

# CSS

---

Ángel Sobrino

cursoprogramadores@gmail.com



# CSS Introducción

## Lo que deberías saber ya

Antes de continuar, debe tener un conocimiento básico de los siguientes temas:

- HTML / XHTML

## ¿Qué es CSS?

- **CSS** es sinónimo de **C**ascading **S**tyle **S**heets
- Los estilos definen la **forma de mostrar** los elementos HTML
- Los Estilos se añadieron a 4.0 HTML
- **Las Hojas de estilo externas** puede ahorrar un montón de trabajo
- Las Hojas de estilo externas se almacenan en **los archivos CSS**

## CSS demo

Visitar [csszengarden.com](http://csszengarden.com)

\*Inciso

Probad el ColorZilla y BlueGriffon (editor WYSIWYG)

## Los Estilos Css resuelven un gran problema

HTML nunca fue pensado para contener etiquetas para dar formato a un documento.

HTML fue pensado para definir el contenido de un documento:

<h1> Este es un encabezado </ h1>

<p> Este es un párrafo. </ p>

Cuando las etiquetas como<font> y los atributos de color se añadieron a la especificación HTML 3.2, comenzó una pesadilla para los desarrolladores web. Los Desarrollo de sitios web de gran tamaño, donde las fuentes y la información de color se agregan a cada página, se convirtieron en un proceso largo y costoso.

Para resolver este problema, el Consorcio World Wide Web (W3C) creó CSS.

En HTML 4.0, todo el formato puede ser eliminado del documento HTML, y almacenarse en un archivo CSS.

Hoy en día todos los navegadores soportan CSS.

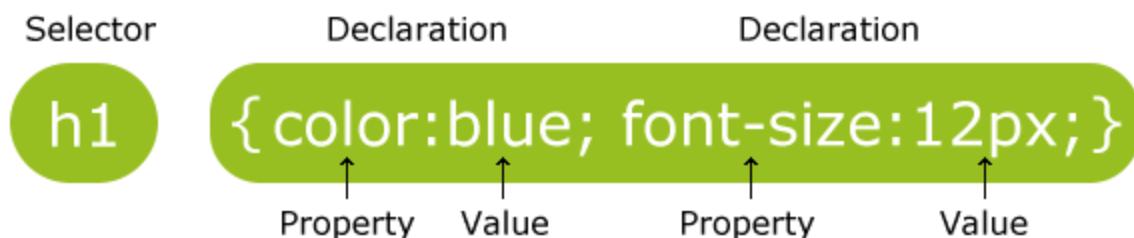
## CSS ahorra mucho trabajo!

CSS define cómo los elementos HTML se van a mostrar.

Los estilos se guardan normalmente en archivos CSS externos. Las Hojas de estilo externas le permiten cambiar el aspecto y el diseño de todas las páginas de un sitio Web, simplemente editando un solo archivo!

## CSS Sintaxis

Una regla CSS consta de dos partes principales: un selector, y de una o más declaraciones:



El selector es normalmente el elemento HTML al que deseamos aplicar el estilo.

Cada declaración consta de una propiedad y un valor.

La propiedad es el atributo de estilo que deseamos cambiar. Cada propiedad tiene un valor.

## Ejemplo CSS

Una declaración CSS siempre termina con un punto y coma, y los grupos de la declaración están rodeadas por llaves:

```
p {color:red;text-align:center;}
```

Para hacer más legible la CSS, puedes poner una declaración en cada línea, así:

### Ejemplo

```
p
{
    color:red;
    text-align:center;
}
```

La especificación HTML5 dice que el atributo type es puramente informativo (advisory) y no explica en detalle cómo deben actuar los navegadores si se omite. No dice explícitamente que omitir un atributo type sea válido o inválido, pero puedes omitirlo con seguridad sabiendo que los navegadores seguirán reaccionando como se espera.

```
<!DOCTYPE html>
<html>
<head>
<!-- o <style> si el navegador es moderno -->
<style type="text/css">
p
{
    color:red;
    text-align:center;
}
</style>
</head>

<body>
<p>Hello World!</p>
<p>This paragraph is styled with CSS.</p>
</body>
</html>
```

Otro ejemplo

```
<html>
<head>
<style>
    body {background-color:yellow;}
    h1 {font-size:36pt;}
    h2 {color:blue;}
    p {margin-left:50px;}
</style>
</head>

<body>

<h1>This header is 36 pt</h1>
<h2>This header is blue</h2>

<p>This paragraph has a left margin of 50 pixels</p>

</body>
</html>
```

## CSS Comentarios

Los comentarios se utilizan para explicar su código, y puede ayudarte a la hora de editar el código fuente en una fecha posterior. Los comentarios son ignorados por los navegadores.

Un comentario CSS comienza con "/ \*" y finaliza con "\* /", así:

```
/*Esto es un comentario*/
p
```

```
{  
text-align:center;  
/*Esto es otro comentario*/  
color:black;  
font-family:arial;  
}
```

## CSS Id y Clase

### Los selectores id y class

Además de establecer un estilo para un elemento HTML, CSS le permite especificar tus propios selectores: "id" y "class".

#### El selector id

El selector id se utiliza para especificar un estilo para un elemento individual, único.

El selector de ID utiliza el atributo id del elemento HTML, y se define con un "#".

La regla de estilo a continuación se aplica al elemento con id = "para1":

#### Ejemplo

```
#para1  
{  
text-align:center;  
color:red;  
}
```

Código completo

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
#para1  
{  
    text-align:center;  
    color:red;  
}  
</style>  
</head>  
  
<body>  
<p id="para1">Hello World!</p>  
<p>This paragraph is not affected by the style.</p>  
</body>  
</html>
```

**NO** empezar un nombre de identificación con un número! No funciona en Mozilla /  
I<sup>!</sup>efox.

## El selector de clase

El selector de clase se utiliza para especificar un estilo para un grupo de elementos. A diferencia del selector de id, el selector de clase es la más utilizada sobre varios elementos.

Esto le permite definir un estilo particular para muchos elementos HTML de la misma clase.

El selector de clase utiliza el atributo de clase HTML, y se define con un ":".

En el siguiente ejemplo, todos los elementos HTML con class = "centro" estarán alineado al centro:

### Ejemplo

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

<!DOCTYPE html>
<html>
<head>
<style>
.centro
{
    text-align:center;
}
</style>
</head>

<body>
<h1 class="centro">Cabecera alineada al centro </h1>
<p class="centro">Párrafo alineado al centro.</p>
</body>
</html>
```

También puede especificar que sólo determinados elementos HTML debería verse afectada por una clase.

En el ejemplo de abajo, todos los elementos p con class = "centro" serán alineado al centro:

### Ejemplo

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

```
<!DOCTYPE html>
<html>
<head>
<style>
p.centro
{
    text-align:center;
}
</style>
```

```
</head>

<body>
<h1 class="centro">Esta cabecera no se verá afectada </h1>
<p class="centro">Este párrafo estará alineado al centro.</p>
<p> Este párrafo no estará alineado al centro.</p>

</body>
</html>
```

 **NO** empezar un nombre de clase con un número! Esto sólo es compatible con Internet Explorer.

## CSS Cómo...

Cuando un navegador lee una hoja de estilo, aplicará el formato al documento de acuerdo con ella.

### Tres formas de insertar CSS

Hay tres formas de insertar una hoja de estilo:

- Hoja de estilos externa
- Hoja de estilo interna
- Estilo en línea

### Hoja de estilos externa

Una hoja de estilos externa es ideal cuando el estilo se aplica a muchas páginas. Con una hoja de estilos externa, puede cambiar el aspecto de un sitio Web completo, cambiando un archivo. Cada página debe enlazar a la hoja de estilo con la etiqueta `<link>`. La etiqueta `<link>` va dentro de la sección "head":

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

Una hoja de estilo externa puede ser escrita en cualquier editor de texto. El archivo no debe contener ninguna etiqueta HTML. Su hoja de estilo debe ser guardado con una extensión. Css. Un ejemplo de un archivo de hoja de estilo se muestra a continuación:

```
hr {color:sienna;}
p {margin-left:20px;}
body {background-image:url("images/back40.gif");}
```

 No agregues un espacio entre el valor de la propiedad y la unidad

(como margin-left: 20px).

(como margin-left: 20px)

La forma correcta es: margin-left:20px

Ejemplo completo

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css"/>
</head>
<body>
<h1 class="centro">no le afectará el centrado</h1>
<p class="centro">le afectará</p>
<p>no le afectará el centrado</p>
</body>
</html>
```

mystyle.css

```
p.centro {color:sienna;text-align:center;}
p {margin-left:20px;background-color:#c0c0c0;}
```

## Hoja de estilo interna

Una hoja de estilo interna debe ser usada cuando tenemos un documento único que tiene un estilo único. Para definir estilos internos utilizamos la etiqueta `<style>` en la sección de cabecera de una página HTML, , así:

```
<head>
<style>
hr {color:sienna;}
p {margin-left:20px;}
body {background-image:url("images/back40.gif");}
</style>
</head>
```

Código completo

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```
p.centro {color:sienna;text-align:center;}  
p {margin-left:20px;}  
</style>  
</head>  
<body>  
<p>no le afectará el centrado</p>  
<p class="centro">le afectará</p>  
</body>  
</html>
```

## Estilos en línea

El estilo en línea pierde muchas de las ventajas de las hojas de estilo ya que el contenido de la página se mezcla con la presentación. Utiliza este método con moderación!

Para utilizar estilos en línea se utiliza el atributo de estilo en la etiqueta correspondiente. El atributo de estilo puede contener cualquier propiedad CSS. El ejemplo muestra cómo cambiar el color y el margen izquierdo de un párrafo:

```
<p style="color:sienna;margin-left:20px">This is a paragraph.</p>
```

## Múltiples hojas de estilo

Si algunas propiedades se han establecido para el mismo selector en diferentes hojas de estilo, los valores se heredará de la hoja de estilo más específico.

Por ejemplo, una hoja de estilo externa tiene estas propiedades para el selector h3:

```
h3  
{  
color:red;  
text-align:left;  
font-size:8pt;  
}
```

Y una hoja de estilo interna consta de las siguientes propiedades para el selector h3:

```
h3  
{  
text-align:right;  
font-size:20pt;  
}
```

Si la página con la hoja de estilo interna también se vincula a la hoja de estilos externa las propiedades h3 será:

```
color:red;  
text-align:right;  
font-size:20pt;
```

El color se hereda de la hoja de estilos externa y la alineación del texto y el tamaño de fuente se sustituye por la hoja de estilo interna.

Código completo

```
<!DOCTYPE html>  
  
<html>  
  
<head>  
  
<link rel="stylesheet" type="text/css" href="micss.css">  
  
<style>  
  
h3  
  
{  
  
    text-align:right;  
  
    font-size:20pt;  
  
}  
  
</style>  
  
</head>  
  
<body>  
  
<h3>Hola a todos</h3>  
  
¿qué tal estáis?  
  
</body>  
  
</html>
```

Contenido micss.css

```
h3  
{  
color:red;  
text-align:left;  
font-size:8pt;  
}
```

## Múltiples estilos en cascada en Uno

Los estilos pueden ser especificados:

- dentro de un elemento HTML
- dentro de la sección de la cabeza de una página HTML
- en un archivo CSS externo

**Nota:** podemos referenciar varias hojas de estilo externas en un mismo documento HTML.

Orden en cascada

### ¿Qué estilo se utiliza cuando hay más de un estilo especificado para un elemento HTML?

En términos generales podemos decir que todos los estilos funcionaran "en cascada" y generarán en una nueva hoja de estilo "virtual" con las siguientes reglas, la número cuatro tiene la prioridad más alta:

1. Navegador predeterminado
2. Hoja de estilos externa
3. Hoja de estilo interna (en la sección de la cabeza)
4. Estilo en línea (dentro de un elemento HTML)

Por lo tanto, un estilo en línea (dentro de un elemento HTML) tiene la mayor prioridad, lo que significa que va a reemplazar un estilo definido dentro de la etiqueta head, o en una hoja de estilos externa, o en un navegador (el valor predeterminado).

 **Nota:** Si el enlace a la hoja de estilos externa se coloca después de la hoja de estilo interna en <head> HTML, la hoja de estilos externa anulará la hoja de estilo interna!

## CSS Background

La Propiedad CSS Background se utilizan para definir el fondo de un elemento.

Las propiedades CSS utilizadas para efectos de fondo:

- background-color
- background-image
- background-repeat
- background-attachment
- background-position

### Color de fondo

La propiedad background-color especifica el color de fondo de un elemento.

El color de fondo de una página se define en el selector de cuerpo:

## Ejemplo

```
body {background-color:#b0c4de; }

<!DOCTYPE html>
<html>
<head>
<style>
body
{
    background-color:#b0c4de;
}
</style>
</head>

<body>

<h1>My CSS web page!</h1>
<p>Hello world! This is a W3Schools.com example.</p>

</body>
</html>
```

Con CSS, un color se suele especificar a través de:

- un valor HEX - como "#ff0000"
- un valor RGB - como "rgb(255,0,0)"
- un nombre de color - como "red"

En el ejemplo siguiente, el h1, p, y los elementos div tienen diferentes colores de fondo:

## Ejemplo

```
h1 {background-color:#6495ed; }
p {background-color:#e0ffff; }
div {background-color:#b0c4de; }

<!DOCTYPE html>
<html>
<head>
<style>
body
{
    background-color:#c0c0c0;
}

h1
{
    background-color:#6495ed;
}
p
{
    background-color:#e0ffff;
}
div
{
    background-color:#b0c4de;
}
</style>
```

```
</head>

<body>

<h1>CSS background-color example!</h1>
<div>
This is a text inside a div element.
<p>This paragraph has its own background color.</p>
We are still in the div element.
</div>

</body>
</html>
```

## Imagen de fondo

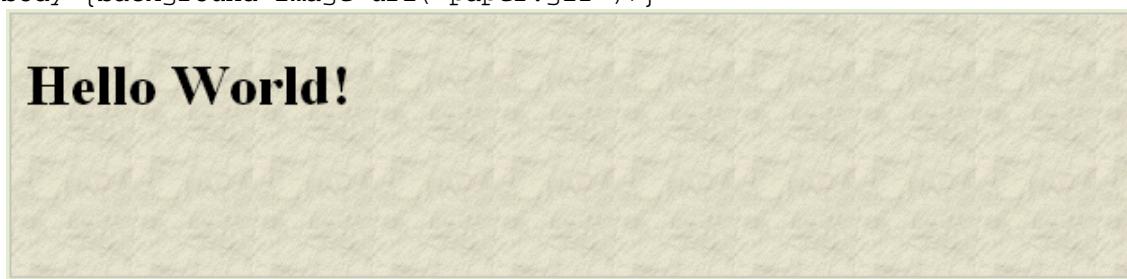
La propiedad `background-image` especifica una imagen para usar como fondo de un elemento.

Por defecto, la imagen se repite por lo que cubre todo el elemento.

La imagen de fondo de una página se puede establecer de la siguiente manera:

### Ejemplo

```
body {background-image:url('paper.gif');}
```



paper.gif

```
<!DOCTYPE html>
<html>
<head>
<style>
body
{
    background-image:url('paper.gif');
}
</style>
</head>

<body>
<h1>Hello World!</h1>
</body>
```

</html>

A continuación se muestra un ejemplo de una mala combinación de texto y la imagen de fondo. El texto es casi ilegible:

### Ejemplo

```
body {background-image:url('bgdesert.jpg');}
```



```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-image:url('bgdesert.jpg');}
</style>
</head>

<body>
<h1>Hello World!</h1>
<p>This text is not easy to read on this background image.</p>
</body>
</html>
```

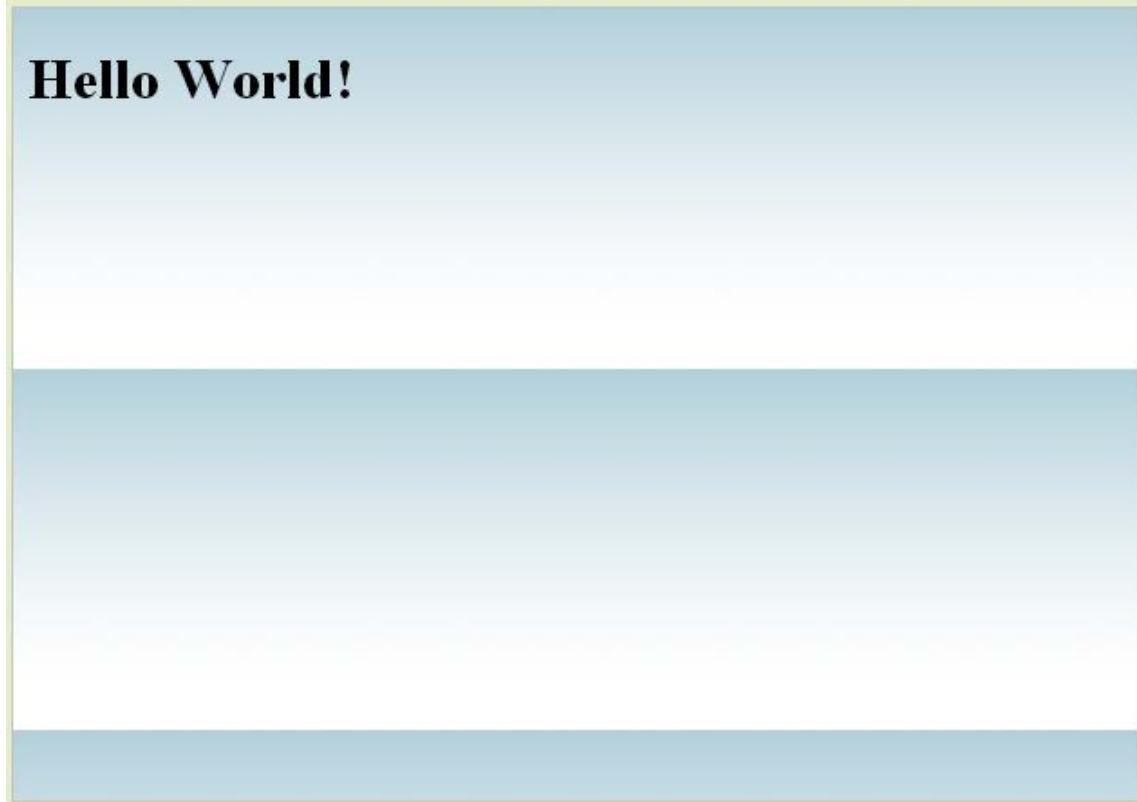
## Imagen de fondo - Repetición horizontal o vertical

De manera predeterminada, la propiedad background-image repite una imagen tanto horizontal como verticalmente.

Algunas de las imágenes sólo se debe repetir en horizontal o vertical, para que no quede demasiado mal:

## Ejemplo

```
body
{
background-image:url('gradient2.png');
```



Hello World!

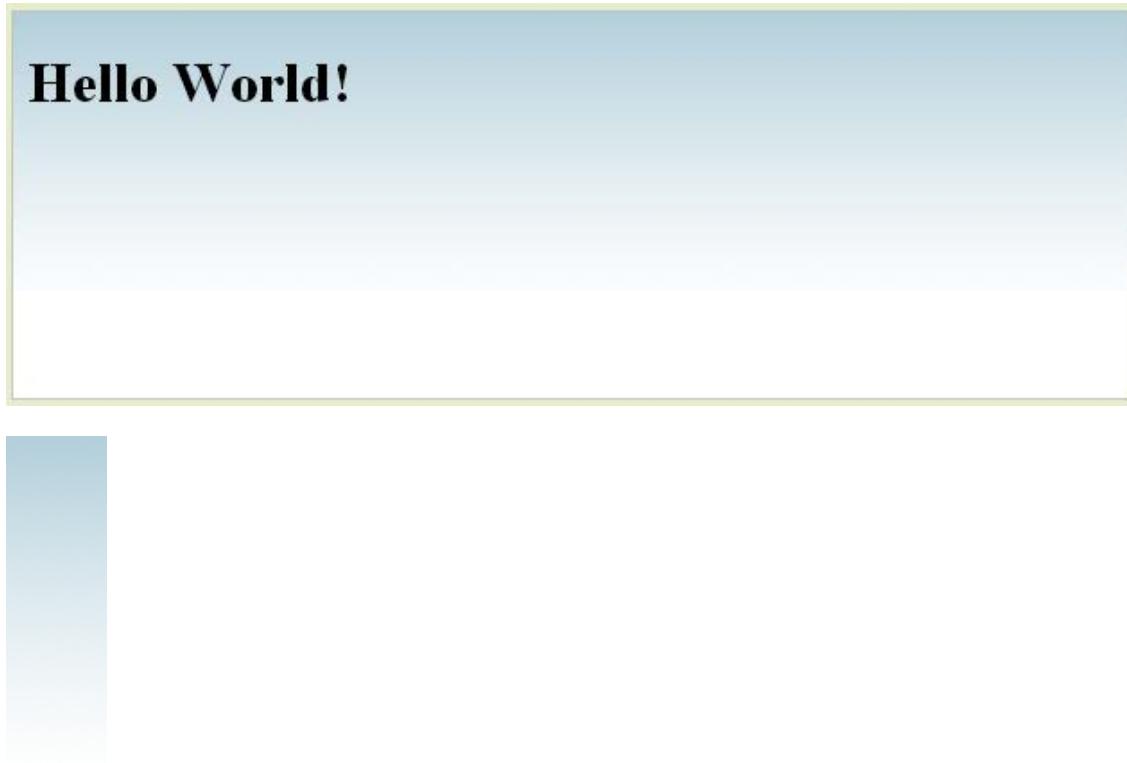
```
<!DOCTYPE html>
<html>
<head>
<style>
body
{
    background-image:url('gradient2.png');
}
</style>
</head>

<body>
<h1>Hello World!</h1>
</body>
</html>
```

Si la imagen es repetida sólo horizontalmente (repeat-x), el fondo se verá mejor:

## Ejemplo

```
body
{
background-image:url('gradient2.png');
background-repeat:repeat-x;
}
```



Gradient.png

```
<!DOCTYPE html>
<html>
<head>
<style>
body
{
background-image:url('gradient2.png');
background-repeat:repeat-x;
}
</style>
</head>

<body>
<h1>Hello World!</h1>
</body>

</html>
```

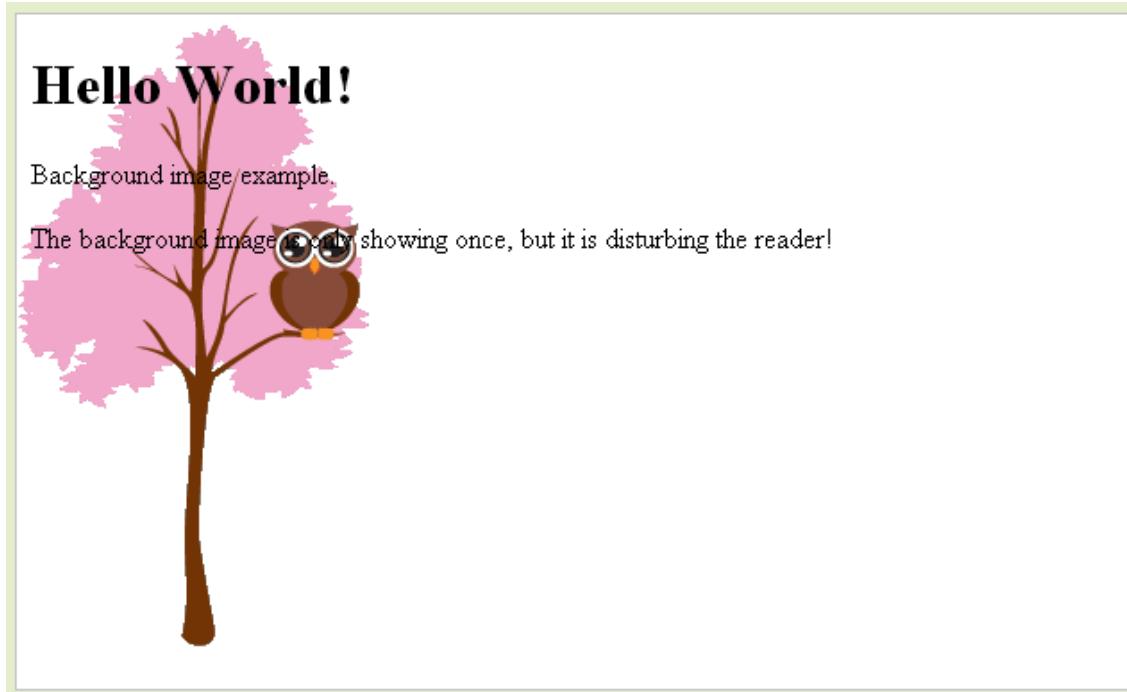
## Imagen de fondo - Ajustar la posición y no repetir-

💡 Cuando se utiliza una imagen de fondo, utiliza una imagen que no perturbe el texto.

Mostrar la imagen sólo una vez se especifica mediante la propiedad background-repeat:

### Ejemplo

```
body
{
background-image:url('img_tree.png');
background-repeat:no-repeat;
}
```



```
<!DOCTYPE html>
<html>
<head>
<style>
body
{
    background-image:url('img_tree.png');
    background-repeat: no-repeat;
}
</style>
</head>

<body>
<h1>Hello World!</h1>
<p>Ejemplo de imagen en el Background.</p>
<p>La imagen de fondo se muestra una sola vez, pero hace que el texto no se lea bien!</p>
</body>

</html>
```

En el ejemplo anterior, la imagen de fondo se muestra en el mismo lugar que el texto. Queremos cambiar la posición de la imagen, de manera que no moleste demasiado al texto.

La posición de la imagen se especifica mediante la propiedad `background-position`:

### Ejemplo

```
body
{
background-image:url('img_tree.png');
background-repeat:no-repeat;
background-position:right top;
}
```

## Hello World!

Background no-repeat, set position example.

Now the background image is only shown once, and positioned away from the text.

In this example we have also added a margin on the right side, so the background image will never disturb the text.



Vamos a separar un poco el texto de la imagen de fondo para que se pueda leer. Añadimos un margen para que el texto se separe de la parte derecha del navegador.

```
<!DOCTYPE html>
<html>
<head>

<style>
body
{
    background-image:url('img_tree.png');
    background-repeat:no-repeat;
    background-position:right top;
    margin-right:200px;
}
</style>

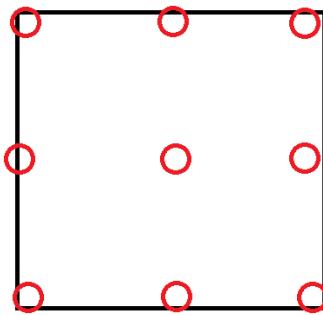
</head>

<body>
<h1>Hello World!</h1>
<p>Background no-repeat, set position example.</p>
<p>Now the background image is only shown once, and positioned away from the text.</p>
<p>In this example we have also added a margin on the right side, so the background image will never disturb the text.</p>
</body>

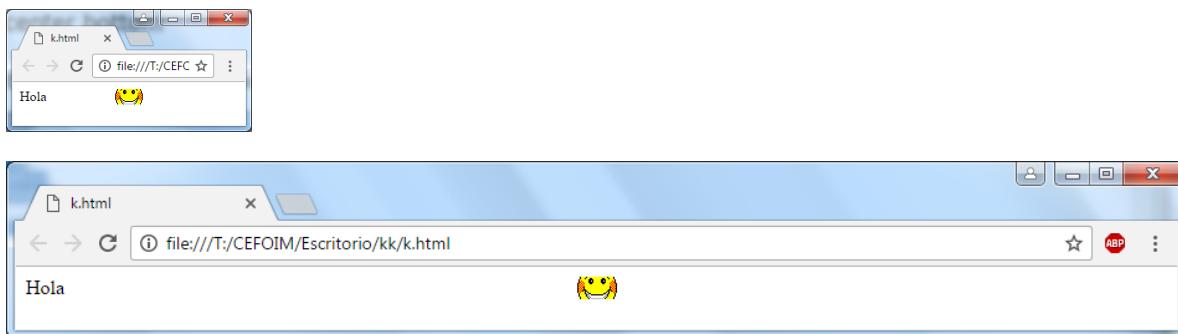
</html>
```

### Posibles valores

- left top
- left center
- left bottom
- right top
- right center
- right bottom
- center top
- center center
- center bottom



Ejemplo con porcentajes (no usar en el body vacío que se raya, meted algo)



```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    background-image: url('smiley.gif');
    background-repeat: no-repeat;
    background-position: 50% 50%; /*en este caso sería equivalente al center center */
}
</style>
</head>

<body>
<div>
Hola
</div>
</body>
</html>

O con pixels
background-position: 50px 150px;
```

### Tamaño del background

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url("img_flwr.gif");
    background-size: 80px 60px;
    background-repeat: no-repeat;
}
```

```
</style>
</head>
<body>

<p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat. Ut wisi enim ad minim veniam, quis nostrud exerci tation ullamcorper suscipit lobortis nisl ut aliquip ex ea commodo consequat.</p>
<p>Imagen original: </p>
</body>
</html>
```

### Ejemplo completo

```
<!DOCTYPE html>
<html>
<head>
<style>
#example1 {
border: 2px solid black;
background: url(mountain.jpg);
background-repeat: no-repeat;
}

#example2 {
border: 2px solid black;
background: url(mountain.jpg);
background-repeat: no-repeat;
background-size: 300px 100px;
}
</style>
</head>
<body>

<h2>background-size: auto (lo habitual): </h2>
<div id="example1">
<h2>Hola Mundo</h2>
<p>La imagen de fondo se muestra con el tamaño original.</p>
</div>

<h2>background-size: 300px 100px: </h2>
<div id="example2">
<h2>Hola Mundo</h2>
<p>La imagen de fondo se muestra con tamaño 300px de ancho y 100px de alto.</p>
</div>
</body>
</html>
```

Para establecer una sola propiedad (ancho o alto) para que la imagen no se deforme  
background-size: **100% 600px;**

<https://developer.mozilla.org/es/docs/Web/CSS/background-size>

## Background - Abreviando las propiedades

Como se puede ver en los ejemplos anteriores, hay muchas propiedades a tener en cuenta cuando se trata de fondos.

Para acortar el código, también es posible especificar todas las propiedades en una sola propiedad. Esto se llama una propiedad resumida.

La propiedad abreviada para el fondo no es más que "background":

### Ejemplo

```
body {background:#ffffff url('img_tree.png') no-repeat right top; }

<!DOCTYPE html>
<html>
<head>
<style>
body
{
    background:#ffffff url('img_tree.png') no-repeat right top;
    margin-right:200px;
}
</style>
</head>
<body>
<h1>Hello World!</h1>
<p>Now the background image is only shown once, and it is also positioned away from the text.</p>
<p>In this example we have also added a margin on the right side, so that the background image will not disturb the text.</p>
</body>

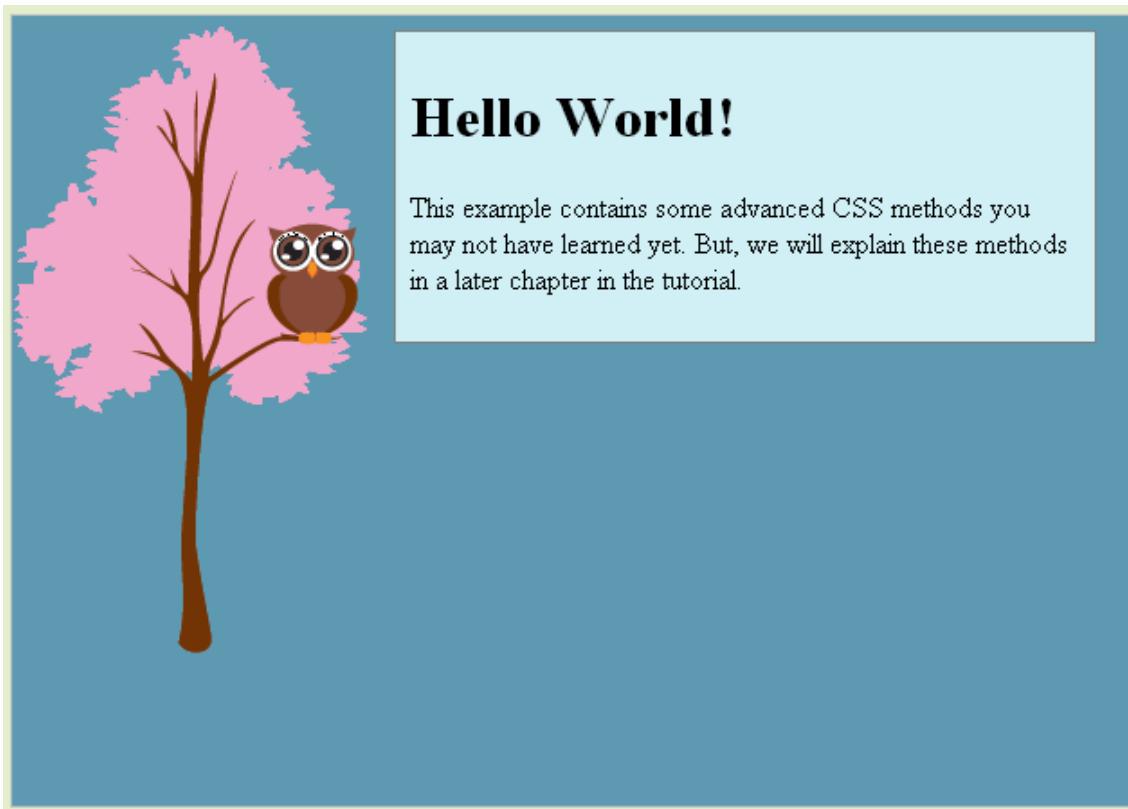
</html>
```

Cuando se utiliza la propiedad resumen el orden de los valores de la propiedad es:

- background-color
- background-image
- background-repeat
- background-attachment //Controla si la imagen de fondo se mueve o permanece fija cuando se hace scroll en la ventana del navegador
- background-position

No importa si uno de los valores de la propiedad está ausente, siempre que los que están presentes estén en este orden.

Este ejemplo usa CSS más avanzado.



```
<!DOCTYPE html>
<html>
<head>
<style>
body
{
    margin-left: 200px;
    background:#5d9ab2 url('img_tree.png') no-repeat top left;
}

.container
{
    text-align:center;
}

.center_div
{
    border: 1px solid gray;
    margin-left: auto;
    margin-right: auto;
    width: 90%;
    background-color: #d0f0f6;
    text-align:left;
    padding: 8px;
}
</style>
</head>

<body>
<div class="container">
<div class="center_div">
```

```
<h1>Hello World!</h1>
<p>Este ejemplo contiene algunos métodos Css avanzados que no conocemos todavía. Pero
que veremos mas tarde.</p>
</div>
</div>
</body>
</html>
```

Para que funcione hay que instalarse el plugin tilt 3d, una vez instalado se da a ctrl+shift+M  
<https://addons.mozilla.org/es/firefox/addon/tilt/>

Solo funciona en Firefox 47, en los siguientes ya no.

<https://ftp.mozilla.org/pub/firefox/releases/47.0/win32/es-ES/>

Probar con esto (año 2018)

<https://www.mozilla.org/en-US/firefox/developer/>

[https://developer.mozilla.org/es/docs/Tools/3D\\_View](https://developer.mozilla.org/es/docs/Tools/3D_View)



## Para activar la aceleración HW en Firefox

### Rendimiento

[Usar configuración de rendimiento recomendada](#) [Saber más](#)

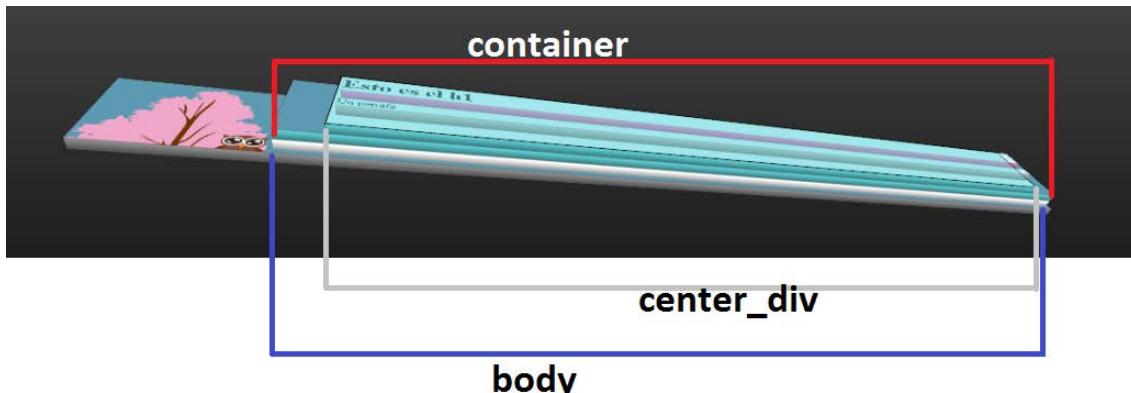
Esta configuración está ajustada al hardware y sistema operativo de su equipo.

[Usar aceleración de hardware cuando esté disponible](#)

Límite de procesos de contenido [8 \(predeterminado\)](#)

Más procesos de contenido pueden mejorar el rendimiento al usar múltiples pestañas, pero también usarán más memoria.

Seguimos sin poder hacer que funcione 2020



Añadimos un color de fondo al div de clase container y un texto para poder apreciar ambos contenedores



```
<!DOCTYPE html>
<html>
<head>
<style>
body
{
margin-left: 200px;
background: #5d9ab2 url('img_tree.png') no-repeat top left;
}

.container
{
background-color: #c0c0c0;
text-align: center;
}

.center_div
{
border: 1px solid gray;
margin-left: auto;
margin-right: auto;
width: 90%;
background-color: #d0f0f6;
text-align: left;
padding: 8px;
}
</style>
</head>

<body>
<div class="container">
Mi web
</div>
<div class="center_div">
Hola Mundo!
<small>Este ejemplo contiene algunos métodos avanzados de CSS</small>
</div>
</body>
```

```
<div class="center_div">
<h1>Hello World!</h1>
<p>This example contains some advanced CSS methods you may not have learned yet. But, we will explain these methods in a later chapter in the tutorial.</p>
</div>
</div>
</body>
</html>
```



## Más ejemplos

### Cómo establecer una imagen de fondo fija

este ejemplo muestra cómo establecer una imagen de fondo fija. La imagen no se desplazará con el resto de la página aunque movamos el scroll.

```
<!DOCTYPE html>
<html>
<head>
<style>
body
{
background-image:url('w3css.gif');
background-repeat:no-repeat;
background-attachment:fixed;
}
</style>
</head>
<body>
<p>La imagen de fondo está fija. mueve el scroll hacia abajo.</p>
<p>La imagen de fondo está fija. mueve el scroll hacia abajo.</p>
<p>La imagen de fondo está fija. mueve el scroll hacia abajo.</p>
<p>La imagen de fondo está fija. mueve el scroll hacia abajo.</p>
<p>La imagen de fondo está fija. mueve el scroll hacia abajo.</p>
<p>La imagen de fondo está fija. mueve el scroll hacia abajo.</p>
<p>La imagen de fondo está fija. mueve el scroll hacia abajo.</p>
<p>La imagen de fondo está fija. mueve el scroll hacia abajo.</p>
<p>La imagen de fondo está fija. mueve el scroll hacia abajo.</p>
<p>La imagen de fondo está fija. mueve el scroll hacia abajo.</p>
<p>La imagen de fondo está fija. mueve el scroll hacia abajo.</p>
<p>La imagen de fondo está fija. mueve el scroll hacia abajo.</p>
<p>La imagen de fondo está fija. mueve el scroll hacia abajo.</p>
<p>La imagen de fondo está fija. mueve el scroll hacia abajo.</p>
<p>La imagen de fondo está fija. mueve el scroll hacia abajo.</p>
<p>La imagen de fondo está fija. mueve el scroll hacia abajo.</p>
<p>La imagen de fondo está fija. mueve el scroll hacia abajo.</p>
</body>
</html>
```

## Bloque centrado con posicionamiento fijo

```
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

Mismo ejercicio pero con porcentajes

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <style>
#fcentrada {
position:fixed;
width:50%;
background:green;
/*lo puedo hacer con margin-left:50%; y left -25% pero no es necesario en este caso*/
left:25%;
top:50%;
}
</style>
    </head>
    <body>
        Hola mundo, hola mundo, hola mundo, hola mundo, hola mundo, hola mundo,
        hola mundo, hola mundo, hola mundo, hola mundo, hola mundo, hola mundo,
        hola mundo, hola mundo, hola mundo, hola mundo, hola mundo, hola mundo,
        hola mundo, hola mundo, hola mundo, hola mundo, hola mundo, hola mundo,
        hola mundo, hola mundo, hola mundo, hola mundo, hola mundo, hola mundo,
        <div id="fcentrada">Este es nuestro div centrado horizontalmente</div>
    </body>
</html>
```

## Todas las Propiedades de fondo CSS

Propiedad	Descripción
background	Establece todas las propiedades de fondo en una declaración
background-attachment	Establece si una imagen de fondo se fija o se desplaza con el resto de la página
background-color	Establece el color de fondo de un elemento
background-image	Establece la imagen de fondo de un elemento
background-position	Establece la posición inicial de una imagen de fondo
background-repeat	Establece una imagen de fondo se repetirá

En Css3

*background: bg-color bg-image position/bg-size bg-repeat bg-origin bg-clip bg-attachment;*

*bg-origin :The background positioning area – the part of a box that a background's position is in relation to.*

<http://htmldog.com/references/css/properties/background/>

**Note:** If one of the properties in the shorthand declaration is the bg-size property, you must use a / (slash) to separate it from the bg-position property, e.g. background:url(smiley.gif) 10px 20px/50px 50px; will result in a background image, positioned 10 pixels from the left, 20 pixels from the top, and the size of the image will be 50 pixels wide and 50 pixels high.

**Note:** If using multiple background-image sources but also want a background-color, the background-color parameter needs to be last in the list.

## Multiples Backgrounds Con CSS3

Utilizando **CSS3** podemos colocar **múltiples imágenes de fondo** a un elemento de html podría ser un div. Antes para conseguir esto teníamos que crear una sola composición con todas las imágenes o múltiples divs para conseguir este objetivo, con las nuevas y mejoradas propiedades de background de css3 conseguiremos ajustarlos a la necesidad presentada. Miremos como funciona.

Bueno para entender esto bien, necesitamos conocer la sintaxis.

**background-image:**  
url(sol.jpg), url(vaca.png), url(paisaje.jpg);

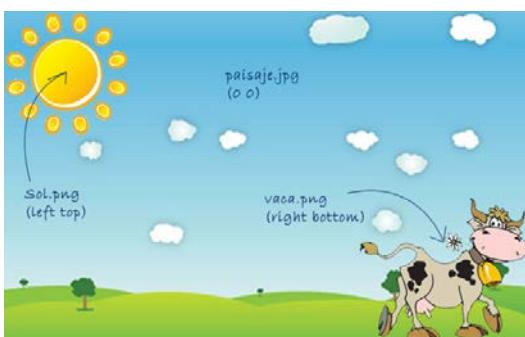
La declaración de la propiedad es la misma de la anterior versión de css, solo que ahora añadimos múltiples imágenes a la propiedad **background-image** y la separamos por medio de comas.

Podemos crear composiciones compuestas por varia imágenes y ubicarlas en diferentes posiciones para lograr lo deseado. Ejemplo, vamos a ver un

paisaje compuesto por 3 imágenes diferentes pero que vamos a integrar en un solo div utilizando múltiples imágenes de fondos en css3.

```
<html>
<head>
    <style text="text/css">
        div{
            height:600px;
            width: 835px;
            background-image: url('sol.png'), url('vaca.png'), url('paisaje.jpg');
            background-repeat: no-repeat;
            background-position: left top, right bottom, 0 0;
            background-size:25%,50%,150%;
        }
    </style>
</head>
<body>
    <div></div>
</body>
</html>
```

Tenemos este resultado:



Las posiciones que asignamos a cada imagen dependerán del orden en que tengamos las imágenes, desde la primera hasta la última.

Podemos abreviar este código de css para aplicar múltiples fondos de imágenes de la siguiente forma.

<http://canciones-infantiles.com.es/wp-content/uploads/2013/09/dibujo-vaca.png>

<https://s-media-cache-ak0.pinimg.com/564x/25/0d/59/250d5973b1cb104700b0c9ee3dcf4e18.jpg>

[https://cdn.pixabay.com/photo/2014/08/04/07/24/landscape-409551\\_960\\_720.png](https://cdn.pixabay.com/photo/2014/08/04/07/24/landscape-409551_960_720.png)

```
div{
background:
url(sol.png) left top no-repeat,
url(vaca.png) right bottom no-repeat,
```

```
url(paisaje.jpg) 0 0 no-repeat;  
}
```

y el resultado sera el mismo.

Video de background



El video está aquí

<https://www.w3schools.com/howto/rain.mp4>

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {
```

```
    margin: 0;
```

```
    font-family: Arial;
```

```
    font-size: 17px;
```

```
}
```

```
#miVideo {
```

```
    position: fixed;
```

```
    left: 0;
```

```
    top: 0;
```

```
    min-width: 100% /*para que el vídeo me ocupe todo el ancho del navegador  
*/
```

```
    min-height: 100% /*para que el vídeo me ocupe todo el alto del navegador */
```

}

```
.content {  
    position: fixed;  
    top: 50px;  
    background: rgba(0, 0, 0, 0.5);  
    color: #f1f1f1;  
    width: 50%;  
    /*margin-left: 50%; */  
    left: 25%;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<video autoplay muted loop id="miVideo">  
<source src="rain.mp4" type="video/mp4">  
Tu navegador no soporta la etiqueta HTML5 video.  
</video>
```

```
<div class="content">  
<h1>Mi Título</h1>  
<p>Lorem ipsum dolor sit amet, an his etiam torquatos. Tollit soleat phaedrum  
te
```

duo, eum cu recteque expetendis neglegentur. Cu mentitum maiestatis  
persequeris

pro, pri ponderum tractatos ei. Id qui nemore latine molestiae, ad mutat oblique

delicatissimi pro.</p>  
</div>  
</body>  
</html>

efecto parallax

<http://www.paellacreativa.com.ar/2012/05/24/efecto-parallax-diseno-web/>  
<https://freefrontend.com/css-parallax/>  
<https://codepen.io/dazulu/pen/VVZrQv>

## Crear un efecto de fondo enmascarado con CSS

<http://tutsplus.github.io/create-a-masked-background-effect-with--background-attachment--fixed--/images/minipadwhite.png>  
<http://tutsplus.github.io/create-a-masked-background-effect-with--background-attachment--fixed--/images/minipadblack.png>  
<http://tutsplus.github.io/create-a-masked-background-effect-with--background-attachment--fixed--/images/miniwhite.png>  
<http://tutsplus.github.io/create-a-masked-background-effect-with--background-attachment--fixed--/images/miniblack.png>

Versión reducida

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Ejemplo background-attachment: fixed;</title>
<meta charset="utf-8">
<link href="style.css" rel="stylesheet" type="text/css">
</head>
<body>
<div class="content right illustration_01">
<p>hola a todos</p>
<p>hola a todos</p>
<p>hola a todos</p>
<p>hola a todos</p>
```





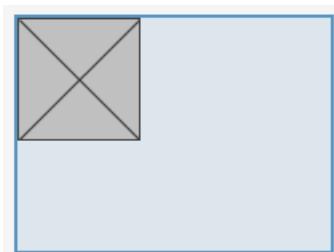
```
</div>
</body>
</html>
Css
html {
    background-color: #262626;
    color: #d9d9d9;
}
body {
    margin: 0;
}
.content {
    font-size: 1.875em;
    color: #262626;
    background-size: 49% auto;
    background-attachment: fixed;
    background-repeat: no-repeat;
}
.right {
    padding: 1.618em 6.472em 3.236em 50%;
    background-position: 0 50%; /*se coloca en el punto 0 50% */
}
.illustration_01 {
    background-color: #00c17b;
    background-image: url("./minipadwhite.png");
}
.separator {
    font-size: 1.875rem;
    padding: 1.618em 0;
    text-align: center;
}
.left {
```

```
padding: 1.618em 50% 3.236em 6.472em;  
background-position: 100% 50%; /*se coloca en el punto 100% 50% */
```

```
}
```

```
.illustration_02 {  
background-color: #e8697b;  
background-image: url("./minipadblack.png");  
}
```

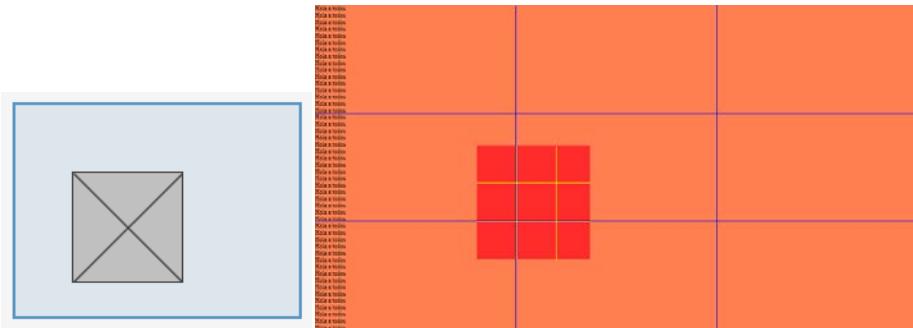
La propiedad background-position también permite el uso de porcentajes para establecer la posición de la imagen de fondo. Cuando se indican dos porcentajes (x% y%) se considera que el punto x% - y% de la imagen de fondo se debe colocar en el punto x% - y% del elemento. Los siguientes ejemplos muestran este comportamiento de forma práctica:



```
div {  
background-image:url("images/imagen_pequena.png");  
background-repeat:no-repeat;  
background-position:0%0%;  
}
```

```
div {  
background-image:url("images/imagen_pequena.png");  
background-repeat:no-repeat;  
background-position:100%100%;  
}
```

En el primer ejemplo, el punto 0% 0% de la imagen de fondo (es decir, su esquina superior izquierda) se coloca en el punto 0% 0% del elemento (también su esquina superior izquierda). El segundo ejemplo, el punto 100% 100% de la imagen de fondo (es decir, su esquina inferior derecha) se coloca en el punto 100% 100% del elemento (también su esquina inferior derecha).



