

# 1. Introducción

**CSS (Cascading Style Sheets)** permite a los desarrolladores Web controlar el estilo y el formato de múltiples páginas Web al mismo tiempo.

Antes del uso de **CSS**, los diseñadores de páginas web debían definir el aspecto de cada elemento dentro de las etiquetas **HTML** de la página. El principal problema de esta forma de definir el aspecto de los elementos es que habría que definir el formato de cada uno de los elementos que formen la página, lo cual hace que sea muy difícil de actualizar.

**CSS** permite separar los contenidos de la página y su aspecto. Para ello se define en una zona reservada el formato de cada uno de los elementos de la web. Cualquier cambio en el estilo marcado para un elemento en la **CSS** afectará a todas las páginas vinculadas a ella en las que aparezca ese elemento. Las hojas de estilo están compuestas por una o más reglas de estilo aplicadas a un documento **HTML** o **XML**.

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, cabecera, texto destacado, etc. Una vez creados los contenidos, se utiliza el lenguaje **CSS** para definir el formato de cada elemento.

**CSS** obliga a crear documentos semánticos **HTML/XHTML**, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinitud de dispositivos diferentes.

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 estilos a los documentos electrónicos. La guerra de navegadores y la falta de un estándar para la definición de los estilos dificultaban la creación de documentos que tuvieran igual apariencia en distintos navegadores.

En 1995, el **W3C** añadió a su grupo de trabajo de **HTML** el desarrollo y estandarización de **CSS**.

**CSS 1**, se publicó en 1996, es la primera recomendación oficial.

**CSS 2**, publicada en 1998, es la segunda recomendación oficial.

**CSS 3**, empezó a desarrollarse en 1998, sus primeros borradores se publicaron en junio de 1999. A diferencia de las versiones anteriores la especificación se divide en varios documentos separados llamados módulos. Esto permite que diferentes módulos se encuentren en diferentes estados de desarrollo y no todos pasen a ser recomendaciones oficiales del **W3C**. Un módulo debe primero pasar por la fase de "candidato" antes de ser aprobado.

**CSS 4**, última versión sobre la que no existe una especificación integrada ya que al igual que **CSS 3** se encuentra dividido en diferentes módulos. A partir de la división en módulos que se hizo en **CSS 3** la evolución de cada uno de ellos es independiente lo que provoca que actualmente la mayoría de módulos estén es especificación de nivel 3 mientras que otros (por ejemplo [Selectores](#)) se encuentre en nivel 4.

En el siguiente enlace encontrarás el soporte que cada uno de los navegadores más actuales tienen sobre las etiquetas CSS.

[CSS Reference With Browser Support - W3school](#)

## 2. Cómo incluir CSS en un documento HTML o XHTML

### 2.1. Declaración en línea

Dentro de la propia etiqueta mediante el atributo style.

**Deben evitarse** para preservar el principio de separación de contenidos y formato.

```
<p style="color: red;"> Loren ipsum </p>
```

### 2.2. Declaración Interna

En el encabezado del documento dentro de las etiquetas **< style>**

```
<style type="text/css"> Loren ipsum </style>
```

### 2.3. Declaración en archivo externo

En el encabezado mediante la etiqueta link:

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

El archivo de estilos tendrá extensión .css.

El código del archivo de estilos no tendrá etiqueta de declaración de estilo, por ejemplo:

```
h1 {background-color: #blue}
```

Otra forma de usar hojas de estilo externas es mediante la etiqueta @import. Es una directiva CSS no html.

```
<style type="text/css"> @import url("formato1.css"); </style>
```

### 3. Sintaxis de las reglas de estilo

Cada uno de los estilos que componen una hoja de estilos CSS se denomina regla. Cada regla se forma por:

- **Selector:** indica el elemento o elementos HTML a los que se aplica la regla CSS
- **Llave de apertura,** {
- **Declaración:** especifica los estilos que se aplican a los elementos.

**Propiedad:** permite modificar el aspecto de un atributo del elemento.

**Valor:** indica el nuevo valor del atributo modificado en el elemento.

- **Llave de cierre,** }.

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

En este caso el **selector** es "p" (párrafo), la **declaración** es "color: blue" donde encontramos la **propiedad** color y el **valor** blue.

Un archivo CSS puede contener infinitas reglas CSS, cada regla puede contener varios selectores y cada declaración puede estar formada por diferentes declaraciones.

