

# 1

---

## CSS3

El lenguaje CSS (Cascading Style Sheets) es un lenguaje de marcado de presentación dedicado a la elaboración de páginas web. Fue definido por primera vez en 1994 y, poco después, fue añadido al grupo de trabajo de la W3C como parte del proceso de desarrollo y estandarización.

El crecimiento de especificación CSS ha sido francamente irregular con respecto a otras evoluciones de lenguajes de marcado. De hecho, sólo hay que investigar un poco para ver que en 1998 se publicó la recomendación 2.1 y, poco después, en 1999, ya apareció el primer borrador de la versión actual de CSS. Desde entonces, ha ido evolucionando hasta la especificación que es la actualmente denominamos CSS3.

El lenguaje de marcado CSS3 está pensado para separar el contenido y estructura de la presentación en un documento HTML, XML o XHTML. Esta separación permite, además, mejorar la accesibilidad y usabilidad de los documentos porque proporciona una mayor flexibilidad, una mejor reutilización de código y una reducción de la complejidad gracias, entre otras cosas, a que se evita la repetición de código.

Entre sus características iniciales podemos destacar que es un lenguaje sencillo que permite diferentes métodos de renderizado y que todo se realiza a través de reglas que se aplican en función de unos selectores previamente definidos.

### 1.1 SOPORTE A LOS NAVEGADORES

---

El soporte de CSS3 en los navegadores actuales está más o menos cubierto, no obstante, Internet Explorer sigue siendo el que más problemas da a la hora de hacer que los documentos se comporten de igual forma.

En líneas generales podemos decir que Chrome, Edge, Firefox, Internet Explorer, Opera y Safari son los navegadores más utilizados por los usuarios y que, prácticamente todos, a excepción de Internet Explorer 10 e inferiores, dan soporte a todos los selectores, pseudo-clases y gran parte de las propiedades.

En la dirección [https://www.w3schools.com/cssref/css3\\_browsersupport.asp](https://www.w3schools.com/cssref/css3_browsersupport.asp) es posible encontrar el soporte específico para todas las propiedades CSS de manera individual.

## 1.2 CÓMO FUNCIONA CSS

Básicamente, el lenguaje CSS es una especificación que define un conjunto de reglas, cada una de ellas, definida a través de un selector y una declaración.

Selector	Declaración
div	{ background: white; font-family: Arial; font-size: 14px }
	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;">↑ Propiedad</div> <div style="text-align: center;">↑ Valor</div> <div style="text-align: center;">↑ Propiedad</div> <div style="text-align: center;">↑ Valor</div> <div style="text-align: center;">↑ Propiedad</div> <div style="text-align: center;">↑ Valor</div> </div>

Como se puede apreciar en la ilustración anterior, la regla CSS está compuesta por un selector y una declaración.

El selector, puede ser un nombre de etiqueta HTML, una clase, un identificador, un comodín o, incluso, una combinación de ellos. Su objetivo es localizar el elemento o grupo de elementos dónde se debe aplicar la declaración.

La declaración, está definida a modo de bloque mediante unas llaves que representa el conjunto de pares de propiedad-valor.

Cada uno de estos pares siempre termina con un punto y coma, aunque si es el final de la declaración puede omitirse, como es el caso de la ilustración. El conjunto de estos pares de propiedad-valor es el que definirá el comportamiento visual del objeto que se haya seleccionado por el selector previamente definido.

## 1.3 DEFINICIÓN DE SELECTOR Y CLASIFICACIÓN

Como se acaba de comentar, un selector, puede ser un nombre de etiqueta HTML, una clase, un identificador, un comodín o, incluso, una combinación de ellos y se utilizan para localizar o seleccionar el conjunto de elementos HTML que se desea manipular.

Aunque los selectores pueden ser de muy diversos tipos, se puede establecer una clasificación general en la que se dividen en universales, simples, combinados,

de atributo, pseudo-clases y pseudo-elementos. A continuación, se describen cada uno de ellos con brevedad ya que, posteriormente, se verán en detalle:

**Los selectores simples** son aquellos que permiten seleccionar los elementos a partir del nombre de una etiqueta, identificador o clase. Por ejemplo:

```
button      { background: #f0f0f0; border: 1px solid #ccc; }
#banner     { background: black; }
.form-group { color: #000; height: 32px; line-height: 32px; }
```

**Los selectores combinados** son aquellos que permiten seleccionar los elementos a partir de una relación preestablecida entre ellos. Por ejemplo:

```
div p      { font-family: Arial; font-size: 14px; }
#banner .title { font-family: Verdana; font-size: 21px; }
```

**Las pseudo-clases** son un tipo especial de selectores que permiten seleccionar los elementos a partir del estado en el que se encuentran y siempre van precedidos del símbolo dos puntos. Por ejemplo:

```
button:hover { font-weight: bold; }
a:active     { color: red; }
input:focus  { background: #000; color: white; }
```

