# Capítulo 1. Introducción

## 1.1. ¿Qué es CSS?

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. CSS es la mejor forma de separar los contenidos y su presentación y es imprescindible para crear páginas web complejas.

Separar la definición de los contenidosy la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados *"documentos semánticos"*). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes.

Al crear una página web, se utiliza en primer lugar el lenguaje HTML/XHTML para *marcar* los contenidos, es decir, para designar la función de cada elemento dentro de la página: párrafo, titular, texto destacado, tabla, lista de elementos, etc.

Una vez creados los contenidos, se utiliza el lenguaje CSS para definir el aspecto de cada elemento: color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.

## 1.2. Breve historia de CSS

Las hojas de estilos aparecieron poco después que el lenguaje de etiquetas SGML, alrededor del año 1970. Desde la creación de SGML, se observó la necesidad de definir un mecanismo que permitiera aplicar de forma consistente diferentes estilos a los documentos electrónicos.

El gran impulso de los lenguajes de hojas de estilos se produjo con el boom de Internet y el crecimiento exponencial del lenguaje HTML para la creación de documentos electrónicos. La guerra de navegadores y la falta de un estándar para la definición de los estilos dificultaban la creación de documentos con la misma apariencia en diferentes navegadores.

El organismo W3C (World Wide Web Consortium), encargado de crear todos los estándares relacionados con la web, propuso la creación de un lenguaje de hojas de estilos específico para el lenguaje HTML y se presentaron nueve propuestas. Las dos propuestas que se tuvieron en cuenta fueron la CHSS (*CascadingHTML Style Sheets*) y la SSP (*Stream-based Style SheetProposal*).

La propuesta CHSS fue realizada por HåkonWium Lie y SSP fue propuesto por BertBos. Entre finales de 1994 y 1995 Lie y Bos se unieron para definir un nuevo lenguaje que tomaba lo mejor de cada propuesta y lo llamaron CSS (*Cascading Style Sheets*).

En 1995, el W3C decidió apostar por el desarrollo y estandarización de CSS y lo añadió a su grupo de trabajo de HTML. A finales de 1996, el W3C publicó la primera recomendación oficial, conocida como "CSS nivel 1".

A principios de 1997, el W3C decide separar los trabajos del grupo de HTML en tres secciones: el grupo de trabajo de HTML, el grupo de trabajo de DOM y el grupo de trabajo de CSS.

El 12 de Mayo de 1998, el grupo de trabajo de CSS publica su segunda recomendación oficial, conocida como "CSS nivel 2". La versión de CSS que utilizan todos los navegadores de hoy en día es CSS 2.1, una revisión de CSS 2 que aún se está elaborando (la última actualización es del 23 de abril de 2009). Al mismo tiempo, la siguiente recomendación de CSS, conocida como "CSS nivel 3", continúa en desarrollodesde 1998 y hasta el momento sólo se han publicado borradores.

La adopción de CSS por parte de los navegadores ha requerido un largo periodo de tiempo. El mismo año que se publicó CSS 1, Microsoft lanzaba su navegador Internet Explorer 3.0, que disponía de un soporte bastante reducido de CSS. El primer navegador con soporte completo de CSS 1 fue la versión para Mac de Internet Explorer 5, que se publicó en el año 2000. Por el momento, ningún navegador tiene soporte completo de CSS 2.1.

## 1.3. Soporte de CSS en los navegadores

El trabajo del diseñador web siempre está limitado por las posibilidades de los navegadores que utilizan los usuarios para acceder a sus páginas. Por este motivo es imprescindible conocer el soporte de CSS en cada uno de los navegadores más utilizados del mercado.Internamente los navegadores están divididos en varios componentes. La parte del navegador que se encarga de interpretar el código HTML y CSS para mostrar las páginas se denomina motor. Desde el punto de vista del diseñador CSS, la versión de un motor es mucho más importante que la versión del propio navegador.La siguiente tabla muestra el soporte de CSS 1, CSS 2.1 y CSS 3 de los cinco navegadores más utilizados por los usuarios:

| **Navegador** | **Motor** | **CSS 1** | **CSS 2.1** | **CSS 3** |
|---|---|---|---|---|
| Mcsoft. edge | Trident | Completo | Completo | Todos los selectores, pseudo-clases y muchas propiedades |
| Firefox | Gecko | Completo | completo | Selectores, pseudo-clases y algunas propiedades |
| Safari | WebKit | Completo | completo | Todos los selectores, pseudo-clases y muchas propiedades |
| Opera | Presto | Completo | completo | Todos los selectores, pseudo-clases y muchas propiedades |
| Google Chrome | WebKit | Completo | completo | Todos los selectores, pseudo-clases y muchas propiedades |

Los navegadores Safari y Opera son los más avanzados en el soporte de CSS, ya que incluyen muchos elementos de la futura versión CSS 3 y un soporte casi perfecto de la actual versión 2.1. El navegador Firefox no tiene un soporte tan avanzado de CSS 3 pero las últimas versiones están alcanzando rápidamente a Safari y Opera.

Por su parte, el navegador Internet Explorer sólo puede considerarse adecuado desde el punto de vista de CSS a partir de su versión 7. Internet Explorer 6, utilizado todavía por un número significativo de usuarios, sufre carencias muy importantes y contiene decenas de errores en su soporte de CSS. Internet Explorer 8 incluye el soporte completo de todas las propiedades y características de CSS 2.1.

La tabla anterior ha sido elaborada a partir de la información que se puede encontrar en la página *Comparison of layoutengines* de la Wikipedia, donde se muestra una comparación exhaustiva sobre el soporte de todas las características de CSS por parte de cada navegador.

## 1.4. Especificación oficial

La especificación o norma oficial que se utiliza actualmente para diseñar páginas web con CSS es la versión CSS 2.1, actualizada por última vez el 23 de abril de 2009 y que se puede consultar libremente en http://www.w3.org/TR/CSS21/

Desde hace varios años, el organismo W3C trabaja en la elaboración de la próxima versión de CSS, conocida como CSS 3. Esta nueva versión incluye multitud de cambios importantes en todos los niveles y es mucho más avanzada y compleja que CSS 2.

