

Programación Aplicada a la Multimedia CSS 2.1

GABRIEL LEON PAREDES

gleon@ups.edu.ec

www.linkedin.com/in/gabrielleonp

Cloud Computing, Smart Cities & High Performance Computing
Cuenca, Ecuador



Introducción

- Las hojas de estilo en cascada (**Cascading Style Sheets**, CSS) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML
- Permiten separar la estructura de un documento de su presentación, es decir, el diseño del contenido
- Permite generar “imagen de marca”.
- Existen pequeñas diferencias entre los distintos navegadores.

CSS 2.1

- Permite aplicar al documento formato de modo mucho más exacto:
 - Podemos definir la distancia entre líneas del documento, párrafo, etc.
 - Se puede aplicar indentado, color de fondo, imagen de fondo, etc., a las primeras líneas del párrafo
 - Podemos colocar elementos en la pagina con mayor precisión
 - Definir la visibilidad de los elementos, márgenes, subrayados, tachados...
 - Medidas en: píxeles (px), porcentaje (%),pulgadas (in), puntos (pt), centímetros (cm)

Documentación

- Estándar CSS2.1: <http://www.w3.org/TR/2004/CR-CSS21-20040225/>
- Traducción del estándar CSS2.1: http://www.w3.org/Style/css21-updates/css21_spa.pdf
- Tutorial en ingles con ejemplos <http://www.w3schools.com/>
- Tutorial en castellano <http://www.tejedoresdelweb.com/w/CSS>
- Libro en castellano <http://librosweb.es/libro/css/>
- Validador CSS2.1: <http://jigsaw.w3.org/css-validator/>

Funcionamiento

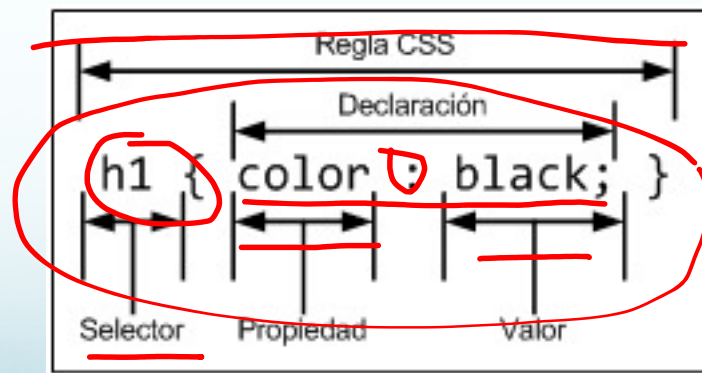
- Antes de que se generalizara el uso de CSS, los diseñadores de páginas web utilizaban etiquetas HTML especiales para modificar el aspecto de los elementos de la página.

```
<body>
  <h1><font color="red" face="Arial" size="5">Titular de la página</font></h1>
  <p><font color="gray" face="Verdana" size="2">Un párrafo de texto no muy largo.</font></p>
</body>
```

- El problema de utilizar este método para definir el aspecto de los elementos se puede ver claramente con el siguiente ejemplo: **si la página tuviera 50 elementos diferentes, habría que insertar 50 etiquetas . Si el sitio web entero se compone de 10.000 páginas diferentes, habría que definir 500.000 etiquetas . Como cada etiqueta tiene tres atributos, habría que definir 1.5 millones de atributos.**

Glosario básico de CSS

- Los diferentes términos se definen a continuación:
 - **Regla:** cada uno de los estilos que componen una hoja de estilos CSS. Cada regla está compuesta de una parte de "selectores", un símbolo de "llave de apertura" ({}), otra parte denominada "declaración" y por último, un símbolo de "llave de cierre" ({}).
 - **Selector:** indica el elemento o elementos HTML a los que se aplica la regla CSS.
 - **Declaración:** especifica los estilos que se aplican a los elementos. Está compuesta por una o más propiedades CSS.
 - **Propiedad:** característica que se modifica en el elemento seleccionado, como por ejemplo su tamaño de letra, su color de fondo, etc.
 - **Valor:** establece el nuevo valor de la característica modificada en el elemento.



Ejemplo

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo de estilos con CSS</title>
<style type="text/css">
    h1{ color: red; font-family: Arial; font-size: large; }
    p { color: gray; font-family: Verdana; font-size: medium; }
</style>
</head>

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

text-style
bold

Incluir estilos en las documentos HTML

- Una de las principales características de CSS es su flexibilidad y las diferentes opciones que ofrece para realizar una misma tarea. De hecho, existen tres opciones para incluir CSS en un documento HTML.
 - 1. Incluir estilos en el mismo archivo:** Los estilos se definen en una zona específica del propio documento HTML. Se emplea la etiqueta **<style>** de HTML y solamente se pueden incluir en la cabecera del documento (sólo dentro de la sección **<head>**)
 - 2. Definir CSS en un archivo externo:** En este caso, todos los estilos CSS se incluyen en un archivo de tipo CSS que las páginas HTML enlazan mediante la etiqueta **<link>**. Un archivo de tipo CSS no es más que un archivo simple de texto cuya extensión es **.css** Se pueden crear todos los archivos CSS que sean necesarios y cada página HTML puede enlazar tantos archivos CSS como necesite.
 - 3. Incluir CSS en los elementos HTML:** El último método para incluir estilos CSS en documentos HTML es el peor y el menos utilizado, ya que tiene los mismos problemas que la utilización de las etiquetas ****.

Ejemplo 1

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo de estilos CSS en el propio documento</title>
<style type="text/css">
  p { color: black; font-family: Verdana; }
</style>
</head>

<body>
<p>Un párrafo de texto.</p>
</body>
</html>
```

Ejemplo 2

`p { color: black; font-family: Verdana; }` Archivo externo *.css

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo de estilos CSS en un archivo externo</title>
<link rel="stylesheet" type="text/css" href="/css/estilos.css" media="screen" />
</head>

<body>
<p>Un párrafo de texto.</p>
</body>
</html>
```

Ejemplo 3

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo de estilos CSS en el propio documento</title>
</head>

<body>
<p style="color: black; font-family: Verdana;">Un párrafo de texto.</p>
</body>
</html>
```

Comentarios

- CSS permite incluir comentarios entre sus reglas y estilos. Los comentarios son contenidos de texto que el diseñador incluye en el archivo CSS para su propia información y utilidad. Los navegadores ignoran por completo cualquier comentario de los archivos CSS, por lo que es común utilizarlos para estructurar de forma clara los archivos CSS complejos.
- El comienzo de un comentario se indica mediante los caracteres `/*` y el final del comentario se indica mediante `*/`, tal y como se muestra en el siguiente ejemplo:

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

Selector Universal

- Se utiliza para seleccionar todos los elementos de la página. El siguiente ejemplo elimina el margen y el relleno de todos los elementos HTML (por ahora no es importante fijarse en la parte de la declaración de la regla CSS):

```
* {  
  margin: 0;  
  padding: 0;  
}
```

- El selector universal se indica mediante un asterisco (*). A pesar de su sencillez, no se utiliza habitualmente, ya que es difícil que un mismo estilo se pueda aplicar a todos los elementos de una página.

Selector de tipo o etiqueta

- Selecciona **todos** los elementos de la página cuya etiqueta HTML coincide con el valor del selector.
- Para utilizar este selector, solamente es necesario indicar el nombre de una etiqueta HTML (**sin los caracteres < y >**) correspondiente a los elementos que se quieren seleccionar.
- CSS permite agrupar todas las reglas individuales en una sola regla con un selector múltiple. Para ello, se incluyen todos los selectores separados por una coma (,)

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


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.
- 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 y todos los selectores anteriores indican el lugar en el que se debe encontrar ese elemento.
- El selector del siguiente ejemplo selecciona todos los elementos **** de la página que se encuentren dentro de un elemento **<p>**:

Selector

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

HTML:

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



Combinación de selectores

- No debe confundirse el selector descendente con la combinación de selectores:

```
/* El estilo se aplica a todos los elementos "p", "a", "span" y "em" */  
p, a, span, em { text-decoration: underline; }
```

```
/* El estilo se aplica solo a los elementos "em" que se  
encuentran dentro de "p a span" */  
p a span em { text-decoration: underline; }
```

Selector de clase (I)

Cómo se pueden aplicar estilos CSS sólo al primer párrafo? El selector universal (*) no se puede utilizar porque selecciona todos los elementos de la página. El selector de tipo o etiqueta (**p**) tampoco se puede utilizar porque seleccionaría todos los párrafos. Por último, el selector descendente (**body p**) tampoco se puede utilizar porque todos los párrafos se encuentran en el mismo sitio.

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

Selector de clase (II)

Una de las soluciones más sencillas para aplicar estilos a un solo elemento de la página consiste en utilizar el atributo **class** de HTML sobre ese elemento para indicar directamente la regla CSS que se le debe aplicar:

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

Selector de clase (III)

- A continuación, se crea en el archivo CSS una nueva regla llamada destacado con todos los estilos que se van a aplicar al elemento. Para que el navegador no confunda este selector con los otros tipos de selectores, se prefija el valor del atributo class con un punto (.) tal y como muestra el siguiente ejemplo:

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

- El selector **.destacado** se interpreta como *"cualquier elemento de la página cuyo atributo **class** sea igual a destacado"*, por lo que solamente el primer párrafo cumple esa condición.

Selector de clase (IV)

- Este tipo de selectores se llaman selectores de clase y son los más utilizados junto con los selectores de **ID**. La principal característica de este selector es que en una misma página HTML varios elementos diferentes pueden utilizar el mismo valor en el atributo **class**:

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


Restringir selector de clase

- En ocasiones, es necesario restringir el alcance del selector de clase.

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

- ¿Cómo es posible aplicar estilos **solamente al párrafo cuyo atributo class sea igual a destacado**? Combinando el selector de tipo y el selector de clase, se obtiene un selector mucho más específico:

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

- El selector **p.destacado** se interpreta como *"aquellos elementos de tipo **<p>** que dispongan de un atributo **class** con valor **destacado**".*

Ejemplos: selectores de clase

```
/* 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 { ... }
```

Estilos de clase sobre un mismo elemento

- También, 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>
```

- Al párrafo anterior se le aplican los estilos definidos en las reglas .especial, .destacado y .error, por lo que en el siguiente ejemplo, el texto del párrafo se vería de color rojo, en negrita y con un tamaño de letra de 15 píxel:

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

Selectores de ID (I)

- En ocasiones, es necesario aplicar estilos CSS a un único elemento de la página. Aunque puede utilizarse un selector de clase para aplicar estilos a un único elemento, existe otro selector más eficiente en este caso.
- El **selector de ID** permite seleccionar un elemento de la página a través del valor de su atributo id. Este tipo de selectores sólo seleccionan un elemento de la página porque el valor del atributo id no se puede repetir en dos elementos diferentes de una misma página.

Selectores de ID (II)

- 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:

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

```
<p>Primer párrafo</p>
```

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

```
<p>Tercer párrafo</p>
```

Selectores de ID (III)

- El valor del atributo **id** debe ser único, de forma que dos elementos diferentes no pueden tener el mismo valor de id.
- El atributo **class** no es obligatorio que sea único, de forma que muchos elementos HTML diferentes pueden compartir el mismo valor para su atributo class.

Restringir selector de ID

- Se puede restringir el alcance del selector mediante la combinación con otros selectores. El siguiente ejemplo aplica la regla CSS solamente al elemento de tipo **<p>** que tenga un atributo id igual al indicado:

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

- En realidad, un selector de tipo **p#aviso** sólo tiene sentido cuando el archivo CSS se aplica sobre muchas páginas HTML diferentes.
- En este caso, algunas páginas pueden disponer de elementos con un atributo id igual a aviso y que no sean párrafos, por lo que la regla anterior no se aplica sobre esos elementos.

Ejemplos: Selector de ID

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

```
/* Todos los elementos con atributo id="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 id="aviso" de la página */  
p, #aviso { ... }
```

Combinación de selectores

- CSS permite la combinación de uno o más tipos de selectores para restringir el alcance de las reglas CSS. A continuación se muestran algunos ejemplos habituales de combinación de selectores.
 - `.aviso .especial { ... }`
 - `div.aviso span.especial { ... }`
 - `ul#menuPrincipal li.destacado a#inicio { ... }`

Herencia en CSS (I)

- Una de las características principales de CSS es la herencia de los estilos definidos para los elementos. Cuando se establece el valor de una propiedad CSS en un elemento, sus elementos descendientes heredan de forma automática el valor de esa propiedad.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo de herencia de estilos</title>
<style type="text/css">
    body { color: blue; }
</style>
</head>

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

Herencia en CSS (II)

- Aunque la herencia de estilos se aplica automáticamente, se puede **anular** su efecto estableciendo de forma explícita otro valor para la propiedad que se hereda.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Ejemplo de herencia de estilos</title>
<style type="text/css">
    body { font-family: Arial; color: black; }
    h1 { font-family: Verdana; }
    p { color: red; }
</style>
</head>

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

Colisiones de estilos

- En las hojas de estilos complejas, es habitual que varias reglas CSS se apliquen a un mismo elemento HTML. El problema de estas reglas múltiples es que se pueden dar colisiones como la del siguiente ejemplo:
 - `p { color: red; }`
 - `p { color: blue; }`
- ¿De qué color se muestra el párrafo anterior?
- El método seguido por CSS para resolver las colisiones de estilos se muestra a continuación:
 1. Determinar todas las declaraciones que se aplican al elemento para el medio CSS seleccionado.
 2. Ordenar las declaraciones según su origen (CSS de navegador, de usuario o de diseñador) y su prioridad (palabra clave **!important**).
 3. Ordenar las declaraciones según lo específico que sea el selector. Cuanto más genérico es un selector, menos importancia tienen sus declaraciones.
 4. Si después de aplicar las normas anteriores existen dos o más reglas con la misma prioridad, se aplica la que se indicó en último lugar.

Unidades de medida

- 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. Todas las medidas se indican como un **valor numérico entero o decimal** seguido de una unidad de medida (sin ningún espacio en blanco entre el número y la unidad de medida).

