

## 2 >> Hojas de estilos CSS

Dado que en HTML no solo se indican los contenidos que se quieren visualizar, sino que se mezclan con las instrucciones de formato y estilo, el W3C creó un sistema mediante el cual las instrucciones del estilo y formato estuvieran separadas del contenido. Así nació CSS (*Cascading Style Sheets*, hojas de estilo en cascada.), un mecanismo para dotar de estilo a los elementos HTML, que desaconseja mezclar su contenido con la forma en que estos son presentados.

### 2.1 > Introducción

Por tanto, el objetivo fundamental de CSS es **separar el contenido y la estructura de un documento HTML, de su estilo y formato de presentación**. La versión recomendada era CSS 2.1, pero actualmente está en desarrollo el nuevo estándar CSS3, y el W3C recomienda el uso de los módulos CSS3 ya implementados.

Para entender el objetivo de CSS, pongamos un ejemplo: sabemos que el elemento de HTML `<h1>` indica que un bloque de texto es un encabezamiento y que es más importante (de mayor tamaño) que un bloque etiquetado como `<h2>`. Versiones más antiguas de HTML permitían atributos dentro de la etiqueta para añadir estilo extra (como el color o el tamaño de fuente). Pero cuando se utiliza CSS, la etiqueta `<h1>` no debería proporcionar información sobre cómo va a ser visualizado, solamente indicar la estructura del documento. La información de estilo, recogida por ejemplo en una hoja de estilos, debe especificar cómo se ha de mostrar `<h1>`, es decir, su color, fuente, alineación del texto, tamaño y otras características visuales.

Por consiguiente, CSS es la herramienta con la que se debe dar formato y estilo a los elementos de HTML. La información de estilo puede ser recogida tanto en un documento separado del HTML, opción recomendada, como en el propio documento HTML. En este último caso pueden definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo *style*, como más adelante se explica.

### 2.2 > Hoja de estilos

Una hoja de estilos es un conjunto de especificaciones que definen el formato con el que presentar los elementos HTML de una página web.

Está compuesta por **reglas de estilo**, cada una de las cuales declara los formatos que adoptarán los elementos de una página web. Así, un conjunto de reglas de estilo conforman una hoja de estilos, información que normalmente se recoge en un fichero con extensión `.css`, aunque existen otras maneras de recoger las reglas que den estilo a los elementos que componen un documento HTML.

Una regla de estilo tiene dos partes: el **selector**, que determina a qué elementos se asignará el estilo definido en la regla, y la **declaración**, que está compuesta por uno o más pares (*propiedad:valor*), que asignan un valor a la propiedad de los elementos seleccionados por el selector.



Estructura de una regla de estilo.

## 2.3 > Formas de aplicar estilo

CSS proporciona tres formas diferentes para aplicar las reglas de estilo a los elementos HTML de una página web:

### Estilo local o en línea

Este método permite insertar el estilo, directamente, dentro de cada etiqueta HTML, añadiendo en ellas el atributo común *style*.

Es la forma menos adecuada, dado que especificar el estilo dentro de cada etiqueta del documento HTML, es contrario al objetivo de CSS, y además se convierte en una tarea larga, tediosa y poco elegante de expresar el formato y estilo de una página web. Solo se recomienda usar este modo de trabajo, de manera excepcional, por ejemplo cuando se necesita aplicar un estilo o formato de manera urgente en una página.

### Hoja de estilos interna

Es una hoja de estilo que está incrustada (etiqueta `<style>`) dentro de un documento HTML. Suele ponerse dentro del elemento `<head>`, aunque puede ir en cualquier parte de la página. Esta forma permite agrupar toda la información sobre el estilo, pero dentro del mismo fichero HTML que contiene el contenido. Una desventaja es que hay que modificar todas las páginas que conforman un sitio cuando se quiera realizar un cambio que afecte a todas ellas; por lo tanto, será útil cuando se aplique estilo a una única página.

### Hoja de estilos externa

Es una hoja de estilo que está almacenada en un archivo diferente al archivo donde se almacena el código HTML de la página web. Esta es la mejor manera de aplicar estilo y la que sigue más fielmente el objetivo de CSS, porque separa completamente las reglas de estilo y formato para la página HTML, de su contenido y estructuración básica.

La etiqueta `<link>` se coloca en la cabecera `<head>` del documento y permite enlazarlo con otro recurso.

Se debe indicar el atributo *rel*, para especificar la relación entre el documento y el recurso.

