

CURSO .CSS

INTERMEDIO



Autor: Jon Vadillo
www.jonvadillo.com

Contenidos

- Selectores CSS
- Pseudo clases
- Especificidad
- Herencia
- Propiedades abreviadas
- Page Layout
 - Display
 - Float

Selectores de ID y de clase

Permiten dar estilos diferentes a un mismo tipo de elemento en función a su ID o clase.

Sintaxis

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

```
#id {  
  color: blue;  
}
```

Sintaxis

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



```
<p class="subtitulo">Texto rojo</p>
```

```
#titular {  
  color: blue;  
}
```



```
<p id="titular">Texto azul</p>
```

Referenciar desde HTML

Podemos aplicar los estilos definidos con selectores de tipo clase o ID utilizando los atributos `id` y `class`.

En el selector de clase también se puede especificar que solo se aplique a un tipo de elementos concreto:

```
p.intro {...}
```

```
<div id="top">  
  <h1>Chocolate curry</h1>  
  <p class="intro">Introducción</p>  
  <p class="contenido">Contenido</p>  
</div>
```

Ejercicio 3

Aprendiendo CSS

Lorem ipsum dolor sit amet consectetur

Lorem ipsum dolor sit amet consectetur adipiscing elit ad **pretium consequat** orci, ridiculus aenean mus faucibus [mauris quam](#) est sociis cursus lacinia. Malesuada pellentesque laoreet nascetur mauris mus, accumsan **habitasse** potenti nec libero, duis sapien viverra imperdiet.

Placerat ullamcorper parturient dui iaculis porttitor **malesuada cum sociosqu consequat** ultricies, mi scelerisque platea eu facilisi urna curabitur **justo** eleifend quam.

Magnis ante inceptos euismod potenti etiam enim non vehicula.

Lorem ipsum dolor sit amet consectetur

Aliquet magna **nostra duis facilisis risus** suscipit cum scelerisque, turpis urna at laoreet luctus pretium mi, semper sociis sapien mus parturient ac justo.

At elementum penatibus etiam **justo** semper blandit pretium [turpis nostra](#), euismod pharetra congue porttitor fusce ad per vestibulum class, torquent aenean ut [facilisis](#) eros nec ante sagittis. At iaculis curae aliquam **vitae sem id semper** dictum mus quisque, vivamus pharetra molestie praesent vestibulum nisl interdum congue dictumst penatibus urna, enim magnis porta a auctor non gravida aliquet sollicitudin.

Lorem ipsum dolor sit amet consectetur

Primis mattis vivamus augue rhoncus dictum nunc fermentum cras bibendum magnis leo risus duis ridiculus phasellus, massa sollicitudin hac posuere vitae consequat nostra quis pulvinar



Aprendiendo CSS

Lorem ipsum dolor sit amet consectetur

Lorem ipsum dolor sit amet consectetur adipiscing elit ad **pretium consequat** orci, ridiculus aenean mus faucibus [mauris quam](#) est sociis cursus lacinia. Malesuada pellentesque laoreet nascetur mauris mus, accumsan **habitasse** potenti nec libero, duis sapien viverra imperdiet.

Placerat ullamcorper parturient dui iaculis porttitor **malesuada cum sociosqu consequat** ultricies, mi scelerisque platea eu facilisi urna curabitur **justo** eleifend quam.

Magnis ante inceptos euismod potenti etiam enim non vehicula.

Lorem ipsum dolor sit amet consectetur

Aliquet magna **nostra duis facilisis risus** suscipit cum scelerisque, turpis urna at laoreet luctus pretium mi, semper sociis sapien mus parturient ac justo.

At elementum penatibus etiam **justo** semper blandit pretium [turpis nostra](#), euismod pharetra congue porttitor fusce ad per vestibulum class, torquent aenean ut [facilisis](#) eros nec ante sagittis. At iaculis curae aliquam **vitae sem id semper** dictum mus quisque, vivamus pharetra molestie praesent vestibulum nisl interdum congue dictumst penatibus urna, enim magnis porta a auctor non gravida aliquet sollicitudin.