```
div {
    background-image: url( "images/imagen_pequena.png" );
    background-repeat: no-repeat;
    background-position: 33% 66%;
}

div {
    background-image: url( "images/imagen_pequena.png" );
    background-repeat: no-repeat;
    background-position: 50% 50%;
}
```

En el primer ejemplo, el punto 33% 66% de la imagen de fondo (es decir, el punto que está a un tercio de su anchura desde la parte izquierda y a dos tercios de su altura desde la parte superior) se coloca en el punto 33% 66% del elemento (un tercio de su anchura desde la parte izquierda y dos tercios de su altura desde la parte superior). El segundo ejemplo, el punto 50% 50% de la imagen de fondo (es decir, justo su centro) se coloca en el punto 50% 50% del elemento (también justo su centro).

Palabra clave	Valor equivalente
top	0% en la posición vertical

Palabra clave	Valor equivalente
right	100% en la posición horizontal
bottom	100% en la posición vertical
left	0% en la posición horizontal
center	50% en la posición horizontal si no se ha establecido la posición horizontal de otra forma. En otro caso, 50% en la posición vertical

Si bien la propiedad funciona de maravilla en todos los navegadores de escritorio, no funciona correctamente con la versión móvil de Chrome y puede ser un poco distinto en los otros navegadores móviles. Así, mientras los visitantes ven las imágenes bien, el efecto de este desplazamiento puede verse mucho mejor en los navegadores de escritorio.

## Cómo se hace

Los pasos básicos para lograr tal efecto como se ve en la versión online son:

1. Crear un elemento contenedor y agregar el contenido a él.
2. Ajusta el contenedor (un `div` en nuestro caso) para tener padding en un lado alrededor del 50% de anchura, y así empuje el contenido hacia el otro lado.
3. Agrega una imagen de fondo, también con un 50% de anchura, y posícnala en el lado contrario al contenido.
4. Ajusta la propiedad `background-attachment: fixed;` y ve la magia del desplazamiento.

Vamos a ver paso a paso cómo hacer todo esto. Necesitarás [los archivos iniciales](#) para este tutorial con las imágenes requeridas.

## 1. Configuración Básica

Empieza por crear la carpeta del proyecto, y agrega un archivo **index.html** dentro, también una carpeta **css** con un archivo de nombre **style.css** y guárdalo.

El div contenedor tiene tres clases:

1. **.content** - usada para establecer algunas propiedades que serán comunes en todos los demás contenedores.
2. **.right** - identifica que este contenedor deberá tener el contenido alineado a la derecha (también vamos a trabajar con un contenedor alineado a la izquierda más adelante)
3. **.illustration\_01** - usado para establecer la imagen de fondo y el color para este contenedor

En la clase **.content**:

```
.content {  
  
    font-size: 1.875rem;  
  
    color: #262626;  
  
    background-size: 49%auto;  
  
    background-attachment: fixed;  
  
    background-repeat: no-repeat;  
  
}
```

En esta clase es donde la mayor parte de la magia sucede. Para el texto, establecemos nuestro tamaño de fuente y color. Para el fondo como verás empezamos por fijar la propiedad **background-size** a un 49% auto.

Esto significa que la imagen de fondo siempre se reducirá o estirará para llenar el 49% del ancho de la página, y la altura se ajustará proporcionalmente. Estamos usando un valor de 49% en lugar de 50% porque de lo contrario Firefox muestra una línea rara en el centro de la pantalla.

A continuación establecemos **background-fixed** que, como usted ya sabe de la descripción anterior, hace que la imagen de fondo se quede fija cuando nos desplazamos, y hace que se posicione con respecto a la ventana en lugar del contenedor al que se aplica.

Finalmente fijamos **background-repeat: no-repeat;** para asegurarnos que nuestra imagen aparezca solo una vez en la página.

A continuación agrega la clase **.right** a tu hoja de estilos:

```
.right{
```

```
padding: 1.618em 6.472em 3.236em 50%;  
background-position: 0 50%;  
}
```

Esta última clase coloca el contenido del texto en una mitad de la pantalla y la imagen de fondo en otra.

La propiedad `background-position` nos dice que la posición de la imagen de fondo será de cero píxeles desde el lado izquierdo de la ventana y alineado a la mitad del mismo hacia abajo desde la parte superior de la ventana.

Por último agrega la clase `.illustration_01`:

```
.illustration_01{  
background-color: #00c17b;  
background-image: url("../images/minipadwhite.png");  
}
```

Esto especifica la imagen de fondo y el color que queremos para el contenido de este contenedor.2. Agrega un Segundo Contenedor

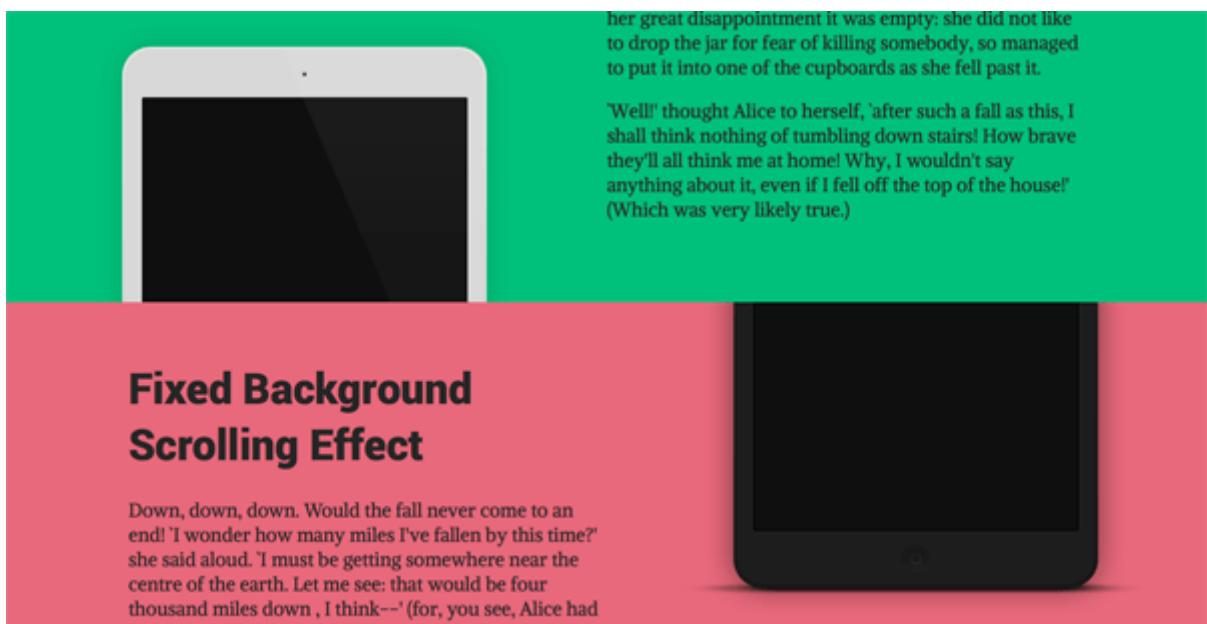
Usamos la clase `.left` en lugar de `.right` y hemos aumentado el número de ilustración en la clase `.illustration_01` por `.illustration_02`

Agrega las siguientes dos clases a tu hoja de estilos:

```
.left{  
padding: 1.618em 50% 3.236em 6.472em;  
background-position: 100% 50%;  
}  
.illustration_02{  
background-color: #e8697b;  
background-image: url("../images/minipadblack.png");  
}
```

Esta vez hemos aplicado el 50% de padding del lado derecho del contenedor de manera que el contenido sea empujado hacia la izquierda, y el fondo está posicionado 100% horizontalmente, es decir, todo a la derecha. También vamos a agregar un color y una imagen de fondo diferente a este contenedor.

Revisa de nuevo tu sitio y empieza por desplazarte hacia abajo. Cuando encuentres el final del primer contenedor deberías empezar a ver el segundo venir, desapareciendo la primer imagen en la parte superior y revelando poco a poco la segunda.



### 3. Agrega un Separador

Sería mejor el efecto de esta técnica si agregamos un separador entre los dos contenedores, así que lo vamos aadir ahora.

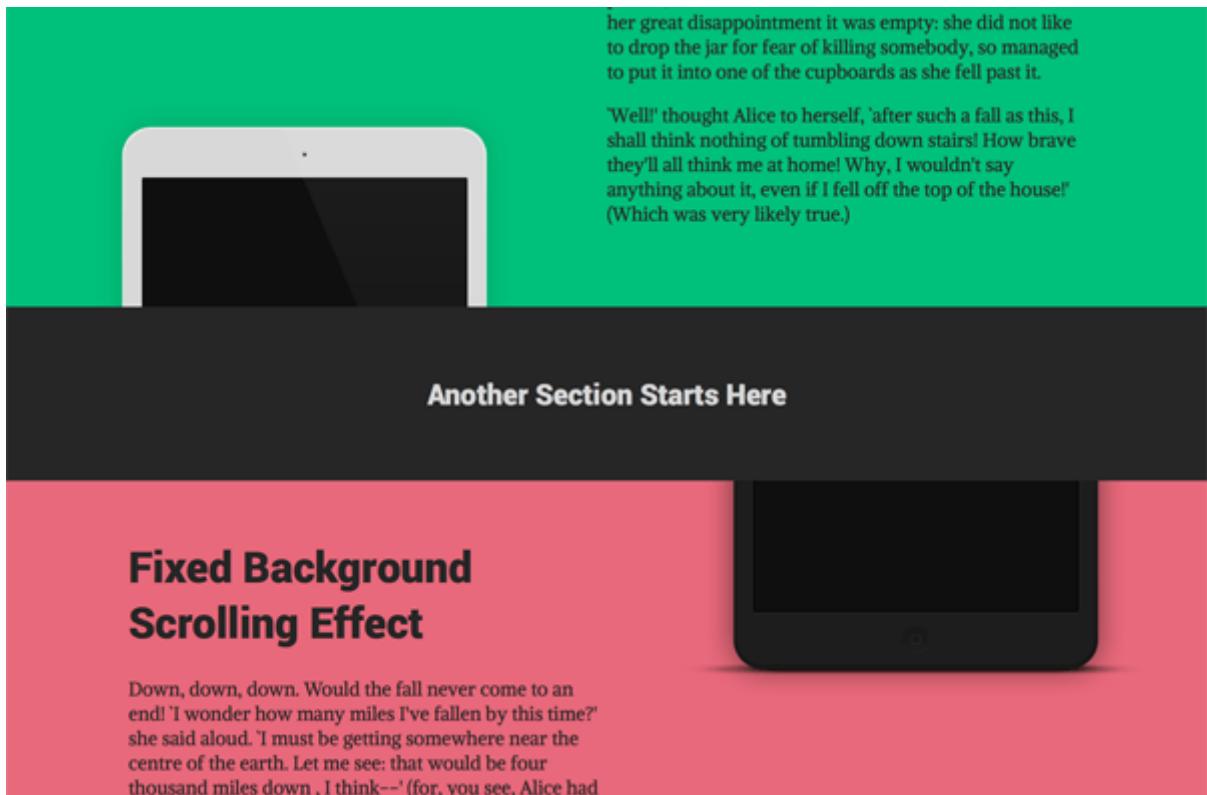
En medio de los dos div contenedores agrega este HTML:

```
<section class="separator">
  <h3>Another Section Starts Here</h3>
</section>
```

Y agrega la clase .separator a tu hoja de estilos:

```
.separator {
  font-size: 1.875rem;
  padding: 1.618em 0;
  text-align: center;
}
```

Cuando actualices tu sitio ahora deberías tener un separador entre tus contenedores:



<http://tutsplus.github.io/create-a-masked-background-effect-with--background-attachment--fixed--/images/minipadwhite.png>  
<http://tutsplus.github.io/create-a-masked-background-effect-with--background-attachment--fixed--/images/minipadblack.png>  
<http://tutsplus.github.io/create-a-masked-background-effect-with--background-attachment--fixed--/images/miniwhite.png>  
<http://tutsplus.github.io/create-a-masked-background-effect-with--background-attachment--fixed--/images/miniblack.png>

Ejercicio de background con video y botones

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
body {
```

```
    margin: 0;
```

```
    font-family: Arial;
```

```
font-size: 17px;  
}  
  
  
#miVideo {  
    position: fixed;  
    right: 0;  
    bottom: 0;  
    min-width: 100% /*para que el vídeo me ocupe todo el ancho del navegador */  
    min-height: 100% /*para que el vídeo me ocupe todo el alto del navegador */  
  
}  
  
  
.content {  
    position: fixed;  
    bottom: 0;  
    background: rgba(0, 0, 0, 0.5);  
    color: #f1f1f1;  
    width: 100%;  
    padding: 20px;  
}  
  
  
#miBtn {  
    width: 200px;  
    font-size: 18px;  
    padding: 10px;  
    border: none;  
    background: #000;  
    color: #fff;  
    cursor: pointer;
```

}

```
#miBtn:hover {  
    background: #ddd;  
    color: black;  
}  
</style>  
</head>  
<body>  
  
<video autoplay muted loop id="miVideo">  
<source src="rain.mp4" type="video/mp4">  
Tu navegador no soporta la etiqueta HTML5 video.  
</video>  
  
<div class="content">  
    <h1>Mi Título</h1>  
    <p>Lorem ipsum dolor sit amet, an his etiam torquatos. Tollit soleat phaedrum te duo, eum cu recteque expetendis neglegentur. Cu mentitum maiestatis persequeris pro, pri ponderum tractatos ei. Id qui nemore latine molestiae, ad mutat oblique delicatissimi pro.</p>  
    <button id="miBtn" onclick="miFuncion()">Pausa</button>  
</div>  
<script>  
var video = document.getElementById("miVideo");  
var btn = document.getElementById("miBtn");  
function miFuncion() {  
    if (video.paused) {  
        video.play();  
        btn.innerHTML = "Pausa";  
    } else {  
        video.pause();  
        btn.innerHTML = "Reproducir";  
    }  
}
```

```
video.pause();  
btn.innerHTML = "Iniciar";  
}  
}  
</script>  
  
</body>  
</html>
```

## CSS Text

### Color del texto

La propiedad color se utiliza para establecer el color del texto.

Con CSS, un color se especifica con mayor frecuencia a través de:

- un valor HEX - como "#ff0000"
- un valor RGB - como "rgb(255,0,0)"
- un nombre de color - como "red"

El color por defecto para una página se define en el selector body.

### Ejemplo

```
body {color:blue;}  
h1 {color:#00ff00;}  
h2 {color:rgb(255,0,0);}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
body {color:red;}  
h1 {color:#00ff00;}  
p.ex {color:rgb(0,0,255);}  
</style>  
</head>  
  
<body>  
<h1>This is heading 1</h1>  
<p>This is an ordinary paragraph. Notice that this text is red. The default text-color for a  
page is defined in the body selector.</p>  
<p class="ex">This is a paragraph with class="ex". This text is blue.</p>  
</body>  
</html>
```

W3C indica que: Si define la propiedad de color, también debe definir la propiedad background-color.

## Alineación del texto

La propiedad text-align se utiliza para establecer la alineación horizontal del texto.

El texto puede ser centrado, alineado a la izquierda, a la derecha, o justificado.

Cuando el valor para text-align es "justificar", cada línea se estira para que cada línea tenga la misma anchura y los márgenes izquierdo y derecho sean rectos (como en revistas y periódicos).

### Ejemplo

```
h1 {text-align:center;}  
p.date {text-align:right;}  
p.main {text-align:justify;}
```

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1 {text-align:center;}  
p.date {text-align:right; }  
p.main {text-align:justify; }  
</style>  
</head>  
  
<body>  
<h1>CSS text-align Example</h1>  
<p class="date">May, 2009</p>  
<p class="main">In my younger and more vulnerable years my father gave me some advice  
that I've been turning over in my mind ever since. 'Whenever you feel like criticizing  
anyone,' he told me,  
'just remember that all the people in this world haven't had the advantages that you've  
had.'</p>  
<p><strong>Note:</strong> Resize the browser window to see how the value "justify"  
works.</p>  
</body>  
  
</html>
```

## Decoración del Texto

La propiedad text-decoration se utiliza para establecer o quitar las decoraciones de texto.

La propiedad text-decoration se utiliza sobre todo para eliminar el subrayado de los hipervínculos (y aplicarle luego otros diseños):

### Ejemplo

```
a {text-decoration:none; }

<!DOCTYPE html>
<html>
<head>
<style>
a {text-decoration:none; }
</style>
</head>

<body>
<p>Link to: <a href="http://www.google.com">Google.com</a></p>
</body>

</html>
```

También se puede utilizar para decorar texto:

### Ejemplo

```
h1 {text-decoration:overline; }
h2 {text-decoration:line-through; }
h3 {text-decoration:underline; }
h4 {text-decoration:blink; }
```

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {text-decoration:overline; }
h2 {text-decoration:line-through; }
h3 {text-decoration:underline; }
h4 {text-decoration:blink; }
</style>
</head>
<body>
    <h1>This is heading 1</h1>
    <h2>This is heading 2</h2>
    <h3>This is heading 3</h3>
    <h4>This is heading 4</h4>
<p><strong>Note:</strong> The "blink" value is not supported in IE, Chrome, Firefox, o
Safari</p>
</body>
</html>
```

 No se recomienda subrayar texto que no sea un enlace, ya que a menudo confunde a los usuarios.

## Transformación de Texto

La propiedad text-transform se utiliza para especificar letras mayúsculas y minúsculas en un texto.

Se puede utilizar para convertir todo en letras mayúsculas o minúsculas, o mayúscula la primera letra de cada palabra.

### Ejemplo

```
p.uppercase {text-transform:uppercase;}  
p.lowercase {text-transform:lowercase;}  
p.capitalize {text-transform:capitalize;}
```

Este Es Un Texto Capitalizado

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.uppercase {text-transform:uppercase;}  
p.lowercase {text-transform:lowercase;}  
p.capitalize {text-transform:capitalize;}  
</style>  
</head>  
  
<body>  
<p class="uppercase">This is some text.</p>  
<p class="lowercase">This is some text.</p>  
<p class="capitalize">This is some text.</p>  
</body>  
</html>
```

## Sangría del Texto

La propiedad text-indentación se utiliza para especificar la sangría de la primera línea de un texto.

### Ejemplo

```
p {text-indent:50px;}
```

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
    p {text-indent: 50px; }  
</style>  
</head>  
<body>  
  
<p>In my younger and more vulnerable years my father gave me some advice that I've  
been turning over in my mind ever since. 'Whenever you feel like criticizing anyone,' he told  
me, 'just remember that all the people in this world haven't had the advantages that you've  
had.'</p>  
  
</body>  
</html>
```



## Más ejemplos

### Especificar el espacio entre caracteres

Este ejemplo demuestra cómo aumentar o disminuir el espacio entre caracteres.

# This is heading 1

## This is heading 2

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {letter-spacing: 2px; }
h2 {letter-spacing: -3px; }
</style>
</head>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
</body>
</html>
```

### Especificando el interlineado

En este ejemplo se muestra cómo especificar el espacio entre las líneas de un párrafo.

This is a paragraph with a standard line-height.

This is a paragraph with a standard line-height.

The default line height in most browsers is about 110% to 120%.

This is a paragraph with a smaller line-height.  
This is a paragraph with a smaller line-height.  
This is a paragraph with a smaller line-height.  
This is a paragraph with a smaller line-height.

This is a paragraph with a bigger line-height.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.small {line-height: 70%; }
p.big {line-height: 200%; }
```

```
</style>
</head>
<body>
<p>
This is a paragraph with a standard line-height.<br/>
This is a paragraph with a standard line-height.<br/>
The default line height in most browsers is about 110% to 120%.<br/>
</p>

<p class="small">
This is a paragraph with a smaller line-height.<br/>
</p>

<p class="big">
This is a paragraph with a bigger line-height.<br/>
</p>

</body>
</html>
```

#### Ajuste de la posición del texto de un elemento

En este ejemplo se muestra cómo cambiar la dirección del texto de un elemento

```
<!DOCTYPE html>

<html>

<head>
<style>

div.ex1 {direction:rtl; }

</style>

</head>

<body>

<div>Some text. Default writing direction.</div>

<div class="ex1">Some text. Right-to-left direction.</div>

</body>

</html>
```



#### Aumentar el espacio en blanco entre las palabras

En este ejemplo se muestra cómo aumentar el espacio en blanco entre las palabras en un párrafo.

```
<!DOCTYPE html>
<html>
<head>
<style>
p
{
    word-spacing: 30px;
}
</style>
</head>
<body>

<p>
This is some text. This is some text.
</p>

</body>
</html>
```

#### Desactive el ajuste de texto dentro de un elemento.

En este ejemplo se muestra cómo deshabilitar el ajuste de texto dentro de un elemento.(no cambia de renglón cuando llega al final y por lo tanto nos aparece un scroll vertical)



(Mas abajo tendremos ejemplos donde veremos las distintas utilidades de esta regla)

```
<!DOCTYPE html>
<html>
<head>
<style>
p
{
    white-space: nowrap;
}
</style>
</head>
<body>
```

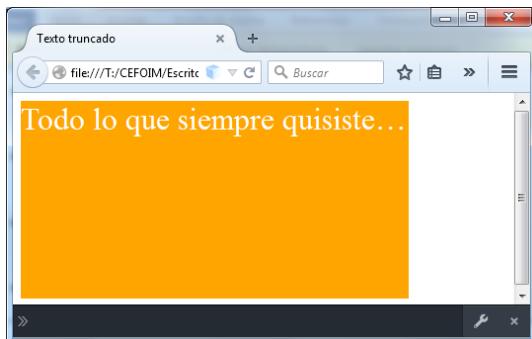
```
<p>
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
</p>
</body>
</html>
```

### Ejemplo con tabla

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>

<table width="400" border="1" cellpadding="10" cellspacing="0" bordercolor="#000000">
<tr>
<td nowrap>Aunque este texto sea más ancho que los 400 píxeles de la tabla, está no puede dividir mediante saltos de línea el contenido de la misma, por lo que se estira para albergar toda la frase.</td>
</tr>
</table>
</body>
</html>
```

Poniendo tres puntos en un texto que se corta (css3)



```
<!doctype html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Texto truncado</title>
<style type="text/css">
    #contenido {
        width: 80%;
        height: 200px;
        background-color: orange;
        color: white;
        font-size: 24pt;
        overflow: hidden;
        white-space: nowrap;
        text-overflow: ellipsis;
    }
</style>
```

```
</head>
<body>
<div id="contenido">
    Todo lo que siempre quisiste saber sobre HTML, CSS y JavaScript y no
    te atreviste a preguntar.
</div>
</body>
</html>
```

Podríamos añadir un una animación que quite ese nowrap cuando pasemos el ratón por encima

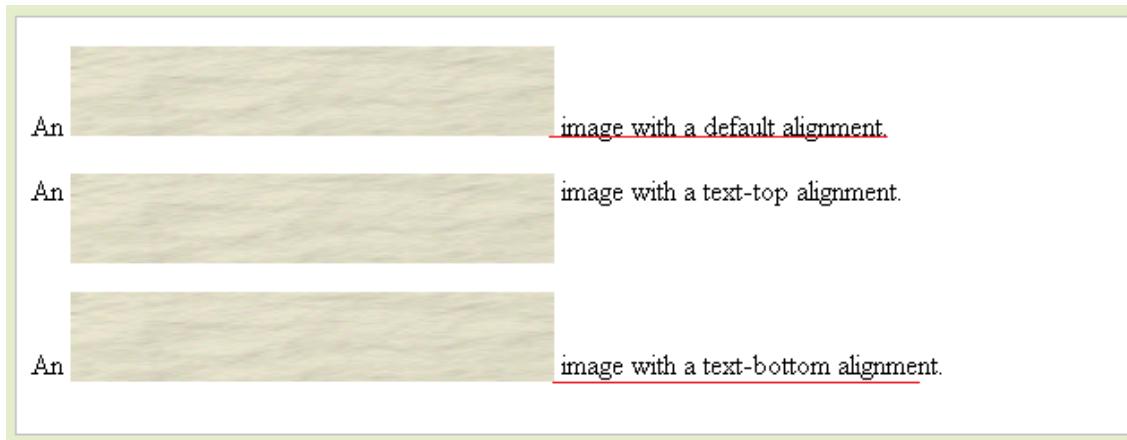
```
<!DOCTYPE html>
<html>
<head>
<style>
div.test {
    white-space: nowrap;
    width: 12em;
    overflow: hidden;
    border: 1px solid #000000;
}
div.test:hover {
    white-space: normal;
}
</style>
</head>
<body>
<p>Pasa el ratón por encima de los párrafos</p>
<div class="test" style="text-overflow:ellipsis;">Esto es un texto largo con
mucho bla, bla, bla, bla, bla, bla, bla, </div>
<br>
<div class="test" style="text-overflow:clip;">Esto es un texto largo con mucho
bla, bla, bla, bla, bla, bla</div>
<div class="test" style="text-overflow:'. Leer mas...'">Esto es un texto largo con
mucho bla, bla, bla, bla, bla, bla, </div>
</body>
</html>
```

Yo en el text-overflow puedo poner el texto que yo quiera  
text-overflow: ". Leer mas...";  
/\* No soportado en muchos navegadores, en firefox 9 tendría que funcionar \*/  
(html no admite o eso creo)

<https://leaverou.github.io/animatable/#border-width>  
(No sé porqué pero en la web no funciona, si se descarga sí)  
<https://github.com/leaverou/animatable>

### La alineación vertical de una imagen

En este ejemplo se muestra cómo establecer la alineación vertical de una imagen en un texto.



```
<!DOCTYPE html>
<html>
<head>
<style>
img.top { vertical-align:text-top; }
img.bottom { vertical-align:text-bottom; }
</style>
</head>

<body>
<p>An  image with a default alignment.</p>
<p>An  image with a text-top alignment.</p>
<p>An  image with a text-bottom alignment.</p>
</body>
</html>
```

También hay vertical-align: middle;

Text shadows

## Hola a todos

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
    text-shadow: 2px 2px #FF0000;
}
</style>
</head>
<body>

<h1>Hola a todos</h1>

</body>
</html>
```

## Todas las propiedades del texto CSS

Propiedad	Descripción
color	Establece el color del texto
direction	Especifica la dirección del texto / escritura dirección
letter-spacing	Aumenta o disminuye el espacio entre caracteres en un texto
line-height	Ajusta la altura de la línea
text-align	Especifica la alineación horizontal del texto
text-decoration	Especifica la decoración añadida al texto
text-indent	Especifica la sangría de la primera línea de un bloque de texto
text-shadow	Especifica el efecto de sombra añadido al texto
text-transform	Controla la capitalización del texto
unicode-bidi	
vertical-align	Establece la alineación vertical de un elemento
white-space	Especifica cómo un espacio en blanco dentro de un elemento se maneja
word-spacing	Aumenta o disminuye el espacio entre las palabras de un texto

Alineaciones mas complejas (no ver de momento)

<http://css-tricks.com/almanac/properties/v/vertical-align/>

## CSS Fuente

Las propiedades CSS de fuente definen la familia de fuentes, el grosor, el tamaño y el estilo de un texto.

## Diferencia entre Serif y fuentes Sans-serif



En las pantallas de ordenador, los textos sans-serif se consideran más fáciles de leer que las fuentes serif.

## Familias de fuentes CSS

En CSS, hay dos tipos de familias de fuentes:

- **Familia genérica** - un grupo de familias de fuentes con un aspecto similar (como "Serif" o "Monospace")
- **Familia tipográfica** - una familia de fuentes específicas (como "Times New Roman" o "Arial")

Familia genérica	Familia de fuentes	Descripción
Serif	Times New Roman <b>Georgia</b>	Las fuentes serif tienen pequeñas líneas en los extremos de algunos caracteres
Sans-serif	<b>Arial</b> Verdana	"Sans" significa sin - estas fuentes no tienen las líneas en los extremos de los caracteres
Monospace	<b>Courier New</b> <b>Lucida Console</b>	Todos los caracteres Monospace tienen el mismo ancho

## Font Family

La familia de fuentes de un texto se establece con la propiedad font-family.

La propiedad font-family debería soportar varios tipos de fuentes. Si el navegador no es compatible con la primera fuente, intenta con la siguiente fuente.

Coloca en primera posición la fuente que deseas, y termina con una familia genérica, para que el navegador elija un tipo de letra similar en la familia genérica, si no hay otras fuentes disponibles.

**Nota :** Si el nombre de una familia de fuentes tiene más que una palabra, debe estar entre comillas, como font-family: "Times New Roman".

Se especifica más de una familia de fuentes en una lista separada por comas:

### Ejemplo

```
p{font-family: "Times New Roman", Times, serif;}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
p.serif{font-family:"Times New Roman",Times,serif; }
p.sansserif{font-family:Arial,Helvetica,sans-serif; }
</style>
</head>

<body>
<h1>CSS font-family</h1>
<p class="serif">This is a paragraph, shown in the Times New Roman font.</p>
<p class="sansserif">This is a paragraph, shown in the Arial font.</p>

</body>
</html>
```

## Listado Fuentes "seguras"

<http://www.elwebmaster.com/articulos/fuentes-seguras-para-la-web-como-y-cuales-son>

## Font Style

La propiedad font-style se utiliza sobre todo para especificar texto en cursiva.

Esta propiedad tiene tres valores:

- normal - El texto se muestra normalmente
- cursiva - El texto se muestra en cursiva
- oblicuo - El texto se "inclina" (oblicua es muy similar a la cursiva, pero está menos soportado, de hecho yo no veo diferencia)

En imprenta, si se vería la diferencia, en Chrome también se ve...



This font is **Italic**  
This font is **Oblique**  
This font is **Normal**

## Ejemplo

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

```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {font-style:normal; }
p.italic {font-style:italic; }
p.oblique {font-style:oblique; }
```

```
</style>
</head>

<body>
<p class="normal">This is a paragraph, normal.</p>
<p class="italic">This is a paragraph, italic.</p>
<p class="oblique">This is a paragraph, oblique.</p>
</body>

</html>
```

Podemos usar dos clases a la vez haciendo:

```
<div class="class1 class2">
    Un texto cualquiera
</div>
```

Ejemplo

```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {font-style: normal; }
p.italic {font-style: italic; }
p.oblique {font-style: oblique; }
p.serif{ font-family: "Times New Roman", Times, serif; }
</style>
</head>

<body>
<p class="normal">This is a paragraph, normal.</p>
<p class="italic">This is a paragraph, italic.</p>
<p class="oblique">This is a paragraph, oblique.</p>

<p class="oblique serif">
Un texto cualquiera
</p>
</body>

</html>
```

## Google Fonts en tu página web

Google nos ha brindado muchas cosas útiles y prácticas en este nuevo universo del internet, el mejor motor de búsqueda, un sistema operativo muy bueno para los teléfonos, Google Maps, la lista es larga. Ahora, uno de esos productos que uso mucho últimamente es [Google Fonts](#). Más de 600 tipos de letras disponibles. Totalmente gratis, para que puedas usar y adornar o arreglar tu página web.

### Usando Google Fonts. La Explicación Rápida

En el encabezado del documento HTML vas a añadir la siguiente línea:

```
<link rel="stylesheet" type="text/css" href="http://fonts.googleapis.com/css?family=Rokkitt">
```

Luego, en tu hoja de estilos vas a especificar la fuente (tipo de letra) que deseas usar, así:

```
<style> body { font-family: 'Rokkitt'; } </style>
```

## Como usar Google Fonts en tu Página Web

```
<!DOCTYPE html>
```

```
<html>
<head>
<link rel="stylesheet" type="text/css"
      href="http://fonts.googleapis.com/css?family=Rokkitt">
<style>
body
{
    font-family: 'Rokkitt';
    font-size: 15px;
}
</style>
</head>
<body>
Hola Don Pepito
</body>
</html>
```

<https://fonts.google.com/>

Mas ejemplos

**change Pacifico Lobster Gloria Hallelujah**

Otro ejemplo

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css"
      href="http://fonts.googleapis.com/css?family=Gloria+Hallelujah">
<style>
body { font-family: 'Gloria Hallelujah'; }
</style>
</head>
<body>
Gloria Hallelujah
```

</body>

</html>

Otro

```
<link rel="stylesheet" type="text/css" href="http://fonts.googleapis.com/css?family=Chango">
```

```
<div>Usando una fuente tipográfica de Google Fonts</div>
```

```
<!-- ### Puedes cambiar el nombre de la fuente a: ### -->
```

```
<!-- ### Hanalei, Ribeye, Fondamento, etc... ##### -->
```

```
<!-- ### No olvides que el nombre tambien lo debes cambiar en el CSS ### -->
```

```
<!-- Para mas fuentes ve a google.com/fonts -->
```

Fichero CSS

```
@charset "UTF-8";
```

```
/* CSS Document */
```

```
body {
```

```
    background-color:white;
```

```
    font-family: 'Chango', serif;
```

```
    font-size: 48px;
```

```
}
```

/\* Puedes cambiar el nombre de la fuente a:

Hanalei, Ribeye, Fondamento, etc... \*/

/\* ##### Si cambias el nombre de la fuente aquí debes de cambiarlo tambien en el archivo HTML ### \*/

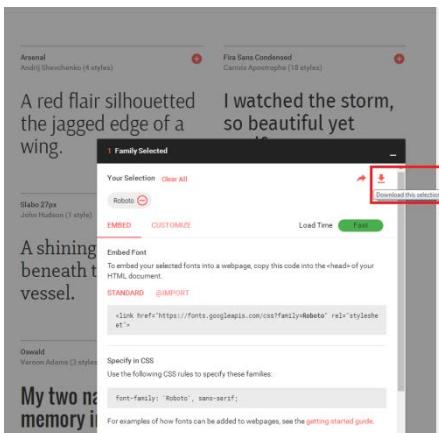
Algunas propiedades de CSS, como `font-family` declaradas con comillas, son sensibles a las mayúsculas y minúsculas

## Para descargar las fuentes:

Selecciono las fuentes que quiera

The screenshot shows a list of font families. At the top, it says 'Viewing 815 of 815 font families'. Below this, there's a list of fonts: 'Roboto' (Christian Robertson (12 styles)), 'Arsenal' (André Silveira (12 styles)), and 'All their equipment and instruments are alive.' (A re the wing). A red box highlights the 'Select this font' button next to the 'Arsenal' font entry.

Y en el menú de abajo a la derecha



## Como añadir varias fuentes de Google a tu proyecto

Puedes añadir varias fuentes de Google en el mismo enlace, separando cada nombre de fuente con el carácter “|” y asegurándote de usar el símbolo “+” para las fuentes con nombres compuestos.

Ejemplo:

```
<link rel="stylesheet"
      href="http://fonts.googleapis.com/css?family=Finger+Paint|Baumans|Ubuntu+Condensed|Lily+Script+One|Roboto|Rosario" />
```

Ejemplo

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" type="text/css"
      href="http://fonts.googleapis.com/css?family=Gloria+Hallelujah|Pacifico">
<style>
body { font-family: 'Gloria Hallelujah'; }
p { font-family: 'Pacifico'; }

</style>
</head>
<body>
Gloria Hallelujah
<p>Pacífico</p>
</body>
</html>
```

## Cómo descargar y utilizar Google Fonts en tu servidor

<http://manuelmorrison.com/blog/vincular-fuentes-con-css/>

## Determinar la fuente que se está usando en un texto

<http://www.omicrono.com/2016/09/identificar-fuentes-tipos-de-letra/>

<https://www.whatfontis.com/>

## Tamaño de la Fuente

La propiedad font-size establece el tamaño del texto.

Ser capaz de manejar el tamaño del texto es importante en el diseño web. Sin embargo, no debes usar ajustes del tamaño de fuente para hacer pasar párrafos como encabezados o títulos como párrafos.

Utiliza siempre las etiquetas HTML adecuadas, como `<h1>` -`<h6>` para los títulos y `<p>` para párrafos.

El valor de font-size puede ser un tamaño absoluto o relativo.

Tamaño absoluto:

- Establece el texto a un tamaño especificado
- No permite al usuario cambiar el tamaño de texto en todos los navegadores (malo por razones de accesibilidad)
- El tamaño absoluto es útil cuando el tamaño físico de la salida se conoce

Tamaño relativo:

- Establece el tamaño en relación con los elementos del entorno
- Permite al usuario cambiar el tamaño del texto en los navegadores

 Si no se especifica un tamaño de fuente, el tamaño predeterminado para el texto normal, como párrafos, es 16px (16px = 1em).

**1EM equivale a la medida de la letra "M" del tamaño de letra que estemos utilizando** (y se pilla la M porque sus dimensiones son más o menos cuadradas, mide más o menos lo mismo de alto que de ancho).

**Por defecto** el tamaño de letra debería ser de 16px, es decir que si no toco nada **1EM = 16px**. PERO, aquí está la clave, **si yo redefino el FONT-SIZE, el tamaño de 1EM VARIARÁ TAMBIÉN**. Es decir que si yo pongo `body { font-size: 10px } 1EM=10px`.

## Establecer Tamaño de fuente con píxeles

Ajustar el tamaño del texto con píxeles ofrece un control total sobre el tamaño del texto:

## Ejemplo

```
h1 {font-size:40px;}  
h2 {font-size:30px;}  
p {font-size:14px;}
```

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
h1 {font-size:40px;}  
h2 {font-size:30px;}  
p {font-size:14px;}  
</style>  
</head>  
<body>  
  
<h1>This is heading 1</h1>  
<h2>This is heading 2</h2>  
<p>This is a paragraph.</p>  
<p>Specifying the font-size in px allows Internet Explorer 9, Firefox, Chrome, Opera, and  
Safari to resize the text.</p>  
<p><strong>Note:</strong> This example does not work in IE, prior version 9.</p>  
  
</body>  
</html>
```

El ejemplo anterior permite a Internet Explorer 9, Firefox, Chrome, Opera y Safari cambiar el tamaño del texto.

**Nota:** El ejemplo anterior no funciona en la versión de IE anterior a la 9.

El texto se puede cambiar el tamaño de todos los navegadores que utilizan la herramienta de zoom (sin embargo, esto cambia el tamaño de la página entera, no sólo el texto).

## Establecer tamaño de fuente con Em

Para evitar problemas de cambio de tamaño con las versiones anteriores de Internet Explorer, muchos desarrolladores usan em en lugar de píxeles.

La unidad de medida recomendado por el W3C es el em.

1em es igual al tamaño actual de la fuente. El tamaño de texto por defecto en los navegadores es 16px. Por lo tanto, el tamaño predeterminado de 1em es 16px.

El tamaño se puede calcular de píxeles a em utilizando esta fórmula:  $píxeles / 16 = em$

## Ejemplo

```
h1 {font-size:2.5em;} /* 40px/16=2.5em */  
h2 {font-size:1.875em;} /* 30px/16=1.875em */  
p {font-size:0.875em;} /* 14px/16=0.875em */
```

```
<!DOCTYPE html>
```

```
<html>
<head>
<style>
h1 {font-size: 2.5em;} /* 40px/16=2.5em */
h2 {font-size: 1.875em;} /* 30px/16=1.875em */
p {font-size: 0.875em;} /* 14px/16=0.875em */
</style>
</head>
<body>

<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>Specifying the font-size in em allows all major browsers to resize the text.
Unfortunately, there is still a problem with older versions of IE. When resizing the text, it
becomes larger/smaller than it should.
</p>
</body>
</html>
```

En el ejemplo anterior, el tamaño del texto en em es el mismo que el ejemplo anterior en píxeles. Sin embargo, con el tamaño de la EM, es posible ajustar el tamaño del texto en todos los navegadores.

Por desgracia, todavía hay un problema con las versiones anteriores de IE. El texto se hace más grande de lo que debiera cuando agrandamos la fuente y más pequeño de lo que debería cuando bajamos el zoom.

## Use una combinación de porcentaje y Em

La solución que funciona en todos los navegadores, es establecer una fuente predeterminada de tamaño en porcentaje para el elemento <body>:

### Ejemplo

```
body {font-size:100%;}
h1 {font-size:2.5em;}
h2 {font-size:1.875em;}
p {font-size:0.875em;}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
body {font-size:100%;}
h1 {font-size:2.5em;}
h2 {font-size:1.875em;}
p {font-size:0.875em;}
</style>
</head>
<body>
<h1>This is heading 1</h1>
<h2>This is heading 2</h2>
<p>This is a paragraph.</p>
<p>Specifying the font-size in percent and em displays the same size in all
major browsers, and allows all browsers to resize the text!</p>
```

```
</body>  
</html>
```

Nuestro código ahora funciona muy bien! Se muestra el mismo tamaño del texto en todos los navegadores, y permite que todos los navegadores hacer zoom o cambiar el tamaño del texto!

Puntos (pt): los puntos se usan tradicionalmente en los medios impresos (todo lo que se va a imprimir en papel, etc.). Un punto es igual a 1/72 de pulgada. Los puntos se parecen mucho a los píxeles, ya que son unidades de tamaño fijo y no pueden escalar en tamaño.

<https://kyleschaeffer.com/development/css-font-size-em-vs-px-vs-pt-vs/>

**1em = 12pt = 16px = 100%.**

	body { font-size: 100%; }	body { font-size: 120%; }
font-size: 1em	The quick brown fox	<b>The quick brown</b>
font-size: 12pt	The quick brown fox	The quick brown fox
font-size: 16px	The quick brown fox	The quick brown fox
font-size: 100%	The quick brown fox	<b>The quick brown</b>

¿Qué es rem?

La unidad rem viene de Root EM, pero es muy importante que el nombre no confunda. rem no es igual al em del elemento root, es igual al font-size del elemento root, que en HTML sería el elemento <html>.

Si el font-size del <html> es 16px, 1rem sería igual a 16px en cualquier parte del documento. Como siempre se refiere al elemento root, no se ve afectado por el font-size de elementos anidados.

Cuando Utilizar Em vs. Rem

<https://webdesign.tutsplus.com/es/tutorials/comprehensive-guide-when-to-use-em-vs-rem--cms-23984>



## Más ejemplos

[Establezca el grosor de la fuente](#)

En este ejemplo se muestra cómo establecer la audacia de una fuente.

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.normal {font-weight: normal; }
```

```
p.light {font-weight: lighter; }
p.thick {font-weight: bold; }
p.thicker {font-weight: 900; }
</style>
</head>

<body>
<p class="normal">This is a paragraph.</p>
<p class="light">This is a paragraph.</p>
<p class="thick">This is a paragraph.</p>
<p class="thicker">This is a paragraph.</p>
</body>

</html>
```

#### Establecer la variante de la fuente

en este ejemplo se muestra cómo establecer la variante de una fuente.

Hola a Todo el Mundo.

HOLA A TODO EL MUNDO.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.normal {font-variant: normal; }
p.small {font-variant: small-caps; }
</style>
</head>

<body>
<p class="normal">Hola a Todo el Mundo.</p>
<p class="small">Hola a Todo el Mundo.</p>
</body>
</html>
```

#### Todas las propiedades de la fuente en una declaración

En este ejemplo se muestra cómo utilizar la propiedad abreviada para establecer todas las propiedades de la fuente en la declaración.

### Propiedad resumen Font

Existe una manera sencilla de reflejar todo los cambios que hemos realizado hasta el momento, y esta es la propiedad "font". Muchos le llamarán atajo, pero sigue siendo una propiedad en sí y básicamente su función consiste en combinar los valores de "font-style", "font-variant", "font-weight", "font-size", "line-height" y "font-family" en una sola línea de texto. La manera de definir su valor es separar cada propiedad separada por un espacio, como se muestra aquí:

```
p { font: bold small-caps 12px 1.5 arial, helvetica, sans-serif }
```

La propiedad compuesta *font* permite definir simultáneamente las propiedades relacionadas con el tipo de letra: *font-style*, *font-variant*, *font-weight*, *font-size*, *line-height* y *font-family* (explicadas más adelante en esta misma página).

Al escribir la propiedad compuesta, el orden en que deben aparecer los valores es el orden en que aparecen las propiedades simples en el párrafo anterior. Las únicas propiedades que deben aparecer obligatoriamente son *font-size* y *font-family*. Si se escribe la propiedad *line-height* debe aparecer separada de *font-size* por una barra (/).

```
p.uno {
    font: 150% sans-serif;
}
```

**Esto es un párrafo con fuente *sans-serif* (en Windows suele ser Arial) al 150%.**

```
p {
    font: bold 90% monospace;
}
```

**Esto es un párrafo con fuente *monospace* (en Windows suele ser Courier) al 90% y en negrita.**

```
p {
    font: oblique bold 120%
cursive;
}
```

**Esto es un párrafo con fuente *cursive* (en Windows suele ser Comic Sans) al 120%, en negrita y en oblicuo.**

```
p {
    font: small-caps 100%/200%
    serif;
}
```

ESTO ES UN PÁRRAFO CON  
FUENTE *SERIF* (EN WINDOWS  
SUELE SER TIMES NEW ROMAN)  
AL 100%, CON UN INTERLINEADO  
DOBLE, Y EN VERSALITAS.

Si no están las propiedades en el orden correcto, o falta alguna de las propiedades obligatorias, los navegadores no hacen caso.

```
p {
    font: monospace 150%;
    /* el orden no es correcto
   */
}
```

Esto debería ser un párrafo con fuente *monospace* (en Windows suele ser Courier) al 150%.

```
p {
    font: bold 90%;
    /* falta font-family */
}
```

Esto debería ser un párrafo al 90% y en negrita.

Si los primeros valores son correctos, pero al final hay valores que deberían estar en otro lugar, los navegadores aplican los valores correctos e ignoran los valores finales (*font-family* también se ignora).

```
p {
    font: oblique 120% sans-serif bold;
    /* bold debería estar al principio */
}
```

*Esto debería ser un párrafo con fuente sans-serif (en Windows suele ser Arial) al 120%, y en negrita.*

```
p {
    font: 120% sans-serif small-caps;
    /* small-caps debería estar al principio
   */
}
```

*Esto debería ser un párrafo con fuente sans-*

*serif* (en Windows suele ser Arial) al 120% y en versalitas.

