



Cascade Style Sheeting. Hoja de estilo en cascada (*).

Se usa para darle apariencia gráfica a una pág. web.

(*) Se llama en cascada porque le asigna a un tag html el estilo que tenga más cercano.

Para más info. ir a : <http://librosweb.es/css/>

Se puede usar de 3 formas:

1) En línea: <p **style="color:red;"**>

2) Forma interna embebida:

```
<head>
  <style type="text/css">
    p{ color : red; }
  </style>
</head>
```

3) Archivo Externo:

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

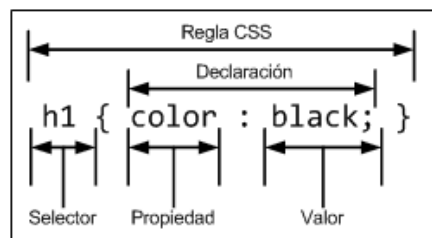
estilo.css

```
/*Esta es una hoja de estilos css*/
```

```
p{
  color : red;
  background-color : #000000;
}
```

Nota: si en una pág. tenemos las 3 formas, y aplicamos diferentes colores, el párrafo queda con el color definido en la forma en línea. Ya que los navegadores dan prioridad **1), 2), 3).**

Componentes de un estilo CSS básico:



SELECTORES:

En el ejemplo anterior se le daba estilo a todos los párrafos de la página. Para evitar esto y especificar un elemento en concreto se usan los selectores.

/*Este se aplica directamente sobre el tag body*/

```
body{  
    background-color : #A9D0F5;  
}
```

/*Sirve para los 2 tags*/

```
h1,h2{  
    color : #990000;  
    text-decoration : underline;  
}
```

/*Este es una clase, todos los tags con esa clase cogen este estilo*/

```
.azulon{  
    color : #006699;  
}
```

/*Esto es para un id, solo un tag puede tener ese id , por lo tanto afecta a 1 solo elemento*/

```
#parrafoVerde{  
    color : #093;  
}
```

/*Solo afecta a los li, que estén dentro de un ul, que a su vez estén dentro de un tag nav*/

```
nav ul li{  
    color : #333;  
}
```

Para que las clases y los id sirvan, es necesario asignarlos como atributos dentro de los tag html:

```
<p class="azulon">Texto de color azul.</p>
```

```
<div class="azulon"></div>
```

```
<p id="parrafoVerde">Texto de color verde.</p>
```

Otros ejemplos:

/* Todos los párrafos con atributo class="aviso" */

```
p.aviso { ... }
```

/* Todos los elementos con atributo class="aviso" que estén dentro de cualquier párrafo */

```
p .aviso { ... }
```

/* Todos los párrafos y todos los elementos con atributo class="aviso" */

```
p, .aviso { ... }
```

➔ Aplicar varias clases de estilos a un elemento:

```
<p class="clase1 clase2 clase3">Párrafo de texto...</p>
```

```
.clase1 { color: red; }
```

```
.clase2 { font-size: 15px; }
```

```
.clase3 { font-weight: bold; }
```

➔ Selector de clase avanzado:

```
<p class="clase1 clase2 clase3">Párrafo de texto...</p>
```

```
.clase3 { color: red; }
```

```
.clase3.clase2 { color: blue; }
```

```
.clase2 { font-size: 15px; }
```

```
.clase1 { font-weight: bold; }
```

En este caso el color de la letra del texto es azul y no rojo. Porque se usó el **selector de clase múltiple: .clase3.clase2**, que se interpreta como "aquellos

elementos de la página que dispongan de un atributo class con al menos los valores clase3 y clase2".

SELECTORES AVANZADOS:

Selector de hijo:

/*Selecciona cualquier que sea hijo directo de un parrafo*/

```
p > span { color: blue; }
```

```
<p><span>Texto1</span></p>
```

```
<p><a href="#"><span>Texto2</span></a></p>
```

Selector Adyacente:

/*Selecciona todos los h2 que se encuentran inmediatamente después de un h1 (que sea hermano del h1)*/

```
h2 { color: green; }
```

```
h1 + h2 { color: red }
```

```
<h1>Titulo1</h1>
```

```
<h2>Subtítulo</h2>...
```

```
<h2>Otro subtítulo</h2>...
```

Nota: en este caso, todos los h2 salen de color verde, y el h2 que se encuentra justo después de un h1 sale de color rojo.

Selector de Atributos:

/* 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 apunten al sitio "http://www.ejemplo.com" */

```
a[href="http://www.ejemplo.com"] { color: blue; }
```

/* 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; }
```

PSEUDO ELEMENTOS:



Permiten aplicar estilos a ciertas partes de la página.

Ej:

```
ul li{  
    list-style-type:none;  
    font-size:50px;  
    font-weight:bold;  
    color:#069;  
}
```



/*inserta antes del elemento una imagen*/

```
ul li::before{  
    content:url(imagen.png);  
}
```

<ul style="list-style-type: none">• Lista Elemento 1.• Lista Elemento 2.	<ul style="list-style-type: none">•  Lista Elemento 1.•  Lista Elemento 2.
---	---

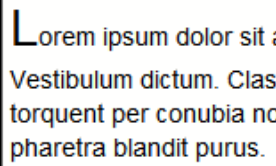
/*inserta una imagen después del elemento li*/

```
ul li::after{  
    content:url(imagen.png);  
}
```

<ul style="list-style-type: none">• Lista Elemento 1.• Lista Elemento 2.	 
---	--

/*Cambia la primera letra de los párrafos, es muy útil para letras Capitales, como en los periódicos*/

```
p::first-letter{
    font-size:36px;
    font-weight:bold;
    color:#FF0000;
}
```



/*Cambia solo la primera línea del párrafo*/

```
p::first-line{
    color:#F00;
    text-decoration:underline;
}
```

PSEUDO CLASES:

Sirve para aplicar diferentes estilos a un elemento, que serán visibles de acuerdo a su estado.

Ej:

```
a{
    color:#333;
    text-decoration:none;
}
```

/*Cambia el color del enlace cuando no se le ha hecho clic (link sin visitar)*/

```
a:link{
    color:#900;
}
```

/*Cambia el color del enlace cuando ya se ha hecho clic (link visitado)*/

```
a:visited{
    color:#003399;
}
```

/*Cambia el color del fondo del enlace, cuando se pasa el mouse por encima (Mouse sobre el link)*/

```
a:hover{
    background-color:#CCC;
}
```

/*Cuando se le hace clic al link, cambia el color de fondo en ese momento que se tiene el botón presionado*/

```
a:active{
    background-color:#222;
}
```


Nota: Las pseudo-clases **:link** y **:visited** solamente están definidas para los enlaces, pero las pseudo-clases **:hover** y **:active** se definen para todos los elementos HTML.

También es posible combinar en un mismo elemento las pseudo-clases que son compatibles entre sí:

```
/* Los estilos se aplican cuando el usuario pasa el ratón por encima de un
   enlace que todavía no ha visitado */
```

```
    a:link:hover { ... }
```

```
/* Los estilos se aplican cuando el usuario pasa el ratón por encima de un
   enlace que ha visitado previamente */
```

```
    a:visited:hover { ... }
```

Para más info. ir a: Cap.7-Enlaces http://librosweb.es/css/capitulo_7.html

/*Cambia el color de fondo y de la letra de un input cuando se le hace clic para escribir en él*/

```
input:focus{
    background-color:#000;
    color:#FFF;
}
```

/*Realiza el efecto o cambio de estilo cuando se pasa el mouse encima*/

```
input:hover{ ... }
```

/*Cambia el color de fondo al primer hijo de una lista, en este caso el primer li*/

```
ul li:first-child{
    background-color:#090;
}
```

/*El ultimo hijo de la lista ul, es decir el ultimo li*/

```
ul li:last-child{ ... }
```

/*El número entre paréntesis, indica que hijo cambiar, en este caso cambia el segundo li de la lista ul*/

```
ul li:nth-child(2){ ... }
```

Nota: para conocer más pseudo-elementos y pseudo-clases ir a:

http://www.w3schools.com/cssref/css_selectors.asp

HERENCIA DE ESTILOS:

Cuando se establece el valor de una propiedad CSS en un elemento, sus elementos descendientes (hijos) heredan de forma automática el valor de esa propiedad.

Ej: Todos los hijos del body (en este caso el h1 y el párrafo), heredan el color azul de texto del body.

```
<style type="text/css">
    body { color: blue; }
</style>
```

```
<body>
    <h1>Titular de la página</h1>
    <p>Un párrafo de texto no muy largo.</p>
</body>
```

Para anular el efecto de la herencia, se definen estilos de forma explícita:

```
<style type="text/css">
    body { font-family: Arial; color: black; }
    h1 { font-family: Verdana; }
    p { color: red; }
</style>
```

En este caso, el h1 hereda el color de texto negro, pero no el tipo de fuente, y el párrafo hereda el tipo de fuente Arial, pero no el color negro del texto.

COLISIÓN DE ESTILOS:

Cuando un mismo elemento tiene varios estilos (reglas), y esos estilos hacen referencia a una misma propiedad (por ejemplo se define 2 veces el color, uno rojo y otro azul), se produce una colisión.

Esas colisiones se solucionan de acuerdo a las siguientes reglas:

1. Cuanto más específico sea un selector, más importancia tiene su regla asociada.
2. A igual especificidad, se considera la última regla indicada.

Ej1: Se muestra el texto del párrafo de color azul.

```
p { color: red; }
```

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

Ej2: se muestra el texto del párrafo de color verde.

```
p { color: red; }
```

```
p#especial { color: green; }
```

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

```
<p id="especial">...</p>
```

UNIDADES DE MEDIDA:

Unidades Absolutas:

in → pulgadas ("inches", en inglés). Una pulgada equivale a 2.54 centímetros.

cm → centímetros.

mm → milímetros.

pt → puntos. Un punto equivale a 1 pulgada/72, es decir, unos 0.35 milímetros.

pc → picas. Una pica equivale a 12 puntos, es decir, unos 4.23 milímetros.

Ej:

/* El cuerpo de la página debe mostrar un margen de media pulgada */

```
body { margin: 0.5in; }
```

/* Los elementos <h1> deben mostrar un interlineado de 2 centímetros */

```
h1 { line-height: 2cm; }
```

/* Las palabras de todos los párrafos deben estar separadas 4 milímetros entre si */

```
p { word-spacing: 4mm; }
```

/* Los enlaces se deben mostrar con un tamaño de letra de 12 puntos */

```
a { font-size: 12pt }
```

/* Los elementos deben tener un tamaño de letra de 1 pica */

```
span { font-size: 1pc }
```

Unidades Relativas:

em → relativa respecto del tamaño de letra del elemento.

1em equivale a la anchura de la letra M ("eme mayúscula") del tipo y tamaño de letra del elemento.

Si se utiliza una tipografía de 12 puntos, 1em equivale a 12 puntos.

ex → relativa respecto de la altura de la letra x ("equis minúscula") del tipo y tamaño de letra del elemento.

1ex se puede aproximar por 0.5 em.

px → (píxel) relativa respecto de la resolución de la pantalla del dispositivo en el que se visualiza la página HTML.

Ej1:

```
p { margin: 1em; }
```

Indica que los párrafos deben mostrar un margen de anchura igual a 1em. Como es una unidad de medida relativa, es necesario realizar un cálculo matemático para determinar la anchura real de ese margen.

Todos los navegadores muestran por defecto el texto de los párrafos con un tamaño de letra de 16 píxeles.

Por tanto, en este caso el margen de 1em equivale a un margen de anchura 16px.

Ej2:

```
p { font-size: 32px; margin: 1em; }
```

El valor real de las márgenes es diferente.

En este caso, el margen tendrá una anchura de 32px, ya que 1em siempre equivale al tamaño de letra del elemento.

Ej3: Si se quiere reducir la anchura del margen a 16px pero manteniendo el tamaño de letra de los párrafos en 32px, se debe utilizar la siguiente regla CSS:

```
p { font-size: 32px; margin: 0.5em; }
```

El valor 0.5em se interpreta como "la mitad del tamaño de letra del elemento", ya que se debe multiplicar por 0.5 su tamaño de letra ($32\text{px} \times 0.5 = 16\text{px}$).

De la misma forma, si se quiere mostrar un margen de 8px de anchura, se debería utilizar el valor 0.25em, ya que $32\text{px} \times 0.25 = 8\text{px}$.

Establecer el margen de un elemento con el valor 1em equivale a indicar que "el margen del elemento debe ser del mismo tamaño que su letra y debe cambiar proporcionalmente".

Ej4: El funcionamiento de la unidad ex es idéntico a em, salvo que en este caso, la referencia es la altura de la letra x minúscula, por lo que su valor es aproximadamente la mitad que el de la unidad em.

Ej5: Las unidades de medida se pueden mezclar en los diferentes elementos de una misma página, como en el siguiente ejemplo:

```
body { font-size: 10px; }
```

```
h1 { font-size: 2.5em; }
```

Se establece un tamaño de letra de 10 píxeles para toda la página.

Se asigna un tamaño de 2.5em al elemento <h1>, por lo que su tamaño de letra real será de $2.5 \times 10\text{px} = 25\text{px}$.

Ej6: Herencia después de calcular el valor de una medida relativa:

```
body { font-size: 12px;
```

```
    text-indent: 3em; }
```

```
h1 { font-size: 15px }
```

La propiedad text-indent, se utiliza para tabular la primera línea de un texto.

El elemento <body> define un valor para esta propiedad, pero el elemento <h1> no lo hace, por lo que heredará el valor de su elemento padre.

Sin embargo, el valor heredado no es 3em, sino 36px.

Primero se calcula el valor real de 3em para el elemento <body>: $3em \times 12px = 36px$.

Una vez calculado el valor real, este es el valor que se hereda para el resto de elementos.

Porcentajes:

También es una unidad relativa. Los porcentajes se pueden utilizar por ejemplo para establecer el valor del tamaño de letra de los elementos:

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

```
h1 { font-size: 200%; } → equivale a 2em
```

```
h2 { font-size: 150%; } →equivale a 1.5em
```

Por lo que es más habitual definirlos mediante em.

Ej: Los porcentajes también se utilizan para establecer la anchura de los elementos:

```
div#contenido { width: 600px; }
```

```
div.principal { width: 80%; }
```

El valor 80% es la anchura de su elemento padre. Por lo tanto, el elemento <div> cuyo atributo class es principal tiene una anchura de $(80\% \times 600px)/100\% = 480px$.

Nota: Se deben usar medidas relativas para que las páginas se adapten a cualquier medio o dispositivo.

- Se usa **em**, para indicar el tamaño del texto y para todas las medidas que sean posibles.
- Se utilizan **píxeles** y **porcentajes**, para definir el layout del documento (la anchura de las columnas y de los elementos de las páginas).
- Y **em** y **porcentajes**, para el tamaño de letra de los textos.

Points	Pixels	Ems	Percent
6pt	8px	0.5em	50%
7pt	9px	0.55em	55%
7.5pt	10px	0.625em	62.5%
8pt	11px	0.7em	70%
9pt	12px	0.75em	75%
10pt	13px	0.8em	80%
10.5pt	14px	0.875em	87.5%
11pt	15px	0.95em	95%
12pt	16px	1em	100%
13pt	17px	1.05em	105%
13.5pt	18px	1.125em	112.5%
14pt	19px	1.2em	120%
14.5pt	20px	1.25em	125%
15pt	21px	1.3em	130%
16pt	22px	1.4em	140%
17pt	23px	1.45em	145%
18pt	24px	1.5em	150%
20pt	26px	1.6em	160%
22pt	29px	1.8em	180%
24pt	32px	2em	200%
26pt	35px	2.2em	220%
27pt	36px	2.25em	225%
28pt	37px	2.3em	230%
29pt	38px	2.35em	235%
30pt	40px	2.45em	245%
32pt	42px	2.55em	255%
34pt	45px	2.75em	275%

COLORES:

Blanco = #FFFFFF

Amarillo = #FFFF00

Rojo = #FF0000

Azul = #0000FF

Verde = #00FF00

Negro = #000000

Cyan = #00FFFF

Gris = #C0C0C0

Turquesa = #7093DB

Purpura = #871F78

Naranja = #7F7F00

Magenta = #FF00FF

Marrón = #A62A2A

Azul Cielo = #3299CC

Gris Oscuro = #2F4F4F

Verde Oscuro = #2F4F2F

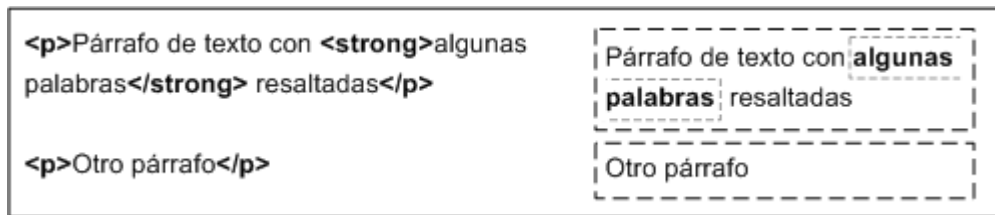
Nota: Para más info. ver: http://en.wikipedia.org/wiki/Web_colors#Web-safe_colors

<http://html-color-codes.info/codigos-de-colores-hexadecimales/>

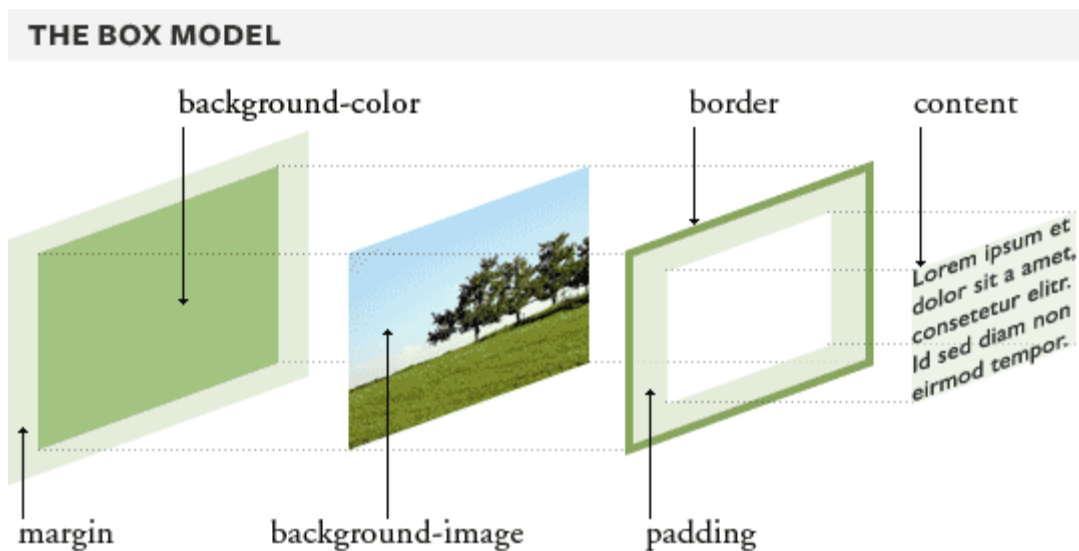
MODELO DE CAJAS:

Es el comportamiento de CSS que hace que todos los elementos de las páginas se representen mediante cajas rectangulares.

Los navegadores crean y colocan las cajas de forma automática, cada vez que se inserta una etiqueta HTML. Las cajas de las páginas no son visibles porque inicialmente no muestran ningún color de fondo ni ningún borde.



CSS permite modificar todas las características de las cajas. Cada una de las cajas está formada por seis partes, tal y como muestra la siguiente imagen:



El “padding” y el “margin” son transparentes.

Las partes que componen cada caja y su orden de visualización desde el punto de vista del usuario son las siguientes:

Contenido (content): el contenido HTML del elemento (las palabras de un párrafo, una imagen, el texto de una lista de elementos, etc.)

Relleno (padding): espacio libre opcional existente entre el contenido y el borde.

Borde (border): línea que encierra completamente el contenido y su relleno.

Imagen de fondo (background-image): imagen que se muestra por detrás del contenido y el espacio de relleno.

Color de fondo (background-color): color que se muestra por detrás del contenido y el espacio de relleno.

Margen (margin): separación opcional existente entre la caja y el resto de cajas adyacentes.

El relleno y la margen son transparentes, por lo que en el espacio ocupado por el relleno se muestra el color o imagen de fondo (si están definidos).

En el espacio ocupado por el margen se muestra el color o imagen de fondo de su elemento padre (si están definidos).

Si ningún elemento padre tiene definido un color o imagen de fondo, se muestra el color o imagen de fondo de la propia página (si están definidos).

Si una caja define tanto un color como una imagen de fondo, la imagen tiene más prioridad y es la que se visualiza.

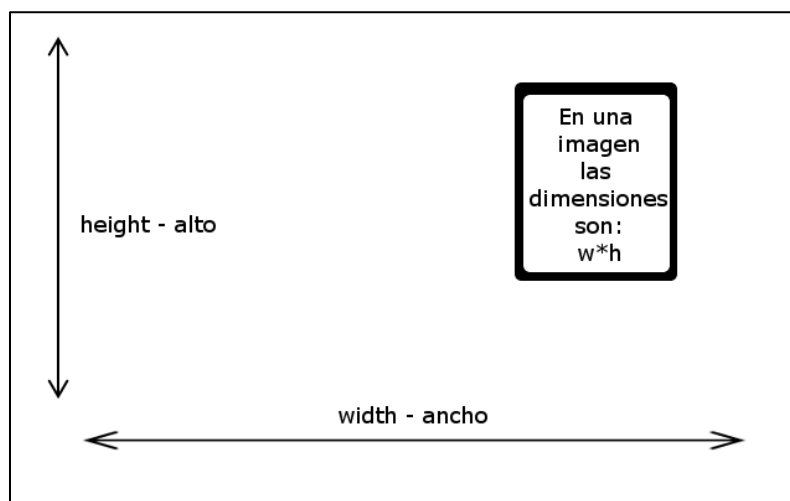
Si la imagen de fondo no cubre totalmente la caja del elemento o si la imagen tiene zonas transparentes, también se visualiza el color de fondo.

Combinando imágenes transparentes y colores de fondo se pueden lograr efectos gráficos muy interesantes.

Anchura y Altura:

Width → controla la anchura de la caja de los elementos.

Height → Altura.



Width y Height, tienen las siguientes características en común:

Valores: unidad de medida | porcentaje | auto | inherit.

Valor por defecto: auto.

Se aplican a: Todos los elementos, salvo los elementos en línea que no sean imágenes, las filas de tabla y los grupos de filas de tabla.

- Width y Height no admiten valores negativos.
- Si se indica un porcentaje, se calculan a partir de la anchura/altura de su elemento padre.
- El valor inherit indica que la anchura/altura del elemento se hereda de su elemento padre. Si el elemento padre no tiene una anchura/altura definida explícitamente, se asigna el valor auto
- El valor auto, indica que el navegador debe calcular automáticamente la anchura/altura del elemento, teniendo en cuenta sus contenidos y el sitio disponible en la página.

Ej:

```
#tamaño { width: 50%; height:200px; border:solid; }
```

```
<div id="tamaño">...
```

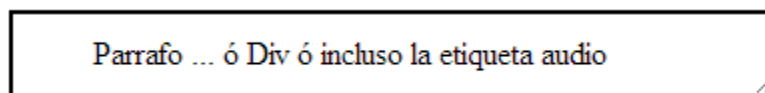
```
</div>
```

Al poner porcentaje (%), tiene la ventaja que se adapta al tamaño de la ventana del navegador.

El tamaño también se puede definir en píxeles (px) y en centímetros (cm).

Con elementos como canvas o video, es mejor no usar el tamaño en css, ya que lo que haría es estirar el contenido.

Para poder modificar rectángulos con contenido como:

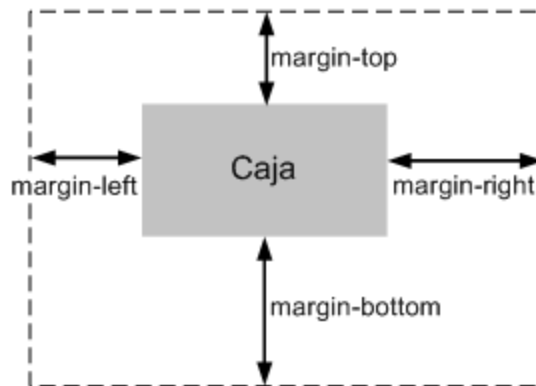


Que al hacer clic en la esquina y arrastrar, se modifica el tamaño, se usa la propiedad resize:

```
#tamaño { width: 30px; height:30px; resize:both; border:solid; }
```

Margen:

Establece la separación entre el borde lateral de la caja y el resto de cajas adyacentes.



Las unidades más utilizadas para indicar los márgenes de un elemento son los píxeles (cuando se requiere una precisión total), los em (para hacer diseños que mantengan las proporciones) y los porcentajes (para hacer diseños líquidos o fluidos).

Los márgenes verticales (**margin-top** y **margin-bottom**) sólo se pueden aplicar a los elementos de bloque (ej: párrafos) y las imágenes.

Los márgenes laterales (**margin-left** y **margin-right**) se pueden aplicar a cualquier elemento (ej: párrafos → elementos de bloque Y enlaces → elementos en línea).

CSS define una propiedad especial (shorthand) que permite establecer los cuatro márgenes al mismo tiempo (**margin**).

margin → admite entre uno y cuatro valores, con el siguiente significado:

1 valor, todos los márgenes tienen ese valor.

2 valores, el primero se asigna al margen superior e inferior y el segundo se asigna a los márgenes izquierdo y derecho.

3 valores, el primero se asigna al margen superior, el tercero se asigna al margen inferior y el segundo valor se asigna los márgenes izquierdo y derecho.

4 valores, el orden de asignación es: margen superior, margen derecho, margen inferior y margen izquierdo.

Ej:

```
div img {  
    margin-top: .5em;  
    margin-bottom: .5em;  
    margin-left: 1em;  
    margin-right: .5em;  
}
```

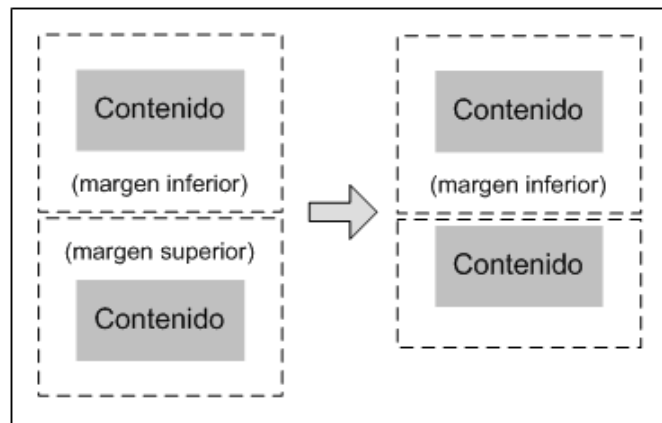
Alternativa directa:

```
div img {  
    margin: .5em .5em .5em 1em;  
}
```

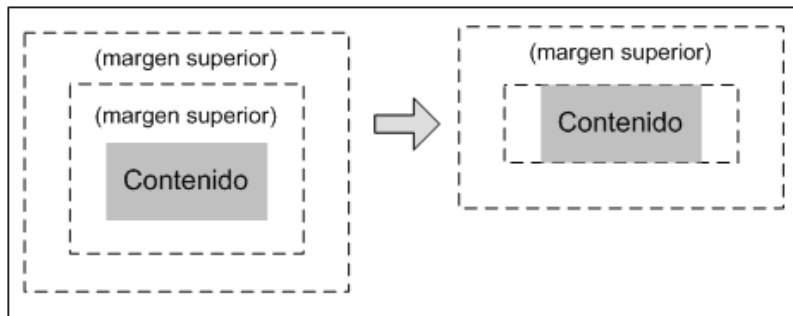
Otra alternativa:

```
div img {  
    margin: .5em;  
    margin-left: 1em;  
}
```

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.



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.



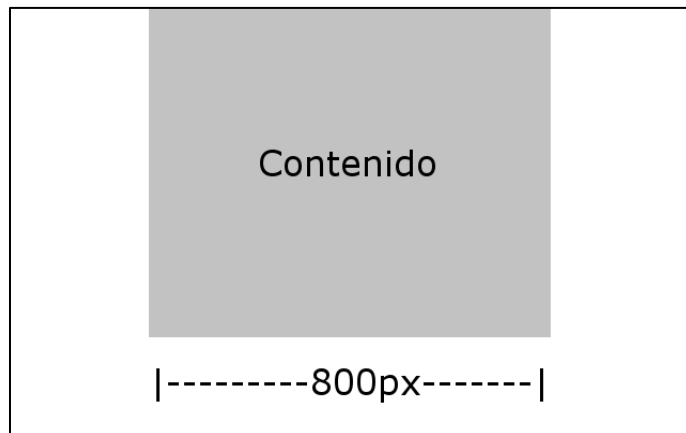
En el caso de un elemento que se encuentra en el interior de otro y sus márgenes se fusionan de forma automática, se puede evitar este comportamiento añadiendo un pequeño relleno (`padding: 1px`) o un borde (`border: 1px solid transparent`) al elemento contenedor.

Ej: En este ejemplo, centramos el contenido al centro del navegador, con “margin: 0 auto;” y no importa si el usuario agranda o achica el navegador, siempre se va a ver centrado.

```
body{
    margin-top:0px;
    margin-right:0px;
    margin-bottom:0px;
    margin-left:0px;
}

#contenido{
    width:800px;
    height:300px;
    margin:0 auto;
    background-color:# c2c2c2;
}
```

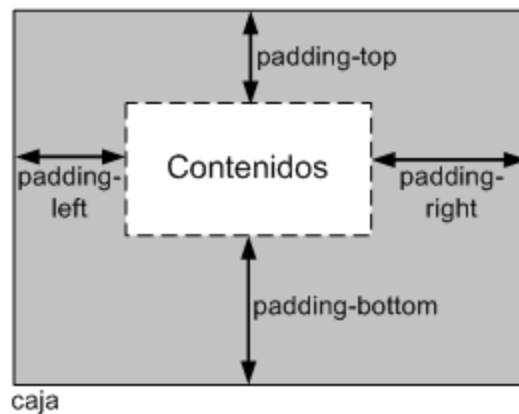
El navegador se ve así:



Si se achica el navegador, aparecen las barras de scroll o navegación, ya que el contenido se definió con un ancho de 800px.

Relleno:

Establece la separación entre el contenido y los bordes laterales de la caja del elemento.



padding → propiedad shorthand para definir el relleno.

Ej:

`/* Todos los rellenos valen 2em */`

```
body {padding: 2em}
```

```
/* Superior e inferior = 1em, Izquierdo y derecho = 2em */
```

```
body {padding: 1em 2em}
```

```
/* Superior = 1em, derecho = 2em, inferior = 3em, izquierdo = 2em */
```

```
body {padding: 1em 2em 3em}
```

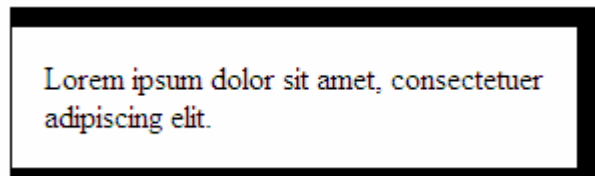
```
/* Superior = 1em, derecho = 2em, inferior = 3em, izquierdo = 4em */
```

```
body {padding: 1em 2em 3em 4em}
```

BORDES:

CSS permite modificar el aspecto de cada uno de los cuatro bordes de la caja de un elemento. Para cada borde se puede establecer su anchura o grosor, su color y su estilo, por lo que en total CSS define 20 propiedades relacionadas con los bordes.

- **Anchura:**



```
div {  
    border-top-width: 10px;  
    border-right-width: 1em;  
    border-bottom-width: thick;  
    border-left-width: thin;  
}
```

Los valores para las 4 propiedades pueden ser:

thin (borde delgado), medium (borde normal), thick (borde ancho), inherit, o una unidad de medida relativa o absoluta, se suele usar **px**, ya que las medidas como por ejemplo “medium”, tiene diferentes valores en cada navegador.

border-width → propiedad shorthand para definir los 4 bordes según la cantidad de valores indicados.

/ superior derecho inferior izquierdo */*



```
p { border-width: thin }
```

/ **thin thin thin thin** */*

```
p { border-width: thin thick }
```

/ **thin** thick **thin** thick */*

```
p { border-width: thin thick medium }
```

/ thin **thick** medium **thick** */*

```
p { border-width: thin thick medium thin } /* thin thick medium thin */
```

- **Color:** se puede definir un color distinto para cada borde.