No obstante, pasarán muchos años hasta que se publique la versión definitiva completa de CSS 3 y hasta que los principales navegadores del mercado incluyan la mayor parte del nuevo estándar.

El sitio web del organismo W3C dispone de una sección en la que se detalla el trabajo que el W3C está desarrollando actualmente en relación a CSS y también dispone de un blog en el que se publican todas las novedades relacionadas con CSS.

## 1.5. Funcionamiento básico de CSS

Antes de que se generalizara el uso de CSS, los diseñadores de páginas web utilizaban etiquetas HTML especiales para modificar el aspecto de los elementos de la página. El siguiente ejemplo muestra una página HTML con estilos definidos sin utilizar CSS:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo de estilos sin CSS</title>
</head>
```

```
<body>
<h1><font color="red" face="Arial" size="5">Titular de la página</font></h1>
<p><font color="gray" face="Verdana" size="2">Un párrafo de texto no muy largo.</font></p>
</body>
</html>
```

El ejemplo anterior utiliza la etiqueta <font> con sus atributos color, face y size para definir el color, el tipo y el tamaño de letra de cada elemento de la página.

El problema de utilizar este método para definir el aspecto de los elementos se puede ver claramente con el siguiente ejemplo: si la página tuviera 50 elementos diferentes, habría que insertar 50 etiquetas <font>. Si el sitio web entero se compone de 10.000 páginas diferentes, habría que definir 500.000 etiquetas <font>. Como cada etiqueta <font> tiene tres atributos, habría que definir 1.5 millones de atributos.

Como el diseño de los sitios web está en constante evolución, es habitual modificar cada cierto tiempo el aspecto de las páginas del sitio. Siguiendo con el ejemplo anterior, cambiar el aspecto del sitio requeriría modificar 500.000 etiquetas y 1.5 millones de atributos.

La solución que propone CSS es mucho mejor, como se puede ver en el siguiente ejemplo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo de estilos con CSS</title>
<style type="text/css">
  h1 { color: red;  font-family: Arial;   font-size: large;  }
p{ color: gray; font-family: Verdana; font-size: medium; }
</style>
</head>

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

CSS permite separar los contenidos de la página y la información sobre su aspecto. En el ejemplo anterior, dentro de la propia página HTML se crea una zona especial en la que se incluye toda la información relacionada con los estilos de la página.

Utilizando CSS, se pueden establecer los mismos estilos con menos esfuerzo y sin *ensuciar* el código HTML de los contenidos con etiquetas <font>. Como se verá más adelante, la etiqueta <style> crea una zona especial donde se incluyen todas las reglas CSS que se aplican en la página.

En el ejemplo anterior, dentro de la zona de CSS se indica que todas las etiquetas <h1> de la página se deben ver de color rojo, con un tipo de letra Arial y con un tamaño de letra grande. Además, las etiquetas <p> de la página se deben ver de color gris, con un tipo de letra Verdana y con un tamaño de letra medio.

Definir los estilos de esta forma ahorra miles de etiquetas y millones de atributos respecto a la solución anterior, pero sigue sin ser una solución ideal. Como los estilos CSS sólo se aplican en la página que los incluye, si queremos que las 10.000 páginas diferentes del sitio tengan el mismo aspecto, se deberían copiar 10.000 veces esas mismas reglas CSS. Más adelante se explica la solución que propone CSS para evitar este problema.

# 1.6. Cómo incluir CSS en un documento XHTML

Una de las principales características de CSS es su flexibilidad y las diferentes opciones que ofrece para realizar una misma tarea. De hecho, existen tres opciones para incluir CSS en un documento HTML.

## 1.6.1. Incluir CSS en el mismo documento HTML

Los estilos se definen en una zona específica del propio documento HTML. Se emplea la etiqueta <style> de HTML y solamente se pueden incluir en la cabecera del documento (sólo dentro de la sección <head>).

Ejemplo:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<metahttp-equiv="Content-Type"content="text/html; charset=iso-8859-1" />
<title>Ejemplo de estilos CSS en el propio documento</title>
<styletype="text/css">
p { color: black; font-family: Verdana; }
</style>
</head>
 <body>
<p>Un párrafo de texto.</p>
</body>
</html>
```

Este método se emplea cuando se define un número pequeño de estilos o cuando se quieren incluir estilos específicos en una determinada página HTML que completen los estilos que se incluyen por defecto en todas las páginas del sitio web.

El principal inconveniente 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.

### 1.6.2. Definir CSS en un archivo externo

En este caso, todos los estilos CSS se incluyen en un archivo de tipo CSS que las páginas HTML enlazan mediante la etiqueta <link>. Un archivo de tipo CSS no es más que un archivo simple de texto cuya extensión es .css Se pueden crear todos los archivos CSS que sean necesarios y cada página HTML puede enlazar tantos archivos CSS como necesite.

Si se quieren incluir los estilos del ejemplo anterior en un archivo CSS externo, se deben seguir los siguientes pasos:

1) Se crea un archivo de texto y se le añade solamente el siguiente contenido:

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

2) Se guarda el archivo de texto con el nombre estilos.css Se debe poner especial atención a que el archivo tenga extensión .css y no .txt

3) En la página HTML se enlaza el archivo CSS externo mediante la etiqueta <link>:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<htmlxmlns="http://www.w3.org/1999/xhtml">
<head>
<metahttp-equiv="Content-Type"content="text/html; charset=iso-8859-1" />
<title>Ejemplo de estilos CSS en un archivo externo</title>
<linkrel="stylesheet"type="text/css"href="/css/estilos.css"media="screen" />
</head>
 <body>
<p>Un párrafo de texto.</p>
</body>
</html>
```

Para indicar que una hoja de estilos CSS es alternativa, se utiliza el atributo rel="alternate stylesheet" y se establece el atributo title en la etiqueta <link>:

```
<linkrel="alternate stylesheet"title="Estilo alternativo"type="text/css"href="css/estilos.css"/>
```

La hoja de estilos del ejemplo anterior no se aplica en la página a menos que el usuario la seleccione entre todas las hojas de estilos alternativas. Esta característica no está disponible en todos los navegadores, por lo que los diseñadores no deben suponer que el usuario podrá utilizarla.

Cuando el navegador carga la página HTML anterior, antes de mostrar sus contenidos también descarga los archivos CSS externos enlazados mediante la etiqueta <link> y aplica los estilos a los contenidos de la página.

Normalmente, la etiqueta <link> incluye cuatro atributos cuando enlaza un archivo CSS:

- **rel**: indica el tipo de relación que existe entre el recurso enlazado (en este caso, el archivo CSS) y la página HTML. Para los archivos CSS, siempre se utiliza el valor stylesheet
- **type**: indica el tipo de recurso enlazado. Sus valores están estandarizados y para los archivos CSS su valor siempre es text/css
- **href**: indica la URL del archivo CSS que contiene los estilos. La URL indicada puede ser relativa o absoluta y puede apuntar a un recurso interno o externo al sitio web.
- **media**: indica el 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.

De todas las formas de incluir CSS en las páginas HTML, esta es la más utilizada con mucha diferencia. La principal ventaja es que se puede incluir un mismo archivo CSS en multitud de páginas HTML, por lo que se garantiza la aplicación homogénea de los mismos estilos a todas las páginas que forman un sitio web.

Con este método, el mantenimiento del sitio web se simplifica al máximo, ya que un solo cambio en un solo archivo CSS permite variar de forma instantánea los estilos de todas las páginas HTML que enlazan ese archivo.

Aunque generalmente se emplea la etiqueta <link> para enlazar los archivos CSS externos, también se puede utilizar la etiqueta <style>. La forma alternativa de incluir un archivo CSS externo se muestra a continuación:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<htmlxmlns="http://www.w3.org/1999/xhtml">
<head>
<metahttp-equiv="Content-Type"content="text/html; charset=iso-8859-1" />
<title>Ejemplo de estilos CSS en un archivo externo</title>
<styletype="text/css"media="screen">
  @import '/css/estilos.css';
</style>
</head>

<body>
<p>Un párrafo de texto.</p>
</body>
</html>
```

En este caso, para incluir en la página HTML los estilos definidos en archivos CSS externos se utiliza una regla especial de tipo @import. Las reglas de tipo @import siempre preceden a cualquier otra regla CSS (con la única excepción de la regla @charset).

La URL del archivo CSS externo se indica mediante una cadena de texto encerrada con comillas simples o dobles o mediante la palabra reservada url(). De esta forma, las siguientes reglas @import son equivalentes:

```
@import '/css/estilos.css';
@import "/css/estilos.css";
@importurl('/css/estilos.css');
@importurl("/css/estilos.css");
```

### 1.6.3. Incluir CSS en los elementos HTML

El último método para incluir estilos CSS en documentos HTMLes el peor y el menos utilizado, ya que tiene los mismos problemas que la utilización de las etiquetas <font>.

Ejemplo:

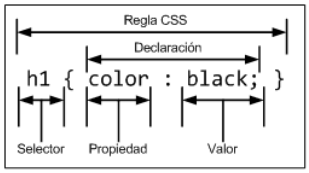
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<metahttp-equiv="Content-Type"content="text/html; charset=iso-8859-1" />
<title>Ejemplo de estilos CSS en el propio documento</title>
</head>