```
<!DOCTYPE html>
<html>
<head>
<style>
/* Orden seguido font-style, font-variant, font-weight, font-size , line-height y font-family */
p.ex1
{
font:15px arial,sans-serif;
}

p.ex2
{
font:italic bold 12px/30px Georgia,serif;
}
</style>
</head>

<body>
<p class="ex1">This is a paragraph. This is a paragraph.</p>
<p class="ex2">This is a paragraph. This is a paragraph.</p>
</body>
</html>
```

## Todas las propiedades de fuente CSS

Propiedad	Descripción
font	Establece todas las propiedades de la fuente en la declaración
font-family	Especifica la familia de fuentes para el texto
font-size	Especifica el tamaño de fuente del texto
line-height	Especifica el tamaño del interlineado
font-style	Especifica el estilo de fuente para el texto
font-variant	Especifica si el texto debe aparecer en una fuente small-caps

font-weight	Especifica la anchura de una fuente
-------------	-------------------------------------

## CSS Links

Los enlaces pueden tener también estilos.

### Styling Links

Los enlaces pueden tener cualquier propiedad CSS (por ejemplo, color, font-family, fondo, etc.)

Los enlaces son que pueden tener diferentes estilos dependiendo del estado en que están

Los cuatro estados enlaces son los siguientes:

- a: link - un enlace normal, no visitado
- a: visited - un enlace que el usuario ha visitado
- a: hover - un enlace cuando el usuario pone el mouse sobre el
- a: active - un enlace el momento en que se hace clic en él

### Ejemplo

```
a:link {color:#FF0000;}      /* unvisited link */  
a:visited {color:#00FF00;}  /* visited link */  
a:hover {color:#FF00FF;}   /* mouse over link */  
a:active {color:#0000FF;}  /* selected link */
```

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
a:link {color:#FF0000;}  /* unvisited link */  
a:visited {color:#00FF00;} /* visited link */  
a:hover {color:#FF00FF;} /* mouse over link */  
a:active {color:#0000FF;} /* selected link */  
</style>  
</head>  
  
<body>  
<p><strong><a href="http://infolibre.es" target="_blank">This is a link</a></strong></p>  
<p><strong>Note:</strong> a:hover MUST come after a:link and a:visited in the CSS  
definition in order to be effective.</p>  
<p><strong>Note:</strong> a:active MUST come after a:hover in the CSS definition in order  
to be effective.</p>  
</body>  
</html>
```

Al configurar el estilo de los enlaces, hay algunas reglas de orden:

- a: hover debe ir después a: link y: visited
- a: active debe venir después de un: hover

## Estilos comunes de enlace

En el ejemplo anterior, el color del enlace cambia en función del estado en que está

Vamos a ver algunas de las formas comunes de dar estilo a los enlaces:

### DecoraciónTexto

La propiedad text-decoration se utiliza sobre todo para eliminar el subrayado de enlaces:

#### Ejemplo

```
a:link {text-decoration:none;}  
a:visited {text-decoration:none;}  
a:hover {text-decoration:underline;}  
a:active {text-decoration:underline;}  
  
<!DOCTYPE html>  
<html>  
<head>  
<style>  
a:link {text-decoration:none;} /* unvisited link */  
a:visited {text-decoration:none;} /* visited link */  
a:hover {text-decoration:underline;} /* mouse over link */  
a:active {text-decoration:underline;} /* selected link */  
</style>  
</head>  
  
<body>  
<p><b><a href="http://infolibre.es" target="_blank">This is a link</a></strong></p>  
<p><b>Note:</strong> a:hover MUST come after a:link and a:visited in the CSS  
definition in order to be effective.</p>  
<p><b>Note:</strong> a:active MUST come after a:hover in the CSS definition in order  
to be effective.</p>  
</body>  
</html>
```

## Color de fondo

La propiedad background-color especifica el color de fondo en los enlaces:

#### Ejemplo

```
a:link {background-color:#B2FF99;}  
a:visited {background-color:#FFFF85;}  
a:hover {background-color:#FF704D;}  
a:active {background-color:#FF704D;}
```

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
a:link {background-color:#B2FF99;} /* unvisited link */  
a:visited {background-color:#FFFF85;} /* visited link */  
a:hover {background-color:#FF704D;} /* mouse over link */  
a:active {background-color:#FF704D;} /* selected link */  
</style>  
</head>  
  
<body>  
<p><strong><a href="http://google.es" target="_blank">This is a link</a></strong></p>  
<p><strong>Note:</strong> a:hover MUST come after a:link and a:visited in the CSS  
definition in order to be effective.</p>
```

```
<p><strong>Note:</strong> a:active MUST come after a:hover in the CSS definition in order  
to be effective.</p>  
</body>  
</html>
```



## Más ejemplos

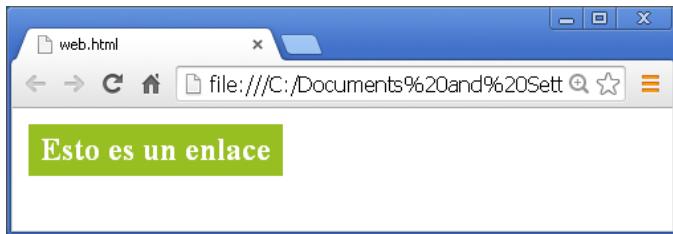
### Añadir estilos a hipervínculos

En este ejemplo se muestra cómo agregar otros estilos a hipervínculos.

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
a.one:link {color:#ff0000;}  
a.one:visited {color:#0000ff;}  
a.one:hover {color:#ffcc00;}  
  
a.two:link {color:#ff0000;}  
a.two:visited {color:#0000ff;}  
a.two:hover {font-size:150%;}  
  
a.three:link {color:#ff0000;}  
a.three:visited {color:#0000ff;}  
a.three:hover {background:#66ff66;}  
  
a.four:link {color:#ff0000;}  
a.four:visited {color:#0000ff;}  
a.four:hover {font-family:monospace;}  
  
a.five:link {color:#ff0000;text-decoration:none;}  
a.five:visited {color:#0000ff;text-decoration:none;}  
a.five:hover {text-decoration:underline;}  
</style>  
</head>  
  
<body>  
<p>Mouse over the links to see them change layout.</p>  
  
<p><strong><a class="one" href="http://google.es" target="_blank">This link changes  
color</a></strong></p>  
<p><strong><a class="two" href="http://google.es" target="_blank">This link changes font-  
size</a></strong></p>  
<p><strong><a class="three" href="http://google.es" target="_blank">This link changes background-  
color</a></strong></p>  
<p><strong><a class="four" href="http://google.es" target="_blank">This link changes font-  
family</a></strong></p>  
<p><strong><a class="five" href="http://google.es" target="_blank">This link changes text-  
decoration</a></strong></p>  
</body>  
  
</html>
```

### Avanzado - Crear cuadros de enlace

Este ejemplo muestra un ejemplo más avanzado en el que se combinan varias propiedades CSS para mostrar los enlaces como cuadros.



```
<!DOCTYPE html>
<html>
<head>
<style>
a:link,a:visited
{
display:block;
font-weight:bold;
color:#FFFFFF;
background-color:#98bf21;
width:120px;
text-align:center;
padding:4px;
text-decoration:none;
}
a:hover,a:active
{
background-color:#7A991A;
}
</style>
</head>

<body>
<a href="http://google.es" target="_blank">Esto es un enlace</a>
</body>
</html>
```

## CSS listas

Las propiedades de la lista de CSS permiten:

- Establecer diferentes marcadores de elementos de lista para las listas ordenadas
- Establecer diferentes marcadores de elementos de lista para listas desordenadas
- Establecer una imagen como marcador de elemento de la lista

### Lista

En HTML, hay dos tipos de listas:

- listas no ordenadas - los elementos de la lista están marcados con viñetas
- listas ordenadas - los elementos de la lista están marcados con números o letras

Con CSS, las listas pueden utilizar imágenes como marcadores de los ítems de la lista.

## Diferentes marcadores para los items de lista

El tipo de marcador de elemento de lista se especifica con la propiedad list-style-type:

### Ejemplo

```
ul.a {list-style-type: circle;}  
ul.b {list-style-type: square;}  
  
ol.c {list-style-type: upper-roman;}  
ol.d {list-style-type: lower-alpha;}
```

Example of unordered lists:

- Coffee
- Tea
- Coca Cola
  
- Coffee
- Tea
- Coca Cola

Example of ordered lists:

- I. Coffee
- II. Tea
- III. Coca Cola
  
- a. Coffee
- b. Tea
- c. Coca Cola

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
ul.a {list-style-type:circle; }  
ul.b {list-style-type:square; }  
ol.c {list-style-type:upper-roman; }  
ol.d {list-style-type:lower-alpha; }  
</style>  
</head>  
  
<body>  
<p>Example of unordered lists:</p>  
  
<ul class="a">  
<li>Coffee</li>  
<li>Tea</li>  
<li>Coca Cola</li>  
</ul>  
  
<ul class="b">  
<li>Coffee</li>  
<li>Tea</li>  
<li>Coca Cola</li>  
</ul>  
  
<p>Example of ordered lists:</p>  
<ol class="c">  
<li>Coffee</li>
```

```
<li>Tea</li>
<li>Coca Cola</li>
</ol>

<ol class="d">
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>

</body>
</html>
```

Unos estilos son para listas desordenadas, y otros para listas ordenadas.

Dando color solo a las viñetas (no ver)

```
<ul>

<li>

<a href="http://google.com" target="_blank">Google</a>

</li>

</ul>
```

Y mi CSS sería:

```
ul > li {
    color: red;
}

ul > li > a {
    color: black;
    text-decoration: none;
}
```

## Una imagen como marcador de elemento de lista

Para especificar una imagen como marcador de elemento de la lista, utiliza la propiedad list-style-image:

Ejemplo

```
ul
{
list-style-image: url('sqpurple.gif');
}

■ Coffee
■ Tea
■ Coca Cola
```

 <http://www2.uca.es/iconos3/sqpurple.gif>

```
<!DOCTYPE html>
<html>
<head>
<style>
ul
{
    list-style-image:url('sqpurple.gif');
}
</style>
</head>

<body>
    <ul>
        <li>Coffee</li>
        <li>Tea</li>
        <li>Coca Cola</li>
    </ul>
</body>
</html>
```

El ejemplo anterior no se muestra igual en todos los navegadores. IE y Opera mostrarán la imagen con una posición un poco más alta que Firefox, Chrome y Safari (en las últimas versiones no se distingue)

Si deseas que la imagen se coloque por igual en todos los navegadores, una solución se explica a continuación.

## Solución Crossbrowser

El siguiente ejemplo muestra la imagen del marcador por igual en todos los navegadores:

### Ejemplo

```
ul
{
list-style-type: none;
padding: 0px;
margin: 0px;
}
ul li
{
background-image: url(sqpurple.gif);
background-repeat: no-repeat;
background-position: 0px 5px;
padding-left: 14px;
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
ul
```

```
{
    list-style-type:none;
    padding:0px;
    margin:0px;
}
ul li
{
    background-image:url('sqpurple.gif');
    background-repeat:no-repeat;
    background-position:0px 5px;
    padding-left:14px;
}
</style>
</head>

<body>
<ul>
<li>Coffee</li>
<li>Tea</li>
<li>Coca Cola</li>
</ul>
</body>

</html>
```

Ejemplo explicado:

- Para ul:
  - Ajusta el list-style-type a none para quitar el marcador de elemento de la lista
  - Establece el relleno y el margen para 0px (para la compatibilidad cross-browser)
- Para todos los li del ul :
  - Establece la URL de la imagen, y mostrar sólo una vez (no-repeat)
  - Coloca la imagen donde deseé (izquierda y abajo 0px 5px)
  - Coloca el texto en la lista con padding-left

## List - propiedad abreviada

También es posible especificar todas las propiedades de lista en una sola propiedad. Esto se llama una propiedad resumida.

La propiedad abreviada utilizada para las listas, es la propiedad list-style:

### Ejemplo

```
ul
{
list-style: square url("sqpurple.gif");
}

<!DOCTYPE html>
<html>
<head>
<style>
ul
{
    list-style:square url("sqpurple.gif");
}
</style>
</head>

<body>
<ul>
<li>Coffee</li>
</ul>
```

```
<li>Tea</li>
<li>Coca Cola</li>
</ul>
</body>
</html>
```

Cuando se utiliza la propiedad resumida, el orden de los valores son:

- list-style-type
- list-style-position
- list-style-image

No importa si uno de los valores anteriores se han perdido, mientras que el resto están en el orden especificado.

La posición de la lista (list-style-position) especifica donde será colocado el marcador de la lista en relación a la caja principal.

list-style-position: inside | outside

outside

el marcador se encuentra fuera de la caja principal.

This is an unordered list:

-  Coffee
-  Tea
-  Coca Cola

inside

el marcador es la primera caja en línea dentro de la caja principal, después de la cual fluye el resto de los elementos.

This is an unordered list:

-  Coffee
-  Tea
-  Coca Cola

[https://www.w3schools.com/css/img\\_marker.png](https://www.w3schools.com/css/img_marker.png)

```
<!DOCTYPE html>
<html>
<head>
<style>
ul {
    list-style: circle inside url('img_marker.png');
}
</style>
</head>
<body>

<p>This is an unordered list:</p>
<ul>
<li>Coffee</li>
<li>Tea</li>
```

```
<li>Coca Cola</li>
```

```
</ul>
```

```
</body>
```

```
</html>
```



## Más ejemplos

[Todos los diferentes elementos de lista para las listas de marcadores](#)

Este ejemplo muestra todos los diferentes marcadores de elementos de lista en CSS.

```
<!DOCTYPE html>
<html>
<head>
<style>
ul.a { list-style-type:circle; }
ul.b { list-style-type:disc; }
ul.c { list-style-type:square; }

ol.d { list-style-type:armenian; }
ol.e { list-style-type:cjk-ideographic; }
ol.f { list-style-type:decimal; }
ol.g { list-style-type:decimal-leading-zero; }
ol.h { list-style-type:georgian; }
ol.i { list-style-type:hebrew; }
ol.j { list-style-type:hiragana; }
ol.k { list-style-type:hiragana-iroha; }
ol.l { list-style-type:katakana; }
ol.m { list-style-type:katakana-iroha; }
ol.n { list-style-type:lower-alpha; }
ol.o { list-style-type:lower-greek; }
ol.p { list-style-type:lower-latin; }
ol.q { list-style-type:lower-roman; }
ol.r { list-style-type:upper-alpha; }
ol.s { list-style-type:upper-latin; }
ol.t { list-style-type:upper-roman; }

ol.u { list-style-type:none; }
ol.v { list-style-type:inherit; }

</style>
</head>

<body>
<ul class="a">
<li>Circle type</li>
<li>Tea</li>
```

<li>Coca Cola</li>  
</ul>

<ul class="b">  
<li>Disc type</li>  
<li>Tea</li>  
<li>Coca Cola</li>  
</ul>

<ul class="c">  
<li>Square type</li>  
<li>Tea</li>  
<li>Coca Cola</li>  
</ul>

<ol class="d">  
<li>Armenian type</li>  
<li>Tea</li>  
<li>Coca Cola</li>  
</ol>

<ol class="e">  
<li>CJK-ideographic type</li>  
<li>Tea</li>  
<li>Coca Cola</li>  
</ol>

<ol class="f">  
<li>Decimal type</li>  
<li>Tea</li>  
<li>Coca Cola</li>  
</ol>

<ol class="g">  
<li>Decimal-leading-zero type</li>  
<li>Tea</li>  
<li>Coca Cola</li>  
</ol>

<ol class="h">  
<li>Georgian type</li>  
<li>Tea</li>  
<li>Coca Cola</li>  
</ol>

<ol class="i">  
<li>Hebrew type</li>  
<li>Tea</li>  
<li>Coca Cola</li>

</ol>

```
<ol class="j">
<li>Hiragana type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol class="k">
<li>Hiragana-iroha type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol class="l">
<li>Katakana type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol class="m">
<li>Katakana-iroha type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol class="n">
<li>Lower-alpha type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol class="o">
<li>Lower-greek type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol class="p">
<li>Lower-latin type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol class="q">
<li>Lower-roman type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>
```

```
<ol class="r">
<li>Upper-alpha type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>

<ol class="s">
<li>Upper-latin type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>

<ol class="t">
<li>Upper-roman type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>

<ol class="u">
<li>None type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>

<ol class="v">
<li>inherit type</li>
<li>Tea</li>
<li>Coca Cola</li>
</ol>

</body>
</html>
```

## Todas las propiedades de la lista de CSS

Propiedad	Descripción
<a href="#"><u>list-style</u></a>	Establece todas las propiedades de una lista en una declaración
<a href="#"><u>list-style-image</u></a>	Especifica una imagen como marcador de ítems de lista
<a href="#"><u>list-style-position</u></a>	Especifica si los marcadores list-item debe aparecer dentro o fuera del flujo de contenido
<a href="#"><u>list-style-type</u></a>	Especifica el tipo de marcador de ítems de lista

## CSS Tablas

El aspecto de una tabla HTML puede ser mejorado con CSS:

Empresa	Contacto	País
Alfreds Futterkiste	Maria Anders	Alemania
Berglunds snabbköp	Christina Berglund	Suecia
Centro comercial Moctezuma	Francisco Chang	México
Ernst Handel	Roland Mendel	Austria
Isla Trading	Helen Bennett	Reino Unido
Königlich Essen	Philip Cramer	Alemania
Riendo Winecellars Bacchus	Yoshi Tannamuri	Canadá
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italia
Norte / Sur	Simon Crowther	Reino Unido
París spécialités	Marie Bertrand	Francia
The Big Cheese	Liz Nixon	EE.UU.
Vaffeljernet	Palle Ibsen	Dinamarca

## Bordes de tabla

Para especificar los bordes de la tabla en CSS, utiliza la propiedad border.

El ejemplo siguiente especifica un borde negro para la tabla, y los elementos th y td:

### Ejemplo

```
table, th, td
{
border: 1px solid black;
}
```

Firstname	Lastname
Peter	Griffin
Lois	Griffin

```
<!DOCTYPE html>
<html>
<head>
<style>
table,th,td
{
    border: 1px solid black;
}
</style>
</head>

<body>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
</tr>
<tr>
<td>Peter</td>
<td>Griffin</td>
</tr>
<tr>
<td>Lois</td>
<td>Griffin</td>
</tr>
</table>
```

```
</body>
</html>
```

Fíjate como la tabla del ejemplo anterior tiene bordes dobles. Esto se debe a que la tabla y los elementos th / td tienen bordes separados.

Para mostrar un borde sencillo para la tabla, utiliza la propiedad border-collapse.

## Border-Collapse

La propiedad border-collapse establece si los bordes de la tabla se colapsan en un solo borde o por lo contrario deben estar separados:

### Ejemplo

```
table
{
border-collapse:collapse;
}
table,th, td
{
border: 1px solid black;
}
```

Firstname	Lastname
Peter	Griffin
Lois	Griffin

Si lo pongo en la misma regla también funciona

```
<style>
table,th,td
{
    border:1px solid black;
    border-collapse:collapse;
}
</style>
```

**Note:** If a !DOCTYPE is not specified, the border-collapse property can produce unexpected results in IE8 and earlier versions.

```
<!DOCTYPE html>
<html>
<head>
<style>
table
{
    border-collapse:collapse;
}
table, td, th
{
    border:1px solid black;
}
</style>
</head>

<body>
<table>
<tr>
<th>Firstname</th>
```

```

<th>Lastname</th>
</tr>
<tr>
<td>Peter</td>
<td>Griffin</td>
</tr>
<tr>
<td>Lois</td>
<td>Griffin</td>
</tr>
</table>
<p>
<strong>Note:</strong>
If a !DOCTYPE is not specified, the border-collapse property can produce unexpected results in IE8 and
earlier versions.
</p>
</body>
</html>

```

## Inciso uso de navegadores

Distintos estudios que cada uno dice una cosa

Browser Statistics w3schools.com

2016	Chrome	IE	Firefox	Safari	Opera
December	73.7 %	4.8 %	15.5 %	3.5 %	1.1 %
November	73.8 %	5.2 %	15.3 %	3.5 %	1.1 %
October	73.0 %	5.2 %	15.7 %	3.6 %	1.1 %
September	72.5 %	5.3 %	16.3 %	3.5 %	1.0 %
August	72.4 %	5.2 %	16.8 %	3.2 %	1.1 %
July	71.9 %	5.2 %	17.1 %	3.2 %	1.1 %
June	71.7 %	5.6 %	17.0 %	3.3 %	1.1 %
May	71.4 %	5.7 %	16.9 %	3.6 %	1.2 %
April	70.4 %	5.8 %	17.5 %	3.7 %	1.3 %
March	69.9 %	6.1 %	17.8 %	3.6 %	1.3 %
February	69.0 %	6.2 %	18.6 %	3.7 %	1.3 %
January	68.4 %	6.2 %	18.8 %	3.7 %	1.4 %

<https://www.w3counter.com/globalstats.php>

<http://gs.statcounter.com/>

<https://www.netmarketshare.com/browser-market-share.aspx?qprid=2&qpcustomd=0>

## Ancho y Alto del Cuadro

La anchura y altura de una tabla está definida por las propiedades anchura y la altura.

El ejemplo siguiente establece el ancho de la tabla a 100%, y la altura de los elementos th a 50px:

## Ejemplo

```
table
{
    width:100%;
}
th
{
    height:50px;
}
```

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

```
<!DOCTYPE html>
<html>
<head>
<style>
table,td,th
{
    border:1px solid black;
}
table
{
    width:100%;
}
th
{
    height:50px;
}
</style>
</head>

<body>
    <table>
        <tr>
            <th>Firstname</th>
            <th>Lastname</th>
            <th>Savings</th>
        </tr>
        <tr>
            <td>Peter</td>
            <td>Griffin</td>
            <td>$100</td>
        </tr>
        <tr>
            <td>Lois</td>
            <td>Griffin</td>
            <td>$150</td>
        </tr>
        <tr>
            <td>Joe</td>
            <td>Swanson</td>
            <td>$300</td>
        </tr>
        <tr>
            <td>Cleveland</td>
        </tr>
    </table>
</body>
```

```
<td>Brown</td>
<td>$250</td>
</tr>
</table>
</body>
</html>
```

## Alineación del texto de la Tabla

El texto de una tabla está alineada con las propiedades `text-align` y `vertical-align`.

La propiedad `text-align` establece la alineación horizontal, como izquierda, derecha o centro:

### Ejemplo

```
td
{
text-align:right;
}

<!DOCTYPE html>
<html>
<head>
<style>
table,td,th
{
    border:1px solid black;
}
td
{
    text-align:right;
}
</style>
</head>

<body>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Savings</th>
</tr>
<tr>
<td>Peter</td>
<td>Griffin</td>
<td>$100</td>
</tr>
<tr>
<td>Lois</td>
<td>Griffin</td>
<td>$150</td>
</tr>
<tr>
<td>Joe</td>
<td>Swanson</td>
<td>$300</td>
</tr>
<tr>
<td>Cleveland</td>
<td>Brown</td>
<td>$250</td>
</tr>
</table>
```

```
</body>
</html>
```

Haciendo que afecte solo a determinadas columnas.

```
<!DOCTYPE html>

<html>
<head>
<title>CSS-ESTILOS TABLAS</title>
<style>

table,th,td
{
    border:1px solid blue;
}
table
{
    width:100%;
}
th
{
    height:50px;
}
td.der
{
    text-align:right;
}

</style>
</head>
<body>

    <table>
        <tr>
            <th>Nombre</th>
            <th>Apellido</th>
            <th>Ahorros</th>
        </tr>
        <tr>
            <td>Peter</td>
            <td>Griffin</td>
            <td class="der">$100</td>
        </tr>
        <tr>
            <td>Lois</td>
            <td>Griffin</td>
            <td class="der">$150</td>
        </tr>
        <tr>
            <td>Joe</td>
            <td>Swanson</td>
            <td class="der">$300</td>
        </tr>
        <tr>
            <td>Cleveland</td>
            <td>Brown</td>
            <td class="der">$250</td>
        </tr>
    </table>

```

```

        </table>

        <p>
            <strong>Note:</strong>
            If a !DOCTYPE is not specified, the border-collapse property can produce unexpected
            results in IE8 and earlier versions
        </p>

    </body>
    <hr/>
</html>

```

La propiedad vertical-align establece la alineación vertical, como arriba, abajo o en medio (middle, top, bottom):

Más valores en

<https://developer.mozilla.org/es/docs/Web/CSS/vertical-align>

### Ejemplo

```

td
{
height:50px;
vertical-align:bottom;
}

```

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

```

<!DOCTYPE html>
<html>
<head>
<style>
table, td, th
{
border: 1px solid black;
}
td
{
height:50px;
vertical-align:bottom;
}
</style>
</head>

<body>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Savings</th>

```

```
</tr>
<tr>
<td>Peter</td>
<td>Griffin</td>
<td>$100</td>
</tr>
<tr>
<td>Lois</td>
<td>Griffin</td>
<td>$150</td>
</tr>
<tr>
<td>Joe</td>
<td>Swanson</td>
<td>$300</td>
</tr>
<tr>
<td>Cleveland</td>
<td>Brown</td>
<td>$250</td>
</tr>
</table>
</body>
</html>
```

## Relleno de Cuadro

Para controlar el espacio entre el borde y el contenido de una tabla, utiliza la propiedad de relleno (padding) en los elementos th y td:

### Ejemplo

```
td
{
    padding:15px;
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
table, td, th
{
    border:1px solid black;
}
td
{
    padding:15px;
}
</style>
</head>

<body>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Savings</th>
</tr>
<tr>
<td>Peter</td>
<td>Griffin</td>
```

```
<td>$100</td>
</tr>
<tr>
<td>Lois</td>
<td>Griffin</td>
<td>$150</td>
</tr>
<tr>
<td>Joe</td>
<td>Swanson</td>
<td>$300</td>
</tr>
<tr>
<td>Cleveland</td>
<td>Brown</td>
<td>$250</td>
</tr>
</table>
</body>
</html>
```

## Colores de Tabla

En el ejemplo siguiente se especifica el color de los bordes y el color del texto y del fondo de los elementos th:

### Ejemplo

```
table, td, th
{
border:1px solid green;
}
th
{
background-color:green;
color:white;
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
table, td, th
{
border: 1px solid green;
}
th
{
background-color:green;
color:white;
}
</style>
</head>

<body>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Savings</th>
</tr>
<tr>
<td>Peter</td>
```

```
<td>Griffin</td>
<td>$100</td>
</tr>
<tr>
<td>Lois</td>
<td>Griffin</td>
<td>$150</td>
</tr>
<tr>
<td>Joe</td>
<td>Swanson</td>
<td>$300</td>
</tr>
<tr>
<td>Cleveland</td>
<td>Brown</td>
<td>$250</td>
</tr>
</table>
</body>
</html>
```



## Más ejemplos

### [Haz una tabla de fantasía](#)

En este ejemplo se muestra cómo crear una tabla de lujo.

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina Berglund	Sweden
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Königlich Essen	Philip Cramer	Germany
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy
North/South	Simon Crowther	UK
Paris spécialités	Marie Bertrand	France

```
<!DOCTYPE html>
<html>
<head>
<style>
#customers
{
    font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
    width: 100%;
    border-collapse: collapse;
}
#customers td, #customers th
{
```

```

        font-size: 1em;
        border: 1px solid #98bf21;
        padding: 3px 7px 2px 7px;
    }
    #customers th
    {
        font-size: 1.1em;
        text-align: left;
        padding-top: 5px;
        padding-bottom: 4px;
        background-color: #A7C942;
        color: #ffffff;
    }
    #customers tr.alt td
    {
        color: #000000;
        background-color: #EAF2D3;
    }

```

</style>

</head>

<body>

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina Berglund	Sweden
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Königlich Essen	Philip Cramer	Germany
Laughing Bacchus Winecellars	Yoshi Tannamuri	Canada
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy
North/South	Simon Crowther	

```

<td>UK</td>
</tr>
<tr class="alt">
    <td>Paris spécialités</td>
    <td>Marie Bertrand</td>
    <td>France</td>
</tr>
</table>
</body>
</html>

```

## REGLAS PARES E IMPARES

Una forma de mejorar la legibilidad de tablas de gran tamaño es pintar las filas de colores alternados. Por ejemplo, en la tabla que aparece a continuación, las filas pares tienen fondo gris claro y las filas impares, fondo blanco. Las reglas para hacerlo son extremadamente sencillas:

(even/par, odd/impar)

```

tr:nth-child(even) {background: #CCC }
tr:nth-child(odd) {background: #FFF}

```

# Superfoods

Salmon	Omega-3's help the brain develop properly, reduce the risk of Alzheimer's, and help prevent heart disease.
Spinach	Great source of folate and lutein. Prevents birth defects, heart disease, stroke, and protects your skin from sun damage.
Sweet Potatoes	Beta carotene protects your skin from sun damage.
Beans	Packed with protein, fiber and antioxidants, beans contribute to weight management and lower risk for heart disease and cancer.
Berries	Antioxidants and phytochemicals prevent cancer, Alzheimer's and heart disease.
Green Tea	Antioxidants lower risk for heart disease, cancer, and stroke.
Dark Chocolate	Packed with antioxidants, chocolate lowers risk for heart disease, hypertension and diabetes.

De hecho, CSS no sólo permite alternar entre pares e impares, sino también según intervalos arbitrarios. Las palabras claves "even" y "odd" no son más que abreviaturas de conveniencia. Por ejemplo, con una lista larga podríamos hacer esto:

```
li:nth-child(5n+3) {font-weight: bold}
```

Lo que quiere decir que cada quinto elemento contando a partir del tercero aparecerá en negrita. Dicho de otro modo, estarán en negrita los elementos con la numeración 3, 8, 13, 18, 23, etc.

Firstname	Lastname	Savings
Peter	Griffin	\$100
Lois	Griffin	\$150
Joe	Swanson	\$300
Cleveland	Brown	\$250

```
<!DOCTYPE html>
<html>
<head>
<style>
table, td, th
{
    border:1px solid black;
}
td
{
    padding:15px;
}
tr:nth-child(even) {background: #CCC; }
tr:nth-child(odd) {background: #FFF; }
</style>
</head>

<body>
<table>
<tr>
<th>Firstname</th>
<th>Lastname</th>
<th>Savings</th>
</tr>
<tr>
<td>Peter</td>
<td>Griffin</td>
<td>$100</td>
</tr>
<tr>
<td>Lois</td>
<td>Griffin</td>
<td>$150</td>
</tr>
<tr>
<td>Joe</td>
<td>Swanson</td>
<td>$300</td>
</tr>
<tr>
<td>Cleveland</td>
<td>Brown</td>
<td>$250</td>
</tr>
</table>
</body>
</html>
```

nth-child cuenta como hijos todos los elementos del body, por lo que en el siguiente ejemplo le estamos diciendo el elemento `<p>` que sea el segundo hijo (el primer hijo es el encabezado `<h1>`).

## También funcionaría con párrafos

```
<!DOCTYPE html>
<html>
<head>
<style>
/*podemos probar con nth-child(odd), nth-child(even) o nth-child(2n+1)*/
p:nth-child(2)
{
background: #ff0000;
}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>The first paragraph.</p>
<p>The second paragraph.</p>
<p>The third paragraph.</p>
<p>The fourth paragraph.</p>
<p><strong>Note:</strong> Internet Explorer does not support the :nth-child() selector.</p>
</body>
</html>
```

# This is a heading

The first paragraph.

The second paragraph.

The third paragraph.

The fourth paragraph.

**Note:** Internet Explorer does not support the :nth-child() selector.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.box1:nth-child(4) { color: red; }
/*Con *:nth-child(4) me cogería el cuarto element fuese el que fuese */
</style>
</head>
<body>
<div class='box1'>
Hello 1
</div>
<div class='box1'>
Hello 2
</div>
<div class='box1'>
Hello 3
</div>
```

```
</div>
<div class='box1'>
Hello 4
</div>
</body>
</html>
```

Si pongo esto sin el comentario

```
<!DOCTYPE html>
<html>
<head>
<style>
div.box1:nth-child(4) { color: red; }
/*Con *:nth-child(4) me cogería el cuarto elemento fuese el que fuese */
</style>
</head>
<body>
    <!-- <p>holo</p> -->
    <div class='box1'>
        Hello 1
    </div>
    <div class='box1'>
        Hello 2
    </div>
    <div class='box1'>
        Hello 3
    </div>
    <div class='box1'>
        Hello 4
    </div>
</body>
</html>
```

holo

Hello 1

Hello 2

**Hello 3**

Hello 4

Si pongo \*:nth-child(2) o \*:nth-child(1) no funciona como esperaríamos porque me cuenta como hijo 1 y como hijo 2 el head y el body respectivamente, a partir del 3 si funcionará como se espera.

```
<!DOCTYPE html>
<html>
<head>
<style>
/*podemos probar con nth-child(odd), nth-child(even) o nth-child(2n+1)*/
*:nth-child(3)
{
background:#ff0000;
}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>The first paragraph.</p>
<p>The second paragraph.</p><!--Esta etiqueta la podemos cambiar por otra para poder ver que
funciona igualmente-->
<p>The third paragraph.</p>
<p>The fourth paragraph.</p>
<p><strong>Note:</strong> Internet Explorer does not support the :nth-child() selector.</p>
</body>
```

</html>

## COLUMNAS PARES E IMPARES

---

Lo mismo vale para columnas de tablas, pero entonces debe haber un elemento del documento que se corresponda con la columna. A tal fin existe en HTML el elemento COL. La tabla debe tener un COL al principio de cada columna:

```
<table>
<col/><col/><col/><col/><col/><col/><col/><col/><col/><col/>
<tr><th>Mes<th>1994<th>1995<th>1996...
```

(El elemento COL sirve para otros usos además de la aplicación de estilos, pero en este caso lo único que necesitamos es que haya elementos COL). Las siguientes reglas ponen fondo amarillo a la primera columna y luego fondo gris a las columnas impares a partir de la número 3:

```
col:first-child {background: #FF0}
col:nth-child(2n+3) {background: #CCC}
```

Mes	1994	1995	1996	1997	1998	1999	2000	2001	2002
ene	14	13	14	13	14	11	11	11	11
feb	13	15	12	15	15	12	14	13	13
mar	16	15	14	17	16	15	14	15	15
abr	17	16	17	17	17	15	15	16	16
may	21	20	20	21	22	20	21	20	19
jun	24	23	25	24	25	23	25	23	24
Jul	29	28	26	26	27	26	25	26	25
ago	29	28	27	28	28	27	26	28	26
sep	24	23	23	26	24	24	24	22	21
oct	20	22	20	22	20	19	20	22	
nov	18	17	16	17	16	15	14	15	
dic	15	13	13	14	13	10	13	11	

El fondo de las filas (TR) se dibuja delante del fondo de las columnas (COL), de modo que no hay que ponerle fondo a las filas, o el fondo de las columnas no será visible.

La etiqueta <colgroup> especifica un grupo de una o mas columnas en una tabla para su formateo. Es útil para aplicar estilos a columnas completas.

```
<!DOCTYPE html>
<html>
<head>
<style>
table.ex2 {border-spacing: 0; }
table.ex2 td, th {padding: 0 0.2em; }
table.ex2 col:first-child {background: #FF0; }
table.ex2 col:nth-child(2n+3) {background: #CCC; }
</style>
</head>
<body>

<pre>
col:first-child {background: #FF0; }
col:nth-child(2n+3) {background: #CCC; }
</pre>
<table class="ex2">
<colgroup>
<col/><col/><col/><col/><col/><col/><col/><col/><col/>
</colgroup>
<tbody>
<tr>
<th>Mes</th>
<th>1994</th>
<th>1995</th>
<th>1996</th>
<th>1997</th>
<th>1998</th>
<th>1999</th>
<th>2000</th>
<th>2001</th>
<th>2002</th>
</tr>
<tr>
<td>ene</td>
<td>14</td>
<td>13</td>
<td>14</td>
<td>13</td>
<td>14</td>
<td>11</td>
<td>11</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>feb</td>
<td>13</td>
<td>15</td>
<td>12</td>
<td>15</td>
<td>15</td>
<td>12</td>
<td>14</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>mar</td>
<td>16</td>
<td>15</td>
<td>14</td>
<td>17</td>
<td>16</td>
```

```
<td>15</td>
<td>14</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>
</body>
</html>
```

#### Ajusta la posición de la leyenda de la tabla

Este ejemplo muestra cómo colocar el título de la tabla.

Los posibles valores son top y bottom

Table 1.1 Customers

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Berglunds snabbköp	Christina Berglund	Sweden
Centro comercial Moctezuma	Francisco Chang	Mexico
Ernst Handel	Roland Mendel	Austria
Island Trading	Helen Bennett	UK
Magazzini Alimentari Riuniti	Giovanni Rovelli	Italy
North/South	Simon Crowther	UK

**Note:** IE8 supports the caption-side property if a !DOCTYPE is specified.

```
<!DOCTYPE html>
<html>
<head>
<style>
caption
{
    caption-side: bottom;
}
</style>
</head>

<body>

<table border="1">
<caption>Table 1.1 Customers</caption>
<tr>
<th>Company</th>
<th>Contact</th>
<th>Country</th>
</tr>
<tr>
<td>Alfreds Futterkiste</td>
<td>Maria Anders</td>
<td>Germany</td>
</tr>
<tr>
<td>Berglunds snabbköp</td>
```

```
<td>Christina Berglund</td>
<td>Sweden</td>
</tr>
<tr>
<td>Centro comercial Moctezuma</td>
<td>Francisco Chang</td>
<td>Mexico</td>
</tr>
<tr>
<td>Ernst Handel</td>
<td>Roland Mendel</td>
<td>Austria</td>
</tr>
<tr>
<td>Island Trading</td>
<td>Helen Bennett</td>
<td>UK</td>
</tr>
<tr>
<td>Magazzini Alimentari Riuniti</td>
<td>Giovanni Rovelli</td>
<td>Italy</td>
</tr>
<tr>
<td>North/South</td>
<td>Simon Crowther</td>
<td>UK</td>
</tr>
</table>

<p><strong>Note:</strong> IE8 supports the caption-side
property if a !DOCTYPE is specified.</p>
</body>
</html>
```

## CSS Box Model

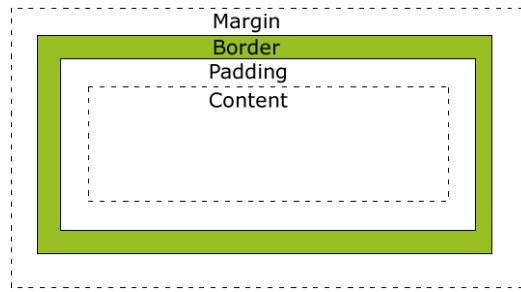
### El modelo de caja CSS

Todos los elementos HTML pueden ser considerados como cajas. En CSS, el término "modelo de caja" se usa cuando se habla de diseño y de layout.

El modelo de caja CSS es esencialmente una caja que envuelve los elementos HTML, y se compone de: los márgenes, bordes, el relleno y el contenido en sí.

El modelo de caja nos permite colocar un borde alrededor de los elementos y los elementos espaciales en función de otros elementos.

La siguiente imagen ilustra el modelo de caja:

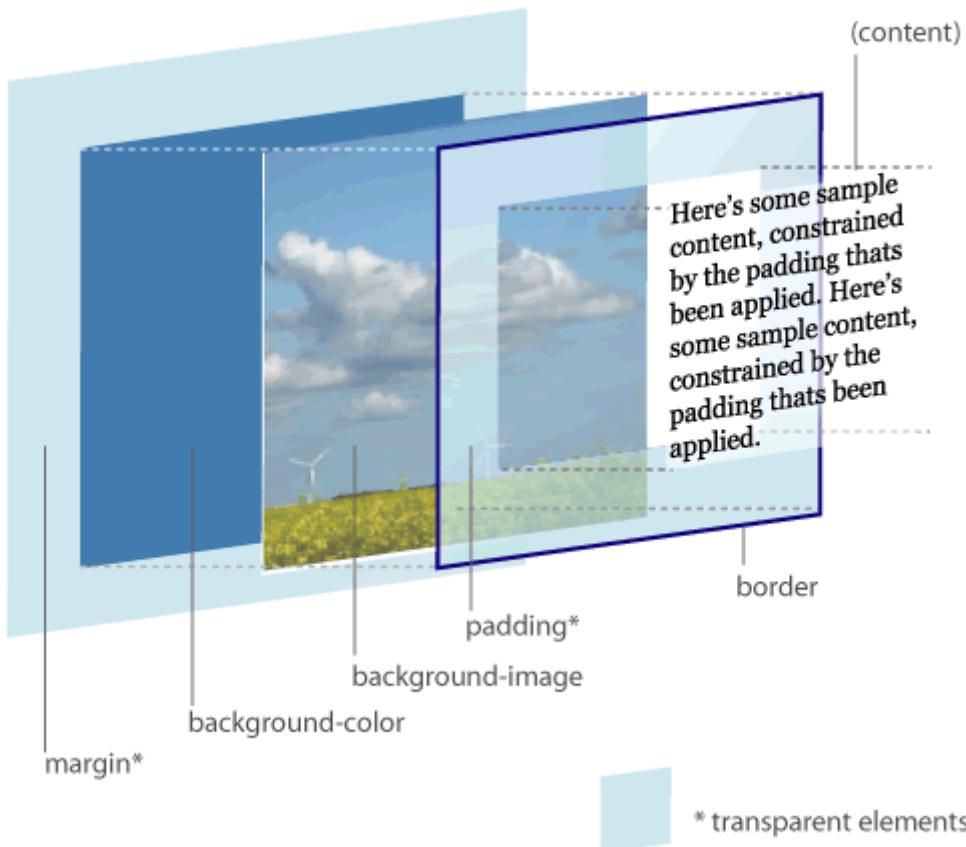


Explicación de las diferentes partes:

- **Margen** - Referencia un área alrededor del borde. El margen no tiene un color de fondo, que es completamente transparente. Se aplica a todos los elementos, salvo margin-top y margin-bottom que sólo se aplican a los elementos de bloque y a las imágenes.
- **Borde** - Un borde que va alrededor del relleno y contenido. El borde se ve afectado por el color de fondo de la caja
- **Relleno** - Hace referencia a un área alrededor del contenido. El relleno se ve afectado por el color de fondo de la caja
- **Contenido** - El contenido de la caja, donde aparecen el texto y las imágenes

Con el fin de establecer el ancho y el alto de un elemento correctamente en todos los navegadores, lo que necesitas saber es cómo funciona el modelo de caja.

## THE CSS BOX MODEL HIERARCHY



\* transparent elements

## Anchura y altura de un elemento

 **Importante:** Cuando estableces el ancho y el alto de un elemento con CSS, solo estableces la anchura y la altura del área de contenido. Para calcular el tamaño total de un elemento, también debe agregar el relleno, los bordes y los márgenes.

La anchura total del elemento en el siguiente ejemplo es 300px:

```
width:250px;  
padding:10px;  
border:5px solid gray;  
margin:10px;
```

Vamos a hacer los cálculos:

250px (anchura)  
+ 20px (relleno izquierdo y derecho)  
+ 10px (borde izquierdo y derecho)  
+ 20px (margen izquierdo y derecho)  
= 300px

Supongamos que sólo tenemos 250 píxeles de espacio. Haremos un elemento con un ancho total de 250px:

### Ejemplo

```
width:220px;  
padding:10px;  
border:5px solid gray;  
margin:0px;
```



The picture above is 250px wide.  
The total width of this element is  
also 250px.

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
div.ex  
{  
width:220px;  
padding:10px;  
border:5px solid gray;  
margin:0px;  
}  
</style>  
</head>  
<body>  
  
<div class="ex">The picture above is 250px wide.
```

The total width of this element is also 250px.</div>  
</body>  
</html>

La anchura total de un elemento debe calcularse así:

Ancho = ancho total del elemento + relleno izquierdo + relleno derecho + borde izquierdo + borde derecho + margen izquierdo + margen derecho

La altura total de un elemento se calcula de este modo:

Altura total del elemento = altura +relleno superior + relleno inferior + borde superior + borde inferior + margen superior + margeninferior

Modificando los valores 

Y el css  
div.ex  
{  
width:250px;  
padding:20px;  
border:5px solid gray;  
**margin:10px;**  
}

Si queremos que la alineación del div y la imagen se mantenga tendremos que modificar nuestro código

```
<!DOCTYPE html>
<html>
<head>
<style>
div.ex
{
    width:250px;
    padding:20px;
    border:5px solid gray;
    margin:10px;
}
img{
    margin:10px;
}
</style>
</head>
<body>

<div class="ex">The picture above is 250px wide.
The total width of this element is also 250px.</div>
</body>
</html>
```

## Problema de compatibilidad de los Navegadores

Versiones de IE8 y versiones anteriores de IE, incluyen el relleno y el borde en la propiedad anchura.

Para solucionar este problema, añade un <!DOCTYPE html> en la página HTML.

## CSS Bordes

## Propiedades CSS para Border

Las propiedades de borde CSS permiten especificar el estilo y el color del borde de un elemento.

### Border Style

La propiedad border-style especifica el tipo de borde que desea mostrar.

 Ninguna de las propiedades de borde tendrá ningún efecto a menos que la propiedad **border-style** se establezca!

### Valores del border-style:

none: Define que no habrá borde

dotted: Define un borde de puntos

dashed: Define un borde discontinuo

solid: Define un borde sólido

double: define dos líneas. La anchura de las dos líneas son el mismo que el valor límite de ancho

groove: Define un borde acanalado 3D. El efecto depende del valor de border-color

ridge: Define un borde estriado 3D. El efecto depende del valor de border-color

inset: Define un borde inset 3D (recuadro). El efecto depende del valor de border-color

outset: define una borde outset 3D. El efecto depende del valor de border-color

### Ejemplo

No border.

A dotted border.

A dashed border.

A solid border.

A double border.

A groove border.

A ridge border.

An inset border.

An outset border.

A hidden border.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.none {border-style:none;}
p.dotted {border-style:dotted;}
p.dashed {border-style:dashed;}
p.solid {border-style:solid;}
p.double {border-style:double;}
p.groove {border-style:groove;}
p.ridge {border-style:ridge;}
p.inset {border-style:inset;}
p.outset {border-style:outset;}
p.hidden {border-style:hidden;}
</style>
</head>

<body>
<p class="none">No border.</p>
<p class="dotted">A dotted border.</p>
<p class="dashed">A dashed border.</p>
<p class="solid">A solid border.</p>
<p class="double">A double border.</p>
<p class="groove">A groove border.</p>
<p class="ridge">A ridge border.</p>
<p class="inset">An inset border.</p>
```

```
<p class="outset">An outset border.</p>
<p class="hidden">A hidden border.</p>
</body>

</html>
```

## Border-width

La propiedad border-width se usa para ajustar el ancho del borde.

El ancho se establece en píxeles, o mediante el uso de uno de los tres valores predefinidos: thin, medium, o thick

**Nota:** La propiedad "border-width" no funciona si se utiliza solo. Utilice la propiedad "border-style" para establecer los bordes primero.

### Ejemplo

```
p.one
{
border-style:solid;
border-width:5px;
}
p.two
{
border-style:solid;
border-width:medium;
}

<!DOCTYPE html>
<html>
<head>
<style>
p.one
{
    border-style:solid;
    border-width:5px;
}
p.two
{
    border-style:solid;
    border-width:medium;
}
p.three
{
    border-style:solid;
    border-width:1px;
}
</style>
</head>

<body>
<p class="one">Some text.</p>
<p class="two">Some text.</p>
<p class="three">Some text.</p>
<p><strong>Note:</strong> The "border-width" property does not work if it is used alone. Use the
"border-style" property to set the borders first.</p>
</body>

</html>
```

Los posibles valores son:

medium  
thin  
thick

## Border-Color

La propiedad border-color se utiliza para establecer el color del borde. El color se puede ajustar por:

- Nombre - indique el nombre del color, como "red"
- RGB - especifica un valorRGB como "rgb(255,0,0)"
- Hex – indica un valor hexadecimal como "#ff0000"

También puede configurar el color del borde como "transparent" (transparente).

**Nota:** La propiedad "border-color" no funciona si se utiliza solo. Utilice la propiedad "border-style" para establecer las fronteras primero.

### Ejemplo

```
p.one
{
border-style:solid;
border-color:red;
}
p.two
{
border-style:solid;
border-color:#98bf21;
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
p.one
{
border-style: solid;
border-color: red;
}
p.two
{
border-style: solid;
border-color: #98bf21;
}
</style>
</head>

<body>
<p class="one">A solid red border</p>
<p class="two">A solid green border</p>
<p><strong>Note:</strong> The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.</p>
</body>
</html>
```

### Transparente

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```
div {
    border-style: solid;
    border-width: 10px;
    border-color: transparent;
}

</style>
</head>
<body>
<div>Este es mi div</div>
</body>
</html>
```

Este ejemplo no tiene mucha gracia porque realmente no estamos viendo nada, así que lo que haremos es modificar el color del borde en el evento hover

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    background-color: pink;
    border-style: solid;
    border-width: 10px;
    border-color: transparent;
}
div:hover {
    border-style: solid;
    border-width: 10px;
    border-color: black;
}
</style>
</head>
<body>
<div>Este es mi div</div>
</body>
</html>
```

## Border - lados individuales

En CSS es posible especificar límites diferentes para cada uno de los lados:

### Ejemplo

```
p
{
border-top-style:dotted;
border-right-style:solid;
border-bottom-style:dotted;
border-left-style:solid;
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
p
{
    border-top-style:dotted;
    border-right-style:solid;
    border-bottom-style:dotted;
    border-left-style:solid;
}
</style>
</head>
```

```
<body>
<p>2 different border styles.</p>
</body>
</html>
```

El ejemplo anterior también se puede ajustar con una sola propiedad:

### Ejemplo

```
border-style:dotted solid;

<!DOCTYPE html>
<html>
<head>
<style>
p
{
border-style:dotted solid;
}
</style>
</head>

<body>
<p>2 different border styles.</p>
</body>
</html>
```

La propiedad estilo de borde puede tener desde uno a cuatro valores.

- **border-style:** dotted solid double dashed;
  - borde superior es dotted
  - borde derecho es solid
  - borde inferior es el double
  - borde izquierdo es dashed
  
- **border-style:** dotted solid double;
  - borde superior es dotted
  - bordes derecho e izquierdo son solid
  - borde inferior es double
  
- **border-style:** dotted solid;
  - bordes superior e inferior es dotted
  - bordes derecho e izquierdo son solid
  
- **border-style: dotted;**
  - los cuatro bordes son dotted

Hemos utilizadola propiedad border-style en el ejemplo anterior. Sin embargo, también funciona con border-width y border-color

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```
p.uno
{
    border-style: dotted solid double dashed;
}
p.dos
{
    border-style: dotted solid double ;
}
p.tres
{
    border-style: dotted solid ;
}
p.cuatro
{
    border-style: dotted;
}
</style>
</head>
<body>
    <p class="uno">Parrafo uno</p>
    <p class="dos">Parrafo dos</p>
    <p class="tres">Parrafo tres</p>
    <p class="cuatro">Parrafo cuatro</p>
</body>
</html>
```

## Border - Abreviatura de las propiedades

Como se puede ver en los ejemplos anteriores, hay muchas propiedades a tener en cuenta cuando se trata de bordes.

Para acortar el código, también es posible especificar todas las propiedades de bordes individuales en una propiedad. Esto se llama una propiedad resumida.

La propiedad border es una forma abreviada de las propiedades de bordes individuales siguientes:

- border-width
- border-style (requerido)
- border-color

### Ejemplo

```
border:5px solid red;
```

```
<!DOCTYPE html>
<html>
<head>
<style>
p
{
    border:5px solid red;
}
</style>
</head>

<body>
<p>This is some text in a paragraph.</p>
</body>
</html>
```



## Más ejemplos

### Todas las propiedades de borde superior en una declaración

Este ejemplo se muestra una propiedad abreviada para establecer todas las propiedades para el borde superior en una declaración.

```
<!DOCTYPE html>
<html>
<head>
<style>
p
{
    border-style:solid;
    border-top:thick double #ff0000;
}
</style>
</head>

<body>
<p>This is some text in a paragraph.</p>
</body>

</html>
```

### Establezca el estilo del borde inferior

Este ejemplo muestra cómo establecer el estilo del borde inferior.

```
<!DOCTYPE html>
<html>
<head>
<style>
p { border-style:solid; }
p.none {border-bottom-style:none; }
p.dotted {border-bottom-style:dotted; }
p.dashed {border-bottom-style:dashed; }
p.solid {border-bottom-style:solid; }
p.double {border-bottom-style:double; }
p.groove {border-bottom-style:groove; }
p.ridge {border-bottom-style:ridge; }
p.inset {border-bottom-style:inset; }
p.outset {border-bottom-style:outset; }
</style>
</head>

<body>
<p class="none">No bottom border.</p>
<p class="dotted">A dotted bottom border.</p>
<p class="dashed">A dashed bottom border.</p>
<p class="solid">A solid bottom border.</p>
<p class="double">A double bottom border.</p>
<p class="groove">A groove bottom border.</p>
<p class="ridge">A ridge bottom border.</p>
<p class="inset">An inset bottom border.</p>
<p class="outset">An outset bottom border.</p>
</body>
</html>
```

### Ajuste el ancho del borde izquierdo

Este ejemplo muestra cómo establecer el ancho del borde izquierdo.