```
div {
```

```
    border-top-color: #CC0000;
```

```
    border-right-color: blue;
```

```
    border-bottom-color: #00FF00;
```

```
    border-left-color: #CCC;
```

```
}
```

border-color → propiedad shorthand para definir el color de los 4 bordes según la cantidad de valores. Se usa igual que border-width.

- **Estilo:**

```
div {
```

```
    border-top-style: dashed;
```

```
    border-right-style: double;
```

```
    border-bottom-style: dotted;
```

```
    border-left-style: solid;
```

```
}
```

Los tipos de borde pueden ser:

None

Hidden

Dotted

Dashed

Solid

Double

Groove

Ridge

Inset

Outset

border-style → propiedad shorthand para definir el estilo de los 4 bordes, según la cantidad de valores, se usa igual que border-width.

- **Propiedades Shorthand:** se usan para definir la anchura, el color, y el estilo de cada borde al mismo tiempo.

```
div {  
  border-top: 1px solid #369;  
  border-right: solid red; /*en este caso la anchura es por defecto medium*/  
  border-bottom: 3px double #369;  
  border-left: solid;  
}
```

border → propiedad shorthand global, establece el valor de todos los atributos de todos los bordes de forma directa.

```
div {  
    border-top: 1px solid red;  
    border-right: 1px solid red;  
    border-bottom: 1px solid red;  
    border-left: 1px solid red;  
}
```

Es equivalente a:

```
div { border: 1px solid red; }
```

Como el valor por defecto de la propiedad border-style es none, si una propiedad shorthand no establece explícitamente el estilo de un borde, el elemento no muestra ese borde:

/* Sólo se establece el color, por lo que el estilo es "none" y el borde no se muestra */

```
div { border: red; }
```

/* Se establece el grosor y el color del borde, pero no su estilo, por lo que es "none" y el borde no se muestra */

```
div { border-bottom: 5px blue; }
```

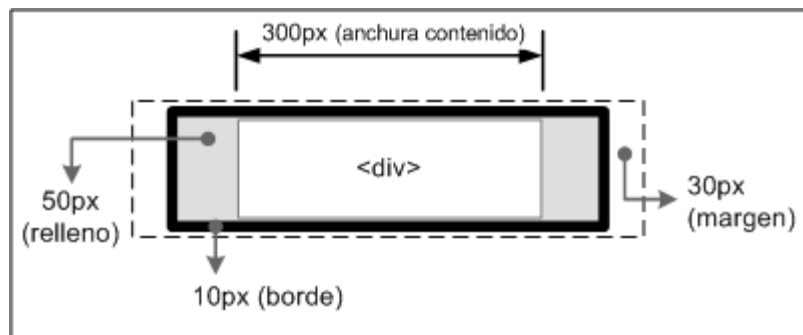
Cuando los cuatro bordes no son idénticos pero sí muy parecidos, se puede utilizar la propiedad border para establecer de forma directa los atributos comunes de todos los bordes y posteriormente especificar para cada uno de los cuatro bordes sus propiedades particulares:

```
h1 {  
    border: solid #000;  
    border-top-width: 6px;  
    border-left-width: 8px;  
}
```

NOTA: La anchura y altura de un elemento no solamente se calculan teniendo en cuenta sus propiedades width y height. El margen, el relleno y los bordes establecidos a un elemento determinan la anchura y altura final del elemento.

```
div {  
    width: 300px;  
    padding-left: 50px;  
    padding-right: 50px;  
    margin-left: 30px;  
    margin-right: 30px;  
    border: 10px solid black;  
}
```

La anchura total con la que se muestra el elemento no son los 300 píxel indicados en la propiedad width, sino que también se añaden todos sus márgenes, rellenos y bordes:



La anchura del elemento en pantalla sería igual a la suma de la anchura original, los márgenes, los bordes y los rellenos:

$$30px + 10px + 50px + 300px + 50px + 10px + 30px = 480 \text{ píxel}$$

La anchura/altura establecida con CSS siempre hace referencia a la anchura/altura del contenido.

FONDOS:

El fondo puede ser un color simple o una imagen. El fondo solamente se visualiza en el área ocupada por el contenido y su relleno.

Para establecer un color o imagen de fondo en la página entera, se debe establecer un fondo al elemento <body>.

Si se establece un fondo a la página, como el valor inicial del fondo de los elementos es transparente, todos los elementos de la página se visualizan con el mismo fondo a menos que algún elemento especifique su propio fondo.

Para establecer el fondo de cada elemento se usan 5 propiedades:

background-color → Establece un color de fondo para los elementos, su valor por defecto es transparent.

```
/*página web con un color gris claro de fondo*/  
  
body {  
    background-color: #F5F5F5;  
}
```

background-image → Establece una imagen como fondo para los elementos, su valor por defecto es none.

```
body { background-image: url("imagenes/fondo.png") }
```

CSS permite establecer de forma simultánea un color y una imagen de fondo. En este caso, la imagen se muestra delante del color, por lo que solamente si la imagen contiene zonas transparentes es posible ver el color de fondo.

Si la imagen que se quiere mostrar es demasiado grande para el fondo del elemento, solamente se muestra la parte de imagen comprendida en el tamaño del elemento.

Si la imagen es más pequeña que el elemento, CSS la repite horizontal y verticalmente hasta llenar el fondo del elemento.

Esto es muy útil, ya que se ahorra ancho de banda al descargar la imagen pequeña y la pág. carga más rápido.



→ En este caso se repetiría en todo el body el fondo, creando un fondo rayado.

background-repeat → Controla la forma en la que se repiten las imágenes de fondo, su valor por defecto es repeat.

Sus posibles valores son:

repeat → la imagen se debe repetir en todas direcciones.

no-repeat → muestra una sola vez la imagen y no se repite en ninguna dirección.

repeat-x → repite la imagen sólo horizontalmente.

repeat-y → repite la imagen solamente de forma vertical.

background-position → Controla la posición en la que se muestra la imagen en el fondo del elemento. Su valor por defecto es 0% 0%.

Si se indican dos porcentajes o dos medidas, el primero indica el desplazamiento horizontal y el segundo el desplazamiento vertical respecto del origen (situado en la esquina superior izquierda).

Si solamente se indica un porcentaje o una medida, se considera que es el desplazamiento horizontal y al desplazamiento vertical se le asigna automáticamente el valor de 50%.

Si el valor de la propiedad background-position se indica mediante dos porcentajes x% y%, el navegador coloca el punto (x%, y%) de la imagen de fondo en el punto (x%, y%) del elemento.

Las palabras clave permitidas son equivalentes a algunos porcentajes significativos: top = 0%, left = 0%, center = 50%, bottom = 100%, right = 100%.

CSS permite mezclar porcentajes y palabras clave, como por ejemplo 50% 2cm, center 2cm, center 10%.

Si se utilizan solamente palabras clave, el orden es indiferente y por tanto, es equivalente indicar top left y left top.

Ej:

```
#caja1 {  
    background-image: url("images/help.png");  
    background-repeat: no-repeat;  
    background-position: bottom left;  
}
```

```
#caja2 {  
    background-image: url("images/help.png");  
    background-repeat: no-repeat;  
    background-position: right top;  
}
```

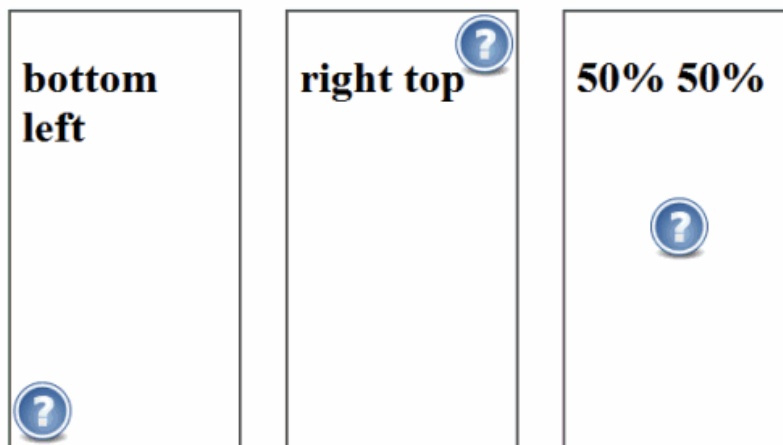
```
#caja3 {  
    background-image: url("images/help.png");  
    background-repeat: no-repeat;  
    background-position: 50% 50%;  
}
```

```
<div id="caja1"><h1>bottom left</h1></div>
```

```
<div id="caja2"><h1>right top</h1></div>
```

```
<div id="caja3"><h1>50% 50%</h1></div>
```

En el navegador muestra:



background-attachment → Controla la forma en la que se visualiza la imagen de fondo: permanece fija cuando se hace scroll en la ventana del navegador o se desplaza junto con la ventana. Su valor por defecto es scroll.

Para hacer que una imagen de fondo se muestre fija al desplazar la ventana del navegador, se usa:

```
background-attachment: fixed
```

background → Propiedad shorthand, para indicar en una sola línea el color, la url de imagen, la repetición y la posición del fondo.

```
body {  
    background-color: #222d2d;  
    background-image: url("../graphics/colorstrip.gif");  
    background-repeat: repeat-x;  
    background-position: 0 0;  
}
```

Es equivalente a:

```
body { background: #222d2d url("../graphics/colorstrip.gif") repeat-x 0 0; }
```

Background permite asignar todos o sólo algunos de todos los valores que se pueden definir para los fondos de los elementos:

```
background: url("../graphics/wide/bg-content-secondary.gif") repeat-y;
```

```
background: url("../graphics/wide/footer-content-secondary.gif") no-repeat  
bottom left;
```

```
background: transparent url("../graphics/navigation.gif") no-repeat 0 -27px;
```

```
background: none;
```

```
background: #293838 url("../graphics/icons/icon-permalink-big.gif") no-repeat  
center left;
```

Mini Resumen Backgrounds:

Como fondo es mejor usar una textura (o imagen pequeña) y repetirla, que una imagen muy grande (como por ej. un wallpaper). Esto por el tiempo de carga.

```
body{  
    background-color : #033;  
↓
```

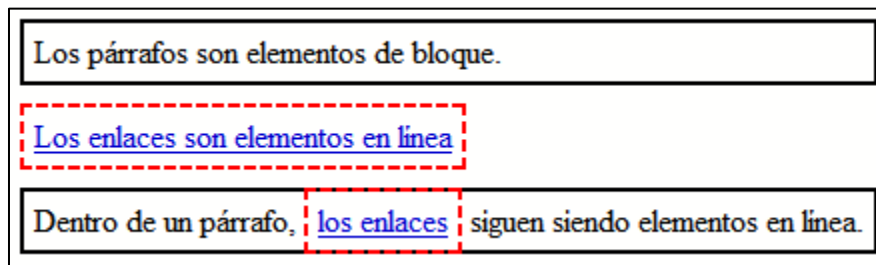
↑
background-image : url(imgs/textura.jpg);
background-repeat : [repeat/repeat-x/repeat-y] → cualquier valor
}

/*Con el siguiente código, podemos tener múltiples fondos, en este caso "linux.png" es una silueta que aparece encima del otro fondo*/

```
body{  
    background-image:url(imgs/linux.png),url(imgs/textura.jpg);  
    background-repeat:no-repeat,repeat;  
    background-position:center,left top;  
    background-size:20% 25%;  
}
```

TIPOS DE ELEMENTOS:

Existen dos tipos de elementos: los elementos de bloque, que siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea. Y los elementos en línea, que no empiezan necesariamente en nueva línea y sólo ocupan el espacio necesario para mostrar sus contenidos.



Los elementos de bloque no pueden insertarse dentro de elementos en línea y sólo pueden aparecer dentro de otros elementos de bloque.

Un elemento en línea puede aparecer tanto dentro de un elemento de bloque como dentro de otro elemento en línea.

Los elementos en línea definidos por HTML son: a, abbr, acronym, b, basefont, bdo, big, br, cite, code, dfn, em, font, i, img, input, kbd, label, q, s, samp, select, small, span, strike, strong, sub, sup, textarea, tt, u, var.

Los elementos de bloque definidos por HTML son: address, blockquote, center, dir, div, dl, fieldset, form, h1, h2, h3, h4, h5, h6, hr, isindex, menu, noframes, noscript, ol, p, pre, table, ul.

Los siguientes elementos también se considera que son de bloque: dd, dt, frameset, li, tbody, td, tfoot, th, thead, tr.

Los siguientes elementos pueden ser en línea y de bloque según las circunstancias: button, del, iframe, ins, map, object, script.

POSICIÓN:

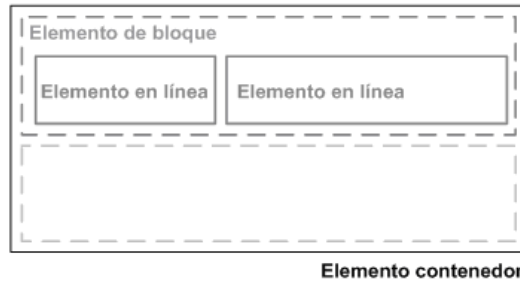
El estándar de CSS define cinco modelos diferentes para posicionar una caja:

- Posicionamiento normal o estático: se trata del posicionamiento que utilizan los navegadores si no se indica lo contrario.
- Posicionamiento relativo: variante del posicionamiento normal que consiste en posicionar una caja según el posicionamiento normal y después desplazarla respecto de su posición original.
- Posicionamiento absoluto: la posición de una caja se establece de forma absoluta respecto de su elemento contenedor y el resto de elementos de la página ignoran la nueva posición del elemento.
- Posicionamiento fijo: variante del posicionamiento absoluto que convierte una caja en un elemento inamovible, de forma que su posición en la pantalla siempre es la misma independientemente del resto de elementos e independientemente de si el usuario sube o baja la página en la ventana del navegador.
- Posicionamiento flotante: se trata del modelo más especial de posicionamiento, ya que desplaza las cajas todo lo posible hacia la izquierda o hacia la derecha de la línea en la que se encuentran.

position → Selecciona el posicionamiento con el que se mostrará el elemento. Su valor por defecto es static, y puede tener los siguientes valores:

- **static:** corresponde al posicionamiento normal o estático. Si se utiliza este valor, se ignoran los valores de las propiedades top, right, bottom y left.

En este posicionamiento normal, sólo se tiene en cuenta si el elemento es de bloque o en línea, sus propiedades width y height y su contenido.



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.

- **relative:** corresponde al posicionamiento relativo. El desplazamiento de la caja se controla con las propiedades top, right, bottom y left.

top, right, bottom, left → Indican el desplazamiento horizontal y vertical del elemento respecto de su posición original.

top → se emplea para mover las cajas de forma descendente.

bottom → mueve las cajas de forma ascendente.

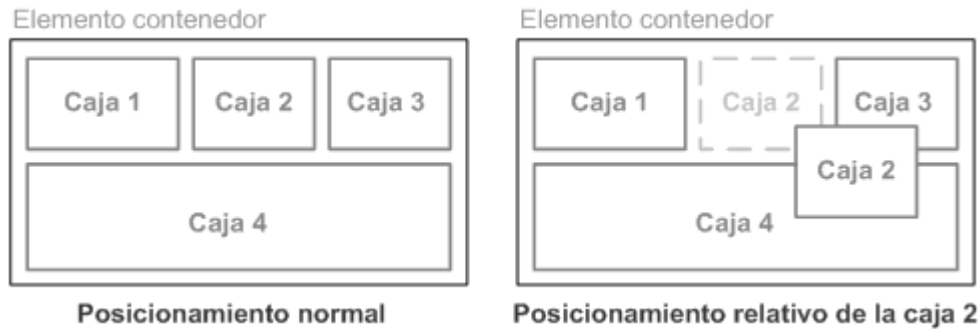
left → se utiliza para desplazar las cajas hacia la derecha.

right → mueve las cajas hacia la izquierda.

Si se utilizan valores negativos en las propiedades top, right, bottom y left, su efecto es el inverso.

Si tanto left como right tienen un valor de auto (valor por defecto) la caja no se mueve de su posición original. Si tanto left como right tienen valores distintos de auto, uno de los dos valores se tiene que ignorar porque son mutuamente excluyentes.

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.



- **absolute:** corresponde al posicionamiento absoluto. El desplazamiento de la caja también se controla con las propiedades top, right, bottom y left, pero su interpretación es mucho más compleja, ya que el origen de coordenadas del desplazamiento depende del posicionamiento de su elemento contenedor.

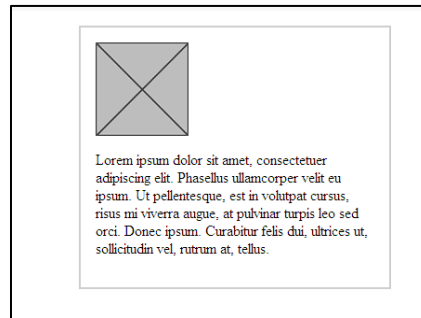
Cuando una caja se posiciona de forma absoluta, el resto de elementos de la página se ven afectados y modifican su posición.



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:

1. Se buscan todos los elementos contenedores de la caja hasta llegar al elemento <body> de la página.
2. Se recorren todos los elementos contenedores empezando por el más cercano a la caja y llegando hasta el <body>
3. 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.
4. Si ningún elemento contenedor está posicionado, la referencia es la ventana del navegador, que no debe confundirse con el elemento <body> de la página.

Ej1:



```
div {  
    border: 2px solid #CCC;  
    padding: 1em;  
    margin: 1em 0 1em 4em;  
    width: 300px;  
}
```

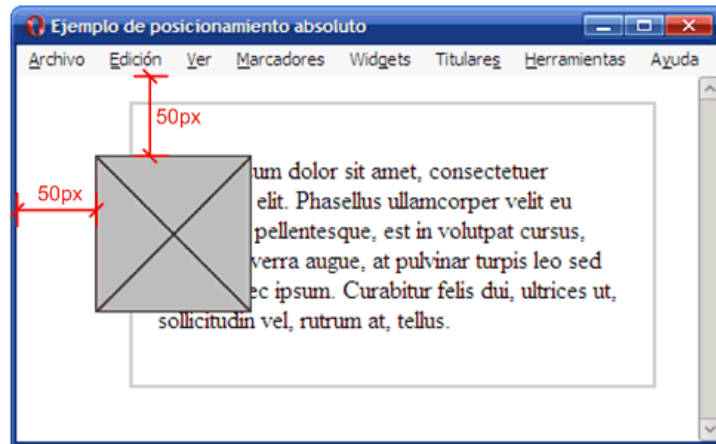
El código html es:

```
<div>  
      
    <p>Lorem ipsum ...</p>  
</div>
```

Se posiciona de forma absoluta la imagen con “position” y se indica su nueva posición mediante las propiedades top y left:

```
div img {  
    position: absolute;  
    top: 50px;  
    left: 50px;  
}
```

La imagen posicionada de forma absoluta no toma como referencia su elemento contenedor <div>, sino 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.

Ej2: se modifica el ejemplo1, 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;
}
```

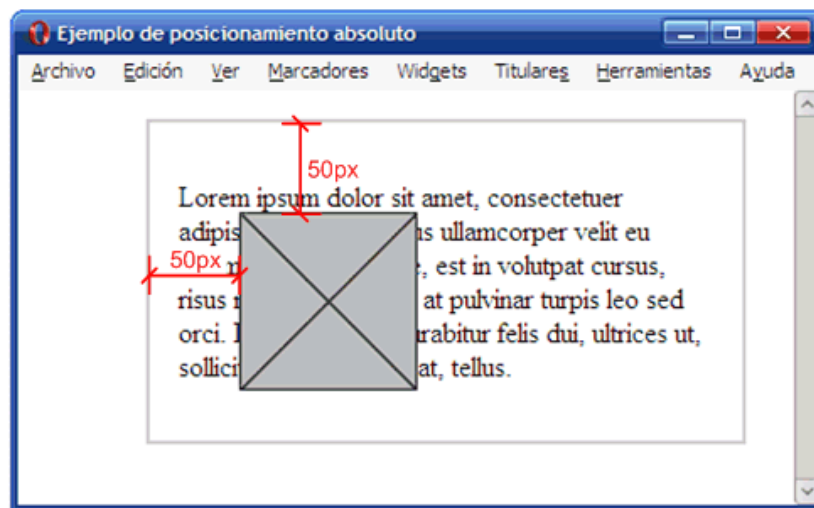
```

    width: 300px;
    position: relative;
}

div img {
    position: absolute;
    top: 50px;
    left: 50px;
}

```

En este caso, como el elemento contenedor de la imagen está posicionado, se convierte en la referencia para el posicionamiento absoluto:



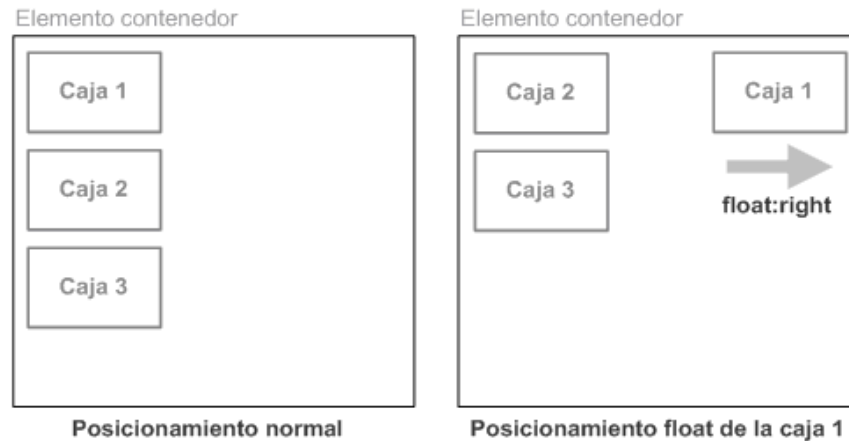
Nota: si se quiere posicionar un elemento de forma absoluta respecto de su elemento contenedor, es imprescindible posicionar este último. Para ello, sólo es necesario añadir la propiedad **position: relative**, por lo que no es obligatorio desplazar el elemento contenedor respecto de su posición original.

- **fixed:** corresponde al posicionamiento fijo. El desplazamiento se establece de la misma forma que en el posicionamiento absoluto, pero en este caso el elemento permanece inamovible en la pantalla.

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.

POSICIONAMIENTO FLOTANTE:

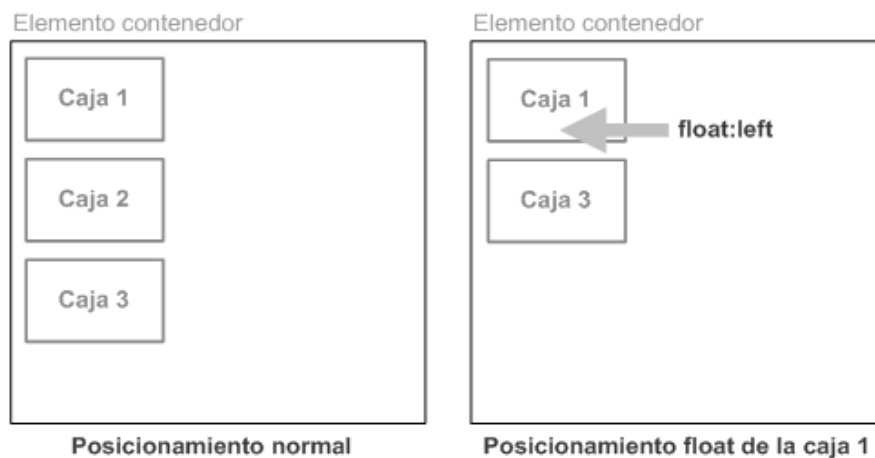
Cuando una caja se posiciona con el modelo de posicionamiento flotante, automáticamente se convierte en una caja flotante, lo que significa que se desplaza hasta la zona más a la izquierda o más a la derecha de la posición en la que originalmente se encontraba.



Cuando se posiciona una caja de forma flotante:

- * La caja deja de pertenecer al flujo normal de la página, lo que significa que el resto de cajas ocupan el lugar dejado por la caja flotante.
- * La caja flotante se posiciona lo más a la izquierda o lo más a la derecha posible de la posición en la que se encontraba originalmente.

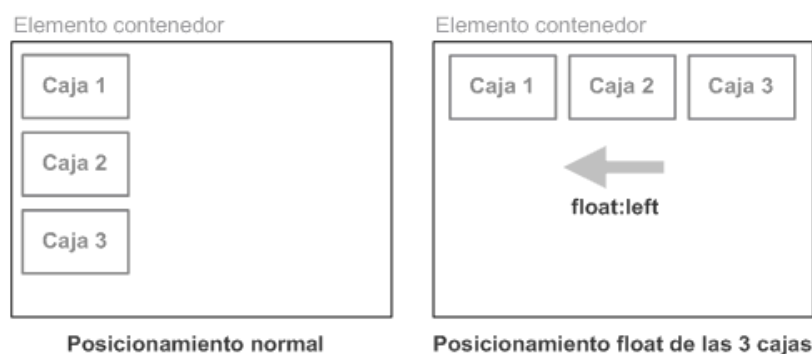
Ej2: Si en el mismo ejemplo, se posiciona hacia la izquierda la caja1, se obtiene lo siguiente:



La caja 2 ahora se muestra dónde estaba la caja 1 y la caja 3 se muestra dónde estaba la caja 2.

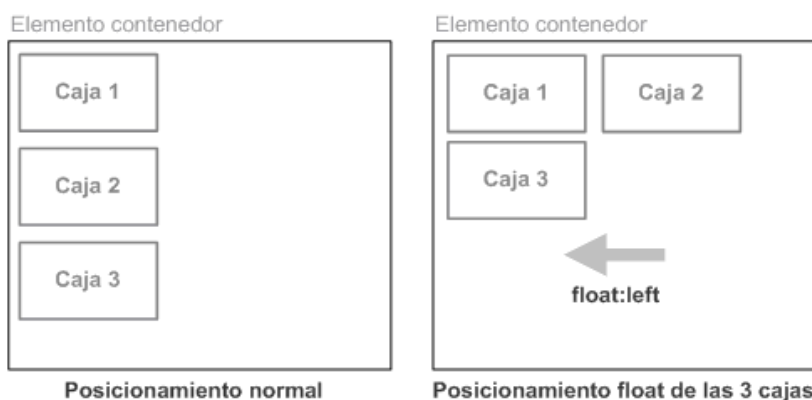
Al mismo tiempo, la caja 1 se desplaza todo lo posible hacia la izquierda de la posición en la que se encontraba. El resultado es que la caja 1 se muestra encima de la nueva posición de la caja 2 y tapa todos sus contenidos.

Ej3: Si existen otras cajas flotantes, al posicionar de forma flotante otra caja, se tiene en cuenta las otras cajas flotantes existentes. En el siguiente ejemplo se posicionan de forma flotante hacia la izquierda las tres cajas:



Como la caja 1 ya estaba posicionada lo más a la izquierda posible, la caja 2 sólo puede colocarse al lado del borde derecho de la caja 1, que es el sitio más a la izquierda posible respecto de la zona en la que se encontraba.

Ej4: Si no existe sitio en la línea actual, la caja flotante baja a la línea inferior hasta que encuentra el sitio necesario para mostrarse lo más a la izquierda o lo más a la derecha posible en esa nueva línea:



Las cajas flotantes influyen en la disposición de todas las demás cajas. Los elementos en línea hacen sitio a las cajas flotantes adaptando su anchura al espacio libre dejado por la caja desplazada.

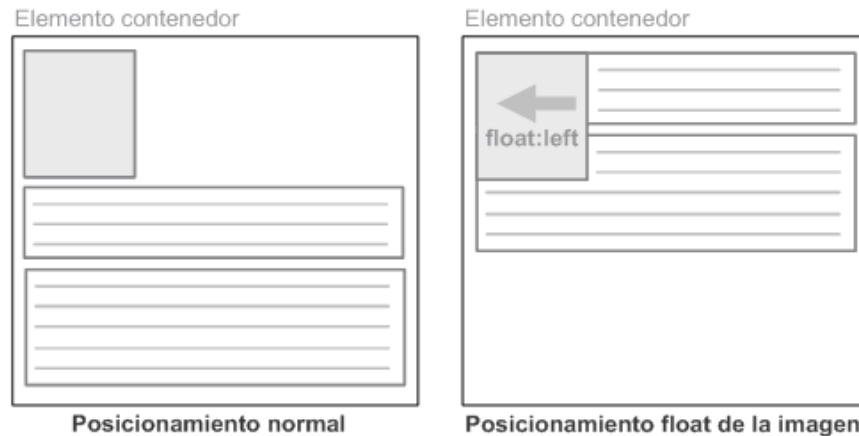
Los elementos de bloque no les hacen sitio, pero sí adaptan sus contenidos para que no se solapen con las cajas flotantes.

float → Establece el tipo de posicionamiento flotante del elemento, su valor por defecto es none. También puede tener los valores: right y left.

Con left la caja se desplaza hasta el punto más a la izquierda posible en esa misma línea (si no existe sitio en esa línea, la caja baja una línea y se muestra lo más a la izquierda posible en esa nueva línea). El resto de elementos adyacentes se adaptan y fluyen alrededor de la caja flotante.

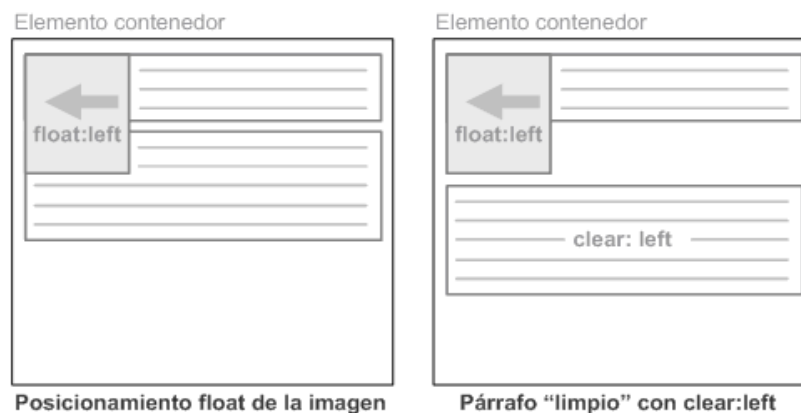
Los elementos que se encuentran alrededor de una caja flotante adaptan sus contenidos para que fluyan alrededor del elemento posicionado:

```
img { float: left; }
```



clear → permite modificar el comportamiento por defecto del posicionamiento flotante para forzar a un elemento a mostrarse debajo de cualquier caja flotante.

```
#parrafo2{ clear: left; }
```



Los valores de clear, pueden ser:

left → el elemento se desplaza de forma descendente hasta que pueda colocarse en una línea en la que no haya ninguna caja flotante en el lado izquierdo.

La especificación oficial de CSS explica este comportamiento como "un desplazamiento descendente hasta que el borde superior del elemento esté por debajo del borde inferior de cualquier elemento flotante hacia la izquierda".

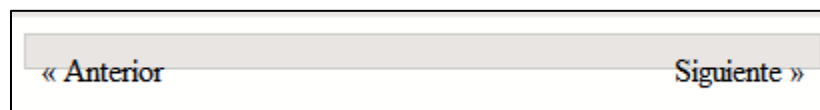
right → el comportamiento es análogo, salvo que en este caso se tienen en cuenta los elementos desplazados hacia la derecha.

both → despeja los lados izquierdo y derecho del elemento, ya que desplaza el elemento de forma descendente hasta que el borde superior se encuentre por debajo del borde inferior de cualquier elemento flotante hacia la izquierda o hacia la derecha.

