1. Introducción

Las CSS (Cascading Style Sheet) u hojas de estilo en cascada, constituyen el lenguaje que se utiliza para controlar el aspecto de los documentos HTML. De manera que será necesario echar mano de nuestros conocimientos de este lenguaje de marcas, ya que, como se verá más adelante, se puede definir un estilo para una etiqueta HTML en concreto o en función de sus atributos.

El uso de hojas de estilo para el diseño Web, conlleva innumerables ventajas, que convergen fundamentalmente en la separación entre contenido y presentación, con lo que ganamos en:

- Mejores definición y significado de los documentos HTML.
- Mejor accesibilidad
- Simplificación de las tareas de mantenimiento
- Posibilidad de visualización del mismo documento, de distinta forma, dependiendo del dispositivo de salida: monitor, impresora, dispositivo móvil…

Obviamente, algunos de los elementos que vamos a poder modificar con las hojas de estilo son: color, bordes, imágenes de fondo, transparencia y tamaño, pero también se va a poder manipular la forma en que se disponen los distintos elementos en la pantalla del navegador, así como determinados efectos para textos, imágenes y cursores.

Para poder trabajar con CSS, hay que tener unos conocimientos básicos, que pasamos a describir en los siguientes apartados.

Fundamentalmente hay que conocer cómo se vinculan el documento .html y el .css, cuáles son los elementos constitutivos de las reglas CSS de las que se componen las hojas de estilo y por último veremos las propiedades que afectan a la disposición en pantalla, a la textura de los elementos, a la estructuración de la página y a la disposición de textos sobre todo con respecto a determinados elementos gráficos que les puedan acompañar.

2. Consideraciones previas

CSS y HTML

La forma en que se incluye CSS en un documento HTML varía en función de nuestros intereses y/o necesidades, sin embargo es preferible, siempre que sea posible, hacerlo mediante un archivo externo, con ello, cada vez que se modifique un estilo, aplicaremos el cambio a todas las páginas en las que se utilice ese estilo, de una sola vez.

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

No obstante podemos insertar estilos utilizando la etiqueta style dentro de la etiqueta <head> del documento HTML,

```
<head>

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

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

</style>
…
</head>
```

o aplicar un estilo en concreto a una etiqueta HTML utilizando el atributo style:

```
<p style="color:black; font-family:Verdana"> Esto es un texto con un estilo </p>
```

Con esto conseguimos aplicar un estilo determinado a un elemento html en concreto o a una etiqueta en particular, pero perdemos las ventajas, mencionadas anteriormente de separar contenido y diseño.

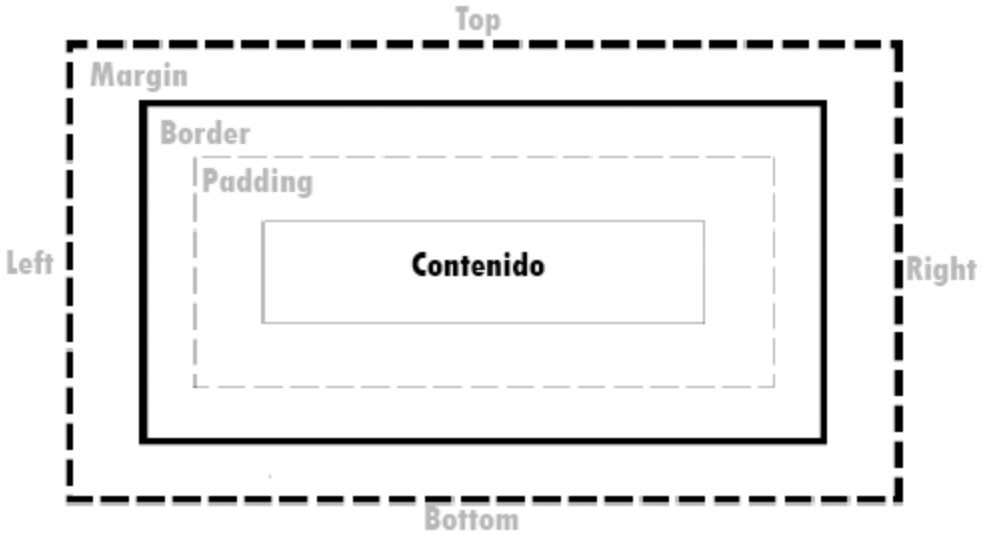
El modelo de cajas

Es sin duda el concepto fundamental del lenguaje CSS ya que, cada vez que incluimos una etiqueta HTML, estamos incluyendo una caja que abarca todo su contenido. Y será mediante las hojas de estilo como controlemos su aspecto.