Unidades absolutas

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

Ejemplo

```
/* 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

- La unidades relativas, a diferencia de las absolutas, no están completamente definidas, ya que su valor siempre está referenciado respecto a otro valor. A pesar de su aparente dificultad, son las más utilizadas en el diseño web por la flexibilidad con la que se adaptan a los diferentes medios.
 - **em**, (no confundir con la etiqueta `` de HTML) relativa respecto del tamaño de letra del elemento.
 - **ex**, relativa respecto de la altura de la letra x ("*equis minúscula*") del tipo y tamaño de letra del elemento.
 - **px**, (píxel) relativa respecto de la resolución de la pantalla del dispositivo en el que se visualiza la página HTML.

Porcentajes

- El porcentaje también es una unidad de medida relativa, aunque por su importancia CSS la trata de forma separada a **em**, **ex** y **px**. Un porcentaje está formado por un valor numérico seguido del símbolo **%** y siempre está referenciado a otra medida. Cada una de las propiedades de CSS que permiten indicar como valor un porcentaje, define el valor al que hace referencia ese porcentaje.
 - `body { font-size: 1em; }`
 - `h1 { font-size: 200%; }`
 - `h2 { font-size: 150%; }`
- Los tamaños establecidos para los elementos **<h1>** y **<h2>** mediante las reglas anteriores, son equivalentes a 2em y 1.5em respectivamente, por lo que es más habitual definirlos mediante **em**.

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 habitual es el del RGB hexadecimal,

Color: Palabras claves

- CSS define 17 palabras clave para referirse a los colores básicos. Las palabras se corresponden con el nombre en inglés de cada color:
- aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, orange, purple, red, silver, teal, white, yellow
- Los navegadores modernos soportan muchos otros nombres de colores.
[en.wikipedia.org/wiki/Websafe.](https://en.wikipedia.org/wiki/Web_safe_colors)

maroon #800000	red #ff0000	orange #ffa500	yellow #ffff00	olive #808000
purple #800080	fuchsia #ff00ff	white #ffffff	lime #00ff00	green #008000
navy #000080	blue #0000ff	aqua #00ffff	teal #008080	
black #000000	silver #c0c0c0	gray #808080		

Color: RGB decimal

- El modelo RGB consiste en definir un color indicando la cantidad de color **rojo, verde y azul** que se debe *mezclar* para obtener ese color.
 - El modelo RGB es un modelo de tipo "aditivo", ya que los colores se obtienen sumando sus componentes.
 - Cada una de las componentes puede tomar un valor entre cero y un valor máximo (255).
 - Si todas las componentes valen 0, el color creado es el negro y si todas las componentes toman su valor máximo, el color obtenido es el blanco.
 - Si se indica un valor menor que 0 para una componente, automáticamente se transforma su valor en 0. Igualmente, si se indica un valor mayor que 255, se transforma automáticamente su valor a 255.
- **Ejemplo:**
 - p { color: rgb(71, 98, 176); }

Color: RGB porcentual

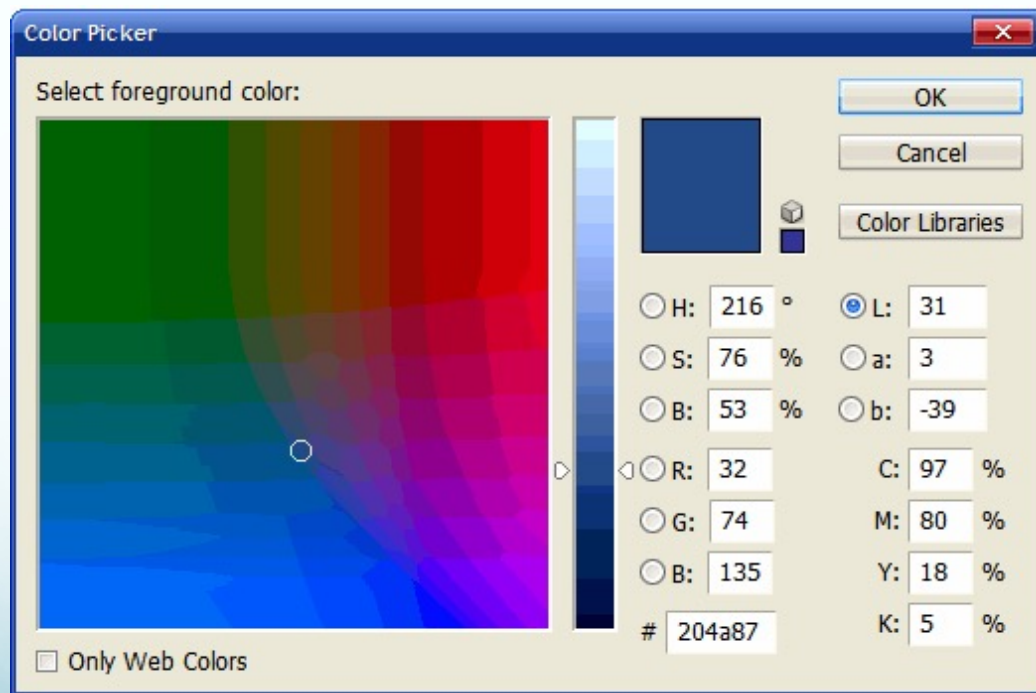
- Las componentes RGB de un color también se pueden indicar mediante un porcentaje.
- El funcionamiento y la sintaxis de este método es el mismo que el del RGB decimal.
- La única diferencia es que en este caso el valor de las componentes RGB puede tomar valores entre 0% y 100%.
- El mismo color del ejemplo anterior se puede representar de forma porcentual:
 - `p { color: rgb(27%, 38%, 69%); }`

Color: RGB hexadecimal

- Es el método más complicado para indicar los colores, pero, se trata del método más utilizado.
 - La mayoría de sitios web reales utilizan exclusivamente este método.
- El sistema hexadecimal utiliza 16 símbolos para representar sus números.
 - Como sólo conocemos 10 símbolos numéricos, el sistema hexadecimal utiliza también seis letras (de la A a la F) para representar los números.
- Definir un color en CSS con el método RGB hexadecimal requiere realizar los siguientes pasos:
 1. Determinar las componentes RGB decimales del color original, por ejemplo: R = 71, G = 98, B = 176.
 2. Transformar el valor decimal de cada componente al sistema numérico hexadecimal.
 3. Se concatenan los valores hexadecimales de las componentes RGB en ese orden y se les añade el prefijo #.
- Siguiendo el mismo ejemplo:
 - `p { color: #4762B0; }`

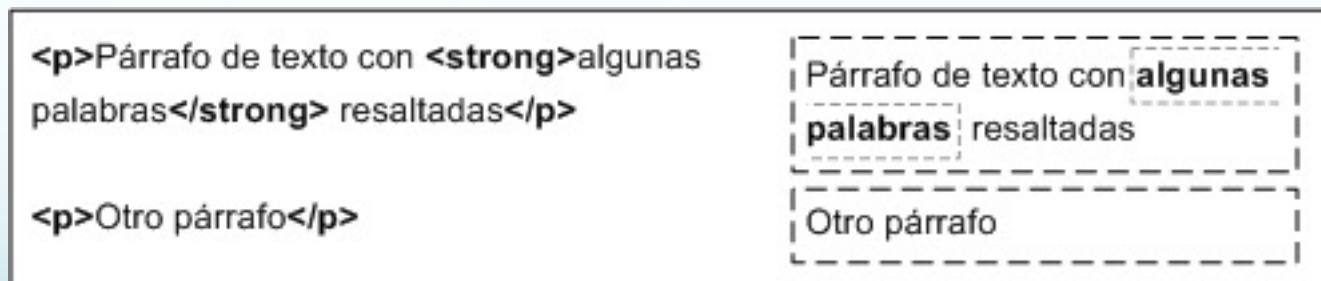
Color: RGB hexadecimal

- La mayoría de programas de diseño gráfico convierten de forma automática los valores RGB decimales a sus valores RGB hexadecimales.



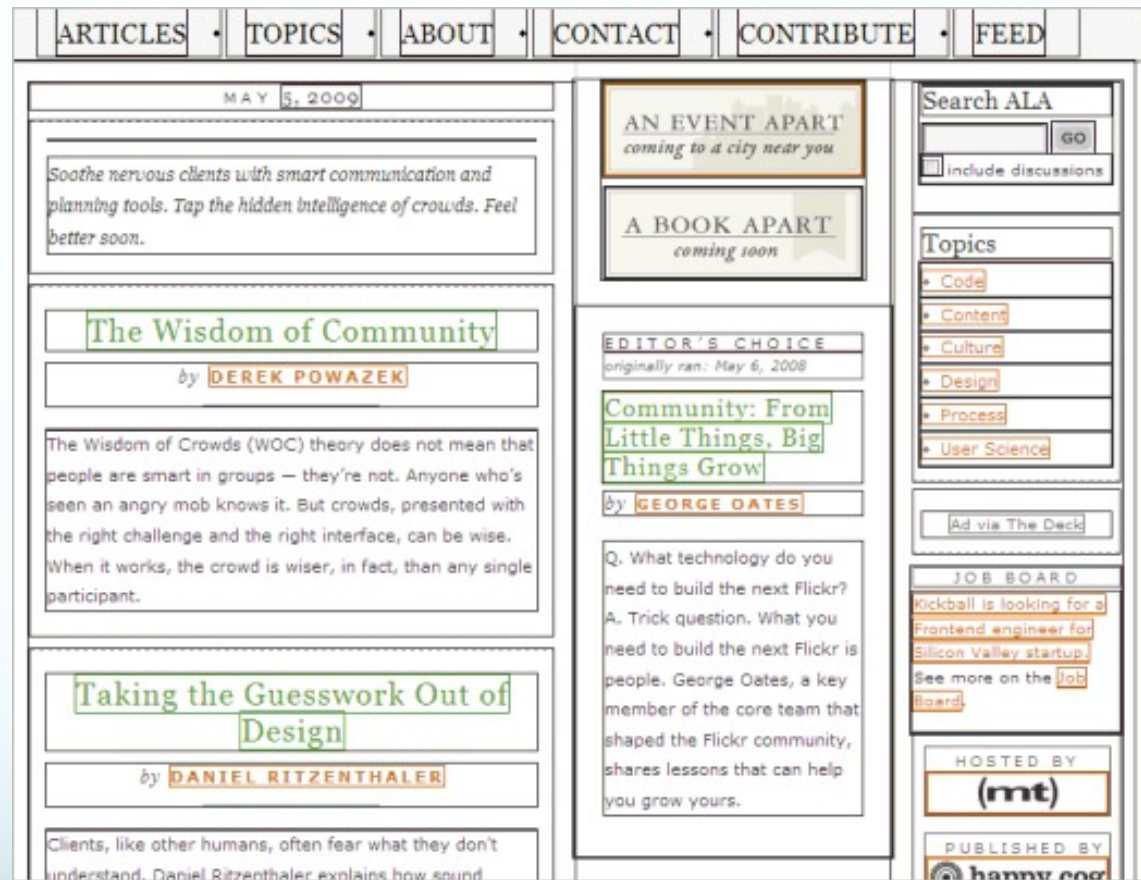
Modelo de cajas (I)

- 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.
- Las cajas de una página se crean automáticamente. Cada vez que se inserta una etiqueta HTML, se crea una nueva caja rectangular que encierra los contenidos de ese elemento.



Modelo de cajas (II)

- Las cajas de las páginas no son visibles a simple vista porque inicialmente no muestran ningún color de fondo ni ningún borde.
- Los navegadores crean y colocan las cajas de forma automática, pero CSS permite modificar todas sus características.

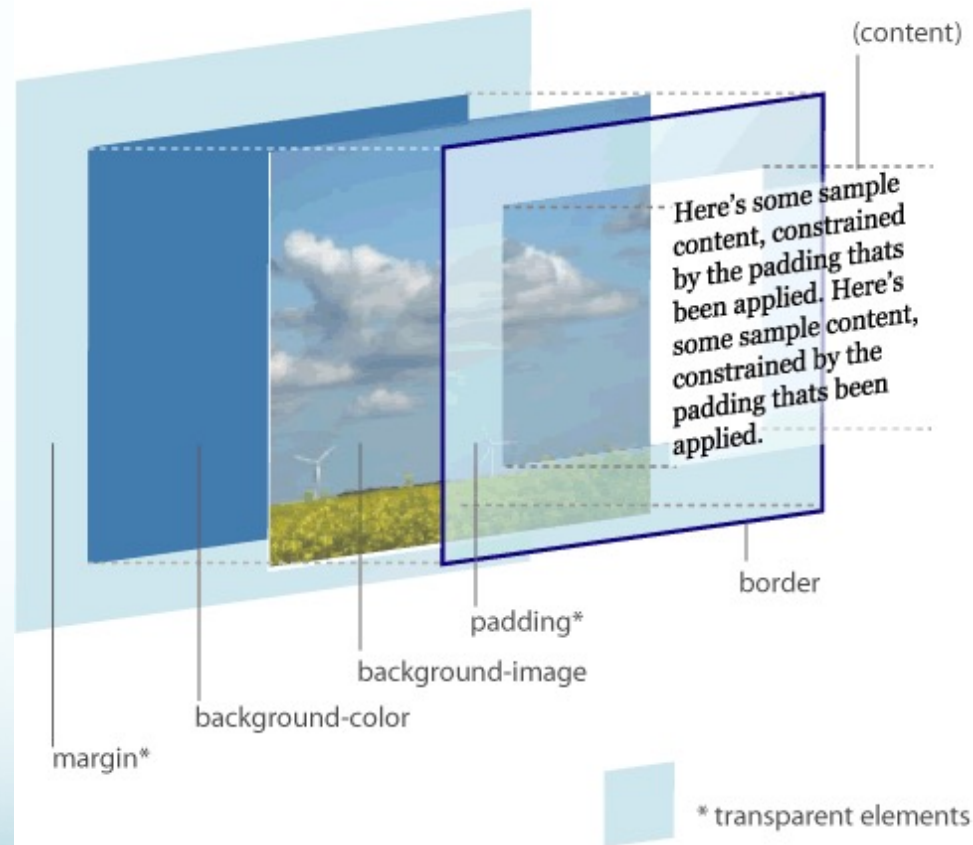


Modelo de cajas (III)

- Las partes que componen cada caja y su orden de visualización desde el punto de vista del usuario son las siguientes:
 - **Contenido** (*content*): se trata del 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.

Modelo de cajas (IV)

THE CSS BOX MODEL HIERARCHY



Anchura

- La propiedad CSS que controla la anchura de la caja de los elementos se denomina **width**.
 - La propiedad **width** no admite valores negativos y los valores en porcentaje se calculan a partir de la anchura de su elemento padre.
 - El valor **inherit** indica que la anchura del elemento se hereda de su elemento padre.
 - El valor **auto**, que es el que se utiliza si no se establece de forma explícita un valor a esta propiedad.
 - Se aplica a todos los elementos, salvo los elementos en línea que no sean imágenes, las filas de tabla y los grupos de filas de tabla
- **Ejemplo:**

```
#lateral { width: 200px; }  
<div id="lateral">  
...  
</div>
```

Altura

- La propiedad CSS que controla la altura de los elementos se denomina **height**.
 - Al igual que sucede con **width**, la propiedad **height** no admite valores negativos. Si se indica un porcentaje, se toma como referencia la altura del elemento padre...
- **Ejemplo:**