**Los pseudo-elementos** son otro tipo especial de selectores que permiten seleccionar partes concretas de los elementos o del documento y siempre van precedidos del símbolo dos puntos repetido dos veces. Por ejemplo:

```
::selection { background: #003366; color: rgba(255,255,255,1); }
::placeholder { background: #333333; color: lightgray; }
.icon::before { content: "\f249"; font-size: 22px; color: #006699; }
```

**Los selectores de atributo** son aquellos que permiten seleccionar los elementos a partir de sus atributos o propiedades y siempre van declarados a través de corchetes. Por ejemplo:

```
a[href^="https"] { color: rgba(255,255,255,1); }
a[href*="blog"]  { color: lightgray; }
a[href$=".php"]  { color: #006699; }
```

El **selector universal** permite selección todos los elementos del documento o del elemento padre que se haya definido.

```
*          { font-family: Arial; font-size: 14px; }
article *  { color: black; }
```

---

## 1.4 UNIDADES DE MEDIDA

---

Existen dos tipos de unidad de medida, las absolutas y las relativas.

### 1.4.1 Unidades absolutas

Las unidades absolutas o fijas son aquellas que no están relacionadas de ninguna forma ni con el contexto, ni con otra entidad. Entre las diferentes propuestas que ofrece CSS como unidades de medida absolutas tenemos:

#### 1.4.1.1 UNIDAD DE MEDIDA PX

El término `px` hace referencia a los puntos de la pantalla, por lo que cada punto en la pantalla es un píxel. Sus posibles valores pueden ser negativos y positivos, dependiendo de la propiedad y contexto.

#### 1.4.1.2 UNIDAD DE MEDIDA PT

El término `pt` hace referencia al tamaño del punto en una pantalla o papel. En lo referente a su equivalencia, un punto se suele traducir a 0.35mm, o lo que es lo mismo, a 1.33 píxeles. Sus posibles valores pueden ser negativos y positivos, dependiendo de la propiedad y contexto.

#### 1.4.1.3 UNIDAD DE MEDIDA IN

El término `in` hace referencia a la longitud del sistema inglés pulgada, es decir, 2.54cm. En lo referente a sus posibles equivalencias, una pulgada se suele traducir en a 96 píxeles o, lo que es lo mismo, 72 puntos. Sus posibles valores pueden ser negativos y positivos, dependiendo de la propiedad y contexto.

#### 1.4.1.4 UNIDADES DE MEDIDA CM, MM Y PC

Los términos `cm`, `mm` y `pc` hacen referencia las conocidas medidas de longitud. En lo referente a su equivalencia, podríamos decir que una pica es equivalente a 12 puntos, o lo que es lo mismo, 4.23 milímetros.

### 1.4.2 Unidades relativas

Las unidades relativas son aquellas que toman como punto de referencia otra entidad o contexto. Entre las diferentes propuestas que ofrece CSS como unidades de medida absolutas tenemos:

#### 1.4.2.1 UNIDAD DE MEDIDA %

El término % hace referencia al porcentaje del marco o contexto actual. Sus valores no tienen por qué ir de cero a cien, ya que también son válidos los valores superiores.

#### 1.4.2.2 UNIDAD DE MEDIDA CH

El término CH hace referencia a la anchura, en píxeles, del carácter 0 del elemento padre, esto es, depende del tamaño de la fuente y letra base realizar la equivalencia.

El tamaño que se toma como referencia es la anchura del carácter 0, por tanto, si se establece el ancho de un elemento a  $80CH$ , lo que se está indicando es que, será de ancho como 80 caracteres 0, todos seguidos sin espacios en blanco.

#### 1.4.2.3 UNIDAD DE MEDIDA EM

El término EM hace referencia a la anchura, en píxeles, de la letra M mayúscula del elemento padre, esto es, si el elemento padre tiene aplicado un tamaño de fuente de 14PX,  $1.2EM$  será equivalente a 16.8PX.

#### 1.4.2.4 UNIDAD DE MEDIDA EX

El término EX hace referencia a la anchura, en píxeles, de la letra X minúscula del elemento padre, esto es, depende del tamaño de la fuente y letra base realizar la equivalencia.

El tamaño que se toma como referencia es la anchura de la X minúscula, por tanto, si se establece el ancho de un elemento a  $80EX$ , lo que se está indicando es que, será de ancho como 80 caracteres x, todos seguidos sin espacios en blanco.

#### 1.4.2.5 UNIDAD DE MEDIDA REM

El término REM hace referencia a la anchura, en píxeles, de la letra M mayúscula del elemento HTML (o raíz), esto es, si el elemento HTML tiene aplicado un tamaño de fuente de 14PX,  $1.2REM$  será equivalente a 16.8PX.