```
<!DOCTYPE html>
```

```
<html>
<head>
<style>
p
{
border-style:solid;
border-left-width:15px;
}
</style>
</head>

<body>
<p><strong>Note:</strong> The "border-left-width" property does not work if it is used alone. Use the "border-style" property to set the borders first.</p>
</body>
</html>
```

#### Ajuste el color de los cuatro bordes

Este ejemplo muestra cómo establecer el color de los cuatro bordes. Se puede tener de uno a cuatro colores.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.one
{
border-style:solid;
border-color:#0000ff;
}
p.two
{
border-style:solid;
border-color:#ff0000 #0000ff;
}
p.three
{
border-style:solid;
border-color:#ff0000 #00ff00 #0000ff;
}
p.four
{
border-style:solid;
border-color:#ff0000 #00ff00 #0000ff rgb(250,0,255);
}
</style>
</head>

<body>
<p class="one">One-colored border!</p>
<p class="two">Two-colored border!</p>
<p class="three">Three-colored border!</p>
<p class="four">Four-colored border!</p>
<p><strong>Note:</strong> The "border-color" property does not work if it is used alone. Use the "border-style" property to set the borders first.</p>
</body>
</html>
```

#### Ajuste el color del borde derecho

Este ejemplo muestra cómo establecer el color del borde derecho.

```
<!DOCTYPE html>
<html>
<head>
<style>
p
```

```
{
border-style:solid;
border-right-color:#ff0000;
}
</style>
</head>

<body>
<p>This is some text in a paragraph.</p>
</body>
</html>
```

## Todas las propiedades de borde CSS

Propiedad	Descripción
Border	Establece todas las propiedades de borde en una declaración
border-bottom	Establece todas las propiedades del borde de fondo en una declaración
border-bottom-color	Establece el color del borde inferior
border-bottom-style	Establece el estilo del borde inferior
border-bottom-width	Establece el ancho del borde inferior
border-color	Establece el color de los cuatro bordes
border-left	Establece todas las propiedades del borde izquierdo en una declaración
border-left-color	Establece el color del borde izquierdo
border-left-style	Establece el estilo del borde izquierdo
border-left-width	Establece el ancho del borde izquierdo
border-right	Establece todas las propiedades del borde derecho en una declaración
border-right-color	Establece el color del borde derecho
border-right-style	Establece el estilo del borde derecho
border-right-width	Establece el ancho del borde derecho
border-style	Establece el estilo de los cuatro bordes
border-top	Establece todas las propiedades de borde superior en una declaración
border-top-color	Establece el color del borde superior
border-top-style	Establece el estilo del borde superior

border-top-width	Establece el ancho del borde superior
border-width	Establece el ancho de los cuatro bordes

## CSS Outlines

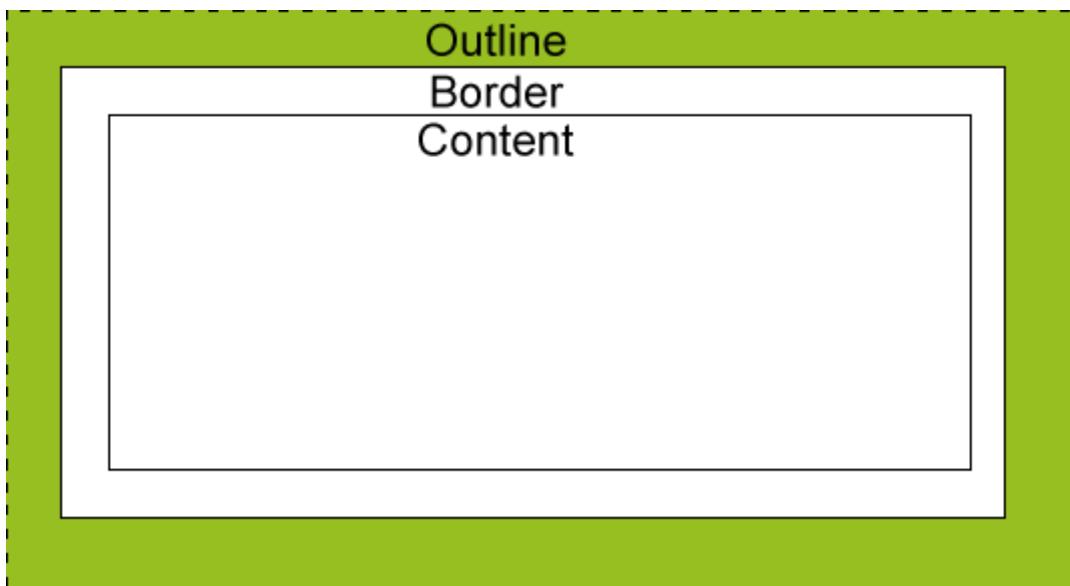
Un contorno es una línea que se dibuja en torno a los elementos (fuera de los bordes) para que el elemento "destaque".

Las propiedades del contorno especifican el estilo, el color y el ancho de un contorno.

### Esquema CSS

Sin embargo, la propiedad Outline es diferente de la propiedad border.

El contorno no es una parte de las dimensiones de un elemento; anchura total del elemento y la altura no se ve afectada por el ancho del contorno.



Se coloca encima del margen, no afecta a las dimensiones del bloque.



```
<!DOCTYPE html>
<html>
```

```
<head>
<style>
p.one
{
    border: 1px solid red;
    outline-style: solid;
    outline-width: 10px;
    margin: 100px;
}

</style>
</head>
<body>

<p class="one">This is some text in a paragraph.</p>

<p><strong>Note:</strong> IE8 supports the outline properties only if a !DOCTYPE is specified.</p>
</body>
</html>
```

## Ejemplos

### Traza una línea alrededor de un elemento (contorno)

En este ejemplo se muestra cómo dibujar una línea alrededor de un elemento, fuera del borde.

**Note:** IE8 supports the outline properties only if a !DOCTYPE is specified.

```
<!DOCTYPE html>
<html>
<head>
<style>
p
{
    border: 1px solid red;
/*El orden en el que se indican los valores de la propiedad outline es indiferente*/
    outline: green dotted thick;
}
</style>
</head>

<body>
<p><strong>Note:</strong> IE8 supports the outline properties only if a !DOCTYPE is specified.</p>
</body>
</html>
```

### Definir el estilo de un outline

En este ejemplo se muestra cómo establecer el estilo de un outline.

```
<!DOCTYPE html>
<html>
<head>
<style>
p {border: 1px solid red; }
p.dotted {outline-style: dotted; }
p.dashed {outline-style: dashed; }
p.solid {outline-style: solid; }
p.double {outline-style: double; }
p.groove {outline-style: groove; }
p.ridge {outline-style: ridge; }
p.inset {outline-style: inset; }
```

```
p.outset {outline-style: outset;}  
</style>  
</head>  
<body>  
  
<p class="dotted">A dotted outline</p>  
<p class="dashed">A dashed outline</p>  
<p class="solid">A solid outline</p>  
<p class="double">A double outline</p>  
<p class="groove">A groove outline</p>  
<p class="ridge">A ridge outline</p>  
<p class="inset">An inset outline</p>  
<p class="outset">An outset outline</p>  
  
<strong>Note:</strong> IE8 supports the outline properties only if a !DOCTYPE is specified.  
</body>  
</html>
```

#### Establecer el color de un outline

En este ejemplo se muestra cómo establecer el color de un esquema.

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p  
{  
border: 1px solid red;  
outline-style: dotted;  
outline-color: #00ff00;  
}  
</style>  
</head>  
  
<body>  
<p><strong>Note:</strong> IE8 supports the outline properties only if a !DOCTYPE is specified.</p>  
</body>  
</html>
```

#### Establecer el ancho de un outline

En este ejemplo se muestra cómo establecer el ancho de un esquema.

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
p.one  
{  
border: 1px solid red;  
outline-style: solid;  
outline-width: thin;  
}  
p.two  
{  
border: 1px solid red;  
outline-style: dotted;  
outline-width: 3px;  
}  
</style>  
</head>  
<body>  
  
<p class="one">This is some text in a paragraph.</p>  
<p class="two">This is some text in a paragraph.</p>  
  
<p><strong>Note:</strong> IE8 supports the outline properties only if a !DOCTYPE is
```

```
specified.</p>
</body>
</html>
```

## Ejemplo del offset

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    margin: 20px;
    border: 2px solid black;
    outline: 2px solid red;
    outline-offset: 15px;
}
</style>
</head>
<body>

<div>Este div tiene un desplazamiento del outline de 15px.</div>

</body>
</html>
```

## Todas las Propiedades del esquema CSS

El número en el "CSS" en la columna indica qué versión de CSS se define la propiedad (CSS1 o CSS2).

Propiedad	Descripción	Valores	CSS
Outline	Establece todas las propiedades de esquema en una declaración	outline-color outline-style outline-width inherit	2
outline-color	Establece el color de un esquema	color_name hex_number rgb_number invert inherit	2
outline-style	Establece el estilo de un esquema	none dotted dashed solid double groove ridge inset outset inherit	2
outline-width	Establece el ancho de un esquema	thin medium thick length inherit	2

## CSS Margen

Las propiedades de los márgenes CSS definen el espacio alrededor de los elementos.

## Margen

El margen hace referencia a un área alrededor de un elemento (fuera del borde). El margen no tiene un color de fondo, y es completamente transparente.

El margen superior, derecho, inferior e izquierdo se pueden cambiar independientemente mediante el uso de propiedades separadas. Una propiedad margen abreviada se puede utilizar también, para cambiar todos los márgenes a la vez.

El comportamiento de los márgenes verticales es más complejo de lo que se puede imaginar. Cuando se juntan dos o más márgenes verticales, se fusionan de forma automática y la altura del nuevo margen será igual a la altura del margen más alto de los que se han fusionado.

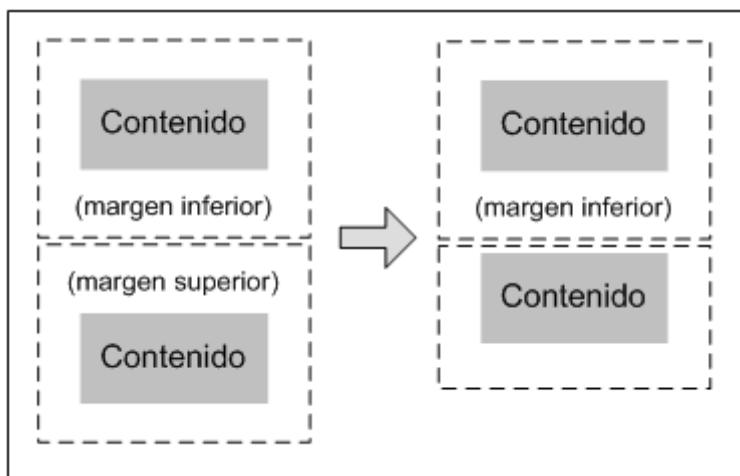


Figura: Fusión automática de los márgenes verticales

De la misma forma, si un elemento está contenido dentro de otro elemento, sus márgenes verticales se fusionan y resultan en un nuevo margen de la misma altura que el mayor margen de los que se han fusionado:

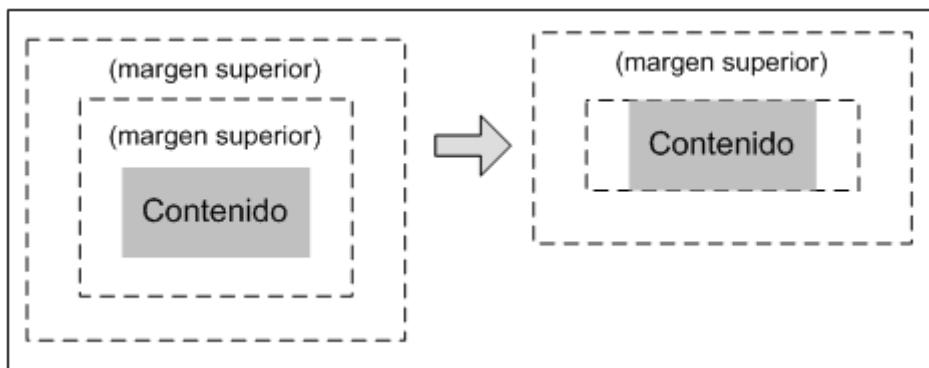


Figura: Fusión de los márgenes de los elementos interiores

Aunque en principio puede parecer un comportamiento extraño, la razón por la que se propuso este mecanismo de fusión automática de márgenes verticales es el de dar uniformidad a las páginas web habituales. En una página con varios párrafos, si no se diera este comportamiento y se estableciera un determinado margen a todos los párrafos, el primer párrafo no mostraría un aspecto homogéneo respecto de los demás.

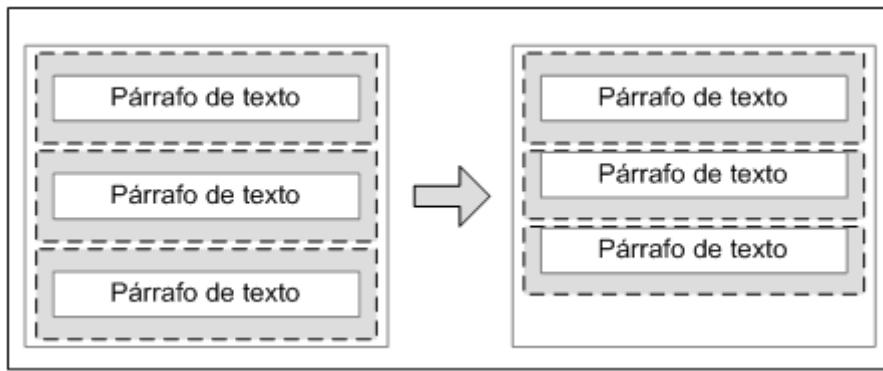


Figura: Motivo por el que se fusionan automáticamente los márgenes verticales

En el caso de un elemento que se encuentra en el interior de otro y sus márgenes se fusionan de forma automática, se puede evitar este comportamiento añadiendo un pequeño relleno (padding: 1px) o un borde (border: 1px solid transparent) al elemento contenedor.

## Valores posibles

Valor	Descripción
auto	El navegador calcula un margen
length	Especifica un margen en px, pt, cm, valor por defecto es 0px etc
%	Especifica un margen en porcentaje de la anchura del elemento que lo contiene
inherit	Especifica que el margen se hereda del elemento padre

💡 Es posible utilizar valores negativos, para solapar contenido.

## Margen - lados individuales

En CSS, es posible especificar diferentes márgenes por cada uno de los lados:

### Ejemplo

```
margin-top:100px;
margin-bottom:100px;
margin-right:50px;
margin-left:50px;
```

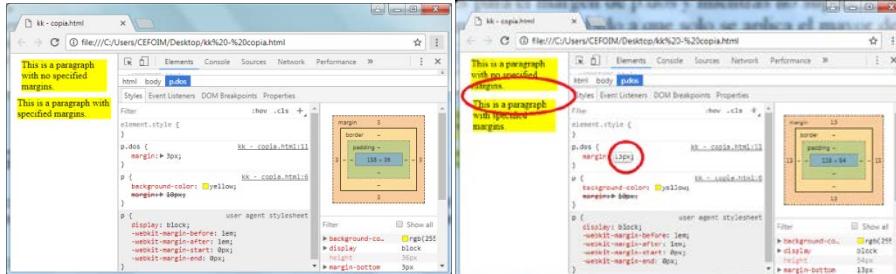
```
<!DOCTYPE html>
<html>
<head>
<style>
p
{
    background-color:yellow;
}
p.margin
{
    margin-top: 100px;
    margin-bottom: 100px;
    margin-right: 50px;
    margin-left: 50px;
}
</style>
</head>

<body>
```

```
<p>This is a paragraph with no specified margins.</p>
<p class="margin">This is a paragraph with specified margins.</p>
</body>

</html>
```

En el siguiente ejemplo, podremos ir modificando en la barra de desarrolladores los valores para el margen de p.dos y mientras no supere el del p normal no veremos diferencia, esto es debido a que solo se aplica el mayor de los dos.



```
<!DOCTYPE html>
<html>
<head>
<style>
p
{
    background-color: yellow;
    margin: 10px;
}
p.dos
{
    margin: 1px;
}
</style>
</head>

<body>
<p>Párrafo uno.</p>
<p class="dos"> Párrafo dos.</p>
</body>

</html>
```

## Margen - Abreviatura de las propiedades

Para acortar el código, es posible especificar todas las propiedades del margen en una propiedad resumen. La propiedad abreviada para todas las propiedades de margen es "margin":

### Ejemplo

```
margin:100px 50px;
```

```
<!DOCTYPE html>
<html>
<head>
<style>
p
{
    background-color: yellow;
}
p.margin
{
    margin:100px 50px;
}
```

```
</style>
</head>

<body>
<p>This is a paragraph with no specified margins.</p>
<p class="margin">This is a paragraph with specified margins.</p>
</body>

</html>
```

La propiedad de margen puede tener de uno a cuatro valores.

- **margin: 25px 50px 75px 100px;**
  - margen superior es de 25px
  - margen derecho es 50px
  - margen inferior es de 75px
  - margen izquierdo es de 100px
- **margin: 25px 50px 75px;**
  - margen superior es de 25px
  - márgenes derecho e izquierdo son 50px
  - margen inferior es de 75px
- **margin: 50px 25px;**
  - márgenes superior e inferior son 25px
  - márgenes derecho e izquierdo son 50px
- **margin: 25px;**
  - los cuatro márgenes son 25px



## Más ejemplos

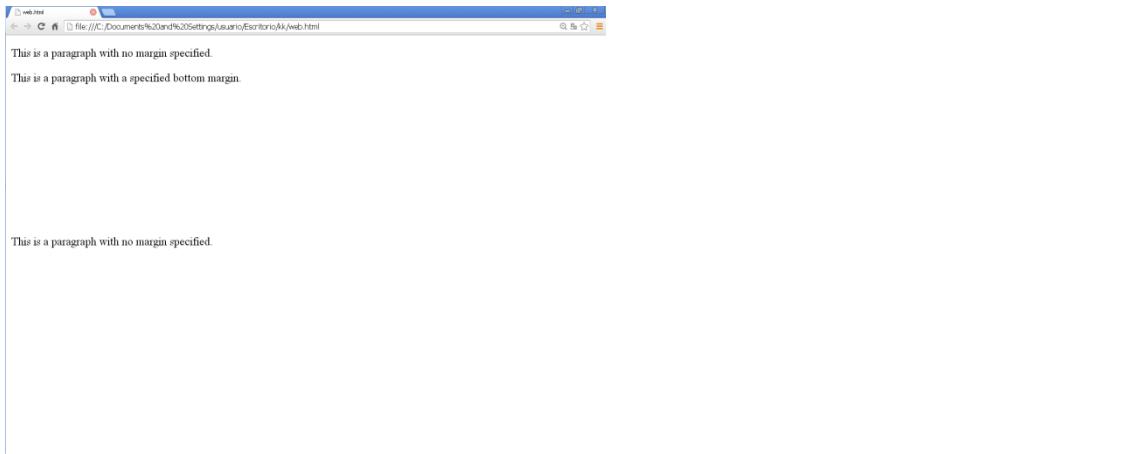
### [Establece el margen superior de un texto](#)

Este ejemplo muestra cómo se establece el margen superior de un texto en cm

```
<!DOCTYPE html>
<html>
<head>
<style>
p.ex1 {margin-top:2cm; }
</style>
</head>
<body>
<p>A paragraph with no margins specified.</p>
<p class="ex1">A paragraph with a 2cm top margin.</p>
<p>A paragraph with no margins specified.</p>
</body>
</html>
```

### [Establece el margen inferior de un texto usando un valor de porcentaje](#)

este ejemplo muestra cómo se establece el margen inferior en porcentaje, [respecto a la altura del elemento que lo contiene.](#)



```
<!DOCTYPE html>
<html>
<head>
<style>
p.bottommargin { margin-bottom:25%; }
body { background-color:yellow; }

</style>
</head>
<body>

<p>This is a paragraph with no margin specified.</p>
<p class="bottommargin">This is a paragraph with a specified bottom margin.</p>
<p>This is a paragraph with no margin specified.</p>

</body>
</html>
```

## Todas las propiedades de los márgenes CSS

Propiedad	Descripción
<a href="#"><u>margin</u></a>	Una propiedad abreviada para establecer las propiedades de los márgenes en una declaración
<a href="#"><u>margin-bottom</u></a>	Establece el margen inferior de un elemento
<a href="#"><u>margin-left</u></a>	Establece el margen izquierdo de un elemento
<a href="#"><u>margin-right</u></a>	Establece el margen derecho de un elemento
<a href="#"><u>margin-top</u></a>	Establece el margen superior de un elemento

## CSS Padding (Relleno)

Las propiedades de relleno CSS definen el espacio entre el borde del elemento y el contenido del elemento.

### Relleno

El relleno reserva un área alrededor del contenido (dentro del borde) de un elemento. El relleno se ve afectado por el color de fondo del elemento.

El relleno superior, derecho, inferior e izquierdo se pueden cambiar independientemente con el uso de propiedades separadas. Una propiedad de relleno abreviadapermite cambiar todos los rellenos a la vez.

## Valores posibles

Valor	Descripción
Length	Define un relleno fijo (en píxeles, pt, em, etc)
%	Define un relleno en% del elemento contenedor

## Relleno - lados individuales

En CSS, es posible especificar el relleno diferente para partes diferentes:

### Ejemplo

```
padding-top:25px;
padding-bottom:25px;
padding-right:50px;
padding-left:50px;
```

```
<!DOCTYPE html>
<html>
<head>
<style>
p
{
    background-color:yellow;
}
p.padding
{
    padding-top:25px;
    padding-bottom:25px;
    padding-right:50px;
    padding-left:50px;
}
</style>
</head>
<body>
<p>This is a paragraph with no specified padding.</p>
<p class="padding">This is a paragraph with specified paddings.</p>
</body>

</html>
```

## Padding - Abreviatura de las propiedades

Para acortar el código, es posible especificar todas las propiedades de relleno en una propiedad.

La propiedad abreviada para todas las propiedades de relleno es "padding":

### Ejemplo

```
padding:25px 50px;
```

```
<!DOCTYPE html>
<html>
<head>
<style>
p
{
    background-color:yellow;
}
p.padding
{
    padding:25px 50px;
}
</style>
</head>

<body>
<p>This is a paragraph with no specified padding.</p>
<p class="padding">This is a paragraph with specified paddings.</p>
</body>

</html>
```

La propiedad padding puede tener de uno a cuatro valores.

- **padding: 25px 50px 75px 100px;**
  - relleno superior es 25px
  - relleno derecho es 50px
  - relleno inferior es 75px
  - margen izquierdo es 100px
- **padding: 25px 50px 75px;**
  - relleno superior es 25px
  - rellenos derecha e izquierda son 50px
  - relleno inferior es 75px
- **padding: 25px 50px;**
  - rellenos superior e inferior son 25px
  - rellenos derecha e izquierda son 50px
- **padding: 25px;**
  - los cuatro rellenos son 25px



## Más ejemplos

### Todas las propiedades de relleno en una declaración

este ejemplo se demuestra una propiedad abreviada para establecer todas las propiedades de relleno en una declaración, pueden tener de uno a cuatro valores.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.ex1 {padding:2cm;}
p.ex2 {padding:0.5cm 3cm;}
</style>
</head>
<body>
```

```
<p class="ex1">This text has equal padding on each side. The padding on each side is 2cm.</p>
<p class="ex2">This text has a top and bottom padding of 0.5cm and a left and right padding of
3cm.</p>
</body>
</html>
```

#### Ajustar el relleno izquierdo

En este ejemplo se muestra cómo establecer el margen izquierdo del elemento P.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.padding {padding-left:2cm;}
p.padding2 {padding-left:50%;}
</style>
</head>
<body>
<p>This is a text with no left padding.</p>
<p class="padding">This text has a left padding of 2 cm.</p>
<p class="padding2">This text has a left padding of 50%.</p>
</body>
</html>
```

#### Ajuste el relleno derecho

de este ejemplo se muestra cómo establecer el margen derecho del elemento p.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.padding {padding-right:2cm;}
p.padding2 {padding-right:50%;}
</style>
</head>

<body>
<p>This is a text with no right padding. This is a text with no right padding. This is a text with no right
padding.</p>
<p class="padding">This text has a right padding of 2 cm. This text has a right padding of 2 cm. This
text has a right padding of 2 cm.</p>
<p class="padding2">This text has a right padding of 50%. This text has a right padding of 50%. This
text has a right padding of 50%.</p>
</body>
</html>
```

#### Establezca el relleno superior

En este ejemplo se muestra cómo establecer el relleno superior de elemento P.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.padding {padding-top:2cm;}
p.padding2 {padding-top:50%;}
</style>
</head>

<body>
<p>This is a text with no top padding. This is a text with no top padding. This is a text with no top
padding.</p>
<p class="padding">This text has a top padding of 2 cm. This text has a top padding of 2 cm. This text
has a top padding of 2 cm.</p>
<p class="padding2">This text has a top padding of 50%. This text has a top padding of 50%. This text
has a top padding of 50%.</p>
</body>
```

</html>

#### Ajusta el relleno inferior

Este ejemplo muestra cómo establecer el margen inferior del elemento ap.

```
<!DOCTYPE html>
<html>
<head>
<style>
p.padding {padding-bottom: 2cm; }
p.padding2 {padding-bottom: 50%; }
</style>
</head>

<body>
<p>This is a text with no bottom padding. This is a text with no bottom padding. This is a text with no bottom padding.</p>
<p class="padding">This text has a bottom padding of 2 cm. This text has a bottom padding of 2 cm. This text has a bottom padding of 2 cm.</p>
<p class="padding2">This text has a bottom padding of 50%. This text has a bottom padding of 50%. This text has a bottom padding of 50%.</p>
</body>
</html>
```

## Todas las propiedades de relleno de CSS

Propiedad	Descripción
<u>padding</u>	Una propiedad abreviada para establecer las propiedades de relleno en una declaración
<u>padding-bottom</u>	Establece el margen inferior de un elemento
<u>padding-left</u>	Establece el margen izquierdo de un elemento
<u>padding-right</u>	Establece el margen derecho de un elemento
<u>padding-top</u>	Define el relleno superior de un elemento

#### Las propiedades CSS por defecto

Sin CSS, una página está desnuda pero, aún así, algo de pudor le queda ya que los navegadores establecen una serie de propiedades por defecto que serán aplicadas a menos que las cambiemos. Tienen ciertas diferencias según se trate de un navegador u otro pero, en general, son justamente esas propiedades no-identificadas, las que nos causan problemas porque no solemos tenerlas en cuenta.

El **body** de una página es donde están definidas por defecto algunas de esas propiedades Es más o menos conocido que el fondo (**background**) de una página es blanco y que el color de los textos (**color**) es negro pero tiene algunas más que suelen perturbar, por ejemplo, el margen (**margin**):

```
body {
  background: #FFF url() repeat left top;
  color: #000;
  font-family: Times New Roman;
  font-size: 16px;
  margin: 8px;
}
```

Los enlaces también tienen propiedades por defecto, son de color azul y se muestran subrayados.

Todas las etiquetas de títulos tienen fuentes en negrita (**font-weight:bold**) y además, un tamaño que está en relación al tamaño de la fuente definida en el **body**. Para colmo, también tiene márgenes:

```
h1 font-size: 2em; margin: 21px 0;
h2 font-size: 1.5em; margin: 19px 0;
```

**h3** font-size: 1.2em; margin: 19px 0;  
**h4** font-size: 1em; margin: 21px 0;  
**h5** font-size: .8em; margin: 21px 0;  
**h6** font-size: .7em; margin: 26px 0;

Ni las etiquetas DIV ni SPAN tienen propiedades por defecto pero si las tiene la etiqueta P; esta posee un margen superior e inferior igual al tamaño de la fuente por defecto (margin:1em 0).

Más etiquetas con propiedades pre-definidas:

**blockquote** margin: 16px 40px;  
**pre** font-family: monospace; margin: 16px 0;

Las imágenes que sirven como enlaces tienen un borde de color igual al color de los enlaces (border: 2px solid #0000EE)

Probablemente, las etiquetas que causan más conflictos son las listas ya que tanto OL como UL tienen varias propiedades incluyendo márgenes y separaciones:

**ol** margin:16px 0; padding-left: 40px; list-style-position: outside; list-style-type: decimal;  
**ul** margin:16px 0; padding-left: 40px; list-style-position: outside; list-style-type: disc;

Es muy común leer que para evitar que todas estas propiedades no nos molesten, lo mejor es "resetearlas", es decir, poner al inicio de nuestra plantilla, algunas definiciones que "limpien" esas cosas. Hay decenas de ejemplos que pueden verse *online* pero, para un uso normal, no es necesario tanto, basta agregar unas pocas, justo al inicio de nuestra plantilla, dentro de etiquetas <style></style> o de <b:skin></b:skin>.

```
* {margin: 0; padding: 0;}  
  
html, body {height: 100%;}  
body {height: 100%;}  
  
a, a:visited, a:link, a:active {outline: none; text-decoration: none;}  
a:hover {outline: none; text-decoration: none;}  
a img {border: none; outline: none; text-decoration: none;}  
  
object, embed {outline: none;}  
  
ol, ul, li {list-style: none;}
```

Por supuesto, esto tendrá que verse caso por caso ya que es posible que algunas de las propiedades por defecto nos sean útiles y por lo tanto, no nos interese sacarlas pero, siempre es bueno saber que allí están y, cuando comenzamos un sitio nuevo, lo mejor es eliminarlas a todas y tomarnos el trabajo de definirlas una por una.

## CSS Selectores Agrupación y anidamiento

### Selectores Agrupación

En las hojas de estilo a menudo existen elementos con el mismo estilo.

```
h1  
{
```

```
color:green;
}
h2
{
color:green;
}
p
{
color:green;
}
```

Para reducir al mínimo el código, puedes agrupar selectores.

Separa cada selector con una coma.

En el siguiente ejemplo hemos agrupado los selectores en el código anterior:

### Ejemplo

```
h1,h2,p
{
color:green;
}
```

```
<!DOCTYPE html>
<html>
<head>
<style>
h1,h2,p
{
color:green;
}
</style>
</head>

<body>
<h1>Hello World!</h1>
<h2>Smaller heading!</h2>
<p>This is a paragraph.</p>
</body>
</html>
```

## Selectores de anidamiento

Es posible aplicar un estilo para un selector dentro de otro selector.

En el siguiente ejemplo, tenemos un estilo especificado para todos los elementos p, un estilo especificado para todos los elementos con class = "marked", y un tercer estilo se especifica sólo para elementos p dentro de los elementos con class = "marked":

### Ejemplo

```
p
{
color:blue;
text-align:center;
}
.marked
{
background-color:red;
```

```
}

.marked p
{
color:white;
}

<!DOCTYPE html>
<html>
<head>
<style>
p
{
    color:blue;
    text-align:center;
}
.marked
{
    background-color:red;
}
.marked p
{
    color:white;
}
</style>
</head>

<body>
<p>Esto es un párrafo azul y centrado</p>
<div class="marked">
    <p>Este párrafo no debería estar en azul.</p>
</div>
<pclass="marked">los párrafos que están dentro de un elemento de clase "marked" mantienen la  
alineación pero tienen un diferente color para el texto.</p>
</body>
</html>
```

## Selectores hijos

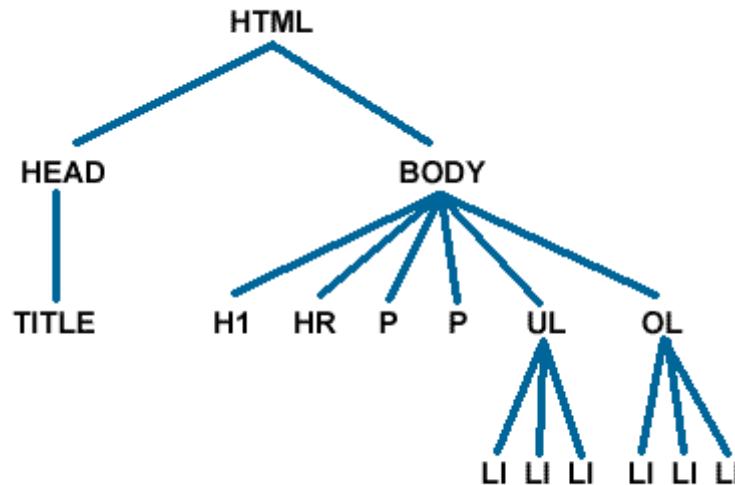
*Hijo: Un elemento A se considera hijo de B si y solo si B es el padre de A (el nodo en el nivel inmediatamente anterior a A).*

*Descendiente: Un elemento A se considera descendiente de B si, o bien,*

*A es hijo de B,  
ó bien,  
A es hijo de C que es descendiente de B.*

*Antecesor: Un elemento A se considera ancestro de B si y solo si B es un descendiente de A.*

Y para los que no entendieron nada, aquí está el árbol del documento HTML del ejemplo.



Un selector hijo se escribe de la forma:

**s1 > s2 { ... }**

donde **s2** es hijo de **s1**. Esto permite aplicar estilos solo a aquellos elementos hijos de **s1**.

Veamos un ejemplo:

**UL > LI { font-size: 1cm }**

En este ejemplo se aplicara el estilo a todos los elementos **LI** que sean hijos de **UL**. Así, en el siguiente ejemplo:

```

<!DOCTYPE html>
<html>
<head>
<title>mi página</title>
<style type="text/css">
p { color: #3366cc; font-family: arial; font-weight: bold; }
h1 { text-align: center; }
ul > li { font-size: 1cm ; }
</style>
</head>
<body>
<h1>bienvenidos todos a mi p&aacute;gina</h1>
<hr/>
<p>un p rrafo cualquiera . </p>
<p>otro p rrafo cualquiera. </p>
<ul>
    <li>elemento 1 de la lista 1</li>
    <ol>
        <li>elemento 1 de la lista 2</li>
        <li>elemento 2 de la lista 2</li>
        <li>elemento 3 de la lista 2</li>
    </ol>
    <li>elemento 2 de la lista 1</li>

```

```
</ul>
<ol>
    <li>elemento 1 de la lista 3</li>
    <li>elemento 2 de la lista 3</li>
    <li>elemento 3 de la lista 3</li>
</ol>
</body>
</html>
```

Solo los elementos LI en una lista sin orden mostraran el estilo.

## Selectores por atributos

Supongamos que nosotros queremos aplicar un estilo a P pero solo si tiene el atributo ALIGN. Para esto nos sirven los selectores por atributos.

Existen 4 formas de utilizar este tipo de selectores de las cuales solo veremos las 2 primeras.

**Estos selectores solo funcionan si el atributo al que se refiere está escrito inline, no hace caso a la herencia de CSS**

### 1. [ATRIBUTO]

Aplica el estilo solo si el elemento contiene el atributo especificado:

`p[align] { font-size: 1cm }`: aplica el estilo a p pero solo si tiene el atributo align.

`*[align] { font-size: 1cm }`: aplica el estilo a cualquier elemento que contenga el atributo align.

`p[align][size]{ font-size: 1cm }`: aplica el estilo a p pero solo si tiene los atributos align y size.

### 2. [ATRIBUTO = VALOR]

Aplica el estilo solo si el elemento contiene el atributo y el valor especificados:

`p[align=center] { font-size: 1cm }`: aplica el estilo a p pero solo si tiene el atributo align y el valor center.

`*[align=center] { font-size: 1cm }`: aplica el estilo a cualquier elemento que contenga el atributo align y el valor center.

`p[align=center][size]{ font-size: 1cm }`: aplica el estilo a p pero solo si tiene los atributos size y align y el valor center para align.

Hasta ahora hemos utilizado los atributos vistos en el temario, sin embargo el siguiente código también es valido:

```
<html>
<head>
<title>mi página</title>
<style type="text/css">
*[nombre="ana"] { color: #006699; font-family: arial; font-weight: bold; font-size: 4pt }
p[nombre="pepe"] { color: #ff0000; font-family: arial; font-weight: bold; }
p[nombre="jorge"] { color: #00ff00; font-family: arial; font-weight: bold; }
p[nombre="juan"] { color: #0000ff; font-family: arial; font-weight: bold; }
</style>
</head>
<body>
<h1 nombre="ana">ana</h1>
<p nombre="pepe">pepe</p>
<p nombre="jorge">jorge</p>
<p nombre="juan">juan</p>
<p nombre="ana">ana</p>
</body>
</html>
```

#### Mas ejemplos

- **[nombre\_atributo~=valor]**, selecciona los elementos que tienen establecido un atributo llamado **nombre\_atributo** y cuyo valor es una lista de palabras separadas por espacios en blanco en la que al menos una de ellas es exactamente igual a **valor**. (ALT+126 o ALT GR+4)

```
/* Se muestran de color azul todos los enlaces que tengan
   un atributo "class" en el que al menos uno de sus valores
   sea "externo" */

a[class~="externo"] { color:blue; }
```

Ejemplo

```
<html>
<head>
<title>mi página</title>
<style type="text/css">
/* Target data-info attr that contains the value "external"
*/
a[data-info~="external"] {
color: red;
```

}

```
/* And which contain the value "image" */  
  
a[data-info~="image"] {  
  
border: 1px solid black;  
  
}  
  
</style>  
  
</head>  
  
<body>  
  
<a href=".//micarpeta/image.jpg" data-info="external image">  
Clica</a>  
  
</body>  
  
</html>
```

#### Otro ejemplo con atributos data

```
<html>  
<head>  
<title>mi página</title>  
<style type="text/css">  
/* Todos los elementos que tengan el atributo "data-vegetable"  
irán en verde */  
  
[data-vegetable] {  
  
color: green;  
  
}  
  
/* Todos los elementos que tengan el atributo "data-vegetable"  
Con el valor "liquid" tendrán un color de fondo dorado */  
  
[data-vegetable="liquid"] {  
  
background-color: goldenrod;  
  
}
```

```

/* Todos los elementos que tengan el atributo "data-vegetable",
   Que contengan el valor "spicy", aunque tengan otros, tendrán un
   color de texto rojo */

[data-vegetable~="spicy"] {

    color: red;

}

</style>
</head>
<body>
Ingredientes de mi receta:

<ul>

<li data-quantity="1kg" data-vegetable>Tomates</li>

<li data-quantity="3" data-vegetable>Cebollas</li>

<li data-quantity="3" data-vegetable>Ajo</li>

<li data-quantity="700g" data-vegetable="not spicy like
chili">Pimiento rojo</li>

<li data-quantity="2kg" data-meat>Pollo</li>

<li data-quantity="optional 150g" data-meat>Bacon</li>

<li data-quantity="optional 10ml" data-vegetable="liquid">Aceite de
oliva</li>

<li data-quantity="25cl" data-vegetable="liquid">vino blanco</li>

</ul>
</body>
</html>

```

- **[nombre\_atributo]=valor**, selecciona los elementos que tienen establecido un atributo llamado **nombre\_atributo** y cuyo valor es una serie de palabras separadas con guiones, pero que comienza con **valor**. Este tipo de selector sólo es útil para los atributos de tipo **lang** que indican el idioma del contenido del elemento.

```

/* Selecciona todos los elementos de la página cuyo atributo
"lang" empiece por "es", es decir, "es", "es-ES", "es-AR", etc. */

*[lang|="es"] { color:red }

```

Para poner el símbolo | hay que poner ALT GR+1

Info sobre lang:

<https://www.w3.org/International/questions/qa-html-language-declarations.es>

A continuación se muestran algunos ejemplos de estos tipos de selectores:

```
/* Se muestran de color azul todos los enlaces que tengan  
   un atributo "class", independientemente de su valor */
```

```
a[class] { color:blue; }
```

```
/* Se muestran de color azul todos los enlaces que tengan  
   un atributo "class" con el valor "externo" */
```

```
a[class="externo"] { color:blue; }
```

```
/* Se muestran de color azul todos los enlaces que apunten  
   al sitio "http://www.ejemplo.com" */
```

```
a[href="http://www.ejemplo.com"] { color:blue; }
```

```
/* Selecciona todos los elementos de la página cuyo atributo  
   "lang" sea igual a "en", es decir, todos los elementos en inglés */  
*[lang=en] { ... }
```

El ^indica que debe empezar por esa palabra (la diferencia con el | es que | busca comienzo y que a continuación tenga un guión)

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
div[class^="test"] {
```

```
  background: #ffff00;  
}
```

```
</style>

</head>

<body>

<div class="first_test">The first div element.</div>

<div class="second">The second div element.</div>

<div class="test">The third div element.</div>

<p class="test">This is some text in a paragraph.</p>

</body>

</html>
```

Final de cadena

```
<!DOCTYPE html>

<html>

<head>

<style>

div[class$="test"] {

    background: #ffff00;

}

</style>

</head>

<body>
```

```
<div class="first_test">The first div element.</div>

<div class="second">The second div element.</div>

<div class="test">The third div element.</div>

<p class="test">This is some text in a paragraph.</p>

</body>

</html>
```

El \* indica que contiene esa palabra

```
<!DOCTYPE html>
<html>
<head>
<style>
div[class*="test"] {
    background: #ffff00;
}
</style>
</head>
<body>
<div class="first_test_hola">The first div element.</div>
<div class="second">The second div element.</div>
<div class="atesto">The third div element.</div>
<p class="testi">This is some text in a paragraph.</p>
</body>
</html>
```

El combinador ~ selecciona hermanos. Esto quiere decir que el segundo elemento sigue al primero (no necesariamente de forma inmediata) y ambos comparten el mismo elemento padre.

**Sintaxis:** A ~ B

**Ejemplo:** La regla p ~ span se aplicará a todos los elementos [span](#) que siguen un elemento [p](#).

```
<!DOCTYPE html>
<html>
<head>
<style>
p ~ ul {
    background: #ff0000;
}
</style>
</head>
<body>

<div>A div element.</div>
<ul>
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ul>

<p>The first paragraph.</p>
<ul>
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ul>

<h2>Another list</h2>
<ul>
<li>Coffee</li>
<li>Tea</li>
<li>Milk</li>
</ul>

</body>
</html>
```

A div element.

- Coffee
- Tea
- Milk

The first paragraph.

- Coffee
- Tea
- Milk

Another list

- Coffee
- Tea
- Milk

## Combinador de hermanos adyacentes

El combinador + selecciona hermanos adyacentes. Esto quiere decir que el segundo elemento sigue directamente al primero y ambos comparten el mismo elemento padre.

**Sintaxis:** A + B

**Ejemplo:** La regla h2 + p se aplicará a todos los elementos <p> que siguen directamente a un elemento <h2>.

```
<!DOCTYPE html>
<html>
<head>
<style>
div + p {
    background-color: yellow;
}
</style>
</head>
<body>

<h1>Welcome to My Homepage</h1>

<div>
<h2>My name is Donald</h2>
<p>I live in Duckburg.</p>
</div>

<p>My best friend is Mickey.</p>

<p>I will not be styled.</p>

</body>
</html>
```

**Todos los selectores:**

[https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)  
<https://www.w3schools.com/cssref/trysel.asp>

[https://www.w3.org/community/webed/wiki/Es/Selectores\\_avanzados\\_de\\_CSS](https://www.w3.org/community/webed/wiki/Es/Selectores_avanzados_de_CSS)

<https://code.tutsplus.com/es/tutorials/the-30-css-selectors-you-must-memorize--net-16048>

**Mas ejemplos**

```
<!DOCTYPE html>
<html>
<head>
<style>
[title]
{
color:blue;
}
</style>
</head>

<body>
<h2>Will apply to:</h2>
<h1 title="Hello world">Hello world</h1>
<a title="google" href="http://google.com">google</a>
<hr>
<h2>Will not apply to:</h2>
<p>Hello!</p>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<style>
[title= 'hola']
{
border:5px solid green;
}
</style>
</head>

<body>
<h2>Will apply to:</h2>

<br>
<a title=" hola" href="http://google.com">Google</a><!--con el espacio
delante del hola no funcionará -->
<hr>
<h2>Will not apply to:</h2>
<p title="greeting">Hi!</p>
<a class="hola" href="http://google.com">Google</a>
</body>
</html>
```

```
<!DOCTYPE html>
<html>
<head>
<style>
[title~=hello]
{
color:blue;
}
</style>
</head>

<body>
<h2>Will apply to:</h2>
<h1 title="hello world">Hello world</h1>
<p title="student hello">Hello CSS students!</p>
<hr>
<h2>Will not apply to:</h2>
<p title="student">Hi CSS students!</p>
</body>
</html>
```

En el ejemplo de abajo aplicaremos los estilos a todos los elementos con un atributo lang que contiene un valor especificado. Esto funciona incluso si el atributo tiene valores separados con un guion (-), (si tiene espacio en blanco no funcionaría):

## Ejemplo

```
[lang|=en] { color:blue; }
```

```
<!DOCTYPE html>
<html>
<head>
<style>
[lang|=en]
{
color:blue;
}
</style>
</head>

<body>
<h2>Will apply to:</h2>
<p lang="en">Hello!</p>
<p lang="en-us">Hi!</p>
<p lang="en-gb">Ello!</p>
<hr>
```

```
<h2>Will not apply to:</h2>
<p lang="us">Hi!</p>
<p lang="no">Hei!</p>
</body>
</html>
```

## Diseño en Formularios

```
<!DOCTYPE html>
<html>
<head>
<style>
input[type="text"]
{
width: 150px;
display: block;
margin-bottom: 10px;
background-color: yellow;
}
input[type="button"]
{
width: 120px;
margin-left: 35px;
display: block;
}
</style>
</head>
<body>
<form name="input" action="" method="GET">
Nombre: <input type="text" name="Name" value="Peter" size="20"/>
Apellido: <input type="text" name="Name" value="Griffin" size="20"/>
<input type="button" value="Pincha"/>
</form>
</body>
</html>
```

## CSS Dimensiones

La propiedad CSS dimension permite controlar la altura y la anchura de un elemento.



## Ejemplos

### Ajuste de la altura de los elementos

este ejemplo se muestra cómo establecer la altura de los diferentes elementos.

```
<!DOCTYPE html>
<html>
<head>
<style>
img.normal
{
    height:auto; /*auto (en este caso, es determinado por el navegador)*/
}
img.big
{
    height:120px;
}
p.ex
{
    height:100px;
    width:100px;
}
</style>
</head>

<body>

This is a paragraph with no height and width specified.  

This is a paragraph with no height and width specified.</p>
</body>
</html>
```

En este caso, como tenemos establecido un valor para el height en el propio elemento html, la regla css lo que hace es sustituirme ese valor, si no tuviésemos establecido ningún valor en el elemento y hubiese espacio, el navegador cogería el tamaño original

Si establezco una altura y luego el contenido ocupa mas, es posible que el contenido de ese div se superponga a otros elementos

### Ajuste de la altura de una imagen utilizando porcentaje

este ejemplo se muestra cómo establecer el alto de un elemento con un valor porcentual. Ese porcentaje es respecto al tamaño de su contenedor (si redimensionamos la ventana, veremos como cambia)

```
<!DOCTYPE html>
<html>
<head>
<style>
/* metemos esto para que los porcentajes nos funcionen bien*/
html {height:100%;}
body {height:100%;}
/* **** */
img.normal {height:auto;}
img.big {height:40%;}
img.small {height:10%;}
</style>
</head>
<body>

```

```

</body>
</html>
diferencia entre css height : 100% y height : auto
```

height:100% implica que el elemento tendrá el 100% de la altura de su contenedor padre.  
height:auto significa que el elemento tendrá una altura flexible, es decir su altura dependerá de la altura de los elementos hijos de la misma

ejemplo:

```
height:100%
<div style="height:50px">
<div id="innerDiv" style="height:100%">
</div>
</div>
#innerDiv tendrá un height:50px
height:auto
<div style="height:50px">
<div id="innerDiv" style="height:auto">
<div id="evenInner" style="height:10px">
</div>
</div>
</div>
Ahora el div #innerDiv tendrá un height:10px
```

Conseguir que las imágenes se redimensionen y no pierdan las proporciones (avanzado)  
<https://nestordominguez.com/ajustar-imagen-al-tamano-del-contenedor/>  
[https://www.w3schools.com/css/css3\\_object-fit.asp](https://www.w3schools.com/css/css3_object-fit.asp)

#### Ajuste el ancho de un elemento con un valor de píxeles

Este ejemplo muestra cómo establecer el ancho de un elemento con un valor de píxel.

```
<!DOCTYPE html>
<html>
<head>
<style>
img {width: 200px;}
</style>
</head>
<body>



</body>
</html>
```

#### Ajuste de la altura máxima de un elemento

este ejemplo se muestra cómo establecer la altura máxima de un elemento.