<body>
<pstyle="color: black; font-family: Verdana;">Un párrafo de texto.</p>
</body>
</html>
```

Esta forma de incluir CSS directamente en los elementos HTML solamente se utiliza en determinadas situaciones en las que se debe incluir un estilo muy específico para un solo elemento concreto.

## 1.7. Glosario básico

CSS define una serie de términos que permiten describir cada una de las partes que componen los estilos CSS. El siguiente esquema muestra las partes que forman un estilo CSS muy básico:



Los diferentes términos se definen a continuación:

- Regla: cada uno de los estilos que componen una hoja de estilos CSS. Cada regla está compuesta de una parte de *"selectores"*, un símbolo de *"llave de apertura"* ({), otra parte denominada *"declaración"* y por último, un símbolo de *"llave de cierre"* (}).
- Selector: indica el elemento o elementos HTML a los que se aplica la regla CSS.
- Declaración: especifica los estilos que se aplican a los elementos. Está compuesta por una o más propiedades CSS.
- Propiedad: característica que se modifica en el elemento seleccionado, como por ejemplo su tamaño de letra, su color de fondo, etc.
- Valor: establece el nuevo valor de la característica modificada en el elemento.

Un archivo CSS puede contener *infinitas* reglas CSS, cada regla puede contener *infinitos* selectores y cada declaración puede estar formada por un número *infinito* de pares propiedad/valor.

El estándar CSS 2.1 define 115 propiedades, cada una con su propia lista de valores permitidos. Por su parte, los últimos borradores del estándar CSS 3 ya incluyen 239 propiedades.

## 1.8. Medios CSS

Una de las características más importantes de las hojas de estilos CSS es que permiten definir diferentes estilos para diferentes medios o dispositivos: pantallas, impresoras, móviles, proyectores, etc.

Además, CSS define algunas propiedades específicamente para determinados medios, como por ejemplo la paginación y los saltos de página para los medios impresos o el volumen y tipo de voz para los medios de audio. La siguiente tabla muestra el nombre que CSS utiliza para identificar cada medio y su descripción:

| **Medio** | **Descripción** |
|---|---|
| all | Todos los medios definidos |
| braille | Dispositivos táctiles que emplean el sistema braille |
| embosed | Impresoras braille |
| handheld | Dispositivos de mano: móviles, PDA, etc. |
| print | Impresoras y navegadores en el modo *"Vista Previa para Imprimir"* |
| projection | Proyectores y dispositivos para presentaciones |

| | |
|---|---|
| screen | Pantallas de ordenador |
| speech | Sintetizadores para navegadores de voz utilizados por personas discapacitadas |
| tty | Dispositivos textuales limitados como teletipos y terminales de texto |
| tv | Televisores y dispositivos con resolución baja |

Los medios más utilizados actualmente son screen (para definir el aspecto de la página en pantalla) y print (para definir el aspecto de la página cuando se imprime), seguidos de handheld (que define el aspecto de la página cuando se visualiza mediante un dispositivo móvil).

Además, CSS clasifica a los medios en diferentes grupos según sus características. La siguiente tabla resume todos los grupos definidos en el estándar:

| **Medio** | **Continuo / Paginado** | **Visual / Auditivo / Táctil / Vocal** | **Mapa de bits / Caracteres** | **Interactivo / Estático** |
|---|---|---|---|---|
| braille | continuo | táctil | caracteres | ambos |
| embossed | paginado | táctil | caracteres | estático |
| handheld | ambos | visual, auditivo, vocal | ambos | ambos |
| print | paginado | visual | mapa de bits | estático |
| projection | paginado | visual | mapa de bits | interactivo |
| screen | continuo | visual, auditivo | mapa de bits | ambos |
| speech | continuo | vocal | (no tiene sentido) | ambos |
| tty | continuo | visual | caracteres | ambos |
| tv | ambos | visual, auditivo | mapa de bits | ambos |

La gran ventaja de CSS es que permite modificar los estilos de una página en función del medio en el que se visualiza. Existen cuatro formas diferentes de indicar el medio en el que se deben aplicar los estilos CSS.

### 1.8.1. Medios definidos con las reglas de tipo @media

Las reglas @media son un tipo especial de regla CSS que permiten indicar de forma directa el medio o medios en los que se aplicarán los estilos incluidos en la regla. Para

especificar el medio en el que se aplican los estilos, se incluye su nombre después de @media. Si los estilos se aplican a varios medios, se incluyen los nombres de todos los medios separados por comas.

A continuación se muestra un ejemplo sencillo:

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

El ejemplo anterior establece que el tamaño de letra de la página cuando se visualiza en una pantalla debe ser 13 píxel. Sin embargo, cuando se imprimen los contenidos de la página, su tamaño de letra debe ser de 10 puntos. Por último, tanto cuando la página se visualiza en una pantalla como cuando se imprimen sus contenidos, el interlineado del texto debe ser de 1.2 veces el tamaño de letra del texto.

### 1.8.2. Medios definidos con las reglas de tipo @import

Cuando se utilizan reglas de tipo @import para enlazar archivos CSS externos, se puede especificar el medio en el que se aplican los estilos indicando el nombre del medio después de la URL del archivo CSS:

```
@importurl("estilos_basicos.css") screen;
@importurl("estilos_impresora.css") print;
```

Las reglas del ejemplo anterior establecen que cuando la página se visualiza por pantalla, se cargan los estilos definidos en el primer archivo CSS. Por otra parte, cuando la página se imprime, se tienen en cuenta los estilos que define el segundo archivo CSS.

Si los estilos del archivo CSS externo deben aplicarse en varios medios, se indican los nombres de todos los medios separados por comas. Si no se indica el medio en una regla de tipo @import, el navegador sobreentiende que el medio es all, es decir, que los estilos se aplican en todos los medios.

### 1.8.3. Medios definidos con la etiqueta <link>

Si se utiliza la etiqueta <link> para enlazar los archivos CSS externos, se puede utilizar el atributo media para indicar el medio o medios en los que se aplican los estilos de cada archivo:

```
<linkrel="stylesheet"type="text/css"media="screen"href="basico.css" />
<linkrel="stylesheet"type="text/css"media="print, handheld"href="especial.css" />
```

En este ejemplo, el primer archivo CSS se tiene en cuenta cuando la página se visualiza en la pantalla (media="screen"). Los estilos indicados en el segundo archivo CSS, se aplican al imprimir la página (media="print") o al visualizarla en un dispositivo móvil (media="handheld"), como por ejemplo en un iPhone.

Si la etiqueta <link> no indica el medio CSS, se sobreentiende que los estilos se deben aplicar a todos los medios, por lo que es equivalente a indicar media="all".

### 1.8.4. Medios definidos mezclando varios métodos

CSS también permite mezclar los tres métodos anteriores para indicar los medios en los que se aplica cada archivo CSS externo:

```
<linkrel="stylesheet"type="text/css"media="screen"href="basico.css" />
@importurl("estilos_seccion.css") screen;
@media print {
  /* Estilos específicos para impresora */
}
```

Los estilos CSS que se aplican cuando se visualiza la página en una pantalla se obtienen mediante el recurso enlazado con la etiqueta <link> y mediante el archivo CSS externo incluido con la regla de tipo @import. Además, los estilos aplicados cuando se imprime la página se indican directamente en la página HTML mediante la regla de tipo @media.

## 1.9. Comentarios

CSS permite incluir comentarios entre sus reglas y estilos. Los comentarios son contenidos de texto que el diseñador incluye en el archivo CSS para su propia información y utilidad. Los navegadores ignoran por completo cualquier comentario de los archivos CSS, por lo que es común utilizarlos para estructurar de forma clara los archivos CSS complejos.

El comienzo de un comentario se indica mediante los caracteres /* y el final del comentario se indica mediante */, tal y como se muestra en el siguiente ejemplo:

```
/* Este es un comentario en CSS */
```

Los comentarios pueden ocupar tantas líneas como sea necesario, pero no se puede incluir un comentario dentro de otro comentario:

```
/* Este es un
comentario CSS de varias
lineas */
```

Aunque los navegadores ignoran los comentarios, su contenido se envía junto con el resto de estilos, por lo que no se debe incluir en ellos ninguna información sensible o confidencial.

La sintaxis de los comentarios CSS es muy diferente a la de los comentarios HTML, por lo que no deben confundirse:

```
<!-- Este es un comentario en HTML -->
<!-- Este es un
comentario HTML de varias
lineas -->
```

# 1.10. Sintaxis de la definición de cada propiedad CSS

A lo largo de los próximos capítulos, se incluyen las definiciones formales de la mayoría de propiedades de CSS. La definición formal se basa en la información recogida en el estándar oficial y se muestra en forma de tabla.

Una de las principales informaciones de cada definición es la lista de posibles valores que admite la propiedad. Para definir la lista de valores permitidos se sigue un formato que es necesario detallar.

Si el valor permitido se indica como una sucesión de palabras sin ningún carácter quelas separe (paréntesis, comas, barras, etc.) el valor de la propiedad se debe indicar tal y como se muestra y con esas palabras en el mismo orden.

Si el valor permitido se indica como una sucesión de valores separados por una barra simple (carácter |) el valor de la propiedad debe tomar uno y sólo uno de los valores indicados. Por ejemplo, la notación <porcentaje> | <medida> | inherit indica que la propiedad solamente puede tomar como valor la palabra reservada inherit o un porcentaje o una medida.

Si el valor permitido se indica como una sucesión de valores separados por una barra doble (símbolo ||) el valor de la propiedad puede tomar uno o más valores de los indicados y en cualquier orden.

Por ejemplo, la notación <color> || <estilo> || <medida> indica que la propiedad puede tomar como valor cualquier combinación de los valores indicados y en cualquier orden. Se podría establecer un color y un estilo, solamente una medida o una medida y un estilo. Además, el orden en el que se indican los valores es indiferente. Opcionalmente, se pueden utilizar paréntesis para agrupar diferentes valores.

Por último, en cada valor o agrupación de valores se puede indicar el tipo de valor: opcional, obligatorio, múltiple o restringido.

El carácter * indica que el valor ocurre cero o más veces; el carácter+ indica que el valor ocurre una o más veces; el carácter ? indica que el valor es opcional y por último, el carácter {número_1, número_2} indica que el valor ocurre al menos tantas veces como el valor indicado en número_1 y como máximo tantas veces como el valor indicado en número_2.

Por ejemplo, el valor [<family-name> , ]* indica que el valor de tipo <family_name> seguido por una coma se puede incluir cero o más veces. El valor <url>? <color> significa que la URL es opcional y el color obligatorio y en el orden indicado. Por último, el valor [<medida> | thick | thin] {1,4} indica que se pueden escribir entre 1 y 4 veces un valor que sea o una medida o la palabra thick o la palabra thin.

No obstante, la mejor forma de entender la notación formal para las propiedades de CSS es observar la definición de cada propiedad y volver a esta sección siempre que sea necesario.

# Capítulo 2. Selectores

Para crear diseños web profesionales, es imprescindible conocer y dominar los selectores de CSS. Como se vio en el capítulo anterior, una regla de CSS está formada por una parte llamada "selector" y otra parte llamada "declaración".

La declaración indica *"qué hay que hacer"* y el selector indica *"a quién hay que hacérselo"*. Por lo tanto, los selectores son imprescindibles para aplicar de forma correcta los estilos CSS en una página.

A un mismo elemento HTML se le pueden asignar *infinitas* reglas CSS y cada regla CSS puede aplicarse a un número *infinito* de elementos. En otras palabras, una misma regla puede aplicarse sobre varios selectores y un mismo selector se puede utilizar en varias reglas.

El estándar de CSS 2.1 incluye una docena de tipos diferentes de selectores, que permiten seleccionar de forma muy precisa elementos individuales o conjuntos de elementos dentro de una página web.

No obstante, la mayoría de páginas de los sitios web se pueden diseñar utilizando solamente los cinco selectores básicos.

## 2.1. Selectores básicos

### 2.1.1. Selector universal

Se utiliza para seleccionar todos los elementos de la página. El siguiente ejemplo elimina el margen y el relleno de todos los elementos HTML (por ahora no es importante fijarse en la parte de la declaración de la regla CSS):

```
* {
margin: 0;
padding: 0;
}
```

El selector universal se indica mediante un asterisco (*). A pesar de su sencillez, no se utiliza habitualmente, ya que es difícil que un mismo estilo se pueda aplicar a todos los elementos de una página.

No obstante, sí que se suele combinar con otros selectores y además, forma parte de algunos *hacks* muy utilizados, como se verá más adelante.

### 2.1.2. Selector de tipo o etiqueta

Selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector. El siguiente ejemplo selecciona todos los párrafos de la página:

```
p {
  ...
}
```

Para utilizar este selector, solamente es necesario indicar el nombre de una etiqueta HTML (sin los caracteres < y >) correspondiente a los elementos que se quieren seleccionar.

El siguiente ejemplo aplica diferentes estilos a los titulares y a los párrafos de una página HTML:

```
h1 {
color: red;
}

h2 {
color: blue;
}

p {
color: black;
}
```

Si se quiere aplicar los mismos estilos a dos etiquetas diferentes, se pueden encadenar los selectores. En el siguiente ejemplo, los títulos de sección `h1`, `h2` y `h3` comparten los mismos estilos:

```
h1 {
color: #8A8E27;
font-weight: normal;
font-family: Arial, Helvetica, sans-serif;
}
h2 {
color: #8A8E27;
font-weight: normal;
font-family: Arial, Helvetica, sans-serif;
```

```
}
h3 {
color: #8A8E27;
font-weight: normal;
font-family: Arial, Helvetica, sans-serif;
}
```

En este caso, CSS permite agrupar todas las reglas individuales en una sola regla con un selector múltiple. Para ello, se incluyen todos los selectores separados por una coma (,) y el resultado es que la siguiente regla CSS es equivalente a las tres reglas anteriores:

```
h1, h2, h3 {
color: #8A8E27;
font-weight: normal;
font-family: Arial, Helvetica, sans-serif;
}
```

En las hojas de estilo complejas, es habitual agrupar las propiedades comunes de varios elementos en una única regla CSS y posteriormente definir las propiedades específicas de esos mismos elementos. El siguiente ejemplo establece en primer lugar las propiedades comunes de los títulos de sección (color y tipo de letra) y a continuación, establece el tamaño de letra de cada uno de ellos:

```
h1, h2, h3 {
color: #8A8E27;
font-weight: normal;
font-family: Arial, Helvetica, sans-serif;
}

h1 {font-size: 2em; }
h2 {font-size: 1.5em; }
h3 {font-size: 1.2em; }
```

### 2.1.3. Selector descendiente

Selecciona los elementos que se encuentran dentro de otros elementos. Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del otro elemento.

El selector del siguiente ejemplo selecciona todos los elementos  de la página que se encuentren dentro de un elemento <p>:

```
p span{color: red; }
```

Si el código HTML de la página es el siguiente:

```
<p>
  ...
<span>texto1</span>
  ...
<a href="">...<span>texto2</span></a>
  ...
</p>
```

explicaciónspan

El selector `p span` selecciona tanto texto1 como texto2. El motivo es que en el selector descendiente, un elemento no tiene que ser *"hijo directo"* de otro. La única condición es que un elemento debe estar dentro de otro elemento, sin importar lo profundo que se encuentre.

Al resto de elementos `<span>` de la página que no están dentro de un elemento `<p>`, no se les aplica la regla CSS anterior.

Los selectores descendientes permiten aumentar la precisión del selector de tipo o etiqueta. Así, utilizando el selector descendiente es posible aplicar diferentes estilos a los elementos del mismo tipo. El siguiente ejemplo amplía el anterior y muestra de color azul todo el texto de los `<span>` contenidos dentro de un `<h1>`:

```
p span  {color: red;  }
h1 span {color: blue; }
```

Con las reglas CSS anteriores:

- Los elementos `<span>` que se encuentran dentro de un elemento `<p>` se muestran de color rojo.
- Los elementos `<span>` que se encuentran dentro de un elemento `<h1>` se muestran de color azul.
- El resto de elementos `<span>` de la página, se muestran con el color por defecto aplicado por el navegador.

La sintaxis formal del selector descendiente se muestra a continuación:

```
selector1 selector2 selector3 ...selectorN
```

Los selectores descendientes siempre están formados por dos o más selectores separados entre sí por espacios en blanco. El último selector indica el elemento sobre el que se aplican los estilos y todos los selectores anteriores indican el lugar en el que se debe encontrar ese elemento.

En el siguiente ejemplo, el selector descendiente se compone de cuatro selectores:

```
p a span em{text-decoration: underline; }
```

Los estilos de la regla anterior se aplican a los elementos de tipo  que se encuentren dentro de elementos de tipo , que a su vez se encuentren dentro de elementos de tipo <a> que se encuentren dentro de elementos de tipo <p>.

No debe confundirse el selector descendiente con la combinación de selectores:

```
/* El estilo se aplica a todos los elementos "p", "a", "span" y "em" */
p, a, span, em{text-decoration: underline; }

/* El estilo se aplica solo a los elementos "em" que se
encuentran dentro de "p a span" */
p a span em{text-decoration: underline; }
```

Se puede restringir el alcance del selector descendiente combinándolo con el selector universal. El siguiente ejemplo, muestra los dos enlaces de color rojo:

```
p a {color: red; }

<p><a href="#">Enlace</a></p>
<p><span><a href="#">Enlace</a></span></p>
```

Sin embargo, en el siguiente ejemplo solamente el segundo enlace se muestra de color rojo:

```
p * a {color: red; }

<p><a href="#">Enlace</a></p>
<p><span><a href="#">Enlace</a></span></p>
```

La razón es que el selector p * a se interpreta como *todos los elementos de tipo <a> que se encuentren dentro de cualquier elemento que, a su vez, se encuentre dentro de un elemento de tipo <p>*. Como el primer elemento <a> se encuentra directamente bajo un elemento <p>, no se cumple la condición del selector p * a.

## 2.1.4. Selector de clase

Si se considera el siguiente código HTML de ejemplo:

```
<body>
<p>Loremipsum dolor sitamet...</p>
<p>Nuncsed lacus etestadipiscingaccumsan...</p>
<p> aptenttacitisociosquadlitora...</p>
</body>
```

¿Cómo se pueden aplicar estilos CSS sólo al primer párrafo? El selector universal (*) no se puede utilizar porque selecciona todos los elementos de la página. El selector de tipo o etiqueta (p) tampoco se puede utilizar porque seleccionaría todos los párrafos. Por último, el selector descendente (body p) tampoco se puede utilizar porque todos los párrafos se encuentran en el mismo sitio.

Una de las soluciones más sencillas para aplicar estilos a un solo elemento de la página consiste en utilizar el atributo class de HTML sobre ese elemento para indicar directamente la regla CSS que se le debe aplicar:

```
<body>
<p class="destacado">Loremipsum dolor sit amet...</p>
<p>Nuncsed lacus etestadipiscingaccumsan...</p>
<p> aptenttacitisociosquadlitora...</p>
</body>
```

A continuación, se crea en el archivo CSS una nueva regla llamada destacado con todos los estilos que se van a aplicar al elemento. Para que el navegador no confunda este selector con los otros tipos de selectores, se prefija el valor del atributo class con un punto (.) tal y como muestra el siguiente ejemplo:

```
.destacado{color: red; }
```

El selector .destacado se interpreta como *"cualquier elemento de la página cuyo atributo class sea igual a destacado"*, por lo que solamente el primer párrafo cumple esa condición.

Este tipo de selectores se llaman selectores de clase y son los más utilizados junto con los selectores de ID que se verán a continuación. La principal característica de este selector es que en una misma página HTML varios elementos diferentes pueden utilizar el mismo valor en el atributo class:

```
<body>
<p class="destacado">Loremipsum dolor sit amet...</p>
<p>Nuncsed lacus et<ahref="#"class="destacado">estadipiscing</a>accumsan...</p>
<p> aptenttaciti<emclass="destacado">sociosqu ad</em>litora...</p>
</body>
```

Los selectores de clase son imprescindibles para diseñar páginas web complejas, ya que permiten disponer de una precisión total al seleccionar los elementos. Además, estos selectores permiten reutilizar los mismos estilos para varios elementos diferentes.

A continuación se muestra otro ejemplo de selectores de clase:

```
.aviso{
padding: 0.5em;
border: 1pxsolid#98be10;
background: #f6feda;
}

.error{
color: #930;
font-weight: bold;
}
<span class="error">...</span>

<div class="aviso">...</div>
```

El elemento  tiene un atributo class="error", por lo que se le aplican las reglas CSS indicadas por el selector .error. Por su parte, el elemento  tiene un atributo class="aviso", por lo que su estilo es el que definen las reglas CSS del selector .aviso.

En ocasiones, es necesario restringir el alcance del selector de clase. Si se considera de nuevo el ejemplo anterior:

```
<body>
<p class="destacado"> Loremipsum dolor sitamet...</p>
<p>Nunc sed lacus et <a href="#"class="destacado"> estadipiscing</a> accumsan...</p>
<p>Classaptenttaciti<em class="destacado">sociosqu ad</em>litora...</p>
</body>
```

¿Cómo es posible aplicar estilos solamente al párrafo cuyo atributo class sea igual a destacado? Combinando el selector de tipo y el selector de clase, se obtiene un selector mucho más específico:

```
p.destacado{color: red}
```

El selector p.destacado se interpreta como *"aquellos elementos de tipo <p> que dispongan de un atributo class con valor destacado"*. De la misma forma, el selector a.destacado solamente selecciona los enlaces cuyo atributo class sea igual a destacado.

De lo anterior se deduce que el atributo .destacado es equivalente a *.destacado, por lo que todos los diseñadores obvian el símbolo * al escribir un selector de clase normal.

No debe confundirse el selector de clase con los selectores anteriores:

```
/* Todos los elementos de tipo "p" con atributo class="aviso" */
p.aviso{ ...}

/* Todos los elementos con atributo class="aviso" que estén dentro
de cualquier elemento de tipo "p" */
p .aviso{ ...}

/* Todos los elementos "p" de la página y todos los elementos con
Atributo class="aviso" de la página */
p, .aviso{ ...}
```

Por último, es posible aplicar los estilos de varias clases CSS sobre un mismo elemento. La sintaxis es similar, pero los diferentes valores del atributo class se separan con espacios en blanco. En el siguiente ejemplo:

```
<p class="especial destacado error">Párrafo de texto...</p>
```

Al párrafo anterior se le aplican los estilos definidos en las reglas .especial, .destacado y .error, por lo que en el siguiente ejemplo, el texto del párrafo se vería de color rojo, en negrita y con un tamaño de letra de 15 píxel:

```
.error { color: red; }
.destacado { font-size: 15px; }
```

```
.especial  { font-weight: bold; }

<p class="especial destacado error">Párrafo de texto...</p>
```

Si un elemento dispone de un atributo `class` con más de un valor, es posible utilizar un selector más avanzado:

```
.error { color: red; }
.error.destacado{ color: blue; }
.destacado{ font-size: 15px; }
.especial  { font-weight: bold; }

<p class="especial destacado error">Párrafo de texto...</p>
```

En el ejemplo anterior, el color de la letra del texto es azul y no rojo. El motivo es que se ha utilizado un selector de clase múltiple `.error.destacado`, que se interpreta como *"aquellos elementos de la página que dispongan de un atributo `class` con al menos los valores `error` y `destacado`"*.

## 2.1.5. Selectores de ID

En ocasiones, es necesario aplicar estilos CSS a un único elemento de la página. Aunque puede utilizarse un selector de clase para aplicar estilos a un único elemento, existe otro selector más eficiente en este caso.

El selector de ID permite seleccionar un elemento de la página a través del valor de su atributo `id`. Este tipo de selectores sólo seleccionan un elemento de la página porque el valor del atributo `id` no se puede repetir en dos elementos diferentes de una misma página.

La sintaxis de los selectores de ID es muy parecida a la de los selectores de clase, salvo que se utiliza el símbolo de la almohadilla (#) en vez del punto (.) como prefijo del nombre de la regla CSS:

```
#destacado{color: red; }

<p>Primer párrafo</p>
<p id="destacado">Segundo párrafo</p>
<p>Tercer párrafo</p>
```

En el ejemplo anterior, el selector `#destacado` solamente selecciona el segundo párrafo (cuyo atributo `id` es igual a `destacado`).

La principal diferencia entre este tipo de selector y el selector de clase tiene que ver con HTML y no con CSS. Como se sabe, en una misma página, el valor del atributo `id` debe ser único, de forma que dos elementos diferentes no pueden tener el mismo valor de `id`. Sin embargo, el atributo `class` no es obligatorio que sea único, de forma que muchos elementos HTML diferentes pueden compartir el mismo valor para su atributo `class`.

De esta forma, la recomendación general es la de utilizar el selector de ID cuando se quiere aplicar un estilo a un solo elemento específico de la página y utilizar el selector de clase cuando se quiere aplicar un estilo a varios elementos diferentes de la página HTML.

Al igual que los selectores de clase, en este caso también se puede restringir el alcance del selector mediante la combinación con otros selectores. El siguiente ejemplo aplica la regla CSS solamente al elemento de tipo <p> que tenga un atributo id igual al indicado:

```
p#aviso{color: blue; }
```

A primera vista, restringir el alcance de un selector de ID puede parecer absurdo. En realidad, un selector de tipo p#aviso sólo tiene sentido cuando el archivo CSS se aplica sobre muchas páginas HTML diferentes.

En este caso, algunas páginas pueden disponer de elementos con un atributo id igual a aviso y que no sean párrafos, por lo que la regla anterior no se aplica sobre esos elementos.

No debe confundirse el selector de ID con los selectores anteriores:

```
/* Todos los elementos de tipo "p" con atributo id="aviso" */
p#aviso{ ...}

/* Todos los elementos con atributo id="aviso" que estén dentro
        de cualquier elemento de tipo "p" */
p #aviso{ ...}

/* Todos los elementos "p" de la página y todos los elementos con
        atributo id="aviso" de la página */
p, #aviso{ ...}
```

### 2.1.6. Combinación de selectores básicos

CSS permite la combinación de uno o más tipos de selectores para restringir el alcance de las reglas CSS. A continuación se muestran algunos ejemplos habituales de combinación de selectores.