#### 1.4.2.6 UNIDAD DE MEDIDA VMAX

El término VMAX hace referencia a la centésima parte de la altura o anchura de la ventana gráfica o área útil.

La regla de para saber qué valor se está aplicando es “cuál de los valores de ancho y alto es mayor”, esto es, si la altura es mayor que la anchura,  $1VMAX$  será equivalente a  $1VH$ , pero si la anchura es mayor que la altura,  $1VMAX$  será equivalente a  $1VW$ .

#### 1.4.2.7 UNIDAD DE MEDIDA VMIN

El término `VMIN` hace referencia a la centésima parte de la altura o anchura de la ventana gráfica o área útil.

La regla de para saber qué valor se está aplicando es “cuál de los valores de ancho y alto es menor”, esto es, si la altura es menor que la anchura, `1VMIN` será equivalente a `1VH`, pero si la anchura es menor que la altura, `1VMIN` será equivalente a `1VW`.

#### 1.4.2.8 UNIDAD DE MEDIDA VH

El término `VH` hace referencia a la centésima parte de la altura del viewport, esto es, un 1% de la altura de la ventana gráfica o área útil. Por tanto, si se establece el alto de un elemento a `100VH`, lo que se está indicando es que sea el 100% del alto del viewport, o lo que es lo mismo, 100% del alto de la ventana del navegador.

#### 1.4.2.9 UNIDAD DE MEDIDA VW

El término `VW` hace referencia a la centésima parte de la anchura del viewport, esto es, un 1% de la anchura de la ventana gráfica o área útil. Por tanto, si se establece el ancho de un elemento a `100VW`, lo que se está indicando es que sea el 100% del ancho del viewport, o lo que es lo mismo, 100% del ancho de la ventana del navegador.

### 1.5 CODIFICACIÓN DE COLORES

---

Los colores en HTML pueden establecerse a través de los nombres preestablecidos por el agente de usuario, mediante codificación RGB, HSL o hexadecimal. Tanto RGB, como HSL, admiten una variación que permite especificar el canal alfa.

#### 1.5.1 Codificación RGB y RGBA

El modelo de color RGB (Red - Green - Blue) es un modelo matemático abstracto que describe cómo realizar la codificación o representación de colores a partir de unos componentes de intensidad sobre los colores primarios. Dado que su formulación es únicamente a través de rojo, verde, azul, la codificación RGB no presenta la posibilidad de definir un color con transparencia.

El espacio de color RGBA (Red - Green - Blue - Alpha) es una combinación del modelo RGB con un cuarto componente denominado canal de alfa. Este canal alfa es el que posibilita que los colores puedan tener una transparencia.

---

El valor de los componentes rojo, verde y azul se establecen a partir de un valor binario de 8 bits bajo el sistema decimal, es decir, con valores que van de 0 a 255. Sin embargo, el valor del canal alfa, se establece con un valor en tanto por uno.

```
.....  
RGB(0,0,255); /* Color azul al 50% de transparencia */  
RGBA(0,0,255, 0.5); /* Color azul al 50% de transparencia */  
.....
```

En lo referente a su compatibilidad, el formato RGB y RGBA resulta ser compatible con IE 9+, Firefox 3+, Chrome, Safari 3.1+ y Opera 10+.

En lo referente a su compatibilidad, la entrada de colores en formato es compatible con todos los agentes de usuario conocidos.

## 1.5.2 Codificación Hexadecimal

La codificación de colores en hexadecimal funciona exactamente igual que el modelo RGB, es decir, los colores pueden formalizarse o construirse a partir de unos valores de rojo, verde, azul, no obstante, su anotación es diferente.

Cabe destacar que existen algunos navegadores que permiten definir colores en hexadecimal y con transparencia a través de un código de ocho caracteres hexadecimales, en vez de seis. No obstante, este tipo de codificación con transparencia no es compatible ni con Internet Explorer, ni con Microsoft Edge.

El valor de los componentes rojo, verde y azul van de 00 a FF, así como el canal alfa, se establecen a partir de un valor binario de 8 bits bajo el sistema hexadecimal, es decir, con valores que van de 00 a FF.

En general, esta especificación se aplica mediante la concatenación del carácter almohadilla y una cadena de seis caracteres, aunque, también es posible codificar los colores mediante una cadena de longitud tres. Cuando se utiliza este tipo de anotación, el color generado se construye a partir de la duplicación y anidación de cada carácter. Es decir, cuando se especifica un color codificado como 0F0, en realidad, se está generando el color 00FF00.

```
.....  
#0000FF; /* Color azul con 6 caracteres */  
#00F /* Color azul con 3 caracteres */  
.....
```

En lo referente a su compatibilidad, el formato hexadecimal resulta ser compatible con IE 3+, Firefox, Chrome, Safari y Opera 3.5+.