```
#cabecera { height: 60px; }
```

```
<div id="cabecera">
```

```
...
```

```
</div>
```

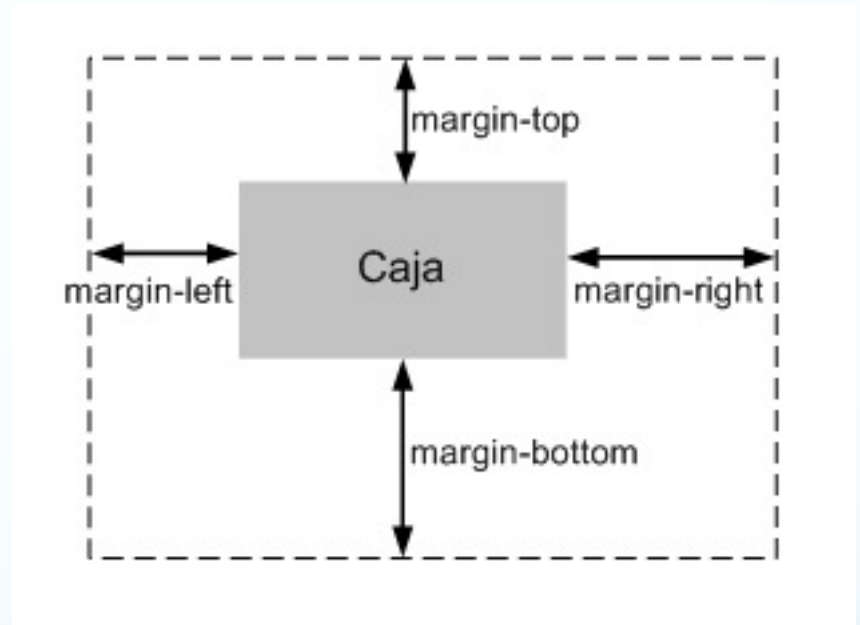
Margen

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

Propiedades	margin-top, margin-right, margin-bottom, margin-left
Valores	unidad de medida porcentaje auto inherit
Se aplica a	Todos los elementos, salvo margin-top y margin-bottom que sólo se aplican a los elementos de bloque y a las imágenes
Valor inicial	0
Descripción	Establece cada uno de los márgenes horizontales y verticales de un elemento

Propiedades de margen

- Cada una de las propiedades 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).



Margen: propiedad shorthand

- 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. Estas propiedades especiales se denominan "*propiedades shorthand*".
 - La propiedad que permite definir de forma simultánea los cuatro márgenes se denomina **margin**.

Propiedad	margin
Valores	(unidad de medida porcentaje auto) {1, 4} inherit
Se aplica a	Todos los elementos salvo algunos casos especiales de elementos mostrados como tablas
Valor inicial	-
Descripción	Establece de forma directa todos los márgenes de un elemento

Ejemplo: Margen

- Código CSS original:

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

- Alternativa directa:

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

- Otra alternativa:

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

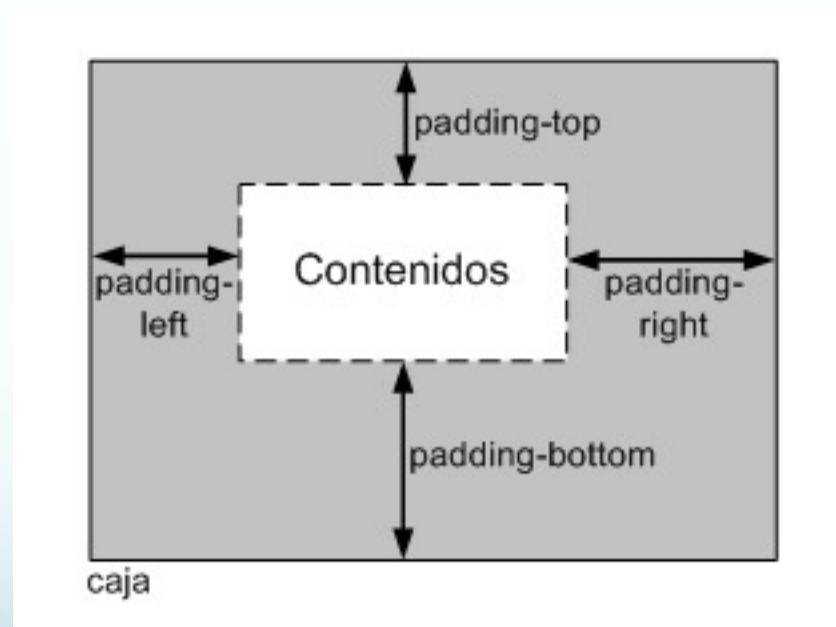
Relleno

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

Propiedades	padding-top, padding-right, padding-bottom, padding-left
Valores	unidad de medida porcentaje inherit
Se aplica a	Todos los elementos excepto algunos elementos de tablas como grupos de cabeceras y grupos de pies de tabla
Valor inicial	0
Descripción	Establece cada uno de los rellenos horizontales y verticales de un elemento

Propiedades de relleno

- Cada una de estas propiedades establece la separación entre el contenido y los bordes laterales de la caja del elemento



Relleno: propiedad shorthand

- Como sucede con los márgenes, CSS también define una propiedad de tipo “*shorthand*” llamada **padding** para establecer los cuatro rellenos de un elemento de forma simultánea.
- La notación {1, 4} de la definición anterior significa que la propiedad **padding** admite entre uno y cuatro valores, con el mismo significado que el de la propiedad **margin**.

```
body {padding: 2em} /* Todos los rellenos valen 2em */  
body {padding: 1em 2em} /* Superior e inferior = 1em, Izquierdo y derecho = 2em */  
body {padding: 1em 2em 3em} /* Superior = 1em, derecho = 2em, inferior = 3em, izquierdo = 2em */  
body {padding: 1em 2em 3em 4em} /* Superior = 1em, derecho = 2em, inferior = 3em, izquierdo = 4em */
```


Borde

- 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.

Borde: anchura

- La anchura de los bordes se indica mediante una medida (en cualquier unidad de medida absoluta o relativa)
- Si se quiere establecer de forma simultánea la anchura de todos los bordes de una caja, es necesario utilizar una propiedad "*shorthand*" llamada **border-width**

Propiedades	border-top-width, border-right-width, border-bottom-width, border-left-width
Valores	(unidad de medida thin medium thick) inherit
Se aplica a	Todos los elementos
Valor inicial	Medium
Descripción	Establece la anchura de cada uno de los cuatro bordes de los elementos

Ejemplo: Borde anchura

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

```
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 */
```

Borde: Color

Propiedades	border-top-color, border-right-color, border-bottom-color, border-left-color
Valores	color transparent inherit
Se aplica a	Todos los elementos
Valor inicial	-
Descripción	Establece el color de cada uno de los cuatro bordes de los elementos

Propiedad	border-color
Valores	(color transparent) {1, 4} inherit
Se aplica a	Todos los elementos
Valor inicial	-
Descripción	Establece el color de todos los bordes del elemento

Ejemplo: Borde color

```
div {  
  border-top-color: #CC0000;  
  border-right-color: blue;  
  border-bottom-color: #00FF00;  
  border-left-color: #CCC;  
}
```

Borde: estilo

Propiedades	border-top-style, border-right-style, border-bottom-style, border-left-style
Valores	none hidden dotted dashed solid double groove ridge inset outset inherit
Se aplica a	Todos los elementos
Valor inicial	none
Descripción	Establece el estilo de cada uno de los cuatro bordes de los elementos

Propiedad	border-style
Valores	(none hidden dotted dashed solid double groove ridge inset outset) {1, 4} inherit
Se aplica a	Todos los elementos
Valor inicial	-
Descripción	Establece el estilo de todos los bordes del elemento

Ejemplo: Borde estilo

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

Borde: propiedad shorthand

Propiedades	border-top, border-right, border-bottom, border-left
Valores	(unidad de medida_borde color_borde estilo_borde) inherit
Se aplica a	Todos los elementos
Valor inicial	-
Descripción	Establece el estilo completo de cada uno de los cuatro bordes de los elementos

Propiedad	border
Valores	(unidad de medida_borde color_borde estilo_borde) inherit
Se aplica a	Todos los elementos
Valor inicial	-
Descripción	Establece el estilo completo de todos los bordes de los elementos

Ejemplo: Borde shorthand

```
h1 {  
  border-bottom: solid red;  
}  
  
div {  
  border-top: 1px solid #369;  
  border-bottom: 3px double #369;  
}  
  
div {  
  border-top: 1px solid red;  
  border-right: 1px solid red;  
  border-bottom: 1px solid red;  
  border-left: 1px solid red;  
}  
  
div {  
  border: 1px solid red;  
}
```

Fondo

- CSS define cinco propiedades para establecer el fondo de cada elemento (**background-color**, **background-image**, **background-repeat**, **background-attachment**, **background-position**) y otra propiedad de tipo "*shorthand*" (**background**).

Fondo: color e imagen

Propiedad	background-color
Valores	color transparent inherit
Se aplica a	Todos los elementos
Valor inicial	transparent
Descripción	Establece un color de fondo para los elementos

Propiedad	background-image
Valores	url none inherit
Se aplica a	Todos los elementos
Valor inicial	none
Descripción	Establece una imagen como fondo para los elementos

Ejemplo: Fondo color e imagen

```
body {  
  background-color: #F5F5F5;  
}
```

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


Fondo: repetir y posición

Propiedad	background-repeat
Valores	repeat repeat-x repeat-y no-repeat inherit
Se aplica a	Todos los elementos
Valor inicial	repeat
Descripción	Controla la forma en la que se repiten las imágenes de fondo

Propiedad	background-position
Valores	((porcentaje unidad de medida left center right) (porcentaje unidad de medida top center bottom)?) ((left center right) (top center bottom)) inherit
Se aplica a	Todos los elementos
Valor inicial	0% 0%
Descripción	Controla la posición en la que se muestra la imagen en el fondo del elemento

Ejemplo: Fondo repetición

```
#hdr {  
    background: url("/images/ds.gif") repeat-x;  
    width: 100%;  
    text-align: center;  
}  
  
.wide #content-secondary {  
    width: 272px;  
    margin: 13px 0 0 0;  
    position: relative;  
    margin-left: -8px;  
    background: url("../graphics/wide/bg-content-  
secondary.gif")  
    repeat-y;  
}
```

Ejemplo: fondo posición

```
#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>
```

Fondo: fijación y shorthand

Propiedad	background-attachment
Valores	scroll fixed inherit
Se aplica a	Todos los elementos
Valor inicial	scroll
Descripción	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

Propiedad	background
Valores	(background-color background-image background-repeat background-attachment background-position) inherit
Se aplica a	Todos los elementos
Valor inicial	-
Descripción	Establece todas las propiedades del fondo de un elemento

Ejemplo: fondo

/* 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;  
}
```

Tipos de elementos (I)

- El estándar HTML clasifica a todos sus elementos en dos grandes grupos: elementos en línea y elementos de bloque.
 - Los elementos de bloque ("*block elements*" en inglés) siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea.
 - Los elementos en línea ("*inline elements*" en inglés) 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. En cambio, un elemento en línea puede aparecer tanto dentro de un elemento de bloque como dentro de otro elemento en línea.

Tipos de elementos (II)

- 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.

Posicionamiento

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

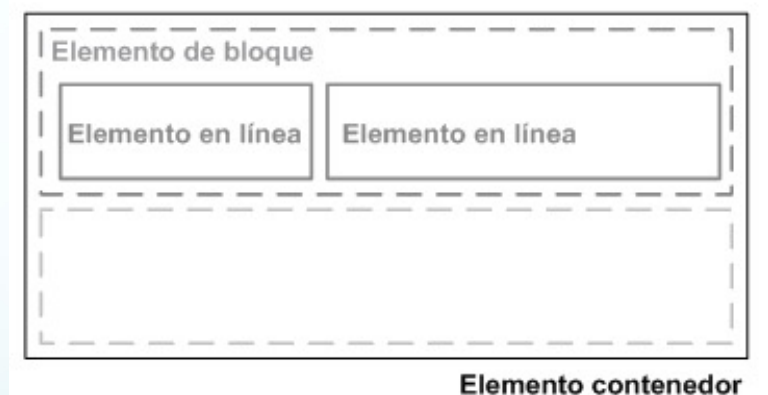
Posicionamiento

Propiedad	position
Valores	static relative absolute fixed inherit
Se aplica a	Todos los elementos
Valor inicial	static
Descripción	Selecciona el posicionamiento con el que se mostrará el elemento

Propiedades	top, right, bottom, left
Valores	unidad de medida porcentaje auto inherit
Se aplica a	Todos los elementos posicionados
Valor inicial	auto
Descripción	Indican el desplazamiento horizontal y vertical del elemento respecto de su posición original

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.



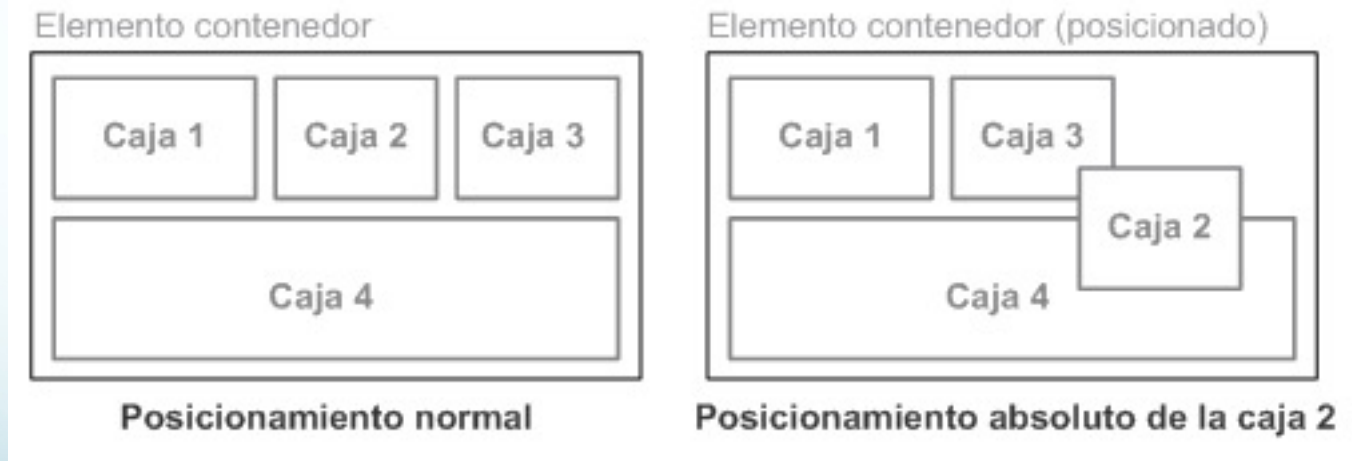
Posicionamiento relativo

- El posicionamiento relativo desplaza una caja respecto de su posición original establecida mediante el posicionamiento normal. El desplazamiento de la caja se controla con las propiedades **top**, **right**, **bottom** y **left**.



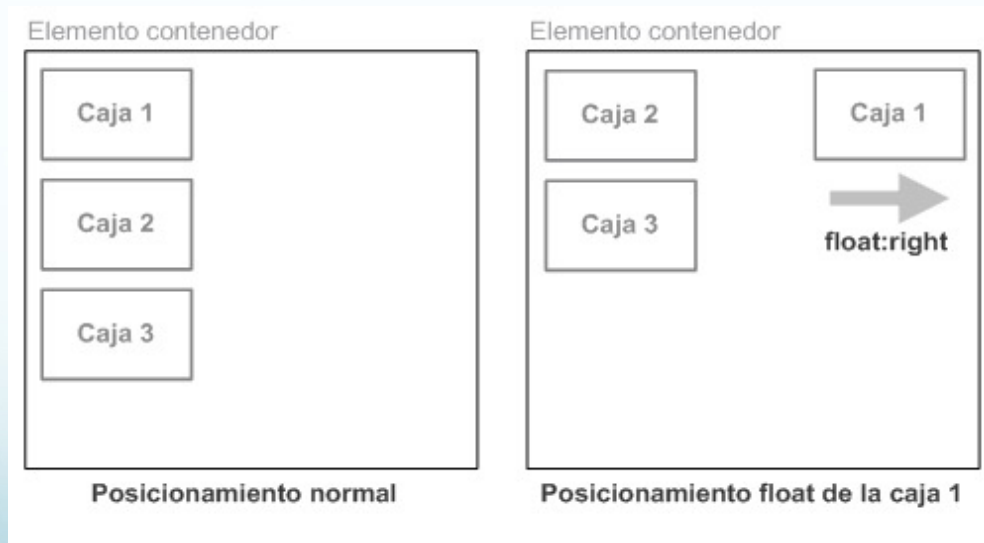
Posicionamiento absoluta

- El posicionamiento absoluto se emplea para establecer de forma exacta la posición en la que se muestra la caja de un elemento. La nueva posición de la caja se indica mediante las propiedades **top**, **right**, **bottom** y **left**.
- La nueva posición depende del elemento contenedor.



