Introducción CSS

El gran impulso de los lenguajes de hojas de estilos se produjo con el boom de Internet y el crecimiento exponencial del lenguaje HTML. Entre la fuerte competencia de navegadores web y la falta de un estándar para la definición de los estilos, se dificultaba la creación de documentos con la misma apariencia en diferentes navegadores.

A mediados de la década de 1990 el W3C, encargado de crear todos los estándares relacionados con la web, propuso la creación de un lenguaje de hojas de estilos específicos para el lenguaje HTML y se presentaron propuestas. Las dos propuestas que se tuvieron en cuenta fueron la CHSS (Cascading HTML Style Sheets) y la SSP (Stream-based Style Sheet Proposal). La propuesta CHSS fue realizada por [Håkon Wium Lie](https://en.wikipedia.org/wiki/H%C3%A5kon_Wium_Lie" \t "_blank) y SSP fue propuesto por [Bert Bos](https://es.wikipedia.org/wiki/Bert_Bos" \t "_blank) . 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](https://developer.mozilla.org/es/docs/Web/CSS) , Cascading Style Sheets.

En 1995, el W3C decidió gestionar el desarrollo y estandarización de CSS y agregó el proyecto a su grupo de trabajo de HTML. A finales de 1996, el W3C publicó la primera recomendación oficial, conocida como " [CSS nivel 1](https://www.ecured.cu/CSS) ".

**¿Qué es CSS?.**

Css es un leguaje que trabaja junto con HTML para proporcionar estilos visuales a los elementos de un documento web.

**Características:**

* .- Ahorra trabajo. Se puede controlar el diseño de varias páginas HTML a la vez.
* .- Se pueden almacenar en archivos \* .css

CSS 3 es la última versión estándar.

¿Para qué utilizar CSS?

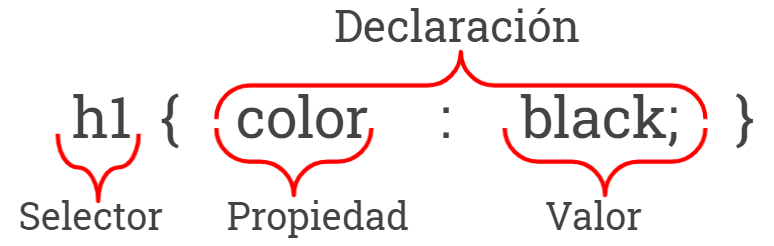
Para definir estilos en los documentos web, incluyendo el diseño, la disposición de los elementos y para responder a las variaciones en la pantalla en cuanto a diferentes dispositivos y tamaños de pantalla.

Ventajas

1. Control centralizado de la presentación de un sitio web completo con lo que se agiliza considerablemente la actualización y mantenimiento.
2. Los Navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web, con lo que aumenta considerablemente la accesibilidad. Por ejemplo, las personas con deficiencias visuales pueden configurar su propia hoja de estilo para aumentar el tamaño del texto o remarcar más los enlaces.
3. Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestra o incluso una elección del usuario. Por ejemplo, para ser impresa, mostrada en un dispositivo móvil, o ser "leída" por un sintetizador de voz.
4. El documento HTML en sí mismo es más claro de entender y se consigue reducir considerablemente su tamaño (siempre y cuando no se utiliza estilo en línea).

## Reglas CSS

CSS define un conjunto de reglas que permiten describir cada una de las partes que componen los estilos CSS.



**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.

**Propiedad:** permite modificar el aspecto de una característica del elemento.

**Valor:** indica el nuevo valor de la característica modificada en el elemento.

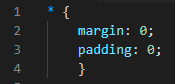
## Selectores

Nos indican qué elemento HTML hay que aplicar el estilo. Una misma regla puede aplicar a varios selectores y, a un mismo selector se le pueden aplicar varias reglas.

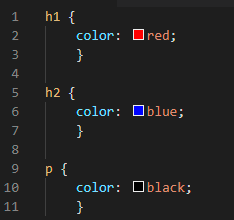
Para mayor información referirse a [este enlace](https://www.w3.org/wiki/CSS_/_Selectores_CSS#Lista_de_selectores_css) .