### 4. Cascada y herencia de estilos

En ocasiones y dependiendo de cómo se haya hecho la definición de estilos es posible que el navegador se encuentre estilos contradictorios. Ante esta situación el navegador aplicará la siguiente precedencia

1. Declaración en línea
2. Declaración interna
3. Declaración externa
4. Propiedades por defecto del navegador

Por otro lado las hojas de estilo también permiten la herencia de propiedades, Si tenemos varios elementos **HTML** anidados los elementos más internos heredan los estilos de los externos en los que están anidados siempre y cuando ellos no los tengan definidos

### 5. Selectores

A la hora de aplicar estilos a nuestros elementos **HTML** necesitamos un mecanismo que permita identificar sobre cuál o cuáles de estos elementos queremos actuar. Para ello se utilizan los **selectores**, estos permiten identificar a qué elementos de nuestro código **HTML** vamos a aplicar el estilo definido. Existe diferentes selectores, a continuación veremos los más importantes.

#### Selector universal (\*)

Sirve para seleccionar todos los elementos de la página.

```
* {margin:10px; padding: 5px};
```

#### Selectores de etiqueta

En este caso los estilos se aplican solo a la etiqueta.

```
p { text-align: center}
```

En este ejemplo, los párrafos serán alineados al centro.

Si queremos ajustar los mismos estilos a dos etiquetas diferentes podemos ponerlos separados por comas.

```
p, h1, h2 { text-align: center}
```

En este ejemplo, tanto los párrafos como los encabezados de tipo 1 y 2 serán alineados al centro.

#### 5.1. Selectores de clase

En ocasiones no vamos a querer aplicar el mismo formato en todas las etiquetas del mismo tipo. Para ello podemos asignar a las etiquetas a una **clase** determinada, esto requiere que identifiquemos las etiquetas afectadas mediante el atributo `class` de la forma.

```
<p class="parrafoCentrado"> ... </p>
```

El selector se especificaría:

```
p.parrafoCentrado {text-align: center; }
```

De manera que este estilo se aplicaría a todas las etiquetas `p` que tengan el atributo `class` a valor `parrafoCentrado`.

Sin embargo si hacemos la misma definición pero omitimos el identificador de etiqueta:

```
.parrafoCentrado { text-align: center; }
```

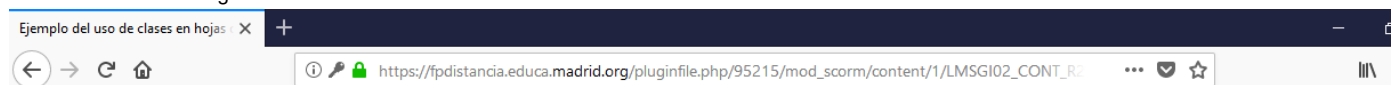
Se aplicaría a todas las etiquetas que perteneciesen a la clase "parrafoCentrado" sean del tipo que sean. Por ejemplo a una que fuese:

```
<h1 class="parrafoCentrado">encabezado 1 centrado</h1>
```

Veamos a continuación un código de ejemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset='UTF-8'>
    <title>Ejemplo del uso de clases en hojas de estilo</title>
    <style type="text/css">
      .clase_azul{color:blue}
      p.clase_roja{color:#ff0000; font-style:italic; font-weight:bolder; font-family:courier;}
    </style>
  </head>
  <body>
    <h3 class="clase_azul">Ejemplo del uso de clases en hojas de estilo</h3>
    <p>Cualquier elemento sobre el que apliquemos la clase clase_azul tendrá el texto azul.</p>
    <p class="clase_azul"> Incluso el párrafo.</p>
    <p class="clase_roja">Sobre el párrafo podemos aplicar la clase clase_roja y el texto será rojo, en negrita cursiva y la familia del texto courier.</p>
    <h3 class="clase_roja"> Pero este texto no aparecerá formateado ya que regla de la clase clase_roja solo actúa sobre párrafos.</h3>
  </body>
</html>
```

El resultado en el navegador sería:



### Ejemplo del uso de clases en hojas de estilo

Cualquier elemento sobre el que apliquemos la clase `clase_azul` tendrá el texto azul.

Incluso el párrafo.

*Sobre el párrafo podemos aplicar la clase `clase_roja` y el texto será rojo, en negrita cursiva y la familia del texto courier.*

**Pero este texto no aparecerá formateado ya que la clase `clase_roja` sólo actúa sobre párrafos.**