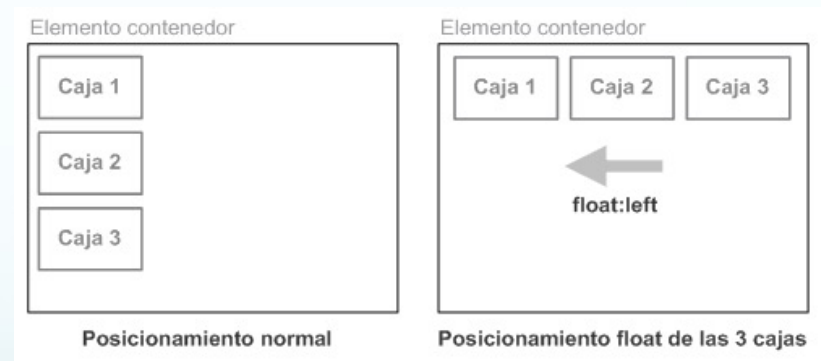
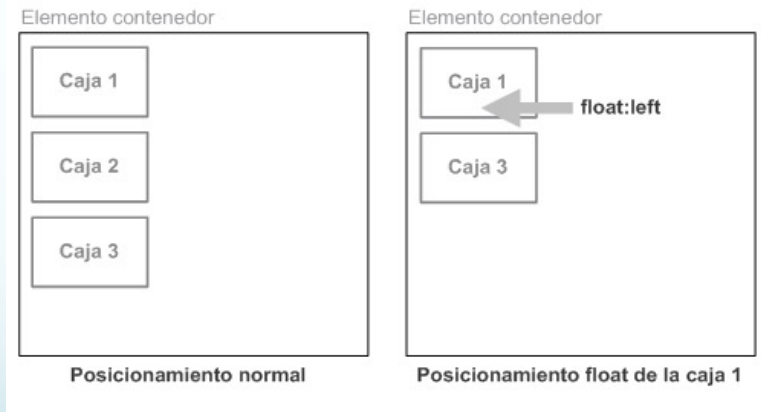
Posicionamiento flotante (I)

- 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.



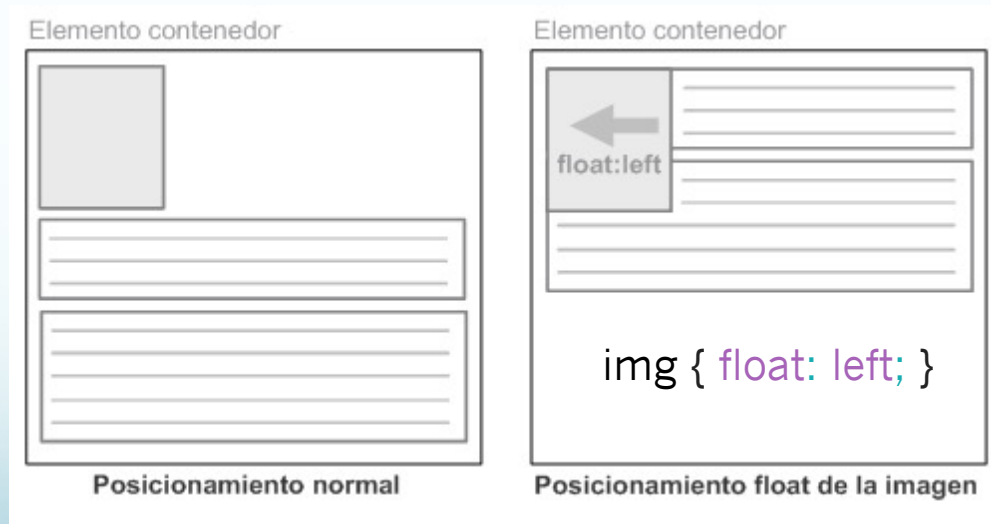
Posicionamiento flotante (II)

- 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.**



Propiedad flotante

Propiedad	float
Valores	left right none inherit
Se aplica a	Todos los elementos
Valor inicial	none
Descripción	Establece el tipo de posicionamiento flotante del elemento

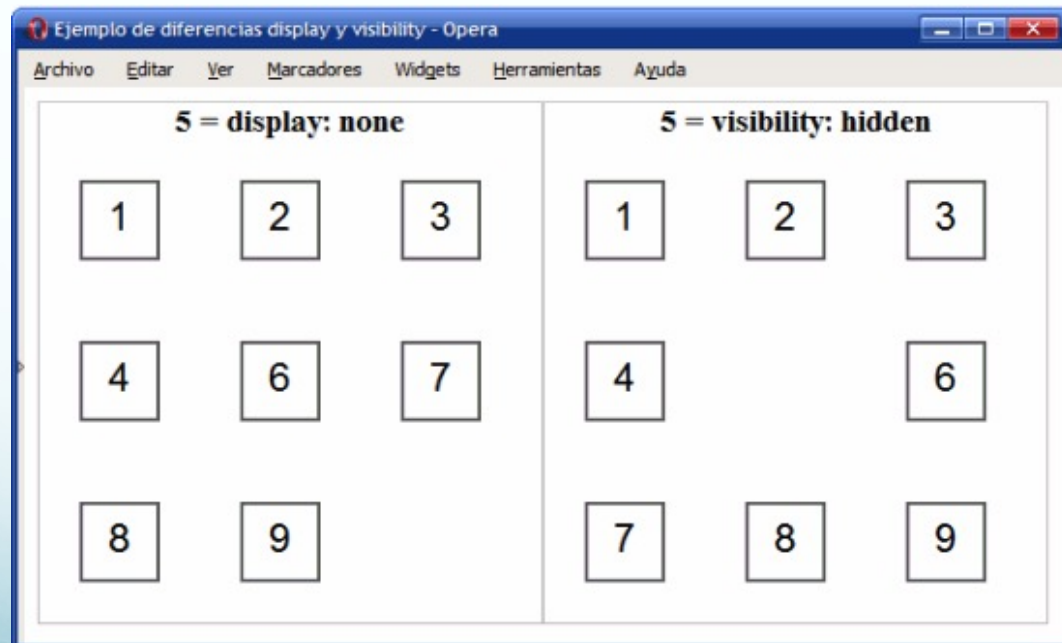


Visualización

- Además de las propiedades que controlan el posicionamiento de los elementos, CSS define otras cuatro propiedades para controlar su visualización: **display, visibility, overflow y z-index.**
- Utilizando algunas de estas propiedades es posible ocultar y/o hacer invisibles las cajas de los elementos, por lo que son imprescindibles para realizar efectos avanzados y animaciones.

Propiedad display y visibility

- La propiedad **display** permite ocultar completamente un elemento 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.
- La propiedad **visibility** permite hacer invisible un elemento, lo que significa que el navegador crea la caja del elemento pero no la muestra. En este caso, el resto de elementos de la página no modifican su posición.

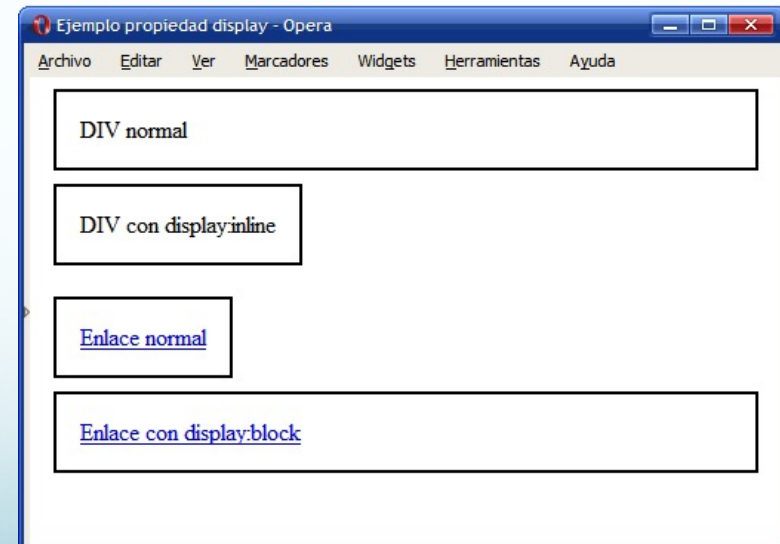


Propiedad display

Propiedad	display
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
Se aplica a	Todos los elementos
Valor inicial	inline
Descripción	Permite controlar la forma de visualizar un elemento e incluso ocultarlo

Los valores más utilizados son:

- El valor **block** muestra un elemento como si fuera un elemento de bloque.
- El valor **inline** visualiza un elemento en forma de elemento en línea.
- El valor **none** oculta un elemento y hace que desaparezca de la página.



Propiedad visibility

Propiedad	visibility
Valores	visible hidden collapse inherit
Se aplica a	Todos los elementos
Valor inicial	visible
Descripción	Permite hacer visibles e invisibles a los elementos

Propiedad overflow

- CSS define la propiedad **overflow** para controlar la forma en la que se visualizan los contenidos que sobresalen de sus elementos.

Propiedad	overflow
Valores	visible hidden scroll auto inherit
Se aplica a	Elementos de bloque y celdas de tablas
Valor inicial	visible
Descripción	Permite controlar los contenidos sobrantes de un elemento

Propiedad z-index

- CSS permite controlar la posición tridimensional de las cajas posicionadas. De esta forma, es posible indicar las cajas que se muestran delante o detrás de otras cajas cuando se producen solapamientos.
- El valor más común de la propiedad z-index es un número entero. Se considera el número 0 como el nivel más bajo.

Propiedad	z-index
Valores	auto numero inherit
Se aplica a	Elementos que han sido posicionados explícitamente
Valor inicial	auto
Descripción	Establece el nivel tridimensional en el que se muestra el elemento

Texto: Tipografía

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

Propiedad	color
Valores	color inherit
Se aplica a	Todos los elementos
Valor inicial	Depende del navegador
Descripción	Establece el color de letra utilizado para el texto

Color

El color por defecto del texto depende del navegador, todos los navegadores principales utilizan el color negro. Para establecer el color de letra de un texto, se puede utilizar cualquiera de las cinco formas que incluye CSS para definir un color.

```
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; }
```

La única excepción de este comportamiento son los enlaces que se crean con la etiqueta <a>

```
body, a { color: #777; }
```

Tipo de letra

Propiedad	font-family
Valores	((nombre_familia familia_generica) (,nombre_familia familia_generica)*) inherit
Se aplica a	Todos los elementos
Valor inicial	Depende del navegador
Descripción	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, como por ejemplo "**Arial**", "**Verdana**", "**Garamond**", etc.
- Mediante el nombre genérico de una *familia* tipográfica, como por ejemplo **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*).

Ejemplo: Tipo Letra

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;

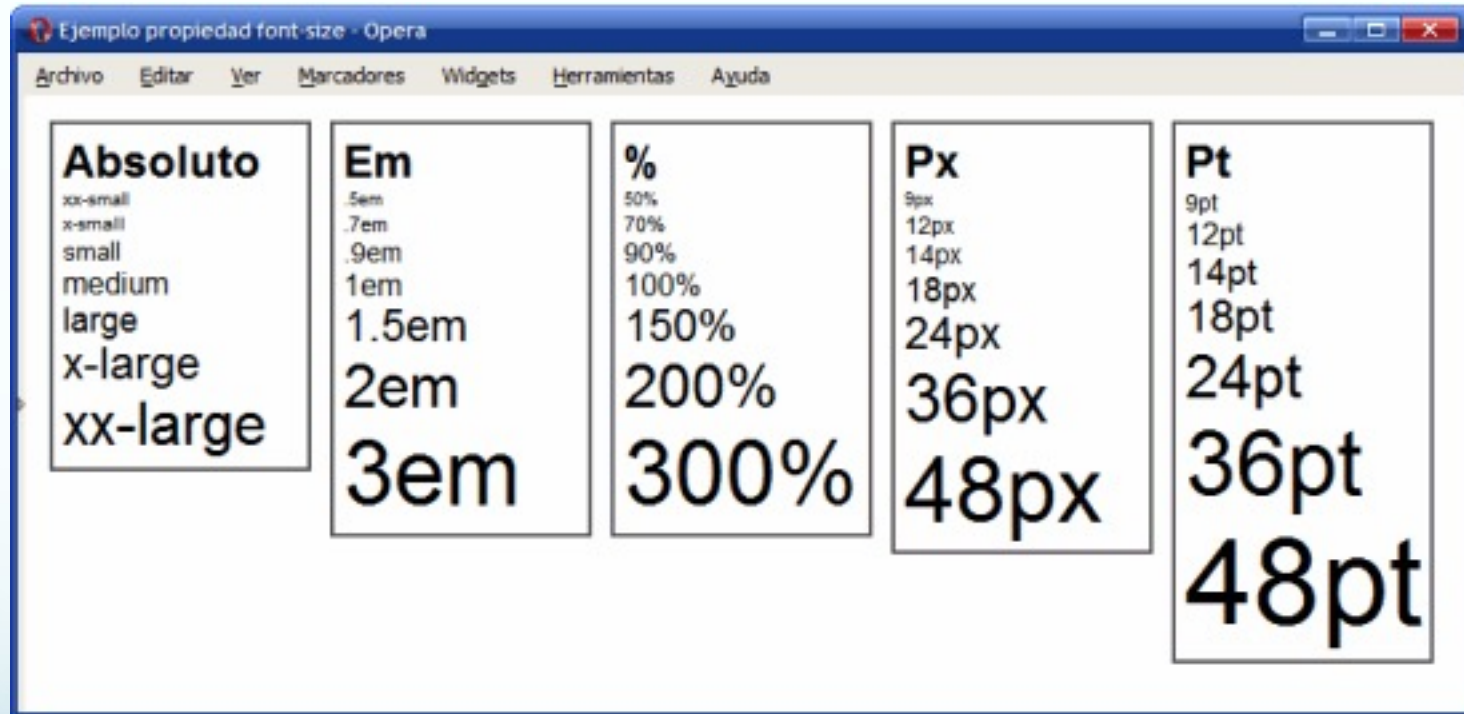
Tamaño de letra

Propiedad	font-size
Valores	tamaño_absoluto tamaño_relativo <u>unidad de medida</u> porcentaje inherit
Se aplica a	Todos los elementos
Valor inicial	medium
Descripción	Establece el tamaño de letra utilizado para el texto

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.

Ejemplo: Tamaño letra



Comparación visual de las distintas unidades para indicar el tamaño del texto

Anchura de la letra

Propiedad	font-weight
Valores	normal bold bolder lighter 100 200 300 400 500 600 700 800 900 inherit
Se aplica a	Todos los elementos
Valor inicial	normal
Descripción	Establece la anchura de la letra utilizada para el texto

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.

Por defecto, los navegadores muestran el texto de los elementos `` en cursiva y el texto de los elementos `` en negrita.

