

TEMA 3. LENGUAJE DE HOJAS DE ESTILO

CSS

1. Introducción a CSS

Cuando trabajamos con páginas web escritas en HTML y XHTML, es deseable que presenten un estilo concreto y un aspecto definido. La forma más eficiente para poder llevar esto a cabo es distinguir entre la parte en la que se sitúan los contenidos, los ficheros programados en HTML, y la parte del proyecto donde se recogen los aspectos relativos a la apariencia, aquí es donde aparece CSS (Cascading Style Sheets), un lenguaje de hojas de estilo que será el encargado de controlar el aspecto de los sitios web.

El uso de hojas de estilo CSS aportar numerosas ventajas: mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento, tanto si va a ser realizado por nosotros como por terceros, y además facilita la visualización del sitio en dispositivos diferentes.

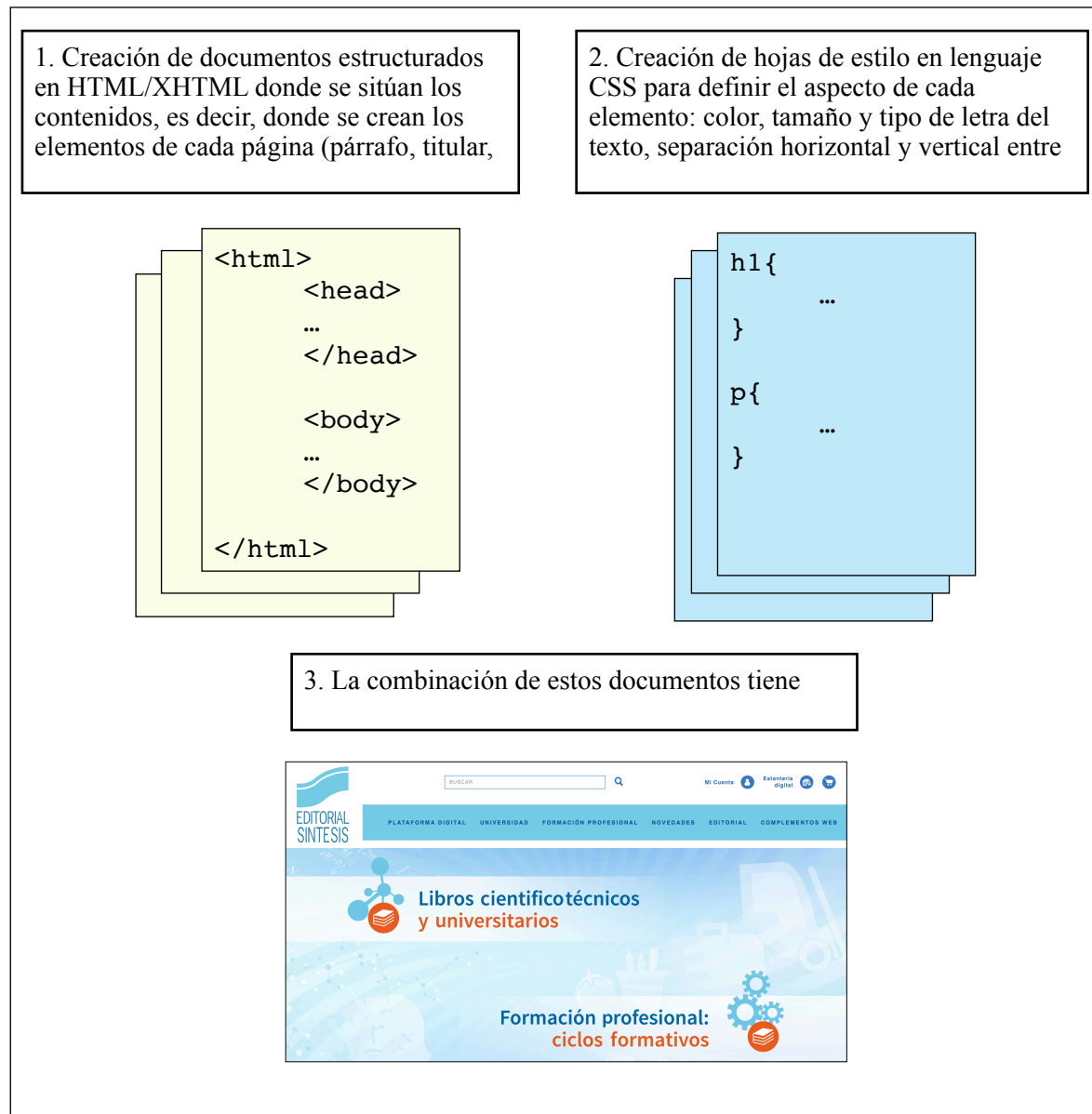


Figura 4.1 Diagrama del proceso seguido para la conexión en HTML y las hojas de estilo CSS

Para añadir estilos a una página web bastaría con añadirlos en el código de la propia página web, en el sitio indicado para ello en la cabecera, y utilizando las etiquetas **<style>**. Esto puede utilizarse para comenzar a familiarizarnos con las hojas de estilo, pero se recomienda realizarlo en documentos aparte, como se explica en el siguiente apartado.

IMPORTANTE. El principal inconveniente de **no** utilizar los documentos CSS y hacerlo desde el propio código en HTML, es que si se quiere hacer una modificación en los estilos definidos, es necesario modificar todas las páginas que incluyen el estilo que se va a modificar.

El código HTML:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <title>
      Ejemplo de estilos CSS en el propio documento
    </title>

    <style type="text/css">
      p {
        color: black; font-family: Verdana;
      }
    </style>
  </head>
  <body>
    <p>Un párrafo de texto.</p>
  </body>
</html>
```

2. Hojas de estilo externas