La principal diferencia entre height y max-height es que height ocupa espacio, incluso cuando el elemento está vacío, pero max-height solo ocupará espacio cuando se introduzca contenido, si está vacío no ocupa

Si lo meto en un div, el alto del contenedor no se adapta



```
<!DOCTYPE html>
<html>
<head>
<style>
```

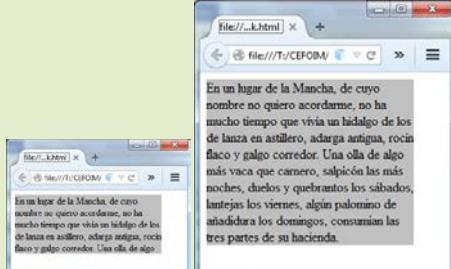
```

div
{
    background-color:blue;
}
p
{
    width:50%;
    max-height:50px;
    background-color:yellow;
}
</style>
</head>

<body>
<div>
<p>The maximum height of this paragraph is set to 50px. The
maximum height of this paragraph is set to 50px. The maximum
height of this paragraph is set to 50px. The maximum height of
this paragraph is set to 50px. The maximum height of this
paragraph is set to 50px. The maximum height of this paragraph
is set to 50px. The maximum height of this paragraph is set to
50px. The maximum height of this paragraph is set to 50px. The
maximum height of this paragraph is set to 50px. The maximum
height of this paragraph is set to 50px.</p>
</div>
</body>
</html>

```

Ejemplo:



```

<!DOCTYPE html>
<html>
<head>
<style>
#test {
    width:250px;
    max-height:100px;
    overflow: hidden;
background-color: #c0c0c0;
}

#test:hover {
max-height:none;
}
</style>
</head>
<body>
<div id="test">
    En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha
    mucho tiempo que vivía un hidalgo de los
    de lanza en astillero, adarga antigua, rocín
    flaco y galgo corredor. Una olla de algo
    más vaca que carnero, salpicón las más
    noches, duelos y quebrantos los sábados,
    lantejas los viernes, algún palomino de
    añadidura los domingos, consumían las
    tres partes de su hacienda.
</div>
</body>

```

En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor. Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados, lantejas los viernes, algún palomino de añadidura los domingos, consumían las tres partes de su hacienda.

vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados, lentejas los viernes, algún palomino de añadidura los domingos, consumían las tres partes de su hacienda.

```
</div>
</body>
</html>
```

#### Ejemplo con numero de líneas por mostrar

```
<!DOCTYPE html>
<html>
<head>
<style>
.max-lines {
    display: block; /* o inline-block */
text-overflow: ellipsis; /*Esto no funciona, solo funciona con el white-space: nowrap; y entonces solo me mostraría una línea*/
/* text-overflow es solo para una línea, si quiero hacerlo multilínea hay que hacer parafernalia*/
    overflow: hidden;
    max-height: 3.6em;
    line-height: 1.8em;
}

</style>
</head>
<body>
<div class="max-lines">
En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor. Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados, lantejas los viernes, algún palomino de añadidura los domingos, consumían las tres partes de su hacienda.
</div>
</body>
</html>
```

donde max-height: = line-height: × <número de líneas > en em.

#### Mismo ejemplo con hover

```
<!DOCTYPE html>
<html>
<head>
<style>
.max-lines {
    display: block; /* o inline-block */
text-overflow: ellipsis; /*Esto no funciona, solo funciona con el white-space: nowrap; y entonces solo me mostraría una línea*/
/* text-overflow es solo para una línea, si quiero hacerlo multilínea hay que hacer parafernalia*/
    overflow: hidden;
    max-height: 3.6em;
    line-height: 1.8em;
}

.max-lines:hover {
max-height:none;
}

</style>
</head>
<body>
<div class="max-lines">
En un lugar de la Mancha, de cuyo nombre no quiero acordarme, no ha mucho tiempo que vivía un hidalgo de los de lanza en astillero, adarga antigua, rocín flaco y galgo corredor. Una olla de algo más vaca que carnero, salpicón las más noches, duelos y quebrantos los sábados, lantejas los viernes, algún palomino de añadidura los domingos, consumían las tres partes de su hacienda.
</div>
</body>
</html>
```

#### Ajuste la anchura máxima de un elemento utilizando porcentaje

este ejemplo se muestra cómo establecer la anchura máxima de un elemento mediante un valor porcentual.

This is some text.  
This is some text.

```
<!DOCTYPE html>
<html>
<head>
<style>
p
{
    max-width: 20%;
    background-color: yellow;
}
</style>
</head>
<body>

<p>This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.
This is some text. This is some text. This is some text.</p>

</body>
</html>
```

¿Cuál es la diferencia entre max-width y width?

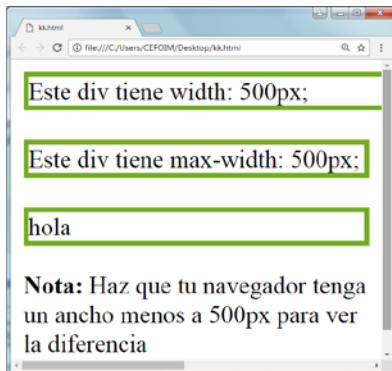
Width hace que me aparezca un scroll horizontal que no es lo mejor para hacer responsive mi web

```
<!DOCTYPE html>
<html>
<head>
<style>
div.ex1 {
    width: 500px;
    margin: auto;
    border: 3px solid #73AD21;
}

div.ex2 {
    max-width: 500px;
    margin: auto;
    border: 3px solid #73AD21;
}
</style>
</head>
<body>
```

```
<div class="ex1">Este div tiene width: 500px;</div>
<br/>
<div class="ex2"> Este div tiene max-width: 500px;</div>
<br/>
<div class="ex2"> hola</div>
<p><strong>Nota:</strong> Haz que tu navegador tenga un ancho menos a 500px para ver la diferencia
</p>

</body>
</html>
```



#### Ajuste la altura mínima de un elemento

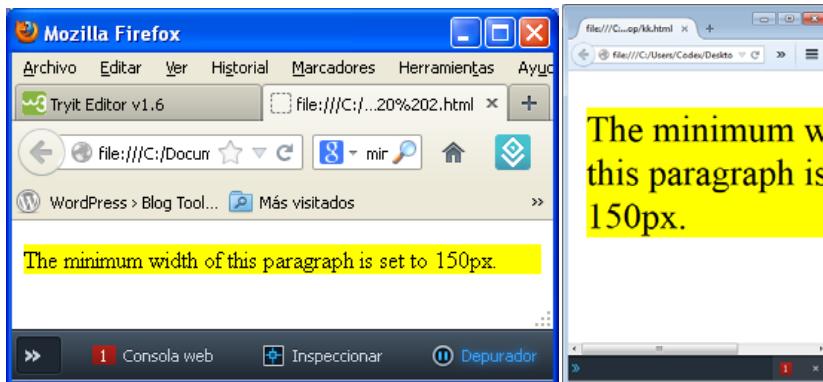
En este ejemplo se muestra cómo establecer la altura mínima de un elemento.

The minimum height of this paragraph is set to 100px.

```
<!DOCTYPE html>
<html>
<head>
<style>
p
{
    min-height: 100px;
    background-color: yellow;
}
</style>
</head>
<body>
<p>The minimum height of this paragraph is set to 100px.</p>
</body>
</html>
```

#### Ajustar el ancho mínimo de un elemento con un valor de píxeles

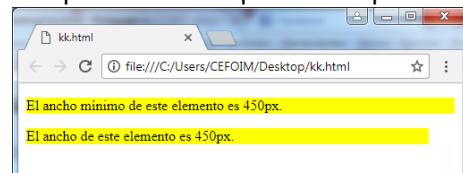
Este ejemplo muestra cómo establecer la anchura mínima de un elemento con un valor de pixel.



```
<!DOCTYPE html>
<html>
<head>
<style>
p
{
    min-width: 450px;
    background-color: yellow;
}
</style>
</head>

<body>
<p>The minimum width of this paragraph is set to 450px.</p>
</body>
</html>
```

### Comparado con un p donde especificamos el ancho



```
<!DOCTYPE html>
<html>
<head>
<style>
p.uno
{
    min-width: 450px;
    background-color: yellow;
}

p.dos
{
    width: 450px;
    background-color: yellow;
}
</style>
</head>

<body>
<p class="uno">El ancho mínimo de este elemento es 450px.</p>
<p class="dos">El ancho de este elemento es 450px.</p>
</body>
</html>
```

## Todas las propiedades de dimensiones CSS

El número en el "CSS" en la columna indica qué versión de CSS se define la propiedad (o CSS1 CSS2).

Propiedad	Descripción	Valores	CSS
height	Establece el alto de un elemento	auto length % inherit	1
max-height	Establece la altura máxima de un elemento	none length % inherit	2
max-width	Define la anchura máxima de un elemento	none length % inherit	2
min-height	Establece la altura mínima de un elemento	length % inherit	2
min-width	Establece el ancho mínimo de un elemento	length % inherit	2
width	Establece el ancho de un elemento	auto length % inherit	1

## CSS Display y Visibility

La propiedad display especifica el como se visualiza un elemento en la pantalla, y la propiedad visibility especifica si un elemento debe ser visible u oculto.



### Cómo ocultar un elemento - display: none o visibility: hidden

Un elemento se puede ocultar mediante el establecimiento de la propiedad display a "none" o estableciendo la propiedad de visibility a "hidden". Sin embargo, hay que darse cuenta que estos dos métodos producen resultados diferentes:

visibility: Oculta un elemento, pero el elemento ocupa el mismo espacio que antes. El elemento se oculta, pero todavía afectan al diseño.

## Ejemplo

```
h1.hidden {visibility:hidden;}\n\n<!DOCTYPE html>\n<html>\n<head>\n<style>\n    h2.hidden {visibility:hidden;}\n</style>\n</head>\n\n<body>\n    <h1>Cabecera visible </h1>\n    <h2 class="hidden">cabecera oculta</h2>\n    <p>Aunque esté oculta, ocupa espacio.</p>\n</body>\n</html>
```

display: none esconde un elemento, y no ocupa espacio. El elemento se oculta, y la página se mostrará como si el elemento no estuviera:

## Ejemplo

```
h1.hidden {display:none;}
```

```
<!DOCTYPE html>\n<html>\n<head>\n<style>\n    h2.hidden {display:none;}\n</style>\n</head>\n<body>\n    <h1>This is a visible heading</h1>\n    <h2 class="hidden">This is a hidden heading</h2>\n    <p>Notice that the hidden heading does not take up space.</p>\n</body>\n</html>
```



Hiding an Element - display:none or visibility:hidden



klematis\_small.jpg



klematis2\_small.jpg



klematis3\_small.jpg

```
<!DOCTYPE html>
<html>
<head>
<script>
function eliminarElemento()
{
    document.getElementById("imgbox1").style.display="none";
}
function cambiarVisibilidad()
{
    document.getElementById("imgbox2").style.visibility="hidden";
}
function reestablecerElementos()
{
    document.getElementById("imgbox1").style.display="block";
    document.getElementById("imgbox2").style.visibility="visible";
}
</script>
</head>
<body>
<div id="imgbox1">
    Box 1<br/>
    <br/>
    <input onclick="eliminarElemento()" value="Borrar" type="button"/>
</div>
<div id="imgbox2">
    Box 2<br/>
    
    <input onclick="cambiarVisibilidad()" value="Esconder" type="button"/>
</div>
<div >
    Box 3<br/>
    
    <input onclick="reestablecerElementos()" value="Restablecer" type="button"/>
</div>
</body>
```

</html>

## CSS Display - Block y inline

Un elemento de bloque es un elemento que ocupa todo el ancho disponible completo, y tiene un salto de línea antes y después de ella.

Ejemplos de elementos de bloque:

- <h1>
- <p>
- <div>

Un elemento en línea sólo ocupa el ancho que sea necesario, y no fuerza saltos de línea.

Ejemplos de elementos in-line:

- <span>
- <a>

## Cambiar la forma en la que un elemento se muestra

Podemos cambiarla visualización de un elemento en línea a un elemento de bloque, o viceversa, puede ser útil para hacer que la página se vea de una manera específica, y todavía siga los estándares web.

El ejemplo siguiente muestra los elementos una de lista como elementos en línea:

### Ejemplo

```
li {display:inline;}

<!DOCTYPE html>
<html>
<head>
<style>
    li{display:inline; }
</style>
</head>
<body>

<p>Display this link list as a horizontal menu:</p>

<ul>
<li><a href="http://google.es" target="_blank">Enlace 1</a></li>
<li><a href="http://google.es" target="_blank">Enlace 2</a></li>
<li><a href="http://google.es" target="_blank">Enlace 3</a></li>
<li><a href="http://google.es" target="_blank">Enlace 4</a></li>
</ul>

</body>
</html>
```

El siguiente ejemplo muestra abarcan elementos como elementos de bloque:

## Ejemplo

```
span {display:block;}

<!DOCTYPE html>
<html>
<head>
<style>
span
{
    display:block;
}
</style>
</head>
<body>

<h2>Nirvana</h2>
<span>Record: MTV Unplugged in New York</span>
<span>Year: 1993</span>
<h2>Radiohead</h2>
<span>Record: OK Computer</span>
<span>Year: 1997</span>

</body>
</html>
```

**Nota:** cambiar el tipo de visualización de un elemento cambia sólo la visualización del elemento en la pantalla, no el tipo de elemento que es. Por ejemplo: Un elemento en línea establecida en display: block no se le permite tener un elemento de bloque anidado dentro de él.

```
<!DOCTYPE html>
<html>
<head>
<style>
li{display:inline; }
</style>
<script>
function displayBlock()
{
    document.getElementById("box1").style.display="block";
}
function displayInline()
{
    document.getElementById("box1").style.display="inline";
}

</script>
</head>
<body>
<ul>
    <li id="box1">
        <a href="http://google.es" target="_blank">
            Enlace 1
        </a>
    </li>
    <li>
        <a href="http://google.es" target="_blank">
            Enlace 2
        </a>
    </li>
</ul>
```

```
<br/>
<input onclick="displayBlock()" value="Block" type="button"/>
<input onclick="displayInline()" value="Inline" type="button"/>
</body>
</html>
```



## Más ejemplos

### Cómo mostrar un elemento como un elemento en línea.

Este ejemplo muestra cómo mostrar un elemento como un elemento en línea.

```
<!DOCTYPE html>
<html>
<head>
<style>
p { display:inline; }
</style>
</head>
<body>
<p>Cuando la propiedad display se establece a "inline" conseguimos que </p>
<p>no haya distancia entre dos elementos.</p>
</body>
</html>
```

### Cómo mostrar un elemento como un elemento de bloque

En este ejemplo se muestra cómo mostrar un elemento como un elemento de bloque.

```
<!DOCTYPE html>
<html>
<head>
<style>
span
{
    display:block;
}
</style>
</head>
<body>

<span>A display property with a value of "block" results in</span><span>a line break between the two
elements.</span>

</body>
</html>
```

### Cómo hacer que se colapse del elemento de tabla

Este ejemplo muestra cómo colapsar elemento de tabla.

Este ejercicio tendrá lógica cuando veamos javascript

En Chrome no funciona.

```
<!DOCTYPE html>
<html>
<head>
<style>
tr.collapse { visibility:collapse; }
</style>
</head>
<body>
<table border="1">
<tr>
```

```
<td>Peter</td>
<td>Griffin</td>
</tr>
<tr class="collapse">
<td>Lois</td>
<td>Griffin</td>
</tr>
</table>
<p><strong>Note:</strong> IE8 and earlier support visibility:collapse only if a !DOCTYPE is specified.</p>
</body>
</html>
```

Uno de los valores más curiosos de `display` es `inline-block`, que crea cajas que son de bloque y en línea de forma simultánea. Una caja de tipo `inline-block` se comporta como si fuera de bloque, pero respecto a los elementos que la rodean es una caja en línea.

El enlace del siguiente ejemplo es de tipo `inline-block`, lo que permite por ejemplo establecerle un tamaño mediante la propiedad `width`:

Loreum ipsum dolor sit amet, consectetuer adipiscing elit. Sed non sem quis tellus vulputate lobortis. Vivamus fermentum, tortor id ornare ultrices, ligula ipsum tincidunt pede, et blandit sem pede suscipit pede. Nulla cursus porta sem. Donec mollis nunc in leo. [\[display: inline-block, width: 25%\]](#) Quisque semper, magna sed pharetra tincidunt, quam urna dapibus dolor, a dignissim sem neque id purus. Etiam luctus viverra nisi. Integer lobortis accumsan felis. Cras venenatis. Morbi cursus, tellus vitae iaculis pulvinar, turpis nibh posuere nisl, sed vehicula massa orci at dui. Morbi pede ipsum, porta quis, venenatis et, ullamcorper in, metus. Nulla facilisi. Quisque laoreet molestie mi. Ut mollis elit eget urna.

Si tu navegador soporta el valor `inline-block`, el ejemplo anterior se debe visualizar tal y como muestra la siguiente imagen:

Loreum ipsum dolor sit amet, consectetuer adipiscing elit. Sed non sem quis tellus vulputate lobortis. Vivamus fermentum, tortor id ornare ultrices, ligula ipsum tincidunt pede, et blandit sem pede suscipit pede. Nulla cursus porta sem. Donec mollis nunc in leo. [\[display: inline-block, width: 25%\]](#) Quisque semper, magna sed pharetra tincidunt, quam urna dapibus dolor, a dignissim sem neque id purus. Etiam luctus viverra nisi. Integer lobortis accumsan felis. Cras venenatis. Morbi cursus, tellus vitae iaculis pulvinar, turpis nibh posuere nisl, sed vehicula massa orci at dui. Morbi pede ipsum, porta quis, venenatis et, ullamcorper in, metus. Nulla facilisi. Quisque laoreet molestie mi. Ut mollis elit eget urna.

## soporte inline-block

Selector	IE 5-5	IE 6	IE 7 pure	IE8b2 as IE7	IE8b2 as IE8	FF 2	FF 3.0	FF 3.1b	Saf 3.0 Win	Saf 3.1 Win	iPhone 3G	Chrome 0.3	Opera 9.51	Opera 9.62	Konqueror 3.5.7
<b>block</b>	yes					yes			yes			yes			yes
<b>inline</b>	yes					yes			yes			yes			yes
<b>none</b>	yes					yes			yes			yes			yes
<b>inline-block</b>	no	incomplete	yes	no		yes			yes			yes			yes

IE 6/7 accepts the value only on elements with a natural display: `inline`.

## Ejemplo

```
<!DOCTYPE html>
<html>
<head>
<style>

</style>
</head>

<body>
<strong> hola</strong>
<div style="width: 100px; display: inline-block; background-color: red; border: 3px solid black;">display: inline-block</div>
<div style="width: 100px; display: inline-block; background-color: red; border: 3px solid black;">display: inline-block</div>
<div style="width: 100px; display: inline-block; background-color: red; border: 3px solid black;">display: inline-block</div>
<strong> hola</strong></body>
</html>
```

hola `display: inline-block` `display: inline-block` `display: inline-block` hola

## Display Flex

<https://www.yunbitsoftware.com/blog/2017/03/30/flexbox-css3-tutorial-descripcion-ejemplos-html/>

[https://www.w3schools.com/css/css3\\_flexbox.asp](https://www.w3schools.com/css/css3_flexbox.asp)  
[https://developer.mozilla.org/es/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout/Conceptos\\_Basicos\\_de\\_Flexbox](https://developer.mozilla.org/es/docs/Web/CSS/CSS_Flexible_Box_Layout/Conceptos_Basicos_de_Flexbox)



FlexBox facilita el diseño del layout permitiéndonos hacerlo flexible y responsive sin utilizar float o posicionamientos.

Flexbox viene de “Flexible Box Layout”, que se puede traducir como “Diseño de caja flexible”, y nos aporta una magnífica solución para todos nuestros desarrollos “responsive”. Lo que nos permite es crear un conjunto de elementos flexibles que se adaptan automáticamente a su contenedor y con el que podemos controlar parámetros tales como la alineación, dirección (horizontal/vertical), ajuste de la fila según tamaños y multitud de posibilidades

## Elementos Flexbox

Un Layout Flexible debe tener un elemento padre con la propiedad *display* establecida a *flex*.

los elementos hijos del contenedor flexible serán de forma automática elementos flexibles.

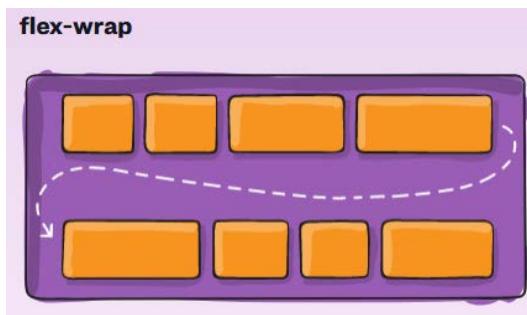
El elemento de arriba representa un flex container (el área azul) con tres elementos.

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  background-color: DodgerBlue;
}

.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  padding: 20px;
  font-size: 30px;
}
</style>
</head>
<body>

<div class="flex-container">
<div>1</div>
<div>2</div>
<div>3</div>
</div>
<p>Un Layout Flexible debe tener un elemento padre con la propiedad
<em>display</em> establecida a<em>flex</em>. </p>
<p>los elementos hijos del contenedor flexible serán de forma automática elementos
flexibles.</p>
</body>
```

Puedo hacer que parezcan elementos flotantes utilizando flex-wrap: wrap;



```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  flex-wrap: wrap;
  background-color: DodgerBlue;
}

.flex-container > div {
  background-color: #f1f1f1;
  margin: 10px;
  padding: 20px;
  font-size: 30px;
}
```

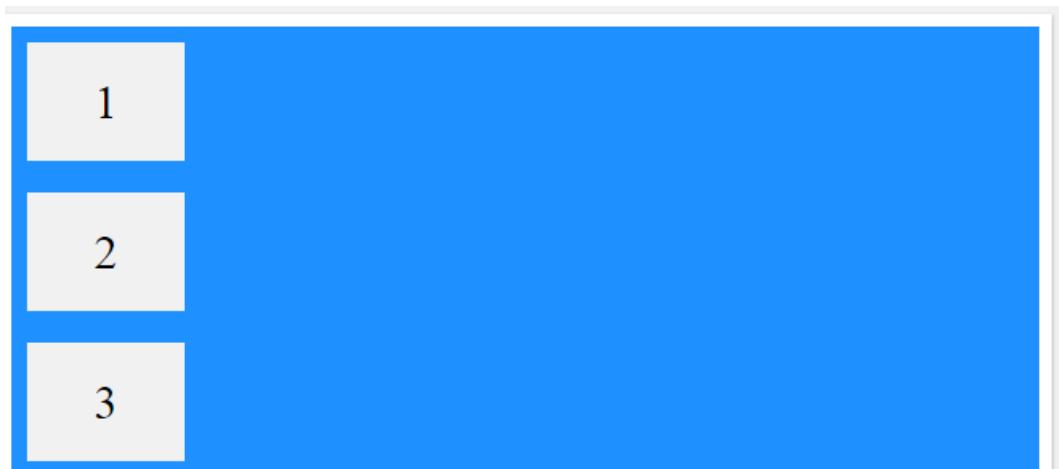
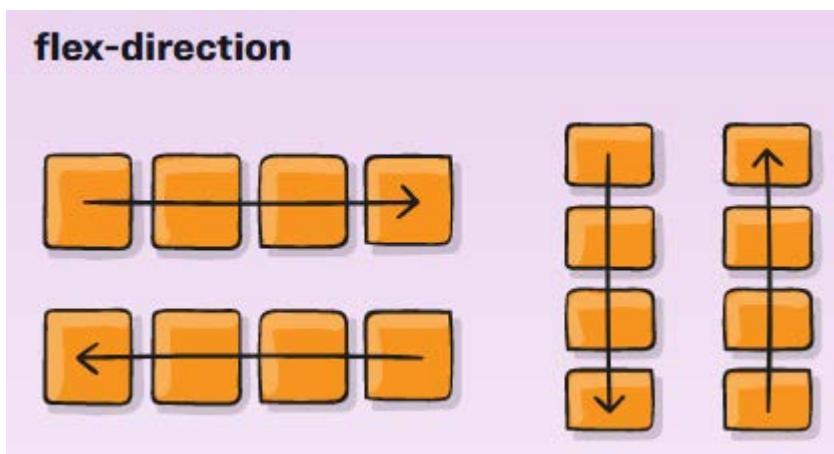
```

}
</style>
</head>
<body>
<div class="flex-container">
<div>1</div>
<div>2</div>
<div>3</div>
</div>
<p>Un Layout Flexible debe tener un elemento padre con la propiedad <em>display</em> establecida a <em>flex</em>. </p>
<p>los elementos hijos del contenedor flexible serán de forma automática elementos flexibles.</p>
</body>

```

Las propiedades del flex container son:

- [flex-direction](#)
- [flex-wrap](#)
- [flex-flow](#)
- [justify-content](#)
- [align-items](#)
- [align-content](#) /\*No hay nada en los apuntes \*/



A container with the `flex-direction` property set to `column` will display its items from top to bottom.

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  flex-direction: column;
  background-color: DodgerBlue;
}
.flex-container > div {
  background-color: #f1f1f1;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<div class="flex-container">
<div>1</div>
<div>2</div>
<div>3</div>
</div>
<p>A container with the flex-direction property set to
<em>column</em> will display its items from top to bottom.</p>
</body>
</html>
```

El valor *column-reverse* coloca os elementos verticalmente pero de abajo a arriba:

```
.flex-container {
  display: flex;
  flex-direction: column-reverse;
}
```



El valor de `row` apila los elementos flexibles horizontalmente (de izquierda a derecha) (comportamiento habitual):

```
.flex-container {  
  display: flex;  
  flex-direction: row;  
}
```

El valor `row-reverse` apila los elementos flexibles horizontalmente (pero de derecha a izquierda):

```
.flex-container {  
  display: flex;  
  flex-direction: row-reverse;  
}
```



## La propiedad flex-wrap

La propiedad `flex-wrap` especifica si los elementos flex deberían hacer wrap o no.

Los siguientes ejemplos tienen 12 elementos flex, para demostrar mejor la propiedad `flex-wrap`.

1	2	3	4	5	6	7
8	9	10	11	12		

## Ejemplo

El valor `wrap` especifica que los elementos flexibles se ajustarán si es necesario:

```
.flex-container {  
  display: flex;  
  flex-wrap: wrap;  
}
```

## Example

El valor de `nowrap` especifica que los elementos flexibles no se ajustarán (por defecto):

```
.flex-container {  
  display: flex;  
  flex-wrap: nowrap;  
}
```

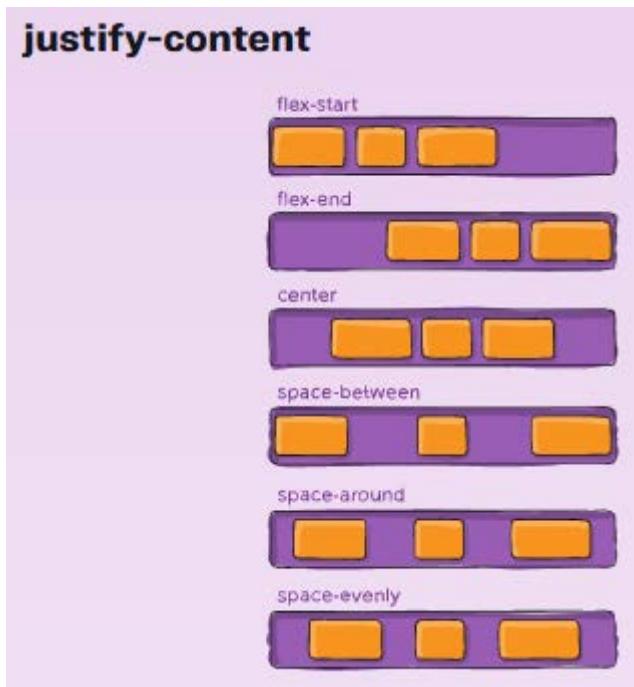
El valor `wrap-reverse` especifica que los elementos flexibles se ajustarán si es necesario, en orden inverso

```
.flex-container {  
  display: flex;  
  flex-wrap: wrap-reverse;  
}
```

11	12				
6	7	8	9	10	
1	2	3	4	5	

# La propiedad `justify-content`

La propiedad `justify-content` se usa para alinear los elementos flexibles:



## Ejemplo

El valor `center` alinea los elementos flexibles en el centro del contenedor:

```
.flex-container {  
  display: flex;  
  justify-content: center;  
}
```

## Ejemplo

El valor `flex-start` alinea los elementos flexibles al comienzo del contenedor (esto es predeterminado):

```
.flex-container {  
  display: flex;  
  justify-content: flex-start;  
}
```

El valor `flex-end` alinea los elementos flexibles al final del contenedor:

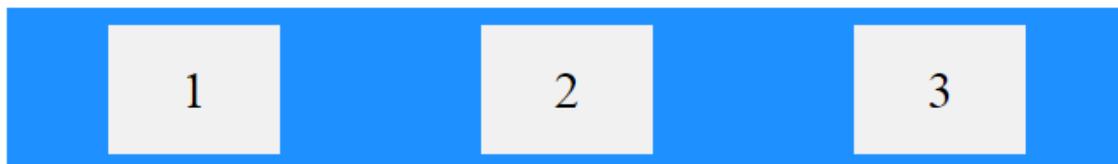
```
.flex-container {  
  display: flex;  
  justify-content: flex-end;  
}
```

The "justify-content: flex-end;" aligns the flex items at the end of the container:



El valor `space-around` muestra los elementos flexibles con espacio antes, entre y después de los items:

```
.flex-container {  
  display: flex;  
  justify-content: space-around;  
}
```



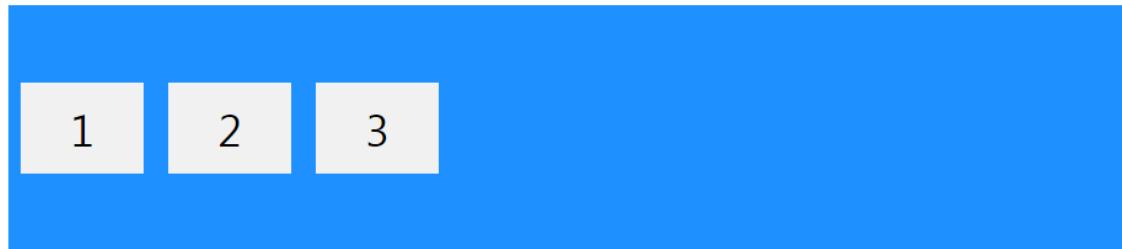
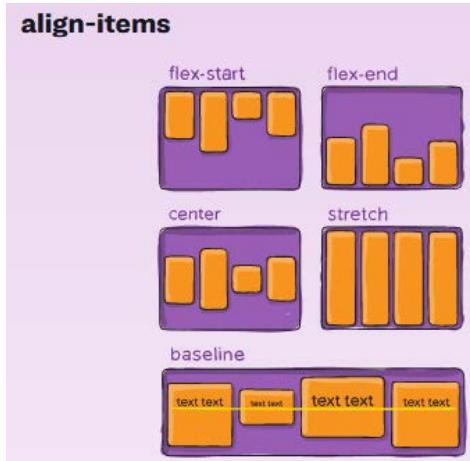
El valor de `space-between` muestra los elementos flexibles con espacio entre los items:

```
.flex-container {  
  display: flex;  
  justify-content: space-between;  
}
```



## La propiedad align-items

La propiedad `align-items` es utilizada para alinear los elementos flexibles verticalmente.



En estos ejemplos, utilizamos un contenedor de 200 píxeles de altura para demostrar mejor la propiedad **align-items**.

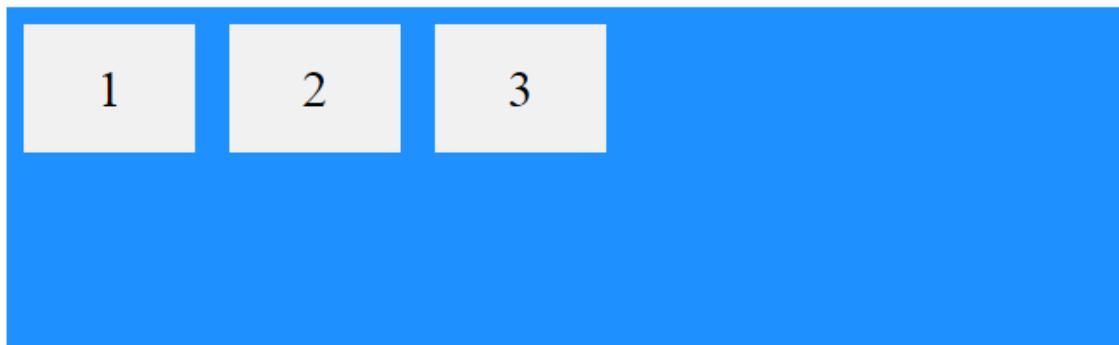
El valor **center** alinea los elementos flexibles en el medio del contenedor::

```
.flex-container {  
  display: flex;  
  height: 200px;  
  align-items: center;  
}
```

El valor **flex-start** alinea los elementos flexibles en la parte superior del contenedor

```
.flex-container {  
  display: flex;  
  height: 200px;  
  align-items: flex-start;  
}
```

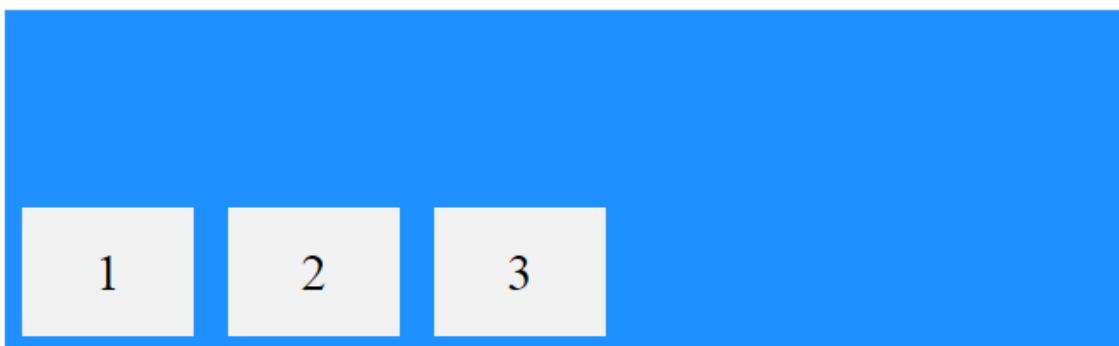
The "align-items: flex-start;" aligns the flex items at the top of the container:



El valor **flex-end** alinea los elementos flexibles en la parte inferior del contenedor:

```
.flex-container {  
  display: flex;  
  height: 200px;  
  align-items: flex-end;  
}
```

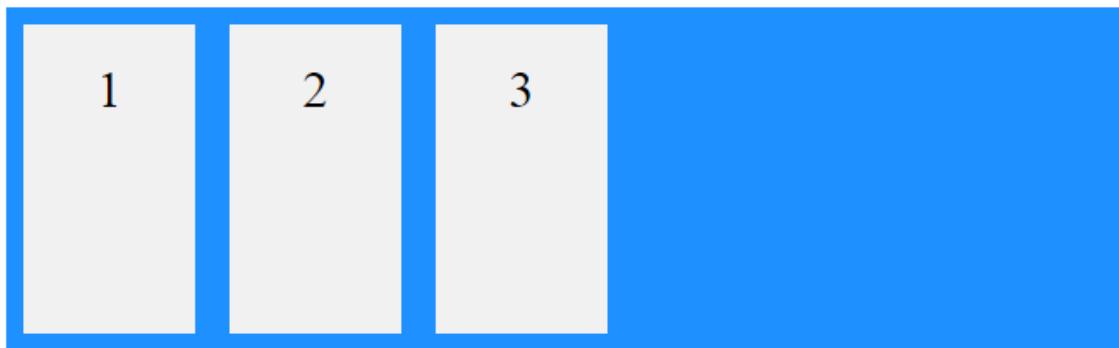
The "align-items: flex-end;" aligns the flex items at the bottom of the container:



El valor de **stretch** estira los elementos flexibles para llenar el contenedor (esto es predeterminado):

```
.flex-container {  
  display: flex;  
  height: 200px;  
  align-items: stretch;  
}
```

The "align-items: stretch;" stretches the flex items to fill the container (this is default):

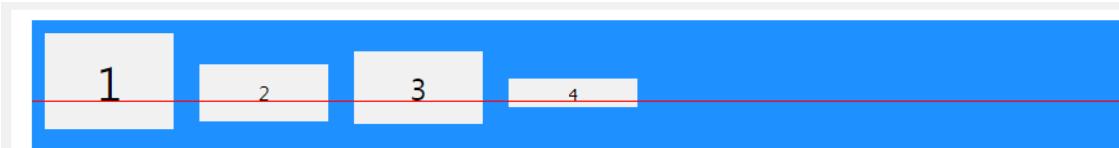


## Ejemplo

El valor de **baseline** alinea los elementos flexibles por la parte de abajo:

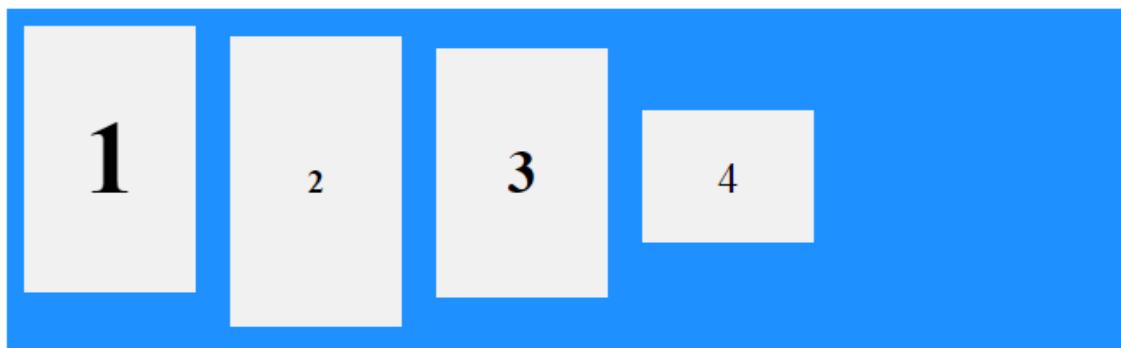
```
.flex-container {  
  display: flex;  
  height: 200px;  
  align-items: baseline;  
}
```

**Note:** el ejemplo usa diferentes tamaños de fuente para demostrar que los elementos se alinean por la parte de abajo del texto:



```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
.flex-container {  
  display: flex;  
  height: 200px;  
  align-items: baseline;  
  background-color: DodgerBlue;  
}  
  
.flex-container > div {  
  background-color: #f1f1f1;  
  width: 100px;  
  margin: 10px;  
  text-align: center;  
  line-height: 75px;
```

```
font-size: 30px;  
}  
</style>  
</head>  
<body>  
<h1>The align-items Property</h1>  
  
<p>The "align-items: baseline;" aligns the flex items such as their  
baselines aligns:</p>  
  
<div class="flex-container">  
<div><h1>1</h1></div>  
<div><h6>2</h6></div>  
<div><h3>3</h3></div>  
<div><small>4</small></div>  
</div>  
</body>  
</html>
```



## Centrado perfecto

En el siguiente ejemplo resolveremos un problema de estilo muy común: el centrado perfecto

**SOLUCIÓN:** establece las propiedades `justify-content` y `align-items:center`, y el elemento flexible estará perfectamente centrado

### Ejemplo

```
.flex-container {  
  display: flex;  
  height: 300px;  
  justify-content: center;
```

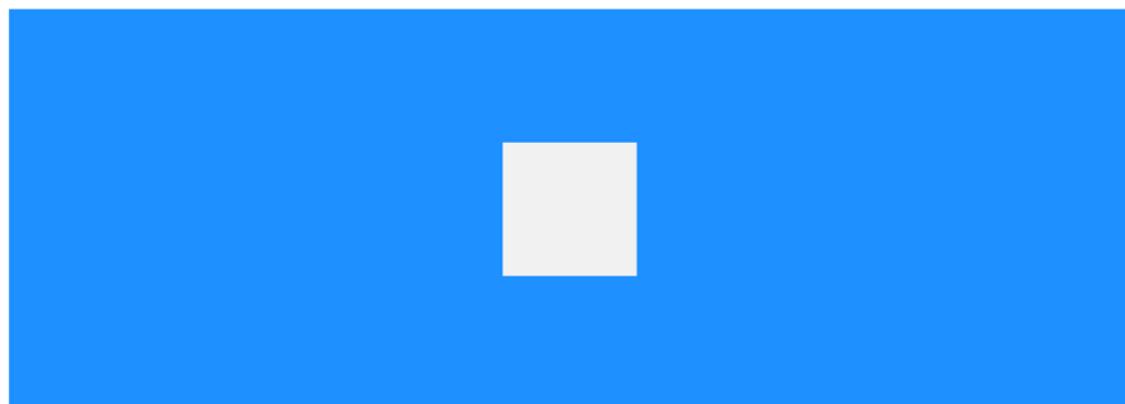
```
align-items: center;  
}
```

## Ejemplo

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
.flex-container {  
    display: flex;  
    justify-content: center;  
    align-items: center;  
    height: 300px;  
    background-color: DodgerBlue;  
}  
  
.flex-container>div {  
    background-color: #f1f1f1;  
    color: white;  
    width: 100px;  
    height: 100px;  
}  
</style>  
</head>  
<body>  
<h1>Centrado Perfecto</h1>  
  
<p>Un contenedor con las propiedades justify-content y align-items establecidas a<em> center </em> alinearán los elementos en el centro (en ambos ejes)</p>  
  
<div class="flex-container">  
<div></div>  
</div>  
</body>  
</html>
```

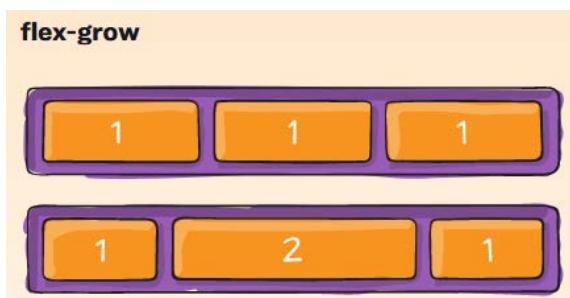
## Centrado Perfecto

Un contenedor con las propiedades `justify-content` y `align-items` establecidas a `center` alineará los elementos en el centro (en ambos ejes)



## La propiedad flex-grow

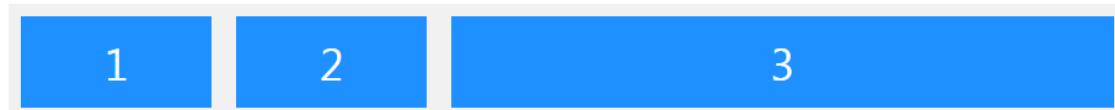
La propiedad `flex-grow` especifica cuánto crecerá un elemento flexible en relación con el resto de los elementos flexibles.



El valor debe ser un número, el valor predeterminado es 0.

### Ejemplo

Haremos que el tercer elemento flexible crezca ocho veces más rápido que los otros elementos flexibles:



```
<div class="flex-container">
  <div style="flex-grow: 1">1</div>
  <div style="flex-grow: 1">2</div>
  <div style="flex-grow: 8">3</div>
</div>
```

## Código completo

```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  align-items: stretch;
  background-color: #f1f1f1;
}

.flex-container > div {
  background-color: DodgerBlue;
  color: white;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
<h1>La propiedad flex-grow </h1>

<p>Hacemos que el tercer elemento flexible crezca ocho veces más rápido que los otros elementos flexibles:</p>

<div class="flex-container">
<div style="flex-grow: 1">1</div>
<div style="flex-grow: 1">2</div>
<div style="flex-grow: 8">3</div>
</div>

</body>
</html>
```

## La propiedad flex-shrink

La propiedad **flex-shrink** especifica cuánto se reducirá un elemento flexible en relación con el resto de los elementos flexibles.

El valor debe ser un número, el valor predeterminado es 1.

## Ejemplo

No permitiremos que el tercer elemento flexible se encoja tanto como los otros elementos flexibles:

The `flex-shrink` property specifies how much a flex item will shrink relative to the rest of the flex items.



The value must be a number, default value is 1.