El atributo opcional *type* especifica el tipo del documento a vincular, para una hoja de estilos *“text/css”*.

El atributo *href* debe indicar una URL, la del fichero CSS que tenga la definición del estilo.

Es recomendable utilizar hojas de estilo externas, pero sobre todo en sitios web que utilizan un mismo estilo para todas sus páginas, dado que una modificación de una regla de estilo afectará a todas ellas.

### Uso de hoja de estilo externa `<link>`

```
<html>
<head>
...
<link
rel="stylesheet"
type="text/css"
href="hojaEstilo.css"
...
</head>
<body>
...
</body>
</html>
```

### Comentarios

Cuando se crea una hoja de estilos (interna o externa), para facilitar la revisión del código CSS es conveniente añadir comentarios entre las reglas de estilo. Para incluir un comentario es necesario incluir el texto a comentar entre los caracteres */\* y \*/*, por ejemplo:

```
/* Esta es la forma de incluir
un comentario válido entre las
reglas de estilo CSS */
```

## Actividades propuestas

4.. Indica una etiqueta HTML que permita vincular la hoja de estilos del fichero *“miHoja.css”*.

### Alto y ancho de objetos CSS

Las propiedades generales *height* y *width* permiten establecer el alto y el ancho de los elementos.

```
#tamaño100
{height:100px;
width:100px;}
```

## 2.4 > Conceptos fundamentales

Los conceptos fundamentales que permiten realizar hojas de estilos en CSS son:

### Identificadores

Todos los elementos HTML tienen un atributo opcional llamado identificador (*id*), cuya finalidad es poder identificar a cada uno de los diferentes elementos que componen la página web. Con estos identificadores podemos aplicar un estilo exclusivo a cada etiqueta, según se seleccione su identificador en el selector de una regla CSS.

Para identificar un determinado elemento en el código HTML, es necesario darle un nombre por medio del atributo *id*, y, posteriormente, será necesario acceder a este elemento desde una regla CSS para aplicarle el diseño. Esto se hace a través del selector, que se formará con el nombre de la etiqueta seguido del símbolo # y del nombre del atributo *id*.

### Clases

Existe otro atributo que se puede utilizar en todos los elementos HTML, el atributo *class*. Su utilidad es la de poder agrupar los elementos por clases o grupos para que, posteriormente, CSS los pueda seleccionar y distinguirlos de los demás. El uso de *class* también es opcional, pero, como el atributo *id*, es muy recomendable y permite precisar selectores.

## Ejemplos

### Aplicar estilo de manera local, con una hoja de estilos interna y a través de clases

Este es un ejemplo de código HTML que utiliza una **hoja de estilo local** al párrafo, haciendo uso del atributo *style*:

```
<html> <body> <p style="color:green"> HOLA MUNDO </p>
</body> </html>
```



Este ejemplo utiliza una **hoja de estilos interna**, a través de la etiqueta *<style>*:

```
<html> <head> <title>Mi primera página con estilo</title>
  <style type="text/css">
    body {color: purple;
    background-color: #d8da3d}</style>
</head> <body>
```



En este otro ejemplo utiliza **clases** para aplicar estilo, a través del atributo *class*; su sintaxis en HTML sería:

```
<p class="bordeRojo">HOLA</p>
```

Un elemento puede pertenecer a varias clases a la vez, ya que estas no son excluyentes. Para ello se especifica dentro del atributo *class* los nombres de las clases separados por espacios:

```
<p class="letraAzul bordeAzul">HOLA</p>
```

Si un elemento HTML ha sido identificado siguiendo este ejemplo, puedes darle formato con reglas de estilo así:

```
.bordeRojo {
  border:solid 1px #000000;
  border-color:red;}
.bordeAzul {
  border:solid 1px #000000;
  border-color:blue;}
.letraAzul {color:blue;}
```

## Casos prácticos

4

### Creación de estilo usando identificadores

•• Crea un estilo, poniendo en negrita el primer párrafo, utilizando el atributo *id* dentro del elemento `<p>`.

**Solución** •• Para probar cómo dar estilo a elementos, abre un editor de texto (Bloc de notas, Gedit o similar) y edita dos documentos, uno que incluya las líneas de "Código HTML", y el otro con el "Código CSS" indicado. Hay que identificar con un nombre a la etiqueta `<p>` del primer párrafo y añadir un identificador al selector del código CSS.