### 1.5.3 Codificación HSL y HSLA

El modelo de color HSL (Hue - Saturation - Lightness) es un modelo matemático abstracto que describe cómo realizar la codificación o representación de colores a partir de unos componentes de matiz, saturación y luminosidad. Dado que su formulación es únicamente a través de estos tres componentes, la codificación HSL no presenta la posibilidad de definir un color con transparencia.

El espacio de color HSLA (Hue - Saturation - Lightness - Alpha) es una combinación del modelo HSL con un cuarto componente denominado canal de alfa. Este canal alfa es el que posibilita que los colores puedan tener una transparencia.

El componente de matiz se mide en grados, por lo que sus posibles valores van de 0 a 360. El valor 0 equivale al rojo, 120 al verde y 240 al azul.

Los componentes de saturación y luminosidad se establecen en términos porcentaje, es decir, con valores que van de 0 a 100 y, el canal alfa se establece en términos de tanto por uno.

```
.....
HSL(0, 240%, 50%);      /* Color azul al 50% de transparencia */
HSLA(0, 240%, 50%, 0.5); /* Color azul al 50% de transparencia */
.....
```

En lo referente a su compatibilidad, el formato hexadecimal resulta ser compatible con IE 9+, Firefox 3+, Chrome, Safari 3.1+ y Opera 10+.

## 1.6 PROPIEDADES

A continuación, se muestra una lista con las propiedades de CSS3 agrupadas por funcionalidad, aunque algunas ellas se verán en otros capítulos más adelante.

### 1.6.1 Texto, fuentes y tipos de letra

#### 1.6.1.1 PROPIEDAD COLOR

Especifica el color del texto. Esta propiedad admite una larga lista de colores a través de su nombre en inglés o mediante su equivalente en RGB(A), HSL(A) o hexadecimal.

#### Ejemplos:

```
p { color: black; }
p { color: #000000; }
p { color: rgb(0, 0, 0); }
p { color: hsl (0, 0%, 0%); }
p { color: rgba(0,0,0,0.0); } /* color transparente */
p { color: transparent; } /* color transparente */
```



### 1.6.1.2 PROPIEDAD DIRECTION

Dirección de la lectura. Puede obtener los valores `ltr` y `rtl` para indicar que el sentido de la lectura es de izquierda a derecha o al revés, respectivamente.

#### Ejemplos:

```
p { direction: rtl; }
article { direction: ltr; }
```

### 1.6.1.3 PROPIEDAD FONT-FAMILY

Especifica el espacio entre caracteres. Entre sus posibles valores podemos encontrar:

- **NORMAL**: Indica el valor prefijado por el agente de usuario.
- **[VALOR]**: Indica un valor establecido en una de las medidas permitidas de CSS.

Si se desea indicar varias fuentes, se puede hacer a través del carácter coma, no obstante, su aplicación será en función de la disponibilidad y de izquierda a derecha, es decir, si la primera no está disponible o no se puede seleccionar, se seguirá intentando con las siguientes.

#### Ejemplos:

```
p { font-family: Arial, sans-serif; }
a { font-family: "Open Sans", sans-serif; } /* Fuente de Google Fonts */
b { font-family: Times New Roman, serif; }
```

### 1.6.1.4 PROPIEDAD FONT-SIZE

Especifica el tamaño del texto. Entre sus posibles valores podemos encontrar:

- **XX-SMALL**: Indica un tamaño de fuente baladí, equivalente a 9PX.
- **X-SMALL**: Indica un tamaño de fuente muy pequeño, equivalente a 10PX.
- **SMALL**: Indica un tamaño de fuente pequeño, equivalente a 13PX.
- **MEDIUM**: Indica un tamaño de fuente medio, equivalente a 16PX. Es el valor por defecto.
- **LARGE**: Indica un tamaño de fuente grande, equivalente a 18PX.
- **X-LARGE**: Indica que el tamaño de fuente muy grande, equivalente a 24PX.
- **XX-LARGE**: Indica un tamaño de fuente enorme, equivalente a 32PX.
- **[VALOR]**: Indica un valor establecido en una de las medidas permitidas de CSS.

### Ejemplos:

```
html, body { font-size: 14px; }
h1         { font-size: 1.5rem; }
h2         { font-size: large; }
li         { font-size: 1em; }
```

#### 1.6.1.5 PROPIEDAD FONT-STYLE

Especifica el estilo del texto. Entre sus posibles valores podemos encontrar:

- **NORMAL**: Indica que se muestre con un estilo de fuente normal.
- **ITALIC**: Indica que se muestre con un estilo de letra cursiva.
- **OBLIQUE**: Indica que se muestre con un estilo de fuente oblicuo.