```
.aviso.especial{ ...}
```

El anterior selector solamente selecciona aquellos elementos con un class="especial" que se encuentren dentro de cualquier elemento con un class="aviso".

Si se modifica el anterior selector:

```
div.avisospan.especial{ ...}
```

Ahora, el selector solamente selecciona aquellos elementos de tipo  con un atributo class="especial" que estén dentro de cualquier elemento de tipo  que tenga un atributo class="aviso".

La combinación de selectores puede llegar a ser todo lo compleja que sea necesario:

ul# menuPrincipalli.destacadoa#inicio{ ... }

El anterior selector hace referencia al enlace con un atributo id igual a inicio que se encuentra dentro de un elemento de tipo <li> con un atributo class igual a destacado, que forma parte de una lista <ul> con un atributo id igual a menuPrincipal.

# Ejercicio 1

A partir del código HTML y CSS que se muestra, añadir los selectores CSS que faltan para aplicar los estilos deseados. Cada regla CSS incluye un comentario en el que se explica los elementos a los que debe aplicarse:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<htmlxmlns="http://www.w3.org/1999/xhtml">
<head>
<metahttp-equiv="Content-Type"content="text/html; charset=iso-8859-1" />
<title>Ejercicio de selectores</title>
<styletype="text/css">
/* Todos los elementos de la pagina */
{ font: 1/1.3em Arial, Helvetica, sans-serif; }

/* Todos los parrafos de la pagina */
{ color: #555; }

/* Todos los párrafos contenidos en #primero */
{ color: #336699; }

/* Todos los enlaces la pagina */
{ color: #CC3300; }

/* Los elementos "em" contenidos en #primero */
{ background: #FFFFCC; padding: .1em; }

/* Todos los elementos "em" de clase "especial" en toda la pagina */
{ background: #FFCC99; border: 1px solid #FF9900; padding: .1em; }

/* Elementos "span" contenidos en .normal */
{ font-weight: bold; }

</style>
</head>

<body>

<div id="primero">
<p>Loremipsum dolor sit amet, <a href="#">consectetueradipiscingelit</a>.
Praesentblanditnibh at felis.Sednecdiam in dolor vestibulumaliquet. Duisullamcorper,
nisi non facilisismolestie, <em>loremsemaliquamnulla</em>, id laciniavelit mi
vestibulumenim.</p>

</div>

<div class="normal">
<p>Phaselluseuvelitsedloremsodalesegestas.Utfeugiat.<span><ahref="#">Donecporttitor</a>,
magna euvariusluctus,</span>metusmassatristiquemassa, in imperdietestvelitvel magna.
Phaselluserat.Duisrisus.<ahref="#">Maecenas dictum</a>, nibh vitae pellentesqueauctor,
tellusvelitconsectetuertellus, temporpretiumfelistellus at metus.</p>
```

```
<p>Cumsociisnatoque<em class="especial">penatibus et magnis</em> dis parturient montes,
nasceturridiculus mus. Proinaliquamconvallis ante.
Pellentesquehabitantmorbitristiquesenectus et netus et malesuadafamesacturpisegestas.
Nunc aliquet. Sed eumetus. Duis justo.</p>

<p>Donecfacilisisblanditvelit. Vestibulumnisi. Proinvolutpat, <emclass="especial">enim
id iaculiscongue</em>, orci justo ultrices tortor, <ahref="#">quis lacinia eros libero
in eros</a>. Sed malesuadaduivelquam. Integer at eros.</p>
</div>

</body>
</html>
```

**SOLUCIÓN:**

```
<style type="text/css">
/* Todos los elementos de la pagina */
* { font: 1em/1.3 Arial, Helvetica, sans-serif; }

/* Todos los parrafos de la pagina */
p { color: #555; }

/* Todos los prrafos contenidos en #primero */
#primero p { color: #336699; }

/* Todos los enlaces la pagina */
a { color: #CC3300; }

/* Los elementos "em" contenidos en #primero */
#primero em { background: #FFFFCC; padding: .1em; }

/* Todos los elementos "em" de clase "especial" en toda la pagina */
em.especial { background: #FFCC99; border: 1px solid #FF9900; padding: .1em; }

/* Elementos "span" contenidos en .normal */
.normal span { font-weight: bold; }
```

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Praesent blandit nibh at felis. Sed nec diam in dolor vestibulum aliquet. Duis ullamcorper, nisi non facilisis molestie, lorem sem aliquam nulla, id lacinia velit mi vestibulum enim.

Phasellus eu velit sed lorem sodales egestas. Ut feugiat. Donec porttitor, **magna eu varius luctus,** metus massa tristique massa, in imperdiet est velit vel magna. Phasellus erat. Duis risus. Maecenas dictum, nibh vitae pellentesque auctor, tellus velit consectetuer tellus, tempor pretium felis tellus at metus.

Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Proin aliquam convallis ante. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nunc aliquet. Sed eu metus. Duis justo.

Donec facilisis blandit velit. Vestibulum nisi. Proin volutpat, enim id iaculis congue, orci justo ultrices tortor, quis lacinia eros libero in eros. Sed malesuada dui vel quam. Integer at eros.