## Selectores básicos

**Selector universal** : se utiliza para seleccionar todos los elementos de la página:



**Selector de tipo o etiqueta:** selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector:



## Selector descendente

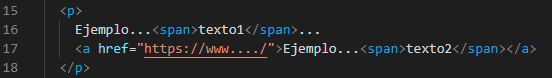
Selecciona los elementos que se encuentran dentro de otros elementos. Un elemento es descendente de otro cuando se encuentra entre las etiquetas de apertura y de cierre de otro elemento.

C:\Users\54381\Downloads\image2.png

La sintaxis formal del selector descendente se muestra a continuación:

**elemento1 elemento2 elemento3 ... elementoN**

C:\Users\54381\Downloads\image3.png

Si el código HTML de la página es el siguiente: 

El selector p span selecciona tanto texto1 como texto2. El motivo es que en el selector descendente, un elemento no tiene que ser descendiente directo del otro. La única condición es que un elemento debe estar dentro de otro elemento, sin importar el nivel de profundidad en el que se encuentre.

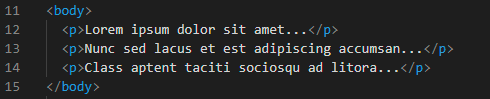
No se les aplica la regla CSS anterior si el resto de los elementos <span> de la página no están dentro de un elemento <p>.

Los selectores descendentes nos permiten aumentar la precisión del selector de tipo o etiqueta. Así, utilizando el selector descendente podemos aplicar diferentes estilos a los elementos del mismo tipo.

Para conocer más te invitamos hacer clic en el este [enlace](https://uniwebsidad.com/libros/css/capitulo-2/selectores-basicos).

## Selector de clase o selector Class

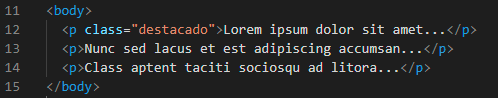
Si consideramos el siguiente código HTML de ejemplo:



**¿Cómo podemos aplicar estilos CSS sólo al primer párrafo?**

El selector universal (\*) no se puede utilizar porque seleccionaría 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:

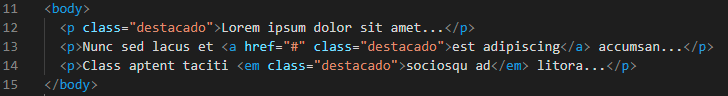


A continuación, creamos en el archivo CSS una nueva regla llamada destacado con todos los estilos que vamos a aplicar al elemento. Para que el navegador no confunda este selector con los otros tipos de selectores, prefijamos el valor del atributo class con un punto (.) Tal como vemos en la siguiente línea:

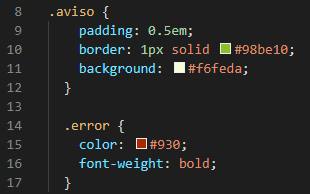
C:\Users\54381\Downloads\image5.png

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 del anterior ejemplo cumple esa condición.

Este tipo de selectores de clase son los más utilizados junto con los selectores de ID que veremos 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:



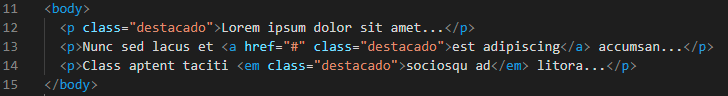
Los selectores de clase resultan imprescindibles para diseñar páginas web complejas, ya que nos permiten disponer de una precisión total al seleccionar los elementos. Además, estos selectores nos permiten reutilizar los mismos estilos para varios elementos diferentes:



C:\Users\54381\Downloads\image9.png

El elemento <span> tiene un atributo class = "error", por lo que se le aplica las reglas CSS indicadas por el selector .error. Por su parte, el elemento <div> tiene un atributo class = "aviso", por lo que su estilo es el que define las reglas CSS del selector .aviso.

En ciertos casos, es necesario restringir el alcance del selector de clase. Si utilizamos el ejemplo anterior:



**¿Cómo podemos 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:

C:\Users\54381\Downloads\image10.png

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

De lo anterior deducimos 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 debemos confundir 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 {...} / \* notar el espacio entre py la clase \* /

/ \* Todos los elementos "p" de la página y todos los elementos con

   atributo class = "aviso" de la página \* /

p, .aviso {...} / \* notar la, (coma) entre py la clase \* /

Para completar, 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.