Ej. Uso de clear: Se tiene el siguiente código:

```
#paginacion {  
    border: 1px solid #CCC;  
    background-color: #E0E0E0;  
    padding: .5em;  
}  
  
.derecha { float: right; }  
  
.izquierda { float: left; }  
  
<div id="paginacion">  
    <span class="izquierda">&laquo; Anterior</span>  
    <span class="derecha">Siguiete &raquo;</span>  
</div>
```

En el navegador se ve así:



Los elementos Anterior y Siguiente se salen de su elemento contenedor y el resultado es visualmente incorrecto.

El motivo de este comportamiento es que el **elemento** posicionado de forma **flotante** ya **no pertenece al flujo normal de la página HTML**. Por lo tanto, el elemento `<div id="paginacion">` en realidad no encierra ningún contenido y por eso se visualiza incorrectamente.

Una de las soluciones es agregar un div sin contenido, y darle un estilo: "clear:both"

```
div.clear {  
    clear: both;  
}  
  
<div id="paginacion">  
    <span class="izquierda">&laquo; Anterior</span>  
    <span class="derecha">Siguiente &raquo;</span>  
    <div class="clear"></div>  
</div>
```

VISUALIZACIÓN:

Para controlar la visualización de los elementos, se definen las siguientes 4 propiedades:

- **display** → Permite controlar la forma de visualizar un elemento e incluso ocultarlo completamente, haciendo que desaparezca de la página. Como el elemento oculto no se muestra, el resto de elementos de la página se mueven para ocupar su lugar.

Su valor por defecto es inline, y puede tener los siguientes valores: inline | block | none | list-item | run-in | inline-block | table | inline-table | table-row-group | table-header-group | table-footer-group | table-row | table-column-group | table-column | table-cell | table-caption | inherit.

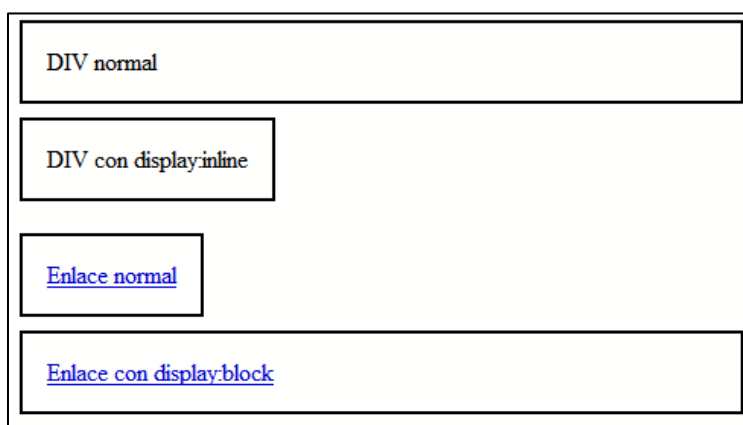
Los valores más utilizados son inline, block y none.

block → muestra un elemento como si fuera un elemento de bloque, independientemente del tipo de elemento que se trate.

inline → visualiza un elemento en forma de elemento en línea, independientemente del tipo de elemento que se trate.

none → oculta un elemento y hace que desaparezca de la página. El resto de elementos de la página se visualizan como si no existiera el elemento oculto, es decir, pueden ocupar el espacio en el que se debería visualizar el elemento.

Ej: se usa `display` para mostrar un elemento de bloque como si fuera un elemento en línea y para mostrar un elemento en línea como si fuera un elemento de bloque:



-
- **visibility** → Permite hacer visibles e invisibles a los elementos.

Su valor por defecto es visible, y puede tener los siguientes valores:

hidden → es posible convertir una caja en invisible para que no muestre sus contenidos. El resto de elementos de la página se muestran como si la caja todavía fuera visible, por lo que en el lugar donde originalmente se mostraba la caja invisible, ahora se muestra un hueco vacío.

collapse → sólo se puede utilizar en las filas, grupos de filas, columnas y grupos de columnas de una tabla. Su efecto es similar al de la propiedad `display`, ya que oculta completamente la fila y/o columna y se pueden mostrar otros contenidos en ese lugar. Si se utiliza el valor `collapse` sobre cualquier otro tipo de elemento, su efecto es idéntico al valor `hidden`.

Las diferencias entre “`display`” y “`visibility`” son las siguientes:

Se tienen 9 cajas, y a la caja número 5, se le aplican las propiedades “`display`” y “`visibility`”, el resultado es el siguiente:

5 = display: none			5 = visibility: hidden		
1	2	3	1	2	3
4	6	7	4		6
8	9		7	8	9

Relación entre display, float y position:

Cuando se establecen las propiedades display, float y position sobre una misma caja, su interpretación es la siguiente:

1. Si display vale none, se ignoran las propiedades float y position y la caja no se muestra en la página.
2. Si position vale absolute o fixed, la caja se posiciona de forma absoluta, se considera que float vale none y la propiedad display vale block tanto para los elementos en línea como para los elementos de bloque. La posición de la caja se determina mediante el valor de las propiedades top, right, bottom y left.
3. En cualquier otro caso, si float tiene un valor distinto de none, la caja se posiciona de forma flotante y la propiedad display vale block tanto para los elementos en línea como para los elementos de bloque.

- **overflow** → Permite controlar los contenidos sobrantes de un elemento.

Normalmente, los contenidos de un elemento se pueden mostrar en el espacio reservado para ese elemento. Sin embargo, en algunas ocasiones el contenido de un elemento no cabe en el espacio reservado para ese elemento y se desborda.

La situación más habitual en la que el contenido sobresale de su espacio reservado es cuando se establece la anchura y/o altura de un elemento mediante la propiedad width y/o height. Otra situación habitual es la de las líneas muy largas contenidas dentro de un elemento <pre>, que hacen que la página entera sea demasiado ancha.

Overflow se aplica a: Elementos de bloque y celdas de tablas.

Su valor por defecto es visible, y puede tener los siguientes valores:

visible → el contenido no se corta y se muestra sobresaliendo la zona reservada para visualizar el elemento.

hidden → el contenido sobrante se oculta y sólo se visualiza la parte del contenido que cabe dentro de la zona reservada para el elemento.

scroll → solamente se visualiza el contenido que cabe dentro de la zona reservada para el elemento, pero también se muestran barras de scroll que permiten visualizar el resto del contenido.

auto → el comportamiento depende del navegador, aunque normalmente es el mismo que la propiedad scroll.

Ej1:

```
div {  
    display: inline;  
    float: left;  
    margin: 1em;  
    padding: .3em;  
    border: 2px solid #555;  
    width: 100px;  
    height: 150px;  
    font: 1em Arial, Helvetica, sans-serif;  
}
```

```
div#oculto{ overflow:hidden; }
```

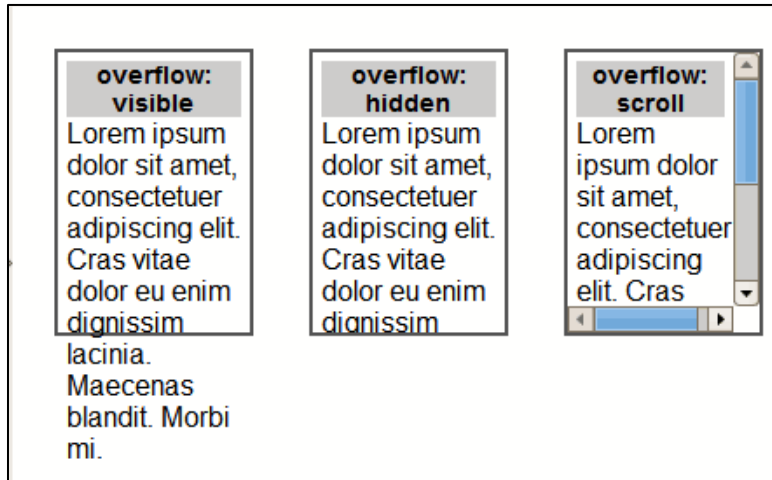
```
div#desplaza{ overflow:scroll; }
```

```
<div><h1>overflow: visible</h1> Lorem ipsum...</div>
```

```
<div id="oculto"><h1>overflow: hidden</h1> Lorem ipsum...</div>
```

```
<div id="desplaza"><h1>overflow: scroll</h1> Lorem ipsum... </div>
```

En el navegador se ve así:



Ej2: recordando el ejemplo del posicionamiento float, en donde se tenía:



Otra forma de solucionar ese comportamiento es usando overflow:

```
#paginacion {
    border: 1px solid #CCC;
    background-color: #E0E0E0;
    padding: .5em;
    overflow: hidden;
}

.derecha { float: right; }

.izquierda { float: left; }
```

- **z-index** → Establece el nivel tridimensional en el que se muestra el elemento. Con esta propiedad es posible indicar las cajas que se muestran delante o detrás de otras cajas cuando se producen solapamientos.

Su valor por defecto es auto, pero se pueden indicar valores numéricos enteros, incluso negativos.

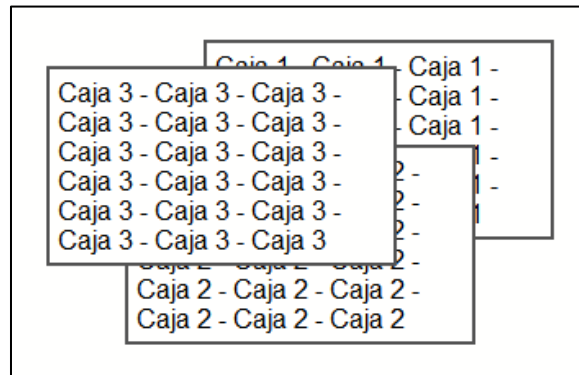
El número 0 es el nivel más bajo.

Cuanto más alto sea el valor numérico, más cerca del usuario se muestra la caja.

Ej:

```
div { position: absolute; }  
  
#caja1 { z-index: 5; top: 1em; left: 8em;}  
  
#caja2 { z-index: 15; top: 5em; left: 5em;}  
  
#caja3 { z-index: 25; top: 2em; left: 2em;}  
  
<div id="caja1">Caja 1 - Caja 1 - Caja 1 ...</div>  
<div id="caja2">Caja 2 - Caja 2 - Caja 2 ...</div>  
<div id="caja3">Caja 3 - Caja 3 - Caja 3 ...</div>
```

En el navegador se ve así:



Nota: La propiedad z-index sólo tiene efecto en los elementos posicionados, por lo que es obligatorio que la propiedad z-index vaya acompañada de la propiedad **position**.

TEXTO:

color → Establece el color de letra utilizado para el texto.

Ej:

```
h1 { color: #369; }  
  
p { color: black; }  
  
a, span { color: #B1251E; }  
  
div { color: rgb(71, 98, 176); }
```

Como el valor de la propiedad color se hereda, normalmente se establece la propiedad color en el elemento body para establecer el color de letra de todos los elementos de la página: `body { color: #777; }`

font-family → Establece el tipo de letra utilizado para el texto.

El tipo de letra del texto se puede indicar de dos formas diferentes:

- Mediante el nombre de una familia tipográfica: en otras palabras, mediante el nombre del tipo de letra, por ejemplo "Arial", "Verdana", "Garamond", etc.
- Mediante el nombre genérico de una familia tipográfica: los nombres genéricos no se refieren a ninguna fuente en concreto, sino que hacen referencia al estilo del tipo de letra. Las familias genéricas definidas son serif (tipo de letra similar a Times New Roman), sans-serif (tipo Arial), cursive (tipo Comic Sans), fantasy (tipo Impact) y monospace (tipo Courier New).

El valor de font-family suele definirse como una lista de tipos de letra alternativos separados por comas. El último valor de la lista es el nombre de la familia tipográfica genérica que más se parece al tipo de letra que se quiere utilizar.

Las listas de tipos de letra más utilizadas son las siguientes:

font-family: Arial, Helvetica, sans-serif;

font-family: "Times New Roman", Times, serif;

font-family: "Courier New", Courier, monospace;

font-family: Georgia, "Times New Roman", Times, serif;

font-family: Verdana, Arial, Helvetica, sans-serif;

font-size → Establece el tamaño de letra utilizado para el texto. Su valor por defecto es médium, y se pueden indicar los siguientes valores: tamaño_absoluto | tamaño_relativo | unidad de medida | porcentaje | inherit.

Además de todas las unidades de medida relativas y absolutas, y el uso de porcentajes, CSS permite utilizar una serie de palabras clave para indicar el tamaño de letra del texto:

tamaño_absoluto: indica el tamaño de letra de forma absoluta mediante alguna de las siguientes palabras clave: xx-small, x-small, small, medium, large, x-large, xx-large.

tamaño_relativo: indica de forma relativa el tamaño de letra del texto mediante dos palabras clave (larger, smaller) que toman como referencia el tamaño de letra del elemento padre.

Absoluto xx-small x-small small medium large x-large xx-large	Em .5em .7em .9em 1em 1.5em 2em 3em	% 50% 70% 90% 100% 150% 200% 300%	Px 9px 12px 14px 18px 24px 36px 48px	Pt 9pt 12pt 14pt 18pt 24pt 36pt 48pt
---	---	---	--	--

CSS recomienda indicar el tamaño del texto en la unidad em o en porcentaje (%).

font-weight → Establece la anchura de la letra utilizada para el texto, esta propiedad se usa para el texto en **NEGRILLA**.

Su valor por defecto es normal, y puede tener los siguientes valores: normal | bold | bolder | lighter | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | inherit.

El valor normal equivale al valor numérico 400 y el valor **bold** al valor numérico 700.

font-style → Establece el estilo de la letra utilizada para el texto. Su valor por defecto es normal, y puede tener los siguientes valores: normal | italic | oblique | inherit.

Para poner un “*Texto en cursiva o itálica*” se usa el valor “*italic*”, la diferencia con “oblique” es casi imperceptible.

font-variant → Establece el estilo alternativo de la letra utilizada para el texto. Sus dos únicos valores son normal o small-caps. Este último valor hace que un texto “así” aparezca “**ASÍ**”.

font → propiedad shorthand para indicar todas o algunas de las propiedades de la tipografía de un texto.

El orden en el que se deben indicar las propiedades del texto es el siguiente:

1. En primer lugar y de forma opcional se indican el font-style, font-variant y font-weight en cualquier orden.
2. A continuación, se indica obligatoriamente el valor de font-size seguido opcionalmente por el valor de line-height.
3. Por último, se indica obligatoriamente el tipo de letra a utilizar.

Ej: /*En este caso el /140% indica el valor de interlineado o line-height*/

font: 76%/140% Verdana,Arial,Helvetica,sans-serif;

font: normal 24px/26px "Century Gothic", "Trebuchet MS", Arial, Helvetica, sans-serif;

font: normal .94em "Trebuchet MS", Arial, Helvetica, sans-serif;

font: bold 1em "Trebuchet MS", Arial, Sans-Serif;

font: normal 0.9em "Lucida Grande", Verdana, Arial, Helvetica, sans-serif;

font: normal 1.2em/1em helvetica, arial, sans-serif;

font: 11px verdana, sans-serif;

font: normal 1.4em/1.6em "helvetica", arial, sans-serif;

font: bold 14px georgia, times, serif;

Mini Resumen de Tipos de Letra:

/*Cambia el color y el Tipo de Fuente para todo el documento*/

```
body{  
    color:red;  
    font-family:Arial,Helvetica,sans-serif;  
}
```

/*Indican el tamaño y el ancho del texto (negrilla) respectivamente*/

```
.parrafo1{  
    font-size:32px;  
    font-weight:bold;  
}
```

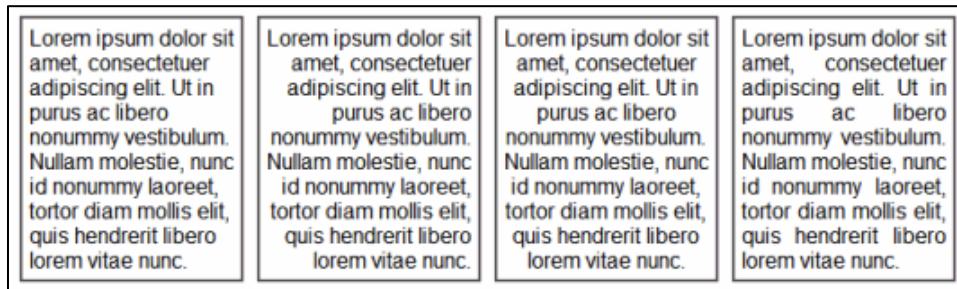
/*Letra *Itálica* o *Cursiva**/

```
.parrafo2{  
    font-style:italic;  
}
```

/*Un texto “así” lo pone “ASÍ”*/

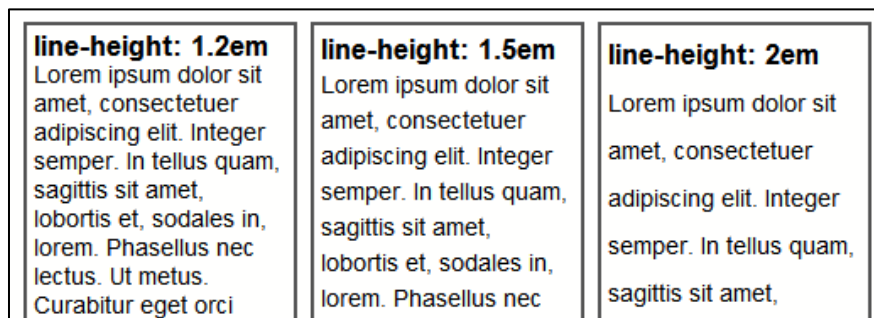
```
.parrafo3{  
    font-variant:small-caps;  
}
```

text-align → Establece la alineación del contenido del elemento, Su valor por defecto es left, y permite los siguientes valores: left | right | center | justify | inherit.



La propiedad text-align no sólo alinea el texto que contiene un elemento, sino que también alinea todos sus contenidos, como por ejemplo las imágenes.

line-height → Permite establecer la altura de línea (interlineado) de los elementos.



Además de todas las unidades de medida y el uso de porcentajes, la propiedad line-height permite indicar un número sin unidades que se interpreta como el múltiplo del tamaño de letra del elemento.

Por lo tanto, estas tres reglas CSS son equivalentes:

p { line-height: 1.2; font-size: 1em }

p { line-height: 1.2em; font-size: 1em }

p { line-height: 120%; font-size: 1em }

text-decoration → Establece la decoración del texto (subrayado, tachado, parpadeante, etc.). Sus valores pueden ser:

underline → subraya el texto.

overline → añade una línea en la parte superior del texto.













line-through → muestra el texto tachado con una línea continua.

blink → muestra el texto parpadeante.

text-transform → Transforma el texto original (lo transforma a mayúsculas, a minúsculas, o con la primera letra de cada palabra en mayúscula.).

Original Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer semper. In tellus quam, sagittis sit amet, lobortis et, sodales in, lorem. Phasellus nec lectus. Ut metus. Curabitur eget orci vitae felis volutpat nonummy.
Text-transform: Capitalize Lorem Ipsum Dolor Sit Amet, Consectetur Adipiscing Elit. Integer Semper. In Tellus Quam, Sagittis Sit Amet, Lobortis Et, Sodales In, Lorem. Phasellus Nec Lectus. Ut Metus. Curabitur Eget Orci Vitae Felis Volutpat Nonummy.
text-transform: lowercase lorem ipsum dolor sit amet, consectetur adipiscing elit. integer semper. in tellus quam, sagittis sit amet, lobortis et, sodales in, lorem. phasellus nec lectus. ut metus. curabitur eget orci vitae felis volutpat nonummy.
TEXT-TRANSFORM: UPPERCASE LOREM IPSUM DOLOR SIT AMET, CONSECTETUER ADIPISCING ELIT. INTEGER SEMPER. IN TELLUS QUAM, SAGITTIS SIT AMET, LOBORTIS ET, SODALES IN, LOREM. PHASELLUS NEC LECTUS. UT METUS. CURABITUR EGET ORCI VITAE FELIS VOLUTPAT NONUMMY.

vertical-align → Determina la alineación vertical de los contenidos de un elemento. Su valor por defecto es baseline. Se aplica a elementos en línea y celdas de tabla.

 baseline Ayuda	 sub Ayuda	 super Ayuda
 top Ayuda	 text-top Ayuda	 middle Ayuda
 bottom Ayuda	 text-bottom Ayuda	 medida 1em Ayuda
 medida -1em Ayuda	 porcentaje 10% Ayuda	 porcentaje 80% Ayuda

text-indent → Tabula desde la izquierda la primera línea del texto original. Se aplica a los elementos de bloque y las celdas de tabla. Puede tener como valores unidades de medida o porcentajes.

<p>text-indent: 0</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum dictum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Etiam pharetra blandit purus.</p> <p>Sed semper adipiscing justo. Maecenas molestie ultrices tellus. In eget magna sed arcu lobortis auctor. Nulla vel leo vel justo feugiat hendrerit. Donec venenatis sollicitudin sapien.</p> <p>Aenean lectus nulla, pulvinar in, posuere et, ultrices et, pede. Vivamus non enim. Donec ac nulla sit amet tortor laoreet consequat.</p> <p>Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Praesent luctus, diam sed posuere porttitor, elit lectus pretium dui, et sagittis leo nunc et lacus.</p>	<p>text-indent: 2em</p> <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum dictum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Etiam pharetra blandit purus.</p> <p>Sed semper adipiscing justo. Maecenas molestie ultrices tellus. In eget magna sed arcu lobortis auctor. Nulla vel leo vel justo feugiat hendrerit. Donec venenatis sollicitudin sapien.</p> <p>Aenean lectus nulla, pulvinar in, posuere et, ultrices et, pede. Vivamus non enim. Donec ac nulla sit amet tortor laoreet consequat.</p> <p>Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Praesent luctus, diam sed posuere porttitor, elit lectus pretium dui, et sagittis leo nunc et lacus.</p>
--	--

letter-spacing → Permite establecer el espacio entre las letras que forman las palabras del texto. Sus valores pueden ser: “normal” o una unidad de medida.

.especial h1 { letter-spacing: .2em; }

word-spacing → Permite establecer el espacio entre las palabras que forman el texto. Sus valores pueden ser: “normal” o una unidad de medida.

.especial p { word-spacing: .5em; }

Nota: Cuando se utiliza un valor numérico en las propiedades letter-spacing y word-spacing, se interpreta como la separación adicional que se añade (si el valor es positivo) o se quita (si el valor es negativo) a la separación por defecto entre letras y palabras respectivamente.

white-space → Establece el tratamiento de los espacios en blanco del texto. Sus valores son los siguientes:

normal: comportamiento por defecto de HTML.

pre: se respetan los espacios en blanco y las nuevas líneas (exactamente igual que la etiqueta <pre>). Si la línea es muy larga, se sale del espacio asignado para ese contenido.

nowrap: elimina los espacios en blanco y las nuevas líneas. Si la línea es muy larga, se sale del espacio asignado para ese contenido.

pre-wrap: se respetan los espacios en blanco y las nuevas líneas, pero ajustando cada línea al espacio asignado para ese contenido.

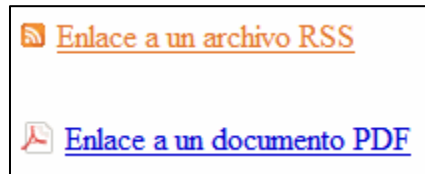
pre-line: elimina los espacios en blanco y respeta las nuevas líneas, pero ajustando cada línea al espacio asignado para ese contenido.

Ej: El párrafo original contiene espacios en blanco y nuevas líneas y se ha limitado la anchura de su elemento contenedor.

white-space: pre Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum dictum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Etiam pharetra blandit purus.	
white-space: pre-wrap Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum dictum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Etiam pharetra blandit purus.	
white-space: nowrap Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum dictum. Class aptent taciti sociosqu ad litora	
white-space: pre-line Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vestibulum dictum. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Etiam pharetra blandit purus.	

PERSONALIZAR ENLACES:

Se pueden personalizar enlaces para añadirles una imagen a su izquierda:



Para esto, es necesario definir los siguientes estilos:

```
a { margin: 1em 0; float: left; clear: left; }
```

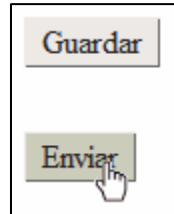
```
.rss {
  color: #E37529;
  padding: 0 0 0 18px;
  background: #FFF url(imagenes/rss.gif) no-repeat left center;
}
```

```
.pdf {
  padding: 0 0 0 22px;
  background: #FFF url(imagenes/pdf.png) no-repeat left center;
}
```

`Enlace a un archivo RSS`

`Enlace a un documento PDF`

Con CSS, también podemos modificar los estilos de un enlace, para que parezcan un botón:



El estilo de estos enlaces es:

```
a { margin: 1em 0; float: left; clear: left; }
```

```
a.boton {  
    text-decoration: none;  
    background: #EEE;  
    color: #222;  
    border: 1px outset #CCC;  
    padding: .1em .5em;  
}
```

```
a.boton:hover {  
    background: #CCB;  
}
```

```
a.boton:active {  
    border: 1px inset #000;  
}
```

`Guardar`

`Enviar`

ESTILOS PARA LISTAS:

list-style-type → Permite establecer el tipo de viñeta mostrada para una lista. Su valor por defecto es “disc”. Con el valor “none” se le quitan las viñetas, y puede tener los siguientes valores que pueden ser de tipo: gráficos, numéricos y alfabéticos.

Los valores gráficos son disc, circle y square y muestran como viñeta un círculo relleno, un círculo vacío y un cuadrado relleno respectivamente.

Los valores numéricos están formados por decimal, decimal-leading-zero, lower-roman, upper-roman, armenian y georgian.

Los valores alfanuméricos se controlan mediante lower-latin, lower-alpha, upper-latin, upper-alpha y lower-greek.

Algunos ejemplos de estos valores son:

<ul style="list-style-type: none">• list-style-type: disc• Elemento• Elemento	<ul style="list-style-type: none">○ list-style-type: circle○ Elemento○ Elemento	<ul style="list-style-type: none">■ list-style-type: square■ Elemento■ Elemento
<ul style="list-style-type: none">i. list-style-type: lower-romanii. Elementoiii. Elemento	<ul style="list-style-type: none">α. list-style-type: lower-greekβ. Elementoγ. Elemento	
<ul style="list-style-type: none">A. list-style-type: upper-latinB. ElementoC. Elemento	<ul style="list-style-type: none">a. list-style-type: lower-latinb. Elementoc. Elemento	<ul style="list-style-type: none">Մ. list-style-type: armenianԲ. ElementoԳ. Elemento
<ul style="list-style-type: none">A. list-style-type: upper-alphaB. ElementoC. Elemento	<ul style="list-style-type: none">01. list-style-type: decimal-leading-zero02. Elemento03. Elemento	

list-style-position → Permite establecer la posición de la viñeta de cada elemento de una lista. Su valor por defecto es “outside” y también puede ser “inside”. Se ve su efecto cuando el contenido de la lista tiene mucho texto:

<ul style="list-style-type: none">■ list-style-position: outside■ Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc in diam. Praesent a justo. Nam odio. Quisque a libero vel massa malesuada scelerisque. Curabitur metus.■ Nunc lobortis tortor. Etiam nec nibh. In tincidunt urna ut erat. Integer velit ante, tempus ut, egestas convallis, euismod in, erat.	<ul style="list-style-type: none">■ list-style-position: inside■ Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nunc in diam. Praesent a justo. Nam odio. Quisque a libero vel massa malesuada scelerisque. Curabitur metus.■ Nunc lobortis tortor. Etiam nec nibh. In tincidunt urna ut erat. Integer velit ante, tempus ut, egestas convallis, euismod in, erat.
--	---

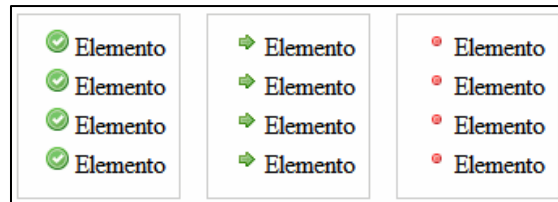
list-style-image → Permite reemplazar las viñetas automáticas por una imagen personalizada. Su valor es la URL de la imagen.

```
ul.ok { list-style-image: url("imagenes/ok.png"); }
```

```
ul.flecha { list-style-image: url("imagenes/flecha.png"); }
```

```
ul.circulo { list-style-image: url("imagenes/circulo_rojo.png"); }
```

En el navegador se ve:



Nota: Si no se encuentra la imagen o no se puede cargar, se muestra la viñeta automática correspondiente (salvo que explícitamente se haya eliminado mediante la propiedad list-style-type).

list-style → Propiedad shorthand que permite establecer de forma simultánea todas las opciones de una lista.

Ej: Si no se puede cargar la imagen, se muestra el tipo de viñeta cuadrada:

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

CREACIÓN DE UN MENÚ VERTICAL:

Para crear un menú, ya sea vertical u horizontal en una página Web, se usan las listas no ordenadas (etiqueta ul) y se modifica su aspecto con CSS.

Ej: la lista normal, sin aplicar estilos es:

```
<ul>
  <li><a href="#">Elemento 1</a></li>
  <li><a href="#">Elemento 2</a></li>
  <li><a href="#">Elemento 3</a></li>
  <li><a href="#">Elemento 4</a></li>
  <li><a href="#">Elemento 5</a></li>
  <li><a href="#">Elemento 6</a></li>
</ul>
```

Los pasos para convertir la lista en menú son:

1) Definir la anchura del menú:

```
ul.menu { width: 180px; }
```

2) Eliminar las viñetas automáticas y todos los márgenes y espaciados aplicados por defecto:

```
ul.menu {  
    list-style: none;  
    margin: 0;  
    padding: 0;  
    width: 180px;  
}
```

3) Añadir un borde al menú de navegación y establecer el color de fondo y los bordes de cada elemento del menú:

```
ul.menu {  
    border: 1px solid #7C7C7C;  
    border-bottom: none;  
    list-style: none;  
    margin: 0;  
    padding: 0;  
    width: 180px;  
}  
  
ul.menu li {  
    background: #F4F4F4;  
    border-bottom: 1px solid #7C7C7C;  
    border-top: 1px solid #FFF;  
}
```

4) Aplicar estilos a los enlaces: mostrarlos como un elemento de bloque para que ocupen todo el espacio de cada del menú, añadir un espacio de relleno y modificar los colores y la decoración por defecto:

```
ul.menu li a {
    color: #333;
    display: block;
    padding: .2em 0 .2em .5em;
    text-decoration: none;
}
```

En el navegador el menú se debería ver así:

Elemento 1
Elemento 2
Elemento 3
Elemento 4
Elemento 5
Elemento 6

Nota: Este es un menú vertical muy básico, si se quieren agregar imágenes para cada elemento, así como el comportamiento del cambio de color cuando se le pone el mouse encima, se puede consultar el ejercicio 10.

CREACIÓN DE UN MENÚ HORIZONTAL:

Para crear un menú horizontal básico (con la misma estructura de lista ul), se realizan los siguientes pasos:

1) Aplicar los estilos CSS básicos para establecer el estilo del menú (similares a los del menú vertical anterior):

```
ul.menu {
    background: #F4F4F4;
    border: 1px solid #7C7C7C;
    list-style: none;
    margin: 0;
    padding: 0;
}
```

```
ul.menu li a {  
    color: #333;  
    display: block;  
    padding: .3em;  
    text-decoration: none;  
}
```

2) Establecer la anchura de los elementos del menú. Como el menú es de anchura variable y contiene cinco elementos, se asigna una anchura del 20% a cada elemento.

Si se quiere tener un control más preciso sobre el aspecto de cada elemento, es necesario asignar una anchura fija al menú.

Además, se **posiciona** de forma **flotante** los elementos de la lista mediante la propiedad float. Esta es la **clave de la transformación** de una lista en un menú horizontal:

```
ul.menu li {  
    float: left;  
    width: 20%;  
}
```

Después de posicionar de forma flotante a todos los elementos de la lista, el elemento es un elemento vacío ya que en su interior no existe ningún elemento posicionado de forma normal.

La solución de este problema es aplicar la propiedad overflow: hidden; al elemento , de forma que encierre a todos los elementos posicionados de forma flotante:

```
ul.menu {  
    overflow: hidden;  
}
```

3) Establecer los bordes de los elementos que forman el menú:

```
ul.menu li a {  
    border-left: 1px solid #FFF;  
    border-right: 1px solid #7C7C7C;  
}
```

4) Por último, se elimina el borde derecho del último elemento de la lista, para evitar el borde duplicado:

```
<ul>
    <li><a href="#">Elemento 1</a></li>...
    <li><a href="#" style="border-right: none">Elemento 5</a></li>
</ul>
```

O se puede obtener el mismo resultado con:

```
ul.menu:last-of-type{
    border-right:none;
}
```

ESTILOS PARA TABLAS:

El siguiente código aplica bordes para la tabla y para cada celda:

```
.normal {
    width: 250px;
    border: 1px solid #000;
}

.normal th, .normal td {
    border: 1px solid #000;
}

<table class="normal" summary="Tabla genérica">
    <tr>
        <th scope="col">A</th>
        <th scope="col">B</th>
        <th scope="col">C</th>
        <th scope="col">D</th>
        <th scope="col">E</th>
    </tr> ...
</table>
```

En el navegador se ve así:

A	B	C	D	E
a	1	2	3	4
b	1	2	3	4
c	1	2	3	4
d	1	2	3	4

border-collapse → Define el mecanismo de fusión de los bordes de las celdas adyacentes de una tabla. Su valor por defecto es “separate”. Y también puede tener el valor “collapse”.

separate fuerza a que cada celda muestre sus cuatro bordes.

collapse fusiona de forma automática los bordes de las celdas adyacentes.

Ej: ejemplo de uso de collapse:

```
.normal {  
    width: 250px;  
    border: 1px solid #000;  
    border-collapse: collapse;  
}  
  
.normal th, .normal td {  
    border: 1px solid #000;  
}
```

En el navegador se ve así:

A	B	C	D	E
a	1	2	3	4
b	1	2	3	4
c	1	2	3	4
d	1	2	3	4

Si se usa “separate”, se puede utilizar la propiedad:

border-spacing → Establece la separación entre los bordes de las celdas adyacentes de una tabla. Su valor por defecto es 0.

Si solamente se indica como valor una medida, se asigna ese valor como separación horizontal y vertical. Si se indican dos medidas, la primera es la separación horizontal y la segunda es la separación vertical entre celdas.

La propiedad border-spacing sólo controla la separación entre celdas.

empty-cells → Define el mecanismo utilizado para el tratamiento de las celdas vacías de una tabla. Su valor por defecto es “show”, y también puede tener el valor “hide”.

Sólo se aplica cuando el modelo de bordes de la tabla es de tipo separate.

hide indica que las celdas vacías no se deben mostrar. Una celda vacía es aquella que no tiene ningún contenido, ni siquiera un espacio en blanco o un .

Ej: arriba se encuentra una tabla normal, y abajo se usa empty-cells:hide;

1		3
4	5	
	8	9
1		3
4	5	
	8	9

caption-side → Establece la posición del título de la tabla (el título se establece mediante <caption>). Su valor por defecto es “top”, y también puede tener el valor “bottom”.

Ej:

```
.especial {
    caption-side: bottom;
}

<table class="especial" summary="Tabla genérica">
    <caption>Tabla 2.- Título especial</caption>
    <tr>
        <td>1</td>
        <td>2</td>
        <td>3</td>
    </tr> ...
</table>
```


En el navegador se ve:

1	2	3
4	5	6
7	8	9
Tabla 2.- Título especial		

ESTILOS PARA FORMULARIOS:

Para mostrar un botón como un enlace:

```
.enlace {  
    border: 0;  
    padding: 0;  
    background-color: transparent;  
    color: blue;  
    border-bottom: 1px solid blue;  
}
```

```
<input class="enlace" type="button" value="Botón como enlace" />
```

Añadir padding (relleno) a los campos input, para que se vean mucho mejores:

```
form.elegante input {  
    padding: .2em;  
}
```

En el navegador el cambio seria:

Formulario sin padding en los input

Nombre

Contraseña

Formulario con padding en los input

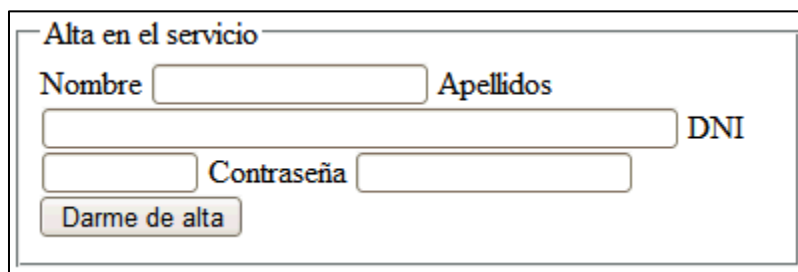
Nombre

Contraseña

Alinear y Formatear labels: Teniendo el siguiente formulario:

```
<form>
  <fieldset>
    <legend>Alta en el servicio</legend>
    <label for="nombre">Nombre</label>
    <input type="text" id="nombre" />
    <label for="apellidos">Apellidos</label>
    <input type="text" id="apellidos" size="50" />
    <label for="dni">DNI</label>
    <input type="text" id="dni" size="10" maxlength="9" />
    <label for="contrasena">Contraseña</label>
    <input type="password" id="contrasena" />
    <input class="btn" type="submit" value="Dar de alta" />
  </fieldset>
</form>
```

Al ser los “labels” y los “input”, elementos en línea, se verían así en el navegador:

A screenshot of a web browser displaying the rendered HTML form. The form is titled "Alta en el servicio" and is enclosed in a box. It contains four input fields: "Nombre", "Apellidos", "DNI", and "Contraseña". The "Nombre" and "Apellidos" fields are on the same line, followed by the "DNI" field. The "Contraseña" field is on a new line. Below the input fields is a button labeled "Dar de alta".

Es posible mejorar su aspecto añadiendo los siguientes estilos:

```
label {
  display: block;
  margin: .5em 0 0 0;
}

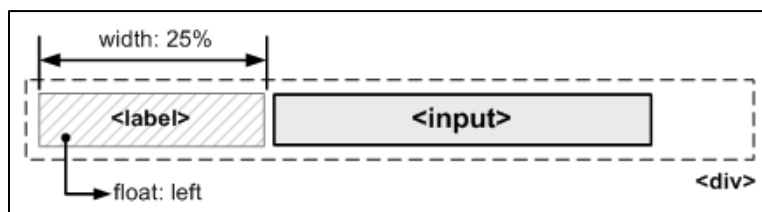
.btn {
  display: block;
}
```

↑
margin: 1em 0;
}

En el navegador se verían así:

Y si se quiere que tanto el label como su input correspondiente se encuentren en una sola línea, es necesario encerrar cada pareja label e input, en un div:

```
<div>
  <label for="nombre">Nombre</label>
  <input type="text" id="nombre" />
</div>...
```

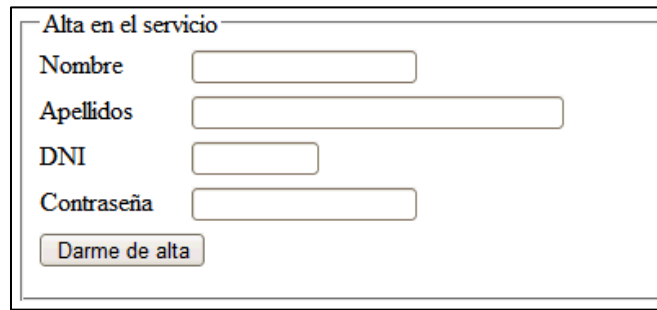


Y luego asignamos los estilos necesarios:

```
div {
  margin: .4em 0;
}

div label {
  width: 25%;
  float: left;
}
```

En el navegador se vería así:



Alta en el servicio

Nombre

Apellidos

DNI

Contraseña

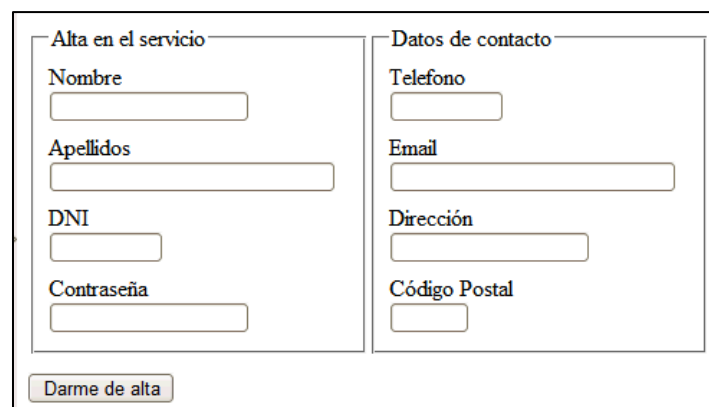
Formulario en varias columnas: sirve para agrupar campos relacionados.

Se aplica la siguiente regla CSS a los <fieldset> del formulario:

```
form fieldset {  
    float: left;  
    width: 48%;  
}
```

```
<form>  
    <fieldset>  
        ...  
    </fieldset>  
    ...  
</form>
```

En el navegador se ve así:



Alta en el servicio

Nombre

Apellidos

DNI

Contraseña

Datos de contacto

Telefono

Email

Dirección

Código Postal

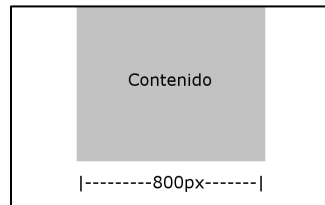
Se pueden agregar más columnas, solo hay que cambiar el valor de “width” de los fieldset.

LAYOUT (Estructura):

Centrar Pág. Horizontalmente: Se centra con la propiedad margin, primero se agrupan todos los contenidos de la página en un elemento <div> y se le asigna a ese <div> unos márgenes laterales automáticos.

El <div> que encierra los contenidos se suele llamar contenedor (en inglés se denomina wrapper o container):

```
#contenedor {  
    width: 800px;  
    margin: 0 auto;  
}  
  
<body>  
    <div id="contenedor"> ...  
    </div>  
</body>
```



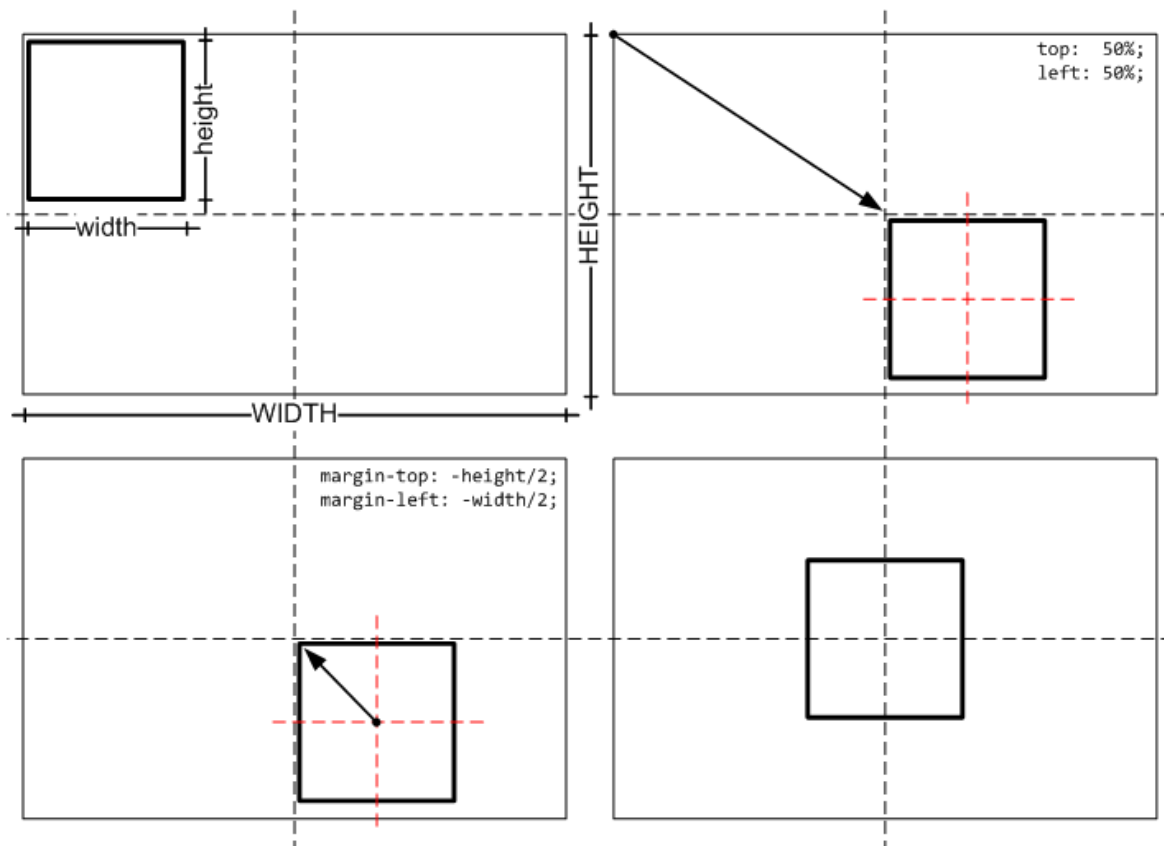
El valor 0 auto significa que los márgenes superior e inferior son iguales a 0 y los márgenes laterales toman un valor de auto.

Cuando se asignan márgenes laterales automáticos a un elemento, los navegadores centran ese elemento respecto de su elemento padre. En este ejemplo, el elemento padre del <div> es la propia página (el elemento <body>), por lo que se consigue centrar el elemento <div> respecto de la ventana del navegador.

Para hacer un diseño dinámico fluido, que se adapta a la anchura de la ventana del navegador y permanece siempre centrado, solo es necesario definir la anchura del contenedor en porcentaje (%):

```
#contenedor {  
    width: 70%;  
    margin: 0 auto;  
}
```

Centrar Pág. Verticalmente: aunque no es muy común centrar una página verticalmente, para hacerlo se siguen los siguientes cálculos:

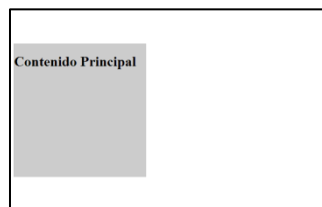


En este ejemplo, el contenedor tiene un tamaño de 500px * 500px:

```
#contenedor {  
    width: 500px;  
    height: 500px;  
  
    position: absolute;  
    top: 50%;  
    left: 50%;  
  
    margin-top: -250px; /* height/2 = 500px / 2 */  
    margin-left: -250px; /* width/2 = 500px / 2 */  
}
```

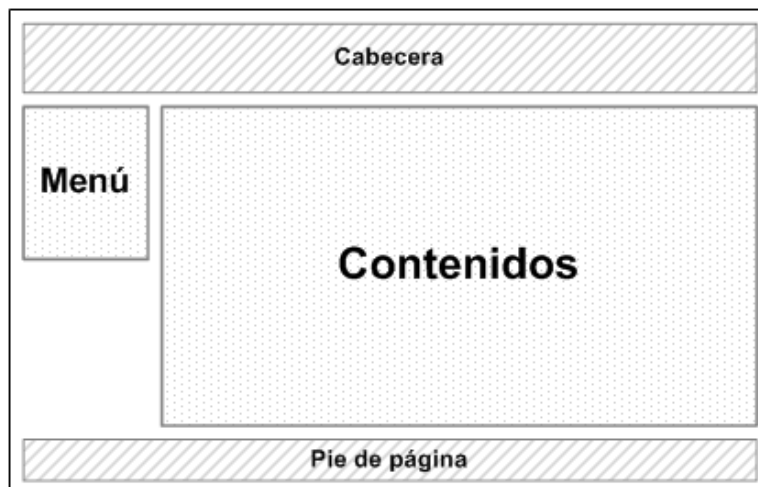
Para centrar una página sólo verticalmente, se debe prescindir tanto del posicionamiento horizontal como del desplazamiento horizontal:

```
#contenedor {  
    width: 500px;  
    height: 500px;  
  
    position: absolute;  
    top: 50%;  
  
    margin-top: -250px; /* height/2 = 500px / 2 */  
}
```

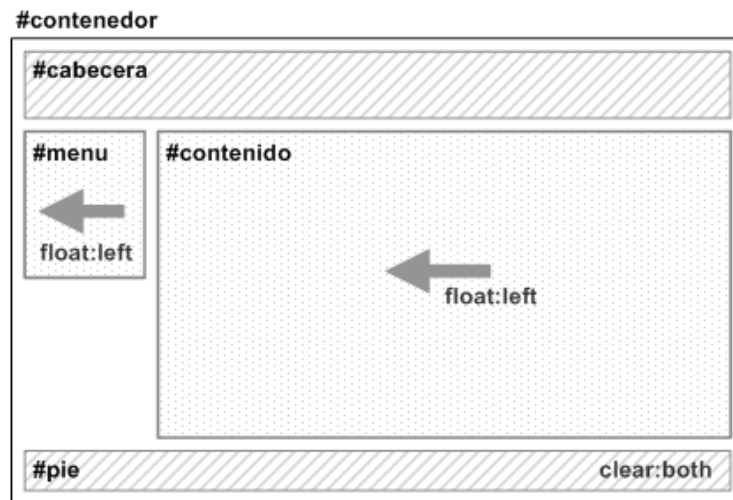


Diseño a 2 columnas con cabecera y pie de página:

El objetivo de este diseño es definir una estructura de página con cabecera y pie, un menú lateral de navegación y una zona de contenidos. La anchura de la página se fija en 700px, la anchura del menú es de 150px y la anchura de los contenidos es de 550px:



Para hacer el diseño, se usa la propiedad “float” para los elementos posicionados como el menú y los contenidos, y se usa “clear” en el pie de página para evitar los solapamientos ocasionados por los elementos posicionados con float.