Estilo de letra

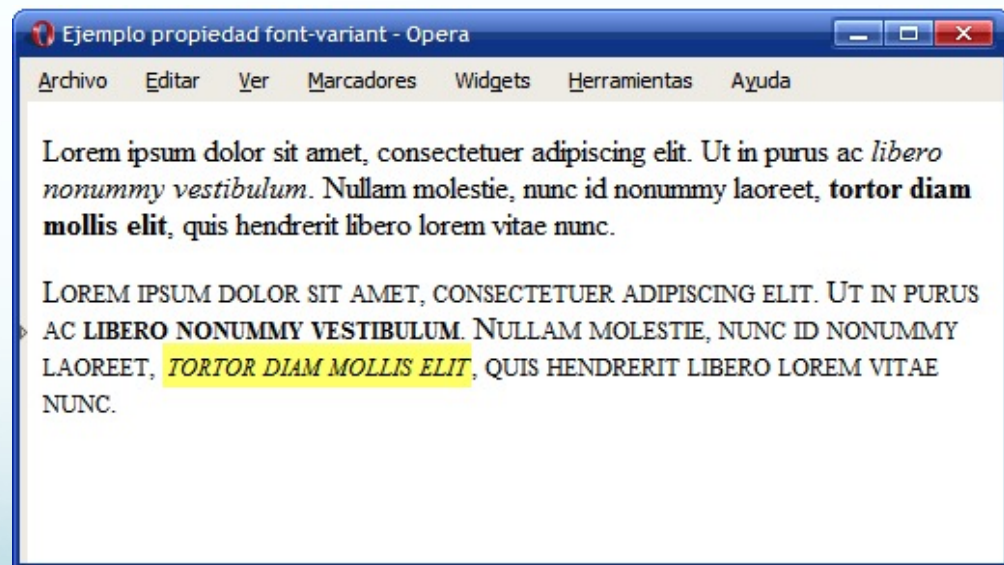
Propiedad	font-style
Valores	normal italic oblique inherit
Se aplica a	Todos los elementos
Valor inicial	normal
Descripción	Establece el estilo de la letra utilizada para el texto

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

Tipo de letra variante

Propiedad	font-variant
Valores	normal small-caps inherit
Se aplica a	Todos los elementos
Valor inicial	normal
Descripción	Establece el estilo alternativo de la letra utilizada para el texto

La propiedad **font-variant** no se suele emplear habitualmente, ya que sólo permite mostrar el texto con *letra versal* (mayúsculas pequeñas).



Fuente: shorthand

Propiedad	font
Valores	((font-style font-variant font-weight)? font-size (/ line-height)? font-family) caption icon menu message-box small-caption status-bar inherit
Se aplica a	Todos los elementos
Valor inicial	-
Descripción	Permite indicar de forma directa todas las propiedades de la tipografía de un texto

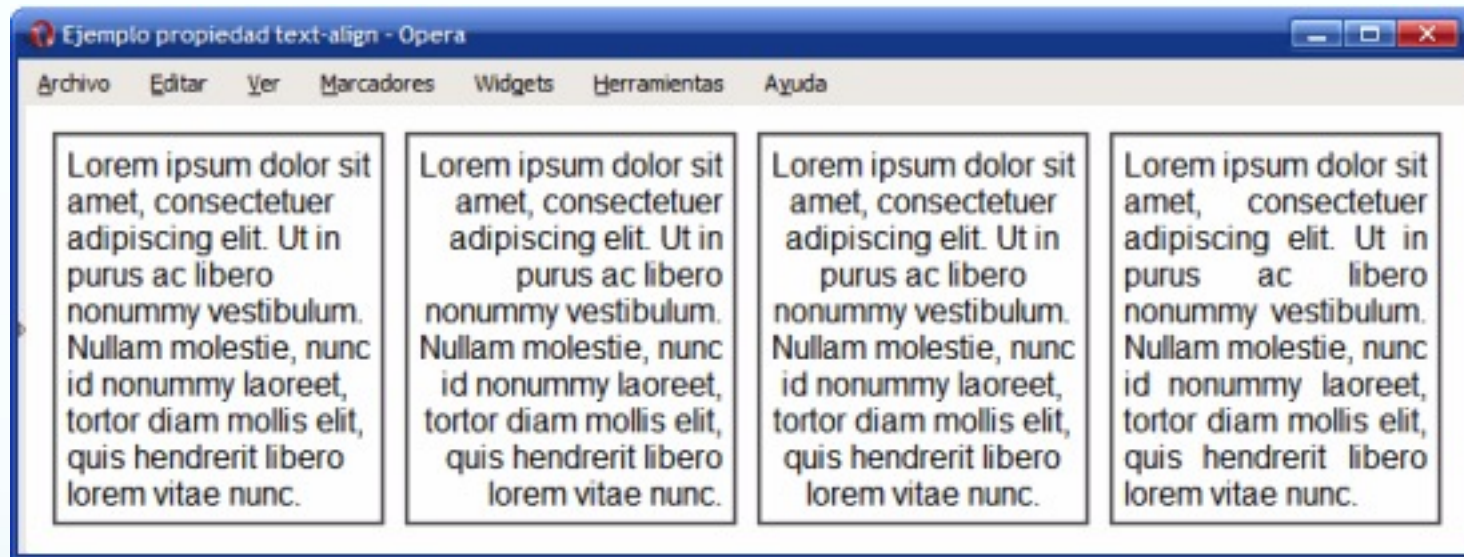
font: 76%/140% Verdana, Arial, Helvetica, **sans-serif**;
font: normal 24px/26px "Century Gothic", "Trebuchet MS", Arial, Helvetica, **sans-serif**;
font: normal .94em "Trebuchet MS", Arial, Helvetica, **sans-serif**;
font: bold 1em "Trebuchet MS", Arial, Sans-Serif;
font: normal 0.9em "Lucida Grande", 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**;

Otras propiedades de texto (I)

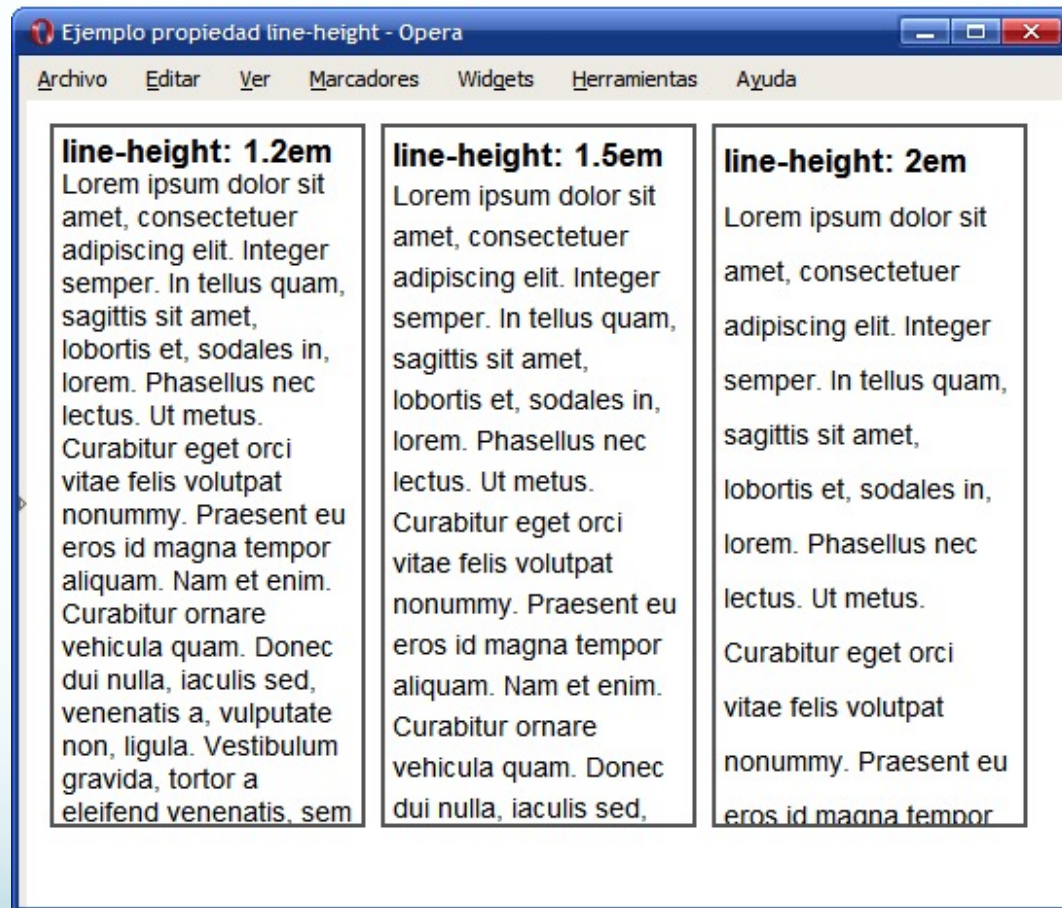
Propiedad	text-align
Valores	left right center justify inherit
Se aplica a	Elementos de bloque y celdas de tabla
Valor inicial	left
Descripción	Establece la alineación del contenido del elemento

Propiedad	line-height
Valores	normal numero unidad de medida porcentaje inherit
Se aplica a	Todos los elementos
Valor inicial	normal
Descripción	Permite establecer la altura de línea de los elementos

Ejemplo: text-align



Ejemplo: line-height

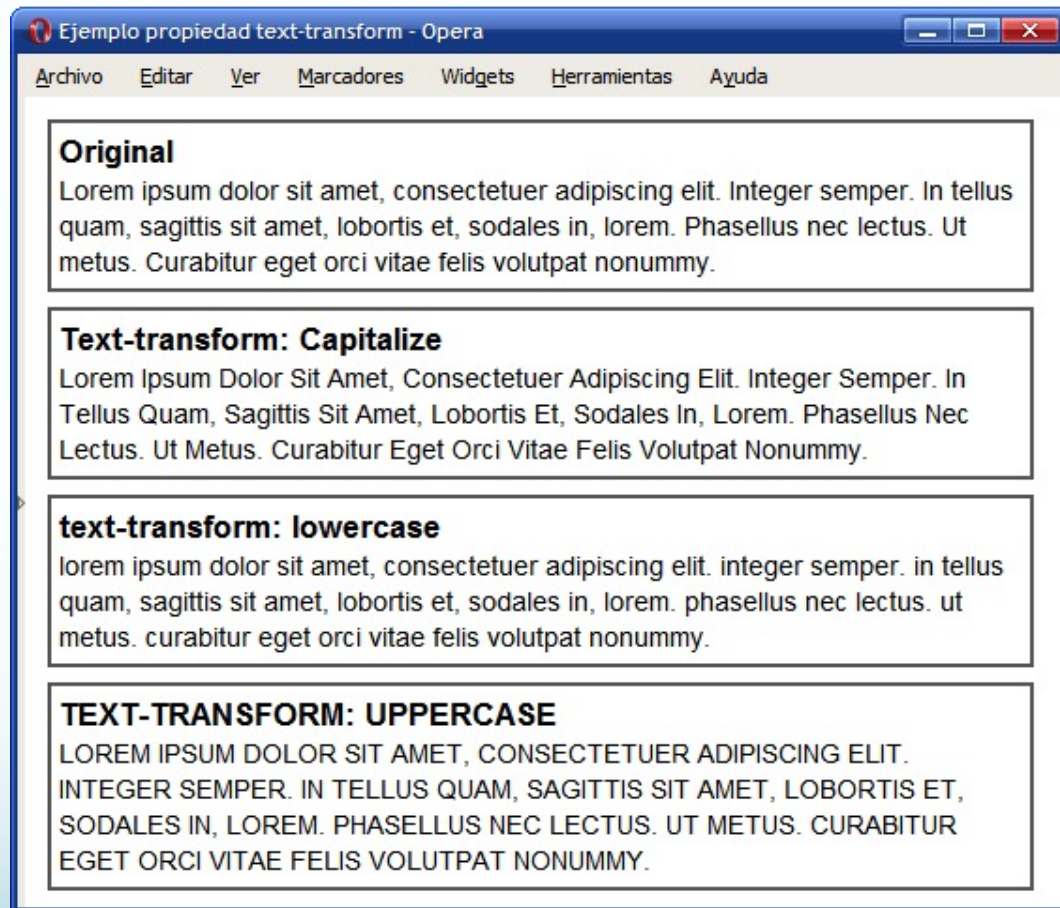


Otras propiedades de texto (II)

Propiedad	text-decoration
Valores	none (underline overline line-through blink) inherit
Se aplica a	Todos los elementos
Valor inicial	none
Descripción	Establece la decoración del texto (subrayado, tachado, parpadeante, etc.)

Propiedad	text-transform
Valores	capitalize uppercase lowercase none inherit
Se aplica a	Todos los elementos
Valor inicial	none
Descripción	Transforma el texto original (lo transforma a mayúsculas, a minúsculas, etc.)

Ejemplo: text-transform



Otras propiedades de texto (III)

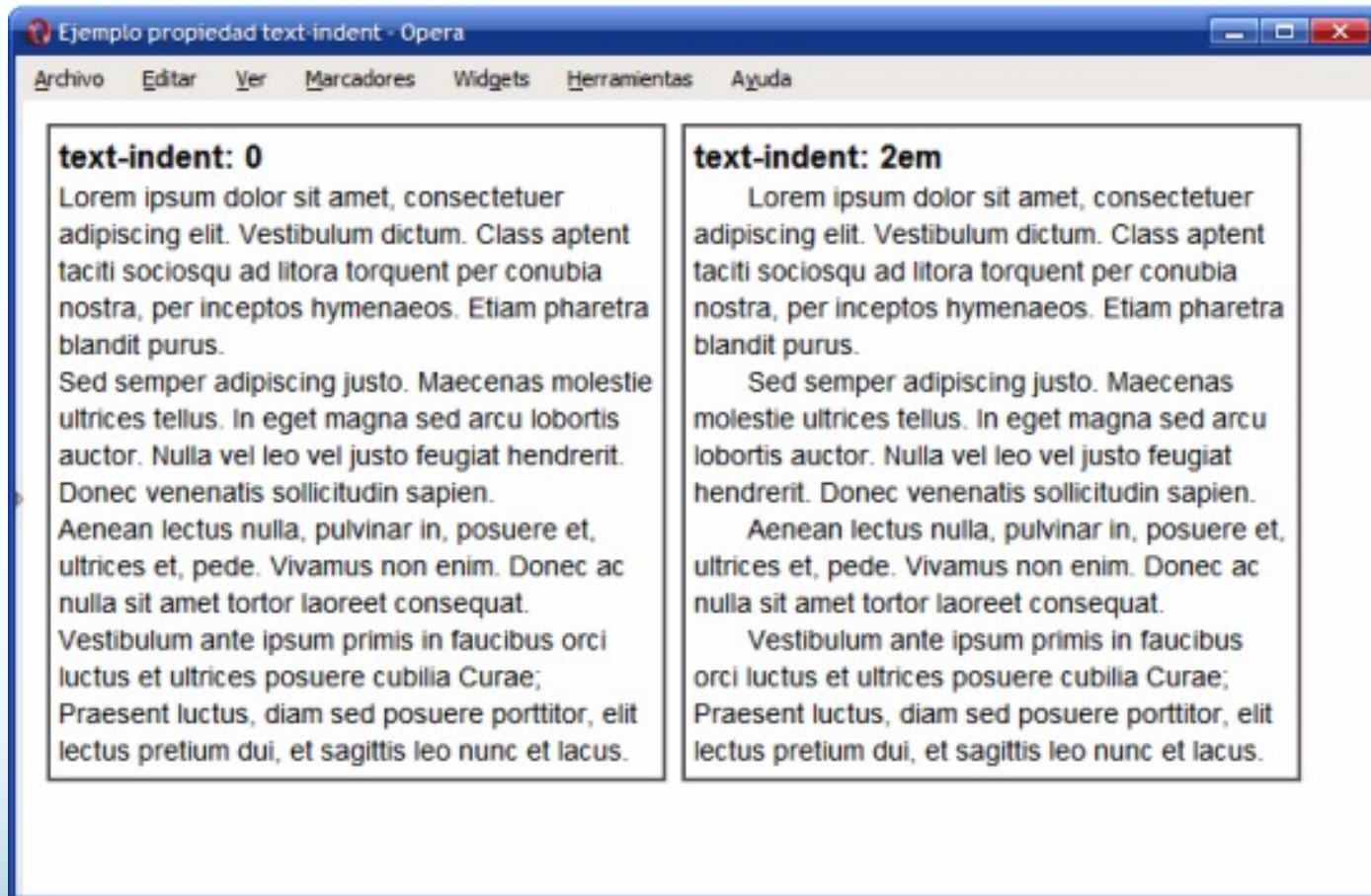
Propiedad	vertical-align
Valores	baseline sub super top text-top middle bottom text-bottom porcentaje unidad de medida inherit
Se aplica a	Elementos en línea y celdas de tabla
Valor inicial	baseline
Descripción	Determina la alineación vertical de los contenidos de un elemento