### Ejemplos:

```
p { font-style: normal; }
em { font-style: italic; }
a { font-style: oblique; }
```

#### 1.6.1.6 PROPIEDAD FONT-STRETCH

Especifica el nivel de condensación o expansión del texto. Entre sus posibles valores podemos encontrar:

- **ULTRA-CONDENSED**: Indica que el texto sea lo más estrecho posible.
- **EXTRA-CONDENSED**: Indica que el texto sea algo menos estrecho que ULTRA-CONDENSED.
- **NARROW**: Indica que el texto sea un poco menos estrecho que su predecesor.
- **CONDENSED**: Indica que el texto sea un poco menos estrecho que NARROW.
- **SEMI-CONDENSED**: Indica que el texto debe ser un poco menos estrecho que CONDENSED, cercano a NORMAL.
- **NORMAL**: Indica que el texto no se debe estirar ni estrechar. Es el valor por defecto.
- **EXPANDED**: Indica que el texto sea un poco más ancho de lo normal.
- **EXTRA-EXPANDED**: Indica que el texto sea un poco más ancho que EXPANDED.
- **ULTRA-EXPANDED**: Indica que el texto sea lo más ancho posible.

### Ejemplos:

```
p { font-stretch: semi-condensed; }
em { font-stretch: narrow; }
a { font-stretch: ultra-expanded; }
```

### NOTA

Sólo las fuentes que ofrezcan caras adicionales con variaciones de caracteres condensados y/o expandidos podrán hacer uso de esta propiedad.

#### 1.6.1.7 PROPIEDAD FONT-VARIANT

Especifica la variación del texto. Esta propiedad presenta múltiples valores, pero el único que, de verdad se utiliza, es `SMALL-CAPS` y `ALL-SMALL-CAPS` para indicar que el texto se muestre con las letras minúsculas convertidas a mayúsculas, pero con un tamaño menor.

### Ejemplos:

```
p { font-variant: small-caps; }
em { font-variant: all-small-caps; }
```

#### 1.6.1.8 PROPIEDAD FONT-WEIGHT

Especifica el grosor del texto. Entre sus posibles valores podemos encontrar:

- **NORMAL**: Indica que se muestre la letra con un grosor normal. Es el valor por defecto.
- **BOLD**: Indica que se muestren los caracteres gruesos.
- **100, 200, 300, 400, 500, 600, 700, 800, 900**: Indica que se muestre el grosor de los caracteres en función de estos valores.

Muchas fuentes vectoriales utilizan esta propiedad para definir diferentes pesos o grosores de carácter. En general, el valor `100` se corresponde con un tipo de letra muy fina o delgada, el valor `400` a un tipo `NORMAL` y `700` o superior a un tipo `BOLD`.

### Ejemplos:

```
p { font-weight: 100; }
em { font-weight: 400; }
a { font-weight: bold; }
```

### 1.6.1.9 PROPIEDAD LETTER-SPACING

Especifica el espacio entre caracteres que se debe aplicar al texto. Entre sus posibles valores podemos encontrar:

- **NORMAL**: Indica que se muestre la letra con un espacio entre caracteres normal. Es el valor por defecto.
- **[VALOR]**: Indica un valor establecido en una de las medidas permitidas de CSS.

#### Ejemplos:

```
p { letter-spacing: normal; } /* Ejemplo a normal */
em { letter-spacing: 10px; } /* E j e m p l o a 10 p x */
```

### 1.6.1.10 PROPIEDAD LINE-HEIGHT

Especifica el espacio entre líneas de un texto. Entre sus posibles valores podemos encontrar:

- **NORMAL**: Indica que se muestre la letra con un espacio entre caracteres normal definido por defecto.
- **[NÚMERO]**: Indica un factor de multiplicación para el tamaño de fuente actual.
- **[VALOR]**: Indica un valor establecido en una de las medidas permitidas de CSS.

#### Ejemplos:

```
p { line-height: normal; }
h1 { line-height: 2; } /* Si font-size es 10px, equivaldrá a 20px */
h2 { line-height: 18px; }
```

### 1.6.1.11 PROPIEDAD SHAPE-OUTSIDE

Especifica cómo se distribuirá el texto alrededor de un elemento flotante, es decir, indica cómo se recolocará el texto alrededor de un elemento que tiene la propiedad FLOAT establecida con un valor diferente a NONE.