```
<div class="flex-container">
  <div>1</div>
  <div>2</div>
  <div style="flex-shrink: 0">3</div>
  <div>4</div>
  <div>5</div>
  <div>6</div>
  <div>7</div>
  <div>8</div>
  <div>9</div>
  <div>10</div>
</div>
```

[Código completo](#)

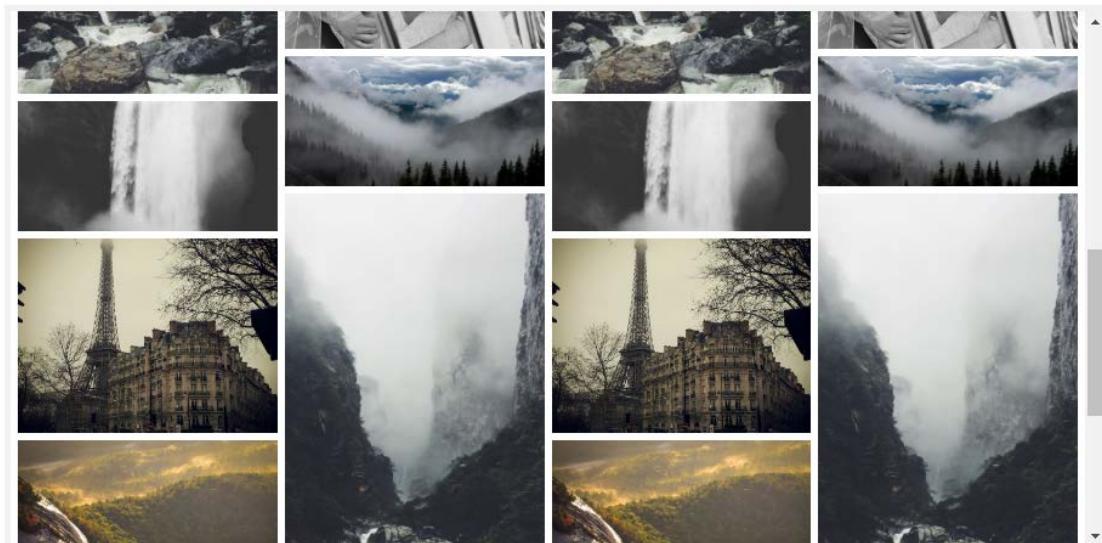
```
<!DOCTYPE html>
<html>
<head>
<style>
.flex-container {
  display: flex;
  align-items: stretch;
  background-color: #f1f1f1;
}

.flex-container > div {
  background-color: DodgerBlue;
  color: white;
  width: 100px;
  margin: 10px;
  text-align: center;
  line-height: 75px;
  font-size: 30px;
}
</style>
</head>
<body>
```

```
<h1>La propiedad flex-shrink</h1>
<p>No permitiremos que el tercer elemento flexible se encoja tanto como los
    otros elementos flexibles</p>
<div class="flex-container">
    <div>1</div>
    <div>2</div>
    <div style="flex-shrink: 0">3</div>
    <!--Probad también con 0.5-->
    <div>4</div>
    <div>5</div>
    <div>6</div>
    <div>7</div>
    <div>8</div>
    <div>9</div>
    <div>10</div>
</div>
</body>
</html>
https://css-tricks.com/snippets/css/a-guide-to-flexbox/
```

## Responsive Image Gallery using Flexbox

Use flexbox to create a responsive image gallery that varies between four, two or full-width images, depending on screen size:



```
<!DOCTYPE html>
```

```
<html>
```

```
<style>
```

```
* {
```

```
/*La propiedad box-sizing nos permite incluir el relleno y el borde en  
el ancho y alto total de un elemento. Si establece box-sizing: border-  
box; en un elemento, el relleno y el borde están incluidos en el ancho  
y la altura*/
```

```
    box-sizing: border-box;
```

```
}
```

```
body {
```

```
    margin: 0;
```

```
    font-family: Arial;
```

```
}
```

```
.header {
```

```
    text-align: center;
```

```
    padding: 32px;
```

```
}
```

```
.row {
```

```
    display: flex;
```

```
    flex-wrap: wrap;
```

```
    padding: 0 4px;
```

```
}
```

```
/* Create four equal columns that sits next to each other */
```

```
.column {
```

```
flex: 25%;  
  
max-width: 25%;  
  
padding: 0 4px;  
  
}
```

```
.column img {  
  
margin-top: 8px;  
  
vertical-align: middle;  
  
}
```

```
/* Responsive layout - makes a two column-layout instead of four columns */
```

```
@media (max-width: 800px) {  
  
.column {  
  
flex: 50%;  
  
max-width: 50%;  
  
}  
  
}
```

```
/* Responsive layout - makes the two columns stack on top of each other instead of next to  
each other */
```

```
@media (max-width: 600px) {  
  
.column {  
  
flex: 100%;  
  
max-width: 100%;  
  
}
```

}

</style>

<body>

<!-- Header -->

<div class="header">

<h1>Responsive Image Grid</h1>

<p>Resize the browser window to see the responsive effect.</p>

</div>

<!-- Photo Grid -->

<div class="row">

<div class="column">















</div>

<div class="column">





```
  
  
  
  
  
  
  
  
</div>
```

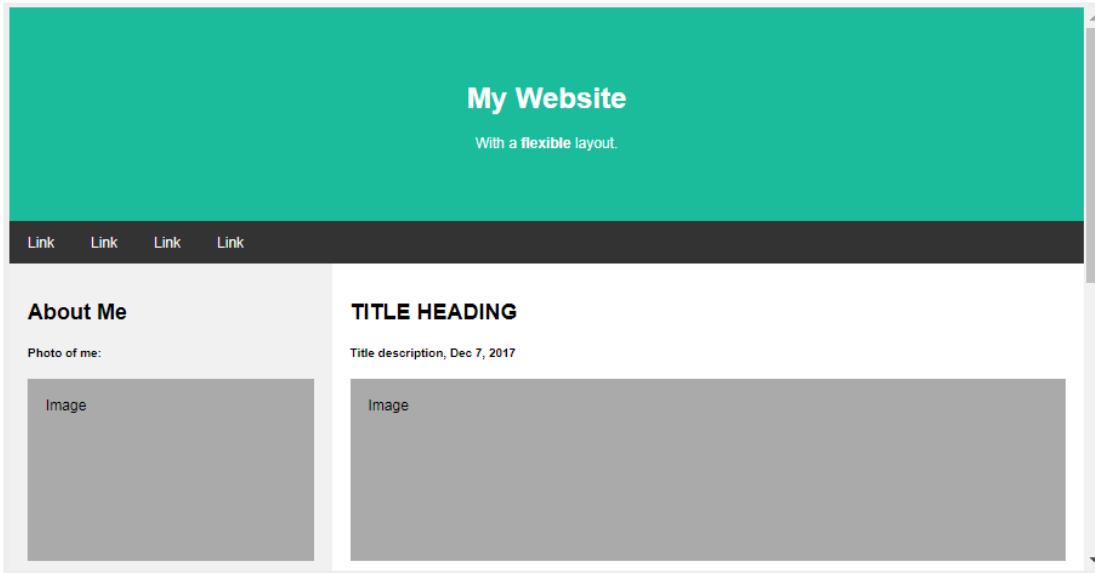
```
<div class="column">  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
</div>
```

```
<div class="column">  
  
  
  
  
  
  
  
  
  
  
  
  
  
</div>  
  
</div>
```

</body>

# Responsive Website using Flexbox

Use flexbox to create a responsive website, containing a flexible navigation bar and flexible content:



```
<!DOCTYPE html>

<html>
  <head>
    <title>Page Title</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <style>
      * {
        box-sizing: border-box;
      }
    </style>
  </head>
  <body>
    <header>
      <h1>My Website</h1>
      <p>With a flexible layout.</p>
    </header>
    <nav>
      <ul>
        <li>Link</li>
        <li>Link</li>
        <li>Link</li>
        <li>Link</li>
      </ul>
    </nav>
    <main>
      <div>
        <div>
          <h2>About Me</h2>
          <p>Photo of me:</p>
          <img alt="Placeholder image for photo of me" data-bbox="164 421 341 501"/>
        </div>
        <div>
          <h2>TITLE HEADING</h2>
          <p>Title description, Dec 7, 2017</p>
          <img alt="Placeholder image for title description" data-bbox="364 421 821 501"/>
        </div>
      </div>
    </main>
  </body>
</html>
```

*/\* Style the body \*/*

```
body {  
    font-family: Arial;  
    margin: 0;  
}
```

*/\* Header/logo Title \*/*

```
.header {  
    padding: 60px;  
    text-align: center;  
    background: #1abc9c;  
    color: white;  
}
```

*/\* Style the top navigation bar \*/*

```
.navbar {  
    display: flex;  
    background-color: #333;  
}
```

*/\* Style the navigation bar links \*/*

```
.navbar a {  
    color: white;  
    padding: 14px 20px;  
    text-decoration: none;
```

text-align: center;

}

/\* Change color on hover \*/

.navbar a:hover {

background-color: #ddd;

color: black;

}

/\* Column container \*/

.row {

display: flex;

flex-wrap: wrap;

}

/\* Create two unequal columns that sits next to each other \*/

/\* Sidebar/left column \*/

.side {

flex: 30%;

background-color: #f1f1f1;

padding: 20px;

}

/\* Main column \*/

.main {

```
flex: 70%;  
  
background-color: white;  
  
padding: 20px;  
  
}  
  
  
/* Fake image, just for this example */  
  
.fakeimg {  
  
background-color: #aaa;  
  
width: 100%;  
  
padding: 20px;  
  
}  
  
  
/* Footer */  
  
.footer {  
  
padding: 20px;  
  
text-align: center;  
  
background: #ddd;  
  
}  
  
  
/* Responsive layout - when the screen is less than 700px wide, make the two columns stack  
on top of each other instead of next to each other */  
  
@media screen and (max-width: 700px) {  
  
.row, .navbar {  
  
flex-direction: column;  
  
}
```

```
}

</style>

</head>

<body>

<!-- Note --&gt;

&lt;div style="background:yellow;padding:5px"&gt;

&lt;h4 style="text-align:center"&gt;Resize the browser window to see the responsive effect.&lt;/h4&gt;

&lt;/div&gt;

<!-- Header --&gt;

&lt;div class="header"&gt;

&lt;h1&gt;My Website&lt;/h1&gt;

&lt;p&gt;With a &lt;strong&gt;flexible&lt;/strong&gt; layout.&lt;/p&gt;

&lt;/div&gt;

<!-- Navigation Bar --&gt;

&lt;div class="navbar"&gt;

&lt;a href="#"&gt;Link&lt;/a&gt;

&lt;a href="#"&gt;Link&lt;/a&gt;

&lt;a href="#"&gt;Link&lt;/a&gt;

&lt;a href="#"&gt;Link&lt;/a&gt;

&lt;/div&gt;

<!-- The flexible grid (content) --&gt;</pre>
```

```
<div class="row">
<div class="side">
<h2>About Me</h2>
<h5>Photo of me:</h5>
<div class="fakeimg" style="height:200px;">Image</div>
<p>Some text about me in culpa qui officia deserunt mollit anim..</p>
<h3>More Text</h3>
<p>Lorem ipsum dolor sit ame.</p>
<div class="fakeimg" style="height:60px;">Image</div><br>
<div class="fakeimg" style="height:60px;">Image</div><br>
<div class="fakeimg" style="height:60px;">Image</div>
</div>
<div class="main">
<h2>TITLE HEADING</h2>
<h5>Title description, Dec 7, 2017</h5>
<div class="fakeimg" style="height:200px;">Image</div>
<p>Some text..</p>
<p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipisciing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>
<br>
<h2>TITLE HEADING</h2>
<h5>Title description, Sep 2, 2017</h5>
<div class="fakeimg" style="height:200px;">Image</div>
<p>Some text..</p>
```

<p>Sunt in culpa qui officia deserunt mollit anim id est laborum consectetur adipisciing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.</p>

</div>

</div>

<!-- Footer -->

<div class="footer">

<h2>Footer</h2>

</div>

</body>

</html>

## CSS Posicionamiento

El posicionamiento puede ser difícil a veces!

Decidir qué elemento va a mostrar en frente!

Los elementos pueden superponerse!

### Posicionamiento

Las propiedades de posicionamiento CSS permiten colocar un elemento. También se puede colocar un elemento detrás de otro, y especificar qué debe ocurrir cuando el contenido de un elemento es demasiado grande.

Los elementos se pueden colocar usando las propiedades `top`, `bottom`, `left`, y `right`. Sin embargo, estas propiedades no funcionarán a menos que la propiedad `position` se establezca en primer lugar. También funcionan de forma diferente dependiendo del método de posicionamiento.

Hay cuatro métodos de posicionamiento diferentes.

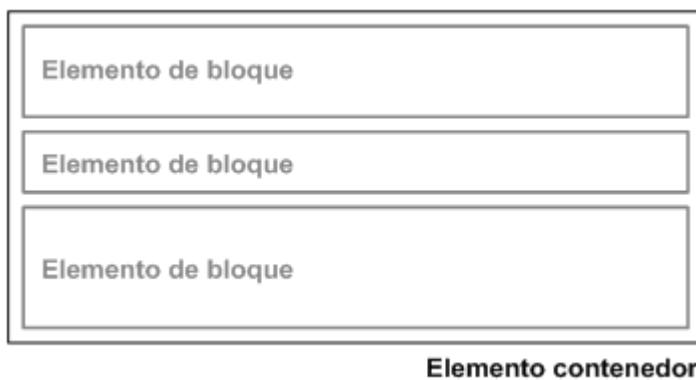
## Posicionamiento normal o estático

Los elementos HTML estático se colocan de forma predeterminada. Un elemento posicionado como estático siempre se coloca de acuerdo con el flujo normal de la página.

Los elementos posicionados como estáticos no se ven afectados por las propiedades top, bottom, left, y right.

El posicionamiento normal o estático es el modelo que utilizan por defecto los navegadores para mostrar los elementos de las páginas. En este modelo, sólo se tiene en cuenta si el elemento es de bloque o en línea, sus propiedades **width** y **height** y su contenido.

Los elementos de bloque forman lo que CSS denomina "*contextos de formato de bloque*". En este tipo de contextos, las cajas se muestran una debajo de otra comenzando desde el principio del elemento contenedor. La distancia entre las cajas se controla mediante los márgenes verticales.

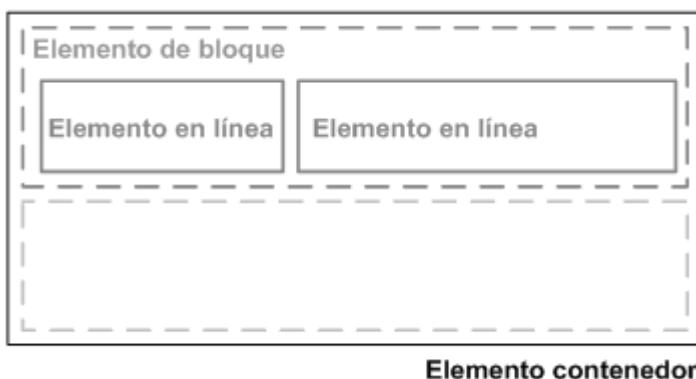


**Figura 5.2** Posicionamiento normal de los elementos de bloque

Si un elemento se encuentra dentro de otro, el elemento padre se llama "*elemento contenedor*" y determina tanto la posición como el tamaño de todas sus cajas interiores.

Si un elemento no se encuentra dentro de un elemento contenedor, entonces su elemento contenedor es el elemento `<body>` de la página. Normalmente, la anchura de los elementos de bloque está limitada a la anchura de su elemento contenedor, aunque en algunos casos sus contenidos pueden desbordar el espacio disponible.

Los elementos en línea forman los "*contextos de formato en línea*". En este tipo de contextos, las cajas se muestran una detrás de otra de forma horizontal comenzando desde la posición más a la izquierda de su elemento contenedor. La distancia entre las cajas se controla mediante los márgenes laterales.



**Figura 5.3** Posicionamiento normal de los elementos en línea

Si las cajas en línea ocupan más espacio del disponible en su propia línea, el resto de cajas se muestran en las líneas inferiores. Si las cajas en línea ocupan un espacio menor que su propia línea, se puede

controlar la distribución de las cajas mediante la propiedad `text-align` para centrarlas, alinearlas a la derecha o justificarlas.

## Posicionamiento fijo

Un elemento con posición fija está situado con relación a la ventana del navegador.

No se moverá incluso si la ventana se desplaza:

El estándar CSS considera que el posicionamiento fijo es un caso particular del posicionamiento absoluto, ya que sólo se diferencian en el comportamiento de las cajas posicionadas.

Cuando una caja se posiciona de forma fija, la forma de obtener el origen de coordenadas para interpretar su desplazamiento es idéntica al posicionamiento absoluto. De hecho, si el usuario no mueve la página HTML en la ventana del navegador, no existe ninguna diferencia entre estos dos modelos de posicionamiento.

La principal característica de una caja posicionada de forma fija es que su posición es inamovible dentro de la ventana del navegador. El posicionamiento fijo hace que las cajas no modifiquen su posición ni aunque el usuario suba o baje la página en la ventana de su navegador.

Si la página se visualiza en un medio paginado (por ejemplo en una impresora) las cajas posicionadas de forma fija se repiten en todas las páginas. Esta característica puede ser útil para crear encabezados o pies de página en páginas HTML preparadas para imprimir.

Internet Explorer 6 y las versiones anteriores no lo soportan.

### Ejemplo

```
p.pos_fixed
{
position:fixed;
top:30px;
right:5px;
}

<!DOCTYPE html>
<html>
<head>
<style>
p.pos_fixed
{
    position:fixed;
    top: 30px;
    right: 5px;
}
</style>
</head>
<body>

<p class="pos_fixed">Algo de texto</p>
<p><strong>Nota:</strong>En IE7 y IE8 hay que poner el !DOCTYPE.</p>
<p>Algo de texto</p><p>Algo de texto</p>
```

**Nota:** En IE7 e IE8 solo es compatible si se especifica un DOCTYPE!.

Los elementos posicionados fijos se eliminan del flujo normal. El documento y los otros elementos se comportan como el elemento fijo no existiera.

Los elementos posicionados fijos pueden superponerse a otros elementos.

Ejemplos:

<http://www.csszengarden.com/136/>

<http://www.caracolesdearagon.com/>

[http://www.csszengarden.com/136/r3\\_zen666\\_tombstone.png](http://www.csszengarden.com/136/r3_zen666_tombstone.png)

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
p.pos_fixed
```

```
{
```

```
    position:fixed;  
  
    bottom:0;  
  
    left: 540px;  
  
    background:url(r3_zen666_tombstone.png) no-repeat top left;  
  
    width:227px;  
  
    height:267px;  
  
    margin:0px;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<p class="pos_fixed"></p>
```

```
<p>Ohh, que miedito!!!</p>
```

```
<p>Ohh, que miedito!!!</p>
```

<p>Ohh, que miedito!!!</p>

</body>

</html>

Menú fijo en la parte de arriba

Inicio Noticias Contacto  
**Menu Fijado a la parte de arriba**

Haz scroll para ver el efecto

  Lorem ipsum dolor sit amet, consectetur adipiscing elit...

  Lorem ipsum dolor sit amet, consectetur adipiscing elit...

<!DOCTYPE html>

<html>

<head>

<style>

body {

  margin:0;

  margin-top:50px;

}

.navbar {

  opacity: 0.9;

  padding-top:10px;

  background-color: #333;

  position: fixed;

  top: 0;

  width: 100%;

  height:32px;

}

.navbar a {

  color: #f2f2f2;

  padding:13px;

  text-decoration: none;

  font-size: 17px;

{

```
.navbar a:hover {
```

```
    background: #c0c0c0;
```

```
    color: black;
```

{}

&lt;/style&gt;

&lt;/head&gt;

&lt;body&gt;

&lt;div class="navbar"&gt;

&lt;a href="#home"&gt;Inicio&lt;/a&gt;

&lt;a href="#news"&gt;Noticias&lt;/a&gt;

&lt;a href="#contact"&gt;Contacto&lt;/a&gt;

&lt;/div&gt;

&lt;div&gt;

&lt;h1&gt;Menu Fijado a la parte de arriba&lt;/h1&gt;

&lt;h2&gt;Haz scroll para ver el efecto&lt;/h2&gt;

&lt;p&gt;Lorem ipsum dolor sit amet, consectetuer adipiscing elit...&lt;/p&gt;

```

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>
</div>
</body>
</html>

```

## Posicionamiento relativo

Un elemento con posicionamiento relativo está situado relativamente a su posición normal.

El estándar CSS considera que el posicionamiento relativo es un caso particular del posicionamiento normal, aunque en la práctica presenta muchas diferencias.

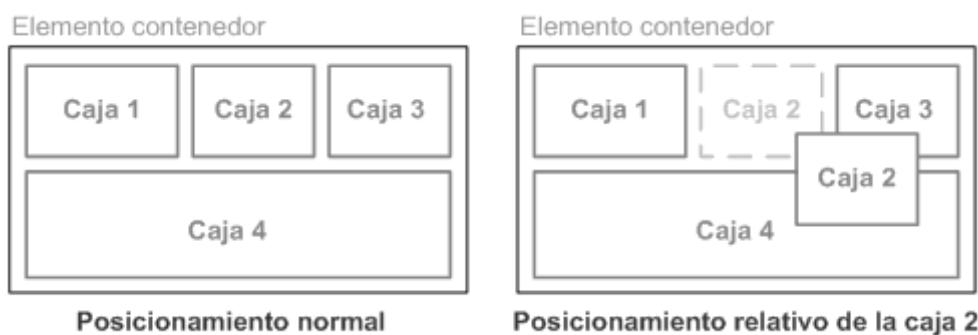
El posicionamiento relativo desplaza una caja respecto de su posición original establecida mediante el posicionamiento normal. El desplazamiento de la caja se controla con las propiedades `top`, `right`, `bottom` y `left`.

El valor de la propiedad `top` se interpreta como el desplazamiento entre el borde superior de la caja en su posición final y el borde superior de la misma caja en su posición original.

De la misma forma, el valor de las propiedades `left`, `right` y `bottom` indica respectivamente el desplazamiento entre el borde izquierdo/derecho/inferior de la caja en su posición final y el borde izquierdo/derecho/inferior de la caja original.

Por tanto, la propiedad `top` se emplea para mover las cajas de forma descendente, la propiedad `bottom` mueve las cajas de forma ascendente, la propiedad `left` se utiliza para desplazar las cajas hacia la derecha y la propiedad `right` mueve las cajas hacia la izquierda. Este comportamiento parece poco intuitivo y es causa de errores cuando se empiezan a diseñar páginas con CSS. Si se utilizan valores negativos en las propiedades `top`, `right`, `bottom` y `left`, su efecto es justamente el inverso.

El desplazamiento relativo de una caja no afecta al resto de cajas adyacentes, que se muestran en la misma posición que si la caja desplazada no se hubiera movido de su posición original.



**Figura 5.4** Ejemplo de posicionamiento relativo de un elemento

En la imagen anterior, la caja 2 se ha desplazado lateralmente hacia la derecha y verticalmente de forma descendente. Como el resto de cajas de la página no modifican su posición, se producen solapamientos entre los contenidos de las cajas.

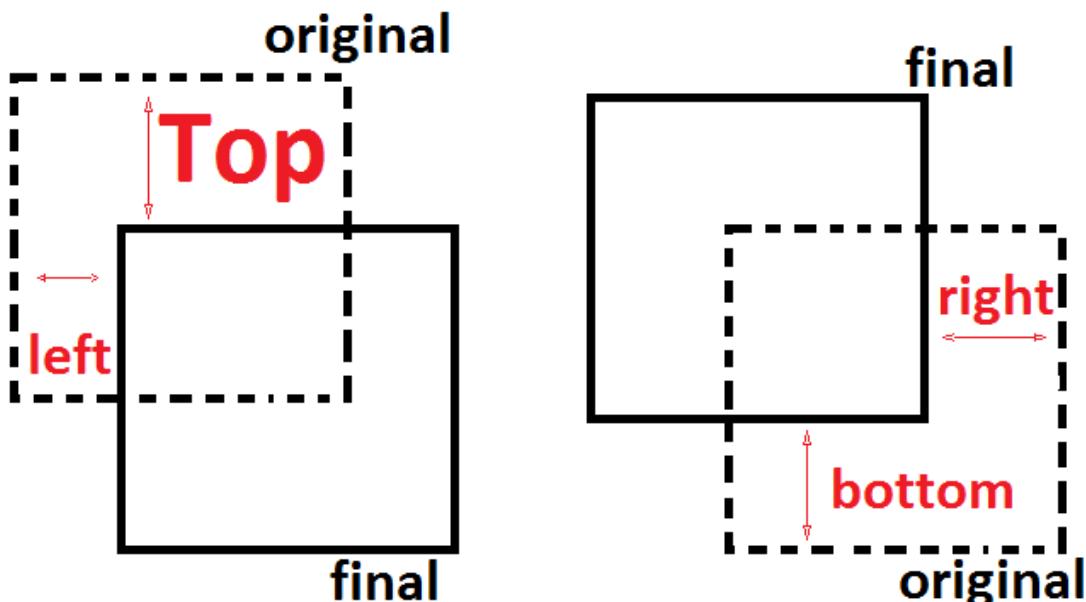
Las cajas desplazadas de forma relativa no modifican su tamaño, por lo que los valores de las propiedades `left` y `right` siempre cumplen que `left = -right`.

Si tanto `left` como `right` tienen un valor de `auto` (que es su valor por defecto) la caja no se mueve de su posición original. Si sólo el valor de `left` es `auto`, su valor real es `-right`. Igualmente, si sólo el valor de `right` es `auto`, su valor real es `-left`.

Si tanto `left` como `right` tienen valores distintos de `auto`, uno de los dos valores se tiene que ignorar porque son mutuamente excluyentes. Para determinar la propiedad que se tiene en cuenta, se considera el valor de la propiedad `direction`.

La propiedad `direction` permite establecer la dirección del texto de un contenido. Si el valor de `direction` es `ltr`, el texto se muestra de izquierda a derecha, que es el método de escritura habitual en la mayoría de países. Si el valor de `direction` es `rtl`, el método de escritura es de derecha a izquierda, como el utilizado por los idiomas árabe y hebreo.

Si el valor de `direction` es `ltr`, y las propiedades `left` y `right` tienen valores distintos de `auto`, se ignora la propiedad `right` y sólo se tiene en cuenta el valor de la propiedad `left`. De la misma forma, si el valor de `direction` es `rtl`, se ignora el valor de `left` y sólo se tiene en cuenta el valor de `right`.



## Ejemplo

```
h2.pos_left
{
position:relative;
left:-20px;
}
h2.pos_right
{
position:relative;
left:20px;
}
```

**This is a heading with no position**

**this heading is moved left according to its normal position**

**This heading is moved right according to its normal position**

Relative positioning moves an element RELATIVE to its original position.

The style "left:-20px" subtracts 20 pixels from the element's original left position.

The style "left:20px" adds 20 pixels to the element's original left position.

```
<!DOCTYPE html>
<html>
<head>
<style>
h2.pos_left
{
background: #c0c0c0;
position: relative;
left: -20px;
}

h2.pos_right
{
background: #c0c0c0;
position: relative;
left: 20px;
}
</style>
</head>

<body>
<h2>Encabezado sin posicionamiento</h2>
<h2 class="pos_left">Este encabezado ha sido desplazado hacia la derecha respecto a su posición normal</h2>
<h2 class="pos_right">Este encabezado ha sido desplazado hacia la izquierda respecto a su posición normal</h2>
<p>El posicionamiento relativo mueve un elemento de forma relativa a su posición original.</p>
<p>"left: -20px" resta 20 pixels a la posición normal de la izquierda del elemento.</p>
<p>"left: 20px" suma 20 pixels a la posición normal de la izquierda del elemento.</p>
</body>

</html>
```

<https://photoshop-kopona.com/es/56791-psd-24-png-perros-divertidos-en-fondo-transparente.html>



```
<!DOCTYPE html>
```

```
<html>
<head>
<style>
#perrete
{
position: relative;
top: 55px;
}
h1{
```

```

margin-left:1px;

}

</style>

</head>

<body>



<h1>Los Perretes</h1>

</body>

</html>

```

El contenido de elementos en posición relativa puede ser movido y se puede superponer a otros elementos, pero el espacio reservado para el elemento se conserva aún en el flujo normal.

### Ejemplo

```

h2.pos_top
{
    position:relative;
    top:-50px;
}

```

**This heading with no position according to its normal position**

**Note:** Even if the content of the relatively positioned element is moved, the reserved space for the element is still preserved in the normal flow.

El elemento sigue "reservando/ocupando" el espacio que ocuparía en su posición "normal"

```

<!DOCTYPE html>
<html>
<head>
<style>
h2.pos_top
{
    position:relative;
    top:-50px;
}
</style>
</head>

<body>
<h2>This is a heading with no position</h2>
<h2 class="pos_top">This heading is moved upwards according to its normal position</h2>
<p><strong>Note:</strong> Even if the content of the relatively positioned element is moved, the reserved space for the element is still preserved in the normal flow.</p>
</body>

</html>

```

Los elementos en posición relativa a menudo se utilizan como bloques contenedor para los elementos con posición absoluta, como veremos a continuación.

El siguiente ejemplo muestra tres imágenes posicionadas de forma normal:

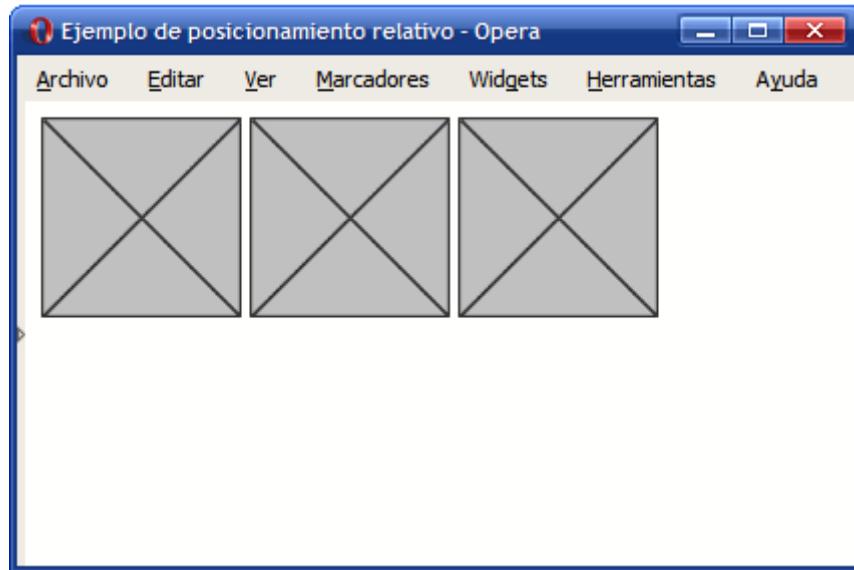


Figura 5.5 Elementos posicionados de forma normal

Aplicando el posicionamiento relativo, se desplaza la primera imagen de forma descendente:

```
img.desplazada {  
    position: relative;  
    top: 8em;  
}
```

```
  
  

```

El aspecto que muestran ahora las imágenes es el siguiente:

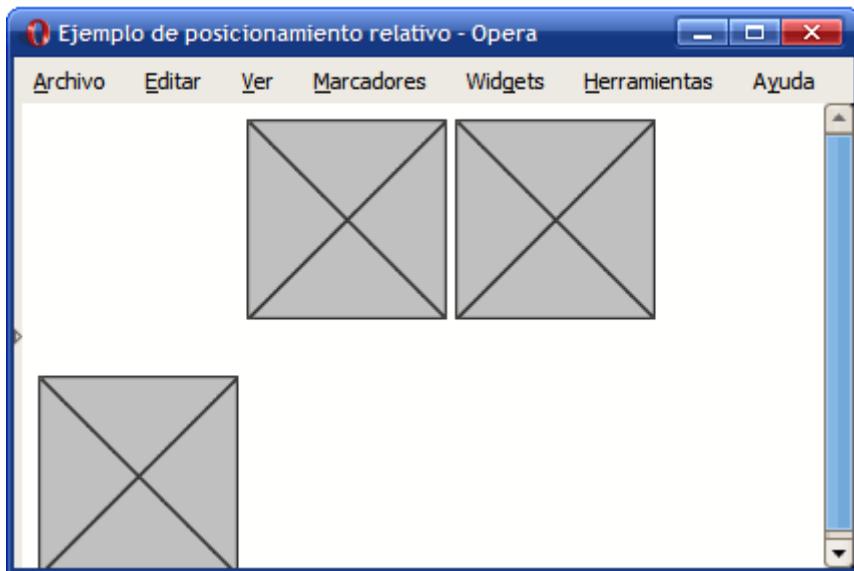


Figura 5.6 Elemento posicionado de forma relativa

El resto de imágenes no varían su posición y por tanto no ocupan el hueco dejado por la primera imagen, ya que el posicionamiento relativo no influye en el resto de elementos de la página. El principal problema de posicionar elementos de forma relativa es que se pueden producir solapamientos con otros elementos de la página.

## Posicionamiento absoluto

Un elemento con posición absoluta está situado con posición relativa al primer elemento padre que tenga una posición que no sea estática. Si no se encuentra dicho elemento, el bloque de contención es <html>:

El posicionamiento absoluto se emplea para establecer de forma exacta la posición en la que se muestra la caja de un elemento. La nueva posición de la caja se indica mediante las propiedades top, right, bottom y left. La interpretación de los valores de estas propiedades es mucho más compleja que en el posicionamiento relativo, ya que en este caso dependen del posicionamiento del elemento contenedor.

Cuando una caja se posiciona de forma absoluta, el resto de elementos de la página se ven afectados y modifican su posición. Al igual que en el posicionamiento relativo, cuando se posiciona de forma absoluta una caja es probable que se produzcan solapamientos con otras cajas.

En el siguiente ejemplo, se posiciona de forma absoluta la caja 2:



Figura 5.7 Ejemplo de posicionamiento absoluto de un elemento

La caja 2 está posicionada de forma absoluta, lo que provoca que el resto de elementos de la página modifiquen su posición. En concreto, la caja 3 deja su lugar original y pasa a ocupar el hueco dejado por la caja 2.

El estándar de CSS 2.1 indica que las cajas posicionadas de forma absoluta "salen del flujo normal de la página", lo que provoca que el resto de elementos de la página se muevan y en ocasiones, ocupen la posición original en la que se encontraba la caja.

Por otra parte, el desplazamiento de una caja posicionada de forma absoluta se controla mediante las propiedades top, right, bottom y left. A diferencia del posicionamiento relativo, la interpretación de los valores de estas propiedades depende del elemento contenedor de la caja posicionada.

Determinar la referencia utilizada para interpretar los valores de top, right, bottom y left de una caja posicionada de forma absoluta es un proceso complejo que se compone de los siguientes pasos:

- Se buscan todos los elementos contenedores de la caja hasta llegar al elemento <body> de la página.
- Se recorren todos los elementos contenedores empezando por el más cercano a la caja y llegando hasta el <body>
- El primer elemento contenedor que esté posicionado de cualquier forma diferente a position: static se convierte en la referencia que determina la posición de la caja posicionada de forma absoluta.
- Si ningún elemento contenedor está posicionado, la referencia es la ventana del navegador, que no debe confundirse con el elemento <body> de la página.

Una vez determinada la referencia del posicionamiento absoluto, la interpretación de los valores de las propiedades top, right, bottom y left se realiza como sigue:

- El valor de la propiedad top indica el desplazamiento desde el borde superior de la caja hasta el borde superior del elemento contenedor que se utiliza como referencia.
- El valor de la propiedad right indica el desplazamiento desde el borde derecho de la caja hasta el borde derecho del elemento contenedor que se utiliza como referencia.
- El valor de la propiedad bottom indica el desplazamiento desde el borde inferior de la caja hasta el borde inferior del elemento contenedor que se utiliza como referencia.
- El valor de la propiedad left indica el desplazamiento desde el borde izquierdo de la caja hasta el borde izquierdo del elemento contenedor que se utiliza como referencia.

## Ejemplo

```
h2
{
position: absolute;
left: 100px;
top: 150px;
}
```

With absolute positioning, an element can be placed anywhere on a page. The heading below is placed 100px from the left of the page and 150px from the top of the page.

## This is a heading with an absolute position

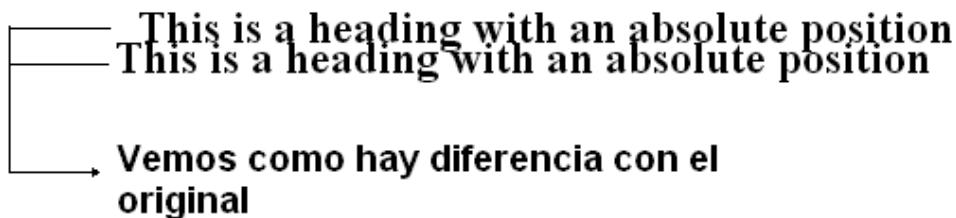
```
<!DOCTYPE html>
<html>
<head>
<style>
h2
{
    position: absolute;
    left: 100px;
    top: 150px;
}
</style>
</head>

<body>
<h2>This is a heading with an absolute position</h2>
<p>With absolute positioning, an element can be placed anywhere on a page. The heading below is placed 100px from the left of the page and 150px from the top of the page.</p>
</body>

</html>
```

Si lo metemos dentro de un div con posicionamiento relativo, vemos que el bloque h2 se colocará en posición relativa a dicho div

With absolute positioning, an element can be placed anywhere on a page. The heading below is placed 100px from the left of the page and 150px from the top of the page.



```
<!DOCTYPE html>
<html>
<head>
<style>
h2
{
    position: absolute;
    left: 100px;
    top: 150px;
}
div
{
    position: relative;
    left: -20px;
}
</style>
</head>

<body>
<div><h2>This is a heading with an absolute position</h2></div>
<p>With absolute positioning, an element can be placed anywhere on a page. The heading below is placed 100px from the left of the page and 150px from the top of the page.</p>
</body>

</html>
```

Los elementos con posición absoluta se eliminan del flujo normal. El documento y otros elementos se comportan como si el elemento con posición absoluta no existiera.

Los elementos con posición absoluta pueden superponer a otros elementos.

En los siguientes ejemplos, se utiliza la página HTML que muestra la siguiente imagen:

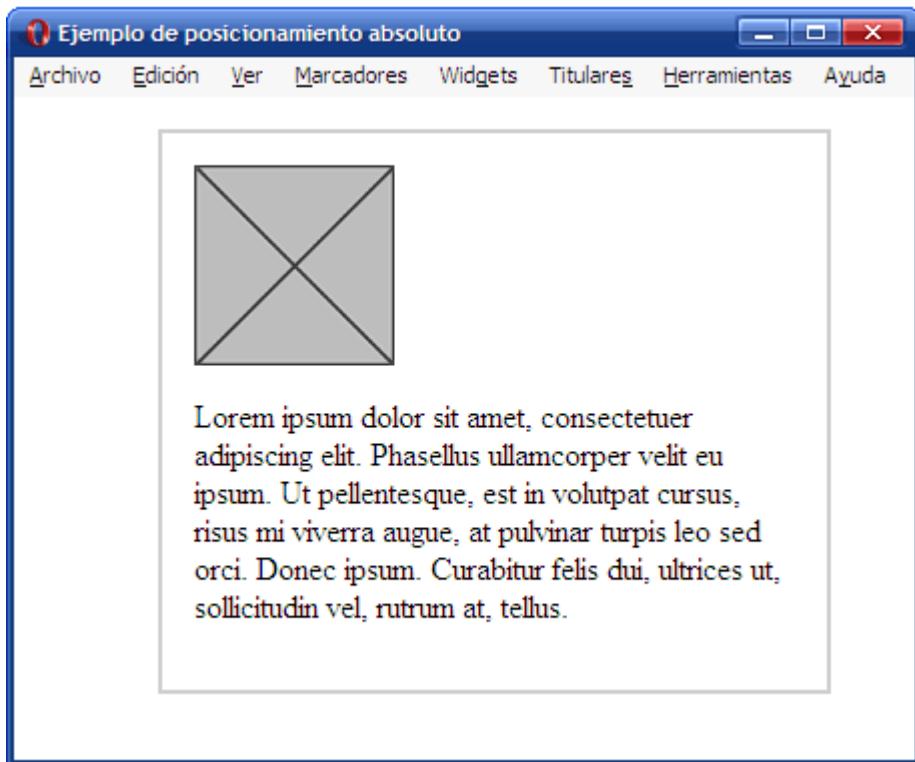


Figura 5.8 Situación original antes de modificar el posicionamiento

A continuación, se muestra el código HTML y CSS de la página original:

```
div {  
    border: 2px solid #CCC;  
    padding: 1em;  
    margin: 1em 0 1em 4em;  
    width: 300px;  
}  
  
<div>  
      
    <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Phasellus  
        ullamcorper velit eu ipsum. Ut pellentesque, est in volutpat cursus, risus  
        mi viverra augue, at pulvinar turpis leo sed orci. Donec ipsum. Curabitur  
        felis dui, ultrices ut, sollicitudin vel, rutrum at, tellus.</p>  
</div>
```

En primer lugar, se posiciona de forma absoluta la imagen mediante la propiedad **position** y se indica su nueva posición mediante las propiedades **top** y **left**:

```
div img {
```

```
position: absolute;  
top: 50px;  
left: 50px;  
}
```

El resultado visual se muestra en la siguiente imagen:

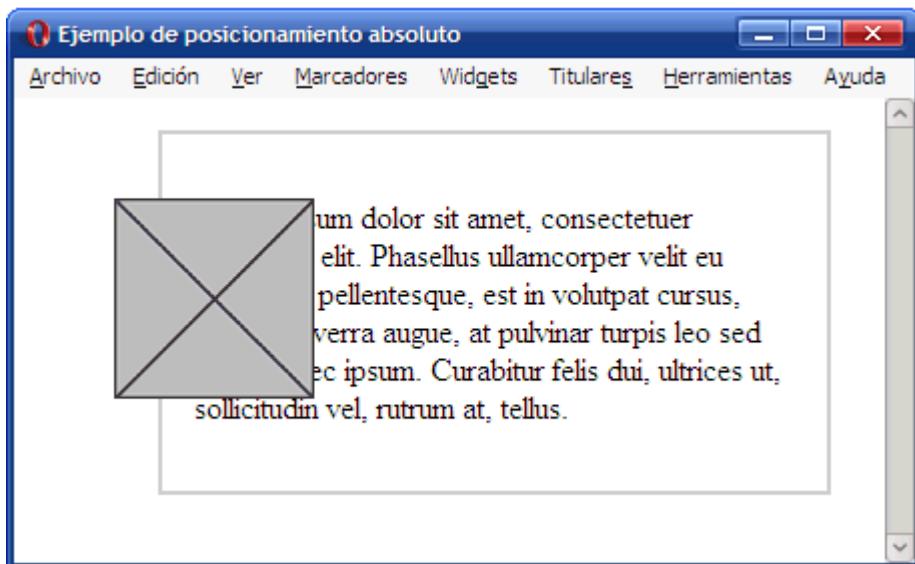


Figura 5.9 Imagen posicionada de forma absoluta

La imagen posicionada de forma absoluta no toma como referencia su elemento contenedor <div>, sino la ventana del navegador, tal y como demuestra la siguiente imagen:

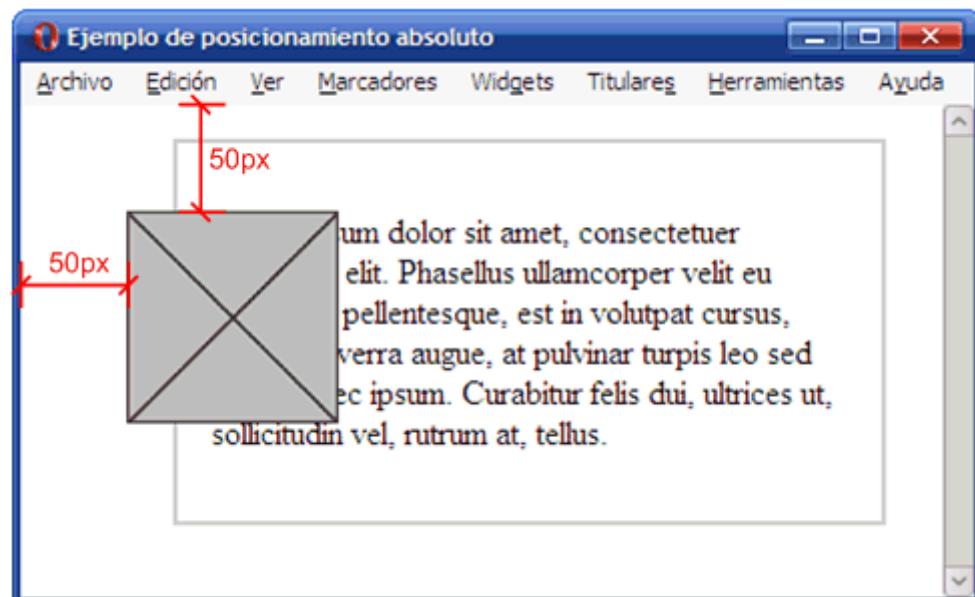


Figura 5.10 La referencia del posicionamiento absoluto es la ventana del navegador

Para posicionar la imagen de forma absoluta, el navegador realiza los siguientes pasos:

1. Obtiene la lista de elementos contenedores de la imagen: <div> y <body>.

2. Recorre la lista de elementos contenedores desde el más cercano a la imagen (el <div>) hasta terminar en el <body> buscando el primer elemento contenedor que esté posicionado.
3. El posicionamiento de todos los elementos contenedores es el normal o estático, ya que ni siquiera tienen establecida la propiedad **position**
4. Como ningún elemento contenedor está posicionado, la referencia es la ventana del navegador.
5. A partir de esa referencia, la caja de la imagen se desplaza 50px hacia la derecha (**left: 50px**) y otros 50px de forma descendente (**top: 50px**).

Como la imagen se posiciona de forma absoluta, el resto de elementos de la página se mueven para ocupar el lugar libre dejado por la imagen. Por este motivo, el párrafo sube hasta el principio del <div> y se produce un solapamiento con la imagen posicionada que impide ver parte de los contenidos del párrafo.

A continuación, se modifica el ejemplo anterior posicionando de forma relativa el elemento <div> que contiene la imagen y el párrafo. La única propiedad añadida al <div> es **position: relative** por lo que el elemento contenedor se posiciona pero no se desplaza respecto de su posición original:

```
div {  
border: 2px solid#CCC;  
padding: 1em;  
margin: 1em 0 1em 4em;  
width: 300px;  
position: relative;  
}  
  
div img {
```

```
position: absolute;  
top: 50px;  
left: 50px;  
}
```

La siguiente imagen muestra el resultado obtenido:

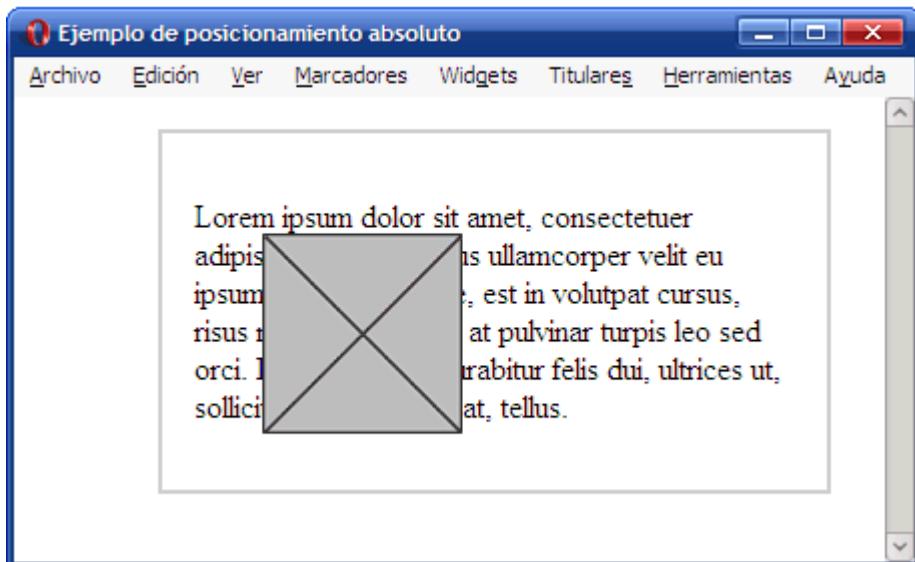


Figura 5.11 Imagen posicionada de forma absoluta

En este caso, como el elemento contenedor de la imagen está posicionado, se convierte en la referencia para el posicionamiento absoluto. El resultado es que la posición de la imagen es muy diferente a la del ejemplo anterior:

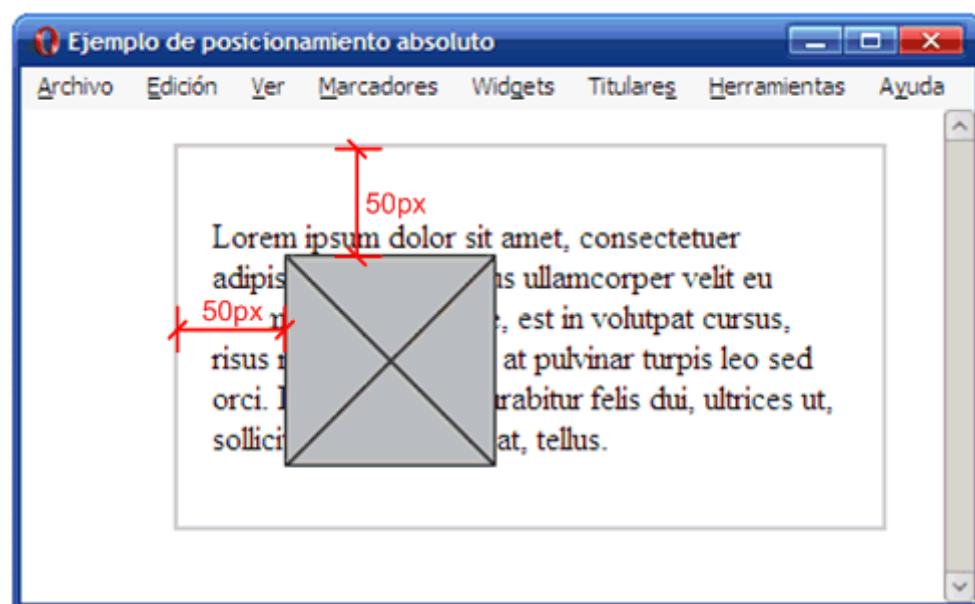


Figura 5.12 La referencia del posicionamiento absoluto es el elemento contenedor de la imagen

Por tanto, si se quiere posicionar un elemento de forma absoluta respecto de su elemento contenedor, es imprescindible posicionar este último. Para ello, sólo es necesario añadir la propiedad `position: relative`, por lo que no es obligatorio desplazar el elemento contenedor respecto de su posición original.

Ejemplo menus:

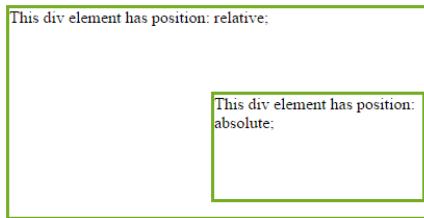
<http://www.tyssendesign.com.au/shed/examples/IR-navbar.html#example1>

<https://escss.blogspot.com/2013/04/drop-down-css.html>

Ejemplo

## position: absolute;

An element with position: absolute; is positioned relative to the nearest positioned ancestor (instead of positioned relative to the viewport, like fixed):



```
<!DOCTYPE html>
<html>
<head>
<style>
div.relativo {
    position: relative;
    width: 400px;
    height: 200px;
    border: 3px solid #73AD21;
}

div.absoluto {
    position: absolute;
    top: 80px;
    right: 0;
    width: 200px;
    height: 100px;
    border: 3px solid #73AD21;
}
</style>
</head>
<body>

<h2>posicion absoluta;</h2>

<p>Un elemento con posicionamiento: absoluto; se posiciona en relación con el ancestro posicionado más cercano (en lugar de posicionado en relación con la ventana gráfica, como con fixed):</p>

<div class="relativo">Este div tiene posicionamiento relative;
<div class="absoluto"> Este div tiene posicionamiento: absolute;</div>
</div>

</body>
</html>
En http://www.csszengarden.com/216/ se usa el posicionamiento absoluto para poner una imagen de fondo a un div.
```

Ejemplo mas chulo

Incialmente

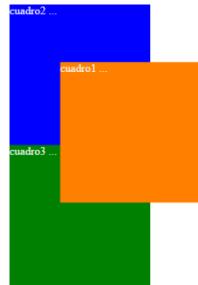


El atributo lang es usado para declarar el idioma de la página web para ayudar a los buscadores y a los navegadores

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>posicionamiento</title>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
<!--
pondríamos en su lugar:
<meta charset="UTF-8"/>
-->
<style type="text/css">
.cuadro1 {
    height:200px; width:200px;
    background-color:#FF8000;
    color:#ffffff;
}
.cuadro2{
    height:200px; width:200px;
    background-color:#0000FF;
    color:#ffffff;
}
.cuadro3{
    height:200px; width:200px;
    background-color:#008000;
    color:#ffffff;
}
</style>
</head>
<body>
<div class="cuadro1">cuadro1 ... </div>
```

```
<div class="cuadro2">cuadro2 ... </div>
<div class="cuadro3">cuadro3 ... </div>
</body>
</html>
```

Con posicionamiento absoluto



```
<!DOCTYPE html >

<html lang="es" >

<head>

<title>posicionamiento absoluto</title>

<meta http-equiv="Content-Type" content="text/html;
charset=iso-8859-1" />

<style type="text/css">

.cuadro1 {height:200px;width:200px;

background-color:#FF8000;

color: #ffffff;

position:absolute;

top:90px;

left:80px;

}

.cuadro2 {height:200px;width:200px;

background-color: #0000FF;

color: #ffffff; }

.cuadro3 {height:200px;width:200px;

background-color: #008000;
```

```
color: #ffffff; }

</style>

</head>

<body>

<div class="cuadro1">cuadro1 ... </div>

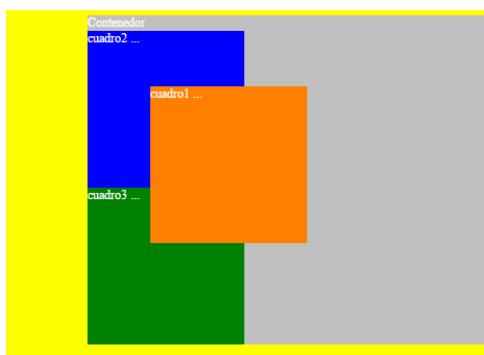
<div class="cuadro2">cuadro2 ... </div>

<div class="cuadro3">cuadro3 ... </div>

</body>

</html>
```

Con contenedor relativo



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml" lang="es" xml:lang="es">

<head>

<title>posicionamiento absoluto</title>

<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />

<style type="text/css">

.cuadro1 { height: 200px; width: 200px;
background-color: #FF8000;
color: #ffffff; position: absolute; top: 90px; left: 80px; }

.cuadro2 { height: 200px; width: 200px;
background-color: #0000FF;
```

```
color: #ffffff; }

.cuadro3 { height: 200px; width: 200px;

background-color: #008000;

color: #ffffff; }

.contenedor {

position: relative;

left: 100px;

background-color: #c0c0c0;

color: #ffffff;

}

html{

background-color: yellow;

}

</style>

</head>

<body>

<div class="contenedor">

Contenedor

<div class="cuadro1">cuadro1 ... </div>

<div class="cuadro2">cuadro2 ... </div>

<div class="cuadro3">cuadro3 ... </div>

</div>

</body>

</html>
```

#### Posicionamiento sticky(pegajoso)

Sticky es un nuevo valor de la propiedad position. Funciona de manera similar al posicionamiento relativo, en cuanto que no elimina ningún elemento del flujo del documento. En otras palabras, un elemento "sticky" no afecta a la posición de sus elementos adyacentes, y no hunde su elemento padre.

Básicamente hace que el elemento se comporte como en position: relative hasta que, debido al scroll de la página, el elemento es alcanzado por la parte superior del viewport (no es exactamente así, pero es el uso más habitual). En ese momento se comporta como fixed, y para que funcione correctamente hay que especificar al menos un valor top, bottom, left o right, en función de si va a ser sticky en un scroll horizontal o vertical.