Propiedad	text-indent
Valores	unidad de medida porcentaje inherit
Se aplica a	Los elementos de bloque y las celdas de tabla
Valor inicial	0
Descripción	Tabula desde la izquierda la primera línea del texto original

Ejemplo: vertical-align



Ejemplo: text-indent

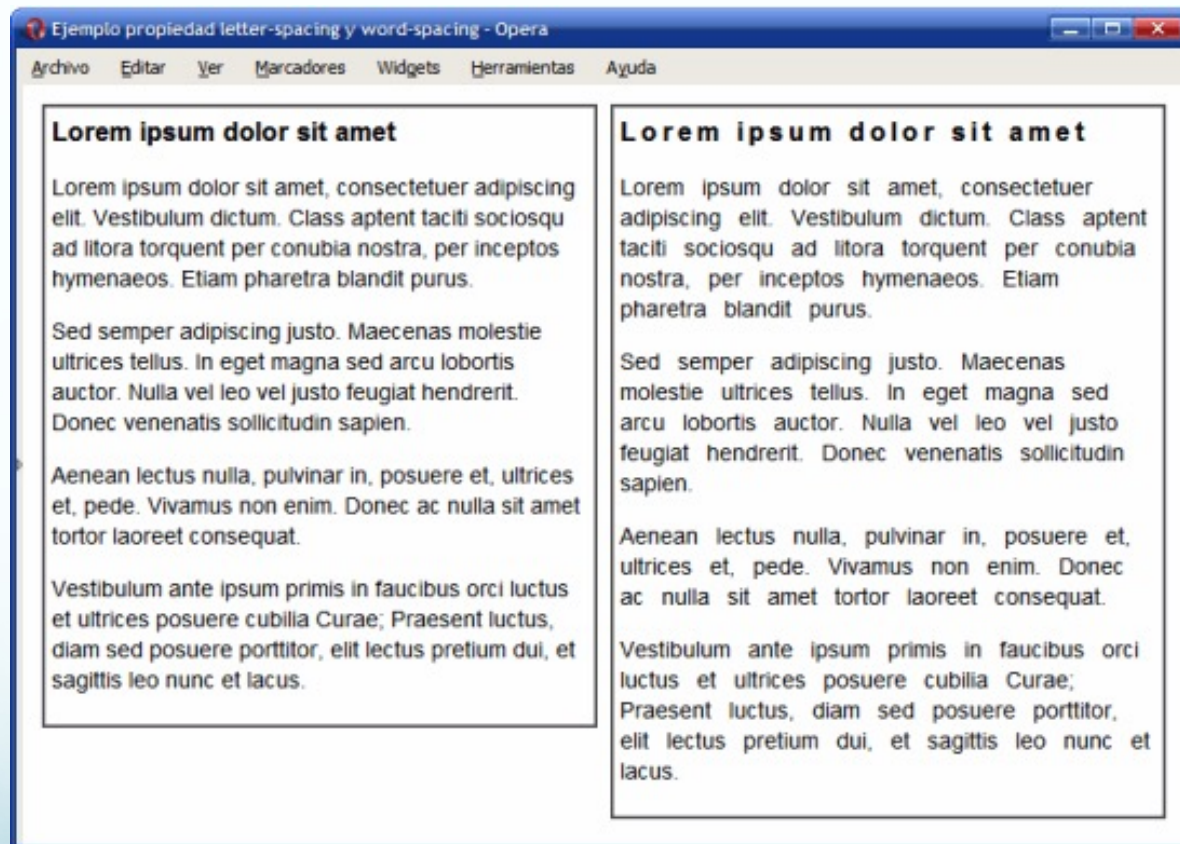


Otras propiedades de texto (IV)

Propiedad	letter-spacing
Valores	normal unidad de medida inherit
Se aplica a	Todos los elementos
Valor inicial	normal
Descripción	Permite establecer el espacio entre las letras que forman las palabras del texto

Propiedad	word-spacing
Valores	normal unidad de medida inherit
Se aplica a	Todos los elementos
Valor inicial	normal
Descripción	Permite establecer el espacio entre las palabras que forman el texto

Ejemplo: letter-spacing y word-spacing

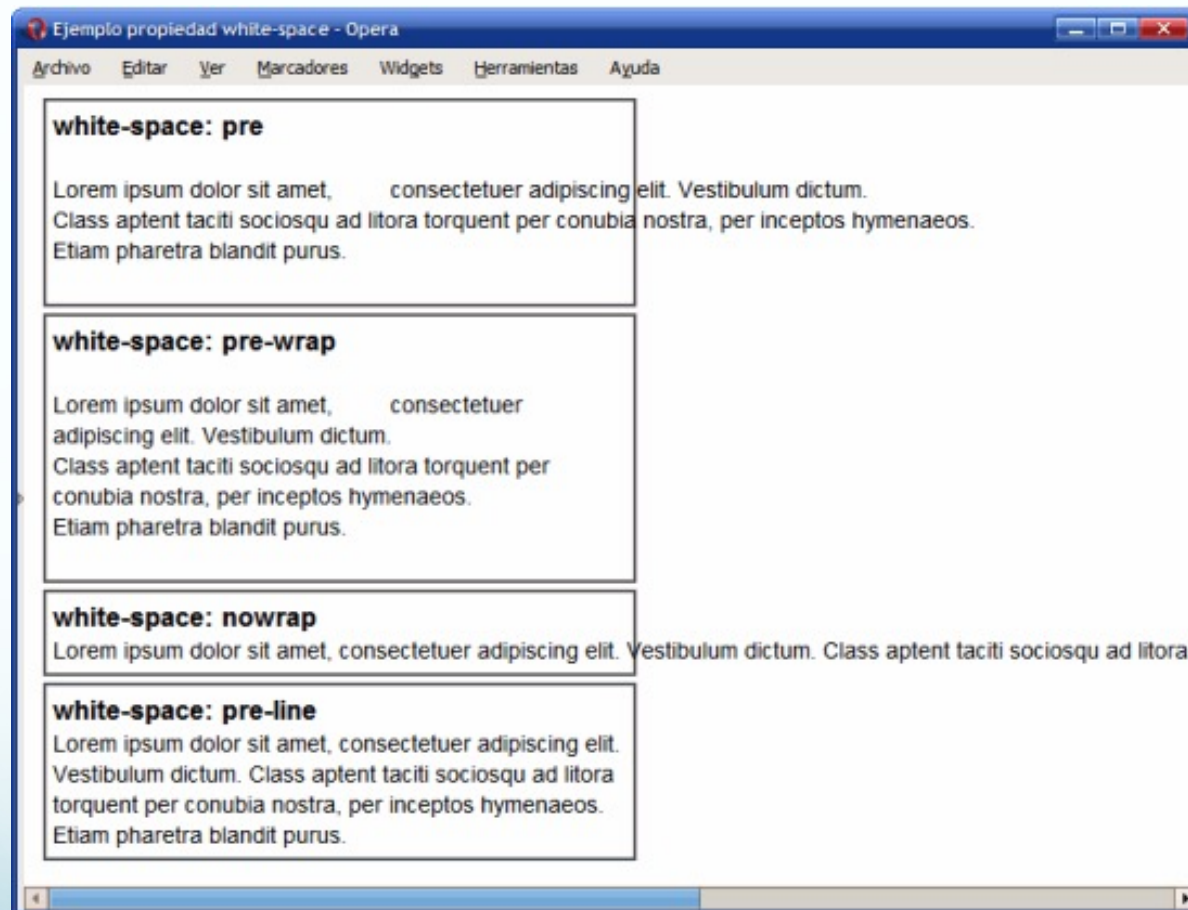


Otras propiedades de texto (V)

Propiedad	white-space
Valores	normal pre nowrap pre-wrap pre-line inherit
Se aplica a	Todos los elementos
Valor inicial	normal
Descripción	Establece el tratamiento de los espacios en blanco del texto

- **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.

Ejemplo: white-space



Ejercicio (I)

- La fuente base de la página debe ser: color negro, tipo Arial, tamaño 0.9em, interlineado 1.4.
- Los elementos `<h2>` de `.articulo` se muestran en color `#CC6600`, con un tamaño de letra de 1.6em, un interlineado de 1.2 y un margen inferior de 0.3em.
- Los elementos del `#menu` deben mostrar un margen a su derecha de 1em y los enlaces deben ser de color blanco y tamaño de letra 1.3em.
- El tamaño del texto de todos los contenidos de `#lateral` debe ser de 0.9em. La fecha de cada noticia debe ocupar el espacio de toda su línea y mostrarse en color gris claro `#999`. El elemento `<h3>` de `#noticias` debe mostrarse de color `#003366`.
- El texto del elemento `#publicidad` es de color gris oscuro `#555` y todos los enlaces de color `#CC6600`.
- Los enlaces contenidos dentro de `.articulo` son de color `#CC6600` y todos los párrafos muestran un margen superior e inferior de 0.3em.
- Añadir las reglas necesarias para que el contenido de `#secundario` se vea como en la imagen que se muestra.
- Añadir las reglas necesarias para que el contenido de `#pie` se vea como en la imagen que se muestra.

Ejercicio (II)

Logotipo

Buscar

>Lorem

Ipsum

Dolor

Sit

Amet

Noticias

dd/mm/aaaa

[Lorem ipsum dolor sit amet](#)

dd/mm/aaaa

[Consectetur adipiscing elit](#)

dd/mm/aaaa

[Donec molestie nunc eu sapien](#)

dd/mm/aaaa

[Maecenas aliquam dolor eget metus](#)

dd/mm/aaaa

[Fusce tristique lorem id metus](#)

[Enlaces relacionados](#)

[Proin placerat](#)

[Nulla in felis](#)

[Nam luctus](#)

Publicidad

Etiam fermentum, nisi tincidunt blandit interdum, massa velit posuere dolor, sed euismod sem odio at mi.

Duis porta placerat arcu.

Nullam felis pede, commodo vel, suscipit a, molestie vel, felis.

Maecenas mattis est vel est.

[Seguir leyendo...](#)

>Lorem ipsum dolor sit amet, consectetur adipiscing elit

Nullam est lacus, suscipit ut, dapibus quis, condimentum ac, risus. Vivamus vestibulum, ipsum sollicitudin faucibus pharetra, dolor metus fringilla dui, vel aliquet pede diam tempor tortor.

Vestibulum pulvinar urna et quam. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Nullam vel turpis vitae dui imperdiet laoreet. Quisque eget ipsum.

Donec commodo, turpis vel venenatis sollicitudin, quam ante convallis justo, sed eleifend justo lectus quis sapien. Ut consequat libero eget est.

[Seguir leyendo...](#)

Vivamus lobortis turpis ac ante fringilla faucibus

Quisque eget ipsum. Donec commodo, turpis vel venenatis sollicitudin, quam ante convallis justo, sed eleifend justo lectus quis sapien. Ut consequat libero eget est.

Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Nullam dictum hendrerit neque. Mauris id ligula non elit mattis semper. Fusce arcu ipsum, tempus eget, tincidunt at, imperdiet in, mi.

Sed fermentum cursus dolor. Aenean a diam. Phasellus feugiat. Donec tempor dignissim sem.

[Seguir leyendo...](#)

Phasellus blandit

Praesent sodales imperdiet augue.

Mauris lorem felis, semper eu, tincidunt eu, sollicitudin eget, sem. Nulla facilisi.

Morbi ut enim ut enim ultricies dapibus.

[Seguir leyendo...](#)

Nullam vel turpis

Donec commodo, turpis vel venenatis sollicitudin, quam ante convallis justo, sed eleifend justo lectus quis sapien. Ut consequat libero eget est.

[Seguir leyendo...](#)

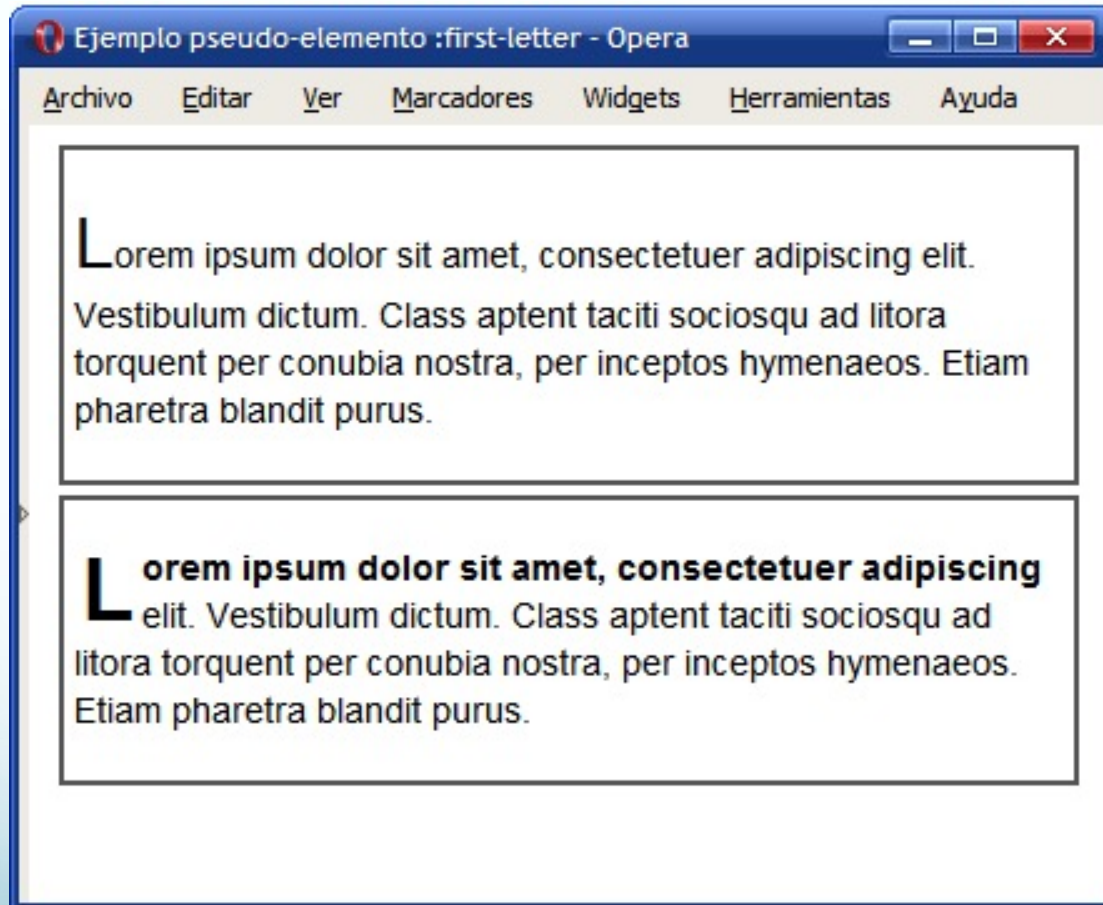
Nulla | Pharetra | Luctus | Ipsum | Proin | Placerat

© Copyright Lorem ipsum

Pseudo-elementos

- 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.
 - El pseudo-elemento **:first-line** permite aplicar estilos a la primera línea de un texto.
 - El pseudo-elemento **:first-letter** permite aplicar estilos a la primera letra de texto

Ejemplo: pseudo-elementos



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


Enlaces: Tamaño, color y decoración

Los estilos más sencillos que se pueden aplicar a los enlaces son los que modifican **su tamaño de letra, su color y la decoración del texto** del enlace. Utilizando las propiedades **text-decoration** y **font-weight**.

```
a { margin: 1em 0; display: block;}
```

```
.alternativo {color: #CC0000;}
```

```
.simple {text-decoration: none;}
```

```
.importante {font-weight: bold; font-size: 1.3em;}
```

```
.raro {text-decoration: underline;}
```

```
<a href="#">Enlace con el estilo por defecto</a>
```

```
<a class="alternativo" href="#">Enlace de color rojo</a>
```

```
<a class="simple" href="#">Enlace sin subrayado</a>
```

```
<a class="importante" href="#">Enlace muy importante</a>
```

```
<a class="raro" href="#">Enlace con un estilo raro</a>
```


Enlaces: pseudo-clases

CSS también permite aplicar diferentes estilos a un mismo enlace en función de su estado. De esta forma, es posible cambiar el aspecto de un enlace cuando por ejemplo el usuario pasa el ratón por encima o cuando el usuario pincha sobre ese enlace.