En la actualidad se utilizan hojas de estilo externas, en este caso, en lugar de tener el código encargado de dar modificar el sitio web, este código se encuentra en otro fichero. Desde el fichero principal de la página desarrollada en HTML se accede al fichero en código CSS, a través de la etiqueta **<link>**, la única particularidad que debemos tener en cuenta, es que el fichero donde se definen los estilos tendrá la extensión “.css”. Se pueden enlazar tantos ficheros CSS como se deseen.

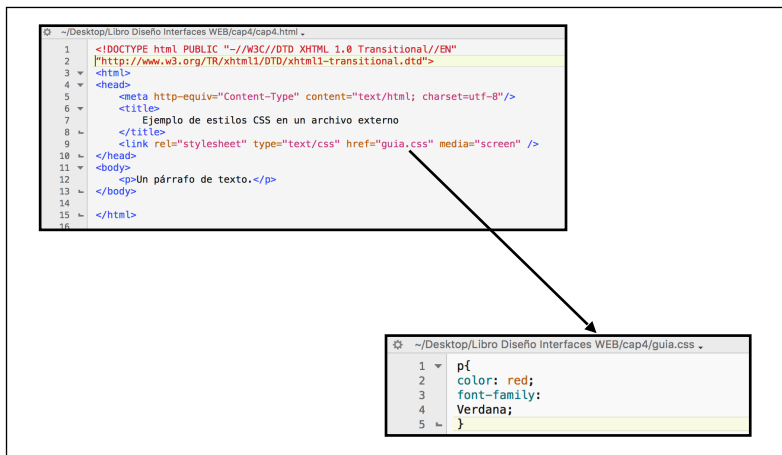


Figura 4.2 Ficheros HTML y CSS en invocación de hojas de estilo externas

La función de cada uno de los atributos usados para enlazar el documento HTML con la hoja de estilos en CSS es:

- **rel.** Indica el tipo de relación que existe entre el recurso enlazado, el archivo CSS y la página HTML. Para los archivos CSS, el valor es *stylesheet*
- **type.** Indica el tipo de recurso enlazado, en el caso de los archivos CSS su valor es *text/css*.
- **href.** Recoge la URL del archivo CSS, esta puede ser relativa o absoluta y puede apuntar a un recurso interno o externo al sitio web.
- **media.** Indica el tipo medio en el que se van a aplicar los estilos del archivo CSS. Más adelante se explican en detalle los medios CSS y su funcionamiento.

En la actualidad, el uso de CSS en HTML está muy extendido, puesto a través de un solo archivo CSS es posible incluir una misma guía de estilo en todas las páginas HTML de un mismo sitio web, de esta forma se cumple uno de los principios más importantes perseguidos a la hora de realizar un correo diseño gráfico, que exista **uniformidad de estilo entre todas las páginas que componen el sitio**. Además, si es necesario realizar un cambio común en el estilo de todas las páginas, basta con hacerlo en la hoja de estilo y se extenderá por todo el sitio web enlazado a este archivo CSS.

El código HTML:

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>
      Ejemplo de estilos CSS en el propio documento
    </title>
    <link rel="stylesheet" type="text/css" media="screen, handheld" href="hojaEst.css" >
  </head>
  <body>
    <p>Un párrafo de texto.</p>
  </body>
</html>
```