Soporte de los navegadores de la posición CSS Sticky

Tomemos el siguiente ejemplo, establecemos la posición del #sidebar a sticky y al mismo tiempo top: 10px. La propiedad top es requerida y especifica la distancia desde el borde superior del área de visualización de la página.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.sticky {
    position: -webkit-sticky;
    position: sticky;
    top: 0;
    padding: 5px;
    background-color: #cae8ca;
    border: 2px solid #4caf50;
}
</style>
</head>
<body>

<p>Intenta hacer <strong>scroll</strong> dentro de este marco para entender cómo funciona el posicionamiento pegajoso.</p>
<p>Nota: En IE/Edge 15 y anteriores no funciona.</p>

<div class="sticky">Elemento pegajoso!</div>

<div style="padding-bottom: 2000px">
<p>En este ejemplo, el elemento pegajoso se adhiere a la parte superior de la página (arriba: 0), cuando llegue a su posición de desplazamiento.</p>
<p>Parte del texto para permitir el desplazamiento... Lorem ipsum dolor sit amet, illum definitiones no quo, maluisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.</p>
<p>Some text to enable scrolling.. Lorem ipsum dolor sit amet, illum definitiones no quo, maluisset concludaturque et eum, altera fabulas ut quo. Atqui causae gloriatur ius te, id agam omnis evertitur eum. Affert laboramus repudiandae nec et. Inciderint efficiantur his ad. Eum no molestiae voluptatibus.</p>
</div>

</body>
</html>
```

Ejemplo de blog con foto de perfil sticky

```
<p>Hola a todos</p>
</div>
</div>
```

```
Hola
</body>
</html>
```

**Ejemplo menu sticky**

<https://css-tianguis.com/position-sticky/>

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    margin: 0;
}

.header {
    height: 100px;
    background-color: #ddcbaf;
    text-align: center;
}

.header h1 {
    margin-top: 0;
    padding-top: 20px;
}

.menu {
    margin: 0;
    padding: 0;
    width: 100%;
    background-color: #BFFF3;
    top: 0;
    position: -webkit-sticky;
    position: sticky;
    opacity: 0.8;
}

.menu li {
    display: inline-block;
    text-align: center;
    width: 25%;
}

.menu li:hover,
.menu li:focus {
    background-color: #66FFE3;
}

.menu a {
    display: block;
    padding: 10px 0;
    text-decoration: none;
}

.container {
    padding: 0 20px;
    color: #989898;
}
```

```
</style>
<body>
<div class="header">
<h1>Esta es mi bonita empresa!</h1>
</div>
<ul class="menu">
<li><a href="#">Inicio</a></li><li>
<!--lo ponemos en la misma línea porque si no los porcentajes exactos no funcionan-->
<a href="#">Servicios</a></li><li>
<a href="#">Portafolio</a></li><li>
<a href="#">Contacto</a></li>
</ul>
<div class="container">
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent luctus urna sit amet sollicitudin venenatis. Aenean odio tortor, varius vitae molestie eu, ultricies vel lacus. Nam viverra fermentum dapibus. Nulla at semper diam. Phasellus sit amet hendrerit sapien, non semper felis. Morbi augue leo, mattis nec leo sed, malesuada porta dui. Maecenas pretium eros quis lorem luctus gravida.</p>
<p>Sed gravida quam odio, euismod accumsan mauris ornare quis. Duis sed condimentum justo. Phasellus ac dui eget velit bibendum viverra. Aenean porttitor commodo diam, quis interdum mi sagittis lobortis. Donec id ipsum dignissim, pharetra nunc sit amet, porttitor elit. Donec iaculis elit et enim interdum, ac laoreet lorem consequat. Sed eu elit ut quam pellentesque cursus. Integer sed condimentum est. Integer tempor placerat bibendum. Sed ut semper arcu, at porttitor nibh. Fusce vulputate pharetra tellus a laoreet. Cras imperdiet enim sed turpis adipiscing placerat. Vestibulum ut rhoncus mauris. In egestas ullamcorper dolor vitae suscipit. Curabitur non orci rutrum, iaculis ligula quis, sollicitudin neque. Aliquam dapibus dignissim tincidunt.</p>
<p>Suspendisse at urna mauris. Vivamus eu lectus et quam viverra accumsan quis nec ligula. Aliquam sed mi sit amet arcu convallis bibendum id eu lorem. Sed pretium eget nibh egestas consectetur. Sed adipiscing, libero sed molestie laoreet, arcu tortor elementum ligula, nec commodo ipsum augue ut tellus. Morbi nibh mauris, facilisis vel bibendum vehicula, dignissim in tortor. Suspendisse augue urna, vestibulum at orci nec, scelerisque cursus mi. Proin congue eget justo et mattis.</p>
<p>Suspendisse suscipit in lectus at aliquet. Integer non sem enim. Vestibulum aliquam imperdiet laoreet. In malesuada sodales augue, ut aliquam elit tempus id. Suspendisse sed hendrerit nibh. Curabitur molestie in libero nec vulputate. Cras ut scelerisque lacus, vitae cursus dui. Fusce ultricies lectus tincidunt, congue elit interdum, pellentesque nunc.</p>
<p>Donec quis diam dapibus, accumsan ipsum vitae, porta turpis. Pellentesque vel ligula suscipit, scelerisque justo sit amet, vestibulum enim. Morbi eget sapien blandit, mattis enim sed, tincidunt eros. Nullam urna lectus, pretium id ultrices non, convallis vitae arcu. Fusce vel justo in nulla pulvinar interdum id sed sem. Ut quis ullamcorper dolor, a bibendum quam. Vivamus pellentesque, enim feugiat tempus venenatis, arcu quam molestie eros, quis bibendum nulla metus vitae nisi. Integer at lectus at metus eleifend molestie. Nulla sit amet pharetra est. Sed commodo ac leo ac volutpat. Vivamus enim sapien, sodales vel cursus sit amet, ornare at risus. Proin a lacus id felis aliquet fringilla.</p>
<p>Donec quis diam dapibus, accumsan ipsum vitae, porta turpis. Pellentesque vel ligula suscipit, scelerisque justo sit amet, vestibulum enim. Morbi eget sapien blandit, mattis enim sed, tincidunt eros. Nullam urna lectus, pretium id ultrices non, convallis vitae arcu. Fusce vel justo in nulla pulvinar interdum id sed sem. Ut quis ullamcorper dolor, a bibendum quam. Vivamus pellentesque, enim feugiat tempus venenatis, arcu quam molestie eros, quis bibendum nulla metus vitae nisi. Integer at lectus at metus eleifend molestie. Nulla sit amet pharetra est. Sed commodo ac leo ac volutpat. Vivamus enim sapien, sodales vel cursus sit amet, ornare at risus. Proin a lacus id felis aliquet fringilla.</p>
<p>Donec quis diam dapibus, accumsan ipsum vitae, porta turpis. Pellentesque vel ligula suscipit, scelerisque justo sit amet, vestibulum enim. Morbi eget sapien blandit, mattis enim sed, tincidunt eros. Nullam urna lectus, pretium id ultrices non, convallis vitae arcu. Fusce vel justo in nulla pulvinar interdum id sed sem. Ut quis ullamcorper dolor, a bibendum quam. Vivamus pellentesque, enim feugiat tempus venenatis, arcu quam molestie eros, quis bibendum nulla metus vitae nisi. Integer at lectus at metus eleifend molestie. Nulla sit amet pharetra est. Sed commodo ac leo ac volutpat. Vivamus enim sapien, sodales vel cursus sit amet, ornare at risus. Proin a lacus id felis aliquet fringilla.</p>
```

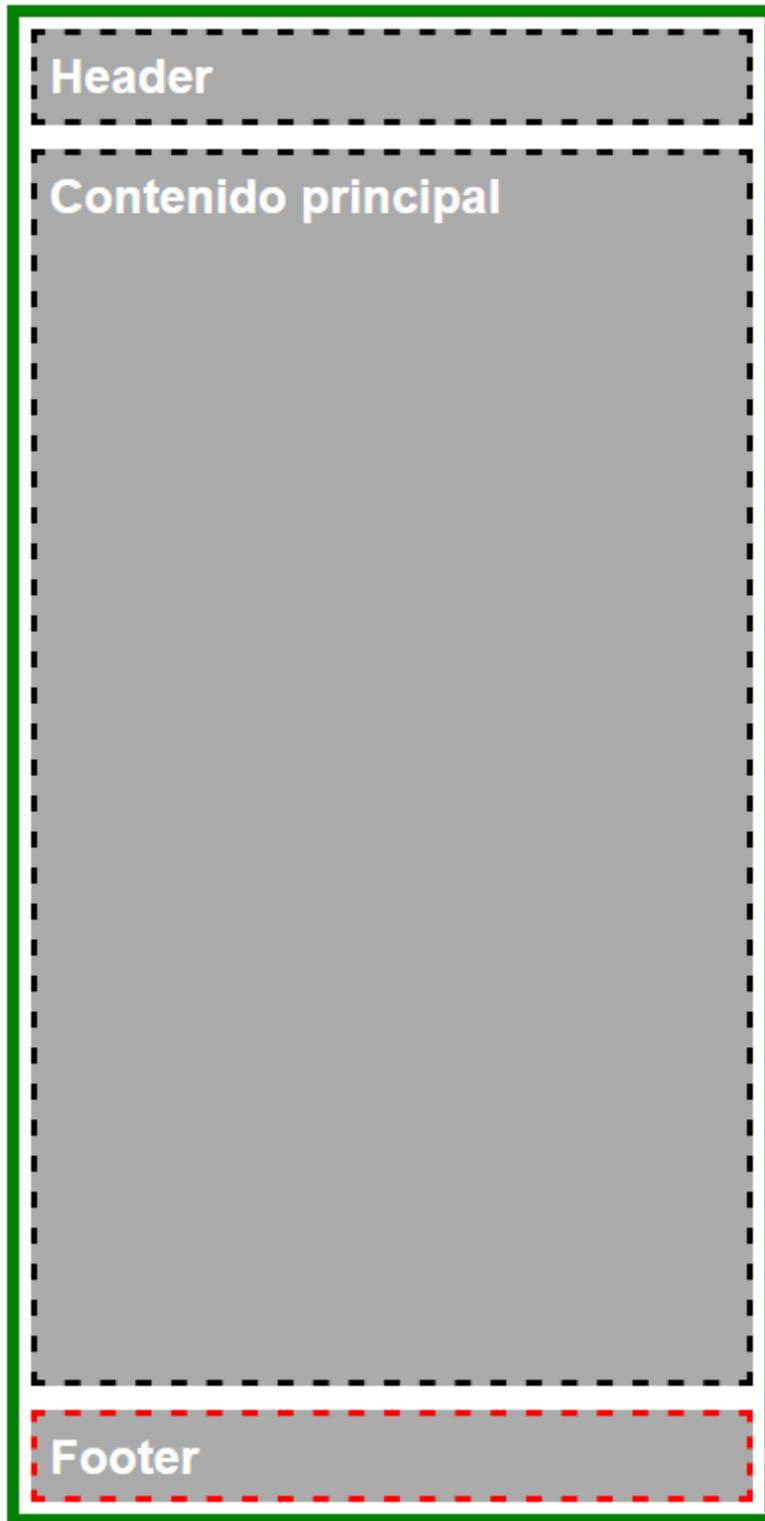


<p>Donec quis diam dapibus, accumsan ipsum vitae, porta turpis. Pellentesque vel ligula suscipit, scelerisque justo sit amet, vestibulum enim. Morbi eget sapien blandit, mattis enim sed, tincidunt eros. Nullam urna lectus, pretium id ultrices non, convallis vitae arcu. Fusce vel justo in nulla pulvinar interdum id sed sem. Ut quis ullamcorper dolor, a bibendum quam. Vivamus pellentesque, enim feugiat tempus venenatis, arcu quam molestie eros, quis bibendum nulla metus vitae nisi. Integer at lectus at metus eleifend molestie. Nulla sit amet pharetra est. Sed commodo ac leo ac volutpat. Vivamus enim sapien, sodales vel cursus sit amet, ornare at risus. Proin a lacus id felis aliquet fringilla.</p>

<p>Donec quis diam dapibus, accumsan ipsum vitae, porta turpis. Pellentesque vel ligula suscipit, scelerisque justo sit amet, vestibulum enim. Morbi eget sapien blandit, mattis enim sed, tincidunt eros. Nullam urna lectus, pretium id ultrices non, convallis vitae arcu. Fusce vel justo in nulla pulvinar interdum id sed sem. Ut quis ullamcorper dolor, a bibendum quam. Vivamus pellentesque, enim feugiat tempus venenatis, arcu quam molestie eros, quis bibendum nulla metus vitae nisi. Integer at lectus at metus eleifend molestie. Nulla sit amet pharetra est. Sed commodo ac leo ac volutpat. Vivamus enim sapien, sodales vel cursus sit amet, ornare at risus. Proin a lacus id felis aliquet fringilla.</p>

```
</div>
</body>
</html>
```

Ejemplo con pie y cabecera fijas



```
<!DOCTYPE html>
<html>
<head>
<style>
body{
    color: #fff;
    font-family: arial;
    font-weight: bold;
    font-size: 40px;
```

```

}

.main-container{
    max-width:600px;
    margin:0 auto;
    border:solid 10px green;
    padding:10px; margin-top:40px;
}
.main-container *{
    padding:10px;
    background:#aaa;
    border:dashed 5px #000;
}

/*
Ejemplo: La regla h2 + p se aplicará a todos los elementos p que siguen directamente a un elemento h2.


#### Ejemplo sticky múltiple animales



```

<!DOCTYPE html>
<html>
<head>
<style>
.heading{
    background: #ccc;
    height: 50px;
    line-height: 50px;

```



Página 218 de 502


```

```
margin-top: 10px;  
font-size: 30px;  
padding-left: 10px;  
position: -webkit-sticky;  
position: sticky;  
top: 0px;  
}  
  
</style>  
<body>  
<h1>Animales por orden alfabético</h1>  
  
<div class="container">  
<div class="heading">A</div>  
<div>Abubilla</div>  
<div>Águila</div>  
<div>Aligator</div>  
<div>Antílope</div>  
  
<div class="heading">B</div>  
<div>Bacalao</div>  
<div>Ballena</div>  
<div>Boa</div>  
  
<div class="heading">C</div>  
<div>Caballo</div>  
<div>Camello</div>  
<div>Cebrera</div>  
<div>Chivo</div>  
  
<div class="heading">D</div>  
<div>Delfín</div>  
<div>Dinosaurio</div>  
<div>Dragón</div>  
<div>Dodo</div>  
</div>  
  
</body>  
</html>
```

Otro ejemplo

<https://css-tricks.com/creating-sliding-effects-using-sticky-positioning/>  
<https://webdesign.tutsplus.com/es/tutorials/sticky-positioning-with-nothing-but-css--cms-24042>

Otro ejemplo

<https://codepen.io/Kseso/pen/ofAkg>

Sticky multiple con un poco de javascript para aquellos navegadores que no lo soportan  
([stickyfill.js](#) → [stickyfill.min.js](#))

<http://jsfiddle.net/psflannery/t2n3qkvh/>  
<http://meridadesignblog.com/stickyfill-polyfill-javascript-aplicar-posicion-pegajosa-cualquier-elemento/>

## Elementos superpuestos

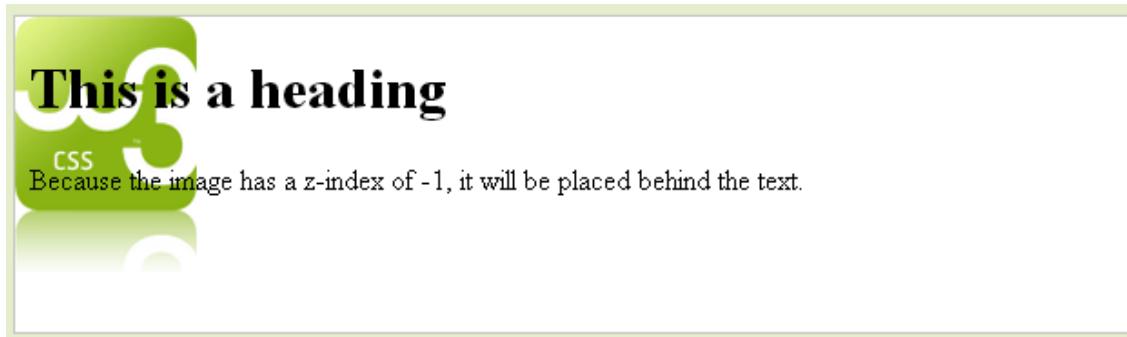
Cuando los elementos están posicionados fuera del flujo normal, pueden superponerse con otros elementos.

La propiedad z-index especifica el orden de apilamiento de un elemento (el elemento que debe ser colocado delante o detrás de los otros).

Un elemento puede tener un orden de apilamiento positivo o negativo:

## Ejemplo

```
img
{
position:absolute;
left:0px;
top:0px;
z-index:-1;
}
```



```
<!DOCTYPE html>
<html>
<head>
<style>
img
{
    position:absolute;
    left:0px;
    top:0px;
    z-index:-1;
}
</style>
</head>

<body>
<h1>Esto es una cabecera</h1>

<p>Debido a que la imagen tiene un z-index = -1, se colocará detrás del texto.</p>
</body>
</html>
```

Un elemento con mayor valor en el eje Z siempre estará delante de un elemento con un valor en el eje Z mas bajo.

**Nota:** Si dos elementos posicionados se superponen sin especificar un z-index, el último elemento posicionado en el código HTML se mostrará en la parte superior.

Variando un poquito el ejemplo

# This is a heading

Because the image has a z-index of -1, it will be placed behind the text.



```
<!DOCTYPE html>
<html>
<head>
<style>
div
{
position: relative;
left: -20px;
}

img
{
position: absolute;
left: 0px;
top: 0px;
z-index: -1;
}
</style>
</head>

<body>
<h1>This is a heading</h1>
<div></div>
<p>Because the image has a z-index of -1, it will be placed behind the text.</p>
</body>
</html>
```

#### Completar este apartado Rejilla

<https://css-tricks.com/snippets/css/complete-guide-grid/>

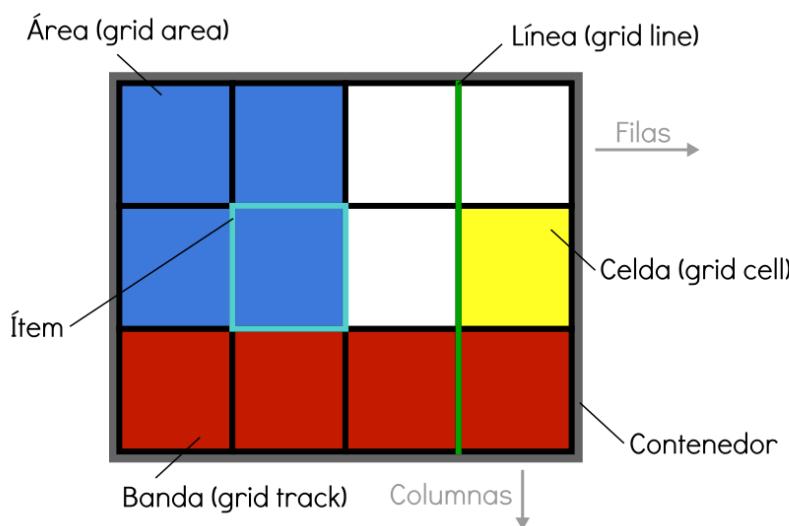
El Layout Grid es el sistema de diseño más potente disponible en CSS. Es un sistema bidimensional, lo que significa que puede manejar tanto columnas como filas, a diferencia de flexbox, que es en gran medida un sistema unidimensional. Trabaja con el Diseño de cuadrícula aplicando reglas CSS tanto a un elemento padre (que se convierte en el Contenedor de la cuadrícula/grid) como a los elementos secundarios de ese elemento (Elementos de la cuadrícula/grid).

El Layout Grid (también conocido simplemente como "Grid"), es un sistema de diseño bidimensional basado en cuadrículas que tiene como objetivo cambiar por completo la forma en que diseñamos interfaces de usuario basadas en rejillas. CSS siempre se ha utilizado para diseñar nuestras páginas web, pero nunca ha hecho un buen trabajo. Primero, usamos tablas, luego floats, posicionamiento y elementos inline-block, pero todos estos métodos eran esencialmente hacks y dejaban de lado una gran cantidad de funcionalidades importantes (centrado vertical, por ejemplo). Flexbox ayudó, pero está destinado a diseños unidimensionales más simples, no a diseños bidimensionales complejos (Flexbox y Grid realmente funcionan muy bien juntos). Grid es el primer módulo CSS creado específicamente para resolver los problemas de diseño que todos hemos estado "hackeando" durante todo el tiempo que hemos estado creando sitios web.

Para comenzar, debe definir un elemento contenedor como display: grid, establecer los tamaños de columna y fila con grid-template-columns y grid-template-rows, después deberemos colocar los elementos secundarios en la cuadrícula con grid-column y grid-row. De manera similar a flexbox, el orden de origen de los elementos de la cuadrícula no importa. Su CSS puede colocarlos en cualquier orden, lo que hace que sea muy fácil reorganizar su cuadrícula con media queries. Imagina que defines el diseño de toda su página y luego lo

reorganizas por completo para acomodar un ancho de pantalla diferente, todo con solo un par de líneas de CSS. Grid es uno de los módulos CSS más potentes jamás presentados.

- Contenedor: Existe un elemento padre que es el contenedor que definirá la cuadrícula o rejilla.
- Ítem: Cada uno de los hijos que contiene la cuadrícula (elemento contenedor).
- Celda (grid cell): Cada uno de los cuadritos (unidad mínima) de la cuadrícula.
- Área (grid area): Región o conjunto de celdas de la cuadrícula.
- Banda (grid track): Banda horizontal o vertical de celdas de la cuadrícula.
- Línea (grid line): Separador horizontal o vertical de las celdas de la cuadrícula.



Un diseño de cuadrícula consta de un elemento primario, con uno o más elementos secundarios.

La propiedad CSS **grid-template-columns** define el nombre de las líneas y las funciones de tamaño de línea de [grid columns](#).

Define las columnas del contenedor. Puedes especificar el ancho de una columna utilizando o bien auto o una longitud (como 10px). El número de columnas está determinado por el número de valores definidos en la lista separada por espacios.

```
grid-template-columns: none;
```



```
grid-template-columns: auto auto auto;  
(3 autos, 3 columnas)
```



```
grid-template-columns: 80px auto 1rem;  
Se pueden mezclar unidades
```



```
grid-template-columns: 40px 1fr 2fr;  
Podemos usar la unidad flexible fr para distribuir el espacio restante  
en todas las columnas flexibles.
```



```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto;
  background-color: #2196F3;
  padding: 10px;
}
.grid-item {
  background-color: rgba(255, 255, 255, 0.8);
  border: 1px solid rgba(0, 0, 0, 0.8);
  padding: 20px;
  font-size: 30px;
  text-align: center;
}
</style>
</head>
<body>

<h1>Elementos Grid</h1>

<p>Un diseño de cuadrícula debe tener un elemento primario con la propiedad <em> display </em> establecida en <em> grid </em> o <em> inline-grid </em>. </p>

<p> Los hijos directos del contenedor de cuadrícula se convierten automáticamente en elementos de cuadrícula. </p>

<div class="grid-container">
<div class="grid-item">1</div>
<div class="grid-item">2</div>
<div class="grid-item">3</div>
<div class="grid-item">4</div>
<div class="grid-item">5</div>
<div class="grid-item">6</div>
<div class="grid-item">7</div>
<div class="grid-item">8</div>
<div class="grid-item">9</div>
</div>

</body>
```

</html>

## Elementos Grid

Un diseño de cuadrícula debe tener un elemento primario con la propiedad `display` establecida en `grid` o `inline-grid`.

Los hijos directos del contenedor de cuadrícula se convierten automáticamente en elementos de cuadrícula.

1	2	3
4	5	6
7	8	9

Para activar la cuadrícula **grid** hay que utilizar sobre el elemento contenedor la propiedad `display` y especificar el valor **grid** o **inline-grid**. Este valor influye en como se comportará la cuadrícula con el contenido exterior. El primero de ellos permite que la cuadrícula aparezca encima/debajo del contenido exterior (*en bloque*) y el segundo de ellos permite que la cuadrícula aparezca a la izquierda/derecha (*en línea*) del contenido exterior.

Tipo de elemento	Descripción
inline-grid	Establece una cuadrícula con ítems en línea, de forma equivalente a inline-block.
grid	Establece una cuadrícula con ítems en bloque, de forma equivalente a block.

Una vez elegido uno de estos dos valores, y establecida la propiedad `display` al elemento contenedor, hay varias formas de configurar nuestra cuadrícula `grid`. Comencemos con las propiedades que se aplican al elemento contenedor (*padre*).

## Ejemplo inline-grid

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: inline-grid;
  grid-template-columns: auto auto auto;
  background-color: #2196F3;
  padding: 10px;
}
.grid-item {
  background-color: rgba(255, 255, 255, 0.8);
  border: 1px solid rgba(0, 0, 0, 0.8);
  padding: 20px;
  font-size: 30px;
  text-align: center;
}
</style>
</head>
<body>

<h1>Elementos Grid</h1>

<p>Un diseño de cuadrícula debe tener un elemento primario con la propiedad <em> display </em> establecida en <em> grid </em> o <em> inline-grid </em>.</p>
```

<p> Los hijos directos del contenedor de cuadrícula se convierten automáticamente en elementos de cuadrícula. </p>

```
<div class="grid-container">
<div class="grid-item">1</div>
<div class="grid-item">2</div>
<div class="grid-item">3</div>
<div class="grid-item">4</div>
<div class="grid-item">5</div>
<div class="grid-item">6</div>
<div class="grid-item">7</div>
<div class="grid-item">8</div>
<div class="grid-item">9</div>
</div>

</body>
</html>
```

Al establecer el valor a inline-grid, el elemento ya no ocupa todo el ancho de la pantalla

### Elementos Grid

Un diseño de cuadrícula debe tener un elemento primario con la propiedad `display` establecida en `grid` o `inline-grid`.

Los hijos directos del contenedor de cuadrícula se convierten automáticamente en elementos de cuadrícula.

1	2	3
4	5	6
7	8	9

### Grid con filas y columnas

Es posible crear cuadrículas con un tamaño explícito. Para ello, sólo tenemos que usar las propiedades CSS `grid-template-columns` y `grid-template-rows`, que sirven para indicar las dimensiones de cada **celda** de la cuadrícula, diferenciando entre columnas y filas. Las propiedades son las siguientes:

Propiedad	Valor	Descripción
<code>grid-template-columns</code>	<code>[ c1 ] [ c2 ] ...</code>	Establece el <b>TAMAÑO</b> de cada columna ( <code>col_1, col_2...</code> ).
<code>grid-template-rows</code>	<code>[ f1 ] [ f2 ] ...</code>	Establece el <b>TAMAÑO</b> de cada fila ( <code>fila_1, fila_2...</code> ).

Una vez elegido uno de estos dos valores, y establecida la propiedad `display` al elemento contenedor, hay varias formas de configurar nuestra cuadrícula grid. Comencemos con las propiedades que se aplican al elemento contenedor (*padre*).

### Grid con filas y columnas

Es posible crear cuadrículas con un tamaño explícito. Para ello, sólo tenemos que usar las propiedades CSS `grid-template-columns` y `grid-template-rows`, que sirven

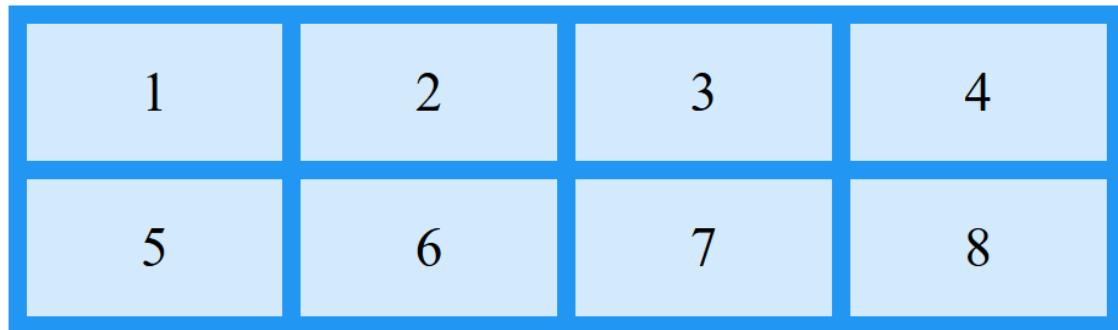
para indicar las dimensiones de cada **celda** de la cuadrícula, diferenciando entre columnas y filas. Las propiedades son las siguientes:

Propiedad	Valor	Descripción
grid-template-columns	[ <u>c1</u> ] [ <u>c2</u> ] ...	Establece el <b>TAMAÑO</b> de cada columna ( <u>col_1, col_2...</u> ).
grid-template-rows	[ <u>f1</u> ] [ <u>f2</u> ] ...	Establece el <b>TAMAÑO</b> de cada fila ( <u>fila_1, fila_2...</u> ).

## Propiedad grid-template-columns

Usa la propiedad *grid-template-columns* para definir el número de columnas en el diseño de la cuadrícula.

Este diseño de cuadrícula tiene cuatro columnas:



La propiedad *grid-gap* define el tamaño del espacio entre las filas y columnas en un diseño de cuadrícula, es una propiedad abreviada de las siguientes propiedades:

- [grid-row-gap](#)
- [grid-column-gap](#)

```
<!DOCTYPE html>
<html>
<head>
<style>

.grid-container {
  display: grid;
  grid-template-columns: auto auto auto auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

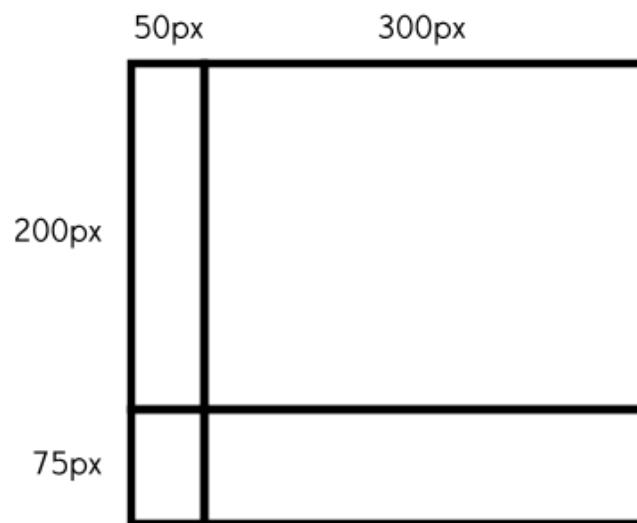

```

```
.grid-container > div {  
background-color: rgba(255, 255, 255, 0.8);  
text-align: center;  
padding: 20px 0;  
font-size: 30px;  
}  
</style>  
</head>  
<body>  
  
<h1>Propiedad grid-template-columns</h1>  
  
<p> Usa la propiedad <em> grid-template-columns </em> para definir el número de columnas en el diseño de la cuadrícula. </p>  
<p> Este diseño de cuadrícula tiene cuatro columnas: </p>  
  
<div class="grid-container">  
<div class="item1">1</div>  
<div class="item2">2</div>  
<div class="item3">3</div>  
<div class="item4">4</div>  
<div class="item5">5</div>  
<div class="item6">6</div>  
<div class="item7">7</div>  
<div class="item8">8</div>  
</div>  
</body>  
</html>
```

Conociendo estas dos propiedades, asumamos el siguiente código CSS:

```
.grid {  
display: grid;  
grid-template-columns: 50px 300px;  
grid-template-rows: 200px 75px;  
}
```

Esto significa que tendremos una cuadrícula con **2 columnas** (*la primera con 50px de ancho y la segunda con 300px de ancho*) y con **2 filas** (*la primera con 200px de alto y la segunda con 75px de alto*). Ahora, dependiendo del número de ítems (*elementos hijos*) que tenga el contenedor **grid**, tendremos una cuadrícula de 2x2 elementos (**4 ítems**), 2x3 elementos (**6 ítems**), 2x4 elementos (**8 ítems**) y así sucesivamente. Si el número de ítems es impar, la última celda de la cuadrícula se quedará vacía.



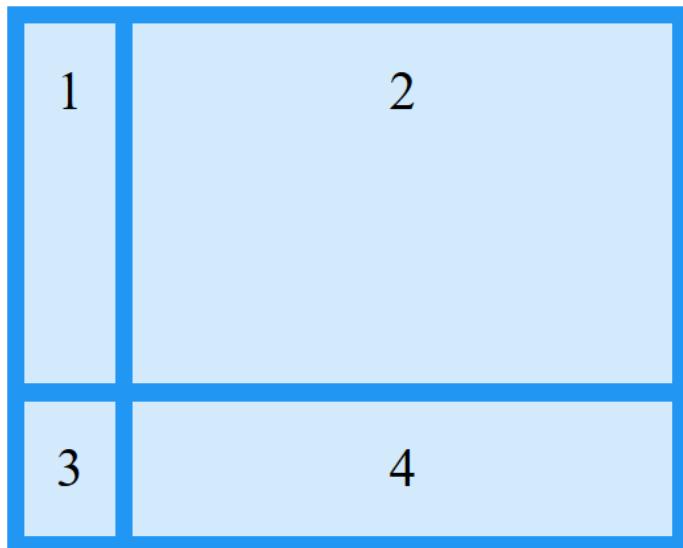
Añadir ítems



# Propiedad grid-template-columns

Usa la propiedad *grid-template-columns* para definir el número de columnas en el diseño de la cuadrícula.

Este diseño de cuadrícula tiene dos columnas:



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.grid-container {  
    width:360px;  
    display: grid;  
    grid-template-columns: 50px 300px;  
    grid-template-rows: 200px 75px;  
    grid-gap: 10px;
```

```
background-color: #2196F3;  
padding: 10px;  
}  
  
.grid-container > div {  
background-color: rgba(255, 255, 255, 0.8);  
text-align: center;  
padding: 20px 0;  
font-size: 30px;  
}  
</style>  
</head>  
<body>
```

## <h1>Propiedad grid-template-columns</h1>

<p> Usa la propiedad <em> grid-template-columns </em> para definir el número de columnas en el diseño de la cuadrícula. </p>

<p> Este diseño de cuadrícula tiene dos columnas: </p>

```
<div class="grid-container">  
<div class="item1">1</div>
```

```
<div class="item2">2</div>
```

```
<div class="item3">3</div>
```

```
<div class="item4">4</div>
```

```
</div>
```

```
</body>
```

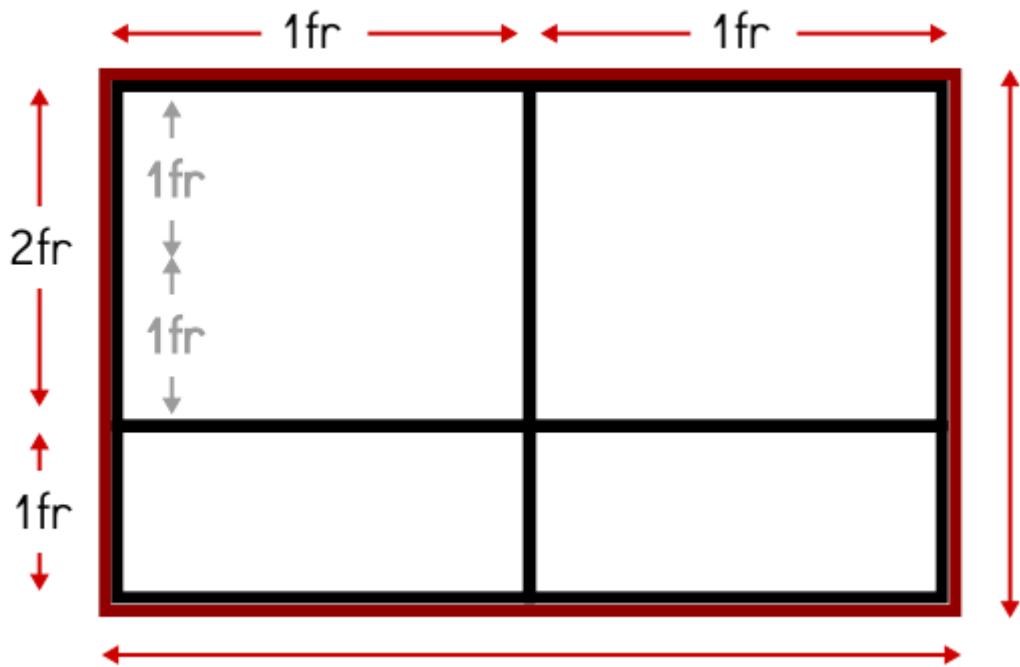
```
</html>
```

fr: Unidad fracción restante

En este ejemplo hemos utilizado píxeles como unidades de las celdas de la cuadrícula, sin embargo, también podemos utilizar otras unidades (e incluso combinarlas) como porcentajes, la palabra clave auto (que obtiene el tamaño restante) o la unidad especial fr (fraction), que simboliza una fracción de espacio restante en el grid. Veamos un código de ejemplo en acción:

```
.grid {  
    display: grid;  
    grid-template-columns: 1fr 1fr;  
    grid-template-rows: 2fr 1fr;  
}
```

Este nuevo ejemplo, se crea una cuadrícula de 2x2, donde el tamaño de ancho de la cuadrícula se divide en **dos columnas** (*mismo tamaño de ancho para cada una*), y el tamaño de alto de la cuadrícula se divide en **dos filas**, donde la primera ocupará el doble (2 *fr*) que la segunda (1 *fr*):



De esta forma, podemos tener un mejor control del espacio restante de la cuadrícula, y como utilizarlo.

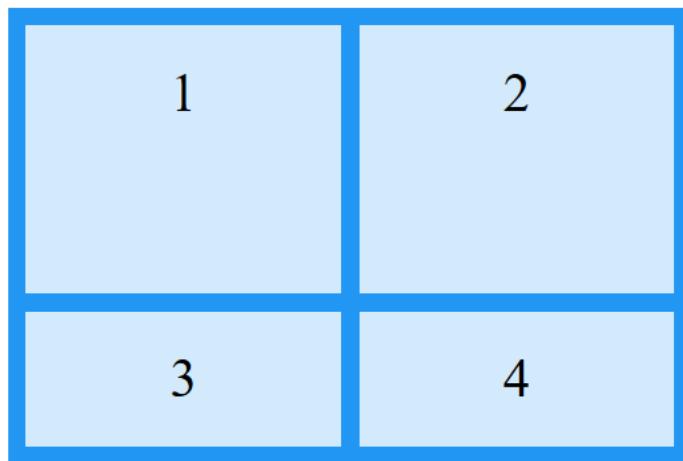
**Nota:** Se pueden combinar varias unidades diferentes, pudiendo utilizar píxeles (px) y fracciones restantes (fr), porcentajes (%) y fracciones restantes (fr) o combinaciones similares.

Ejemplo:

# Propiedad grid-template-columns

Usa la propiedad *grid-template-columns* para definir el número de columnas en el diseño de la cuadrícula.

Este diseño de cuadrícula tiene dos columnas:



```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
```

```
<style>
```

```
.grid-container {  
    width:360px;  
    display: grid;  
    grid-template-columns: 1fr 1fr;  
    grid-template-rows: 2fr 1fr;  
    grid-gap: 10px;  
    background-color: #2196F3;  
    padding: 10px;  
}
```

```
.grid-container > div {  
background-color: rgba(255, 255, 255, 0.8);  
text-align: center;  
padding: 20px 0;  
font-size: 30px;  
}  
</style>  
</head>  
<body>  
<h1>Propiedad grid-template-columns</h1>  
<p> Usa la propiedad <em> grid-template-columns </em> para definir el número de columnas en el diseño de la cuadrícula. </p>  
<p> Este diseño de cuadrícula tiene dos columnas: </p>  
<div class="grid-container">  
<div class="item1">1</div>  
<div class="item2">2</div>  
<div class="item3">3</div>  
<div class="item4">4</div>  
</div>  
</body>  
</html>
```

## Filas y columnas repetitivas

En algunos casos, en las propiedades `grid-template-columns` y `grid-template-rows` podemos necesitar indicar las mismas cantidades un número alto de veces, resultando repetitivo y molesto. Se puede utilizar la expresión `repeat()` para indicar

repetición de valores, indicando el número de veces que se repiten y el tamaño en cuestión.

La expresión a utilizar sería la siguiente: `repeat([número de veces], [valor o valores]):`

```
.grid {
  display: grid;
  grid-template-columns: 100px repeat(2, 50px) 200px;
  grid-template-rows: repeat(2, 50px 100px);
}
```

Asumiendo que tuvieramos un contenedor grid con 8 ítems hijos (*o más*), el ejemplo anterior crearía una cuadrícula con **4 columnas** (*la primera de 100px de ancho, la segunda y tercera de 50px de ancho y la cuarta de 200px de ancho*). Por otro lado, tendría **2 filas** (*la primera de 50px de alto, y la segunda de 100px de alto*). En el caso de tener más ítems hijos, el patrón se seguiría repitiendo.

El ejemplo anterior sería equivalente al código CSS siguiente:

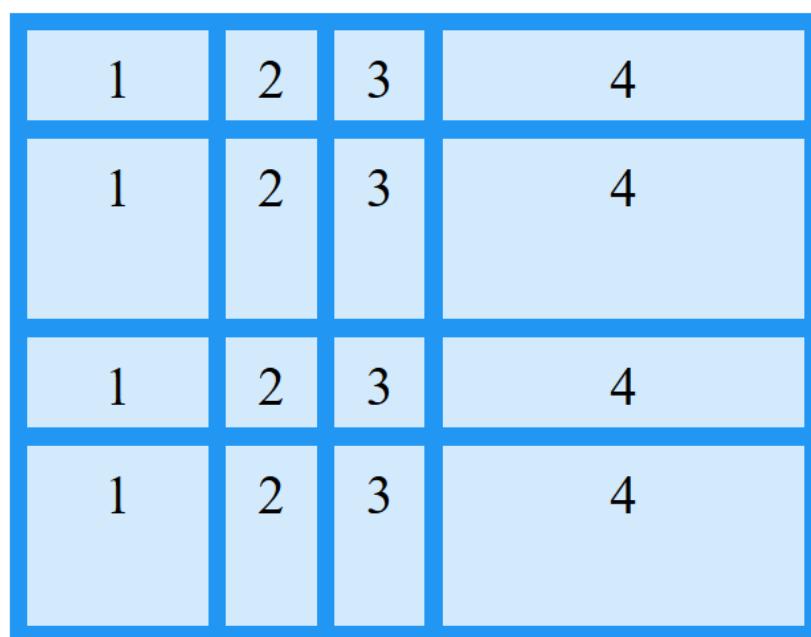
```
.grid {
  display: grid;
  grid-template-columns: 100px 50px 50px 200px;
  grid-template-rows: 50px 100px 50px 100px;
}
```

Ejemplo:

## Propiedad grid-template-columns

Usa la propiedad `grid-template-columns` para definir el número de columnas en el diseño de la cuadrícula.

Este diseño de cuadrícula tiene cuatro columnas:



```
<!DOCTYPE html>
<html>
<head>
<style>

.grid-container {
  width:430px;
  display: grid;
  grid-template-columns: 100px repeat(2, 50px) 200px;
  grid-template-rows: repeat(2, 50px 100px);
  height:330px;

  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 10px 0;
  font-size: 30px;
}
</style>
</head>
<body>

<h1>Propiedad grid-template-columns</h1>

<p> Usa la propiedad <em> grid-template-columns </em> para definir el número de columnas en el diseño de la cuadrícula. </p>
<p> Este diseño de cuadrícula tiene cuatro columnas: </p>

<div class="grid-container">
<div class="item1">1</div>
<div class="item2">2</div>
<div class="item3">3</div>
<div class="item4">4</div>
<div class="item1">1</div>
<div class="item2">2</div>
<div class="item3">3</div>
<div class="item4">4</div>
<div class="item1">1</div>
<div class="item2">2</div>
<div class="item3">3</div>
<div class="item4">4</div>
</div>
</body>
</html>
```

## Grid por áreas

<https://developer.mozilla.org/es/docs/Web/CSS/grid-template-areas>

Mediante los **grids CSS** es posible indicar el nombre y posición concreta de cada área de una cuadrícula. Para ello utilizaremos la propiedad `grid-template-areas`, donde debemos especificar el orden de las áreas en la cuadrícula. Posteriormente, en cada ítem hijo, utilizamos la propiedad `grid-area` para indicar el nombre del área del que se trata:

Propiedad	Descripción
<code>grid-template-areas</code>	Indica la disposición de las áreas en el grid. Cada texto entre comillas simboliza una fila.
<code>grid-area</code>	Indica el nombre del área. Se usa sobre ítems hijos del grid.

De esta forma, es muy sencillo crear una cuadrícula altamente personalizada en apenas unas cuantas líneas de CSS, con mucha flexibilidad en la disposición y posición de cada área:

```
.grid {  
display: grid;  
grid-template-areas: "head head"  
"menu main"  
"foot foot";  
}  
  
.a {grid-area: head; background:blue}  
.b {grid-area: menu; background:red}  
.c {grid-area: main; background:green}  
.d {grid-area: foot; background:orange}
```

Aplicando este código, conseguiríamos una cuadrícula donde:

- El Item 1, la cabecera (*head*), ocuparía toda la parte superior.
- El Item 2, el menú (*menu*), ocuparía el área izquierda del grid, debajo de la cabecera.
- El Item 3, el contenido (*main*), ocuparía el área derecha del grid, debajo de la cabecera.
- El Item 4, el pie de cuadrícula (*foot*), ocuparía toda la zona inferior del grid.



OJO: Ten en cuenta añadir contenido de texto en cada celda del grid, para que Grid CSS detecte que esa celda no está vacía.

En la propiedad `grid-template-areas`, en lugar de indicar el nombre del área a colocar, también podemos indicar una palabra clave especial:

- La palabra clave **none**: Indica que no se colocará ninguna celda en esta posición.
- Uno o más puntos (.): Indica que se colocará una celda vacía en esta posición.