En este caso, el menú y el contenido se les aplicó “float:left”, pero también se hubiera podido hacer con “float:left” para el menú y “float:right” para el contenido.

```
#contenedor {  
    width: 700px;  
}  
  
#cabecera { }  
  
#menu {  
    float: left;  
    width: 150px;  
}  
  
#contenido {  
    float: left;  
    width: 550px;  
}  
  
#pie {  
    clear: both;  
}
```



```

<body>
  <div id="contenedor">
    <div id="cabecera">
    </div>
    <div id="menu">
    </div>
    <div id="contenido">
    </div>
    <div id="pie">
    </div>
  </div>
</body>

```

El diseño anterior es de anchura fija, para que se adapte de forma dinámica a la ventana del navegador, se deben aplicar las siguientes reglas CSS:

```

#contenedor {
}

#cabecera {
}

#menu {
  float: left;
  width: 15%;
}

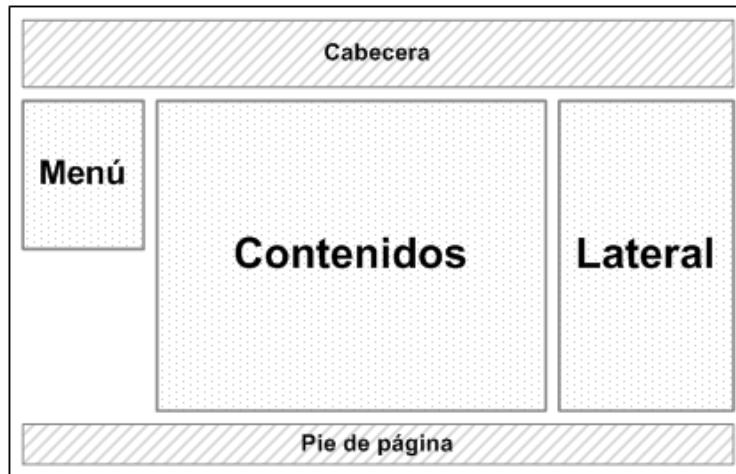
#contenido {
  float: left;
  width: 85%;
}

#pie {
  clear: both;
}

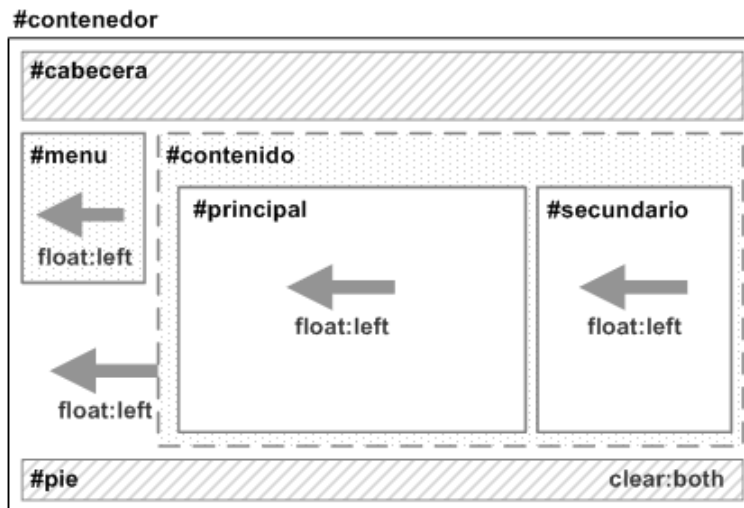
```

Diseño a 3 columnas con cabecera y pie de página:

Además del diseño a dos columnas, el diseño más utilizado es el de tres columnas con cabecera y pie de página. En este caso, los contenidos se dividen en dos zonas diferenciadas: zona principal de contenidos y zona lateral de contenidos auxiliares:



Al igual que con el diseño a dos columnas se usa float left:

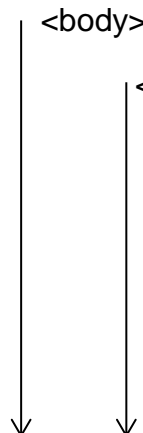


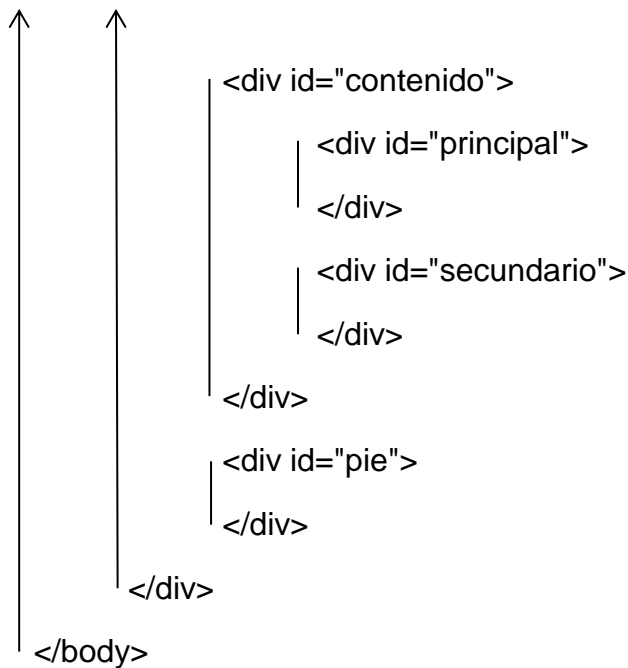
Para la sección de #contenido, se tienen dos sub contenedores: #principal y #secundario, y en vez de usar “float:left” para todos, se hubiera podido usar “float:right” para los contenedores: #contenido y #secundario, y “float:left” para los contenedores #menu y #principal.

Este diseño es de anchura variable (dinámico):

```
#contenedor {  
    }  
  
#cabecera {  
    }
```

```
#menu {  
    float: left;  
    width: 15%;  
}  
  
#contenido {  
    float: left;  
    width: 85%;  
}  
  
#contenido #principal {  
    float: left;  
    width: 80%;  
}  
  
#contenido #secundario {  
    float: left;  
    width: 20%;  
}  
  
#pie {  
    clear: both;  
}  
  
<body>  
    <div id="contenedor">  
        <div id="cabecera">  
        </div>  
        <div id="menu">  
        </div>  
    </div>  
</body>
```





ANCHURAS Y ALTURAS MÁXIMAS Y MÍNIMAS:

Para hacer una página de diseño dinámico, que se ajuste al tamaño de la ventana del usuario, pero que conserve un tamaño mínimo o máximo, para que no se deformen sus contenidos, se usan las siguientes propiedades:

max-width → Permite definir la anchura máxima de un elemento.

min-width → Permite definir la anchura mínima de un elemento.

max-height → Permite definir la altura máxima de un elemento.

min-height → Permite definir la altura mínima de un elemento.

Ej: Para conseguir un diseño de anchura variable pero controlada se usa:

```
#contenedor {  
    min-width: 500px;  
    max-width: 900px;  
}
```

Estas propiedades no están definidas para IE, por lo que se usan así:

➔ max-width equivalente para Internet Explorer:

```
div {  
    max-width: 800px;  
    width: expression(document.body.clientWidth > 801 ? "800px": "auto");}
```

➔ min-width equivalente para Internet Explorer:

```
div {  
    min-width:800px;  
    width: expression(document.body.clientWidth < 801? "800px": "auto" );  
}
```

➔ max-height equivalente para Internet Explorer:

```
div {  
    max-height: 300px;  
    overflow: hidden;  
    height: expression(this.scrollHeight > 301? "300px" : "auto" );  
}
```

➔ min-height equivalente para Internet Explorer:

```
div {  
    min-height:300px;  
    overflow: hidden;  
    height: expression(this.scrollHeight < 301? "300px" : "auto" );  
}
```

ESTRUCTURACIÓN DEL CÓDIGO CSS:

Las reglas CSS de las hojas de estilos complejas se suelen agrupar según su funcionalidad y se suelen incluir en el siguiente orden:

- Estilos básicos (<body>, tipo de letra por defecto, márgenes de , y , etc.)
- Estilos de la estructura o layout (anchura, altura y posición de la cabecera, pie de página, zonas de contenidos, menús de navegación, etc.)
- Enlaces (estilos normales, estilos :hover, etc.)
- Estilos de cada una de las zonas (elementos de la cabecera, titulares y texto de la zona de contenidos, enlaces, listas e imágenes de las zonas laterales, etc.)

PREFIJOS:

Sirven para hacer compatibles con los navegadores algunas características de CSS3.

Ej:

```
<body>
  <div class="rectángulo"></div>
  Chrome y Safari tienen motor -webkit
  Mozilla Firefox motor -moz
  Opera -o
  I.Explorer -ms
</body>

.rectángulo{
  height:200px;
  width:200px;
  background-color:#003333;

  transform:rotate(20deg);

  /*Esta línea rota 20 grados el cuadrado,
  pero si tiene problemas para visualizarse,
  se agregan las siguientes líneas:
  */

  -webkit-transform:rotate(20deg);
  -moz-transform:rotate(20deg);
  -o-transform:rotate(20deg);
}
```

Nota: para evitar usar los prefijos y hacer el CSS mas limpio, se puede usar una librería de JavaScript llamada prefix free.

<http://leaverou.github.io/prefixfree/>

<http://www.cristalab.com/tutoriales/prefijos-automaticos-en-css3-con-prefix-free-c103046/>

FUENTES PERSONALIZADAS:

```
@font-face{
    font-family:"Digital";
    src:url(fonts/dited.ttf),url(fonts/dited.eot),url(fonts/dited.ott);
}

.parrafo-personalizado{
    font-family:"Digital";
}
```

"Digital" es el nombre que se le dio a la fuente. Los archivos dited.ttf .eot y .ott, corresponden a las fuentes en diferentes extensiones para cada navegador.

Nota: para usar muchas fuentes online en nuestra pág.
<https://www.google.com/fonts>

TRUCO MEDIDAS DE TEXTO:

Para usar las medidas em y % en el tamaño de texto, pero pensando en pixeles se usa el siguiente truco:

```
body {
    font-size: 62.5%;
}
```

El tamaño de letra del elemento <body>, y por tanto el tamaño de letra base del resto de elementos de la página, se establece como el 62.5% del tamaño por defecto (16px). Si se calcula el resultado de $16\text{px} * 62.5\%$ se obtienen 10px.

La ventaja de establecer el tamaño de letra del <body> de esa forma es que ahora se pueden utilizar em mientras se piensa en px. En efecto, las siguientes reglas muestran el truco en la práctica:

```
body {
    font-size: 62.5%;
}

h1 {
    font-size: 2em; /* 2em = 2 * 10px = 20px */
}

p {
    font-size: 1.4em; /* 1.4em * 10px = 14px */
}
```

Como el tamaño base son 10px, cualquier valor de em cuya referencia sea el elemento <body> debe multiplicarse por 10, por lo que se puede trabajar con em mientras se piensa en px.

Quitar el misterioso Margen Inferior de una Imagen:

```
<!DOCTYPE html>
<html>
<head>
  <title>Misterioso margen inferior</title>
  <style>
    .contenedor{
      background-color: #939393;
    }
  </style>
</head>
<body>
  <div class="contenedor">
    
  </div>
</body>
</html>
```

1ª forma:

Añadir al estilo de la imagen esto:

```
display: block;
```

2ª forma:

Otra opción es añadir al estilo de la imagen lo siguiente:

```
vertical-align: bottom;
```

3ª forma:

Una última manera, es añadir al estilo del div este código:

```
line-height: 0;
```


RECURSOS ÚTILES:

- Para usar degradados como color de fondo para el body, botones, divs u otros elementos se puede generar el código CSS en línea en:

<http://www.colorzilla.com/gradient-editor/>

- Menús verticales y horizontales avanzados con CSS:

<http://alvit.de/css-showcase/css-navigation-techniques-showcase.php>

- Referencia de CSS:

<http://librosweb.es/referencia/css/>

http://tympanus.net/codrops/css_reference/

- **Complementos o extensiones** Útiles en **Firefox:**

Firebug: para ver errores y muchas otras cosas.

Web Developer: es una barra de herramientas que muestra información de la página, como las medidas, los divs, etc. También nos da la opción de redimensionar la ventana del navegador a cualquier dimensión.

ColorZilla: permite con un cuentagotas, ver el color de cualquier parte de la pág.

MeasureIt: permite medir la altura y anchura de cualquier elemento de la pág.

Screengrab: permite tomar un pantallazo de toda la página completa, solo la parte visible, o un recorte de alguna parte que queramos.

- **Aplicaciones Web:**

Tinypng: sirve para reducir el tamaño de las imágenes PNG sin perder calidad. <https://tinypng.com>

Para imágenes jpg, png, gif y otros formatos se usa **RIOT**, pero ese programa toca instalarlo en el equipo: <http://luci.criosweb.ro/riot/>

Y para los mismos formatos (jpg,png,gif,svg) online: <https://compressor.io/>

Clean CSS: optimiza, ordena, limpia, corrige y reduce el tamaño de las hojas de estilos. <http://www.cleancss.com>

Typetester: permite comparar los tipos de tipografías, así como el tamaño, su color, interlineado y muchas otras propiedades CSS relacionadas con la tipografía y el texto. <http://www.typetester.org>

Stripe Generator: sirve para generar imágenes con rayas que pueden repetirse en todas las direcciones de forma correcta, y por lo tanto se pueden usar como imagen de fondo. <http://www.stripegenerator.com>

Find Icons: para encontrar iconos para usar en aplicaciones o páginas. <http://findicons.com>

- **Sitios Web de Inspiración:**

Web Creme: incluye diariamente varios ejemplos de las mejores páginas diseñadas con CSS y permite realizar búsquedas a partir del color utilizado en la página. <http://www.webcreme.com>

CSS Remix: muestra centenares de páginas diseñadas exclusivamente con CSS y con la posibilidad de puntuar su diseño. <http://cssremix.com>

CSS Zen Garden: es una galería diferente a las tradicionales, pero se ha convertido en una referencia en cuanto a diseños complejos realizados mediante CSS. <http://www.csszengarden.com>

Open Source Web Design: sitio web que ofrece cientos de plantillas gratuitas con posibilidad de utilizarlas libremente en aplicaciones personales y comerciales. <http://www.oswd.org>

- **CSS Sprite Generator:** herramienta online que nos permite generar un sprite, pasándole un archivo .zip, con todas las imágenes individuales para unir en el sprite, también nos genera el código CSS con todas las coordenadas de cada sprite. <http://spritegen.website-performance.org>

- **Colores y paletas de colores:**

<http://html-color-codes.info>

<http://html-color.codes/>

<http://www.color-hex.com/>