Lorem ipsum dolor sit amet consectetur

Primis mattis vivamus augue rhoncus dictum nunc fermentum cras bibendum magnis leo risus duis ridiculus phasellus, massa sollicitudin hac posuere vitae consequat nostra quis pulvinar odio ultrices quam commodo.

Phasellus dictum bibendum lacus praesent ultricies felis facilisi potenti mus ultrices, sociosqu nunc pharetra aenean conubia maecenas [turpis nostra](#) habitant tellus dis velit, cursus varius mattis suscipit ridiculus blandit natoque ad faucibus.

Ejercicio 4

HTML & CSS: Curso práctico avanzado

Aunque los inicios de [Internet](#) se remontan a los años sesenta, no ha sido hasta los años noventa cuando, gracias a la Web, se ha extendido su uso por todo el mundo. En pocos años, la Web ha evolucionado enormemente: se ha pasado de páginas sencillas, con pocas imágenes y contenidos estáticos que eran visitadas por unos pocos usuarios a *páginas complejas, con contenidos dinámicos que provienen de bases de datos y que son visitadas por miles de usuarios al mismo tiempo*.

Todas las páginas están internamente construidas con la misma tecnología, con el *Lenguaje de marcas de hipertexto* (Hypertext Markup Language, [HTML](#)) y con las *Hojas de estilo en cascada* (Cascading Style Sheets, [CSS](#)).

Este libro es *adecuado para cualquiera que tenga interés en aprender a desarrollar sus propias páginas web*. No son necesarios conocimientos previos para aprender con este libro, lo único que es necesario es saber utilizar un ordenador y saber navegar por la Web.



HTML & CSS: Curso práctico avanzado

Aunque los inicios de [Internet](#) se remontan a los años sesenta, no ha sido hasta los años noventa cuando, gracias a la Web, se ha extendido su uso por todo el mundo. En pocos años, la Web ha evolucionado enormemente: se ha pasado de páginas sencillas, con pocas imágenes y contenidos estáticos que eran visitadas por unos pocos usuarios a *páginas complejas, con contenidos dinámicos que provienen de bases de datos y que son visitadas por miles de usuarios al mismo tiempo*.

Todas las páginas están internamente construidas con la misma tecnología, con el *Lenguaje de marcas de hipertexto* (Hypertext Markup Language, [HTML](#)) y con las *Hojas de estilo en cascada* (Cascading Style Sheets, [CSS](#)).

Este libro es *adecuado para cualquiera que tenga interés en aprender a desarrollar sus propias páginas web*. No son necesarios conocimientos previos para aprender con este libro, lo único que es necesario es saber utilizar un ordenador y saber navegar por la Web.

Pseudoclasses

Sirven para especificar un estado especial del elemento seleccionado.

Sintaxis de pseudoclases

```
/* Selecciona cualquier elemento <a>
cuando está "mantenido (hovered)" */
a:hover {
    color: orange;
}
/* Selecciona cualquier <input>
deshabilitado */
input:disabled {
    background: #ccc;
}
/* Selecciona cualquier <input>
requerido */
input:required {
    border: 1px dashed red;
}
/* Coincide con cualquier
checked/selected radio, checkbox, u
option */
:checked {
    margin-left: 25px;
    border: 1px solid blue;
}
/* Elemento Checkbox, cuando está
marcado */
input[type="checkbox"]:checked{
    box-shadow: 0 0 0 3px orange;
}
```

```
selector:pseudoclase {
    propiedad: valor;
}
```