- **:link**, aplica estilos a los enlaces que apuntan a páginas o recursos que aún no han sido visitados por el usuario.
- **:visited**, aplica estilos a los enlaces que apuntan a recursos que han sido visitados anteriormente por el usuario.
 - El historial de enlaces visitados se borra automáticamente cada cierto tiempo y el usuario también puede borrarlo manualmente.
- **:hover**, aplica estilos al enlace sobre el que el usuario ha posicionado el puntero del ratón.
- **:active**, aplica estilos al enlace que está pinchando el usuario.
 - Los estilos sólo se aplican desde que el usuario pincha el botón del ratón hasta que lo suelta, por lo que suelen ser unas pocas décimas de segundo.

Ejemplo: pseudo-clases

- Las pseudo-clases siempre incluyen dos puntos (:) por delante de su nombre y siempre se añaden a continuación del elemento al que se aplican, sin dejar ningún espacio en blanco por delante:

/* Incorrecto: */

a: visited { ... }

/* Incorrecto */

a :visited { ... }

/* Correcto*/

a:visited { ... }

- Las pseudo-clases también se pueden combinar con cualquier otro selector complejo:

a.importante:visited { ... }

a#principal:hover { ... }

div#menu ul li a.primer:active { ... }

Ejercicio (I)

1. En su estado normal, los enlaces se muestran de color rojo #CC0000.
2. Cuando el usuario pasa su ratón sobre el enlace, se muestra con un color de fondo rojo #CC0000 y la letra de color blanco #FFF.
3. Los enlaces visitados se muestran en color gris claro #CCC.

Imágenes: Ancho y Alto

Utilizando las propiedades **width** y **height**, es posible mostrar las imágenes con cualquier altura/anchura, independientemente de su altura/anchura real.

No obstante, si se utilizan alturas/anchuras diferentes de las reales, el navegador deforma las imágenes y el resultado estético es muy desagradable.

```
#destacada {  
  width: 120px;  
  height: 250px;  
}
```

```

```

Por lo tanto, se debe crear una nueva regla CSS (y un nuevo selector) para cada una de las diferentes imágenes del sitio web.

En la práctica, esta forma de trabajo produce una sobrecarga de estilos que la hace inviable.

```

```

Imágenes enlace

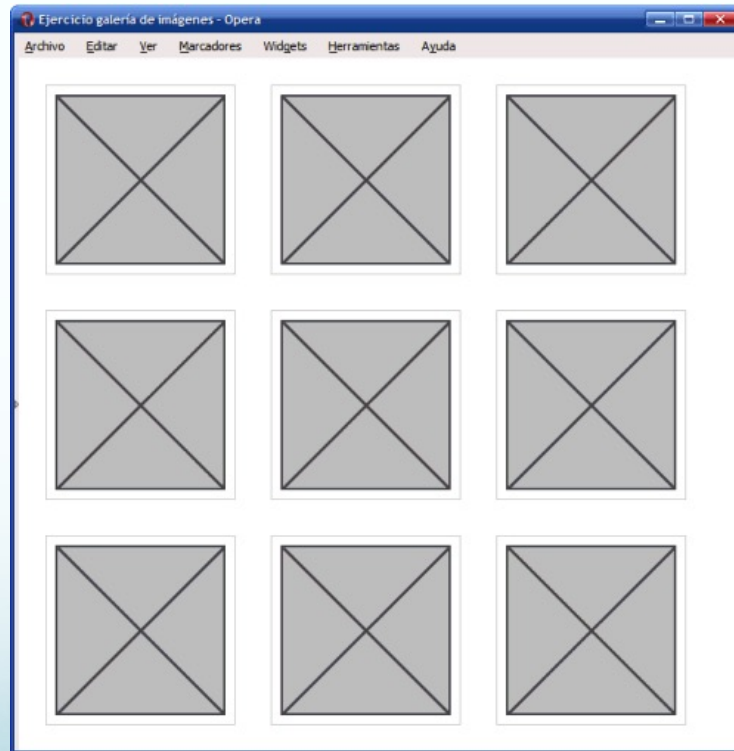
Cuando una imagen forma parte de un enlace, los navegadores muestran por defecto un borde azul grueso alrededor de las imágenes.

Por tanto, una de las reglas más utilizadas en los archivos CSS es la que elimina los bordes de las imágenes con enlaces:

```
img { border: none; }
```

Ejercicio

Definir las reglas CSS que permiten mostrar una galería de imágenes similar a la que se muestra en la siguiente imagen:



Listas: viñetas personalizadas

- Por defecto, los navegadores muestran los elementos de las listas no ordenadas con una viñeta formada por un **pequeño círculo** de color negro.
- Los elementos de las listas ordenadas se muestran por defecto con la **numeración decimal** utilizada en la mayoría de países.
- CSS define varias propiedades para controlar el **tipo de viñeta que muestran** las listas, además de poder **controlar la posición** de la propia viñeta.

Tipo de viñeta

Propiedad	list-style-type
Valores	disc circle square decimal decimal-leading-zero lower-roman upper-roman lower-greek lower-latin upper-latin armenian georgian lower-alpha upper-alpha none inherit
Se aplica a	Elementos de una lista
Valor inicial	disc
Descripción	Permite establecer el tipo de viñeta mostrada para una lista

```
<ul style="list-style-type: square">
  <li>list-style-type: square</li>
  <li>Elemento</li>
  <li>Elemento</li>
</ul>
<ol style="list-style-type: lower-roman">
  <li>list-style-type: lower-roman</li>
  <li>Elemento</li>
  <li>Elemento</li>
</ol>
```

```
<ol style="list-style-type: decimal-leading-zero;
padding-left: 2em;">
  <li>list-style-type: decimal-leading-zero</li>
  <li>Elemento</li> <li>Elemento</li>
</ol>
```

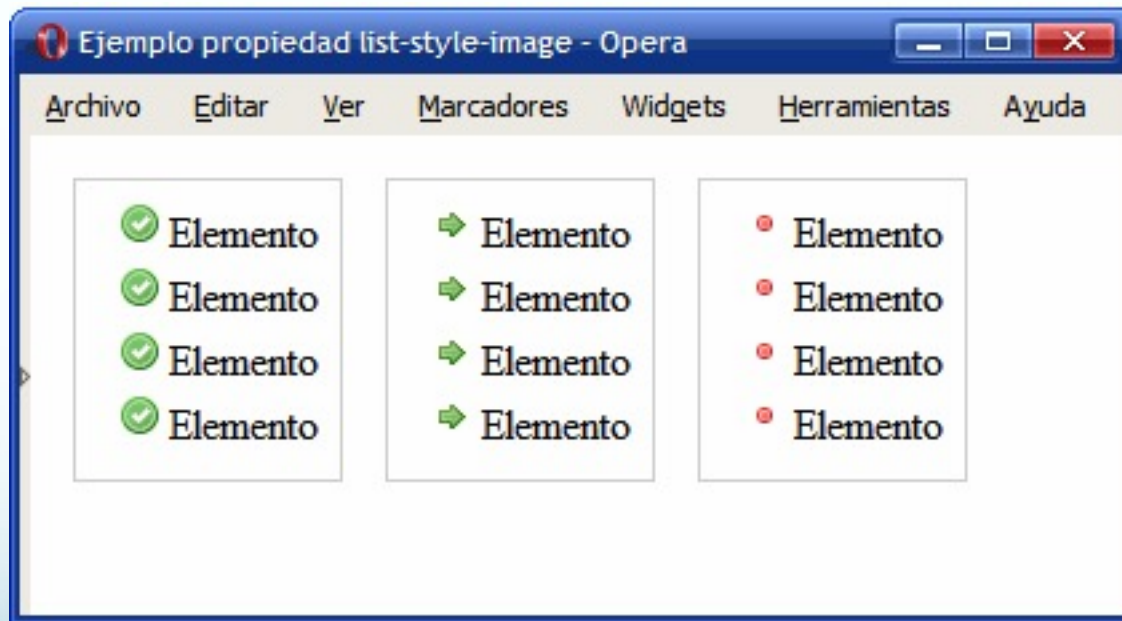
Posición viñeta

Propiedad	list-style-position
Valores	inside outside inherit
Se aplica a	Elementos de una lista
Valor inicial	outside
Descripción	Permite establecer la posición de la viñeta de cada elemento de una lista

Propiedad	list-style-image
Valores	url none inherit
Se aplica a	Elementos de una lista
Valor inicial	none
Descripción	Permite reemplazar las viñetas automáticas por una imagen personalizada

Ejemplo: list-style-image

```
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"); }
```



Listas shorthand

Propiedad	list-style
Valores	(list-style-type list-style-position list-style-image) inherit
Se aplica a	Elementos de una lista
Valor inicial	-
Descripción	Propiedad que permite establecer de forma simultánea todas las opciones de una lista

Menú Vertical

- Los sitios web correctamente diseñados emplean las listas de elementos para crear todos sus menús de navegación. Utilizando la etiqueta `` de HTML se agrupan todas las opciones del menú y haciendo uso de CSS se modifica su aspecto para mostrar un menú horizontal o vertical.
 1. Definir la anchura del menú
 2. Eliminar las viñetas automáticas y todos los márgenes y espaciados aplicados por defecto
 3. Añadir un borde al menú de navegación y establecer el color de fondo y los bordes de cada elemento del menú
 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.
- Los tipos de menús verticales que se pueden definir mediante las propiedades CSS son innumerables, como se puede observar en la página
<http://www.exploding-boy.com/images/EBmenus/menus.html>

Ejemplo: Menú Vertical

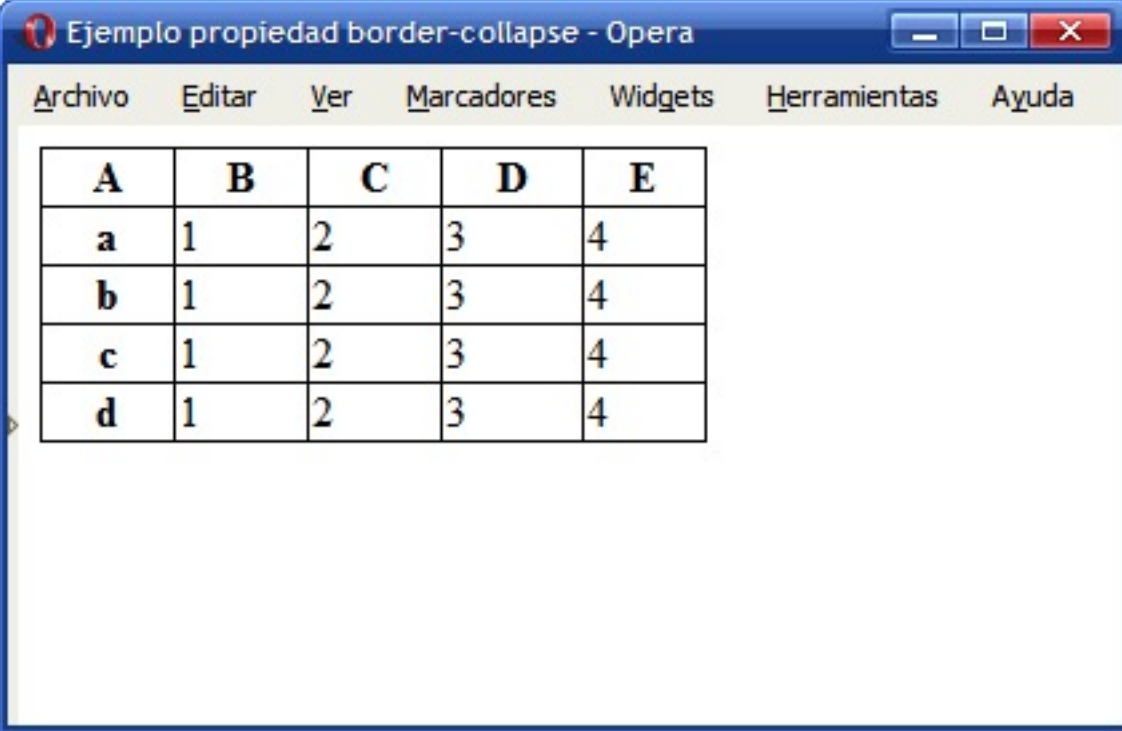


Tablas (I)

Propiedad	border-collapse
Valores	collapse separate inherit
Se aplica a	Todas las tablas
Valor inicial	separate
Descripción	Define el mecanismo de fusión de los bordes de las celdas adyacentes de una tabla

- El modelo **collapse** fusiona de forma automática los bordes de las celdas adyacentes, mientras que el modelo **separate** fuerza a que cada celda muestre sus cuatro bordes. Por defecto, los navegadores utilizan el modelo separate

Ejemplo: border-collapse



The screenshot shows a web browser window with the title "Ejemplo propiedad border-collapse - Opera". The browser's menu bar includes "Archivo", "Editar", "Ver", "Marcadores", "Widgets", "Herramientas", and "Ayuda". The main content area displays a table with the following structure:

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

Tablas (II)

Propiedad	border-spacing
Valores	unidad de medida unidad de medida? inherit
Se aplica a	Todas las tablas
Valor inicial	0
Descripción	Establece la separación entre los bordes de las celdas adyacentes de una tabla

- 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.