El código CSS (hojaEst.css):

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

3. Medios

Las hojas de estilo permiten especificar como se representa un documento, en función del tipo de medio de visualización. Entendemos por **medio**, como los canales a través de los cuales se muestra la información del sitio web: canales auditivos, visuales, impresos, etc. Lo habitual es que utilicemos la propiedad definida para cada tipo de canal. En la actualidad el medio más común es el utilizado para definir el aspecto del sitio web en pantalla, **screen**, y el que define la página para ser impresa, **print**. En la siguiente tabla se recogen los tipos más frecuentes.

Medio	Descripción
all	Todos los medios definidos
print	Impresoras
braille	Dispositivos táctiles que emplean el sistema braille
embosed	Impresoras braille
handheld	Dispositivos de mano: móviles, PDA, etc.
projection	Proyectores y dispositivos para presentaciones
screen	Pantallas de ordenador
speech	Sintetizadores para navegadores de voz utilizados por personas discapacitadas
tv	Televisores y dispositivos con resolución baja

Cuadro 4.2 Tipos de medios, canales a través de los cuales es posible visualizar el sitio web

Para especificar el medio de destino, se utilizan varios mecanismos, desde la hoja de estilo en CSS o desde el propio fichero HTML del sitio web.

4.3.1. @media

La regla **@media** especifica los tipos de medios de destino, a través de esta opción es posible que en la misma hoja estilo aparezcan reglas para distintos medios. Si un mismo estilo se aplica a varios medios basta con indicar el nombre de estos separados por comas.

```
@media nombreTipoMedio{
  selector{
    propiedad: valor;
  }
}
```

En el siguiente diagrama se muestra un ejemplo de uso de `@media` para indicar el tipo de medio donde se va a visualizar el sitio web. En este caso, se observan dos tipos de medio, impresión y visualización por pantalla. Para el tamaño de la letra se especifica para cada medio por separado, mientras que para el interlineado se va a utilizar el mismo en ambos casos, por lo tanto, se puede indicar de forma conjunta.

```
@media print {  
    body {  
        font-size: 10pt  
    }  
}  
@media screen {  
    body {  
        font-size: 13px  
    }  
}  
@media screen, print {  
    body {  
        line-height: 1.2  
    }  
}
```

4.3.2. @import

A través **@import** es posible importar nuevas reglas desde otros ficheros CSS, e indicar el medio o medios sobre los que debe aplicarse el nuevo estilo. En este caso, a diferencia de lo que ocurría con **@media**, en primer lugar aparece la url que apunta al fichero CSS que contiene la guía de estilo, y a continuación se indica el medio sobre el que se va a aplicar. Es importante tener en cuenta, que estas reglas deben preceder a todas las demás reglas.

Si los datos de estilo del fichero se van de aplicar a varios medios al mismo tiempo, se indica el nombre de estos separados por comas.

@import url(nombreFichero.css) nombreTipoMedio;

```
@import "ficheroEstilo1.css" print;  
@import "ficheroEstilo2.css" screen;  
@import "ficheroEstilo3.css" print, screen;  
@import "ficheroEstilo4.css"; //Por defecto se aplica a todo
```

En el ejemplo anterior las dos primeras reglas aplicarán los estilos contenidos en los ficheros “ficheroEstilo1.css” y “ficheroEstilo2.css”, en medio impreso y por pantalla, respectivamente. La

tercera regla aplicará el estilo de “ficheroEstilo3.css” en ambos medios. Finalmente, la última regla al no indicar ningún medio se aplica en todos (all).