#### Código HTML:

```
<link rel="stylesheet"
type="text/css"
href="negrita.css">...
<p id="EnNegrita">
Párrafo 1</p>
<p>Párrafo 2</p>
<p>Párrafo 3</p>...
```

#### Código CSS:

```
p#EnNegrita {font-weight:bold;}
```

Párrafo 1

Párrafo 2

Párrafo 3

Comprueba que incluyes el atributo *id* en el código HTML y utiliza su valor en el código CSS, incluyéndolo en el selector. Al ser un identificador, tienes que anteponerle la almohadilla al indicarlo en la regla de estilo. Así se puede dar un diseño exclusivo a cualquier elemento de una página web, tablas, listas, etc.; en este caso práctico se hace solo sobre el primer párrafo. Crea identificadores para los otros párrafos y aplícales reglas de estilo CSS.

### Pseudo-clases

Las pseudo-clases permiten clasificar los elementos HTML según el estado en el que se encuentren, es decir, los elementos serán seleccionados por una regla de estilo CSS en función de otros sucesos que hayan ocurrido, por ejemplo haber visitado un vínculo, o que el cursor del ratón se sitúe sobre un elemento, o en función del idioma especificado con el atributo común *lang*, en el elemento HTML.

El elemento `<a>` de HTML, utilizado para crear hiperenlaces, tiene definidas las siguientes pseudo-clases:

```
a:visited {color:blue;}
a:hover {color:blue;}
a:link {color:blue;}
a:focus {color:blue;}
a:active {color:blue;}
```

Vínculo visitado  
Vínculo donde está el ratón  
Vínculo no visitado  
Vínculo seleccionado por teclado  
Vínculo que recibe clic

La pseudo-clase para dar estilo a párrafos según el idioma sería así:

```
p:lang(en) {color:blue;}
```

Y la pseudo-clase para asignar un estilo al primer hijo de un elemento:

```
p:first-child {color:blue;}
```

### Selectores CSS3

Una de las diferencias en el uso de los selectores CSS3 es utilizar el símbolo (`::`), doble dos puntos, en lugar de los dos puntos (`:`) que se usaba para los pseudo-elementos en CSS 2.1. Algunos ejemplos son:

```
p::first-letter
{color:blue}
```

Aplica la propiedad a la primera letra de cada párrafo.

```
p::first-line {color:blue}
```

Aplica la propiedad a la primera línea de cada párrafo.

## Pseudo-elementos

Los pseudo-elementos son una parte de algún elemento HTML, que, aunque no tienen identidad propia, pueden ser identificados por las reglas de estilos y ser formateados de manera distinta a los elementos a los que pertenecen; algunos son: `::first-letter` y `::first-line`.

## 2.5 > Criterios de selección, selectores

Con los criterios de selección podemos indicar cuál es el selector de una regla de estilo CSS y así expresar a qué elementos de HTML se le aplicará el valor sobre la propiedad indicada en dicha regla CSS. Los diferentes criterios establecen, por lo tanto, distintos tipos de selectores:

### Selector universal

Si se utiliza el asterisco (\*) como selector, las propiedades especificadas afectarán a todos los elementos de la página; el asterisco es el selector universal. Para declarar el color azul a todos los elementos sería:

```
* {color:blue}
```

### Selector de tipo o nombre de elemento

Permite aplicar un determinado estilo a todos los elementos del tipo seleccionado, con independencia de posibles clases, identificadores, etc. Para poder afinar más la selección se deben utilizar otros criterios de selección. Un ejemplo que declara el color azul a todos los párrafos sería:

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

### Selector por combinación de elementos

Este criterio permite seleccionar en función de los elementos ascendientes/descendientes, padres/hijos o hermanos del elemento al cual se quiere dar un estilo. Estos elementos se pueden definir como:

- **Ascendientes/Descendientes:** ascendiente es el elemento que contiene al que se desea formatear, el descendiente. Puede haber más de un elemento entre uno y otro. Si B es un descendiente arbitrario de algún elemento antepasado A, entonces A se considera ascendiente de B. Para seleccionar elementos en función del ascendiente, se usan dos selectores separados por un espacio ( ), siendo el primero de ellos el ascendiente y el segundo el descendiente. Por ejemplo, si se quiere formatear de color rojo los elementos *p* que estén dentro de un elemento *div*, se debe escribir la siguiente regla:

```
div p {color : #00FF00}
```

- **Padres/Hijos:** padre es el elemento que contiene directamente al que se desea formatear, al que llamamos hijo. Para seleccionar elementos en función del padre, se usan dos selectores separados por el símbolo “mayor que” (>), siendo el primero el padre y el segundo el hijo. Por ejemplo, con la siguiente declaración solo se verían afectados los elementos *p* que estuvieran directamente dentro de un elemento *div*, es decir, los elementos *p* que son hijos de *div*, los cuales serán seleccionados para ponerles el color verde; sería así:

```
div > p {color : #00FF00;}
```

– **Hermanos:** hay dos combinaciones de hermanos, la combinación de hermanos adyacentes y la combinación de hermanos en general.