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid {
  display: grid;
  grid-template-areas: "head head"
                      "menu main"
                      "foot foot";
}
```

```
.a { grid-area: head; background: blue }
.b { grid-area: menu; background: red }
```

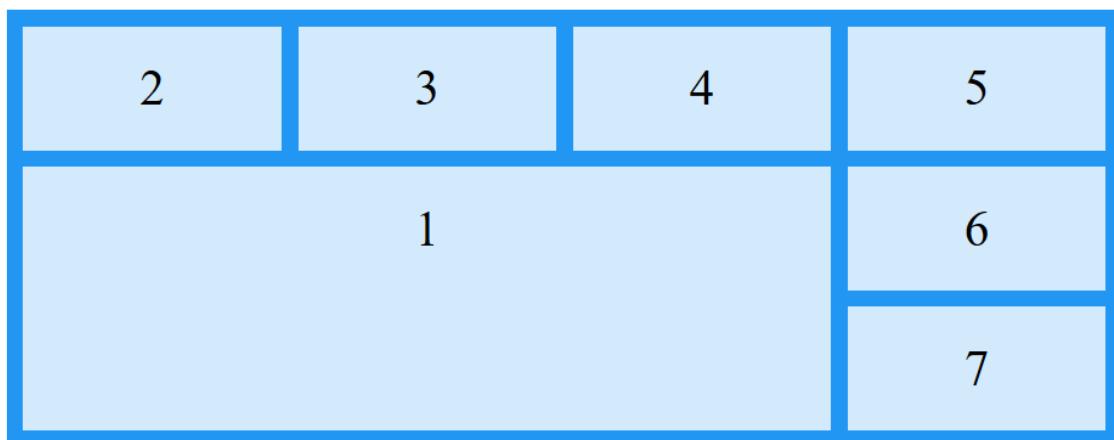
```
.c { grid-area: main; background: green }  
.d { grid-area: foot; background: orange }  
</style>  
</head>  
<body>  
<div class="grid">  
<div class="a">1</div>  
<div class="b">2</div>  
<div class="c">3</div>  
<div class="d">4</div>  
</div>  
Hola  
</body>  
</html>
```

## La propiedad grid-area

Puedes usar la propiedad grid-area para especificar dónde colocar un elemento.

La sintaxis es grid-row-start / grid-column-start / grid-row-end / grid-column-end.

En este ejemplo el elemento 1 comenzará en la fila 2 y la columna 1, y abarcará 2 filas y 3 columnas:



```
<!DOCTYPE html>
<html>
<head>
<style>
.grid-container {
  display: grid;
  grid-template-columns: auto auto auto auto;
  grid-gap: 10px;
  background-color: #2196F3;
  padding: 10px;
}

.grid-container > div {
  background-color: rgba(255, 255, 255, 0.8);
  text-align: center;
  padding: 20px 0;
  font-size: 30px;
}

.item1 {
  grid-area: 2 / 1 / span 2 / span 3;
}

</style>
</head>
<body>
<h1> La propiedad grid-area </h1>

<p> Puedes usar la propiedad <em> grid-area </em> para especificar dónde colocar un elemento. </p>
<p> La sintaxis es grid-row-start / grid-column-start / grid-row-end / grid-column-end. </p>
```

<p> El elemento 1 comenzará en la fila 2 y la columna 1, y abarcará 2 filas y 3 columnas: </p>

```
<div class="grid-container">
<div class="item1">1</div>
<div class="item2">2</div>
<div class="item3">3</div>
<div class="item4">4</div>
<div class="item5">5</div>
<div class="item6">6</div>
<div class="item7">7</div>
</div>
</body>
</html>
```

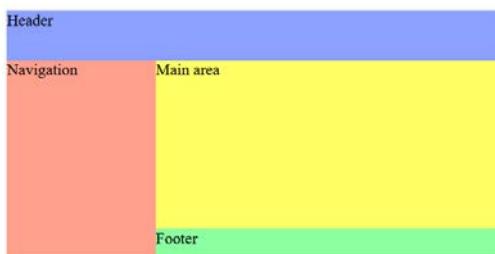
Mas info:

[https://www.w3schools.com/cssref/pr\\_grid-area.asp](https://www.w3schools.com/cssref/pr_grid-area.asp)

Mas avanzado

<https://css-tricks.com/snippets/css/complete-guide-grid/>

## Ejemplo maquetación



```
<!DOCTYPE html>
<html>
<head>
<style>
#page {
    display: grid;
```

```
width: 100%;  
height: 250px;  
grid-template-areas: "head head"  
                      "nav main"  
                      "nav foot";  
grid-template-rows: 50px 1fr 30px;  
grid-template-columns: 150px 1fr;  
}  
  
  
#page > header {  
  grid-area: head;  
  background-color: #8ca0ff;  
}  
  
  
#page > nav {  
  grid-area: nav;  
  background-color: #ffa08c;  
}  
  
  
#page > main {  
  grid-area: main;  
  background-color: #ffff64;  
}  
  
  
#page > footer {  
  grid-area: foot;  
  background-color: #8cffa0;  
}  
</style>  
</head>
```

```
<body>
<section id="page">
<header>Header</header>
<nav>Navigation</nav>
<main>Main area</main>
<footer>Footer</footer>
</section>
</body>
</html>
```

## Grid con huecos

Por defecto, la cuadrícula tiene todas sus celdas pegadas a sus celdas contiguas. Aunque sería posible darle un `margin` a las celdas dentro del contenedor, existe una forma más apropiada, que evita los problemas clásicos de los modelos de caja: los huecos (*gutters*).

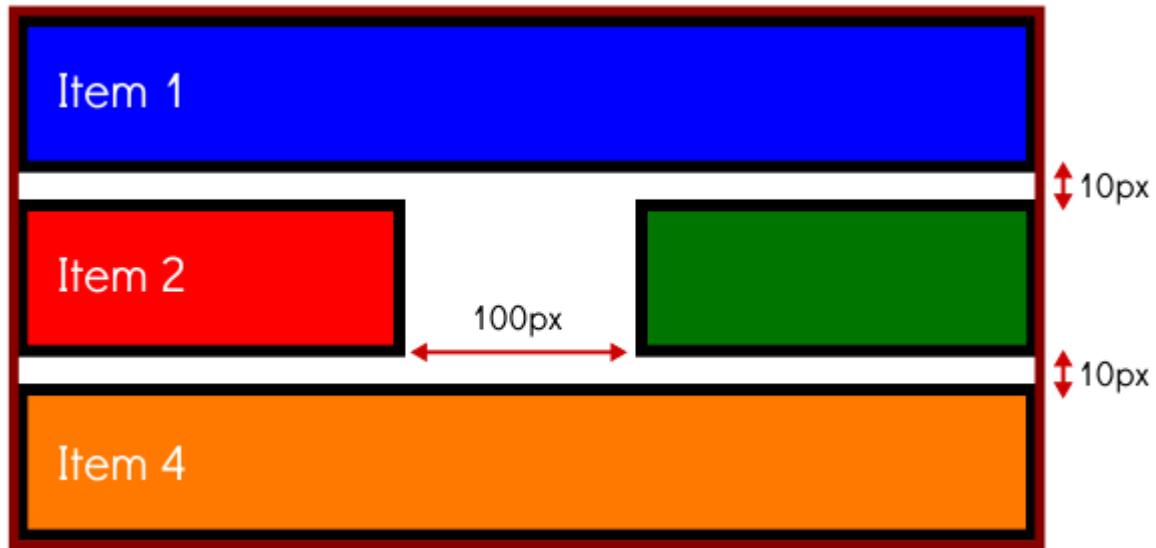
Para especificar los **huecos** (*espacio entre celdas*) podemos utilizar las propiedades `grid-column-gap` o `grid-row-gap`. En ellas indicaremos el tamaño de dichos huecos:

Propiedad	Descripción
<code>grid-column-gap</code>	Establece el <b>TAMAÑO</b> de los huecos entre columnas ( <i>líneas verticales</i> ).
<code>grid-row-gap</code>	Establece el <b>TAMAÑO</b> de los huecos entre filas ( <i>líneas horizontales</i> ).

Tomemos el ejemplo anterior como base. En él, le indicamos estas propiedades para colocar **huecos** entre las celdas de la cuadrícula. El código a añadir al ejemplo anterior sería el siguiente:

```
.grid {
  grid-column-gap: 100px;
  grid-row-gap: 10px;
}
```

Con esto, obtendríamos un resultado como el siguiente, indicando huecos entre columnas de 100px y huecos entre filas de 10px:



### Código completo

```
<!DOCTYPE html>

<html>
<head>
<style>

.grid {
    display: grid;
    grid-template-areas: "head head"
                         "menu main"
                         "foot foot";
    grid-column-gap: 100px;
    grid-row-gap: 10px;
}

.a { grid-area: head; background: blue }
.b { grid-area: menu; background: red }
```

```
.c { grid-area: main; background: green }

.d { grid-area: foot; background: orange }

</style>

</head>

<body>

<div class="grid">

<div class="a">1</div>

<div class="b">2</div>

<div class="c">3</div>

<div class="d">4</div>

</div>

Hola

</body>

</html>
```

## Atajo: Grid con huecos

Existe una propiedad de atajo para las propiedades `grid-column-gap` y `grid-row-gap`, permitiéndonos la posibilidad de no tener que indicarlas por separado.

La propiedad en cuestión sería `grid-gap` y se utilizaría de la siguiente forma:

```
.grid {
/* grid-gap: <row-gap><column-gap> */
grid-gap: 20px 80px;

/* grid-gap: <row-column-gap> */
grid-gap: 40px;
/* Equivalente a grid-gap: 40px 40px; */
}
```

## Posición en el grid

Existen una serie de propiedades que se pueden utilizar para colocar los ítems dentro de la cuadrícula. Con ellas podemos distribuir los elementos de una forma muy sencilla y cómoda. Dichas propiedades son `justify-items` y `align-items`, que ya conocerás del módulo CSS **flexbox**:

<https://developer.mozilla.org/es/docs/Web/CSS/justify-items>

<https://developer.mozilla.org/es/docs/Web/CSS/align-items>

<https://css-tricks.com/snippets/css/complete-guide-grid/>

<https://lenguajecss.com/p/css/propiedades/grid-css>

Propiedad	Valores	Descripción
justify-items	start   end   center   stretch	Distribuye los elementos en el eje horizontal.
align-items	start   end   center   stretch	Distribuye los elementos en el eje vertical.

Estas propiedades se aplican sobre el elemento contenedor padre, pero afectan a los ítems hijos, por lo que actúan sobre la distribución de cada uno de los hijos. En el caso de que queramos que uno de los ítems hijos tengan una distribución diferente al resto, aplicamos la propiedad `justify-self` o `align-self` sobre el ítem hijo en cuestión, sobreescribiendo su distribución.

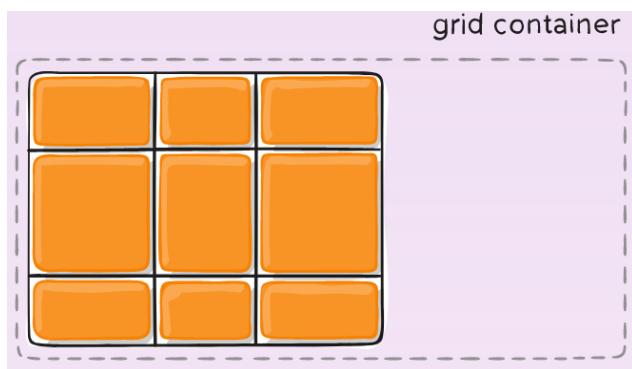
Estas propiedades funcionan exactamente igual que sus análogas `justify-items` o `align-items`, sólo que en lugar de indicarse en el elemento padre contenedor, se hace sobre un elemento hijo. Las propiedades sobre ítems hijos las veremos más adelante.

También podemos utilizar las propiedades `justify-content` o `align-content` para modificar la distribución de todo el contenido en su conjunto, y no sólo de los ítems por separado:

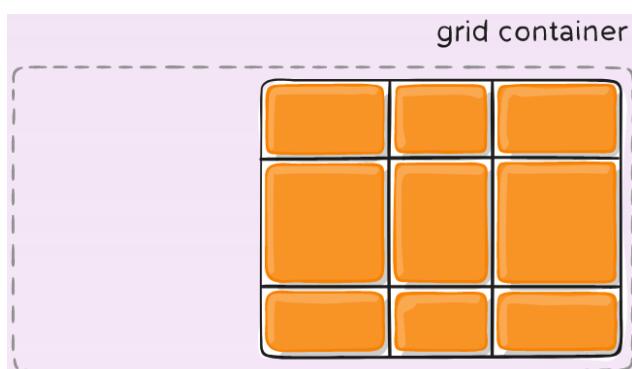
<https://developer.mozilla.org/es/docs/Web/CSS/justify-content>

Propiedad	Valores
justify-content	start   end   center   stretch   space-around   space-between   space-evenly
align-content	start   end   center   stretch   space-around   space-between   space-evenly

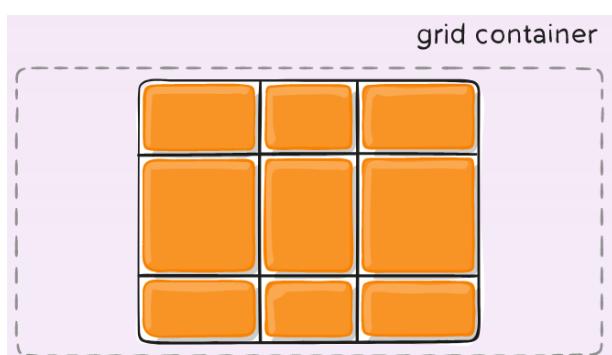
```
.container {
    justify-content: start;
}
```



```
.container {  
  justify-content: end;  
}
```



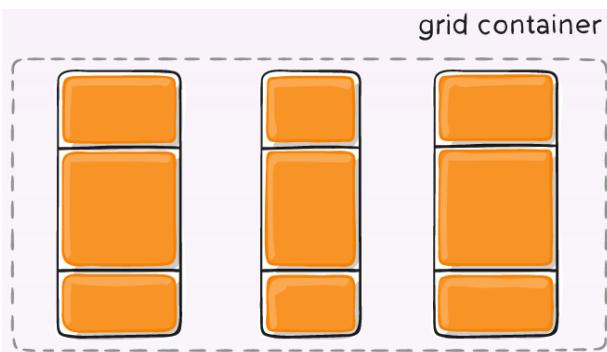
```
.container {  
  justify-content: center;  
}
```



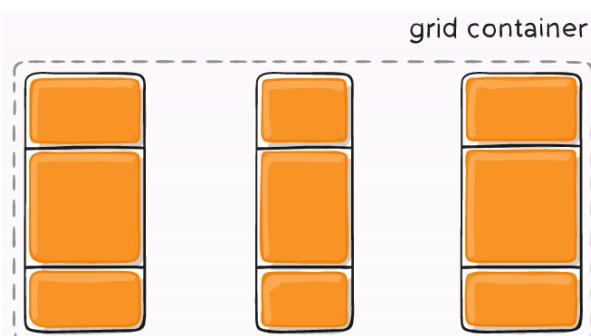
```
.container {  
  justify-content: stretch;  
}
```



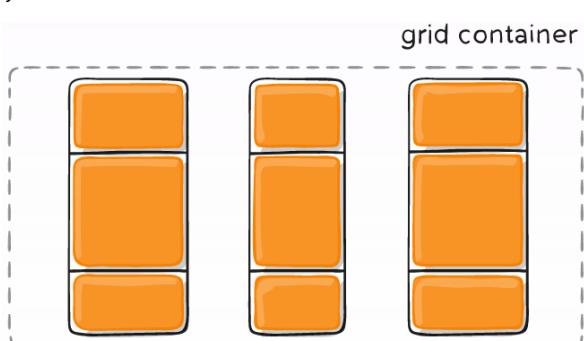
```
.container {  
  justify-content: space-around;  
}
```



```
.container {  
  justify-content: space-between;  
}
```



```
.container {  
  justify-content: space-evenly;  
}
```



De esta forma, podemos controlar prácticamente todos los aspectos de posicionamiento de la cuadrícula directamente desde los estilos CSS de su contenedor padre:

```
<!DOCTYPE html>
<html>
<head>
<style>
.grid {
    display: grid;
    grid-template-areas: "head head"
                         "menu main"
                         "foot foot";
    justify-items:stretch; /*Distribuye los elementos en el eje horizontal*/
    align-items: stretch; /*Distribuye los elementos en el eje vertical*/
    justify-content:stretch; /* distribución de todo el contenido y no solo los items por separado
en el eje horizontal*/
    align-content:stretch; /* distribución de todo el contenido y no solo los items por separado en
el eje vertical*/
}

.a { grid-area: head; background: blue }
.b { grid-area: menu; background: red }
.c { grid-area: main; background: green }
.d { grid-area: foot; background: orange }

</style>
</head>
<body>
<div class="grid">
<div class="a">Item 1</div>
<div class="b">Item 2</div>
<div class="c">Item 3</div>
<div class="d">Item 4</div>
</div>
</body>
```

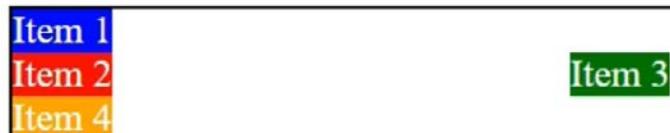
</html>

```
15 | justify-items:stretch;
16 | align-items:stretch;
17 |
18 | justify-content:stretch;
19 | align-content:stretch;
20 }
21
22 + .a { grid-area: head; background:blue }
23 + .b { grid-area: menu; background:red }
24 + .c { grid-area: main; background:green }
25 + .d { grid-area: foot; background:orange }
```



Siguiente ejemplo:

```
15 | justify-items:start;
16 | align-items:stretch;
17 |
18 | justify-content:stretch;
19 | align-content:stretch;
20 }
21
22 + .a { grid-area: head; background:blue }
23 + .b { grid-area: menu; background:red }
24 + .c { grid-area: main; background:green }
25 + .d { grid-area: foot; background:orange }
```



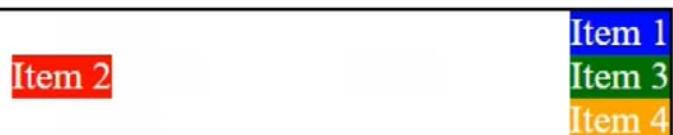
Siguiente ejemplo:

```
15   justify-items:center;
16   align-items:stretch;
17
18   justify-content:stretch;
19   align-content:stretch;
20 }
21
22 .a { grid-area: head; background:blue }
23 .b { grid-area: menu; background:red }
24 .c { grid-area: main; background:green }
25 .d { grid-area: foot; background:orange }
```



Siguiente ejemplo:

```
15   justify-items:end;
16   align-items:stretch;
17
18   justify-content:stretch;
19   align-content:stretch;
20 }
21
22 .a { grid-area: head; background:blue }
23 .b { grid-area: menu; background:red }
24 .c { grid-area: main; background:green }
25 .d { grid-area: foot; background:orange }
```



Siguiente ejemplo:

```
15   justify-items:stretch;
16   |
17   align-items:stretch;
18
19   justify-content:stretch;
20   align-content:stretch;
21 }
22
23 .a { grid-area: head; background:blue }
24 .b { grid-area: menu; background:red }
25 .c { grid-area: main; background:green }
26 .d { grid-area: foot; background:orange }
```



Siguiente ejemplo:

```
15   justify-items:stretch;
16   height:200px;
17   align-items:stretch;
18
19   justify-content:stretch;
20   align-content:stretch;
21 }
22
23 .a { grid-area: head; background:blue }
24 .b { grid-area: menu; background:red }
25 .c { grid-area: main; background:green }
26 .d { grid-area: foot; background:orange }
```



Siguiente ejemplo:

```
15   justify-items:stretch;
16   height:200px;
17   align-items:start;
18
19   justify-content:stretch;
20   align-content:stretch;
21 }
22
23 .a { grid-area: head; background:blue }
24 .b { grid-area: menu; background:red }
25 .c { grid-area: main; background:green }
26 .d { grid-area: foot; background:orange }
```



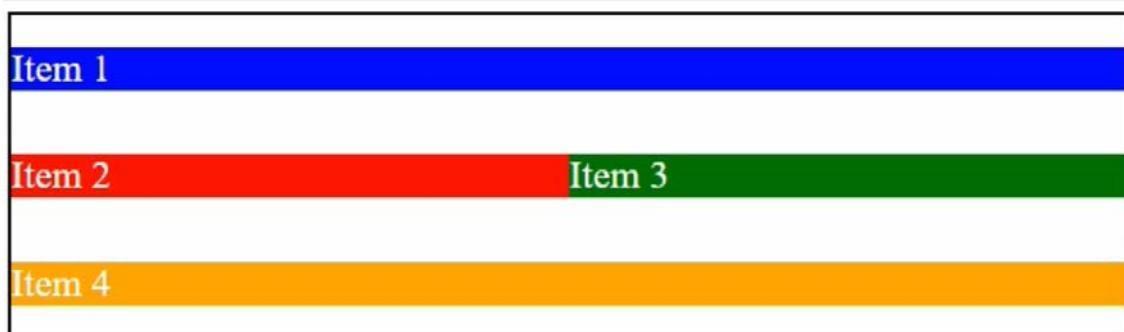
Siguiente ejemplo:

```
15   justify-items:stretch;
16   height:200px;
17   align-items:end;
18
19   justify-content:stretch;
20   align-content:stretch;
21 }
22
23 .a { grid-area: head; background:blue }
24 .b { grid-area: menu; background:red }
25 .c { grid-area: main; background:green }
26 .d { grid-area: foot; background:orange }
```



Siguiente ejemplo:

```
15 justify-items:stretch;
16 height:200px;
17 align-items:center;
18
19 justify-content:stretch;
20 align-content:stretch;
21 }
22
23 .a { grid-area: head; background:blue }
24 .b { grid-area: menu; background:red }
25 .c { grid-area: main; background:green }
26 .d { grid-area: foot; background:orange }
```



Siguiente ejemplo:

```
15 justify-items:stretch;
16 height:200px;
17 align-items:stretch;
18 |
19 justify-content:stretch;
20 align-content:stretch;
21 }
22
23 .a { grid-area: head; background:blue }
24 .b { grid-area: menu; background:red }
25 .c { grid-area: main; background:green }
26 .d { grid-area: foot; background:orange }
```



Siguiente ejemplo:

```
15 justify-items:stretch;
16 height:200px;
17 align-items:stretch;
18
19 justify-content:start;
20 align-content:stretch;
21 }
22
23 .a { grid-area: head; background:blue }
24 .b { grid-area: menu; background:red }
25 .c { grid-area: main; background:green }
26 .d { grid-area: foot; background:orange }
```



Siguiente ejemplo:

```
15 justify-items:stretch;
16 height:200px;
17 align-items:stretch;
18
19 justify-content:end;
20 align-content:stretch;
21 }
22
23 .a { grid-area: head; background:blue }
24 .b { grid-area: menu; background:red }
25 .c { grid-area: main; background:green }
26 .d { grid-area: foot; background:orange }
```



Siguiente ejemplo:

```
15   justify-items:stretch;
16   height:200px;
17   align-items:stretch;
18
19   justify-content:center;
20   align-content:stretch;
21 }
22
23 .a { grid-area: head; background:blue }
24 .b { grid-area: menu; background:red }
25 .c { grid-area: main; background:green }
26 .d { grid-area: foot; background:orange }
```



Siguiente ejemplo:

```
15   justify-items:stretch;
16   height:200px;
17   align-items:stretch;
18
19   justify-content:center;
20   align-content:end;
21 }
22
23 .a { grid-area: head; background:blue }
24 .b { grid-area: menu; background:red }
25 .c { grid-area: main; background:green }
26 .d { grid-area: foot; background:orange }
```



Siguiente ejemplo:

```
15 justify-items:stretch;
16 height:200px;
17 align-items:stretch;
18
19 justify-content:center;
20 align-content:start;
21 }
22
23 .a { grid-area: head; background:blue }
24 .b { grid-area: menu; background:red }
25 .c { grid-area: main; background:green }
26 .d { grid-area: foot; background:orange }
```



Siguiente ejemplo:

```
15 justify-items:stretch;
16 height:200px;
17 align-items:stretch;
18
19 justify-content:stretch;
20 align-content:start;
21 }
22
23 .a { grid-area: head; background:blue }
24 .b { grid-area: menu; background:red }
25 .c { grid-area: main; background:green }
26 .d { grid-area: foot; background:orange }
```



Siguiente ejemplo:

The screenshot shows a code editor with the following CSS code:

```

15 justify-items:stretch;
16 height:200px;
17 align-items:stretch;
18
19 justify-content:stretch;
20 align-content:stretch;
21 }
22
23 .a { grid-area: head; background:blue }
24 .b { grid-area: menu; background:red }
25 .c { grid-area: main; background:green }
26 .d { grid-area: foot; background:orange }

```

Below the code is a visual representation of the layout. It consists of four horizontal colored boxes: blue (Item 1), red (Item 2), green (Item 3), and orange (Item 4). Item 1 spans the entire width. Item 2 is narrower than Item 3. Item 4 is the widest.

Ver imagen:

<https://i.lenguajecss.com/assets/css/propiedades/justify-align-grid-css.gif>

## Ajuste automático de celdas

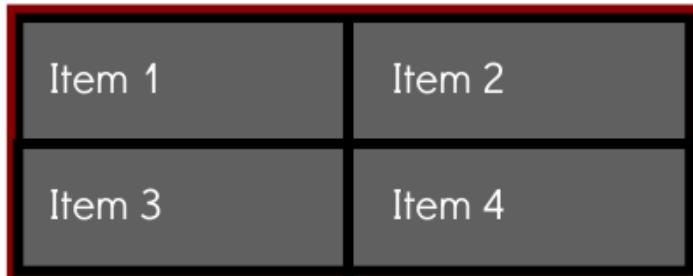
Es posible utilizar las propiedades `grid-auto-columns` y `grid-auto-rows` para darle un tamaño automático a las celdas de la cuadrícula. Para ello, sólo hay que especificar el tamaño deseado en cada una de las propiedades. Además, también podemos utilizar `grid-auto-flow` para indicar el flujo de elementos en la cuadrícula, y especificar por donde se irán añadiendo. Las propiedades son las siguientes:

Propiedad	Valores	Descripción
<code>grid-auto-columns</code>	TAMAÑO	Indica el tamaño automático de ancho que tendrán las columnas.
<code>grid-auto-rows</code>	TAMAÑO	Indica el tamaño automático de alto que tendrán las filas.
<code>grid-auto-flow</code>	row   column   dense	Utiliza un algoritmo de autocolocación (intenta llenar huecos).

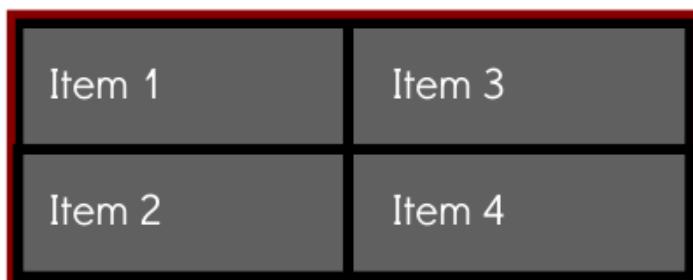
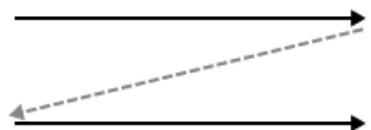
Mas info:

<https://developer.mozilla.org/es/docs/Web/CSS/grid-auto-columns>

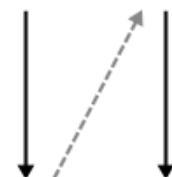
Un ejemplo de como se insertarían los elementos en una cuadrícula de 2x2 utilizando grid-auto-flow por columnas o por filas:



grid-auto-flow: row



grid-auto-flow: column



## Atajo: Grid

Por último, existe una propiedad `grid` que sirve de atajo para la mayoría de propiedades CSS relativas a cuadrículas. Su esquema de utilización sería el siguiente, junto a algunos ejemplos:

```
.grid {
/* grid: <grid-template><grid-auto-flow><grid-auto-rows> / <grid-auto-columns> */

grid:100px 20px;
grid:200px repeat(2, 100px)300px;
grid: row;
grid: column dense;
grid: row 200px;
grid: row 400px / 150px;
}
```

## Propiedades para ítems hijos

Hasta ahora, salvo algunas excepciones como `justify-self`, `align-self` o `grid-area`, hemos visto propiedades CSS que se aplican solamente al contenedor padre de una cuadrícula. A continuación, vamos a ver ciertas propiedades que en su lugar, se aplican a cada ítem hijo de la cuadrícula, para alterar o cambiar el comportamiento específico de dicho elemento, que no se comporta como la mayoría.

Algunas de las propiedades vistas hasta ahora son las siguientes:

Propiedad	Descripción
justify-self	Altera la justificación del ítem hijo en el eje horizontal.
align-self	Altera la alineación del ítem hijo en el eje vertical.
grid-area	Indica un nombre al área especificada, para su utilización con <a href="#">grid-template-areas</a> .

Sin embargo, existen algunas propiedades más, referentes en este caso, a la posición de los hijos de la cuadrícula donde va a comenzar o terminar una fila o columna. Las propiedades son las siguientes:

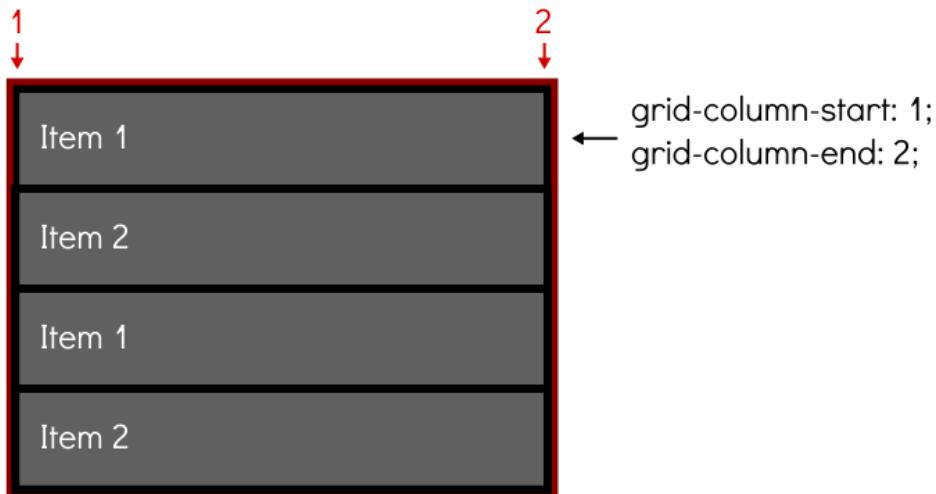
Propiedad	Descripción
grid-column-start	Indica en que columna empezará el ítem de la cuadrícula.
grid-column-end	Indica en que columna terminará el ítem de la cuadrícula.
grid-row-start	Indica en que fila empezará el ítem de la cuadrícula.
grid-row-end	Indica en que fila terminará el ítem de la cuadrícula.

Con dichas propiedades, podemos indicar el siguiente código CSS sobre el primer ítem de una cuadrícula de 4 ítems:

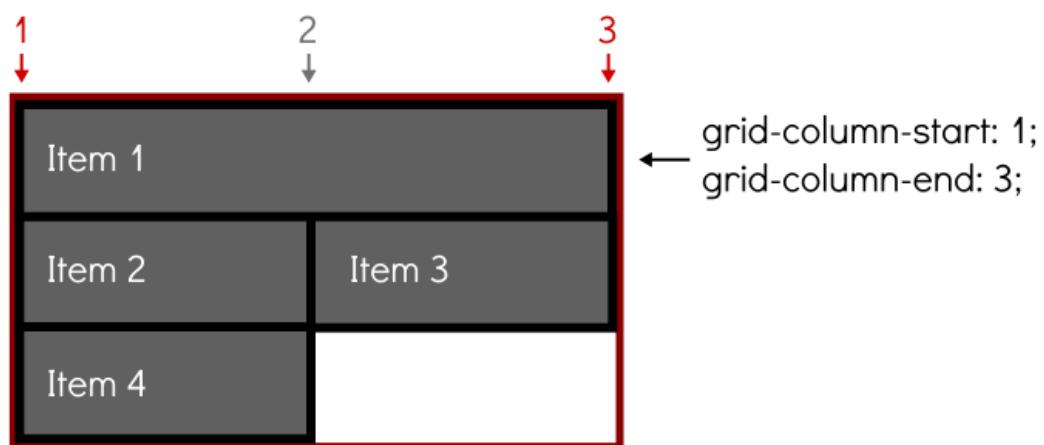
```
.grid {
display:grid;
}

.a {
grid-column-start:1;
grid-row-end:2;
}
```

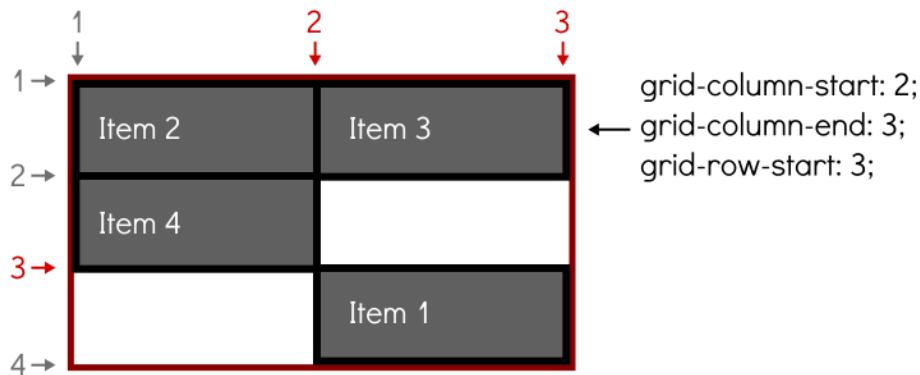
De esta forma, tenemos una cuadrícula de 4 elementos, en el que indicamos la posición del ítem 1 (*elemento HTML con clase .a*): comenzando en la columna 1 y acabando en el inicio de la columna 2:



Ese sería el funcionamiento normal. Donde se ve la utilidad de estas propiedades, es si variamos los valores de forma que tomen posiciones diferentes, como por ejemplo, si indicamos que el ítem 1 debe comenzar en la columna 1, pero acabar en la columna 3 (*ocupando la hipotética primera y segunda celda*):



En este nuevo ejemplo, comenzamos el primer ítem en la columna 2 y lo acabamos al principio de la columna 3, por lo que la celda permanecerá en la posición de la segunda columna. Además, añadimos la propiedad `grid-row-start` que hace lo mismo que hasta ahora, pero con las filas. En este caso, le indicamos que comience en la fila 3, por lo que el ítem 1 se desplaza a una nueva fila de la cuadrícula, dejando en la anterior el ítem 4:



También es posible utilizar la palabra clave **span** seguida de un número, que indica que abarque hasta esa columna o celda.

### Atajo: **grid-column** y **grid-row**

El módulo grid de CSS proporciona las propiedades de atajo **grid-column** y **grid-row** donde se nos permite escribir en un formato abreviado las propiedades anteriores. Su sintaxis sería la siguiente:

```
.grid {
  display: grid;
}

.a {
/* grid-column: <grid-column-start><grid-column-end> */
/* grid-row: <grid-row-start><grid-row-end> */
  grid-column: auto;
  grid-column: 4 / 6;
  grid-column: span 3;
  grid-column: span 3 / 6;
}
```

La especificación de Grid CSS ya tiene buen soporte en la actualidad y puede utilizarse en producción. Eso sí, si requieres soporte para navegadores antiguos, sé cuidadoso. Actualmente, su soporte es el siguiente:



## Más ejemplos

### Establecer la forma de un elemento(**Clip** está obsoleto, pero es el que funciona)

En este ejemplo se muestra cómo establecer la forma de un elemento. El elemento está recortado

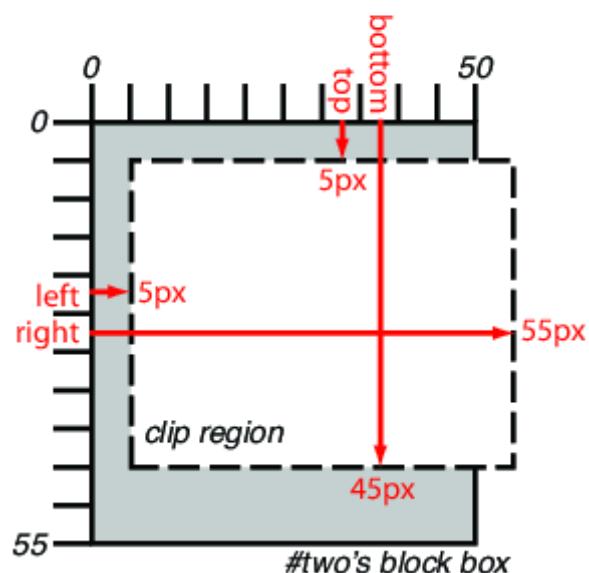
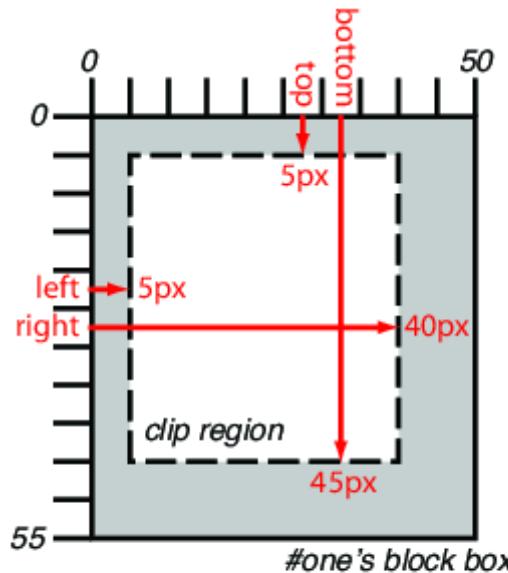
Ejemplo:

```
p#one { clip: rect(5px, 40px, 45px, 5px); }
p#two { clip: rect(5px, 55px, 45px, 5px); }
```

asumiendo que ambos “Ps” tienen unas dimensiones de 50px X 55 px, crearemos respectivamente las regiones recortadas delimitadas por puntos de la siguiente imagen:

**Las distancias siempre se miden desde la parte de arriba y desde la parte de la izquierda.**

**Solo funciona con posicionamiento absolute y fixed (COMPROBAR)**



<http://w3.unpocodetodo.info/css3/clip-path.php>

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```
img
{
position: absolute;
clip: rect(0px,60px,200px,0px);

}
</style>
</head>

<body>

</body>
</html>
```

**clip-path es el nuevo equivalente:**

```
<!DOCTYPE html>
<html>
<head>
<style>
img
{
position: absolute;
-webkit-clip-path: rect(0px,60px,200px,0px);
clip-path: rect(0px,60px,200px,0px);
}
</style>
</head>

<body>

</body>
</html>
```

podemos crear cuatro formas geométricas básicas: círculos, elipses, rectángulos y polígonos de x lados (también podemos hacer máscaras y animaciones)

## Inset()

```
inset( <shape-arg>{1,4} [round <border-radius>]? )
```

La función `inset()` permite la creación de formas rectangulares. La primera parte de la función sigue las mismas reglas que la versión abreviada de la propiedad `margin`, permitiendo uno, dos o cuatro valores. Si se usan cuatro valores el orden es superior, derecha, inferior e izquierda y definen la distancia entre los lados del rectángulo creado con respecto a los bordes del elemento sobre el que se aplica la función.

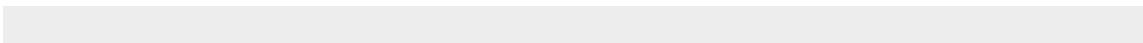
```
.objeto {
    clip-path: inset(100px 100px 100px 100px);
}
```

Clip-path:inset(50px); 50 px en todos los lados

Clip-path:inset(50px 30px); 50 px hacia arriba y hacia abajo.

Clip-path:inset(50px 30px 40px); 50 px hacia arriba, 30px a la derecha y a la izquierda y 40px abajo.

Clip-path:inset(20px 10px 30px 15px); El orden sería: 20px arriba, 10px a la derecha, 30px abajo y 15px a la izquierda.



## Circle()

```
circle( [<shape-radius>]? [at <position>]? )
```

La función `circle()` se usa para crear círculos y ambos valores son opcionales. Si algunos de ambos valores se omite se tomará un valor por defecto marcado por el navegador.

```
.objeto { clip-path: circle(10px at 20px 30px); }
```

## Ellipse()

```
ellipse( [<shape-radius>{2}]? [at <position>]? )
```

La función `ellipse()` se usa para crear una ellipse y toma la misma lista de argumentos que la función `circle()`, a excepción que en lugar de tomar un valor para el radio toma dos, un radio para el eje x y otro para el eje y, en ese orden en concreto.

```
.objeto {
    clip-path: ellipse(10px 20px at 10px 10px);
}
```

En el ejemplo recortamos una ellipse posicionada a 10px horizontalmente y 10px verticalmente, y cuyos radios son de 10px y 20px respectivamente.

## Polygon()

```
polygon( [<fill-rule> , ]? [<shape-arg><shape-arg>]# )
```

La función `polygon()` se usa para construir formas complejas usando un número determinado de puntos. Estos puntos se definen mediante pares de coordenadas (`<shape-arg><shape-arg>`). El primer argumento determina la posición x del punto y el segundo la posición y del mismo. La forma poligonal final se consigue uniendo la sucesión de puntos, para al final unir el último punto al primero de la sucesión. Cada par de coordenadas debe ser separado mediante comas.

```
.objeto { clip-path: polygon(0 40px, 75px 0, 150px 40px, 150px 120px, 75px 200px, 0 120px); }
```

La propiedad `clip-path` es relativamente nueva, de hecho pasó a "candidate recommendation" el 20 de Marzo de 2014, y ya sabemos lo lentas que van estas cosas.

El soporte es aún pobre, de hecho ni IE ni Edge la soportan aún, Firefox lo hace de forma parcial, y Chrome, Opera y Safary necesitan el prefijo `-webkit-`. Puedes mantenerte al tanto de la evolución de su compatibilidad en [caniuse.com](http://caniuse.com).

Otro ejemplo (en Firefox no va, de momento)



```
<!DOCTYPE html>
<html>
<head>
<style>
img {
    width: 120px;
    margin: 20px;
}

.clip-circle {
    clip-path: circle(60px at center);
-webkit-clip-path: circle(60px at center);
}

.clip-ellipse {
    clip-path: ellipse(60px 40px at center);
-webkit-clip-path: ellipse(60px 40px at center);
}

}
.clip-polygon {
    clip-path: polygon(5% 5%, 100% 0%, 100% 75%, 75% 75%, 75%
100%, 50% 75%, 0% 75%);
-webkit-clip-path: polygon(5% 5%, 100% 0%, 100% 75%, 75% 75%, 75%
100%, 50% 75%, 0% 75%);
```

}

```
body {  
    background: linear-gradient(  
        to bottom,  
        red,  
        yellow  
    );  
}  
  
body, html {  
    height: 100%;  
    margin: 0;  
}  
.clip-path {  
    clip-path: inset(100px 50px);  
}  
</style>  
</head>  
  
<body>  
  
  
  
  
  
  
  
  
</body>  
</html>
```

#### Mas ejemplos

<https://css-tricks.com/clipping-masking-css/>

<http://www.desarrollolibre.net/blog/tema/140/css/la-propiedad-clip-path-en-css-para-seleccionar-regiones-a-mostrar-en-elementos#.Wm7Tm7zibIU>

<https://bennettfeely.com/clippy/>

<http://www.sarasoueidan.com/demos/css-svg-clipping/html-img-clipped-with-css-svg-clippath/index.html>

### ¿Cómo mostrar desbordamiento en un elemento mediante desplazamiento?

En este ejemplo se muestra cómo establecer la propiedad de desbordamiento para crear una barra de desplazamiento cuando el contenido de un elemento es demasiado grande para caber en un área específica.

```
<!DOCTYPE html>
<html>
<head>
<style>
div.scroll
{
background-color: #00FFFF;
width: 100px;
height: 100px;
overflow: scroll;
}

div.hidden
{
background-color: #00FF00;
width: 100px;
height: 100px;
overflow: hidden;
}
</style>
</head>

<body>
<p>The overflow property specifies what to do if the content of an element exceeds the size of the element's box.</p>

<p>overflow: scroll</p>
<div class="scroll">You can use the overflow property when you want to have better control of the layout. The default value is visible.</div>

<p>overflow: hidden</p>
<div class="hidden">You can use the overflow property when you want to have better control of the layout. The default value is visible.</div>
</body>
</html>
```

overflow:scroll

You can use the overflow property when you want to have better control of the layout. The default value is visible.

overflow:hidden

You can use the overflow property when you want to have better control of the layout. The default value is visible.

### Cómo configurar el navegador para manejar automáticamente desbordamiento

Este ejemplo muestra cómo configurar el navegador para manejar automáticamente desbordamiento.

```
<!DOCTYPE html>
<html>
<head>
<style>
div
{
    background-color: #00FFFF;
    width: 150px;
    height: 150px;
    overflow: auto;
}
</style>
</head>

<body>
<p>The overflow property decides what to do if the content inside an element exceeds the given width and height properties.</p>

<div>
You can use the overflow property when you want to have better control of the layout. Try to change the overflow property to: visible, hidden, scroll, or inherit and see what happens. The default value is visible.
</div>
</body>

</html>
```

Mostrar solo scroll Horizontal, no vertical

"overflow-x:scroll; overflow-y:hidden;"

<http://www.brunildo.org/test/Overflowxy2.html>

```
<!DOCTYPE html>

<html>

<head>

<style>

div.scroll
{
    background-color: #00FFFF;
    width: 200px;
    height: 100px;
    overflow-x: scroll;
    overflow-y: hidden;
}

/*nowrap La diferencia con el modo normal es que no se aplican los saltos de línea a menos que se encuentren incluidos en el texto original, por lo que el contenido puede desbordar los límites del elemento */

white-space: nowrap;
```

```
border:1px solid #000000; /* Solo lo puse para que se vea el cuadro*/\n\n

\n{\nbackground-color: #00FF00;\nwidth:100px;\nheight:100px;\noverflow:hidden;\n}\n</style>\n</head>\n\n<body>\n

The overflow property specifies what to do if the content of an element exceeds the size of the element's box.

\n<p>overflow:scroll</p>\n<div class="scroll">You can use the overflow property when you want to have better control of the layout. The default value is visible.you can use the overflow property when you want to have better control of the layout. The default value is visible.you can use the overflow property when you want to have better control of the layout. The default value is visible.</div>\n\n<p>overflow:hidden</p>\n<div class="hidden">You can use the overflow property when you want to have better control of the layout. The default value is visible.</div>\n</body>\n</html>


```

#### Cambiar el cursor

En este ejemplo se muestra cómo cambiar el cursor.

Value	Description
alias	The cursor indicates an alias of something is to be created
all-scroll	The cursor indicates that something can be scrolled in any direction
auto	Default. The browser sets a cursor
cell	The cursor indicates that a cell (or set of cells) may be selected
context-menu	The cursor indicates that a context-menu is available
col-resize	The cursor indicates that the column can be resized horizontally
copy	The cursor indicates something is to be copied
crosshair	The cursor render as a crosshair
default	The default cursor
e-resize	The cursor indicates that an edge of a box is to be moved right (east)
ew-resize	Indicates a bidirectional resize cursor
grab	The cursor indicates that something can be grabbed
grabbing	The cursor indicates that something can be grabbed
help	The cursor indicates that help is available
move	The cursor indicates something is to be moved
n-resize	The cursor indicates that an edge of a box is to be moved up (north)
ne-resize	The cursor indicates that an edge of a box is to be moved up and right (north/east)
nesw-resize	Indicates a bidirectional resize cursor
ns-resize	Indicates a bidirectional resize cursor
nw-resize	The cursor indicates that an edge of a box is to be moved up and left (north/west)
nwse-	Indicates a bidirectional resize cursor

resize

no-drop	The cursor indicates that the dragged item cannot be dropped here
none	No cursor is rendered for the element
not-allowed	The cursor indicates that the requested action will not be executed
pointer	The cursor is a pointer and indicates a link
progress	The cursor indicates that the program is busy (in progress)
row-resize	The cursor indicates that the row can be resized vertically
s-resize	The cursor indicates that an edge of a box is to be moved down (south)
se-resize	The cursor indicates that an edge of a box is to be moved down and right (south/east)
sw-resize	The cursor indicates that an edge of a box is to be moved down and left (south/west)
text	The cursor indicates text that may be selected
<i>URL</i>	A comma separated list of URLs to custom cursors. <b>Note:</b> Always specify a generic cursor at the end of the list, in case none of the URL-defined cursors can be used
vertical-text	The cursor indicates vertical-text that may be selected
w-resize	The cursor indicates that an edge of a box is to be moved left (west)
wait	The cursor indicates that the program is busy
zoom-in	The cursor indicates that something can be zoomed in
zoom-out	The cursor indicates that something can be zoomed out
initial	Sets this property to its default value. <a href="#">Read about initial</a>
inherit	Inherits this property from its parent element. <a href="#">Read about inherit</a>

```
<body>
<p>Pasa el ratón por encima de los span para ver como cambia el puntero del ratón..</p>
<span style="cursor: auto">auto</span><br>
<span style="cursor: crosshair">crosshair</span><br>
<span style="cursor: default">default</span><br>
<span style="cursor: e-resize">e-resize</span><br>
<span style="cursor: help">help</span><br>
<span style="cursor: move">move</span><br>
<span style="cursor: n-resize">n-resize</span><br>
<span style="cursor: ne-resize">ne-resize</span><br>
<span style="cursor: nw-resize">nw-resize</span><br>
<span style="cursor: pointer">pointer</span><br>
<span style="cursor: progress">progress</span><br>
<span style="cursor: s-resize">s-resize</span><br>
<span style="cursor: se-resize">se-resize</span><br>
<span style="cursor: sw-resize">sw-resize</span><br>
<span style="cursor: text">text</span><br>
<span style="cursor: w-resize">w-resize</span><br>
<span style="cursor: wait">wait</span><br>
</body>
</html>
```

Cursos personalizados (creo que necesita servidor web)

<http://www.desarrolloweb.com/articulos/cursor-imagen-css.html>

Si quieres aumentar tus conocimientos a la hora de crear tus propios punteros puedes echarle un vistazo al [programa JustCursors](#).

[https://download.cnet.com/JustCursors/3000-2317\\_4-10406297.html](https://download.cnet.com/JustCursors/3000-2317_4-10406297.html)

<https://desarrolloweb.com/articulos/cursor-imagen-css.html>

```
body {
cursor:url(foo51.ani), url(dw.gif), auto;
}
```

<https://giphy.com/stickers/pointer-cursor-mouse-Eq6GFCidPULNm>

```
<!DOCTYPE html>
<html>
<head>
<style>
.alias{
    cursor: url('cursor.gif'),auto;
}
</style>
</head>
<body>
<p class="alias">alias</p>
</body>
</html>
```

## Ejemplo funcional

<https://codepen.io/CybMeta/pen/LVLoJp>

## Todas las propiedades de posición CSS

El número en el "CSS" en la columna indica qué versión de CSS se define la propiedad (o CSS1 CSS2).

Propiedad	Descripción	Valores	CSS
<u>bottom</u>	Define el borde del margen inferior de una caja posicionada	auto length % inherit	2
<u>clip</u>	Clips de un elemento con posición absoluta	shape auto inherit	2
<u>cursor</u>	Especifica el tipo de cursor que se mostrará	url auto crosshair default pointer move e-resize ne-resize nw-resize n-resize se-resize sw-resize s-resize w-resize text wait help	2
<u>left</u>	Establece el margen izquierdo de una caja posicionada	auto length % inherit	2
<u>overflow</u>	Establece lo que sucede si se desborda contenido de cuadros de un elemento	auto hidden scroll visible inherit	2
<u>position</u>	Especifica el tipo de posicionamiento para un elemento	absolute fixed relative static inherit	2
<u>right</u>	Establece el borde del margen derecho de una caja posicionada	auto length % inherit	2
<u>top</u>	Define el borde del margen superior de una caja posicionada	auto length % inherit	2
<u>z-index</u>	Establece el orden de apilamiento de un elemento	number auto inherit	2

## ¿Qué es Float en CSS?



[Eliminar imagen](#)



[Eliminar imagen](#)



[Eliminar imagen](#)



[Eliminar imagen](#)

Con floaten CSS, un elemento puede ser desplazado hacia la izquierda o hacia la derecha, permitiendo que otros elementos lo envuelvan.

Float se utiliza a menudo para las imágenes, pero también es útil cuando se trabaja con cualquier diseño.

¿Cómo hacer que los elementos de “floten”

Que los elementos estén flotando en horizontal, significa que un elemento sólo se puede flotar hacia la izquierda o la derecha, no hacia arriba o hacia abajo.

Un elemento flotante se moverá hacia la izquierda o hacia la derecha como pueda. Normalmente esto significa que el elemento se desplace completamente a la izquierda o derecha del elemento contenedor.

**Los elementos colocados después del elemento flotante fluirán alrededor del mismo envolviéndolo.**

**Los elementos antes del elemento flotante no se verán afectado.**

Si una imagen flota a la derecha, un texto siguiente la envolverá alrededor de ella, a la izquierda:

## Ejemplo

```
img  
{  
float:right;  
}
```

In the paragraph below, we have added an image with style **float:right**. The result is that the image will float to the right in the paragraph.

css