4. El modelo de cajas de CSS

El modelo de cajas es una de las características más importantes de las hojas de estilos. Cuando se crea una página en HTML, el uso de cada etiqueta simula lo que podríamos considerar como una caja, dotando a cada una de ella de un identificador, indicado por el nombre de la etiqueta. Este sistema de cajas permite desde el fichero CSS seleccionar cada caja y dotarle del estilo deseado.

El uso de las guías de estilo con CSS permite modificar todas las características de cada una de las cajas creadas, analizamos a continuación cada una de estas partes a través de un ejemplo que irá incorporando nuevas propiedades.

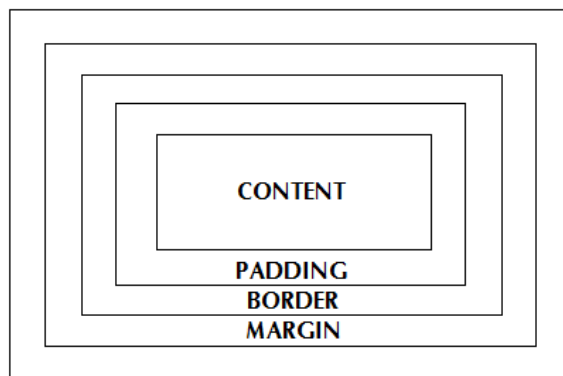


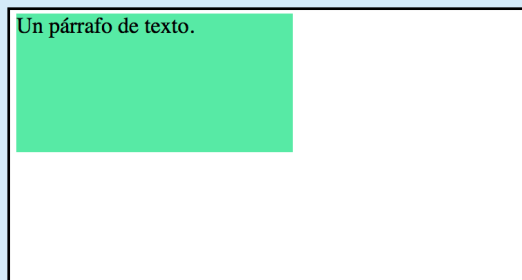
Figura 4.3 Diagrama modelo de cajas en CSS

Las propiedades principales con las que se trabaja en el modelo de cajas son **width** y **height**, relativas a las dimensiones de la **caja de contenido o content**, el ancho y el alto, se trata del área donde se muestra el contenido de la caja.

Otra de las propiedades más importantes del área de contenido es el fondo de la caja, diferenciamos entre **background image** (imagen de fondo), se trata de la imagen que se muestra por detrás del contenido y el espacio de relleno y **background color** (color de fondo) color que se muestra por detrás del contenido y el espacio de relleno.

```
article {
    background: #5DEAAB;
    height: 100px;
    width: 200px;
}
```

Se verá:



La segunda área es la de relleno, llamada **padding**, esto es el espacio que existe entre el la caja de contenido y el borde. Las características de esta caja que pueden modificarse son su altura y anchura, se distinguen por tanto, cuatro zonas posibles de acción: zona inferior, zona superior, zona izquierda y zona derecha. El nombre de las propiedades que definen estas zonas son: padding-top, padding-right, padding-left y padding-bottom.

```
article {  
  padding-top: 10px;  
  padding-right: 30px;  
  padding-bottom: 10px;  
  padding-left: 20px;  
}
```

Si solo escribimos la palabra padding, se aplicará el mismo formato a todos los extremos. En el siguiente ejemplo los límites inferior, superior, izquierdo y derecho tendrían un valor de 10px.

```
article {  
  padding: 10px;  
}
```

Existen más tipos de empleo de esta regla que simplifican su uso, se basan en el empleo de la palabra **padding** seguida de varias cifras separadas sin comas. Si aparecen dos valores, el primero es asignado al límite superior y al inferior, mientras que la segunda cifra será la que se aplique al relleno horizontal. En el caso de tener tres valores, el primero y el último serán los aplicados a los límites superior e inferior respectivamente, y la segunda cifra indica el valor de los bordes izquierdo y derecho.