Entre sus posibles valores podemos encontrar:

- **CIRCLE**: Indica que el texto debe flotar siguiendo una forma circular. El radio y valores de posición pueden establecerse a través de cualquiera de las unidades de medida estándar de CSS y, adicionalmente, esta función permite el uso de CLOSEST-SIDE y FARTHEST-SIDE que sirven para

indicar que el círculo debe llegar hasta el lado más cercano o hasta la esquina más lejana, respectivamente. Su sintaxis es:

```
.....
shape-outside: circle();
shape-outside: circle(radio at posX posY);
.....
```

- **ELLIPSE:** Indica que el texto debe flotar siguiendo una forma elíptica. El radio y valores de posición pueden establecerse a través de cualquiera de las unidades de medida estándar de CSS y, adicionalmente, esta función permite el uso de CLOSEST-SIDE y FARTHEST-SIDE que sirven para indicar que el círculo debe llegar hasta el lado más cercano o hasta la esquina más lejana, respectivamente. Su sintaxis es:

```
.....
shape-outside: ellipse();
shape-outside: ellipse(radio en X Y at posX posY);
.....
```

- **POLYGON:** Indica que el texto debe flotar siguiendo una forma poligonal. La especificación de pares de coordenadas X e Y se deben separar a través de comas y usando las unidades de medida estándar de CSS. Su sintaxis es:

```
.....
shape-outside: polygon(X Y, X Y, ..., X Y);
.....
```

- **INSET:** Indica que el texto debe flotar siguiendo una forma rectangular. Adicionalmente, admite la palabra clave ROUND seguido del valor de redondeo para los valores de la diagonal superior izquierda y diagonal superior derecha que hacen que la forma se vuelva redondeada. Para especificar el valor de las coordenadas X e Y se puede usar cualquiera de las unidades de medida estándar de CSS, separando, esos sí, los valores a través de comas. Su sintaxis es:

```
.....
shape-outside: inset(top right bottom left);
shape-outside: inset(top right bottom left round r1 r2);
.....
```

- **URL:** Indica que el texto debe flotar siguiendo las transparencias de la imagen proporcionada por parámetro. Para especificar el valor de esta propiedad se puede utilizar codificación en Base64 o una URI. Su sintaxis es:

```
.....
shape-outside: url([Base64]);
shape-outside: url([URI]);
.....
```

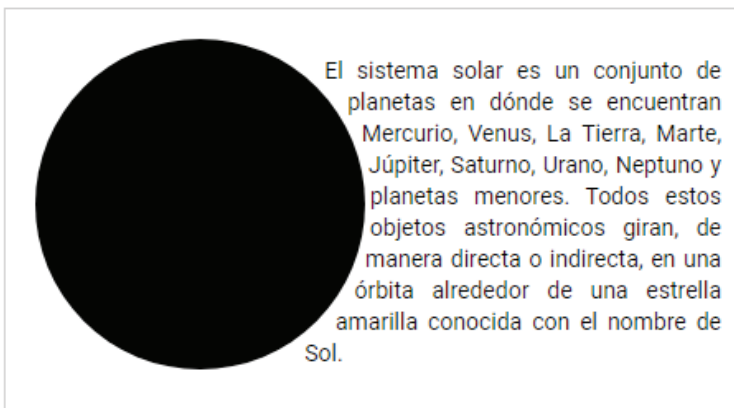
Aunque las opciones anteriores son las más utilizadas, esta propiedad también permite dar forma a partir de las siguientes palabras clave:

- **BORDER-BOX:** Indica debe flotar por fuera del borde del elemento flotante.
- **CONTENT-BOX:** Indica debe flotar alrededor del contenido del elemento flotante.
- **MARGIN-BOX:** Indica debe flotar por fuera del margen externo del elemento flotante.
- **PADDING-BOX:** Indica debe flotar por fuera del margen interno del elemento flotante.

### Ejemplo gráfico 1

```
<style>
  article { border:1px solid #ccc; line-height:1.42; margin:0 auto;
            padding:1em; text-align: justify; width: 450px; }
  article div{ background: #000000; border-radius: 100%;
               float:left; height: 200px; margin: 0.25rem;
               shape-outside: circle(); width: 200px; }
</style>

<article>
  <div></div>
  <p>El sistema solar es un conjunto de planetas en dónde se encuentran Mercurio, Venus, La Tierra, Marte, Júpiter, Saturno, Urano, Neptuno y planetas menores. Todos estos objetos astronómicos giran, de manera directa o indirecta, en una órbita alrededor de una estrella amarilla conocida con el nombre de Sol.</p>
</article>
```



### Ejemplo gráfico 2

```
<style>
article      { border:1px solid #ccc; line-height:1.42; margin: 1rem auto;
              padding:1em; text-align: justify; width: 450px; }
article img{ float: left; height: 150px; object-fit: contain;
              padding: 0 0.8rem; shape-image-threshold: 0.5;
              shape-outside: url(/img/sistema-solar.png); width: auto; }
</style>

<article>
  
  <p>El sistema solar es un conjunto de planetas en dónde se encuentran Mercurio, Venus, La Tierra, Marte, Júpiter, Saturno, Urano, Neptuno y planetas menores. Todos estos objetos astronómicos giran, de manera directa o indirecta, en una órbita alrededor de una estrella amarilla conocida con el nombre de Sol.</p>
</article>
```



Figura 1.1. Imagen extraída de [https://es.m.wikibooks.org/wiki/Archivo:Draw\\_Solar\\_System.png](https://es.m.wikibooks.org/wiki/Archivo:Draw_Solar_System.png)

Cabe destacar que, cuando se recurre a este tipo de efectos o presentaciones, también se suele indicar la propiedad `SHAPE-IMAGE-THRESHOLD` y `SHAPE-MARGIN`. La primera, define el rango de valores válidos del canal alfa para extraer la forma y, la segunda, establece un margen alrededor de la forma.