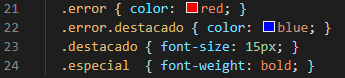
C:\Users\54381\Downloads\image13.png

Al párrafo anterior le aplicamos 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:

C:\Users\54381\Downloads\image15.png

C:\Users\54381\Downloads\image13 (1).png

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



C:\Users\54381\Downloads\image13 (1).png

En el ejemplo anterior, el color de la letra del texto es azul y no rojo. Esto se debe a que hemos utilizado un selector de clase múltiple .error.destacado, que se interpreta como "aquellos elementos de la página que están disponibles de un atributo class con al menos los valores error y destacado".

Para saber más podes hacer clic en [este enlace](https://uniwebsidad.com/libros/css/capitulo-2/selectores-basicos).

Selectores de ID

En ocasiones, es necesario aplicar estilos CSS a un único elemento de la página. Aunque puede aplicar un selector de clase para 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 selecciona 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:

C:\Users\54381\Downloads\image17.png

C:\Users\54381\Downloads\image18.png

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. 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 queremos aplicar un estilo determinado a varios elementos diferentes de la página HTML.

Al igual que los selectores de clase, en este caso también podemos 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:

C:\Users\54381\Downloads\image19.png

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 {...}

## Selectores avanzados

Se trata de un selector similar al selector descendente, pero muy diferente en su funcionamiento. Se utiliza para seleccionar un elemento que es hijo directo de otro elemento y se indica mediante el "signo de mayor que" (>):

C:\Users\54381\Downloads\image20.png

C:\Users\54381\Downloads\image21.png

En el ejemplo anterior, el selector p> span se interpreta como "cualquier elemento <span> que sea hijo directo de un elemento <p>", por lo que el primer elemento <span> cumple la condición del selector. Sin embargo, el segundo elemento <span> no la cumple porque es descendiente pero no es hijo directo de un elemento <p>.

El siguiente ejemplo muestra las diferencias entre el selector descendente y el selector de hijos:

C:\Users\54381\Downloads\image22.png

C:\Users\54381\Downloads\image23.png

El primer selector es de tipo descendente y por tanto se aplica a todos los elementos <a> que se encuentran dentro de los elementos <p>. En este caso, los estilos de este selector se aplican a los dos enlaces.

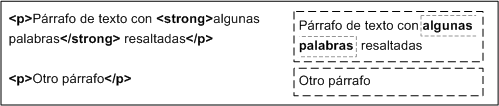
Por otra parte, el selector de hijos obliga a que el elemento <a> sea hijo directo de un elemento <p>. Por lo tanto, los estilos del selector p> a no se aplican al segundo enlace del ejemplo anterior.

## Modelo de Cajas

Tal como dijimos en la unidad de HTML, si hacemos una analogía con una estructura de cajas de cartón, podemos decir que hay ciertas cajas que van dentro de otras y ciertas cajas que van una al lado de otra.

El "modelo de caja" es probablemente la característica más importante del lenguaje de hojas de estilos CSS, ya que condiciona el diseño de todas las páginas web. El modelo de cajas es el comportamiento de CSS que hace que todos los elementos de las páginas sean representados mediante cajas rectangulares.

Cada vez que se inserta un elemento HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento. La siguiente imagen muestra tres cajas rectangulares que crean los tres elementos HTML de una porción de página de ejemplo.



Los navegadores web crean y colocan las cajas de forma automática, pero CSS permite modificar todas sus características. Cada una de las cajas está formada por seis partes que se describen a continuación:

1. **Content** (contenido): se trata del contenido HTML del elemento (las palabras de un párrafo, una imagen, el texto de una lista de elementos, etc.)
2. **Padding** (relleno): espacio libre opcional existente entre el contenido y el borde.
3. **Border** (borde): línea que encierra completamente el contenido y su relleno.
4. **Background-image** (Imagen de fondo): imagen que se muestra por detrás del contenido y el espacio de relleno.
5. **Background-color** (color de fondo): color que se muestra por detrás del contenido y el espacio de relleno.
6. **Margin** (margen): separación opcional existente entre la caja y el resto de cajas adyacentes.