Las partes de que se compone una caja son, de dentro hacia fuera:

- Contenido
- Relleno *(Padding)*
- Borde *(Border)*
- Imagen y/o color de fondo
- Margen *(Margin)*

Es importante recordar que la suma del contenido, el relleno, el borde y el margen dan como resultado el espacio total que ocupará un elemento en la ventana del navegador, y no sólo el alto y/o ancho que le apliquemos.



Herencia

Es otro de los conceptos que hay que tener en cuenta a la hora de trabajar con hojas de estilo. Con él se indica que muchas de las características que demos a las etiquetas HTML, las tomarán también los elementos que se encuentren dentro de ellas.

Por ejemplo, y explicado muy por encima, aunque tendremos oportunidad de verlo más adelante, si indicamos a la etiqueta body que el texto debe tener un tamaño determinado, todos los elementos de la página mostrarán el texto que contengan en ese tamaño de letra, a no ser que especifiquemos lo contrario.

CSS y los navegadores

Finalmente, hay que tener en cuenta que, desgraciadamente, no hay consenso entre los distintos navegadores a la hora de interpretar las reglas CSS.

Pese a la especificación oficial de CSS, que se puede consultar en www.3w.org ,el navegador más utilizado, IExplorer en sus distintas versiones, no tiene buen soporte de CSS, siendo uno de los que mejor interpretación hacen de ellas Firefox.

Esto dificulta enormemente la creación de un diseño homogéneo para todos ellos, incluso para las diferentes versiones de IExplorer, dado que la implementación de CSS varía en cada una de ellas. Para intentar controlar en mayor medida estas situaciones, lo que se suele hacer es utilizar *hacks* (diferentes selectores para una misma visualización, en los distintos navegadores y/o versiones).

Medios

Es muy importante, también, tener en cuenta que con CSS, vamos a poder definir distintos estilos dependiendo del dispositivo de salida.

Así, con la palabra reservada handheld, definiremos las reglas para dispositivos móviles como teléfonos, PDA's... y con print haremos crear las reglas para un documento html cuando vaya a ser impreso.

En este sentido hay que tener en cuenta, por un lado que la mayoría de los usuarios, aún no dispone de dispositivos móviles con tarjetas de vídeo similares a los PC's, con lo que habrá que tener en cuenta que para esos casos es preferible utilizar una paleta de colores segura para la web. Por otra parte, cuando un usuario quiere imprimir una página web, lo más probable es que quiera imprimir sólo la información que ésta contiene y no sus imágenes ni su disposición para ser vista en pantalla.

Los métodos para definir las reglas que se van a aplicar dependiendo del método son varios.

Con reglas del tipo @media:

```
@media print {
body { font-size: 10pt }
}
```

Con reglas del tipo @import:

```
@import url("estilos_impresora.css") print;
```

Con la etiqueta link

```
<link rel="stylesheet" type="text/css" media="print" href="especial.css" />
```

Sintaxis

Cada uno de los estilos que forman una CSS se llama “Regla”. Las reglas, están compuestas por un selector y una apertura y cierre de llave dentro de las cuales está lo que llamamos declaración.

```
body{
        font-size: 1em;
}
```

La forma en que indicamos qué elemento o elementos del documento HTML se verán afectados por una regla es utilizando “Selectores”. Éstos representan a los elementos HTML, body en el ejemplo, a los que se les aplicarán las propiedades definidas, con su correspondiente valor, entre las llaves de apertura y cierre.

El formato que se aplicará va dentro de las llaves de apertura y cierre en la regla en lo que se llama “Declaración” y se compone de “Propiedad”, es decir, de la característica del elemento que se modificará, y de “Valor” o aspecto final que se da al elemento/s HTML seleccionado/s.

Gran parte de las propiedades que se pueden definir mediante CSS se refieren a los elementos anteriormente descritos en el modelo de cajas, pero también a la fuente empleada y a la disposición de las diferentes etiquetas cuando las visualicemos en el navegador.

En cuanto a los valores que esas propiedades van a tomar, son en su mayoría unidades de medida, normalmente se utilizan unidades relativas, ya que estas hacen que sea más fácil la adaptación a cualquier medio y/o dispositivo. Por regla general suelen utilizarse “píxel” y “porcentaje” para la estructura del sitio y “em” y “porcentaje” para el tamaño del texto (Con “em” se toma como medida de referencia el tamaño de la M mayúscula del tipo de letra definida por defecto).

Por último, no hay que olvidar que en CSS, como en cualquier otro lenguaje, podemos insertar comentarios utilizando los caracteres /*…*/

3.Selectores

Los selectores son los elementos de la regla CSS que indican a cual o cuales elementos del documento HTML se van a aplicar los valores especificados en las propiedades que en esa regla se definen.

Este apartado se centra en la explicación de este componente de las reglas CSS, por lo que sólo se exponen reglas sencillas y en algún caso código HTML aclaratorio, pero no el resultado visual de la aplicación del estilo.

Selectores básicos

- Selector universal (*) significa todos los elementos y suele utilizarse en combinación con otros selectores

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

- Selector de etiqueta. Selecciona todos los elementos del documento HTML que tengan esa etiqueta.

```
h1 {
        margin: 0;
        padding: 0;
        font: italic 197% "Times New Roman", Times, serif;
        color: #FFFFFF;
}
```

Se pueden encadenar para aplicar los mismos estilos a diferentes etiquetas

Por ejemplo,

```
h1 {
        color: #313131;
}
h2 {
        color: #313131;
}
h3 {
        color: #313131;
}
```

Es equivalente a:

```
h1, h2, h3 {
        color: #313131;
}
```

- Selector descendente. Selecciona los elementos que están dentro de otros.

```
p span {
        font-weight: bold;
}
```

Se pueden utilizar varios selectores descendentes seguidos:

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

El elemento al que se aplica el estilo siempre es el último selector indicado (em en el ejemplo anterior) y todos los selectores anteriores indican cuál es la ruta para que encontrar al elemento al que se le aplicará el estilo definido (p a span en el ejemplo anterior).

Si se utiliza en combinación con el selector universal, restringe el alcance del selector

```
p * a {
        color: #ff0000;
}
```

La regla anterior aplicada sobre el código que sigue:

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

Hace que solamente el segundo enlace se muestre en color rojo, ya que con esta combinación de selectores indicamos se visualicen en color rojo los vínculos que se encuentren dentro de cualquier elemento que esté dentro de un párrafo. Condición que sólo se cumple en el segundo caso.

- Selector de clase. Permite seleccionar todos los elementos cuyo atributo class HTML coincida con el del selector.

Este selector está formado por un signo de punto (.) y el nombre del atributo class que se quiere seleccionar. Por tanto, en el siguiente ejemplo, se mostrarán de color rojo el segundo párrafo y la etiqueta blockquote:

Regla:

```
.destacado {
        color: #ff0000;
}
```

Código HTML

```
<p>Ejemplo de párrafo sencillo</p>
<p class="destacado">Ejemplo de párrafo con atributo class</p>
<blockquote class="destacado">Texto de ejemplo</blockquote>
```

Este selector suele aparecer unido al selector de etiqueta para aplicar distintos estilos sólo a aquellas que tengan el atributo class que se indique. Si al código HTML anterior le modificamos el estilo que se aplica de la siguiente manera:

```
p.destacado {
        color: #ff0000;
}
```

Sólo se visualizará de color rojo el contenido de la etiqueta p y el estilo dejará de afectar al texto que hay en la etiqueta blockquote.

- Selector de ID. Permite seleccionar todos los elementos cuyo atributo id HTML coincida con el atributo id determinado en el selector.

```
#destacado {
        color: #ff0000;
}
```

```
<p>Primer párrafo</p>
<p id="destacado">Segundo párrafo</p>
<blockquote id="destacado">Texto de ejemplo</blockquote>
```

Exactamente igual que en el caso anterior podemos seleccionar sólo determinados elementos uniendo este selector al de etiqueta:

```
p#destacado {
        color: #ff0000;
}
```

Selectores avanzados

Ni el navegador IExplorer 6, ni sus versiones anteriores soportan estos selectores, por lo que su uso se aprovecha sobre todo para definir *hacks*.

Para facilitar esta tarea desde la siguiente URL. http://www.css3.info/selectors-test/ se puede realizar un test para averiguar los selectores que admite el navegador con el que se realiza.

- Selector de hijos. Un selector de hijo afecta a un elemento cuando es el hijo de otro. Un selector de hijo se compone de dos o más selectores separados por el signo ">".

La siguiente regla asigna el estilo de todos los párrafos que son hijos de body:

```
body > p {
        line-height: 1.3;
}
```

- Selector adyacente. El selector adyacente tiene la siguiente estructura:

elemento1 +elemento2

Donde elemento1 y elemento2 deben tener el mismo padre y además, elemento1 debe preceder inmediatamente a elemento2.

El siguiente ejemplo reduce el espacio vertical que separa un h1 y un h2 que lo sigue inmediatamente:

```
h1 + h2 {
        margin-top: -5mm;
}
```

- Selectores de atributos. Los selectores de atributos se pueden escribir de cuatro maneras:

[att] Se refiere a los elementos que tienen asignado el atributo att, cualquiera sea el valor del atributo.

[att=val] Se refiere a aquellos elementos cuyo valor de atributo att sea exactamente el valor señalado val.

[att~=val] Afecta a los elementos cuyo atributo att sea una lista de "palabras" separadas por espacios, una de las cuales es exactamente val. Si se usa este selector, las palabras en el valor no deben contener espacios (ya que ellas están separadas por espacios).

[att|=val] Selecciona los elementos cuando el valor de su atributo att es una lista de "palabras" separadas por guiones, comenzando con val. La correspondencia siempre comienza al principio del valor del atributo. etc. */

Los valores de los atributos deben ser identificadores o cadenas. La distinción entre mayúsculas/minúsculas en los nombres y valores de los atributos de los selectores depende del lenguaje del documento.

Por ejemplo, el siguiente selector de atributo equivale a todas las etiquetas acronym que especifican el atributo title, cualquiera que sea su valor:

```
acronym [title] {
    color: #0000ff;
    }
```

Aplicado al siguiente código HTML:

```
<acronym>Titular</acronym>
<acronym title="Extensible Markup Language">XML</acronym>
<acronym title="Société Nationale de Chemins de Fer">SNCF</acronym>
```

Afectaría solamente a los acrónimos segundo y tercero, puesto que tienen definido un atributo title.

Sin embargo, si modificásemos el estilo de la siguiente manera:

```
acronym [title= Extensible Markup Language] {
        color: #0000ff;
}
```

Sólo modificaríamos el color del primer acrónimo ya que es el único cuyo atributo title tiene el valor “Extensible Markup Language”.

Los selectores de atributos pueden utilizarse unidos para referirse a varios elementos de un atributo y ser más preciso a la hora de seleccionar un elemento en concreto.

En el ejemplo siguiente seleccionamos los span que tengan un atributo id en concreto y, demás su atributo class coincida con el especificado.

```
span[id="destacado"][class ="especial"]{
        color: #0000ff;
}
```

Los siguientes selectores ilustran la diferencia entre "=" y "~=". El primer selector equivale, por ejemplo, al valor "copyright copyleft copyeditor" para el atributo rel. El segundo selector sólo será equivalente cuando el atributo href tenga el valor "http://www.w3.org/".

```
a[rel~="copyright"]
a[href="http://www.w3.org/"]
```

La siguiente regla esconde a todos los elementos cuyo valor del atributo lang es "fr" (es decir, la lengua es el francés).

```
*[LANG=fr] {
        display: none;
}
```

La siguiente regla será equivalente a valores del atributo lang que empiecen con "en", incluyendo "en", "en-US" y "en-cockney":

```
*[LANG|="en"] {
        color: #ff0000;
  }
```

Pseudo-elementos y pseudo-clases

CSS introduce los conceptos de *pseudo-elementos* y *pseudo-clases* para permitir aplicar el formato basado en información que está fuera de la estructura del documento.

- Las pseudo-clases clasifican a los elementos en base a características más allá de su nombre, atributos o contenido, es decir, en, atributos que al menos en principio no pueden deducirse de la estructura del documento. Las pseudo-clases suelen ser dinámicas, en el sentido de que un elemento puede adquirir o perder una pseudo-clase a medida que el usuario interactúa con el documento. La excepción es ':first-child', que puede deducirse de la estructura del documento.

:first-child. La pseudo-clase :first-child equivale a un elemento que es el primer hijo de algún otro elemento.

En el ejemplo siguiente, el selector equivale a cualquier elemento p que sea el primer hijo de un elemento div.

```
div > p:first-child {
        text-indent: 0;
}
```

Las aplicaciones del usuario normalmente muestran los vínculos no visitados de un modo diferenciado de aquellos previamente visitados. CSS proporciona las pseudo-clases ':link' y ':visited' para distinguirlos.

:link. Se aplica a los vínculos que aún no han sido visitados.

:visited. Se aplica una vez que el vínculo ha sido visitado por el usuario.

Los dos estados son mutuamente excluyentes.

```
a:link {
        color: #ff0000;
}

a:visited {
        color: #0000ff;
}
```

Con estas reglas definimos el color para los vínculos y el color para los vínculos cuando el usuario los haya visitado.

:hover. Se aplica mientras el usuario pasa el puntero del ratón sobre un elemento.

:active. Se aplica mientras un elemento está siendo activado por el usuario. Por ejemplo, el lapso durante el cual el usuario presiona el botón del ratón y lo suelta.

:focus. Se aplica mientras un elemento tiene el foco (acepta eventos del teclado u otras formas de entrada de texto).

Estas pseudo-clases no son mutuamente excluyentes.

```
/* vínculos no visitados */

        a:link{
                color: # ff0000;
        }


/* vínculos visitados */

        a:visited{
                color: #0000ff;
        }

/* el usuario señala el vínculo */

        a:hover{
                color: #ffff00;
        }

/* vínculos activos */

        a:active{
                color: #00ff00;
        }
```

Es conveniente recordar que a:hover debe ir después que las reglas a:link y a:visited, de otro modo las reglas de cascada ocultarán la propiedad 'color' de la regla a:hover. También, debido a que a:active está ubicada después de a:hover, el color activo será aplicado cuando el usuario active y señale el elemento a.

- Los pseudo-elementos crean abstracciones acerca de la estructura del documento más allá de aquellas especificadas por el lenguaje del documento. Por ejemplo, los lenguajes de documento no ofrecen mecanismos para acceder a la primera letra o a la primera línea del contenido de un elemento. Los pseudo-elementos de CSS permiten hacer referencia a esta información inaccesible por otros medios. Los pseudoelementos también pueden utilizarse para asignar estilos a un elemento que no existe en el documento fuente

:first-line. El pseudo-elemento :first-line aplica estilos especiales a la primera línea de un párrafo. Por ejemplo:

```
p:first-line {
        text-transform: uppercase;
}
```

La regla anterior significa "convertir las letras de la primera línea de cada párrafo en mayúsculas".

:first-letter. El pseudo-elemento :first-letter puede ser usado para las "letras capitales"

```
p:first-letter {
        font-size: 200%;
        font-style: italic;
        font-weight: bold;
        float: left;
}
```

Las propiedades y valores necesarios para explicar este ejemplo requieren de la lectura del siguiente capítulo, por lo que se verá un caso práctico más adelante.

4.Propiedades

Propiedades para el tipo de fuente

NOMBRE	DESCRIPCION	VALORES
font-family	Tipo de letra	Nombre de la/s familia/s tipográfica/s en las que deba mostrarse el texto
font-size	Tamaño de letra	Unidad de medida o porcentaje
font-style	Estilo de letra	Normal o italic
line-height	Interlineado	Normal o unidad de medida
font-weight	Anchura de la letra	Normal, Bold o valor entre 100 y 900
text-transform	Transformación del texto	Capitalize, uppercase, lowercase, none, inherit

color	Color del texto	Valor hexadecimal

Disponemos también de la propiedad font que permite definir directamente todas las propiedades de tipografía de un texto. De la siguiente manera:

- Primero se indican font-style, font-variant y font-weight
- Luego el valor de font-size seguido de line-height
- Finalmente se señala el tipo de fuente

Propiedades para bloques de texto

NOMBRE	DESCRIPCION	VALORES
Text-align	Alineación del texto	Left, right, center, justify
Vertical-align	Alineación vertical	Baseline, sub, super, top, text-top, middle, bottom, text-bottom, porcentaje, unidad de medida
Text-indent	Tabula desde la izquierda la primera línea de un párrafo	Medida, porcentaje
Letter-spacing	Espacio entre letras	Unidad de medida
Word-spacing	Espacio entre palabras	Unidad de medida

Propiedades para el fondo de los distintos elementos

NOMBRE	DESCRIPCION	VALORES
background-color	Color de fondo	Valor hexadecimal
background-image	Imagen de fondo	URL de la imagen a cargar
background-repeat	Repetición de la imagen de fondo	Repeat, repeat-x, repeat-y, no-repeat
background-position	Posición de la imagen de fondo	Porcentaje, unidad de medida, left, center, right, top, bottom
background-attachment	Comportamiento de la imagen de fondo	Scroll, fixed

Propiedades de caja

a. Ancho y alto

NOMBRE	DESCRIPCION	VALORES
width	Establece la anchura de un elemento	Unidad de medida, porcentaje
height	Establece la altura de un elemento	Unidad de medida, porcentaje

b. Margen

NOMBRE	DESCRIPCION	VALORES
Margin--top	Margen superior	Unidad de medida, porcentaje
Margin-right	Margen derecho	Unidad de medida, porcentaje
Margin-bottom	Margen inferior	Unidad de medida, porcentaje
Margin-left	Margen izquierdo	Unidad de medida, porcentaje
Margin	Establece de forma directa todos los márgenes de un elemento	Unidad de medida, porcentaje

c. Relleno

El relleno (padding) de un elemento es el espacio que va desde su contenido hasta su borde.

NOMBRE	DESCRIPCION	VALORES
padding-top	Relleno superior	Unidad de medida, porcentaje
padding-right	Relleno derecho	Unidad de medida, porcentaje
padding -bottom	Relleno inferior	Unidad de medida, porcentaje
padding –left	Relleno izquierdo	Unidad de medida, porcentaje
padding	Establece de forma directa todos los rellenos de un elemento	Unidad de medida, porcentaje

d. Flotar y Borrar

NOMBRE	DESCRIPCION	VALORES
float	Tipo de posicionamiento	Left, right
clear	Permite limpiar sus lados para que no se muestren elementos adyacentes	Left, right, both

Propiedades para el borde

El borde sigue siendo una propiedad para elementos de caja, pero dada la cantidad de posibilidades que CSS aporta para esta propiedad, hemos preferido tratar de ella aparte.

Con CSS podemos modificar independientemente cada uno de los cuatro bordes de un elemento, y para cada uno de ellos es posible indicar anchura, color y estilo.

a. Ancho de borde

NOMBRE	DESCRIPCION	VALORES
border-top-width	Ancho del borde superior	Unidad de medida, thin, médium, thick
border -right-width	Ancho del borde derecho	
border -bottom-width	Ancho del borde inferior	
border -left-width	Ancho del borde izquierdo	
border-width	Establece de forma directa todos los anchos de borde de un elemento	

b. Color del borde

NOMBRE	DESCRIPCION	VALORES
border-top-color	Color del borde superior	Valor hexadecimal
border -right- color	Color del borde derecho	
border -bottom- color	Color del borde inferior	
border -left- color	Color del borde izquierdo	
border- color	Establece de forma directa los colores de todos los bordes de un elemento	

c. Estilo de borde

NOMBRE	DESCRIPCION	VALORES	
border-top-style	Estilo del borde superior	Dotted Dashed	Punteado Rayado
border -right- style	Estilo del borde derecho	Solid Doble	Sólido Doble
border -bottom- style	Estilo del borde inferior	Groove Ridge	Surco Rugoso
border -left- style	Estilo del borde izquierdo	outset	altorelieve
border- style	Establece de forma directa los estilos de todos los bordes de un elemento		

Propiedades para listas

NOMBRE	DESCRIPCION	VALORES
List-style-type	Permite establecer el tipo de viñeta mostrada para una lista	Disc, circle, decimal, upperroman…
List-style-position	Establece la posición de la viñeta	Inside, outside
List-style- image	Reemplaza las viñetas por una imagen personalizada	url de la imagen

Propiedades de posicionamiento y visualización

a. Posicionamiento

NOMBRE	DESCRIPCION	VALORES
Position	Marca el tipo de posicionamiento para un elemento	Relative, absolute, fixed
Top, right, bottom y left	Indican el desplazamiento horizontal y vertical de un elemento	Unidad de medida, porcentaje

b. Visualización

NOMBRE	DESCRIPCION	VALORES
Display	Controla la forma de visualizar de un elemento y posibilita su ocultación	Inline, block, none…
Visibility	Permite hacer visibles e invisibles los elementos	Visible, hidden
Overflow	Controla la parte sobrante del contenido de un elemento	Visible, hidden, scroll
z-index (solo para elementos posicionados)	Establece el nivel tridimensional en el que se mostrará un elemento	Valor numérico