También es importante entender que, no todas las imágenes con transparencias generan el efecto deseado. Esto suele ocurrir porque las imágenes no siempre guardan las transparencias de manera correcta. La forma de solventar este problema suele pasar por un editor de imágenes como PhotoShop, seleccionar todo el contenido externo a la figura de la imagen con la herramienta de varita mágica y eliminar el fragmento de imagen pulsando la tecla suprimir (o la opción de menú Edición / Borrar).

### 1.6.1.12 PROPIEDAD TEXT-ALIGN

Especifica la alineación del texto. Entre sus posibles valores podemos encontrar LEFT, RIGHT, CENTER y JUSTIFY que indican que el texto se alinee a la izquierda, a la derecha, de manera centrada o de forma justificada, respectivamente.

Cabe destacar que la alineación justificada no suele ser una buena opción si se desea que la página sea accesible.

#### Ejemplos:

```
p { text-align: left; }
td.number { text-align: right; }
```

### 1.6.1.13 PROPIEDAD TEXT-DECORATION

Especifica la decoración agregada al texto. Aunque esta propiedad presenta múltiples valores, los más recurrentes y frecuentemente utilizados son:

- **UNDERLINE**: Indica que el texto se muestre subrayado por debajo del texto.
- **OVERLINE**: Indica que el texto se muestre subrayado por encima del texto.
- **LINE-THROUGH**: Indica que el texto se muestre tachado.

#### Ejemplos:

```
a { text-decoration: underline; }
h1 { text-decoration: overline; }
.error { text-decoration: line-through; }
```

### 1.6.1.14 PROPIEDAD TEXT-INDENT

Especifica la sangría que se debe aplicar a la primera línea del texto. Entre sus posibles valores podemos encontrar:

- **[PORCENTAJE]**: Indica que define un tanto por ciento de sangría con respecto al ancho del elemento actual.
- **[VALOR]**: Indica un valor establecido en una de las medidas permitidas de CSS.

Aunque no es frecuente, el valor de la sangría puede ser negativo. Si esto es así, el texto se sangrará a la izquierda en vez de hacia la derecha.



### Ejemplos:

```
li { text-indent: 15px; }
div { text-indent: -1.5em; }
span { text-indent: 6%; }
```

#### 1.6.1.15 PROPIEDAD TEXT-SHADOW

Es una propiedad compuesta que agrega una sombra al texto. Su sintaxis es:

```
text-shadow: PosX PosY radio_desenfoque color;
```

Si el valor de la posición horizontal es positivo, la sombra avanzará en sentido hacia la derecha, por lo que, si es negativo, avanzará en sentido hacia la izquierda. Algo similar pasa con el segundo parámetro. Si el valor de la posición vertical es positivo, la sombra avanzará en sentido hacia abajo, por lo que, si es negativo, avanzará en sentido hacia arriba.

Cabe destacar que, salvo excepciones, la propiedad TEXT-SHADOW no se debe utilizar porque, además de dificultar su lectura y disminuir la legibilidad, puede proporcionar una imagen corporativa “desaliñada”.

### Ejemplos:

```
P { text-shadow: 0 0 3px #333; }
div { text-shadow: -1px -1px 5px HSL(0, 0%, 0%); }
span { text-shadow: 1% 1% 5px red; }
```

#### 1.6.1.16 PROPIEDAD TEXT-TRANSFORM

Especifica la capitalización del texto. Entre sus posibles valores, los más recurrentes y frecuentemente utilizados son:

- **NONE**: Indica que no se debe capitalizar el texto. Es el valor por defecto.
- **CAPITALIZE**: Indica que se capitalice el primer carácter de cada palabra.
- **UPPERCASE**: Indica que se transforme todo el texto a mayúsculas.
- **LOWERCASE**: Indica que se transforme todo el texto a minúsculas.