Todos los selectores de atributo en:

https://developer.mozilla.org/es/docs/Web/CSS/Selectores_atributo

Listado de pseudoclases

<code>:active</code>	<code>:indeterminate</code>	<code>:only-child</code>
<code>:checked</code>	<code>:in-range</code>	<code>:only-of-type</code>
<code>:default</code>	<code>:invalid</code>	<code>:optional</code>
<code>:dir()</code>	<code>:lang()</code>	<code>:out-of-range</code>
<code>:disabled</code>	<code>:last-child</code>	<code>:read-only</code>
<code>:empty</code>	<code>:last-of-type</code>	<code>:read-write</code>
<code>:enabled</code>	<code>:left</code>	<code>:required</code>
<code>:first</code>	<code>:link</code>	<code>:right</code>
<code>:first-child</code>	<code>:not()</code>	<code>:root</code>
<code>:first-of-type</code>	<code>:nth-child()</code>	<code>:scope</code>
<code>:fullscreen</code>	<code>:nth-last-child()</code>	<code>:target</code>
<code>:focus</code>	<code>:nth-last-of-type()</code>	<code>:valid</code>
<code>:hover</code>	<code>:nth-of-type()</code>	<code>:visited</code>


Hands on!

- Crea un nuevo Pen en CodePen (<https://codepen.io>) con un listado de enlaces que cambien su color de letra y color de fondo cuando el usuario pasa el ratón por encima.

Más específico = Más prioridad


- Si un elemento tiene más de una regla, CSS da mayor prioridad a la regla que tiene el selector más específico.
- Un selector ID es más específico que un selector de class, que a su vez es más específico que un selector de etiquetas.

Selector	a	b	c	d	Specificity
h1				1	0,0,0,1
.foo			1		0,0,1,0
#bar		1			0,1,0,0




a
1 x element selector

Sith power: 0,0,1




p a
2 x element selectors

Sith power: 0,0,2



.foo
1 x class selector *

Sith power: 0,1,0



a.foo
1 x element selector
1 x class selector

Sith power: 0,1,1



p a.foo
2 x element selectors
1 x class selector

Sith power: 0,1,2



.foo .bar
2 x class selectors

Sith power: 0,2,0



p.foo a.bar
2 x element selectors
2 x class selectors

Sith power: 0,2,2



#foo
1 x id selector

Sith power: 1,0,0



a#foo
1 x element selector
1 x id selector

Sith power: 1,0,1



.foo a#bar
1 x element selector
1 x class selector
1 x id selector

Sith power: 1,1,1




.foo .foo #foo
2 x class selectors
1 x id selector

Sith power: 1,2,0




style
1 x style attribute

Sith power: 1,0,0,0



* Same specificity
class selector =
attribute selector =
pseudo-classes



!important

- **p** tiene especificidad de 1 (1 HTML selector)
- **div p** tiene especificidad de 2 (2 HTML selectors, 1+1)
- **.tree** tiene especificidad de 10 (1 class selector)
- **div p.tree** tiene especificidad de 12 (2 HTML selectors + a class selector, 1+1+10)
- **#baobab** tiene especificidad de 100 (1 id selector)
- **body #content .alternative p** tiene especificidad de 112 (HTML selector + id selector + class selector + HTML selector, 1+100+10+1)

```
/* specificity: 0001 */  
a {  
    background-color: red;  
}
```

```
/* specificity: 0011 */  
.outer a {  
    background-color: red;  
}
```

```
/* specificity: 0101 */  
#outer a {  
    background-color: red;  
}
```

```
/* specificity: 0111 */  
#outer .inner a {  
    background-color: red;  
}
```

```
/* specificity: 0201 */  
#outer #inner a {  
    background-color: blue;  
}
```

```
/* specificity: 0104 */  
#outer div ul li a {  
    color: yellow;  
}
```

```
/* specificity: 0113 */  
#outer div ul .nav a {  
    color: white;  
}
```

CSS Specificity by example

Test paragraph one

Test paragraph two

Test paragraph three

Test paragraph four

Test paragraph one.

Test paragraph two.

Test paragraph three.

<https://codepen.io/jonvadillo/pen/mdJyzja>

¿Y si tiene la misma especificidad?

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

```
/* Esta regla gana sobre la  
anterior */
```

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

El siguiente factor que determina qué regla vence es el orden del código:

Las últimas reglas prevalecen sobre las primeras

!important

```
.mi-clase {  
    color: red!important;  
}
```

Herencia

Aparte de conocer los selectores y su especificidad, es importante entender la herencia para conocer qué estilos se aplicarán a los elementos.

Existen propiedades CSS que se heredan a los descendientes de forma automática (p.ej. font-family)

Herencia

This paragraph has *emphasized text* in it.

Another parahtaph without border in it.

<https://codepen.io/jonvadillo/pen/oNXgQvP>

Propiedades abreviadas

Es posible agrupar algunas propiedades en una única, indicando los valores separados por un espacio:

```
.principal {  
  border: 2px solid #f00;  
}
```

Margin & Padding

```
#div {  
margin-top: 0;  
margin-right: 5px;  
margin-bottom: 10px;  
margin-left: 15px;  
(auto, 0, px, pt, em or %)  
}
```

```
#div {  
margin:0 5px 10px 15px;  
(top right bottom left)  
}
```

```
#div {  
margin-top: 10px;  
margin-right: 20px;  
margin-bottom: 0;  
margin-left: 20px;  
}
```

```
#div {  
margin:10px 20px 0;  
(top right/left bottom)  
}
```

```
#div {  
margin-top: 0;  
margin-right: auto;  
margin-bottom: 0;  
margin-left: auto;  
}
```

```
#div {  
margin:0 auto;  
(top/bottom left/right)  
}
```

```
#div {  
margin-top: 50px;  
margin-right: 50px;  
margin-bottom: 50px;  
margin-left: 50px;  
}
```

```
#div {  
margin:50px;  
(top/right/bottom/left)  
}
```

Border

```
#div {  
border-width: 5px;  
(thin, thick, medium or set value) (default = medium)  
border-style: dotted;  
(solid, dashed, dotted, double, etc) (default = none)  
border-color: blue;  
(named, hex, rgb or 0-255) (default = value of elements/  
elements parent color property)  
}
```

```
#div {  
border:5px dotted blue;  
}
```

```
#div {  
border-right-width: 2px;  
border-right-style: solid;  
border-right-color:  
#666666;  
}
```

```
#div {  
border-right:2px solid #666;  
}
```

```
#div {  
border-top-width: 3px;  
border-right-width: 2px;  
border-bottom-width: 3px;  
border-left-width: 2px;  
}
```

```
#div {  
border-width:3px 2px;  
}
```

Background

```
#div {  
background-color: #CCCCCC;  
(named, hex, rgb or 0-255) (default = transparent)  
background-image: url(images/bg.gif);  
(url or none) (default = none)  
background-repeat: no-repeat;  
(repeat, repeat-x, repeat-y or no-repeat) (default = repeat)  
background-attachment: scroll;  
(fixed or scroll) (default = scroll)  
background-position: top left;  
(top, right, left, bottom or center) (default = 0% 0%)  
}
```

```
#div {  
background: #CCC url(images/bg.gif) no-repeat 0 0;  
}
```

Font

```
#div {  
font-family: Verdana, Arial, Helvetica;  
(Verdana, Arial, "Times New Roman", etc) (default = browse based)  
font-size: 12px;  
(xx-small, medium, x-large, set value, etc) (default = medium)  
font-weight: bold;  
(normal, bold, bolder, lighter, 100-900 or inherit) (default = normal)  
font-style: italic;  
(normal, italic or oblique) (default = normal)  
font-variant: normal;  
(normal or small-caps) (default = normal)  
line-height: 1.5px;  
(normal, px, pt, em or %) (default = normal)  
}
```

```
#div {  
font: italic bold 12px/1.5px Verdana, Arial, Helvetica;  
}
```

List

```
#div {  
list-style-image: url(images/bullet.gif);  
(url or none) (default = none)  
list-style-position: inside;  
(inside or outside) (default = outside)  
list-style-type: square;  
(circle, disc, square, etc) (default = disc)  
}
```

```
#div {  
list-style: square inside url(images/bullet.gif);  
}
```

Color

Aqua: #00ffff to #0ff
Black: #000000 to #000
Blue: #0000ff to #00f
Dark Grey: #666666 to #666
Fuchsia: #ff00ff to #f0f
Light Grey: #cccccc to #ccc
Lime: #00ff00 to #0f0
Orange: #ff6600 to #f60
Red: #ff0000 to #f00
White: #ffffff to #fff
Yellow: #ffff00 to #ff0

Display

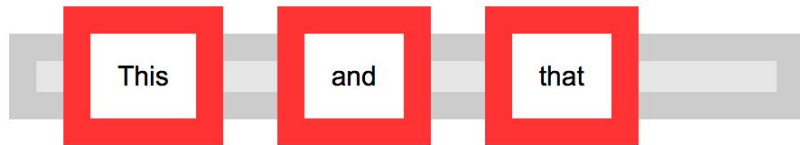
Los elementos tienen un tipo de caja por defecto que afecta a su presentación, por ejemplo block o inline.

La propiedad display permite alterar el tipo de representación del elemento

```
div {  
  display: inline;  
  display: inline-block;  
  display: block;  
  display: none;  
}
```


display: inline

- Es el valor por defecto de la mayoría de elementos.
- Acepta margin y padding, pero no empuja a otros elementos verticalmente.
- No acepta las propiedades height y width.



Pellentesque *inline element* morbi tristique senectus
Donec eu libero sit amet quam egestas semper. Aenean

Pellentesque *inline element* morbi tristique senectus et netus et
malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae,
ultrices eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas
semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

display: inline-block

- Al igual que inline, se posiciona de forma natural en la misma línea.
- La mayor diferencia es que adquiere otras propiedades de block, como permitir especificar height y width

Pellentesque *inline block* *inline block* *inline block* morbi tristique

senectus et netus et malesuada fames ac turpis egestas. Vestibulum tortor quam, feugiat vitae, ultricies eget, tempor sit amet, ante. Donec eu libero sit amet quam egestas semper. Aenean ultricies mi vitae est. Mauris placerat eleifend leo.

The "display" property

Titulo normal y corriente

There are plenty of more exotic display values, such as list-item and table. Here is an exhaustive list. We'll discuss inline-block and flex later on.

Titulo solo inline

As I mentioned, every element has a default display type. However, you can always override this! Though it wouldn't make sense to make an inline div, you can use this to customize the display of elements that have particular semantics. A common example is making inline li elements for horizontal menus.

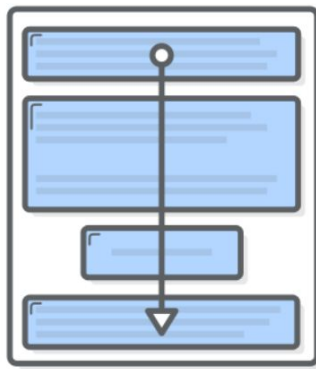
Titulo inline-block

As I mentioned, every element has a default display type. However, you can always override this! Though it wouldn't make sense to make an inline div, you can use this to customize the display of elements that have particular semantics. A common example is making inline li elements for horizontal menus.

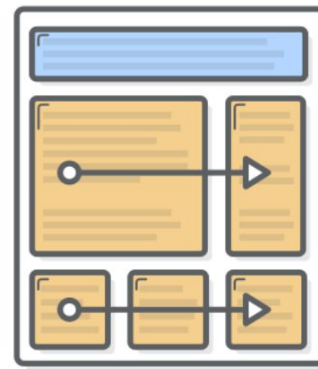
<https://codepen.io/jonvadillo/pen/dyoPQYw>

Float

Hasta ahora los elementos de tipo bloque seguían el flujo vertical de la página. La propiedad CSS **float** permite posicionar elementos de tipo bloque unos junto a otros.



VERTICAL FLOW

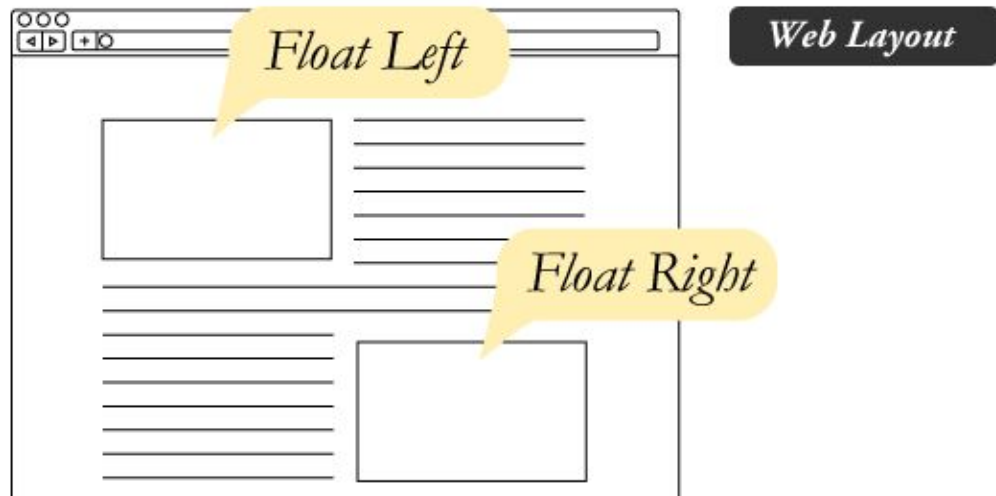


HORIZONTAL FLOW

Float

Hay 4 valores distintos:

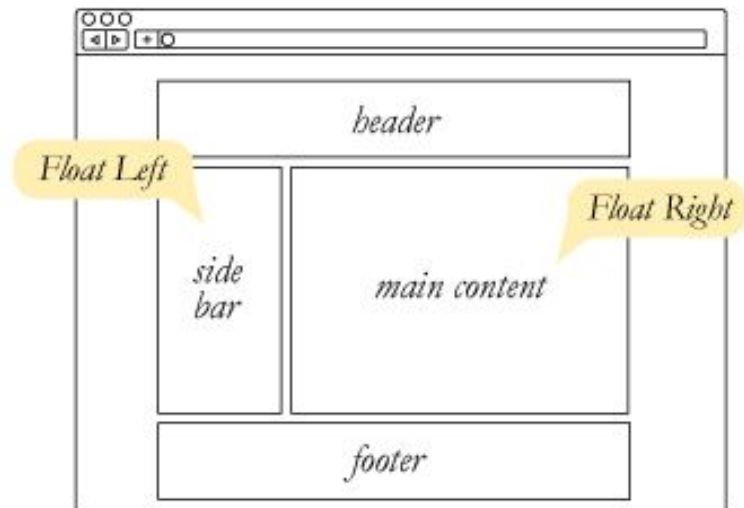
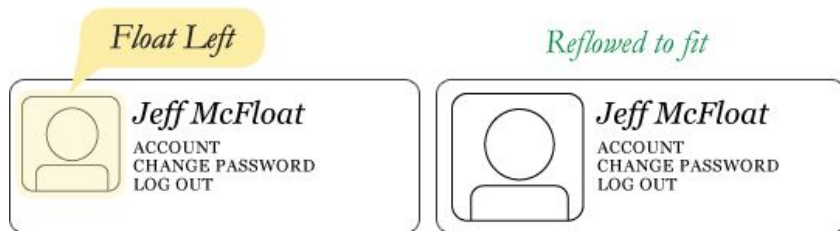
- **left y right** para llevar los elementos en esas direcciones.
- **none**: el valor por defecto (no altera el comportamiento)
- **inherit**: hereda el mismo valor que el padre.



source: <https://css-tricks.com/>

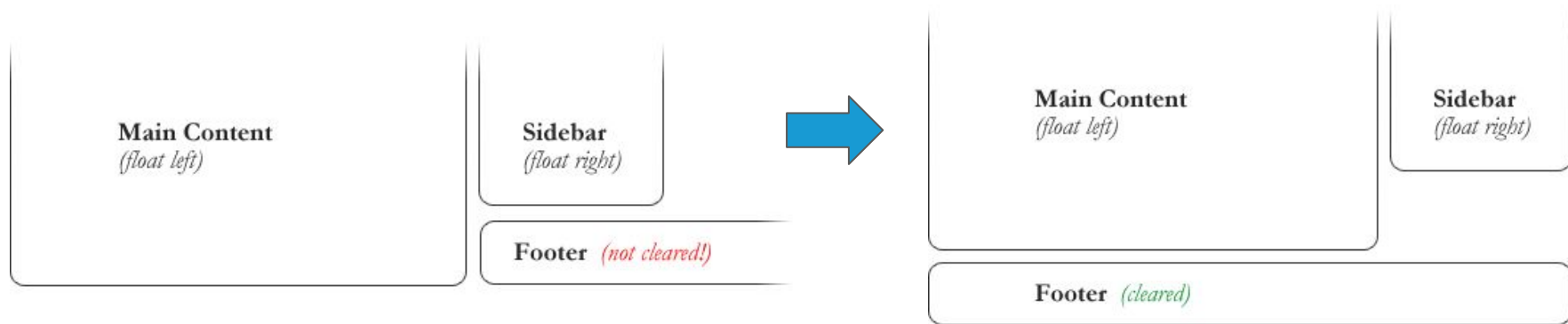
¿Cuál es su uso?

- Se pueden utilizar para construir layouts completos o pequeñas estructuras.

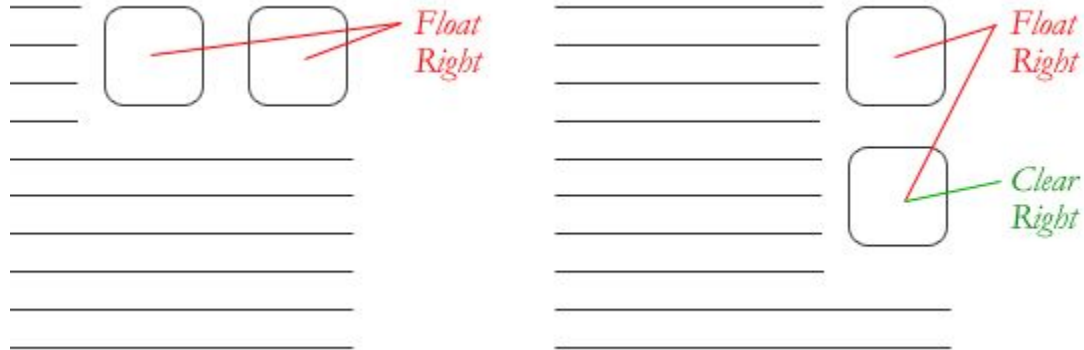


El anti-float: clear

Indica qué lados de la caja no deben ser adyacentes a un elemento posicionado con float. Sus posibles valores son: **left**, **right**, **both** y **none**



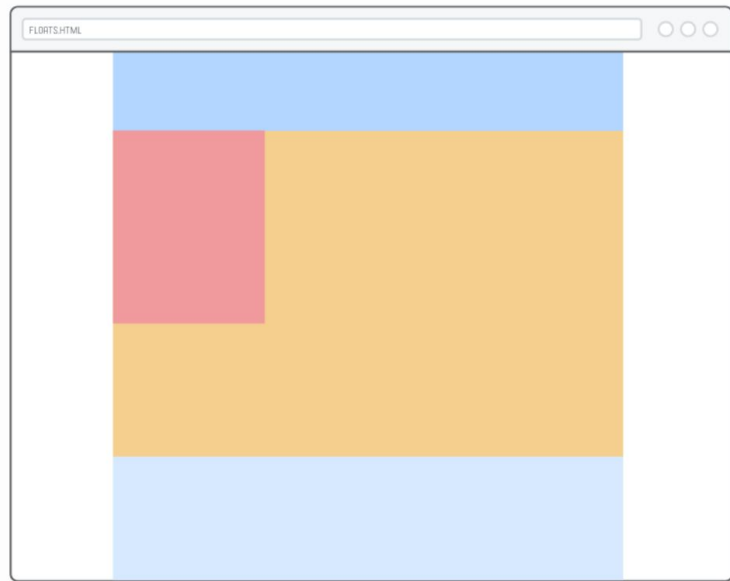
El anti-float: clear



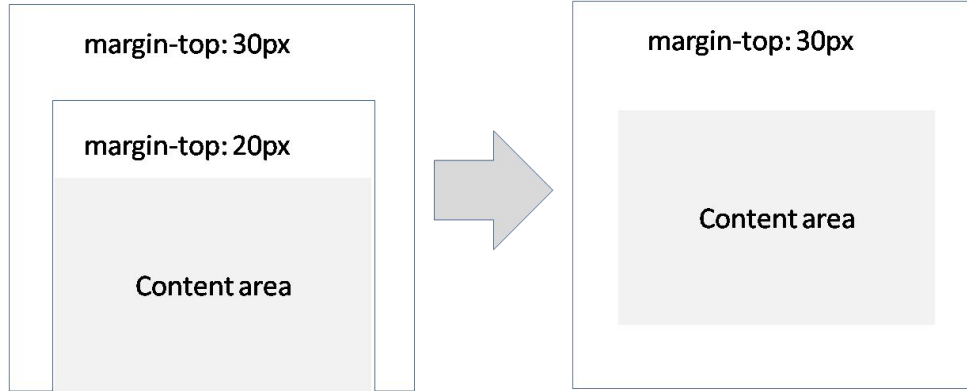
source: <https://css-tricks.com/>

Centrar página

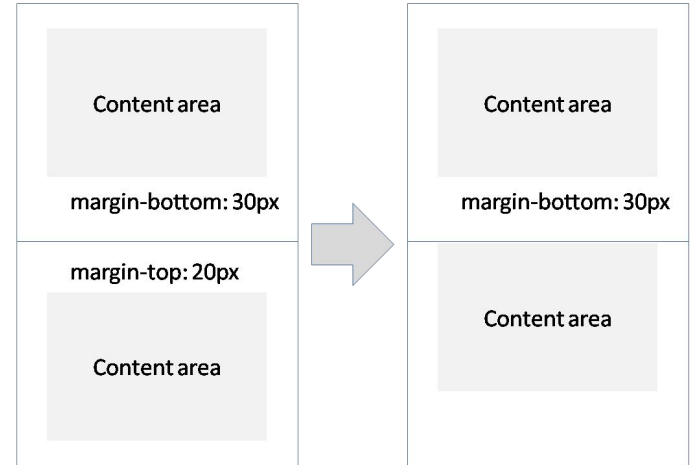
- Normalmente tendremos un ancho fijo de página (`max-width: 900px`) y centraremos el contenedor de nuestro contenido (`margin: 0 auto`).



Problemas con márgenes: Collapsing margins



Margin collapsing



Problemas con márgenes: Collapsing margins

- Los márgenes Top y Bottom de los bloques a veces se “*colapsan*” en un solo margen, cuyo tamaño es el mayor de los márgenes
- Los márgenes de flotantes y elementos con posición absoluta nunca colapsan.
- Cuando los elementos tienen borde o padding, no se solapan.
- Una opción para solucionarlo es hacer que el elemento sea “float” y ocupe el 100% (width) o convertirlo a tipo inline-block

Sources

- Mozilla MDN Web Docs: <https://developer.mozilla.org/>
- HTML Dog: <http://htmldog.com>
- Interneting is Hard: <https://internetingishard.com/>