ACTIVIDAD. Realiza un fragmento de código en el que se ponga en uso todos los métodos de padding vistos en este apartado.

La siguiente área que nos encontramos en el diagrama de cajas, es el correspondiente al borde, llamado **border**, esta zona es la que encierra el contenido y el relleno, *content* y *padding*. Es posible modificar su grosor, estilo y color, además al igual que ocurre con padding, es posible dar formato a los cuatro extremos de la caja a la vez utilizando solo la palabra border o hacerlo de forma individual.

```
article {  
  border: 1px solid #000;  
}
```

Pero para aquellas ocasiones en las que se desee dar un formato diferente a cada uno de los extremos, se utilizarán algunas de las propiedades individuales destinadas a tal fin.

border-top border-right border-bottom border-left	Modifican a la vez el grosor, estilo y color de cada lado del borde
border-width border-style border-color	Modificar de forma individual el grosor, el estilo y el color, de los cuatros extremos a la vez
border-top-width	Modifica el grosor del borde superior
border-top-style	Modifica el estilo del borde superior
border-top-color	Modifica el color del borde superior

Finalmente, el margen que envuelve al resto de elementos CSS, **margin**, y sostiene a otras cajas del diseño. Las propiedades individuales son margin-top, margin-right, margin-bottom, y margin-left. El funcionamiento es el mismo que el de la propiedad padding.

```
article {
  margin-top: 10px;
  margin-right: 10px;
  margin-bottom: 10px;
  margin-left: 10px;
}
```

Cuadro 4.3 Tipologías de la propiedad border en el modelo de cajas

5. Herencia

La herencia en CSS es el mecanismo mediante el cual determinadas propiedades de un elemento padre se transmiten a sus hijos. No todas las propiedades CSS son heredadas, los márgenes no se heredan porque es poco probable que un elemento hijo necesite los mismos márgenes que su padre. Cuando se establece el valor de una propiedad CSS en un elemento, sus elementos descendientes heredan de forma automática el valor de esa propiedad.

```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
    <title>Ejemplo de herencia de estilos</title>
    <style type="text/css">
      body {color: blue; }
    </style>
  </head>

  <body>
    <h1>Titular de la página</h1>
    <p>Un párrafo de texto no muy largo.</p>
  </body>
</html>
```

En el ejemplo, el selector `body` solo establece el color de la letra para el elemento `<body>`. Esta propiedad se hereda de forma automática, ¿qué significa esto? que los elementos descendientes muestran ese mismo color. Ahora bien, la herencia de estilos se aplica automáticamente, pero su efecto puede ser anulado estableciendo de forma explícita otro valor para la propiedad que se hereda.

En cambio, existen otras propiedades cuyo valor no se hereda, como es el caso de los bordes (`border`), es decir, si se define un estilo relativo al borde para el cuerpo, este no será aplicado al resto de elementos HTML. En la tabla se resumen algunas de las propiedades que sí son heredadas por defecto en los elementos HTML.

color	text-align
font	visibility
letter-spacing	white-space
direction	border-collapse
word-spacing	quotes
line-height	list-style-image

Cuadro 4.4 Resumen propiedades heredadas por defecto en HTML

Ahora bien, como hemos dicho existen ciertas propiedades que no son heredadas de padres a hijos, pero existe un mecanismo mediante el cual es posible indicar que un elemento herede el valor de una propiedad. Para ello se utilizan los valores: inherit, initial y unset que modifican el valor de la propiedad respecto a la herencia.

- inherit. Aplica el valor de la propiedad del elemento padre
- initial. Aplica el valor que presenta inicialmente la propiedad
- unset. Hereda el valor de la propiedad del elemento padre, si este no existe, aplica el valor inicial

```
body {  
    border-width: 10px;  
    border-color: blue;  
    color:red;  
}  
  
h1 {  
    border:inherit;  
}
```

ACTIVIDAD 1. Reescribir el código anterior para que el elemento h1 sea de color verde, el resto de elementos heredará su valor de body.

En este ejemplo, el elemento h1 hereda todas las propiedades establecidas en el body. El color se hace forma automática, puesto es una de las propiedades que se heredan, y al no establecerse otro valor para el color en h1 quedará de color rojo. Ahora bien, en este ejemplo las propiedades de borde también serán heredadas puesto que así es indicado por el valor inherit.

6. CSS3

CSS3 (Cascading Style Sheets 3) consiste en la última versión del lenguaje de hojas de estilo CSS, en este apartado se verán algunas de las novedades que presenta frente a su versión anterior, CSS2.1. Esta nueva actualización presenta todas las características de su predecesora y además aporta un mayor dinamismo en el diseño de la página web. En cuanto a los aspectos relativos a la programación, esta no varía, se sigue empleando la estructura de propiedades y selectores vista a lo largo del capítulo.

- **Múltiples imágenes en el fondo.** En esta nueva versión es posible poner varias imágenes de fondo en un mismo elemento. En el siguiente ejemplo, se puede ver como se utilizan varias imágenes para el fondo y a continuación, de l url de cada una de ellas indica las podridas de background.

```
#contenidoPrincipal {
    width: 500px; height: 500px;
    background:
        url(imagen1.gif) bottom right no-repeat,
        url(imagen2.jpg) center center no-repeat,
        url(imagen3.jpg) repeat;
}
```

- **Nuevas propiedades para el fondo de los elementos.** Estos son: background-origin, background-clip, background-size.
- **Esquinas redondeadas.** Se añade una propiedad que permite redondear las esquinas de las áreas de la página, en concreto, a través del atributo **border-radius**. Cabe tener en cuenta, que en función del navegador el nombre del atributo varía: para Chrome y Safari se utiliza **-webkit-border-radius** y para Mozilla **-moz-border-radius**.

```
div {
    border: 2px solid #ffffff;
    -moz-border-radius: 9px;
    -webkit-border-radius: 9px;
}
```

- **Sombras.** Se incorpora una propiedad para crear sombras en los elementos, incluso en los textos. El atributo que se utiliza es **box-shadow** y está formado por cuatro valores: desplazamiento horizontal de la sombra, desplazamiento vertical de la sombra, difuminado y color de la sombra, estos valores aparecen siempre en el mismo orden.

```
div {
    box-shadow: 5px -9px 3px #000;
}
```

- **Transparencias de color.** Otra de las nuevas incorporaciones en CSS3 es la posibilidad de añadir transparencia en el color, desde un color totalmente sólido hasta el transparente. La propiedad que define la transparencia se denomina opacity, que toma un valor decimal entre 0.0 y 1.0, donde 0.0 es la máxima transparencia y 1.0 la máxima opacidad.

```
div {
    opacity: 0.5;
}
```

- **Texto en varias columnas.** En CSS3 se incorpora la opción de maquetar el texto en varias columnas, a través de varios atributos: **column-width** (define el ancho de las columnas), **column-gap** (define el espacio en blanco entre columnas) y **column-rule** (crea una línea entre las columnas). Aunque en la actualidad aún no son soportadas por ningún navegador, de manera experimental sí son implementadas por Safari, Google Chrome y Firefox. Al igual que en el caso de las de border-radius, se utiliza un prefijo para su uso en función del navegador en el que se usan, "-moz-" en Firefox y "-webkit-" en Safari y Chrome.

```
div{  
    -moz-column-width: 20em;  
    -moz-column-gap: 10px;  
    -webkit-column-width: 20em;  
    -webkit-column-gap: 10px;  
    -webkit-column-rule: 2px solid #ffffff;  
    -moz-column-rule: 2px solid #ffffff;
```

- **Animaciones.** Unas de las novedades más importantes es la creación de animaciones, estas se crean directamente desde CSS3 sin la necesidad de utilizar otros lenguajes, además con este tipo de implementación ya no será necesario tener instaladores plugins que antes eran inevitables para la correcta visualización de las animaciones.

ACTIVIDAD. Diseña y escribe un programa en HTML y CSS3 que tenga tres columnas del mismo tamaño. Además, el color de fondo de la página debe ser azul con una transparencia del 30%.