### Ejemplos:

```
li { text-transform: uppercase; }
div { text-transform: lowerrcase; }
span { text-transform: capitalize; }
```

### 1.6.1.17 PROPIEDAD VERTICAL-ALIGN

Especifica la alineación vertical para el elemento y es particularmente útil cuando se trabaja con texto e imágenes. Entre sus posibles valores podemos encontrar:

- **BASELINE**: Indica que el texto debe alinearse con respecto a la línea base del elemento padre. Es el valor por defecto.
- **SUB**: Indica que el texto debe alinearse con respecto a la línea base del subíndice del elemento padre.
- **SUPER**: Indica que el texto debe alinearse con respecto a la línea base del superíndice del elemento padre.
- **TOP**: Indica que el texto debe alinearse en la parte superior con respecto al elemento más alto de la línea.
- **TEXT-TOP**: Indica que el texto debe alinearse en la parte superior con respecto al elemento padre.
- **MIDDLE**: Indica que el texto debe alinearse en el medio del elemento padre.
- **BOTTOM**: Indica que el texto debe alinearse en la parte inferior con respecto al elemento más bajo de la línea.
- **TEXT-BOTTOM**: Indica que el texto debe alinearse en la parte inferior con respecto al elemento padre.

#### Ejemplos:

```
img { vertical-align: top; }  
div { vertical-align: middle; }  
span { vertical-align: bottom; }
```

### 1.6.1.18 PROPIEDAD WHITE-SPACE

Especifica como se debe gestionar el espacio en blanco en el elemento textual actual. Entre sus posibles valores podemos encontrar:

- **NORMAL**: Indica que las secuencias de espacios en blanco y tabulaciones se reducirán a un único elemento y el texto saltará a la línea siguiente cuando sea necesario. Es el valor por defecto.
- **NOWRAP**: Indica que las secuencias de espacios en blanco y tabulaciones se reducirán a un único elemento, pero el texto nunca saltará a la siguiente línea hasta que encuentre un elemento BR.

- **PRE:** Indica que las secuencias de espacios en blanco y tabulaciones se conservarán, pero el texto nunca saltará a la siguiente línea hasta que encuentre un elemento BR. Actúa como el elemento PRE de HTML.
- **PRE-LINE:** Indica que las secuencias de espacios en blanco y tabulaciones se reducirán a uno y el texto saltará a la siguiente línea cuando sea necesario.
- **PRE-WRAP:** Indica que las secuencias de espacios en blanco y tabulaciones se conservarán y el texto saltará a la siguiente línea cuando sea necesario.

### Ejemplos:

```
li { white-space: normal; }  
div { white-space: nowrap; }  
span { white-space: pre; }
```

#### 1.6.1.19 PROPIEDAD WORD-BREAK

Especifica como se deben cortar las palabras cuando no entran en el espacio asignado al elemento. Entre sus posibles valores podemos encontrar:

- **NORMAL:** Indica que las palabras deben cortarse en función de las reglas predeterminadas. En general, esto se convierte en que no se cortan y la palabra entera cae hacia la siguiente línea. Es el valor por defecto.
- **BREAK-ALL:** Indica que las palabras deben cortarse por cualquier carácter para que entren en el espacio del elemento.
- **KEEP-ALL:** Indica que las palabras NO deben cortarse bajo ninguna circunstancia, siempre y cuando, sean textos CJK (chino, japonés y coreano). Para el resto, es equivalente a NORMAL.
- **BREAK-WORD:** Indica que las palabras pueden cortarse en puntos arbitrarios para que entren en el espacio del elemento. Es el comportamiento habitual para textos no CJK.

### Ejemplos:

```
li { word-break: break-all; }  
div { word-break: normal; }  
span { word-break: break-word; }
```

### 1.6.1.20 PROPIEDAD WORD-SPACING

Especifica el espacio que debe haber entre las palabras del texto. Entre sus posibles valores podemos encontrar:

- **NORMAL**: Indica que las palabras deben estar separadas por un espacio equivalente a 0.25EM. Es el valor por defecto.
- **[VALOR]**: Indica un valor establecido en una de las medidas permitidas de CSS.

#### Ejemplos:

```
li { word-spacing: normal; }
div { word-spacing: 0.25em; }
span { word-spacing: 0.6vw; }
```

### 1.6.1.21 PROPIEDAD WORD-WRAP

Especifica que las palabras del texto pueden ajustarse o separarse a la siguiente línea. Entre sus posibles valores podemos encontrar:

- **NORMAL**: Indica que se deben cortar las palabras sólo en los puntos de ruptura permitidos, lo que se suele convertir en que no se corten. Es el valor por defecto.
- **BREAK-WORD**: Indica que se pueden cortar las palabras en cualquier punto.

#### Ejemplos:

```
li { word-wrap: break-word; }
p { word-wrap: normal; }
div { word-wrap: normal; }
```

## 1.6.2 Márgenes internos y externos

Aunque se haga referencia con terminología similar en nuestro idioma, el uso de los márgenes internos y externos tiene diferentes objetivos. Mientras que el margen interno puede estar enfocado a provocar un efecto decorativo, el externo puede estar enfocado a provocar una mejor legibilidad.