### Dimensiones de las cajas

#### **Anchura**

Para los elementos de bloque y las imágenes, la propiedad width (anchura) permite establecer la anchura directamente mediante una medida.

Anchura = width

**En CSS:**

C:\Users\54381\Downloads\a1.jpg

**En HTML:**

C:\Users\54381\Downloads\a2.jpg

Si se utilizan unidades de medida, los valores indicados no pueden ser negativos. Si en vez de una unidad de medida se indica un porcentaje, la referencia de ese valor es la anchura del elemento que lo contiene. El valor heredado indica que la anchura del elemento se hereda de su elemento padre.

Si se establece la anchura de un elemento con la unidad de medida em, el valor indicado toma como referencia el tamaño de letra del propio elemento.

El valor auto es el valor por defecto de la anchura de todos los elementos. En este caso, el navegador determina la anchura de cada elemento teniendo en cuenta, entre otros, el tipo de elemento que se trata (de bloque o en línea), el sitio disponible en la pantalla del navegador y los contenidos de los elementos.

#### **Altura**

Al igual que sucede con ancho, la propiedad altura no admite valores negativos. Si se indica un porcentaje, se toma como referencia la altura del elemento padre. Si el elemento padre no tiene una altura definida explícitamente, se asigna el valor auto a la altura.

El valor heredado indica que la altura del elemento se hereda de su elemento padre. El valor automático, que es el que se utiliza si no se establece altura de forma explícita un valor a esta propiedad, indica que el navegador debe calcular automáticamente el elemento, teniendo en cuenta sus contenidos y el sitio disponible en la página.

Altura = height

**En CSS:**

C:\Users\54381\Downloads\a3.jpg

**En HTML:**

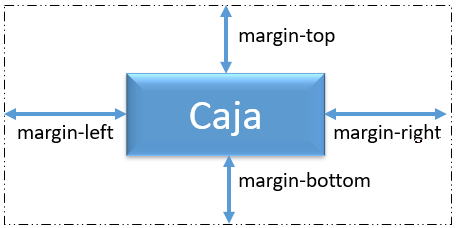
C:\Users\54381\Downloads\a4.jpg

Margen

CSS define cuatro propiedades para controlar cada uno de los márgenes horizontales y verticales de un elemento.

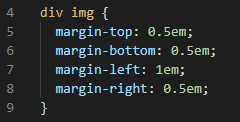
1. margin-top => Margen superior
2. margin-right => Margen derecho
3. margin-bottom => Margen inferior
4. margin-left => Margen izquierdo

Cada una de las propiedades establece la separación entre el borde lateral de la caja y el resto de cajas adyacentes:

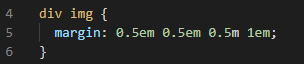


Los márgenes verticales (margin-top y margin-bottom) sólo se pueden aplicar a los elementos de bloque y las imágenes, mientras que los márgenes laterales (margin-left y margin-right) se pueden aplicar a cualquier elemento.

**Código CSS:**



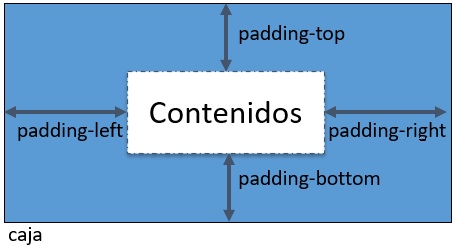
**Alternativa:**



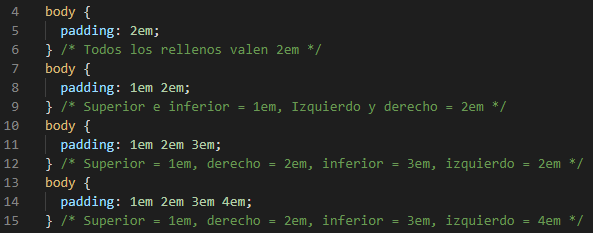
Relleno o acolchado

Relleno o acolchado. CSS define cuatro propiedades para controlar cada uno de los espacios de relleno horizontales y verticales de un elemento:

1. padding-top => Relleno superior
2. padding-right => Relleno derecho
3. padding-bottom => Relleno inferior
4. padding-left => Relleno izquierdo



**Algunos ejemplos de acolchado:**

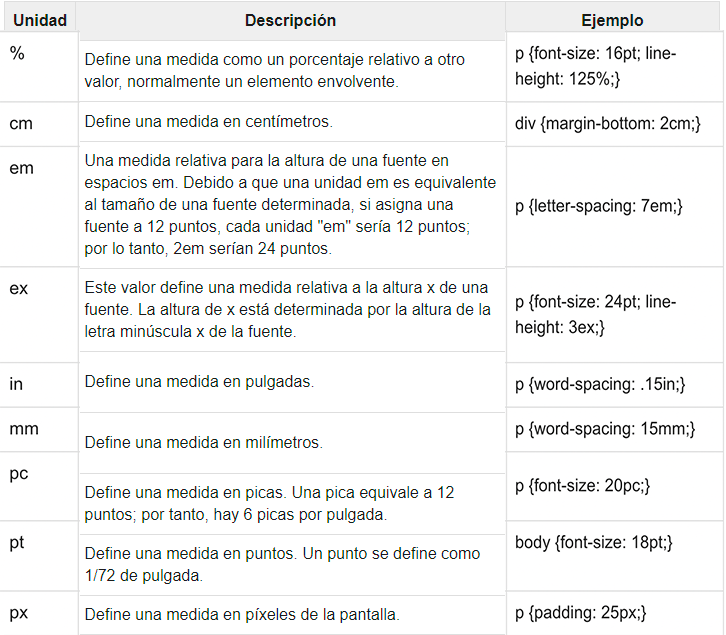


## Unidades de medida

Antes de comenzar el ejercicio real, nos gustaría dar una breve idea sobre las unidades de medida CSS.

CSS admite una serie de medidas que incluyen unidades absolutas como pulgadas, centímetros, puntos, etc., así como medidas relativas como porcentajes y unidades em. Necesita estos valores al especificar varias medidas en sus reglas de estilo, por ejemplo,  **border = "1px solid red"**  .

Hemos enumerado todas las unidades de medida CSS junto con los ejemplos adecuados:



## Colores

CSS usa valores de color para especificar un color. Por lo general, se utilizan para establecer un color para el primer plano de un elemento (es decir, su texto) o para el fondo del elemento. También se puede usar para afectar el color de los bordes y otros efectos decorativos.

Puede especificar sus valores de color en varios formatos. La siguiente tabla enumera todos los formatos posibles:

|  |  |  |
| --- | --- | --- |
| **Formato** | **Sintaxis** | **Ejemplo** |
| Código hexadecimal | #RRGGBB | p {color: #FF0000;} |
| Código hexadecimal corto | #RGB | p {color: #6A7;} |
| RGB % | rgb (rrr%, ggg%, bbb%) | p {color: rgb (50%, 50%, 50%);} |
| Absoluto RGB | rgb (rrr, ggg, bbb) | p {color: rbg (0,0,255);} |
| Palabra Clave | aquamarine, black | p {color: teal;} |

## Fuentes

Veamos cómo configurar las fuentes de un contenido, disponible en un elemento HTML. Puedes establecer las siguientes propiedades de fuente de un elemento:

* La propiedad **font-family** se utiliza para cambiar la cara de una fuente.
* La propiedad de **font-style** se usa para hacer una fuente en cursiva u oblicua.
* La propiedad **font-variant** se utiliza para crear un efecto de versalitas.
* La propiedad de **font-weight** se utiliza para aumentar o disminuir la negrita o la luz de una fuente.
* La propiedad de **font-size** se utiliza para aumentar o disminuir el tamaño de una fuente.
* La propiedad de **font** se utiliza como forma abreviada para especificar otras propiedades de fuente.

## Establecer la familia de fuentes

A continuación se muestra el ejemplo, que demuestra cómo configurar la familia de fuentes de un elemento. El valor posible podría ser cualquier nombre de familia de fuentes: