

CSS 2.0

POR ARKAITZ GARRO

I.
INTRO-
DUCCIÓN

2.
SELEC-
TORES

3.
UNIDADES
DE
MEDIDA Y
COLORES

4.
MODELO
DE
CAJAS

5.
POSICIO-
NAMIENTO

6.
TEXTO

7.
ENLACES

8.
LISTAS

9.
FORMU-
LARIOS

10.
TABLAS

II.
LAYOUT
BÁSICO

12.
OTROS

I.
INTRO-
DUCCIÓN

2.
SELEC-
TORES

3.
UNIDADES
DE
MEDIDA Y
COLORES

4.
MODELO
DE
CAJAS

5.
POSICIO-
NAMIENTO

6.
TEXTO

7.
ENLACES

8.
LISTAS

9.
FORMU-
LARIOS

10.
TABLAS

II.
LAYOUT
BÁSICO

12.
OTROS

CSS es un lenguaje de hojas de estilos creado para controlar el aspecto de los documentos HTML.

Una vez creados los contenidos en HTML, se utiliza el lenguaje CSS para definir el aspecto de cada elemento:

Color, tamaño y tipo de letra del texto, separación horizontal y vertical entre elementos, posición de cada elemento dentro de la página, etc.

INCLUIR CSS EN UN DOCUMENTO

```
<html>

  <head>

    <title>Documento HTML</title>

    <link rel="stylesheet" type="text/css" href="/css/normalize.css"/>
    <link rel="stylesheet" type="text/css" href="/css/main.css"/>

  </head>

  <body>

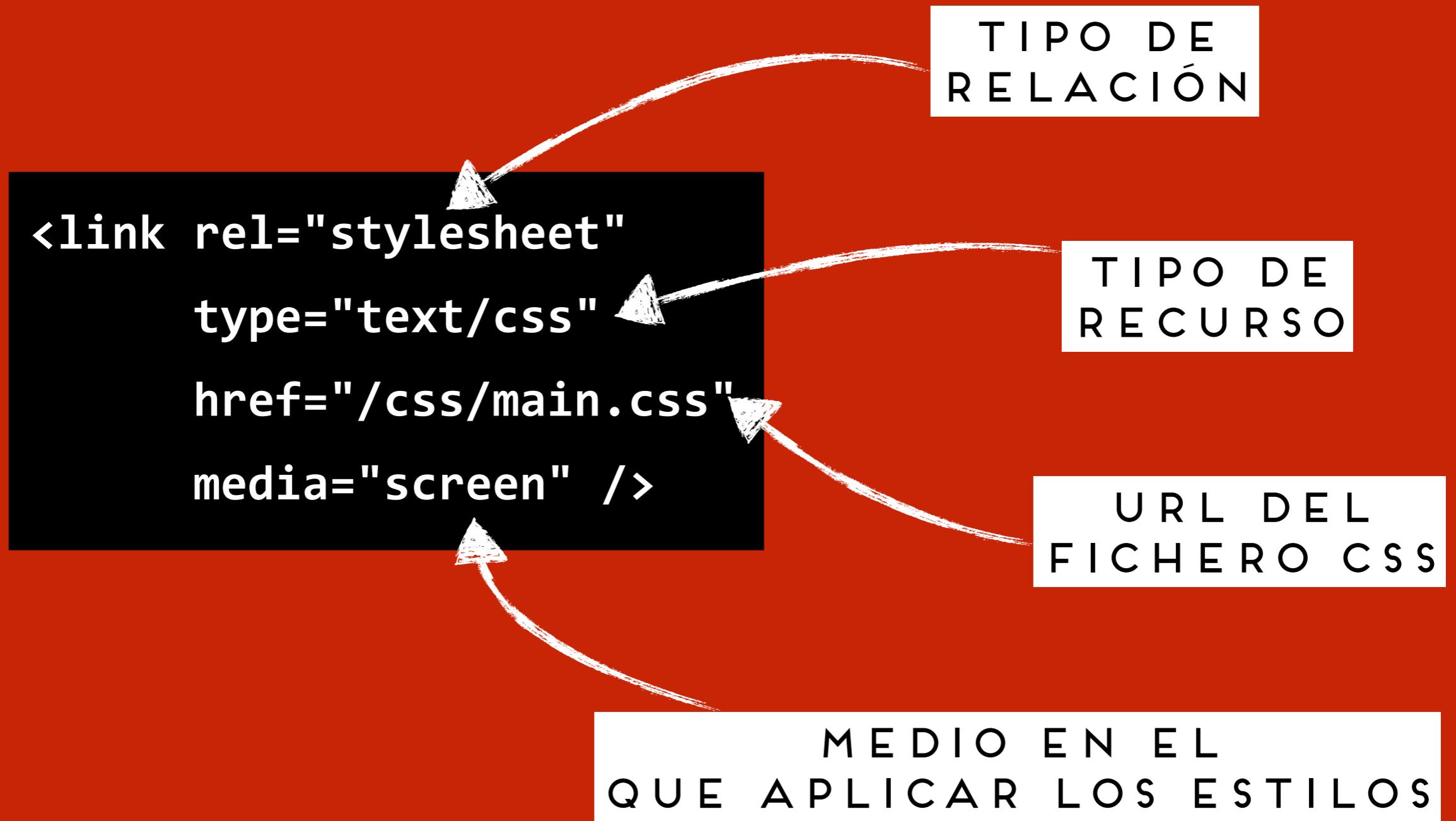
    <p><strong>HTML</strong> (HyperText Markup Language) es un lenguaje de marcas usado, de manera predominante, para la elaboración de páginas web.</p>

  </body>

</html>
```

INCLUIR CSS EN UN DOCUMENTO



IMPORTAR FICHEROS CSS

Dentro de un fichero CSS, es posible hacer referencia o “importar” otros ficheros CSS.

```
@import '/css/colores.css';  
  
@import "/css/colores.css";  
  
@import url('/css/colores.css');  
  
@import url("/css/colores.css");
```

GLOSARIO



TIPOS DE MEDIOS CSS

all

braille

embossed

handled

print

projection

screen

speech

tty

tv

TIPOS DE MEDIOS CSS

all

braille

embossed

handled

print

projection

screen

speech

tty

tv

T I P O S D E M E D I O S C S S

Las reglas `@media` permiten indicar de forma directa el medio o medios en los que se aplicarán los estilos incluidos en la regla.

Para especificar el medio en el que se aplican los estilos, se incluye su nombre después de `@media`.

```
@media print {  
    body { font-size: 10pt }  
}  
  
@media screen {  
    body { font-size: 13px }  
}  
  
@media screen, print {  
    body { line-height: 1.2 }  
}
```

COMENTARIOS

Como en cualquier otro lenguaje, CSS permite añadir comentarios al código.

```
/* Este es un comentario en CSS */
```

```
/* Este es un  
comentario CSS de varias  
lineas */
```

I.
INTRO-
DUCCIÓN

2.
SELEC-
TORES

3.
UNIDADES
DE
MEDIDA Y
COLORES

4.
MODELO
DE
CAJAS

5.
POSICIO-
NAMIENTO

6.
TEXTO

7.
ENLACES

8.
LISTAS

9.
FORMU-
LARIOS

10.
TABLAS

II.
LAYOUT
BÁSICO

12.
OTROS

Para trabajar con CSS, es imprescindible conocer y dominar los SELECTORES.

En una regla CSS, la declaración indica "*qué hay que hacer*" y el selector indica "*a quién hay que aplicarlo*".

A un mismo elemento HTML se le pueden aplicar **varias reglas CSS** y cada regla CSS puede aplicarse a un número ilimitado de elementos.

SELECTOR UNIVERSAL

Selecciona **todos** los elementos de la página.

```
* {  
    padding: 0;  
    margin: 0;  
    color: #FFFFFF;  
}
```

SELECTOR DE TIPO

Selecciona todos los elementos de la página cuya **etiqueta HTML coincide** con el valor del selector.

```
h1 {  
    color: red;  
}
```

```
<h1 class="intro" id="header">CSS2</h1>
```

AGRUPAR SELECTORES

Si se quiere aplicar los mismos estilos a dos etiquetas diferentes, se pueden **encadenar** los selectores.

Para ello, se incluyen todos los selectores separados por una coma (,).

```
h1, h2, h3 {  
    color: #8A8E27;  
    font-weight: normal;  
    font-family: Arial, sans-serif;  
}
```

AGRUPAR SELECTORES

Es habitual agrupar las propiedades **comunes** de varios elementos en una única regla CSS y posteriormente definir las propiedades **específicas** de esos mismos elementos.

```
h1, h2, h3 {  
    color: #8A8E27;  
    font-weight: normal;  
    font-family: Arial, sans-serif;  
}  
  
h1 { font-size: 2em; }  
h2 { font-size: 1.5em; }  
h3 { font-size: 1.2em; }
```

SELECTOR DESCENDENTE

Selecciona los elementos que se encuentran **dentro de otros elementos.**

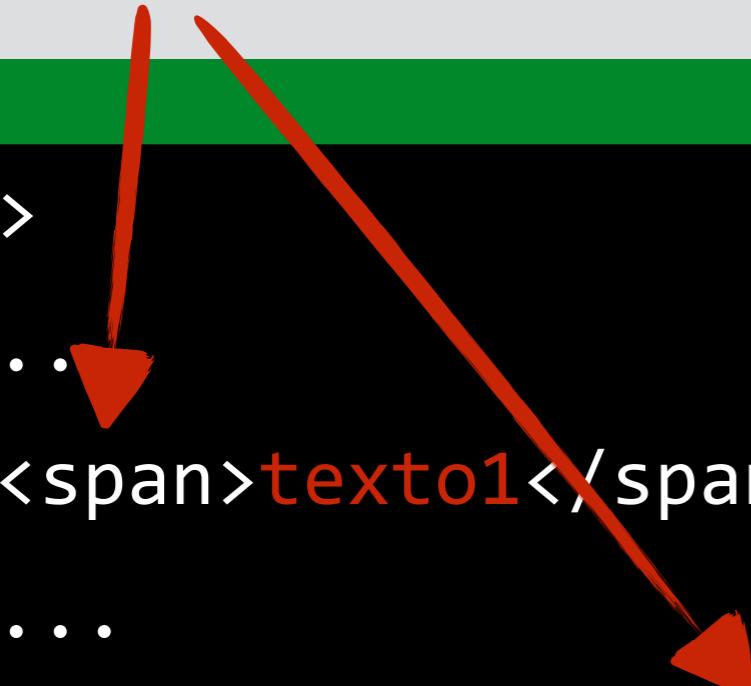
Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del otro elemento.

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

SELECTOR DESCENDENTE

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

```
<p>
...
<span>texto1</span>
...
<a href="">...<span>texto2</span></a>
...
</p>
<span> . . . </span>
```



S E L E C T O R D E S C E N D E N T E

Los selectores descendentes permiten aumentar la precisión del selector de tipo o etiqueta. Así, utilizando el selector descendente es posible aplicar diferentes estilos a los elementos del mismo tipo.

```
p span { color: red; }  
h1 span { color: blue; }
```

SELECTOR DESCENDENTE

La sintaxis formal del selector descendente se muestra a continuación:

```
selector1 selector2 selector3 ... selectorN
```

Los selectores descendentes siempre están formados por dos o más selectores separados entre sí por espacios en blanco. **El último selector** indica el elemento sobre el que se aplican los estilos.

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

SELECTOR DE CLASE

Selecciona los elementos cuyo atributo **class** contiene la clase indicada en el selector.

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

```
<body>
  <p class="fleft destacado">Lorem ipsum dolor sit amet...
  <p>Nunc sed lacus et est adipiscing accumsan...
  <p>Class aptent taciti sociosqu ad litora...
</body>
```

El selector `.destacado` se interpreta como
"cualquier elemento de la página cuyo atributo class sea igual a destacado".

SELECTOR DE CLASE

Los selectores de clase anteponen el carácter . al valor del atributo **class** que se quiere seleccionar.

Este tipo de selectores son los más utilizados junto con los selectores de ID. La principal característica de este selector es que es posible aplicar el mismo esto a diferentes elementos.

```
<body>
  <p class="destacado">Lorem ipsum dolor sit amet...</p>
  <p>Nunc sed <a href="#" class="destacado">est adipiscing</a></p>
  <p>Class aptent <em class="destacado">sociosqu ad</em></p>
</body>
```

S E L E C T O R D E C L A S E

En CSS disponemos de varias maneras de **restringir** los elementos a los que aplicamos los estilos. Una de ellas, es limitar la aplicación de una clase a un elemento en concreto.

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

SELECTOR DE CLASE

No debemos confundir el anterior selector con los siguientes:

```
/* Todos los elementos de tipo "p" con atributo class="aviso" */
p.aviso { ... }

/* Todos los elementos con atributo class="aviso" que estén dentro
   de cualquier elemento de tipo "p" */
p .aviso { ... }

/* Todos los elementos "p" de la página y todos los elementos con
   atributo class="aviso" de la página */
p, .aviso { ... }
```

SELECTOR DE CLASE

Por último, es posible aplicar los estilos de **varias clases** CSS sobre un **mismo elemento**. La sintaxis es similar, pero los diferentes valores del atributo **class** se separan con espacios en blanco.

```
<p class="especial destacado error">Párrafo de texto...</p>
```

```
.error { color: red; }  
.destacado { font-size: 15px; }  
.especial { font-weight: bold; }  
.error.destacado { color: blue; }
```

S E L E C T O R D E I D

En ocasiones, es necesario aplicar estilos CSS a un **único elemento** de la página. El **selector de ID** permite seleccionar un elemento de la página a través del valor de su **atributo id**.

La sintaxis de los selectores de ID es muy parecida a la de los selectores de clase, salvo que se utiliza el símbolo de la almohadilla (#) en vez del punto (.) como prefijo del nombre de la regla CSS:

SELECTOR DE ID

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

```
<p>Primer párrafo</p>
<p id="destacado">Segundo párrafo</p>
<p>Tercer párrafo</p>
```

COMBINACIÓN DE SELECTORES

CSS permite la **combinación** de uno o más tipos de selectores para **restringir el alcance** de las reglas CSS.

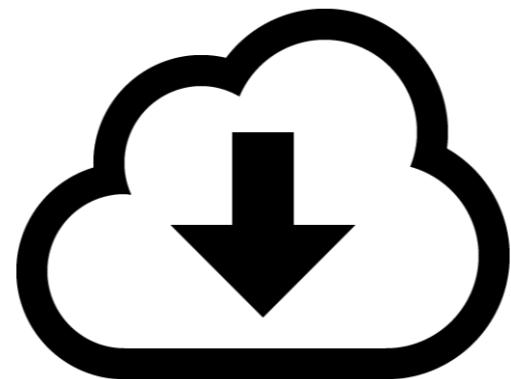
```
.aviso .especial { ... }
```

```
div.aviso span.especial { ... }
```

```
ul#menuPrincipal li.destacado a#inicio { ... }
```

E J E R C I C I O

A partir del código HTML y CSS que se proporcionan, añadir los selectores CSS que faltan para aplicar los estilos deseados. Cada regla CSS incluye un comentario en el que se explica los elementos a los que debe aplicarse.



www.arkaitzgarro.com/css2/snippets/cap14/ej01.zip

I.
INTRO-
DUCCIÓN

2.
SELEC-
TORES

3.
UNIDADES
DE
MEDIDA Y
COLORES

4.
MODELO
DE
CAJAS

5.
POSICIO-
NAMIENTO

6.
TEXTO

7.
ENLACES

8.
LISTAS

9.
FORMU-
LARIOS

10.
TABLAS

II.
LAYOUT
BÁSICO

12.
OTROS

Muchas de las propiedades de CSS permiten indicar MEDIDAS y COLORES en sus valores. Además, CSS es tan flexible que permite indicar las medidas y colores de muchas formas diferentes.

Las medidas en CSS se emplean, entre otras, para definir la altura, anchura y márgenes de los elementos y para establecer el tamaño de letra del texto.

Por otra parte, los colores de CSS se pueden emplear para indicar el color de un texto, fondos, bordes o degradados.

UNIDADES DE MEDIDA

Todas las medidas se indican como un valor **numérico entero** o **decimal** seguido de una unidad de medida.

Absolutas

Establecen de forma completa el valor de una medida, por lo que su valor real es directamente el valor indicado.

Relativas

Definen su valor en relación con otra medida, por lo que para obtener su valor real, se debe realizar alguna operación.

UNIDADES ABSOLUTAS

Una medida indicada mediante unidades absolutas está **completamente definida**, ya que su valor **no depende** de otro valor de referencia.

in

cm

mm

pt

pc

La principal ventaja de las unidades absolutas es que su valor es directamente el valor que se debe utilizar, sin necesidad de realizar cálculos intermedios. Su principal **desventaja** es que son muy poco flexibles y no **se adaptan** fácilmente a los diferentes medios.

UNIDADES ABSOLUTAS

Algunos ejemplos de utilización de unidades absolutas:

```
/* 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 <span> deben tener un tamaño de letra de 1 pica */
span { font-size: 1pc }
```

UNIDADES RELATIVAS

Su valor siempre **depende** de otro valor. Son las **más utilizadas** en el diseño web por la **flexibilidad** con la que se adaptan a los diferentes medios.

em

Relativa respecto del tamaño de letra del elemento.

ex

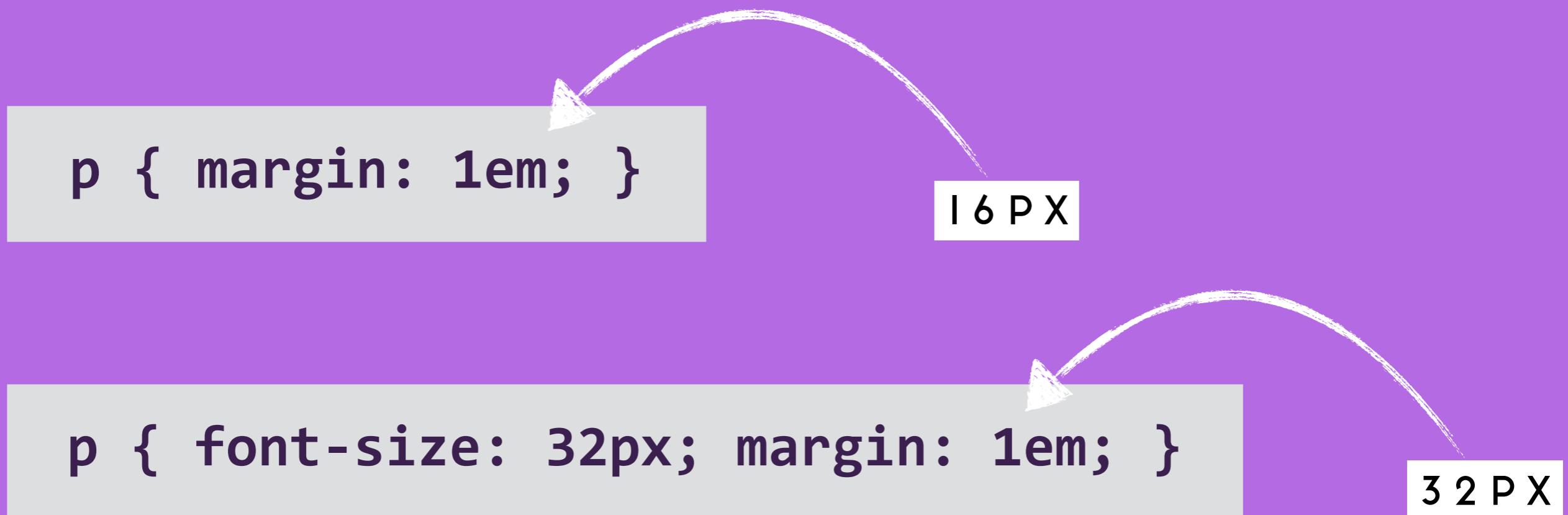
Relativa respecto de la altura de la letra **x**, del tipo y tamaño de letra del elemento.

px

Relativa respecto de la resolución de la pantalla del dispositivo en el que se visualiza la página HTML.

UNIDADES RELATIVAS - EM

La unidad **em** hace referencia al tamaño en puntos de la letra que se está utilizando. Si se utiliza una tipografía de 12 puntos, **1em** equivale a **12 puntos**.



UNIDADES RELATIVAS - %

El porcentaje **%** también es una unidad de medida relativa, y siempre hace referencia a otro valor para calcular el tamaño final.

```
body { font-size: 16px; }  
h1 { font-size: 200%; }  
h2 { font-size: 150%; }
```

32PX == 2EM

24PX == 1.5EM

UNIDADES RELATIVAS - %

Los porcentajes son ampliamente utilizados para calcular el ancho de los elementos. En estos casos, esta unidad relativa establece la medida en función de su contenedor.

```
<div class="content">  
  <div class="main">  
    ...  
  </div>  
</div>
```

```
.content {width: 960px; }  
.main {width : 50%; }
```

50% == 480px

RECOMENDACIONES

En general, se recomienda el uso de **unidades relativas** siempre que sea posible, ya que **mejora la accesibilidad** y permite que los documentos se **adapten** fácilmente a cualquier medio y dispositivo.

Normalmente se utilizan **píxel** y **porcentajes** para definir el **layout** del documento (básicamente, la anchura de las columnas y de los elementos de las páginas) y **em** y **porcentajes** para el **tamaño de letra** de los textos.

COLORES

Los colores en CSS se pueden indicar de cinco formas diferentes: palabras clave, colores del sistema, **RGB hexadecimal**, **RGB numérico** y RGB porcentual.

El método **más utilizado** para representar los colores es **RGB hexadecimal**.

R G B N U M É R I C O

En el modelo RGB un color se define indicando sus tres componentes **R** (rojo), **G** (verde) y **B** (azul).

Cada una de las componentes puede tomar un valor entre 0 y un valor máximo de 255. De esta forma, el color rojo puro en RGB se crea mediante el máximo valor de la componente R y un valor de 0 para las componentes G y B.

```
p { color: rgb(255, 0, 0); }
```

R G B H E X A D E C I M A L

Aunque es el método más complicado para indicar los colores, se trata del método más utilizado con mucha diferencia.

Definir un color en CSS con el método RGB hexadecimal requiere realizar los siguientes pasos:

- Determinar las componentes RGB decimales del color original, por ejemplo: R = 71, G = 98, B = 176
- **Transformar** el valor decimal de cada componente al sistema numérico **hexadecimal**.
- **Concatenar** los valores hexadecimales de las componentes RGB en ese orden y se les añade el **prefijo #**.

R G B H E X A D E C I M A L

```
p { color: rgb(255, 0, 0); }
```

```
p { color: #FF0000; }  
p { color: #F00; }
```

I.
INTRO-
DUCCIÓN

2.
SELEC-
TORES

3.
UNIDADES
DE
MEDIDA Y
COLORES

4.
MODELO
DE
CAJAS

5.
POSICIO-
NAMIENTO

6.
TEXTO

7.
ENLACES

8.
LISTAS

9.
FORMU-
LARIOS

10.
TABLAS

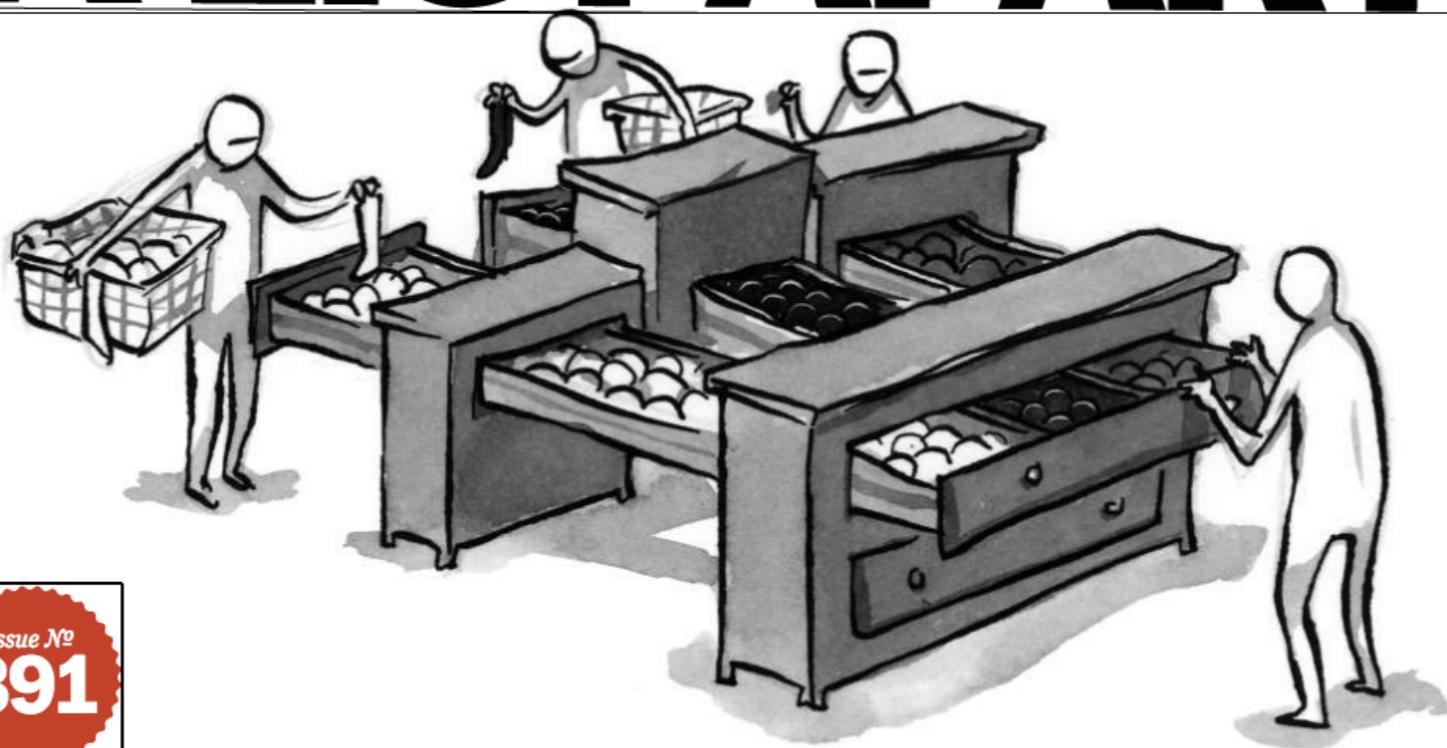
II.
LAYOUT
BÁSICO

12.
OTROS

El modelo de cajas o "**box model**" es seguramente la característica más importante del lenguaje de hojas de estilos CSS, ya que condiciona el diseño de todas las páginas web.

El modelo de cajas es el comportamiento de CSS que hace que **todos los elementos** de las páginas se representen mediante **cajas** rectangulares.

Cada vez que se inserta una etiqueta HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento.



Issue №
391

The Battle for the Body Field

by [JEFF EATON](#) · February 25, 2014

Published in [Content Strategy](#), [Information Architecture](#) · [11 Comments](#)

In the early '90s, every page was a handcrafted labor of love. Sadly, anyone who managed a large site eventually hit the wall: writing piles of custom HTML that tangled valuable content with boilerplate markup, gnarly design tweaks, and other difficult-to-maintain cruft.

Illustration:

[Kevin Cornell](#)

Share This:

[Twitter](#)

[Facebook](#)

[Google+](#)

Soon, large sites abandoned handcrafted pages entirely. The meat of a page got stored in a database, then passed through HTML templates to “wrap” it in design elements like footers, sidebars, and banner ads. Today, even individual elements like the name of a book, a photo of its cover, and an author’s bio are often teased out of design-heavy HTML and stored as individual chunks. Content editors fill out input forms rather than wrestling with a blank HTML canvas, and CMS templates reshuffle the elements as needed.

Trouble in Chunkytown

This fields-and-templates approach works great for content that follows predictable

MODELO DE CAJAS

Los navegadores crean y colocan las cajas de forma automática, pero **CSS permite modificar** todas sus **características**. Cada una de las cajas está formada por **seis partes**.



CONTENT

PADDING

BORDER

left

top

right

bottom

BACKGROUND-IMAGE
&
BACKGROUND-COLOR

MARGIN

ANCHURA

La propiedad CSS que controla la anchura de la caja de los elementos se denomina **width**

TODOS LOS ELEMENTOS

EXCEPTO LOS ELEMENTOS EN LÍNEA QUE NO SEAN IMÁGENES Y LAS FILAS DE TABLAS.

```
<div class="content">  
...  
</div>
```

```
.content {width: 960px; }  
.content {width: 50%; }
```

ALTURA

La propiedad CSS que controla la altura de los elementos se denomina **height**

TODOS LOS ELEMENTOS

EXCEPTO LOS ELEMENTOS EN LÍNEA QUE NO SEAN IMÁGENES Y LAS FILAS DE TABLAS.

```
<div class="content">  
...  
</div>
```

```
.content {height: 960px;}  
.content {height: 50%;}
```

MARGEN

CSS define cuatro propiedades para controlar cada uno de los márgenes horizontales y verticales de un elemento.

margin-top

margin-right

margin-bottom

margin-left

**TODOS LOS
ELEMENTOS**

**SALVO margin-top Y
margin-bottom QUE
SOLO SE APLICAN
A ELEMENTOS EN
BLOQUE E IMÁGENES**

```
<p class="destacado">  
...  
</p>
```

```
.destacado {  
    margin-top: 1em;  
    margin-left: 1em;  
}
```

MARGEN

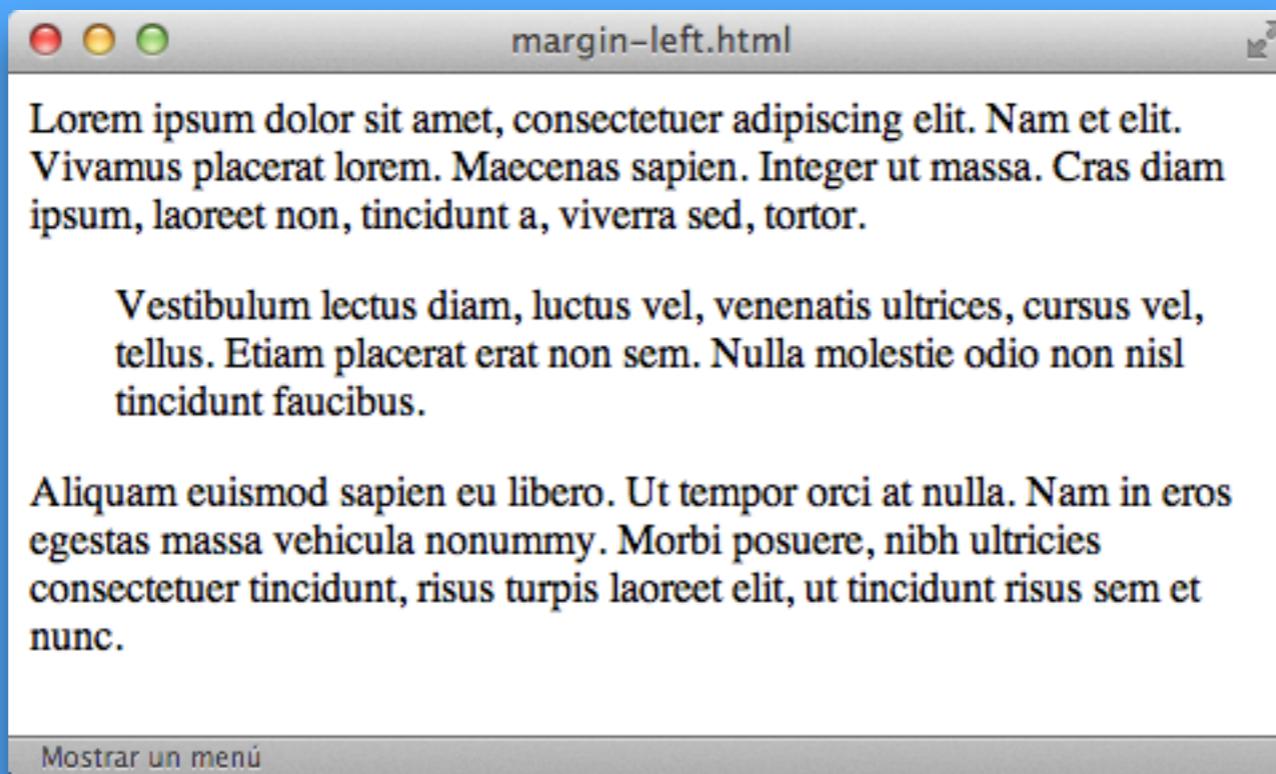
```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam et elit.  
Vivamus placerat lorem. Maecenas sapien. Integer ut massa. Cras diam ipsum,  
laoreet non, tincidunt a, viverra sed, tortor.</p>
```

```
<p class="destacado">Vestibulum lectus diam, luctus vel, venenatis  
ultrices, cursus vel, tellus. Etiam placerat erat non sem. Nulla  
molestie odio non nisl tincidunt faucibus.</p>
```

```
<p>Aliquam euismod sapien eu libero. Ut tempor orci at nulla. Nam in eros  
egestas massa vehicula nonummy. Morbi posuere, nibh ultricies consectetur  
tincidunt, risus turpis laoreet elit, ut tincidunt risus sem et nunc.</p>
```

```
.destacado { margin-left: 1em; }
```

MARGEN



Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Nam et elit.
 Vivamus placerat lorem. Maecenas sapien. Integer ut massa. Cras diam
 ipsum, laoreet non, tincidunt a, viverra sed, tortor.

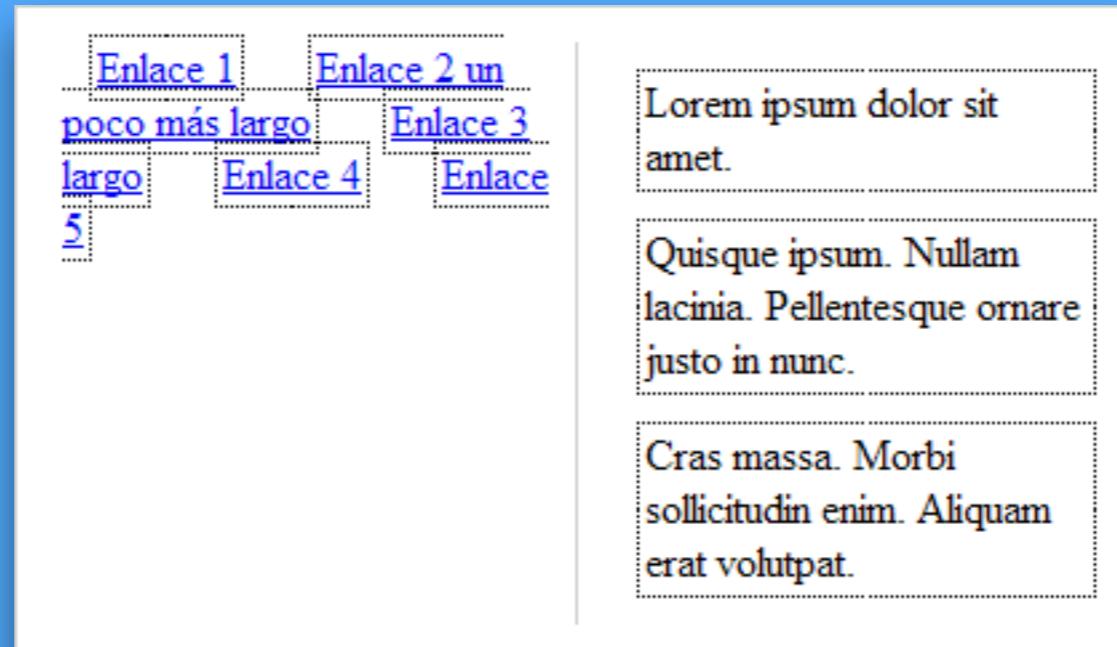
Vestibulum lectus diam, luctus vel, venenatis ultrices, cursus vel,
 tellus. Etiam placerat erat non sem. Nulla molestie odio non nisl
 tincidunt faucibus.

Aliquam euismod sapien eu libero. Ut tempor orci at nulla. Nam in eros
 egestas massa vehicula nonummy. Morbi posuere, nibh ultricies
 consectetur tincidunt, risus turpis laoreet elit, ut tincidunt risus sem et
 nunc.

Mostrar un menú

MARGEN

Los márgenes verticales (`margin-top` y `margin-bottom`) sólo se pueden aplicar a los elementos de bloque y las imágenes, mientras que los márgenes laterales (`margin-left` y `margin-right`) se pueden aplicar a cualquier elemento.



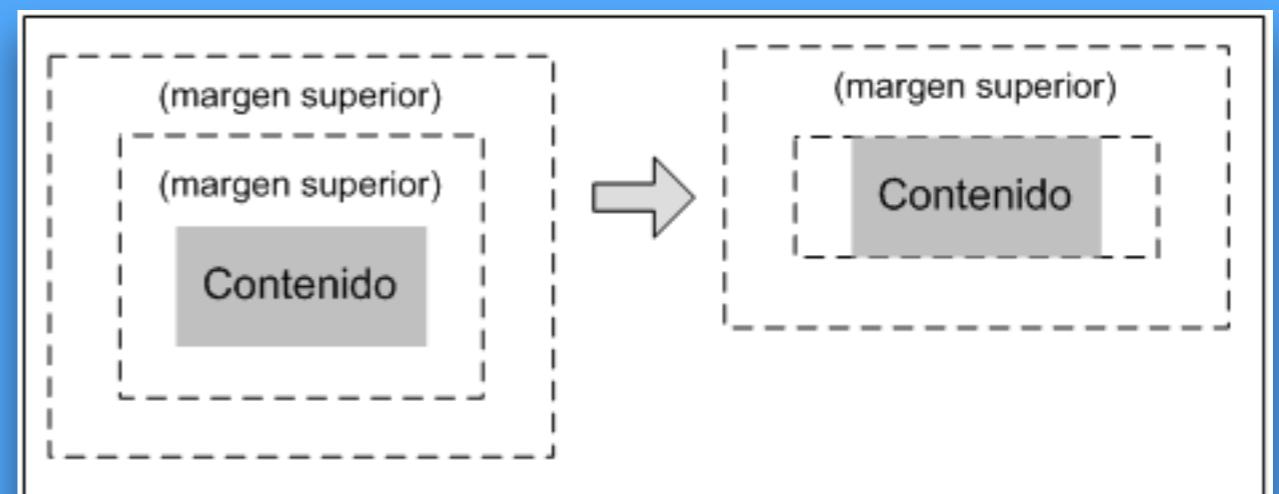
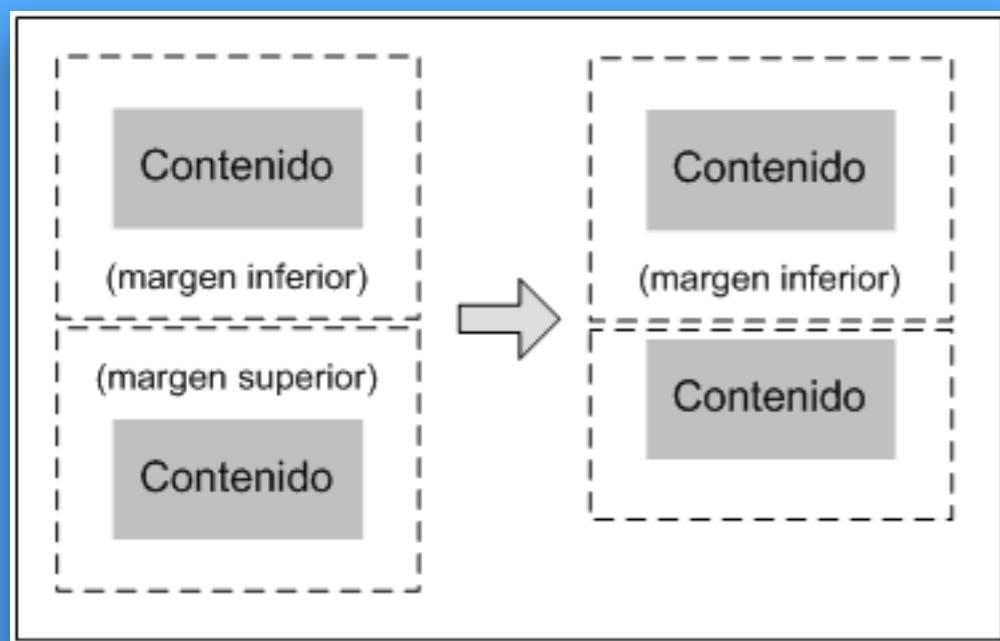
MARGEN

Además de las cuatro propiedades que controlan cada uno de los márgenes del elemento, CSS define una propiedad especial que permite **establecer** los cuatro márgenes de **forma simultánea**.

```
.destacado { margin: 1em 0.5em 10px 5%; }
```

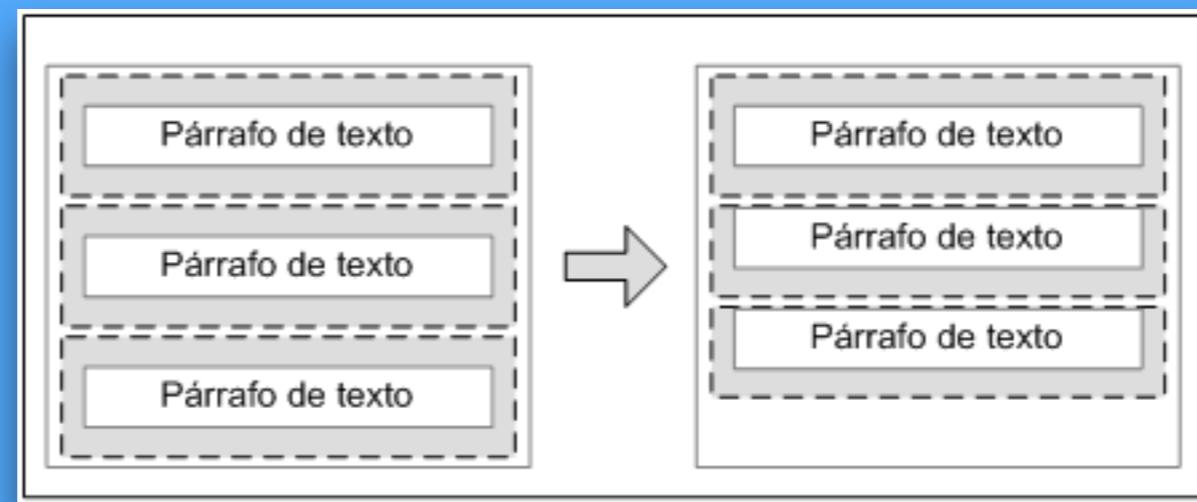
MARGEN

El comportamiento de los márgenes verticales es especial. 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.



MARGEN

La razón por la que se propuso este mecanismo de fusión automática de márgenes verticales es el de dar uniformidad a las páginas web.



MARGEN

.header

.feature

.body

```
.feature {  
    margin: 20px 0 40px 0;  
}
```

MARGEN

.header

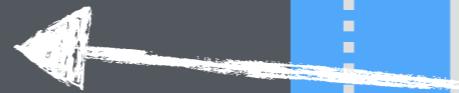
.body



```
.feature {  
    margin: 20px 0 40px 0;  
}
```

MARGEN

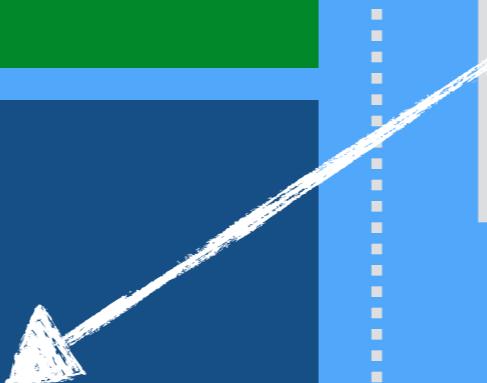
.header



.feature



.body



```
.header {  
    margin: 20px 0 40px 0;  
}  
.feature {  
    margin: 20px 0;  
}  
.body {  
    margin: 10px 0;  
}
```

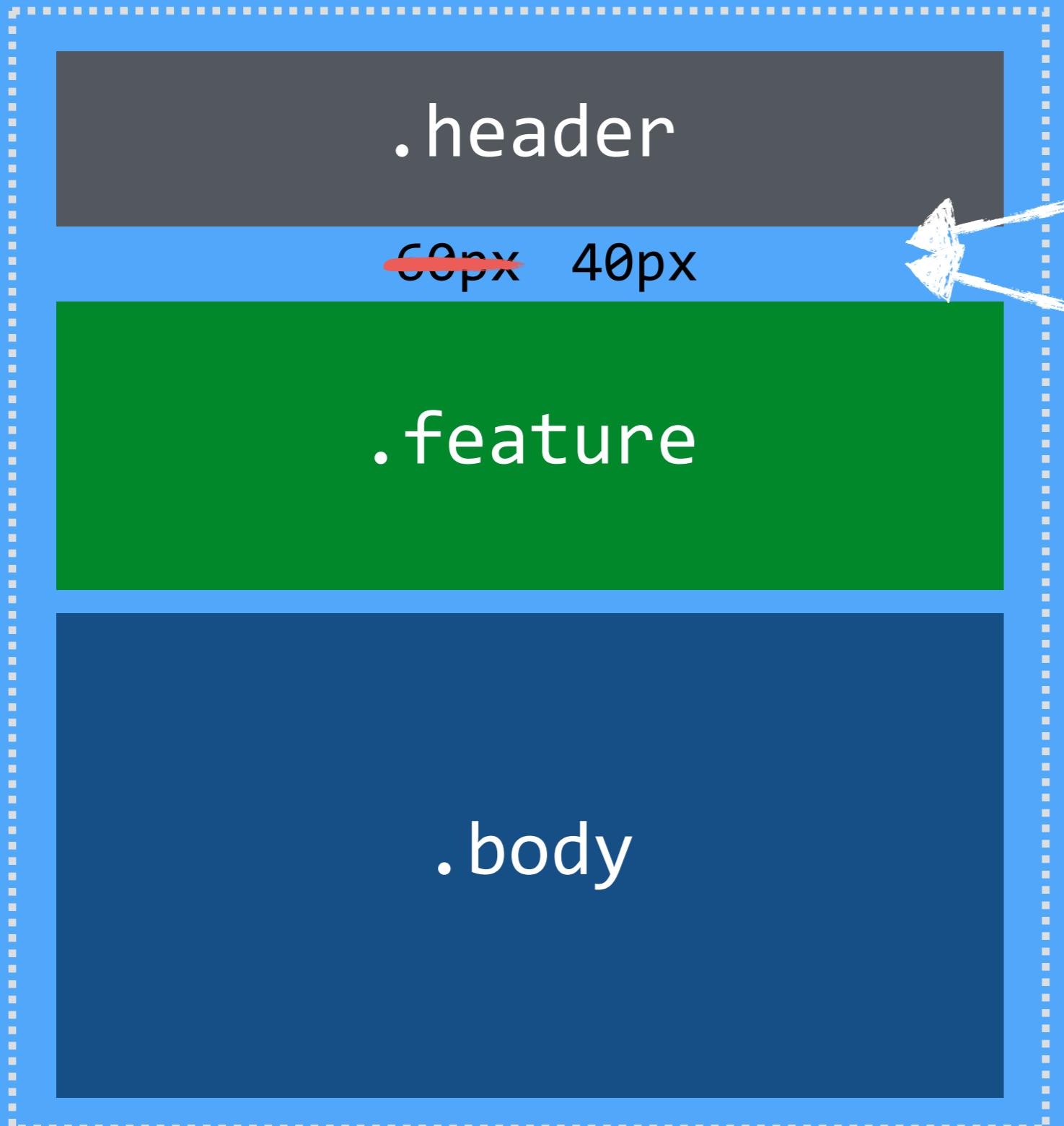
MARGEN

.header

.body

```
.header {  
    margin: 20px 0 40px 0;  
}  
.feature {  
    margin: 20px 0;  
}  
.body {  
    margin: 10px 0;  
}
```

MARGEN



```
.header {  
    margin: 20px 0 40px 0;  
}  
.feature {  
    margin: 20px 0;  
}  
.body {  
    margin: 10px 0;  
}
```

MARGEN

.header

.feature

40px 20px

.body

```
.header {  
    margin: 20px 0 40px 0;  
}  
.feature {  
    margin: 20px 0;  
}  
.body {  
    margin: 10px 0;  
}
```

MARGEN

Centrando elementos de bloque:



MARGEN

Centrando elementos de bloque:



RELENO

CSS define cuatro propiedades para controlar cada uno de los espacios de relleno horizontales y verticales de un elemento.

padding-top

padding-right

padding-bottom

padding-left

**TODOS LOS
ELEMENTOS**

```
<p class="destacado">  
...  
</p>
```

```
.destacado {  
    padding-top: 1em;  
    padding-bottom: 1em;  
}
```

R E L L E N O

Como sucede con los márgenes, CSS también define una propiedad llamada **padding** para establecer los cuatro rellenos de un elemento de forma simultánea.

```
.destacado { padding: 1em 0.5em 10px 5%; }
```

B O R D E S

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.

`border-top-width`

`border-right-width`

`border-bottom-width`

`border-left-width`

`border-top-color`

`border-right-color`

`border-bottom-color`

`border-left-color`

`border-top-style`

`border-right-style`

`border-bottom-style`

`border-left-style`

ANCHURA DE BORDES

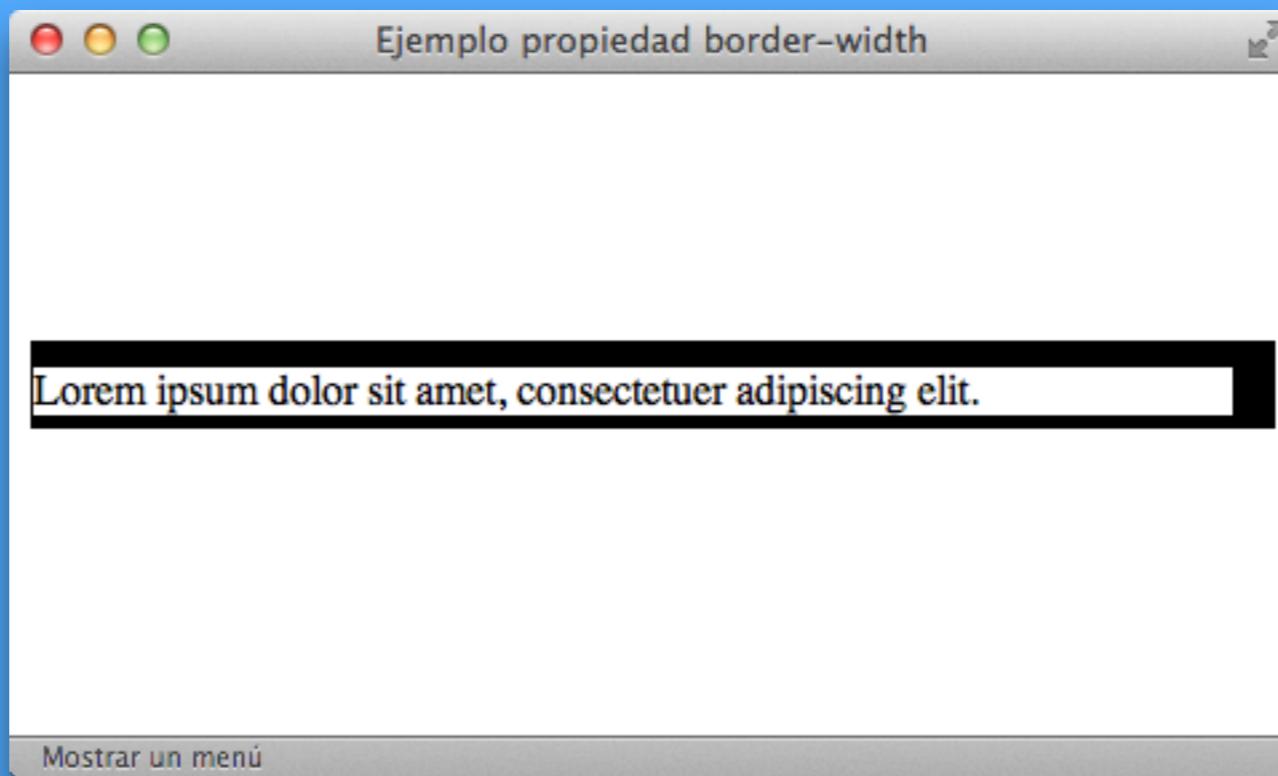
La anchura de los bordes se indica mediante una medida (en cualquier unidad de medida absoluta o relativa) o mediante las palabras clave thin (borde delgado), medium (borde normal) y thick (borde ancho).

**TODOS LOS
ELEMENTOS**

```
<div> . . . </div>
```

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

ANCHURA DE BORDES



COLOR DE BORDES

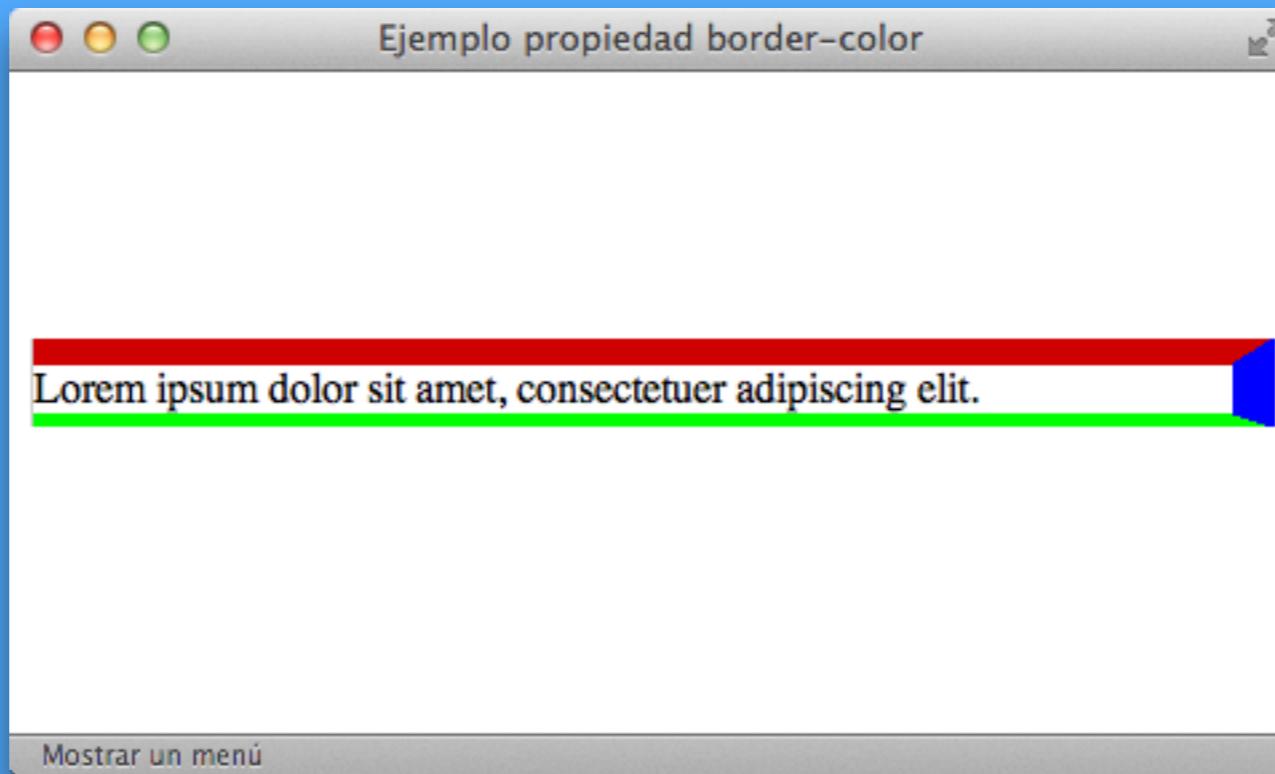
Con estas propiedades podemos establecer el color de cada uno de los cuatro bordes de los elementos.

TODOS LOS ELEMENTOS

```
<div> . . . </div>
```

```
div {  
    border-top-color: red;  
    border-right-color: blue;  
    border-bottom-color: #0F0;  
    border-left-color: blue;  
}
```

COLOR DE BORDES



ESTILO DE BORDES

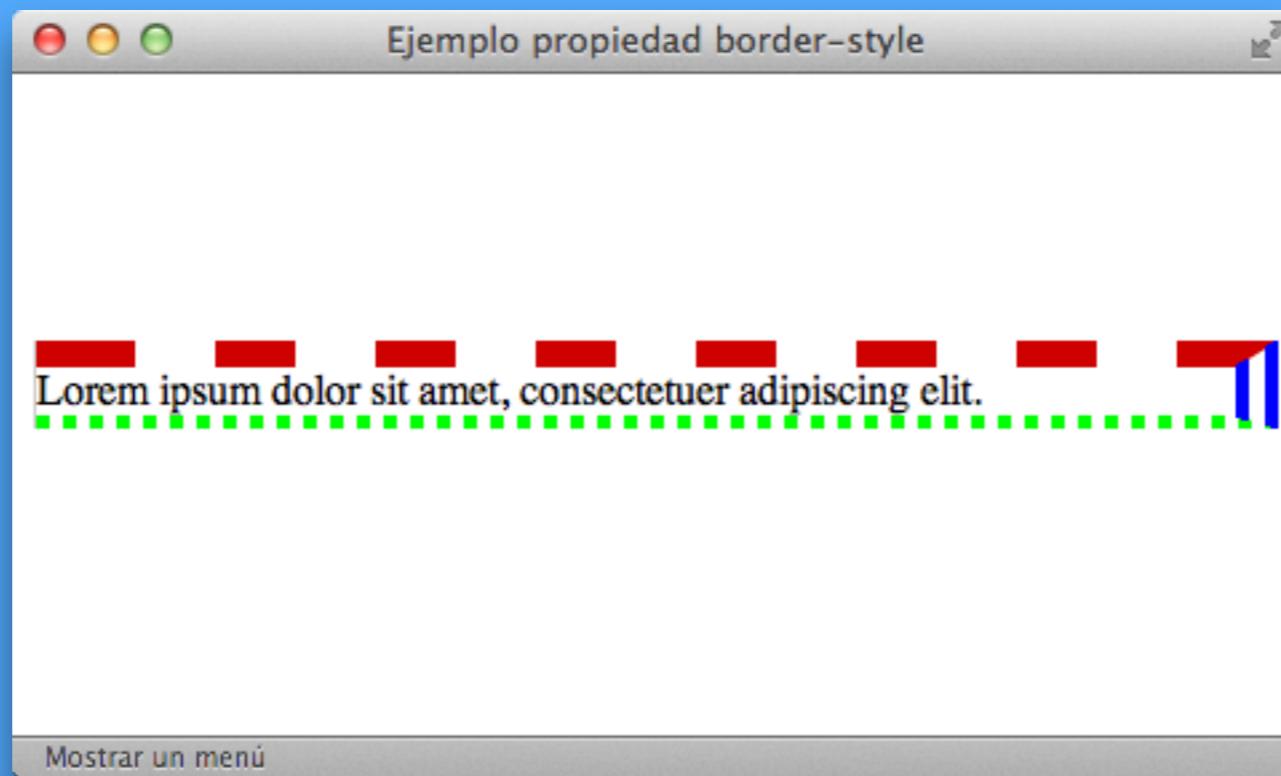
Con estas propiedades podemos establecer el estilo de cada uno de los bordes mediante las siguientes propiedades:

TODOS LOS ELEMENTOS

```
<div> . . .</div>
```

```
div {  
    border-top-style: dashed;  
    border-right-style: double;  
    border-bottom-style: dotted;  
    border-left-style: solid;  
}
```

ESTILO DE BORDES



B O R D E S

Como sucede con los márgenes y los rellenos, CSS define una serie de propiedades de tipo "**shorthand**" que permiten establecer todos los atributos de los bordes de forma simultánea.

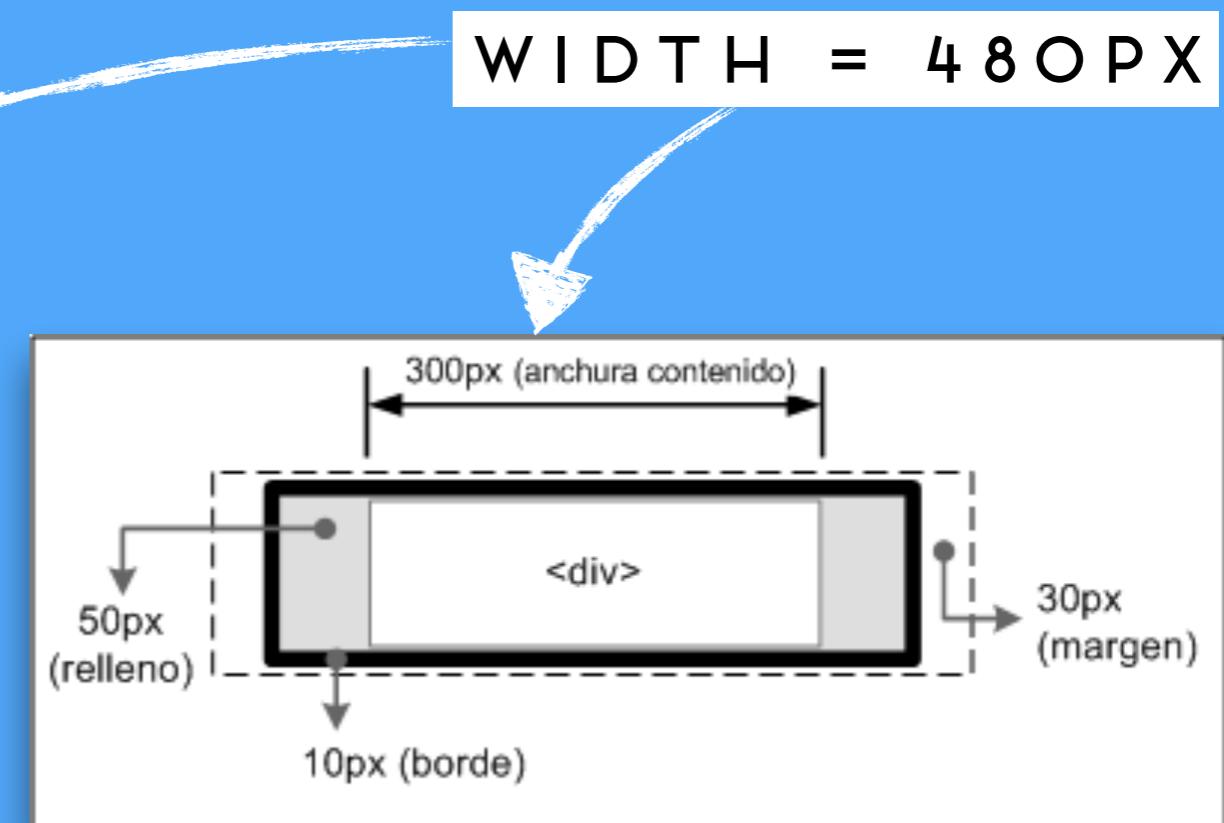
```
h1 { border-bottom: 3px double #369; }
```

```
h1 { border: 1px solid red; }
```

TAMAÑO DE ELEMENTOS

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;  
}
```



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.

CSS define cinco propiedades para establecer el fondo de cada elemento:

background-color

background-image

background-repeat

background-attachment

background-position

FONDOS

```
/* Color e imagen de fondo de la página mediante una propiedad shorthand */
body { background: #222d2d url("./graphics/colorstrip.gif") repeat-x 0 0; }

/* La propiedad shorthand anterior es equivalente a las siguientes propiedades */
body {
    background-color: #222d2d;
    background-image: url("./graphics/colorstrip.gif");
    background-repeat: repeat-x;
    background-position: 0 0;
}
```

I.
INTRO-
DUCCIÓN

2.
SELEC-
TORES

3.
UNIDADES
DE
MEDIDA Y
COLORES

4.
MODELO
DE
CAJAS

5.
POSICIO-
NAMIENTO

6.
TEXTO

7.
ENLACES

8.
LISTAS

9.
FORMU-
LARIOS

10.
TABLAS

II.
LAYOUT
BÁSICO

12.
OTROS

Cuando los navegadores descargan el contenido HTML y CSS de las páginas web, aplican un **procesamiento** muy **complejo** antes de mostrar las páginas en la pantalla del usuario.

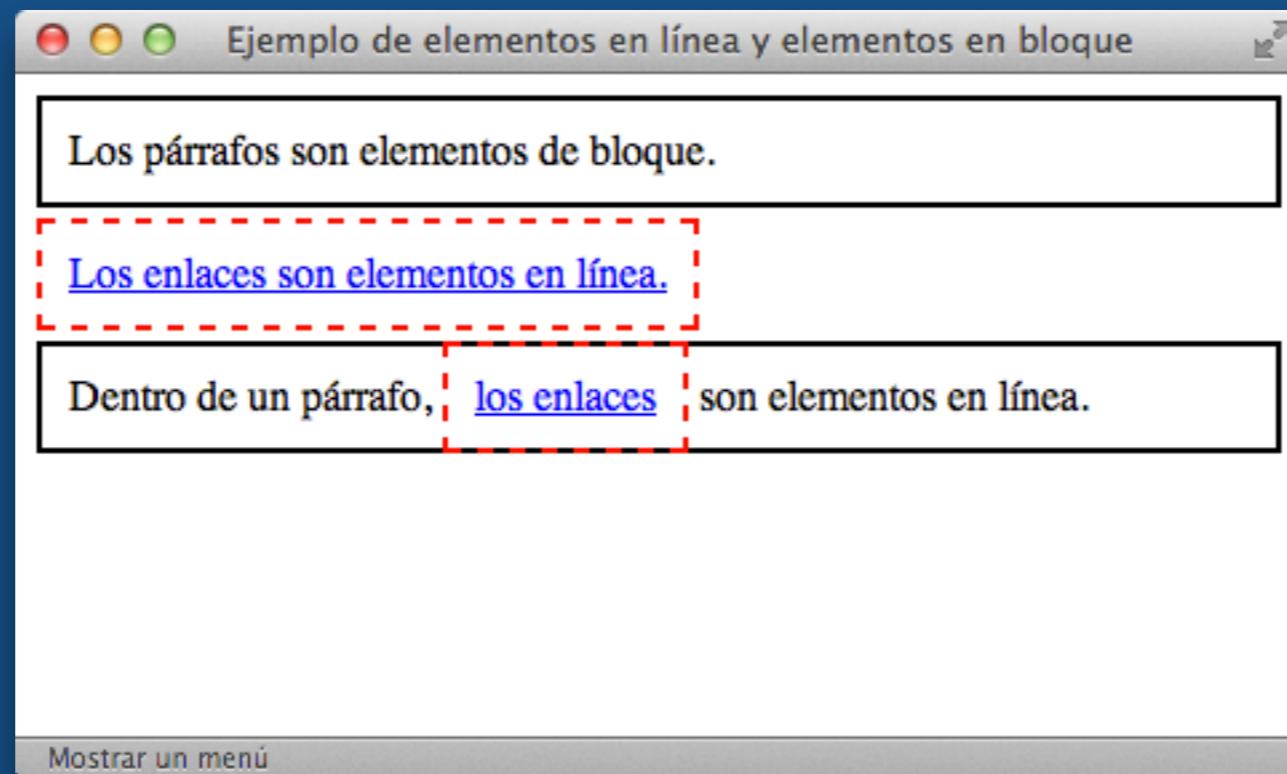
Los navegadores crean una caja para representar a cada elemento de la página HTML.

Los **factores** que se tienen en cuenta para **generar cada caja** son:

- Las propiedades width y height de la caja.
- El tipo de cada elemento HTML (bloque o en línea).
- Posicionamiento de la caja (normal, relativo, absoluto, fijo o flotante).
- Las relaciones entre elementos (dónde se encuentra cada elemento, elementos descendientes, etc.)
- Otro tipo de información, como por ejemplo el tamaño de las imágenes y el tamaño de la ventana del navegador.

TIPOS DE ELEMENTOS

El estándar HTML clasifica a todos sus elementos en dos grandes grupos: **elementos en línea** y **elementos de bloque**.



POSICIONAMIENTO

Los navegadores crean y **posicionan de forma automática** todas las cajas que forman cada página HTML. No obstante, **CSS permite modificar** la posición en la que se muestra cada caja.

Utilizando las propiedades que proporciona CSS para alterar la posición de las cajas es posible realizar efectos muy avanzados y **diseñar estructuras** de páginas que de otra forma no serían posibles.

POSICIONAMIENTO

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

Normal

Relativo

Absoluto

Fijo

Flotante

POSICIONAMIENTO

El posicionamiento de una caja se establece mediante la

propiedad **position**

**TODOS LOS
ELEMENTOS**

```
<ul class="menu"> . . . </ul>
```

```
.menu {  
    position: fixed;  
    top: 10px;  
    left: 10px;  
}
```

POSICIONAMIENTO

El significado de cada uno de los posibles valores de la propiedad **position** es el siguiente:

static

Corresponde con el posicionamiento normal.

relative

Posicionamiento relativo. El desplazamiento se controla a través de **top**, **right**, **bottom** y **left**.

absolute

Posicionamiento absoluto. El desplazamiento también se controla a través de **top**, **right**, **bottom** y **left**.

fixed

Posicionamiento fijo. Mismo comportamiento que el absoluto, pero el elemento permanece fijo en pantalla.

POSICIONAMIENTO

La propiedad position sólo indica **cómo se posiciona** una caja, pero no la desplaza. Para ello es necesario utilizar las propiedades **top**, **right**, **bottom**, y **left**.

**TODOS LOS
ELEMENTOS
POSICIONADOS**

```
<ul class="menu"> . . . </ul>
```

```
.menu {  
    position: fixed;  
    top: 10px;  
    left: 10px;  
}
```

POSICIONAMIENTO

En el caso del **posicionamiento relativo**, cada una de estas propiedades indica el **desplazamiento** del elemento **desde la posición** original de su borde superior/derecho/inferior/izquierdo.

Si el **posicionamiento** es **absoluto**, las propiedades indican el desplazamiento del elemento **respecto del borde** superior/derecho/inferior/izquierdo **de su primer elemento padre posicionado**.

En cualquiera de los dos casos, si el **desplazamiento** se indica en **forma de porcentaje**, se refiere al porcentaje **sobre la anchura** (propiedades right y left) **o altura** (propiedades top y bottom) **del elemento**.

POSICIONAMIENTO NORMAL

El posicionamiento **normal** o estático es el modelo que utilizan **por defecto** los navegadores para mostrar los elementos de las páginas. En este modelo, sólo se tiene en cuenta **si el elemento es de bloque o en línea**, sus propiedades width y height y su contenido.

POSICIONAMIENTO NORMAL

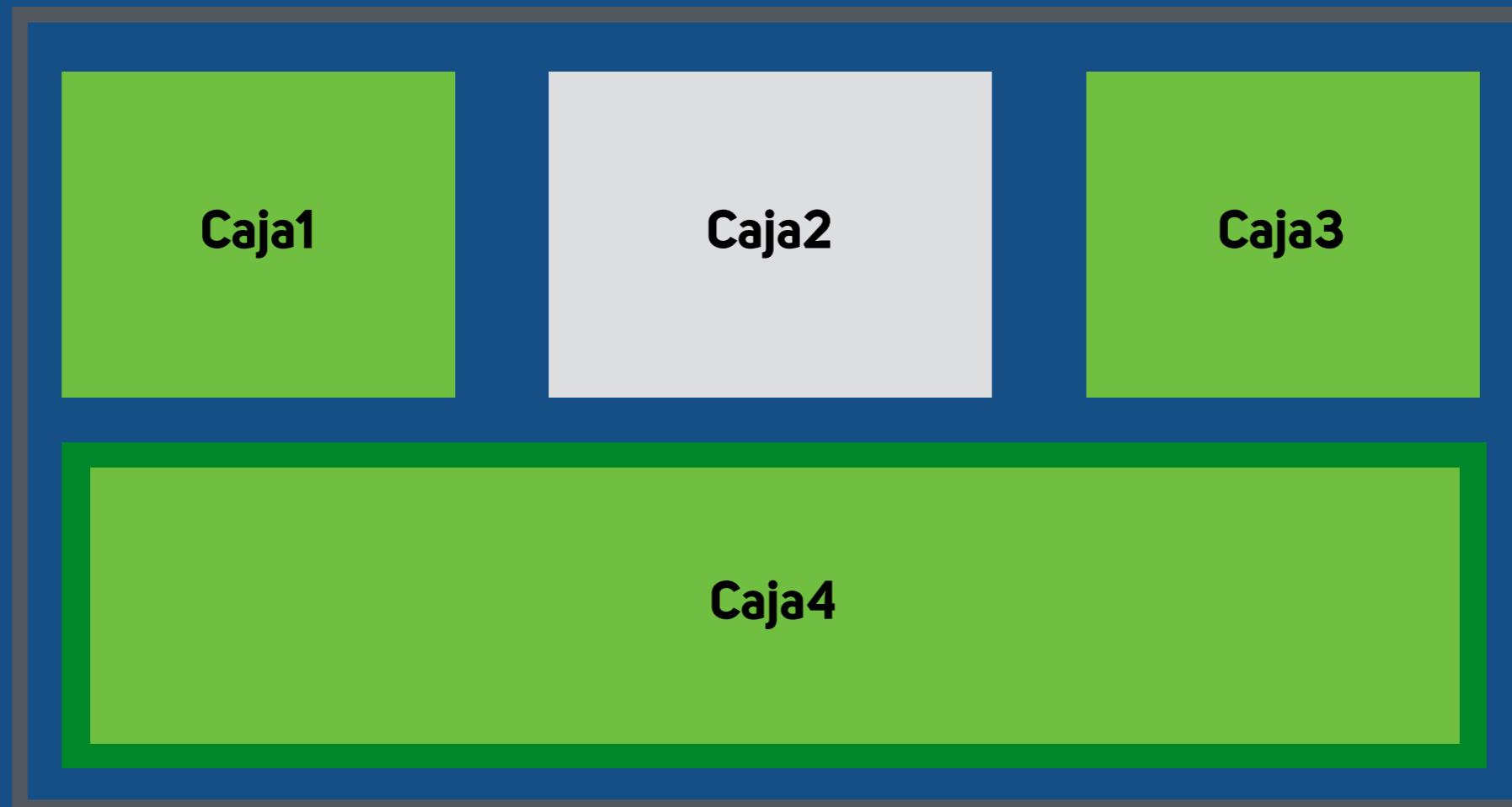


POSICIONAMIENTO RELATIVO

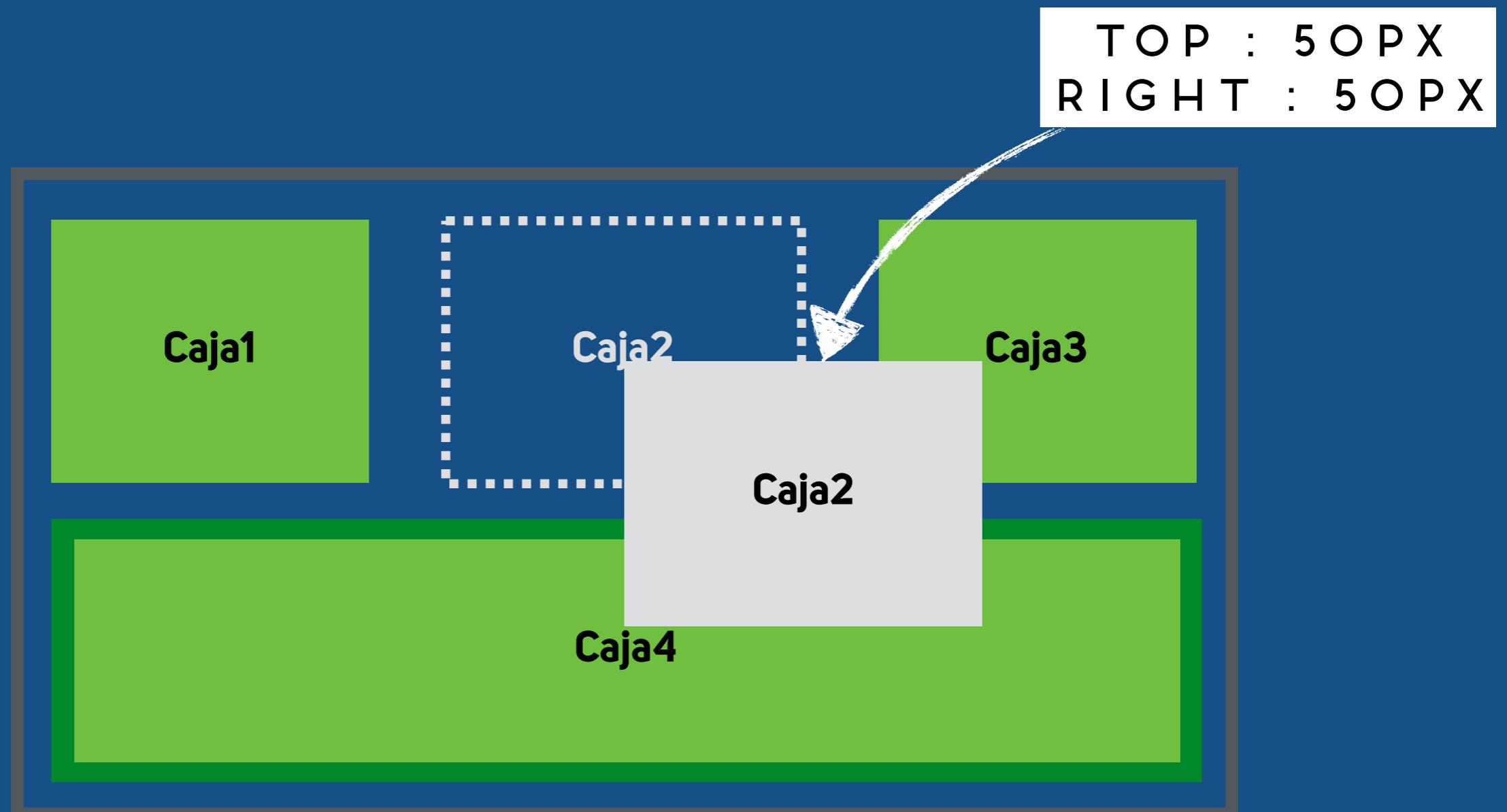
El posicionamiento **relativo** desplaza una caja respecto de su posición original establecida mediante el posicionamiento normal.

El valor de la propiedad **top** se interpreta como el **desplazamiento** entre el borde superior de la caja en su **posición final** y el borde superior de la misma caja en su **posición original**.

POSICIONAMIENTO RELATIVO



POSICIONAMIENTO RELATIVO

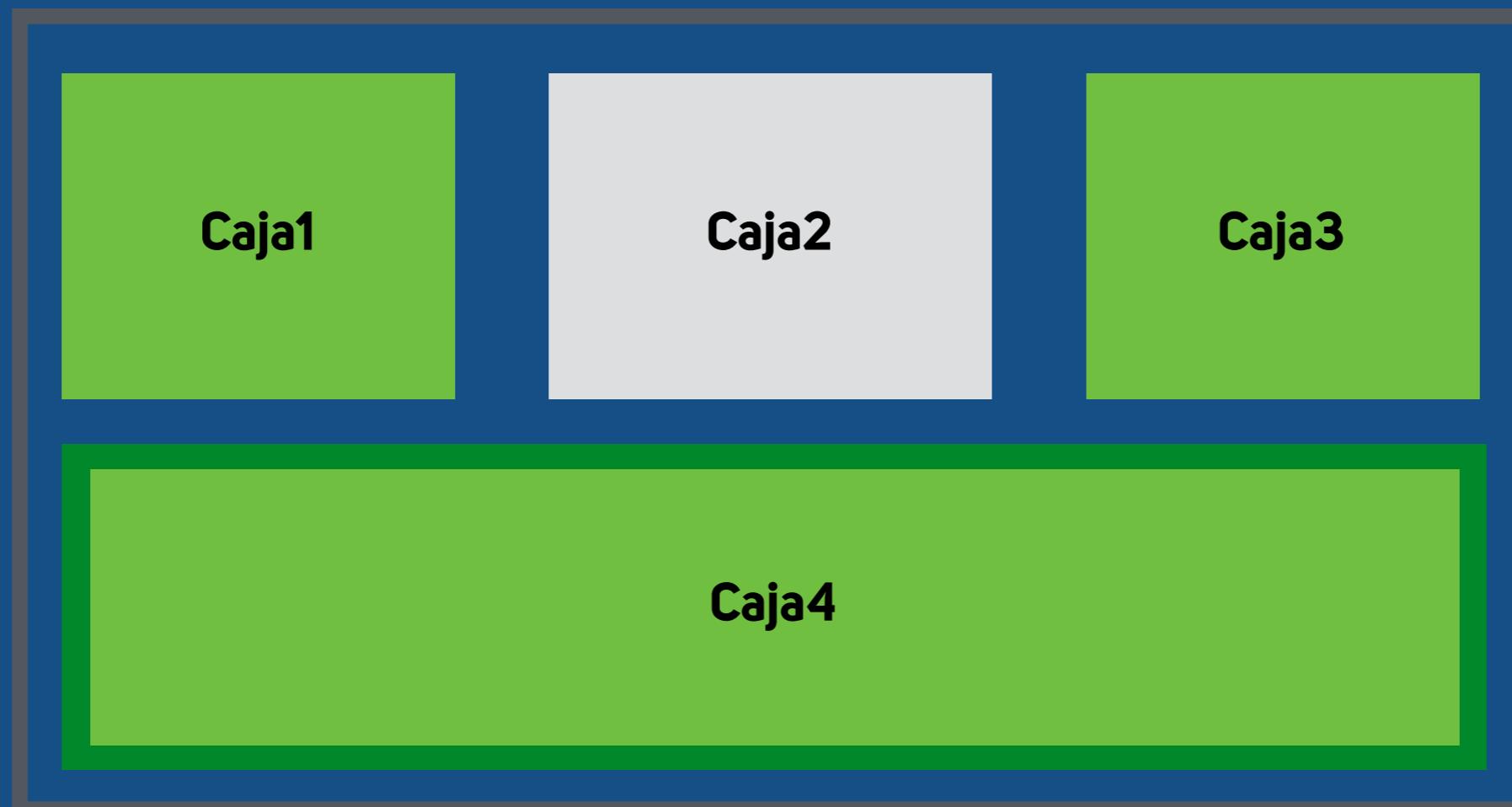


POSICIONAMIENTO ABSOLUTO

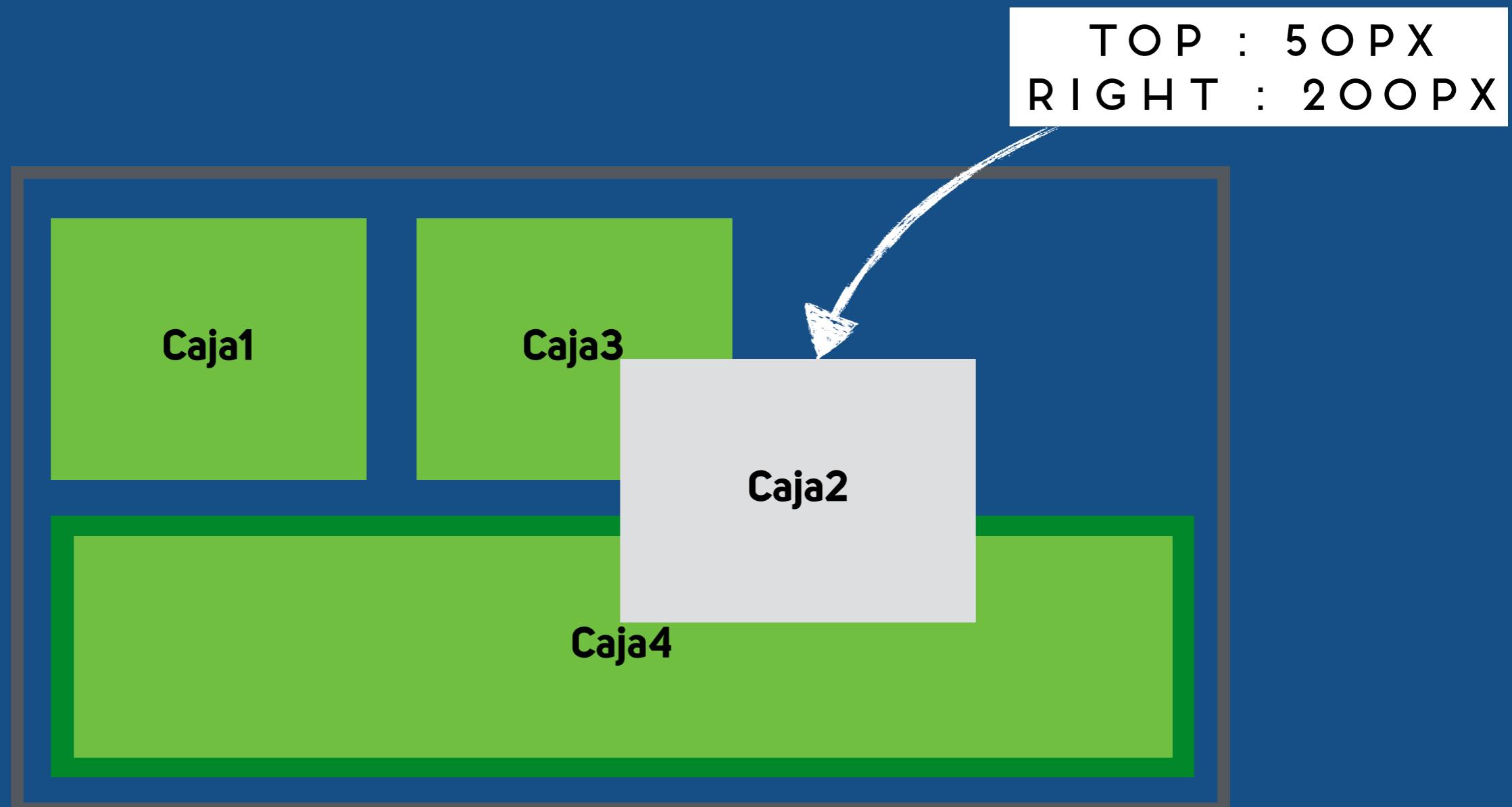
El posicionamiento **absoluto** se emplea para **establecer de forma exacta la posición** en la que se muestra la caja de un elemento.

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

POSICIONAMIENTO ABSOLUTO

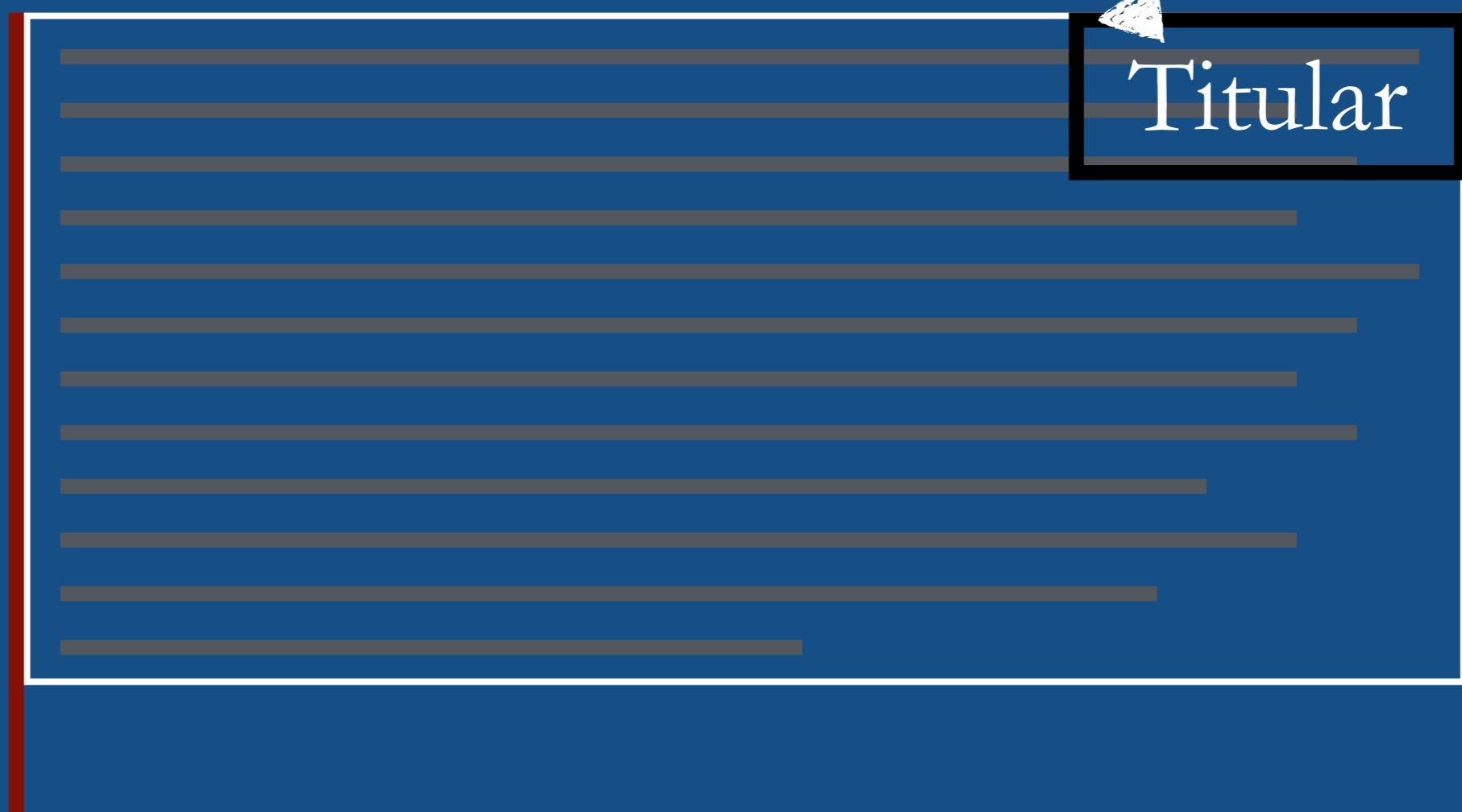


POSICIONAMIENTO ABSOLUTO



POSICIONAMIENTO ABSOLUTO

TOP : 0
RIGHT : 0



POSICIONAMIENTO ABSOLUTO

TOP : 0
RIGHT : 0



POSICIONAMIENTO ABSOLUTO + RELATIVO

POSITION: RELATIVE

TOP : 0
RIGHT : 0



POSICIONAMIENTO FIJO

Cuando una caja se posiciona de forma **fija**, la forma de obtener el origen de coordenadas para interpretar su desplazamiento es **idéntica al posicionamiento absoluto**.

La principal característica de una caja posicionada de forma fija es que su **posición es inamovible** dentro de la ventana del navegador.

POSICIONAMIENTO FIJO



POSICIONAMIENTO FLOTANTE

El posicionamiento flotante es el **más difícil** de comprender pero al mismo tiempo es el **más utilizado**.

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.

POSICIONAMIENTO FLOTANTE



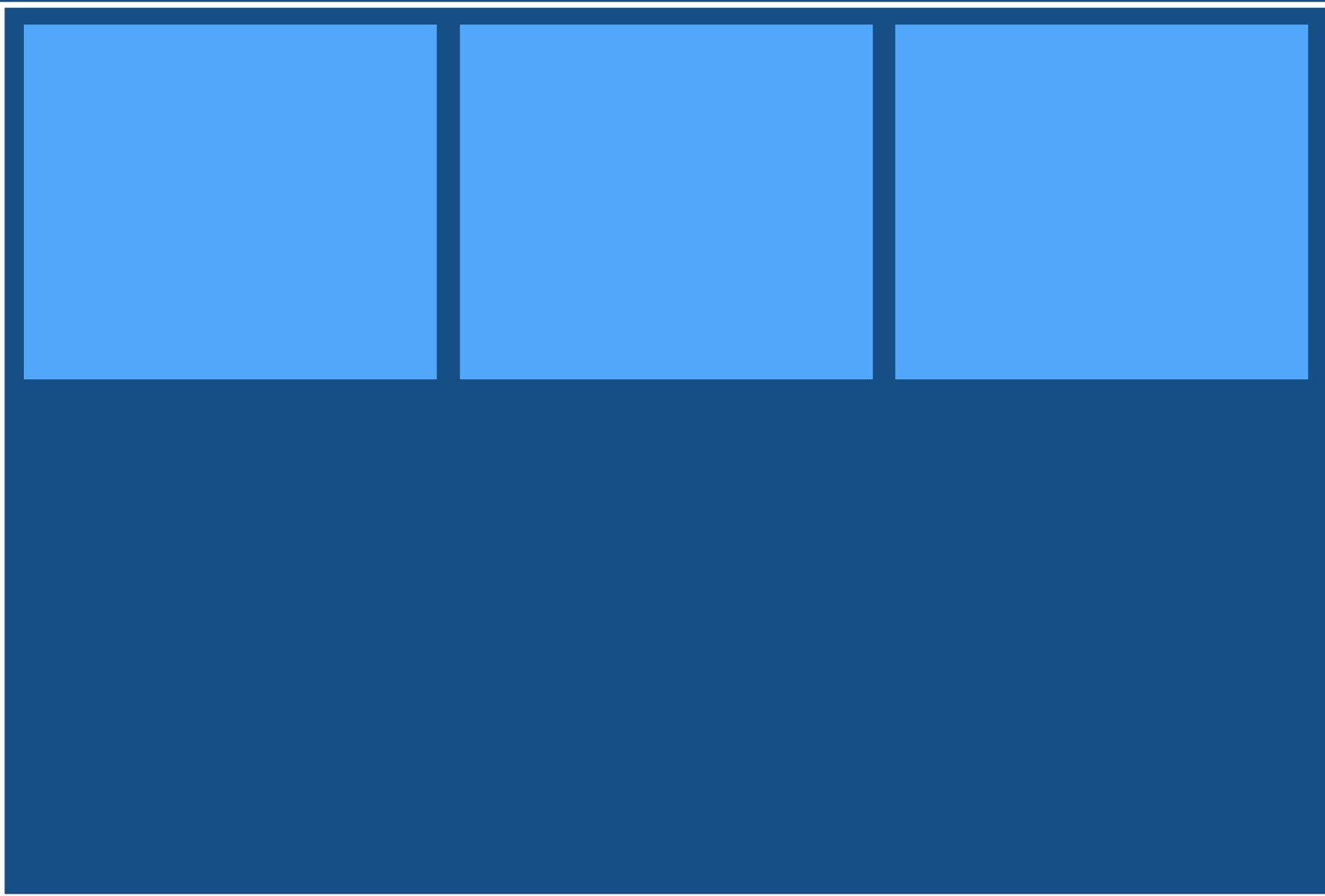
```
img {  
    float: left;  
}
```

POSICIONAMIENTO FLOTANTE

Cuando se posiciona una caja de forma flotante:

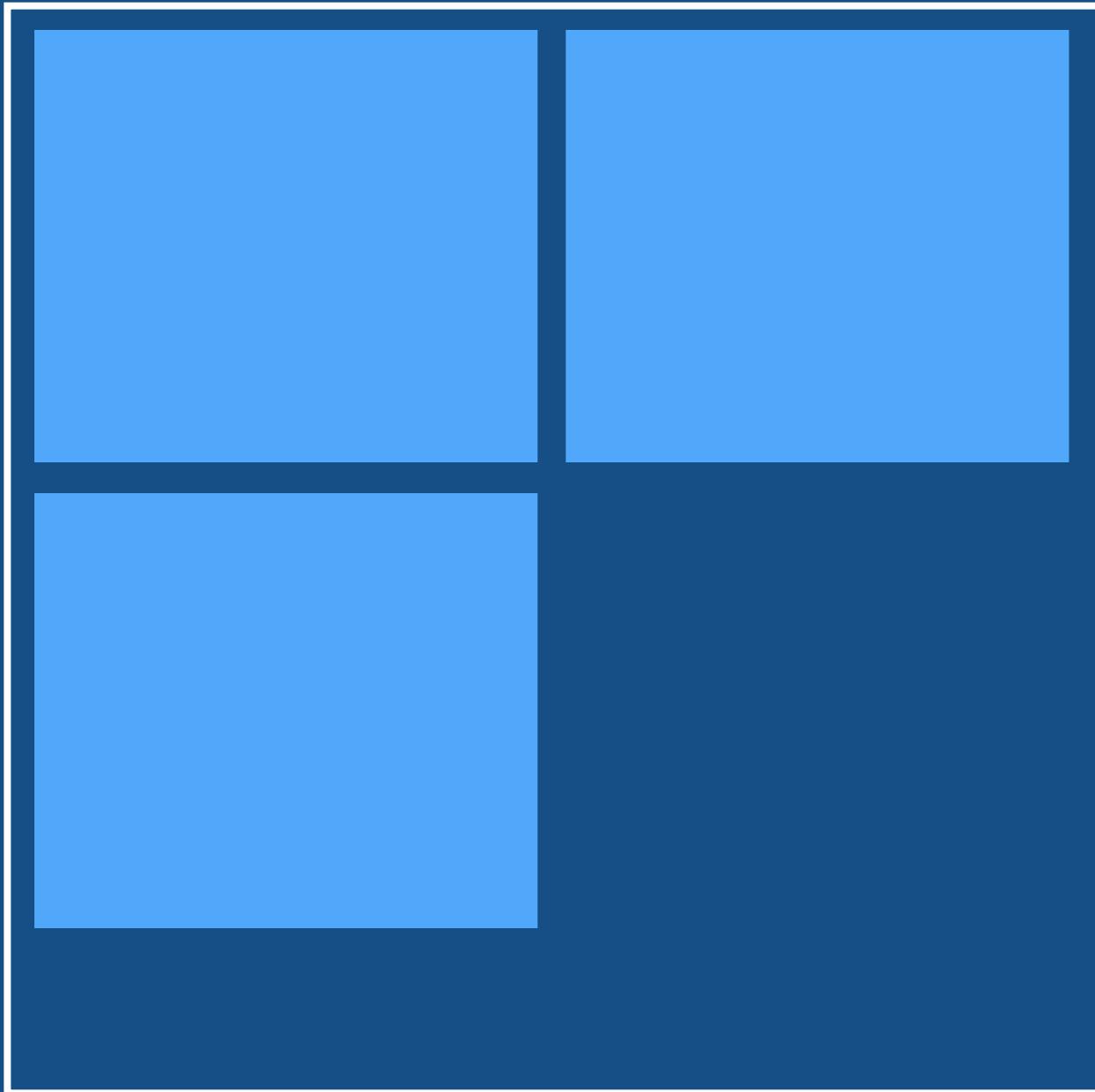
- La caja **deja de pertenecer al flujo normal de la página**, lo que significa que el resto de cajas ocupan el lugar dejado por la caja flotante.
- La caja flotante **se posiciona lo más a la izquierda o lo más a la derecha posible** de la posición en la que se encontraba originalmente.

POSICIONAMIENTO FLOTANTE



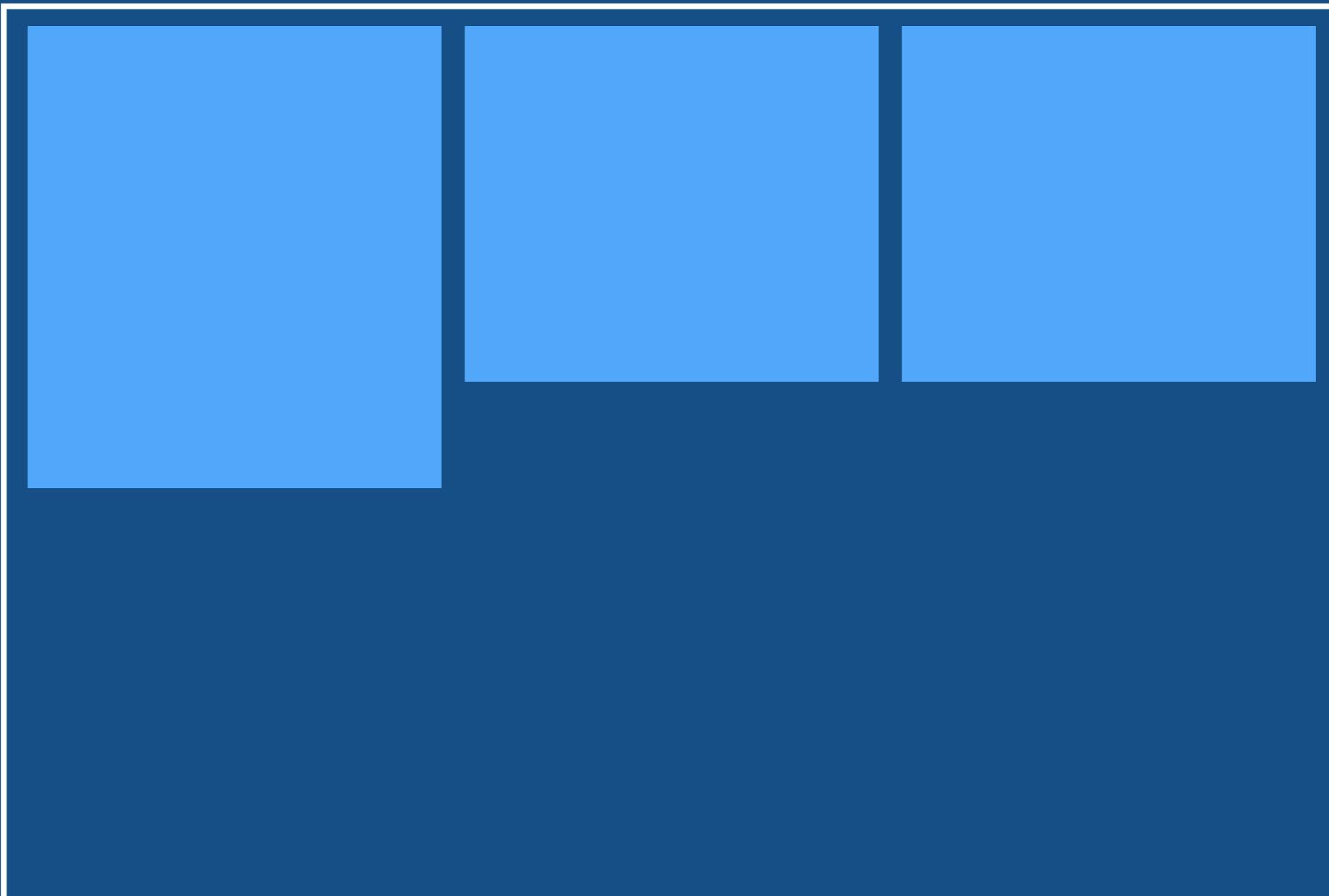
```
.container {  
    width: 100%;  
}  
  
div {  
    float: left;  
}
```

POSICIONAMIENTO FLOTANTE

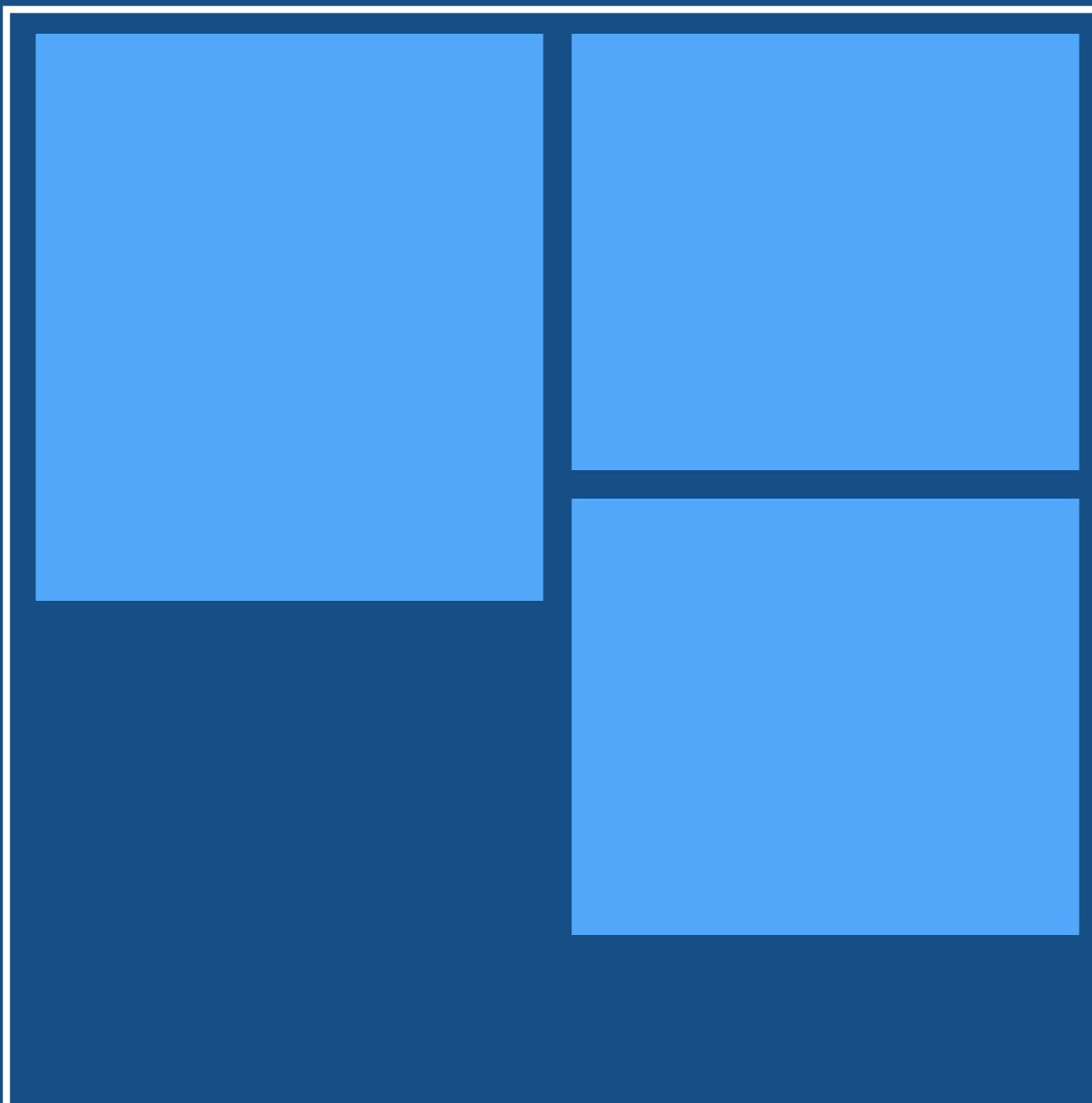


```
.container {  
    width: 50%;  
}  
  
div {  
    float: left;  
}
```

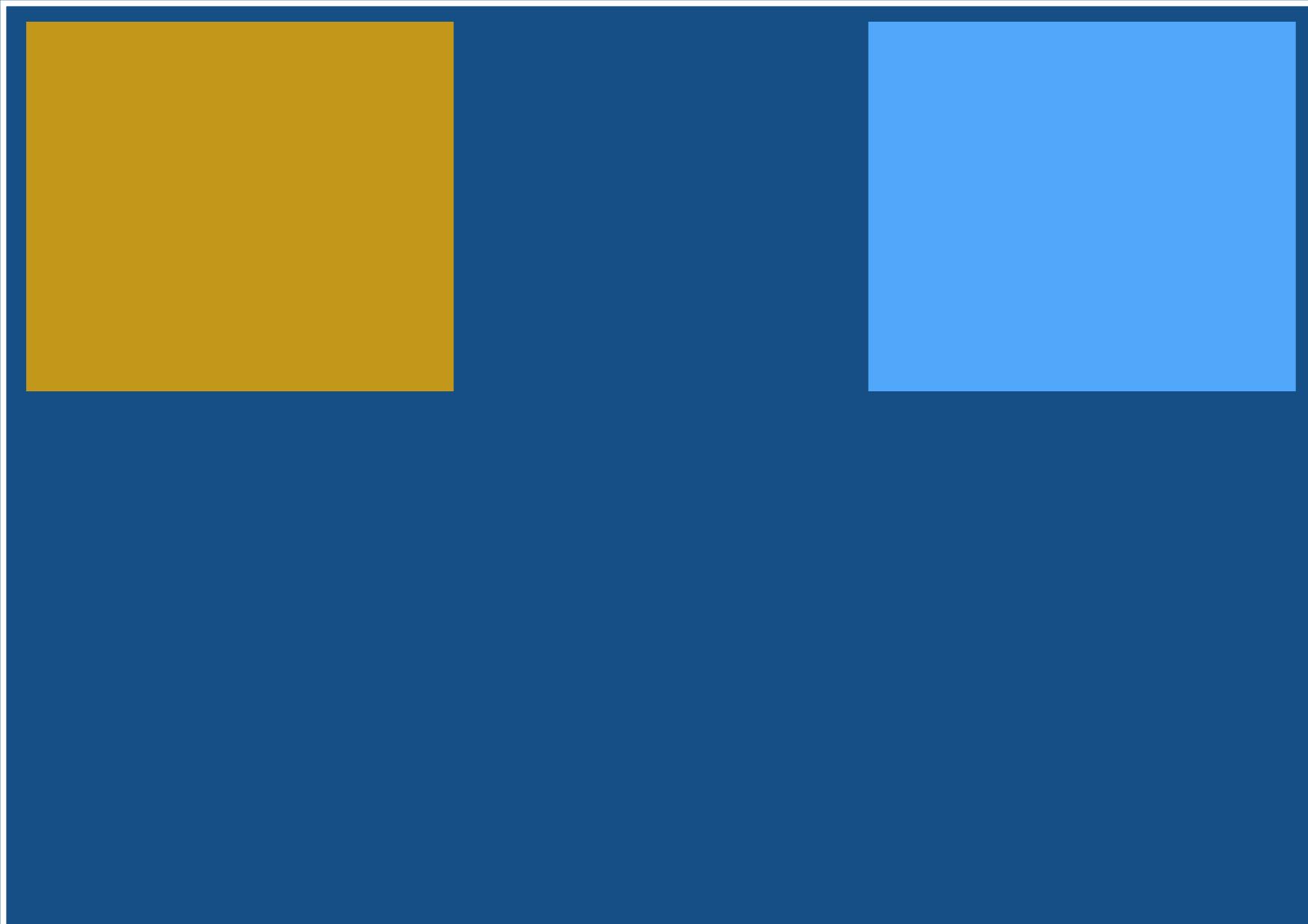
POSICIONAMIENTO FLOTANTE



POSICIONAMIENTO FLOTANTE

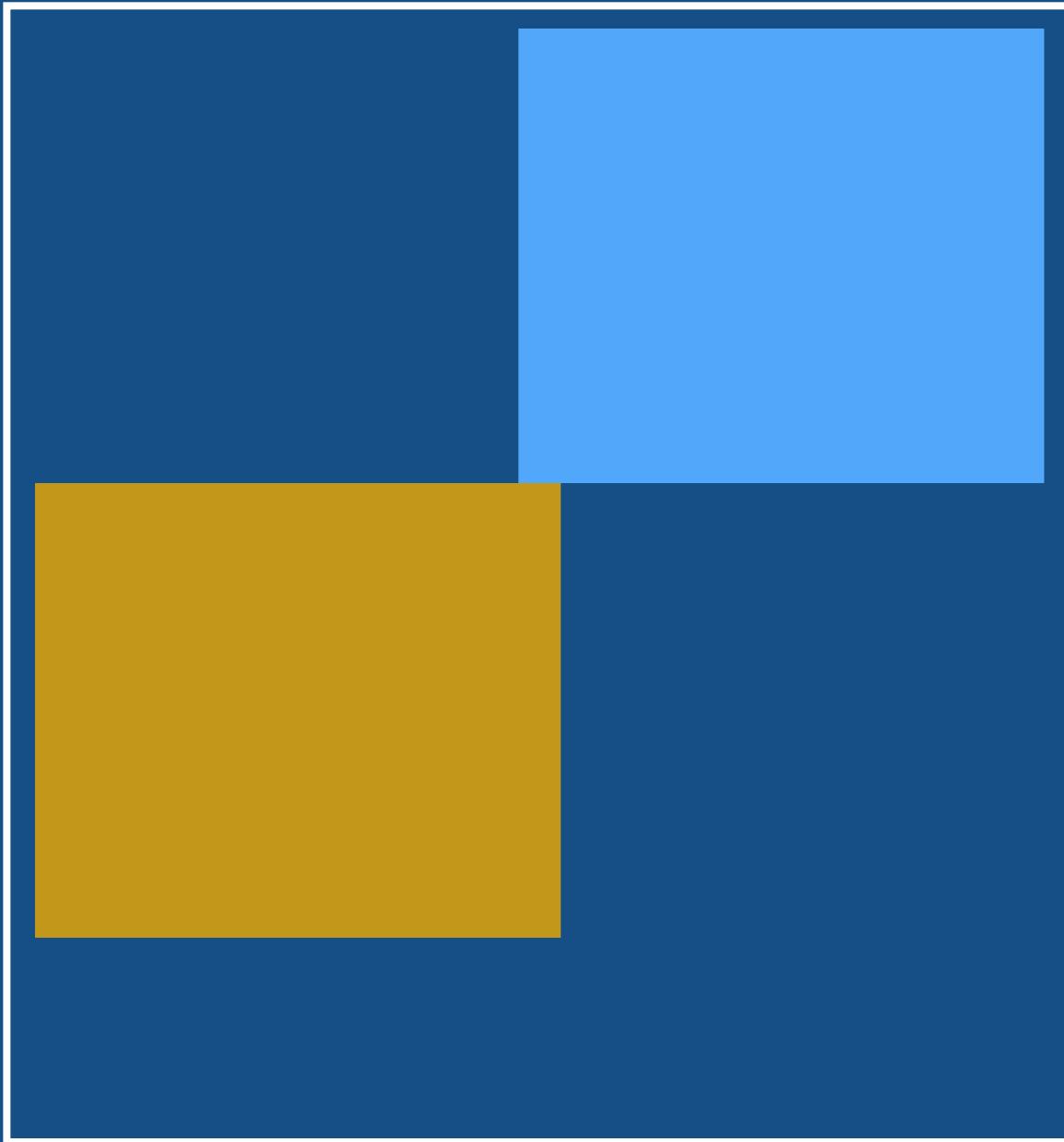


POSICIONAMIENTO FLOTANTE



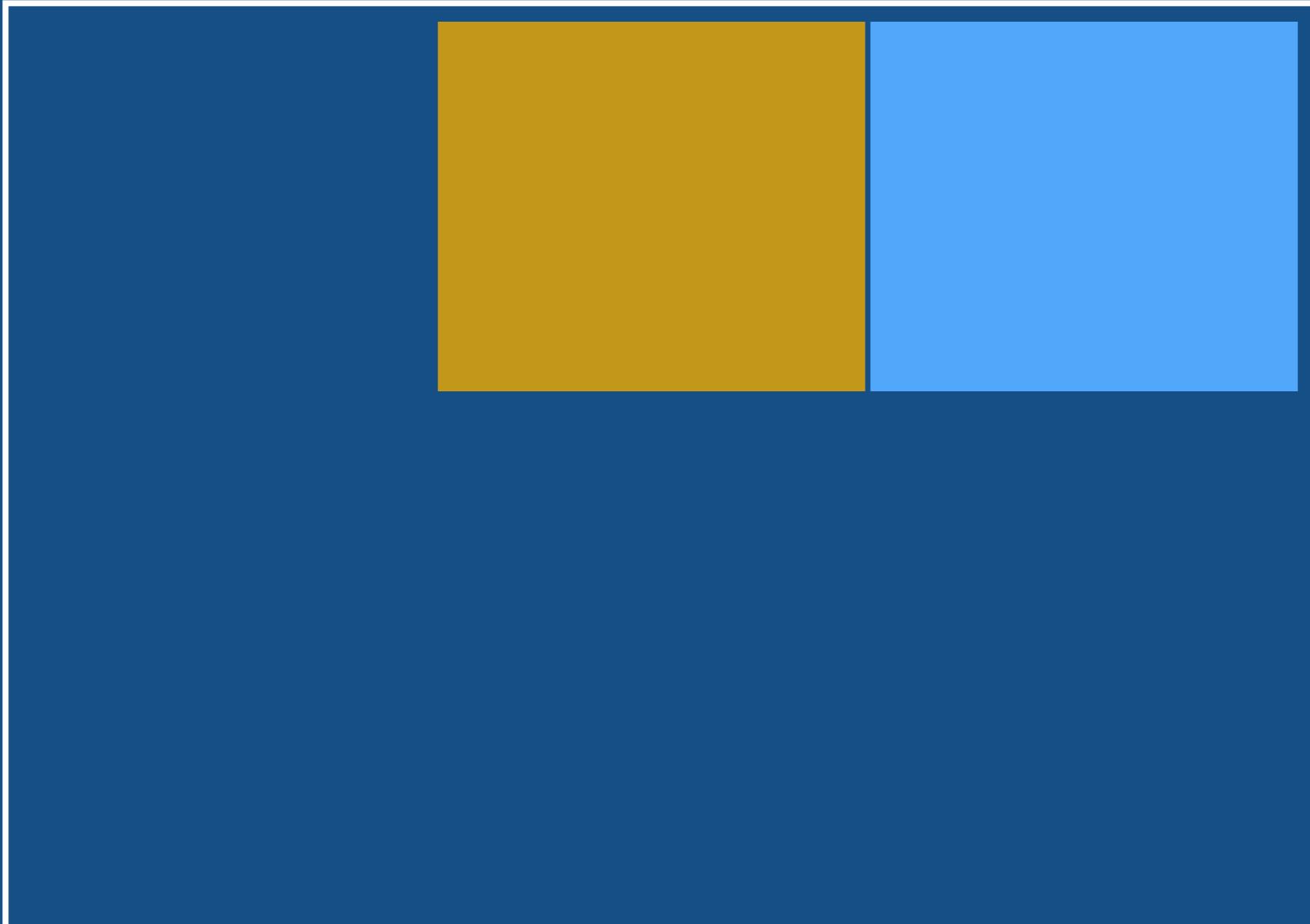
```
.fleft {  
    float: left;  
}  
  
.fright {  
    float: right;  
}
```

POSICIONAMIENTO FLOTANTE



```
.fleft {  
    float: left;  
}  
  
.fright {  
    float: right;  
}
```

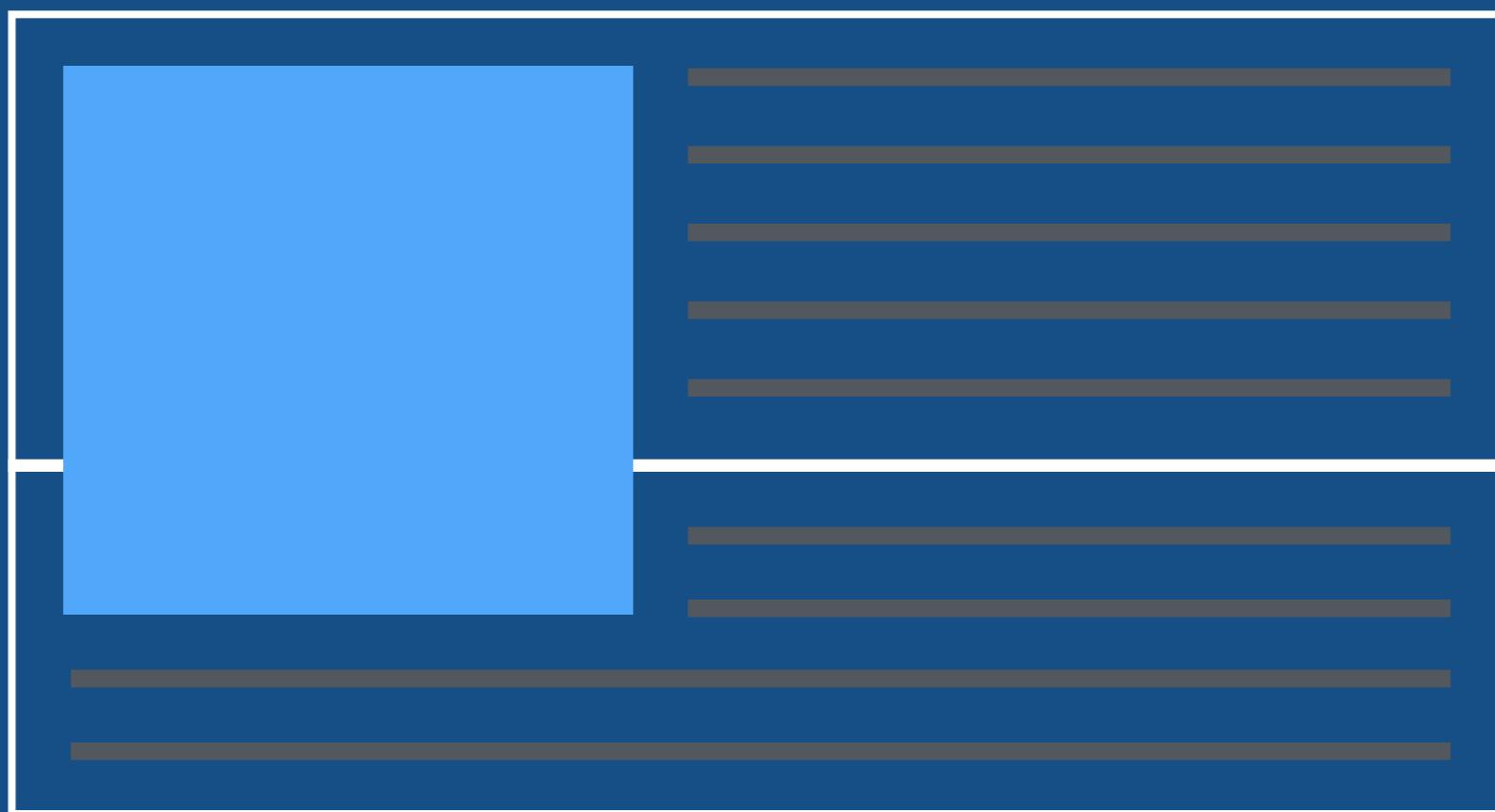
POSICIONAMIENTO FLOTANTE



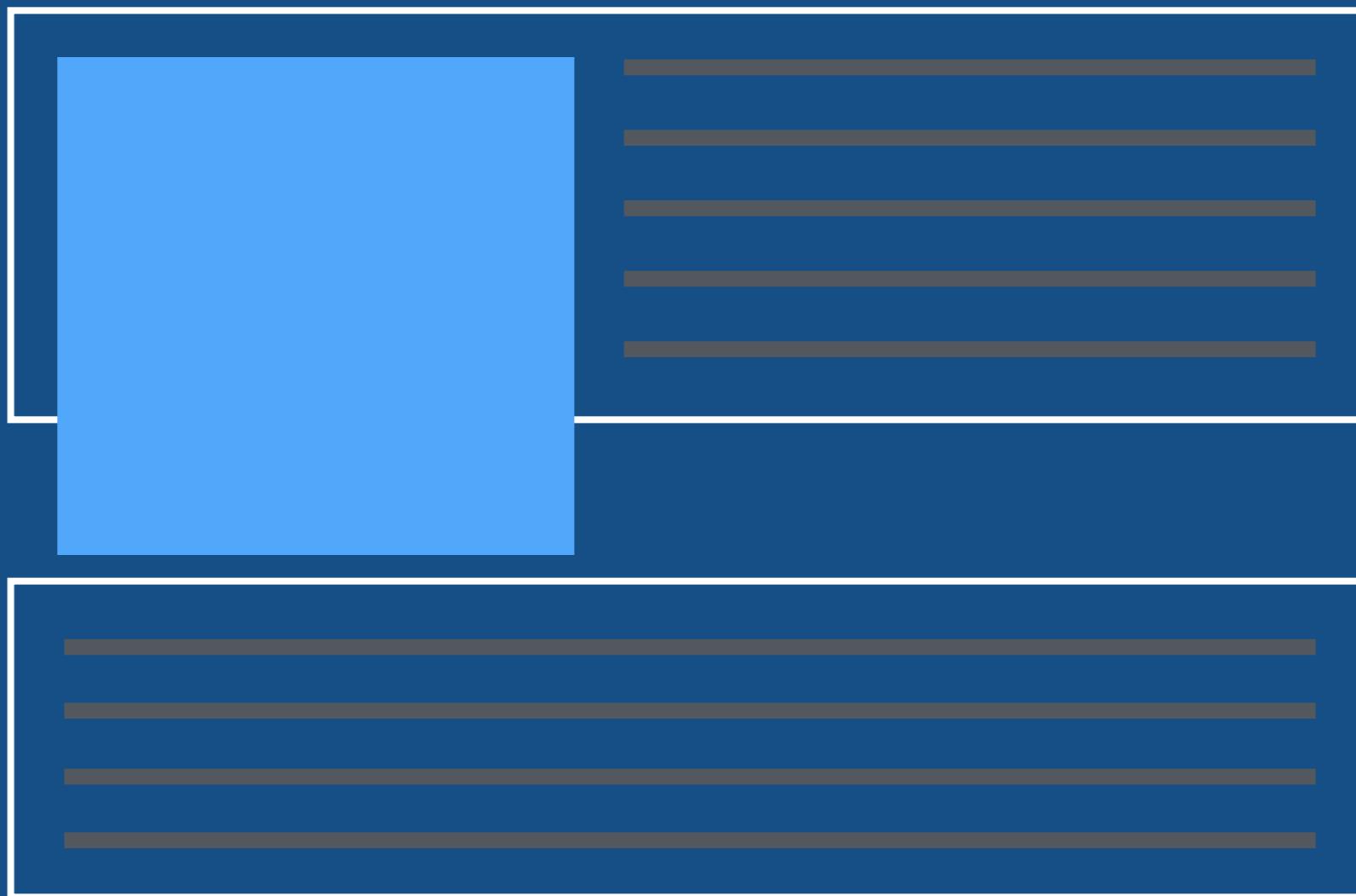
```
.fright {  
    float: right;  
}
```

POSICIONAMIENTO FLOTANTE

En algunas ocasiones, el elemento que flota, puede ser mayor (en altura) que su contenedor, interfiriendo con el resto de las cajas.



POSICIONAMIENTO FLOTANTE



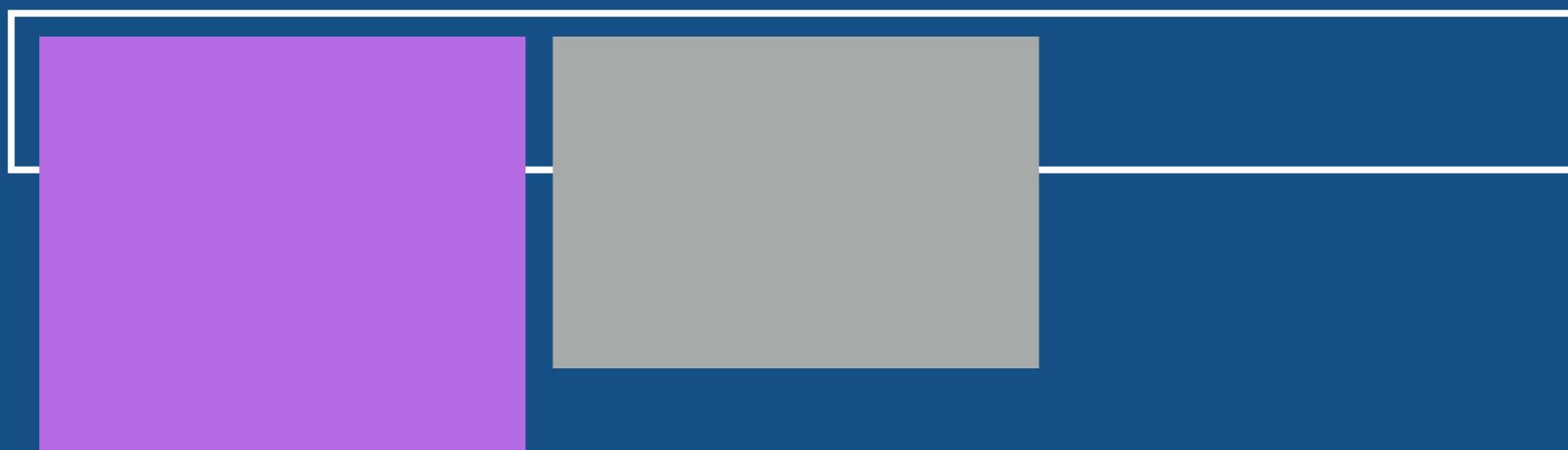
PROPIEDAD CLEAR

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

Si se indica el valor **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**. El valor **right** se comporta de manera análoga.

El valor **both** despeja los lados izquierdo y derecho del elemento.

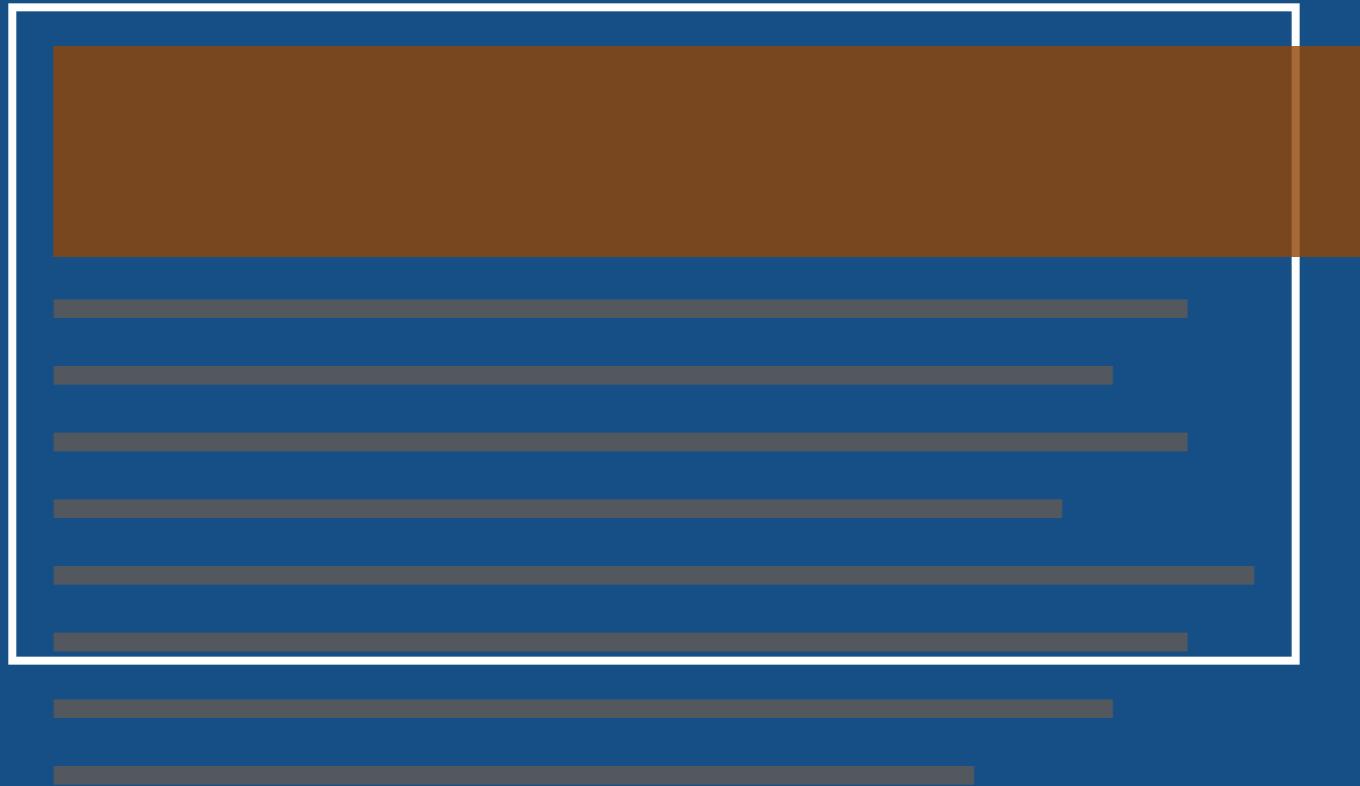
PROPIEDAD CLEAR

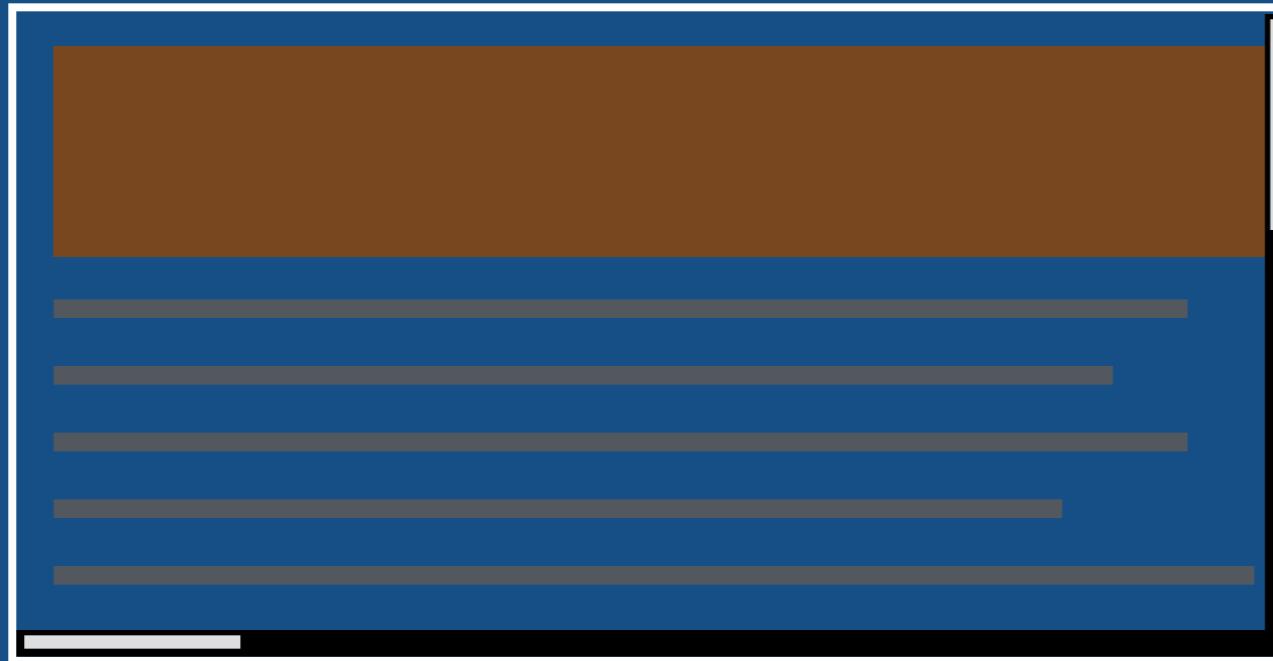


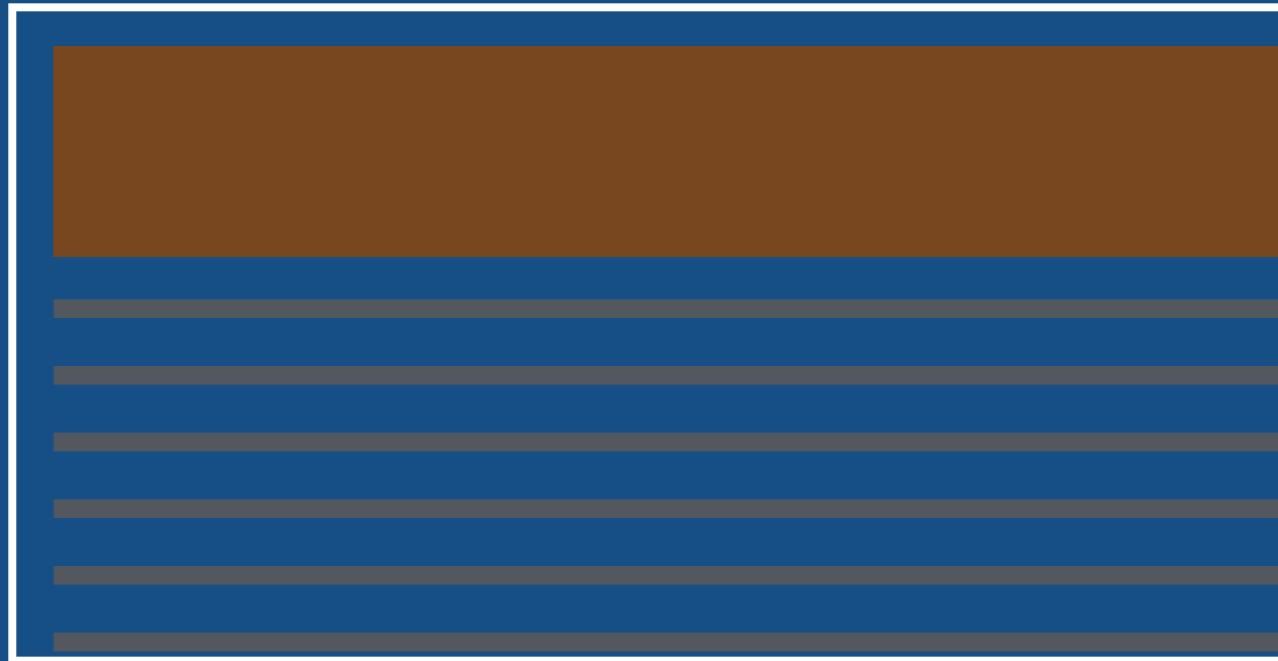
PROPIEDAD CLEAR

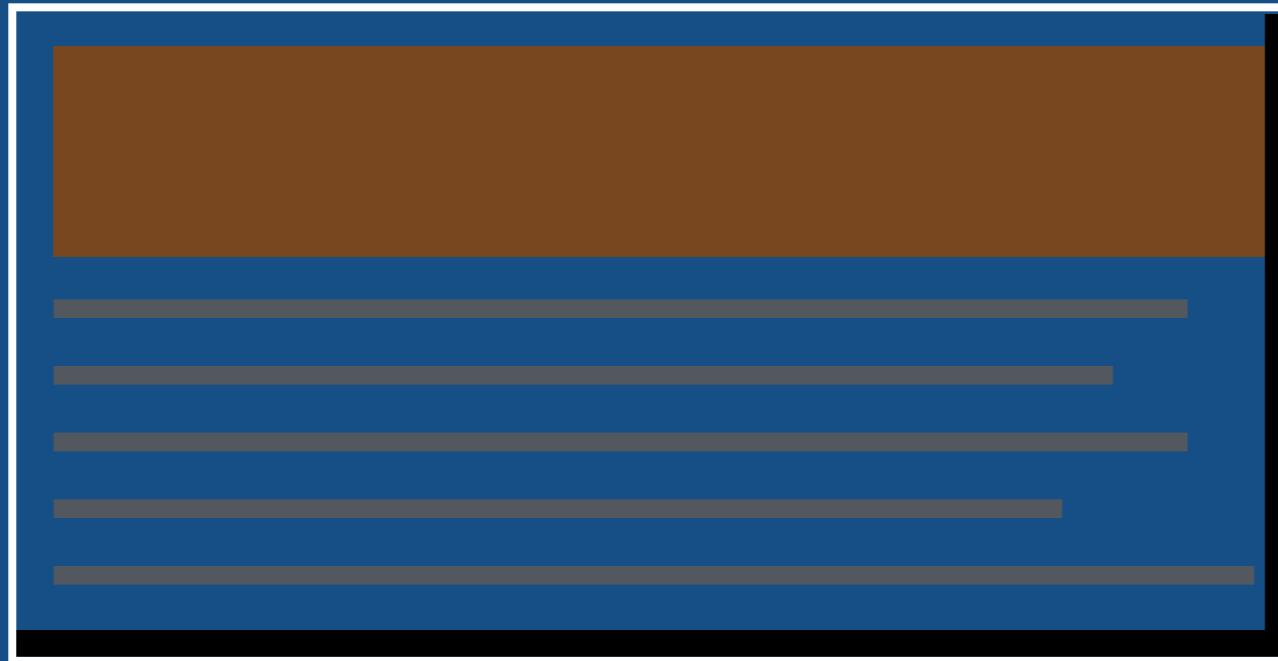


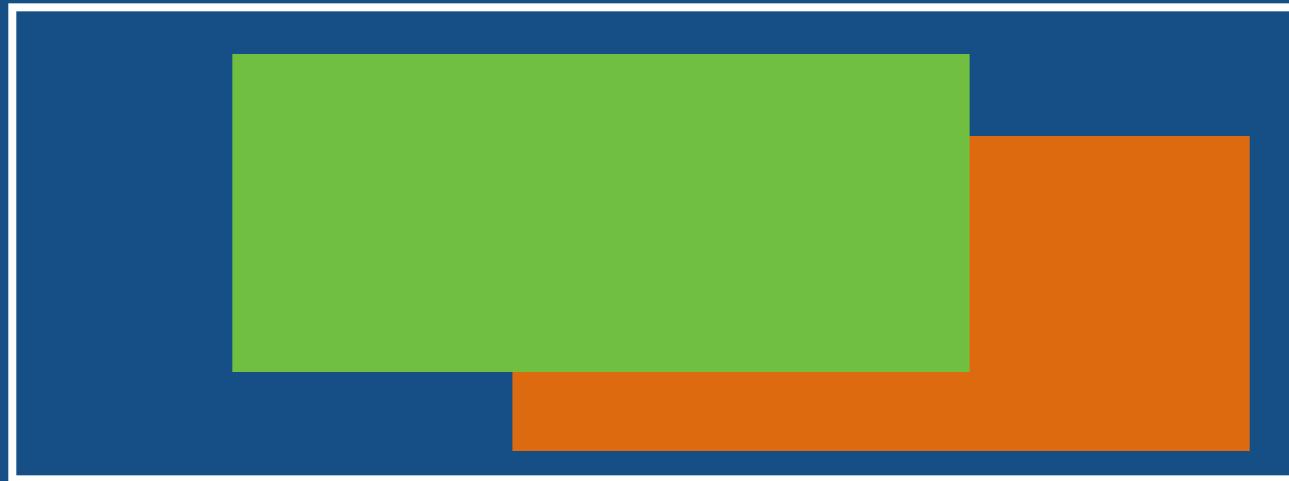
```
.clearfix:before, .clearfix:after {  
    content: "";  
    display: table;  
}  
.clearfix:after {  
    clear: both;  
}  
.clearfix {  
    zoom: 1;  
}
```





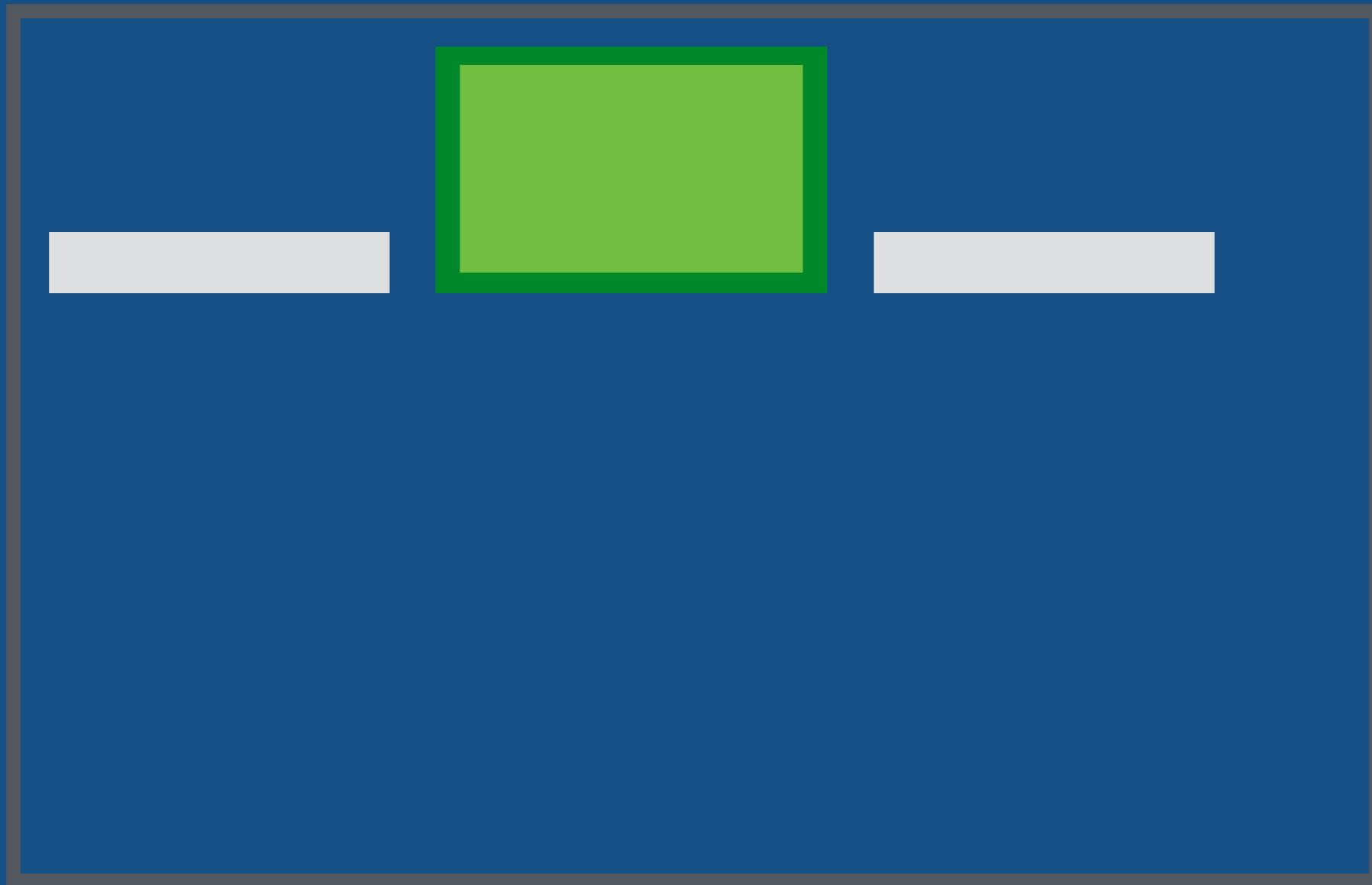












.sidebar

Home
Book Trip
FAQs

.content

Rentals
Tips
Packages



.sidebar

Home
Book Trip
FAQs

.content

Rentals
Tips
Packages



.sidebar

- ul.featured**
- Home*
- Book Trip*
- FAQs*

.content

- ul.featured**
- Rentals*
- Tips*
- Packages*

I.
INTRO-
DUCCIÓN

2.
SELEC-
TORES

3.
UNIDADES
DE
MEDIDA Y
COLORES

4.
MODELO
DE
CAJAS

5.
POSICIO-
NAMIENTO

6.
TEXTO

7.
ENLACES

8.
LISTAS

9.
FORMU-
LARIOS

10.
TABLAS

II.
LAYOUT
BÁSICO

12.
OTROS

CSS define numerosas propiedades para modificar la apariencia del texto. A pesar de que no dispone de tantas posibilidades como los lenguajes y programas específicos para crear documentos impresos, CSS permite aplicar estilos complejos y muy variados al texto de las páginas web.

TIPOGRAFÍA

La propiedad básica que define CSS relacionada con la tipografía se denomina **color** y se utiliza para establecer el color de la letra.

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

EXCEPCIÓN

El **valor** de la propiedad color se hereda pero hay una excepción: **los enlaces**.

```
/* Establece el mismo color a todos los elementos  
de la página salvo los enlaces */  
body { color: #777; }  
  
/* Establece el mismo color a todos los elementos  
de la página, incluyendo los enlaces */  
body, a { color: #777; }
```

TIPOGRAFÍA

La otra propiedad básica que define CSS relacionada con la tipografía se denomina **font-family** y se utiliza para indicar el tipo de letra con el que se muestra el texto.

```
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;
```

TIPOGRAFÍA

Una vez seleccionado el tipo de letra, se puede modificar su tamaño mediante la propiedad:

`font-size`

TAMAÑO
ABSOLUTO

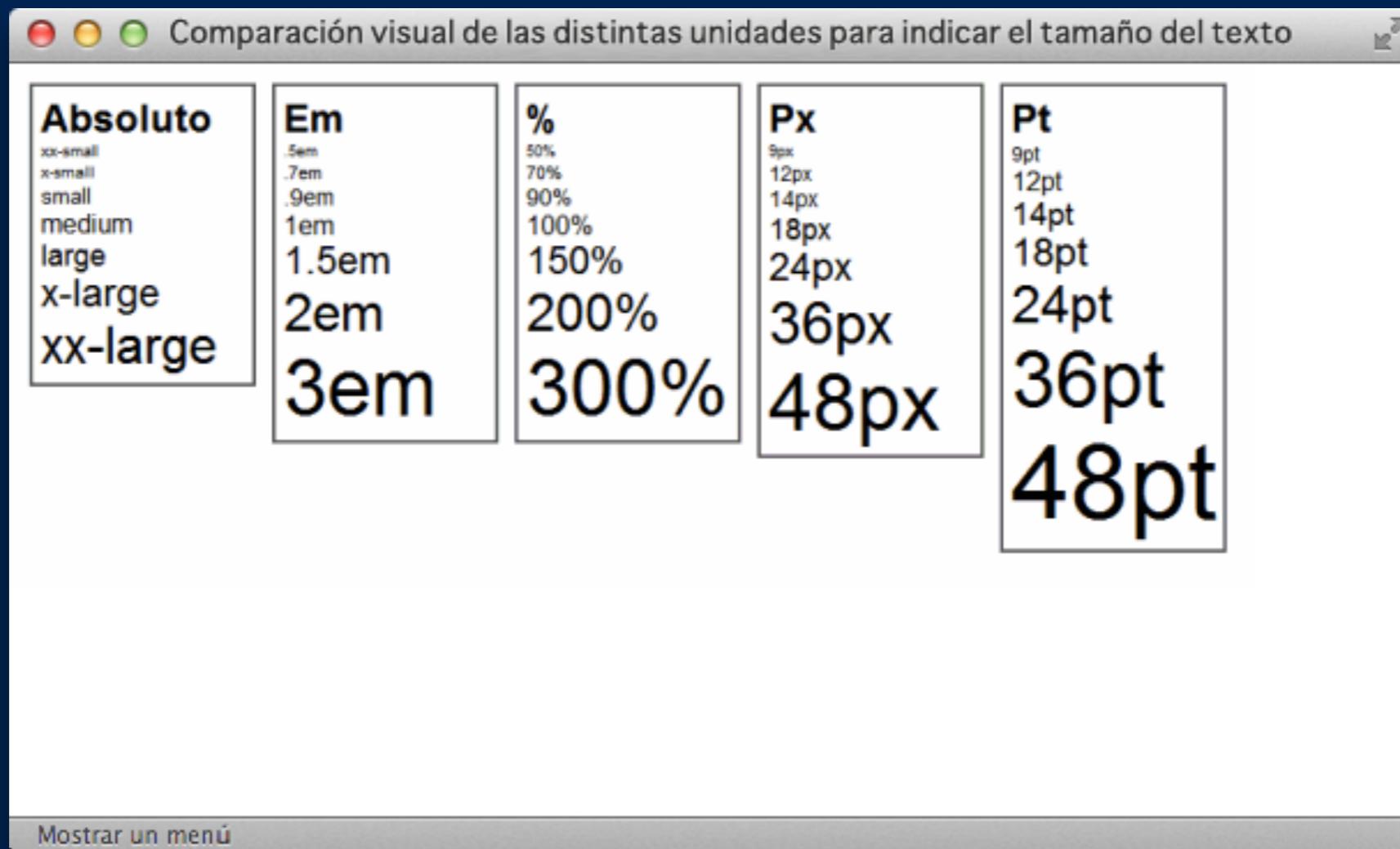
`xx-small`
`x-small`
`small`
`medium`
`large`
`x-large`
`xx-large`

TAMAÑO
RELATIVO

`larger`
`smaller`

Referencia:
tamaño de letra
del elemento
padre.

TIPOGRAFÍA



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

TIPOGRAFÍA

La propiedad que controla la anchura de la letra es:

`font-weight`

VALORES

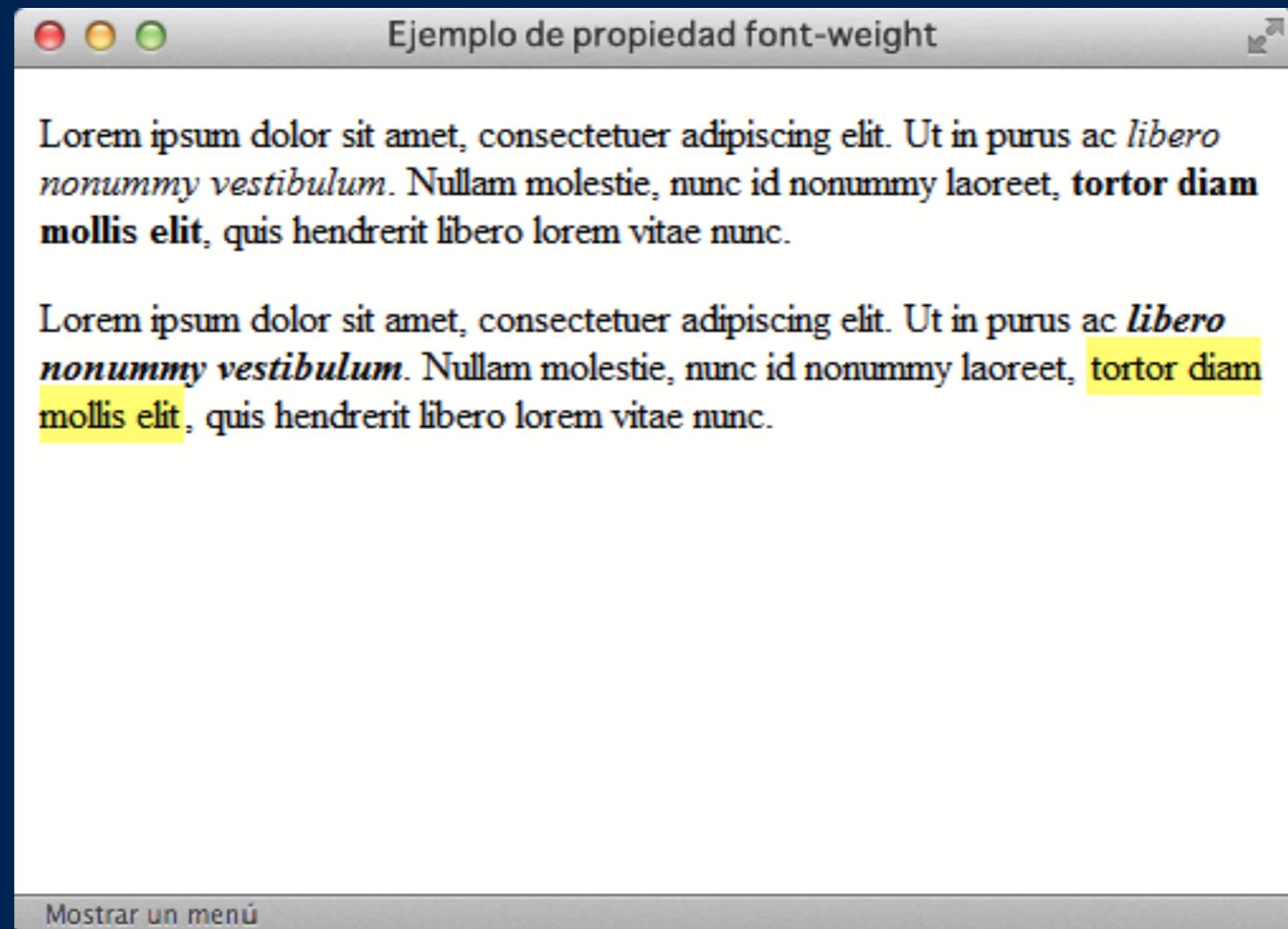
`normal`
`bold`
`bolder`
`lighter`
`inherit`

Los valores que normalmente se utilizan son `normal` (el valor por defecto) y `bold` para los textos en negrita. El valor `normal` equivale al valor numérico `400` y el valor `bold` al valor numérico `700`.

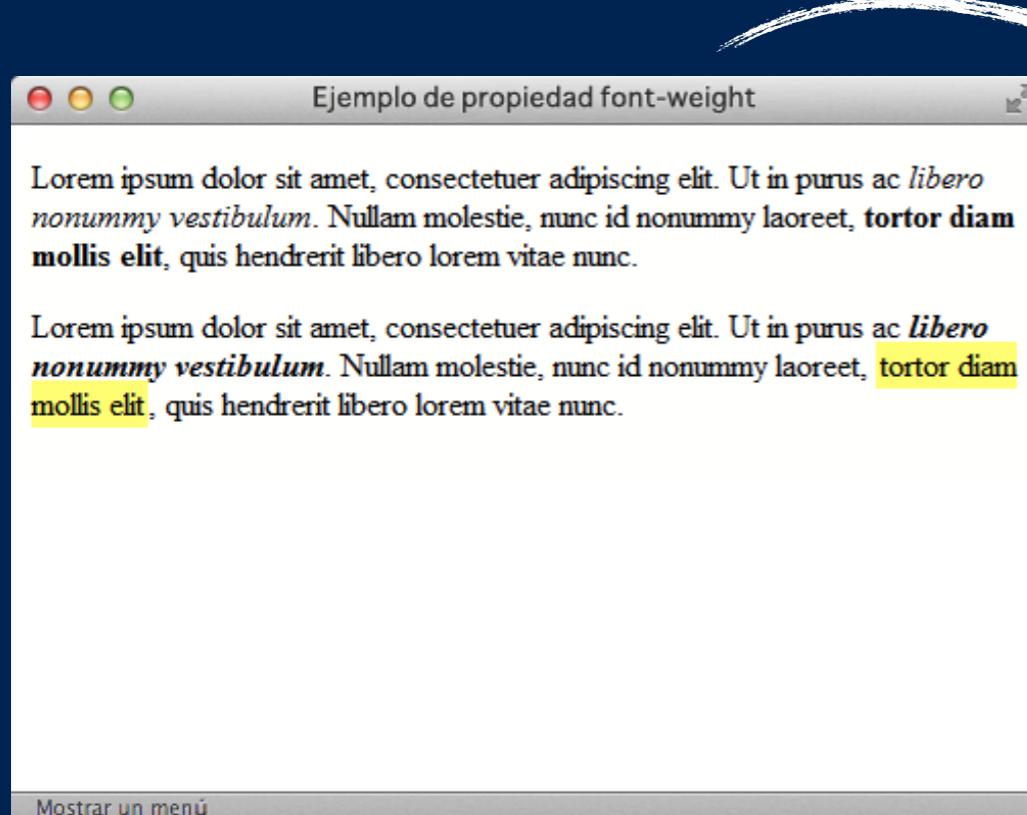
VALORES
NUMÉRICOS

`100`
`200`
`300`
`400`
`500`
`600`
`700`
`800`
`900`

TIPOGRAFÍA



TIPOGRAFÍA



```
#especial em {  
    font-weight: bold;  
}  
  
#especial strong {  
    font-weight: normal;  
    background-color: #FFFF66;  
    padding: 2px;  
}
```

```
<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut in  
purus ac <em>libero nonummy vestibulum</em>. Nullam molestie, nunc id  
nonummy laoreet, <strong>tortor diam mollis elit</strong>, quis hendrerit  
libero lorem vitae nunc.</p>  
  
<p id="especial">Lorem ipsum dolor sit amet, consectetur adipiscing elit.  
Ut in purus ac <em>libero nonummy vestibulum</em>. Nullam molestie, nunc id  
nonummy laoreet, <strong>tortor diam mollis elit</strong>, quis hendrerit  
libero lorem vitae nunc.</p>
```

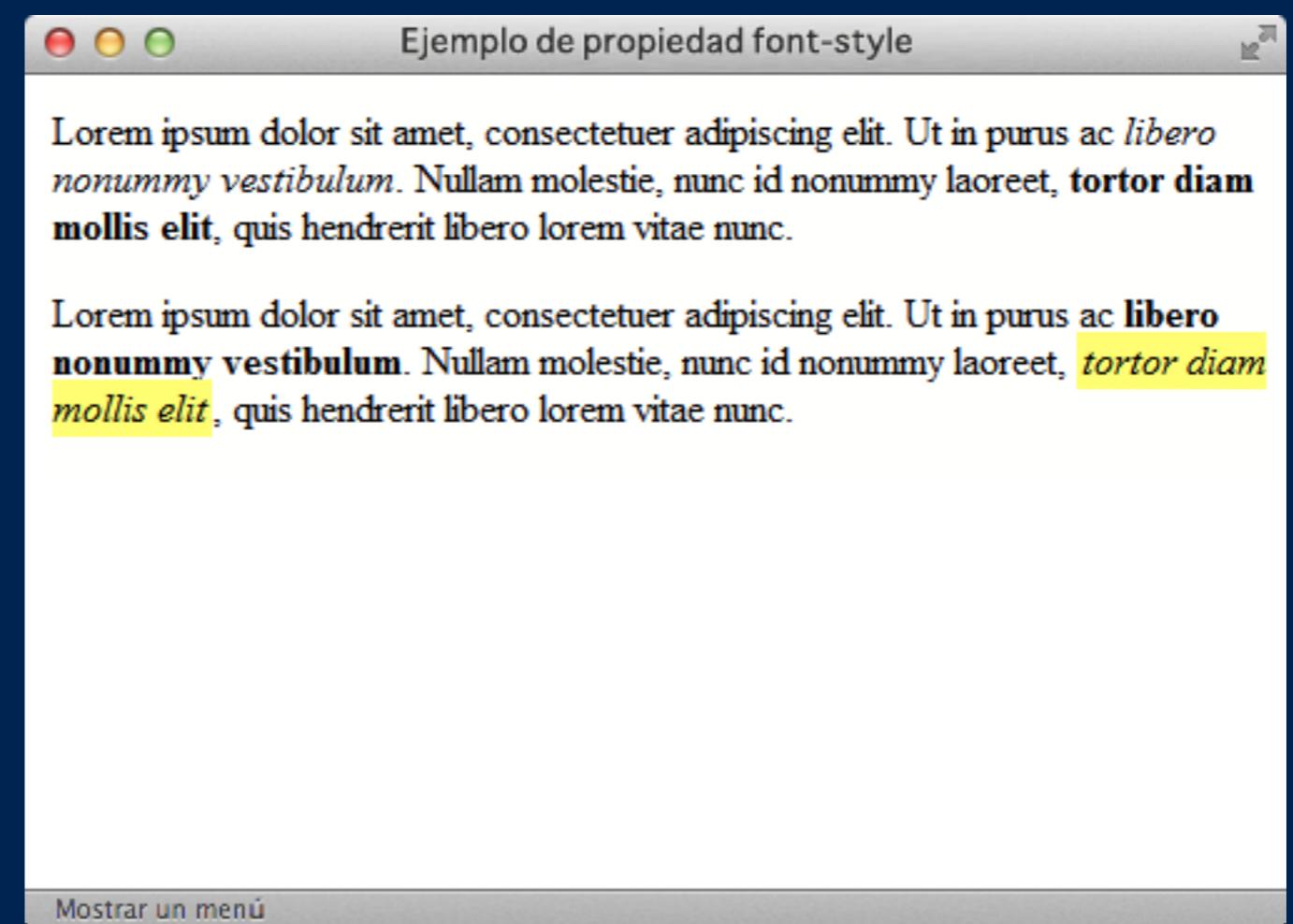
TIPOGRAFÍA

Además de la anchura de la letra, CSS permite variar su estilo mediante la propiedad:

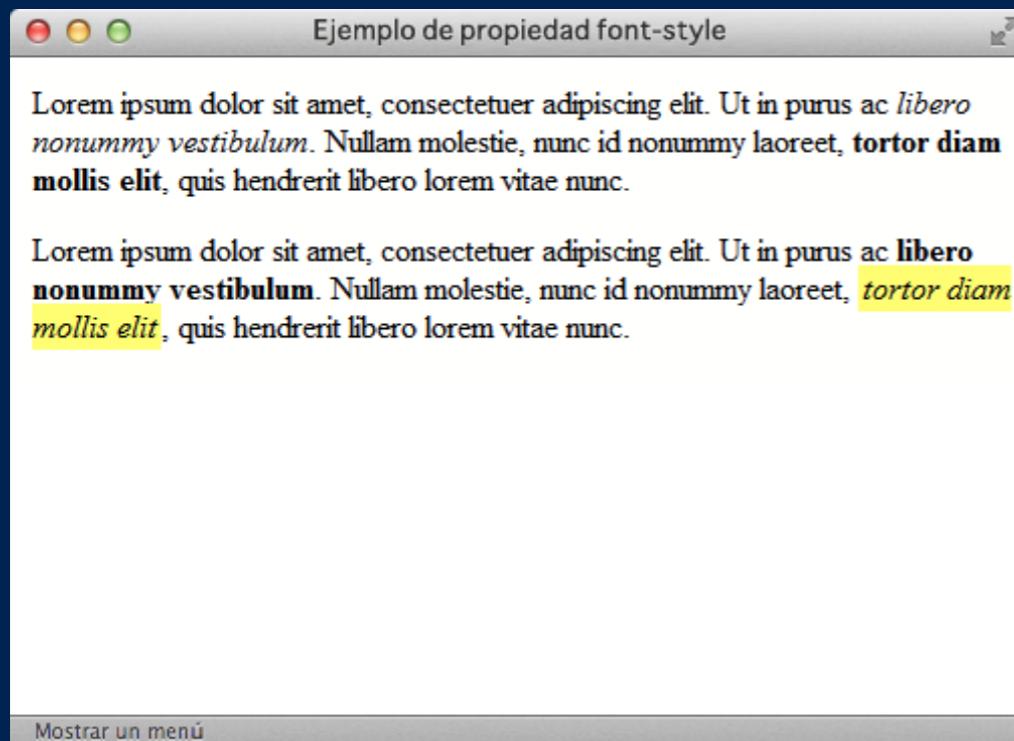
font-style

VALORES

normal
italic
oblique
inherit



TIPOGRAFÍA



```
#especial em {  
    font-weight: bold;  
    font-style: normal;  
}  
  
#especial strong {  
    font-weight: normal;  
    font-style: italic;  
    background-color:#FFFF66;  
    padding: 2px;  
}
```

Ahora, el texto del elemento `` se muestra como un texto en negrita y el texto del elemento `` se muestra como un texto en cursiva con un color de fondo muy destacado.

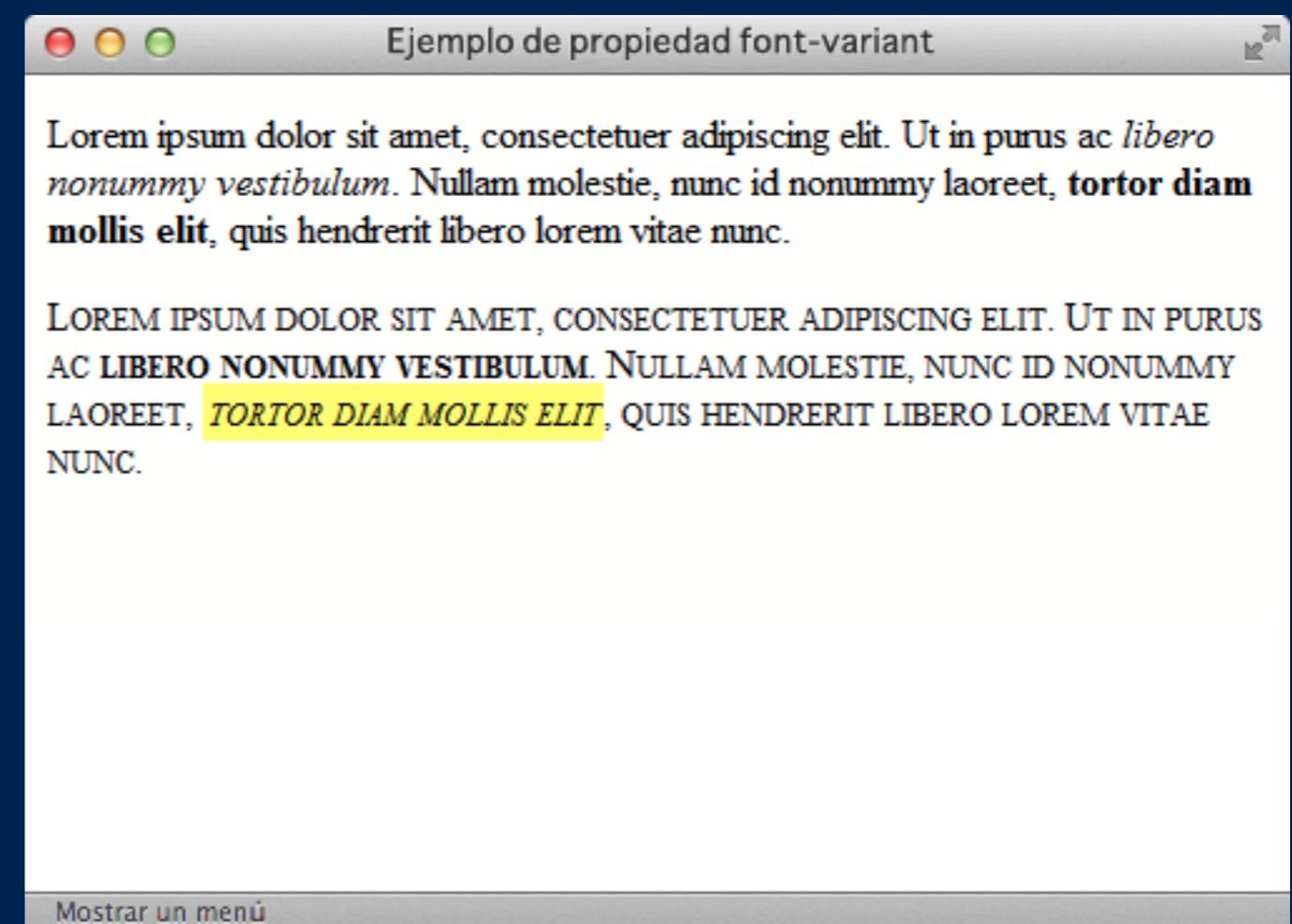
TIPOGRAFÍA

CSS permite otra variación en el estilo del tipo de letra, controlado mediante la propiedad:

font-variant

VALORES

normal
small-caps
inherit



TIPOGRAFÍA

Por otra parte, CSS proporciona una propiedad tipo "shorthand" denominada **font** y que permite indicar de forma directa algunas o todas las propiedades de la tipografía de un texto.

```
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;
```

TEXTO

Además de las propiedades relativas a la tipografía del texto, CSS define numerosas propiedades que determinan la apariencia del texto en su conjunto.

La propiedad que define la alineación del texto se denomina:

`text-align`

TEXTO

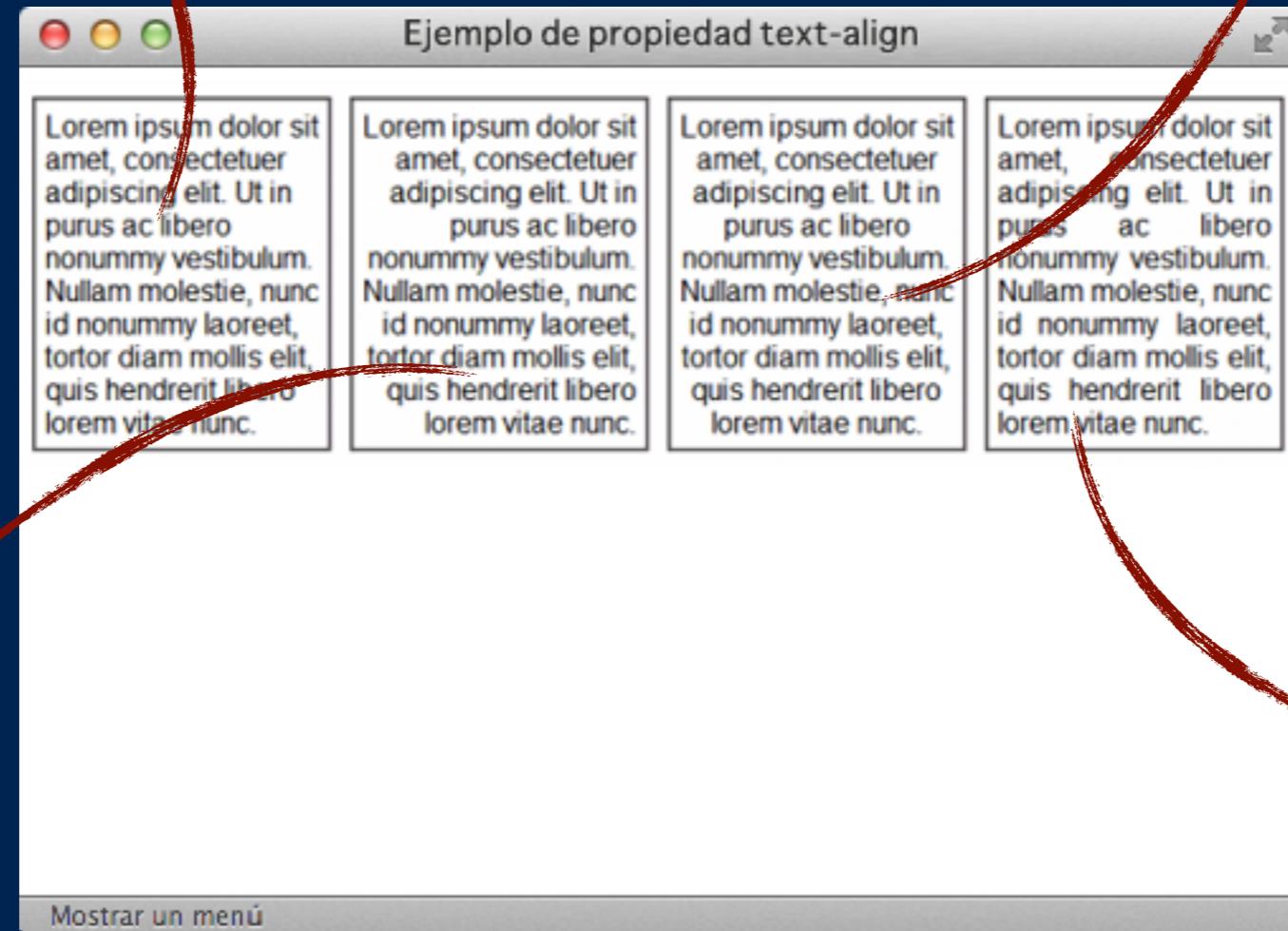
left

center

text-align

right

justify



TEXTO

El interlineado de un texto se controla mediante la propiedad:

line-height

Además de todas las unidades de medida y el uso de porcentajes, esta propiedad permite indicar un número sin unidades que se interpreta como el múltiplo del tamaño de letra del elemento. Por 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 }
```

TEXTO

line-height

line-height: 1.2em

line-height: 1.5em

line-height: 2em

The screenshot shows a Mac OS X application window titled "Ejemplo de propiedad line-height". Inside the window, there are three vertical columns of text, each demonstrating a different line height setting. The first column has a line height of 1.2em, the second of 1.5em, and the third of 2em. The text in all columns is identical: "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. Praesent eu eros id magna tempor aliquam. Nam et enim. Curabitur ornare vehicula quam. Donec dui nulla, iaculis sed, venenatis a, vulputate non, ligula. Vestibulum gravida, tortor a eleifend venenatis, sem". The window has the standard OS X title bar with red, yellow, and green buttons, and a close button in the top right corner.

TEXTO

La propiedad que decora el texto se denomina:

`text-decoration`



VALORES

none
underline
overline
line-through
blink
inherit

TEXTO

Una propiedad bastante desconocida que puede variar de forma sustancial el aspecto del texto es:

text-transform

VALORES

capitalize
uppercase
lowercase
none
inherit



Ejemplo de propiedad text-transform

Original
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.

Text-transform: Capitalize
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.

text-transform: lowercase
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.

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.

Mostrar un menú

TEXTO

text-transform

Las reglas CSS del ejemplo anterior son:

```
<div style="text-transform: none">
  <h1>Original</h1>
  Lorem ipsum dolor sit amet...</div>

<div style="text-transform: capitalize">
  <h1>text-transform: capitalize</h1>
  Lorem ipsum dolor sit amet...
</div>

<div style="text-transform: lowercase">
  <h1>text-transform: lowercase</h1>
  Lorem ipsum dolor sit amet...
</div>

<div style="text-transform: uppercase">
  <h1>text-transform: uppercase</h1>
  Lorem ipsum dolor sit amet...
</div>
```

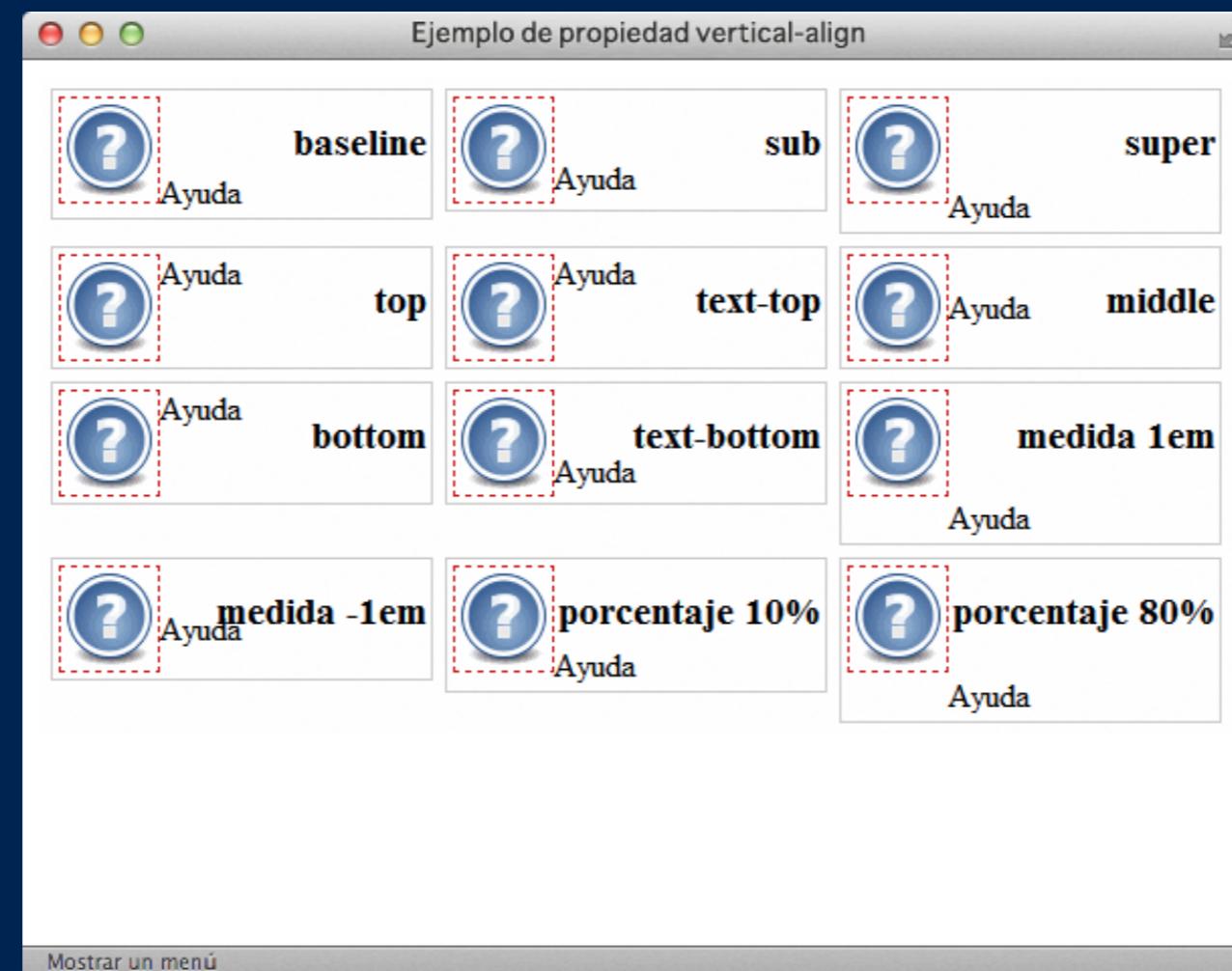
TEXTO

Uno de los principales problemas del diseño de documentos y páginas mediante CSS consiste en la alineación vertical en una misma línea de varios elementos diferentes como imágenes y texto. Para controlar esta alineación, CSS define la propiedad:

vertical-align

TEXTO

vertical-align



El valor por defecto es baseline y el valor más utilizado cuando se establece la propiedad vertical-align es middle.

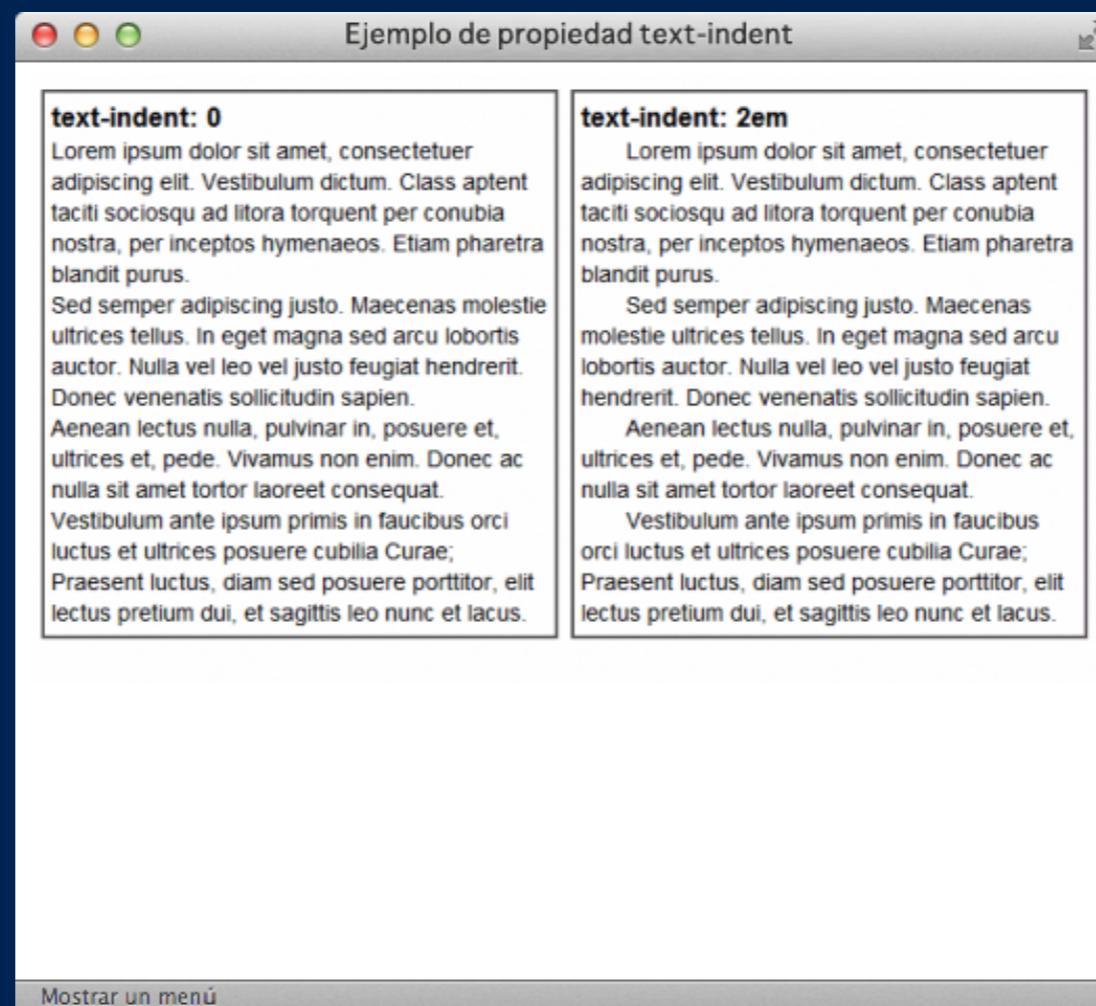
TEXTO

En muchas publicaciones impresas suele ser habitual tabular la primera línea de cada párrafo para facilitar su lectura. CSS permite controlar esta tabulación mediante la propiedad:

`text-indent`

TEXTO

text-indent



La siguiente imagen muestra la comparación entre un texto largo formado por varios párrafos sin tabular y el mismo texto con la primera línea de cada párrafo tabulada.

TEXTO

CSS también permite controlar la separación entre las letras que forman las palabras y la separación entre las palabras que forman los textos. La propiedad que controla la separación entre letras se llama

letter-spacing

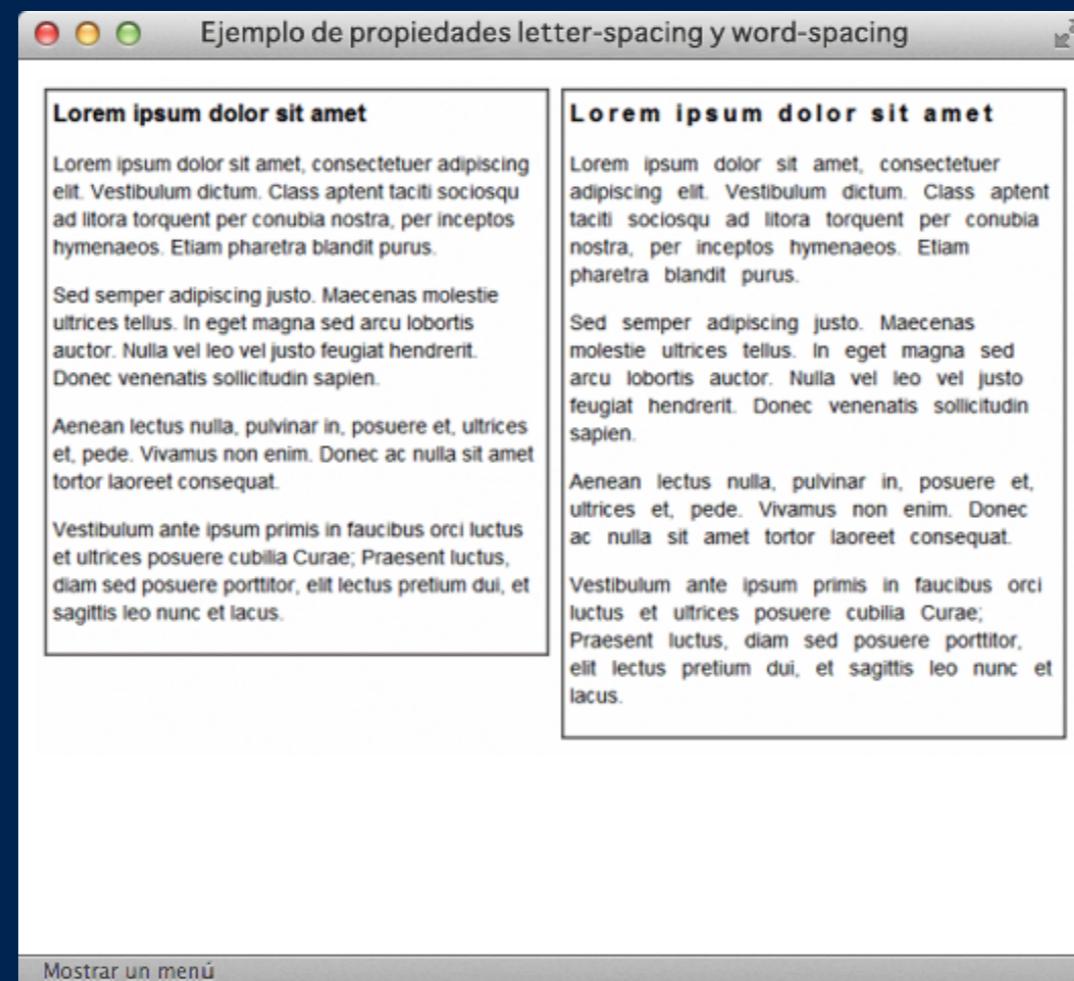
y la separación entre palabras se controla mediante

word-spacing

TEXTO

letter-spacing

word-spacing



La siguiente imagen muestra la comparación entre un texto normal y otro con las propiedades aplicadas.

TEXTO

Las reglas CSS del ejemplo anterior se muestran a continuación:

```
.especial h1 { letter-spacing: .2em; }  
.especial p { word-spacing: .5em; }
```

```
<div>  
  <h1>Lorem ipsum dolor sit amet</h1>  
  <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vestibulum  
  dictum. Class aptent taciti sociosqu ad litora torquent per conubia nostra,  
  per inceptos hymenaeos. Etiam pharetra blandit purus.</p>  
  ...  
</div>  
  
<div class="especial">  
  <h1>Lorem ipsum dolor sit amet</h1>  
  <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vestibulum  
  dictum. Class aptent taciti sociosqu ad litora torquent per conubia nostra,  
  per inceptos hymenaeos. Etiam pharetra blandit purus.</p>  
  ...  
</div>
```

TEXTO

CSS también permite controlar el tratamiento de los espacios en blanco de los textos mediante la propiedad:

white-space



VALORES

normal
pre
nowrap
pre-wrap
pre-line
inherit

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.

TEXTO

white-space

Ejemplo de propiedad white-space

white-space: pre

 Lorem ipsum dolor sit amet, consectetuer 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, consectetuer
 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 torquent per conubia nostra, per inceptos hymenaeos. Etiam pharetra blandit purus.

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.

Mostrar un menú

TEXTO

Por último, CSS define unos elementos especiales llamados "pseudo-elementos" que permiten aplicar estilos a ciertas partes de un texto. En concreto, CSS permite definir estilos especiales a la primera frase de un texto y a la primera letra de un texto.

Ejemplo de pseudo-elemento first-line

```
LOREM IPSUM DOLOR SIT AMET, CONSECTETUER ADIPISCING ELIT. VESTIBULUM DICTUM  
Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos.  
Etiam pharetra blandit purus.
```

```
LOREM IPSUM DOLOR SIT AMET, CONSECTETUER  
adipiscing elit. Vestibulum dictum. Class aptent taciti sociosqu  
ad litora torquent per conubia nostra, per inceptos hymenaeos.  
Etiam pharetra blandit purus.
```

```
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.
```

Mostrar un menú

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

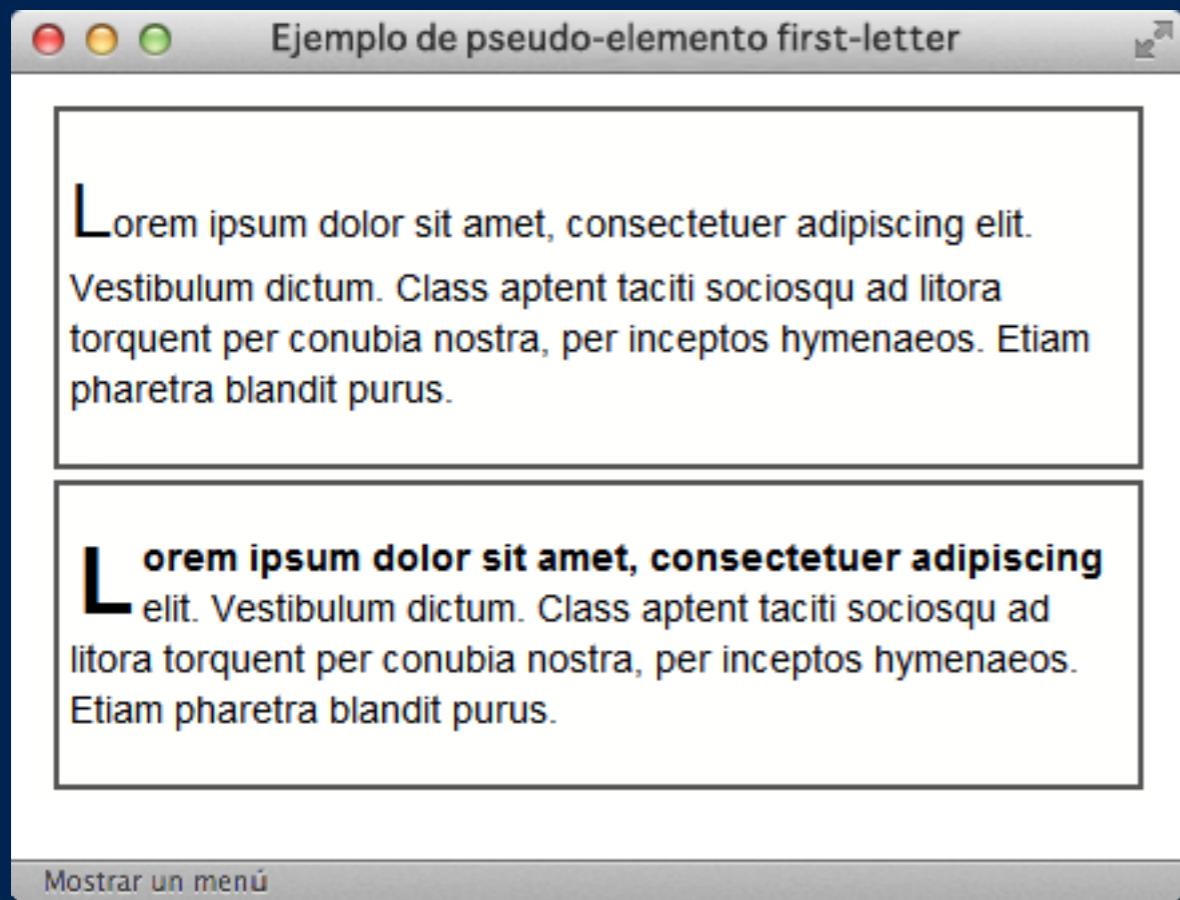
Ejemplo de pseudo-elemento first-letter

```
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.
```

```
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.
```

Mostrar un menú

TEXTO



```
#normal p:first-letter {  
    font-size: 2em;  
}  
  
#avanzado p:first-letter {  
    font-size: 2.5em;  
    font-weight: bold;  
    line-height: .9em;  
    float: left;  
    margin: .1em;  
}  
  
#avanzado p:first-line {  
    font-weight: bold;  
}
```

```
<div id="normal">  
    <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vestibulum  
    dictum. Class aptent taciti sociosqu ad litora torquent per conubia  
    nostra,per inceptos hymenaeos. Etiam pharetra blandit purus.</p>  
</div>  
  
<div id="avanzado">  
    <p>Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Vestibulum  
    dictum. Class aptent taciti sociosqu ad litora torquent per conubia  
    nostra, per inceptos hymenaeos. Etiam pharetra blandit purus.</p>
```