- **Hermanos adyacentes**, son elementos que se sitúan de forma que uno precede inmediatamente al otro, perteneciendo ambos al mismo elemento padre.

Para seleccionar elementos en función de hermanos adyacentes, se usan dos selectores separados por el símbolo suma (+), el primero de ellos representa el hermano precedente y el segundo el hermano al cual se quiere dar el formato. Por ejemplo, para seleccionar y poner en color verde el contenido de un elemento *pre*, que es hermano adyacente de un párrafo *p*:

```
p + pre {color : #00FF00;}
```

- **Hermanos en general**, son elementos que comparten el mismo elemento padre. Para indicar un selector en función de hermanos en general, se usan dos selectores separados por el símbolo de tilde (~), siendo el primero el elemento que precede (no necesariamente inmediatamente) al segundo selector, que es al que se quiere dar formato. Por ejemplo, para seleccionar un elemento de tipo *pre*, que es hermano de un elemento *h1*, sin serlo de manera adyacente, y ponerlo con color verde:

```
h1 ~ pre {color : #00FF00;}
```

### Selector por clase o identificador de elemento

Forma de selección muy utilizada y de gran precisión, que se utiliza para dar formato y estilo solo a aquellos elementos que se hayan etiquetado con una clase (atributo *class*) o con un identificador (atributo *id*) concreto.

Para ello se escribirá el nombre del elemento seguido de un separador (el punto para las clases y la almohadilla para los identificadores), y detrás el nombre de la clase o del identificador correspondiente.

Si al usar un selector por clase o *id*, se omite el nombre del elemento, entonces se verán afectados todos los elementos que pertenezcan a dicha clase o todos los elementos con ese identificador. Por ejemplo, para alinear a la izquierda todos los elementos de la clase *Clase\_Primer*a y a los que tienen como identificador *Nombre\_Identificador*:

```
.Clase_Primer {text-align:left;}
#Nombre_Identificador {text-align:left;}
```

Para dar estilo a los párrafos de la clase “uno”, y a los que tengan el identificador “dos”, definidos en un documento HTML, se tienen que escribir las reglas CSS:

```
p.uno {text-align:#00FF00;}
p#dos {color : #00FF00;}
```

### Selector por pseudo-clase del elemento

Las pseudos-clases clasifican elementos basándose en el estado en el cual se encuentren los mismos. Se utilizan a menudo con los hiperenlaces.

#### Elementos Hermanos

```
<h1>Título</h1>
<p>Párrafo</p>
<pre>Elemento seleccionado
para dar formato</pre>
```

Así, en función de si, por ejemplo, el enlace ha sido visitado o no, si el usuario está pasando por encima con el cursor, poder darles diferente estilo. Por ejemplo: `a:link {color:red;}`

### Selector por atributo o valor del elemento

Se utiliza cuando se pretende aplicar estilo a elementos que tienen un determinado atributo o un valor de atributo dado. La forma de declararlo es *elemento[atributo]*. Por ejemplo, para seleccionar las celdas de una tabla (*td*) que tengan definido el atributo *fgcolor*, cualquiera que sea el valor del atributo se escribiría: `td[colspan]{color:#007777}`

Otros modos de uso del selector por atributo o valor del elemento son:

**[att = valor]** Representa un elemento con el atributo *att*, cuyo valor es exactamente “valor”.

**[att ~ = valor]** Representa un elemento con el atributo *att*, cuyo valor es una lista de palabras separadas por espacios en blanco, una de las cuales es “valor”.

**[att | = valor]** Representa un elemento con el atributo *att*, que tiene su valor exactamente como “valor” o que comienzan con “valor” e inmediatamente le sigue un guión (“-”).

## 2.6 > Unión y combinación de selectores

Con el objetivo de afinar en la definición de los selectores, a veces es conveniente realizar agrupaciones o combinaciones de selectores.

### Unión de selectores

Cuando sea necesario aplicar las mismas reglas de estilo a varios elementos, se pueden reutilizar declaraciones, agrupando selectores, en lugar de repetir las mismas reglas de estilo con diferentes selectores. Para ello se separan los selectores con una coma (.). Por ejemplo, si se quiere formatear los elementos *ul* y *ol* con las mismas propiedades, se puede escribir la siguiente regla: `ul, ol {color:#00FF00;}`

### Combinación de selectores

Los criterios de selección desarrollados en el punto anterior pueden combinarse para realizar una selección más concreta. Un ejemplo, que permite formatear a todos los elementos *li* que estén dentro de elementos *ul*, que pertenezcan a la clase *datos* y que tengan el atributo *id*, es esta combinación de selectores: `ul li.datos[id] {color:#00FF00;}`

## 2.7 > Elementos HTML <div> y <span>

Los elementos de HTML `<span>` y `<div>`, junto con los atributos *class* e *id*, permiten seleccionar o agrupar elementos de un documento HTML y así precisar a qué elementos se quieren asignar las reglas de estilo CSS. Ambos son elementos que no implican ningún cambio de estilo por sí mismos, y tampoco añaden contenido al HTML, sino que sirven para ayudar a organizar su estructura y servir como selectores en reglas CSS.



## Agrupación con <span>

El elemento **<span>** permite definir una especie de gancho, una parte de un texto, para añadirle posteriormente un estilo diferente:

### Casos prácticos

5

#### Creación de estilo usando la etiqueta <span>

•• Dale estilo a un párrafo destacando determinadas palabras, haciendo que aparezcan enfatizadas con colores.

**Solución** •• Crea un nuevo documento, *span.html*, e incluye en el cuerpo un párrafo que contenga la cita de Benjamín Franklin donde considera las ventajas de madrugar, tal y como puedes leer en el siguiente párrafo del documento:

```
<p>El que pronto se acuesta y pronto se levanta, es hombre saludable, rico y sabio.</p>
```

Modifica el párrafo incluyendo cada una de las ventajas dentro de un elemento `<span>` `</span>`, así:

```
<p>El que pronto se acuesta y pronto se levanta, es hombre <span> saludable </span>, <span> rico </span> y <span> sabio </span>.</p>
```

A continuación, añade el atributo **class** a cada elemento `<span>`, asignándole en los tres elementos el mismo valor; por ejemplo, el valor *“verde”* con los respectivos atributos, de la siguiente manera:

```
<p>El que pronto se acuesta y pronto se levanta, es hombre <span class = "verde"> saludable </span>, <span class = "verde"> rico </span> y <span class = "verde"> sabio </span>.</p>
```

Ahora crea un documento CSS, denominado *reglaSpan.css* que contenga esta sencilla regla de estilo CSS; con ella el selector permite seleccionar todos los elementos `<span>` que pertenezcan a la clase *“verde”*.

```
span.verde {color:green;}
```

Añade una cabecera al documento HTML, que contenga una línea de código, para enlazar con la hoja de estilos externa recién creada:

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

El efecto producido es este: **El que pronto se acuesta y pronto se levanta, es hombre saludable, rico y sabio.**

Consigue ahora que el color asignado sea diferente para cada una de estas tres ventajas. Para ello, modifica cada elemento `<span>`, elimina los atributos de tipo *class*, y, posteriormente, añade un atributo *id* a cada uno de ellos, asignándoles diferente valor de identificador; deja el párrafo así:

```
<p>El que pronto se acuesta y pronto se levanta, es hombre <span id="EnVerde"> saludable </span>, <span id="EnRojo"> rico </span> y <span id="EnAzul"> sabio </span>.</p>
```

Modifica ahora el documento CSS denominado *reglaSpan.css* y añade estas reglas de estilo CSS, que tienen como selectores los nombres de los identificadores, para conseguir el efecto y asignar el color deseado:

```
span#EnVerde {color:green;} span#EnRojo {color:red;} span#EnAzul {color:blue;}
```

**El que pronto se acuesta y pronto se levanta, es hombre saludable, rico y sabio.**



### Agrupación con <div>

El elemento **<div>** permite crear bloques, capas o cajas, es decir, una agrupación de elementos para darles a todos ellos el mismo estilo y con ello poder estructurar una página web con estas divisiones.

Los elementos **<div>** y **<span>** sirven para realizar cosas muy sencillas, como, por ejemplo, dar estilo a los textos y colores. Pero ambos elementos tienen un gran potencial para realizar cosas mucho más avanzadas, sobre todo la etiqueta **<div>**, que se utiliza a menudo para crear efectos visuales en la información contenida, a modo de cajas que agrupan los elementos, para ser representadas en un lugar concreto que interese para facilitar la navegación del usuario de la página web.

## Ejemplos

### Diferenciar elementos por división (etiqueta <div>) con una hoja de estilo externa

Supongamos un fichero HTML que contenga estas dos listas de presidentes de los Estados Unidos, una para cada afiliación política mayoritaria, los demócratas y los republicanos, agrupadas ambas con **<div>**.

```
<div id="democratas">
<ul>
<li>Franklin D. Roosevelt</li>
<li>Harry S. Truman</li>
<li>John F. Kennedy</li>
<li>Lyndon B. Johnson</li>
<li>Jimmy Carter</li>
<li>Bill Clinton</li>
</ul>
</div>

<div id="republicanos">
<ul>
<li>Dwight D. Eisenhower</li>
<li>Richard Nixon</li>
<li>Gerald Ford</li>
<li>Ronald Reagan</li>
<li>George Bush</li>
<li>George W. Bush</li>
</ul>
</div>
```

El ejemplo mostrará cada lista con un color diferente, **azul** para la lista de presidentes demócratas y **rojo** para representar la lista de los presidentes republicanos.

Para ello se recomienda definir el estilo con una hoja de estilos externa y utilizar las agrupaciones realizadas con **<div>** y sus respectivos identificadores, en este ejemplo *"democratas"* y *"republicanos"*. Estas son las dos reglas de estilo CSS necesarias en este ejemplo:

```
#democratas {background:blue;}
#republicanos {background:red;}
```

Así será la visualización en el navegador; se consigue resaltar, con fondo en color azul, a los presidentes demócratas y, en color rojo, a los presidentes republicanos.

- Franklin D. Roosevelt
- Harry S. Truman
- John F. Kennedy
- Lyndon B. Johnson
- Jimmy Carter
- Bill Clinton

- Dwight D. Eisenhower
- Richard Nixon
- Gerald Ford
- Ronald Reagan
- George Bush
- George W. Bush

## 2.8 > Colores, fondos, textos y fuentes

Para dar estilo a un documento HTML existen muchas propiedades CSS. Estas son algunas de las que permiten establecer colores y tipos de letra:

Propiedad	Utilidad	Valor	Declaración
color	Color de texto	Hexadecimal   RGB   RGBA   HSL   HSLA   Predefinido   inherit	color:#ff0000; color:rgb(255,0,0); color:Cyan;
background-color	Color de fondo	color   transparent   inherit	background-color:#00ff00; background-color:rgb(255,0,255) background-color:yellow;
background-image	Imagen de fondo	url('URL')	background-image: url('paper.gif');
text-align	Justificar texto	left   right   center   justify   inherit	text-align:center
text-decoration	Decorar texto	none   underline   overline   blink   line-through   inherit	text-decoration:underline
text-indent	Indentar o sangrar texto (margen)	longitud   %   inherit	text-indent
font-family	Tipo de fuente	family-name   generic-family	font-family: 'Arial', 'Times-New-Roman', sans-serif;
font-size	Tamaño de fuente	xx-small   x-small   small   medium   large   x-large   xx-large   smaller larger   unidad de medida	font-size:1cm; font-size:12px; font-size:80%;
font-weight	Grueso de los caracteres de texto	normal   bold   bolder   lighter   100   200   300   400   500   600   700 800   900	font-weight:normal;
font-style	Estilo de fuente	normal   italic   oblique	font-style:normal; font-style:italic;
font-size-adjust	Ajusta el tamaño, con independencia del tipo de fuente de un texto	none   número	font-size-adjust:0.5; font-size-adjust:none;
<b>inherit:</b> indica que el valor de la propiedad será el heredado del elemento padre.			
<b>unidad de medida:</b> pixels (px), centímetros (cm), milímetros (mm), pulgadas (in), picas (pc) o porcentaje (%) sobre el tamaño del padre.			

### Actividades propuestas

**5••** Escribe la etiqueta `<h1>SALUDOS CSS</h1>` y aplica localmente el estilo necesario para que salga en rojo y con fondo amarillo. Haz lo mismo con una hoja de estilos interna y añade una imagen de fondo al documento.

