiedades de estilo heredadas. Por ejemplo, si en el estilo "hijo" incluimos la propiedad `font-size:10px`, el texto contenido en el `<div>` tendrá un tamaño de letra de 10px, manteniendo los valores de las restantes propiedades heredadas.