**6••** Crea un documento HTML que contenga un párrafo de texto, y enlázalo a una nueva hoja de estilos que incluya en ella los ejemplos contenidos en la columna DECLARACIÓN de la tabla superior.

## 2.9 > Bloques, capas o cajas

Sobre los elementos HTML se pueden definir los bloques o capas, también denominadas cajas. Son muy utilizadas con la etiqueta `<div>`, y permiten representar con el mismo estilo a los elementos agrupados, según las propiedades específicas relacionadas con las cajas. Algunas de estas propiedades se recogen en la siguiente tabla:

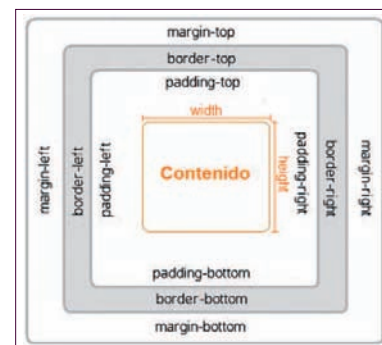
Propiedad	Utilidad	Valor	Declaración
<b>border</b>	<b>Borde, ancho, estilo y color</b> de la caja	border-width border-style color	border:5px solid red;
border-top	Borde superior	border-width border-style color	border-top:thick double #ff0000;
border-right	Borde derecho	border-width border-style color	border-right:medium solid #00ff00;
border-width	Propiedades del <b>tamaño del ancho</b> de la caja. Según el número de valores utilizados se aplica a diferentes zonas del margen	top right bottom left top (right and left) bottom (top and bottom ) (right and left) (top right bottom left)	border-width:thin 10px thick 10px; border-width:thin medium thick; border-width:thin medium; border-width:thin;
border-style	Propiedades del <b>estilo</b> de la caja. Se aplica según el número de valores utilizados	top right bottom left	border-style:dotted solid double dashed;
border-color	Propiedades del <b>color</b> de la caja. Se aplica según el número de valores utilizados	top right bottom left	border-color:red green blue pink;
<b>margin</b>	Propiedades del <b>tamaño del margen</b> de la caja. Según el número de valores utilizados se aplica a diferentes zonas del margen	top right bottom left top (right and left) bottom (top and bottom ) (right and left) (top right bottom left)	margin:10px 5px 15px 20px; margin:10px 5px 15px; margin:10px 5px; margin:10px;
margin-bottom	Margen inferior	unidad de medida   inherit	margin-bottom:1cm;
margin-left	Margen izquierdo	unidad de medida   inherit	margin-left:2cm;
<b>padding</b>	Propiedades del <b>tamaño del relleno</b> de la caja. Según el número de valores utilizados se aplica a diferentes zonas del margen	top right bottom left top (right-left) bottom (top-bottom ) (right-left) (top-right-bottom-left)	padding:10px 5px 15px 20px; padding:10px 5px 15px; padding:10px 5px; padding:10px;

## Actividades propuestas

**7••** Crea cajas con elementos `<div>` que agrupen elementos HTML básicos, e incluye en una hoja de estilos las declaraciones de ejemplo de la columna DECLARACIÓN de la tabla anterior, y pruébalas sobre las cajas creadas.

## 2.10 > Tamaño y posicionamiento

Existen propiedades CSS que permiten establecer la posición de los diferentes elementos de contenido de un fichero HTML. El método de posicionamiento de elementos se establece con la propiedad *position*, siendo su valor por defecto *static*, que muestra los elementos en el mismo orden en el que aparecen en el documento HTML, sin afectarle otras propiedades de posicionamiento. Con el valor *absolute*, los elementos se sitúan en la posición indicada, en las propiedades *top*, *bottom*, *left* y *right*, con respecto a su primer elemento antecesor que tenga un posicionamiento diferente a *static*. Con el valor *fixed*, los elementos se sitúan en una posición indicada con respecto a la ventana del navegador, unos encima de otros. Con el valor *relative*, los elementos se sitúan en una posición (admite valores negativos) relativa a su posición normal por defecto (*static*). Se resumen en la tabla la propiedad *position* y otras propiedades utilizadas:



Propiedades CSS para cajas con div.

Propiedad	Utilidad	Valor	Declaración
<i>position</i>	Método de posicionamiento de los elementos HTML en la ventana del navegador	<i>static</i>   <i>absotute</i>   <i>fixed</i>   <i>relative</i>   <i>inherit</i>	<i>position: absolute;</i>
<i>top</i>	Distancia de la parte superior de un elemento respecto a su posición normal. Si <i>position</i> es " <i>static</i> ", esta propiedad no tiene efecto	unidad de medida   <i>inherit</i>	<i>top: 5px;</i>
<i>right</i>	Distancia de la parte derecha de un elemento respecto a su posición normal. Si <i>position</i> es " <i>static</i> ", esta propiedad no tiene efecto	unidad de medida   <i>inherit</i>	<i>right: 5px;</i>
<i>bottom</i>	Distancia de la parte inferior de un elemento respecto a su posición normal. Si <i>position</i> es " <i>static</i> ", esta propiedad no tiene efecto	unidad de medida   <i>inherit</i>	<i>botton: 5px;</i>
<i>left</i>	Distancia de la parte izquierda de un elemento respecto a su posición normal. Si <i>position</i> es " <i>static</i> ", esta propiedad no tiene efecto	unidad de medida   <i>inherit</i>	<i>left: 20px;</i>
<i>float</i>	Modifica el posicionamiento normal desplazando el elemento hacia el lado especificado. Permite colocar texto alrededor de imágenes. Si <i>position</i> es " <i>absolute</i> ", esta propiedad no tiene efecto	<i>left</i>   <i>right</i>   <i>none</i>   <i>inherit</i>	<i>float: left;</i>
<i>clear</i>	Indica los lados de un elemento donde otros elementos flotantes no están permitidos	<i>right</i>   <i>left</i>   <i>both</i>   <i>none</i>   <i>inherit</i>	<i>clear: both;</i>
<i>display</i>	Define cómo mostrar un determinado elemento, el tipo de caja	<i>inline</i>   <i>block</i>   <i>list-item</i>   <i>none</i>	<i>display: block;</i>
<i>z-index</i>	Nivel de profundidad en la colocación superpuesta de elementos. Cuanto mayor sea el valor indicado más arriba aparecerá el elemento	<i>valor</i>	<i>z-index: 2;</i>
<i>visibility</i>	Especifica si el elemento es visible o no. El valor predeterminado es visible.	<i>visible</i>   <i>hidden</i>   <i>collapse</i>	

## Casos prácticos

6

Hacer flotar una imagen respecto a los párrafos con *float* y *clear*

•• Poner una imagen flotando a la izquierda de un párrafo, y hacer que otro párrafo no lo permita.

**Solución** •• Crea el siguiente documento HTML que expresa dos reglas de estilo para practicar el uso de las propiedades *float* y *clear*. Observa que al elemento imagen se le aplica la propiedad *float*, para que flote a la izquierda respecto al siguiente elemento que le sigue, que en este caso es el primer párrafo; este párrafo permite que los elementos floten sobre cualquiera de sus lados, ya que no tiene definida la propiedad *clear*. Fíjate que al segundo párrafo, que pertenece a la clase "clear", se le asigna esa misma propiedad *clear* y el valor *both*, para que no permita elementos flotantes a ninguno de sus lados, ni a su derecha ni a su izquierda.

```
<html>
<head>
  <style>
    img {float:left;}
    p.clear {clear:both;}
  </style>
</head>
<body>
  
  <p>Este es un párrafo que permite que el resto de elementos floten sobre él.
  Este es un párrafo que permite que el resto de elementos floten sobre él.</p>

  <p class="clear">Este párrafo NO permite que el resto de elementos floten sobre
  él. Este párrafo NO permite que el resto de elementos floten sobre él.</p>
</body>
</html>
```

Guarda este fichero con el nombre float.html y ábrelo desde el navegador. Observarás el siguiente resultado:



Este es un párrafo que permite que el resto de elementos floten sobre él. Este es un párrafo que permite que el resto de elementos floten sobre él.

Este párrafo NO permite que el resto de elementos floten sobre él. Este párrafo NO permite que el resto de elementos floten sobre él.

Modifica las propiedades *float* y *clear*, asignándoles los valores *right* y *none*, respectivamente, y comprueba los resultados obtenidos y cómo se posiciona la imagen respecto al segundo párrafo.

## Actividades propuestas

**8••** Modifica la hoja de estilos asociada de la actividad número 7, incluyendo nuevas reglas de estilo para cambiar los tamaños de las cajas creadas en esa actividad.

**9••** Añade reglas de estilo que modifiquen el posicionamiento de las cajas del ejercicio anterior.

**10••** Coloca superpuestas varias cajas de las creadas en los ejercicios anteriores, utilizando z-index en cada una de ellas.