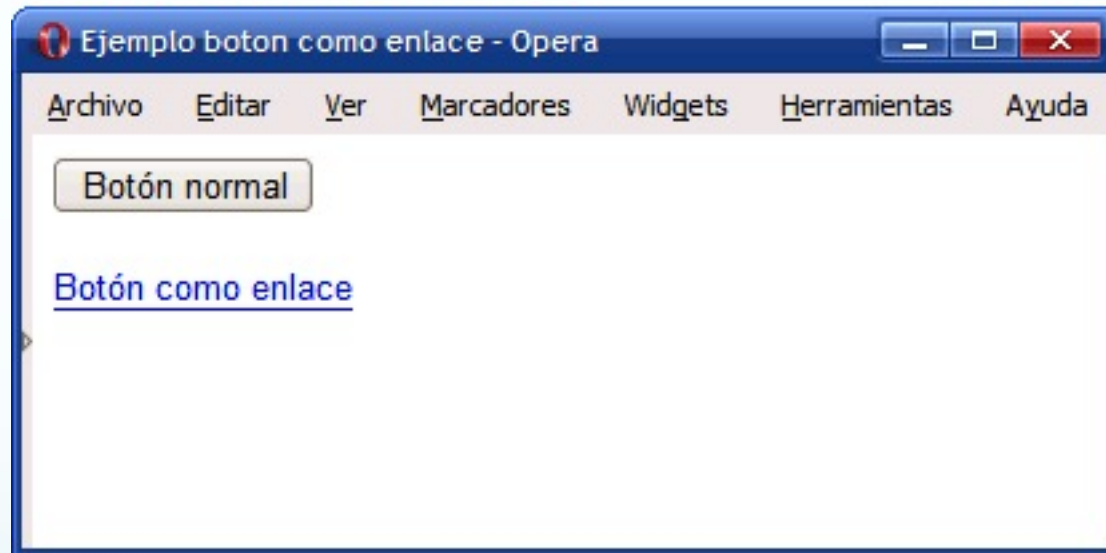
Formularios (I)

- **Mostrar un botón como un enlace**

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

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

Ejemplo

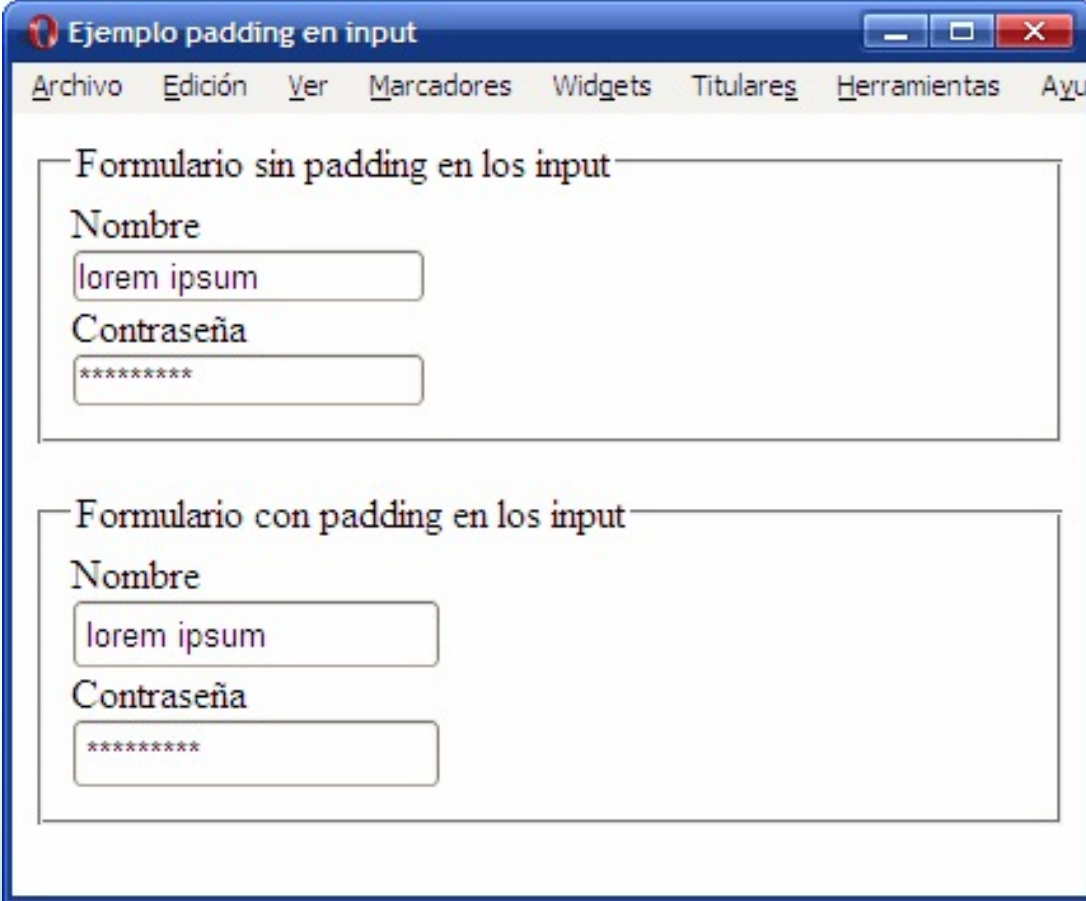


Formularios (II)

- **Mejoras en los campos de texto**
 - Añadiendo un pequeño padding a cada elemento `<input>`, se mejora notablemente el aspecto del formulario:

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

Ejemplo: cajas de texto



The screenshot shows a web application window titled "Ejemplo padding en input". It contains two forms for comparison. The first form, titled "Formulario sin padding en los input", shows two input fields. The first field contains the text "lorem ipsum" and the second field contains "*****". The second form, titled "Formulario con padding en los input", shows two input fields. The first field contains the text "lorem ipsum" and the second field contains "*****". The padding in the second form makes the text appear more centered and consistent in width between the two fields.

Ejemplo padding en input

Archivo Edición Ver Marcadores Widgets Titulares Herramientas Ayu

Formulario sin padding en los input

Nombre
lorem ipsum

Contraseña

Formulario con padding en los input

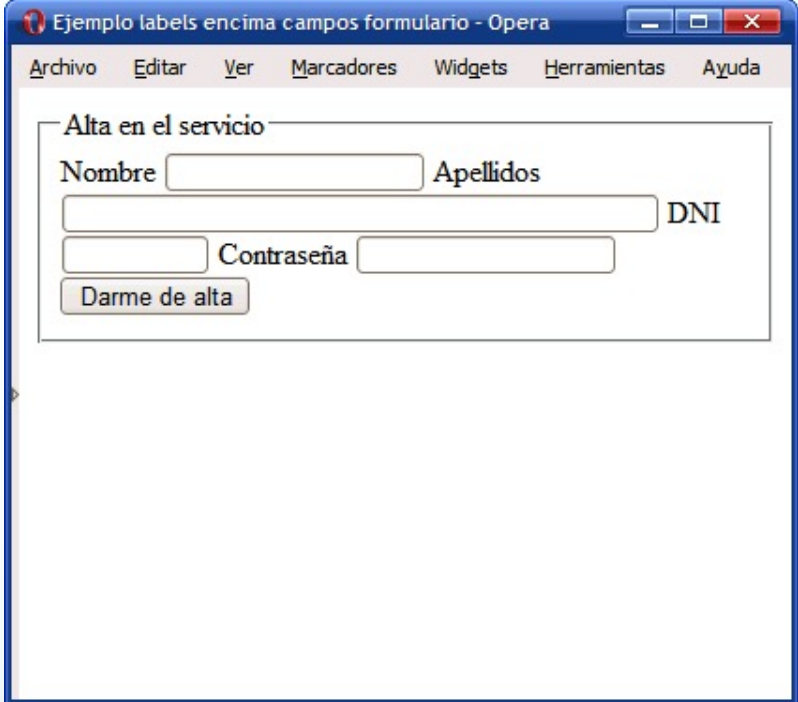
Nombre
lorem ipsum

Contraseña

Formularios (III)

- **Labels alineadas y formateadas**

Los elementos `<input>` y `<label>` de los formularios son elementos en línea, por lo que el aspecto que muestran los formularios por defecto, es similar al de la siguiente imagen:



Ejemplo labels encima campos formulario - Opera

Archivo Editar Ver Marcadores Widgets Herramientas Ayuda

Alta en el servicio

Nombre Apellidos

DNI

Contraseña

Labels en formularios

```
<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" />

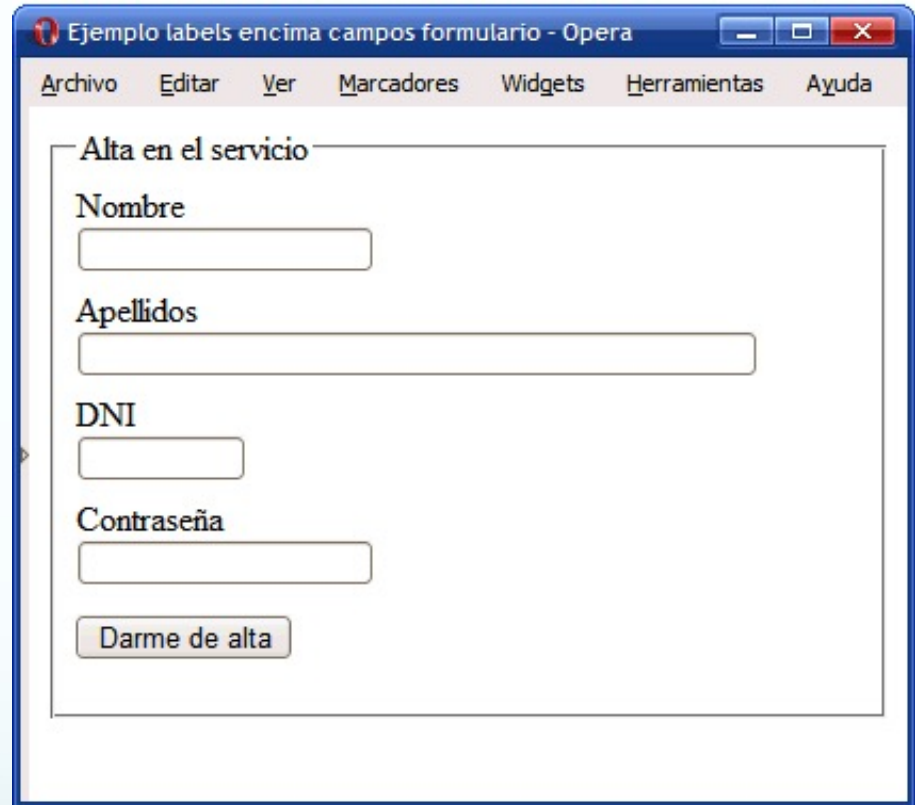
    <label for="dni">DNI</label>
    <input type="text" id="dni" size="10" maxlength="9" />

    <label for="contrasena">Contraseña</label>
    <input type="password" id="contrasena" />

    <input class="btn" type="submit" value="Darme de alta" />
  </fieldset>
</form>
```

Ejemplo: labels alineación

```
label {  
    display: block;  
    margin: .5em 0 0 0;  
}  
  
.btn {  
    display: block;  
    margin: 1em 0;  
}
```



The screenshot shows a web browser window with the title "Ejemplo labels encima campos formulario - Opera". The browser's menu bar includes "Archivo", "Editar", "Ver", "Marcadores", "Widgets", "Herramientas", and "Ayuda". The form content is as follows:

- Section header: "Alta en el servicio"
- Form fields:
 - "Nombre" with a single-line text input.
 - "Apellidos" with a multi-line text input.
 - "DNI" with a single-line text input.
 - "Contraseña" with a single-line text input.
- Submit button: "Dar de alta"

Ejercicio

Ejemplo labels al lado de campos formulario - Opera

Archivo Editar Ver Marcadores Widgets Herramientas Ayuda

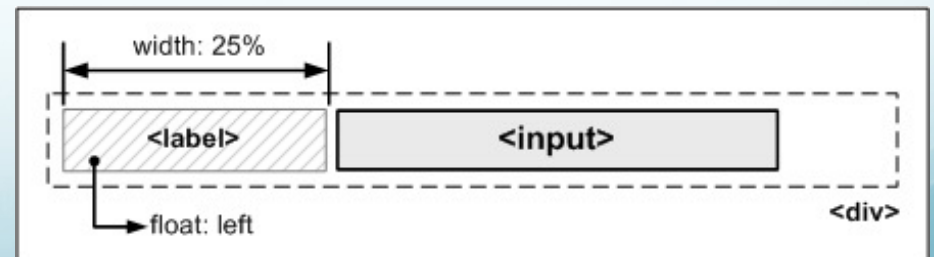
Alta en el servicio

Nombre

Apellidos

DNI

Contraseña

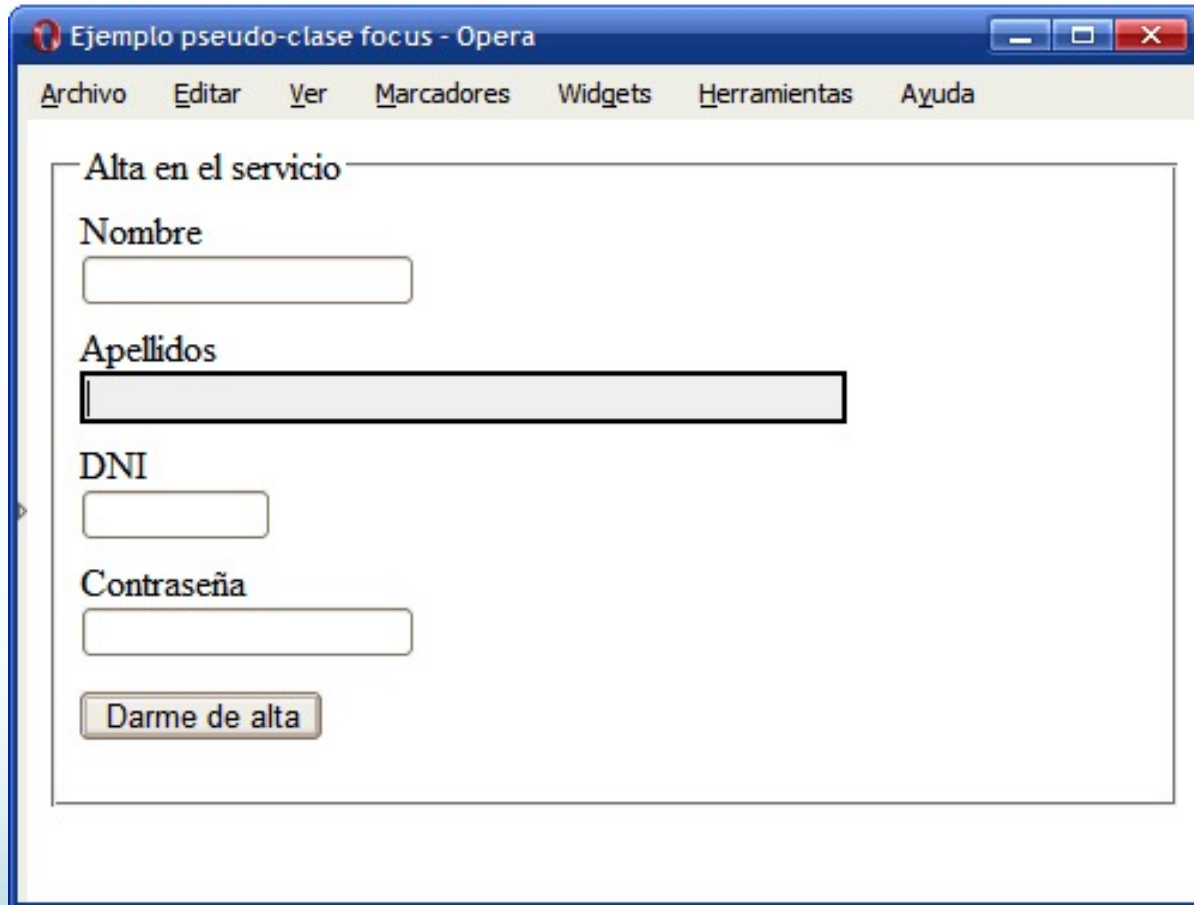


Resaltar el campo seleccionado

- Una de las mejoras más útiles para los formularios HTML consiste en resaltar de alguna forma especial el campo en el que el usuario está introduciendo datos. Para ello, CSS define la pseudo-clase **:focus**, que permite aplicar estilos especiales al elemento que en ese momento tiene el *foco* o atención del usuario.

```
input:focus {  
    border: 2px solid #000;  
    background: #F3F3F3;  
}
```

Ejemplo



The screenshot shows a web browser window with the title "Ejemplo pseudo-clase focus - Opera". The browser's menu bar includes "Archivo", "Editar", "Ver", "Marcadores", "Widgets", "Herramientas", and "Ayuda". The main content area displays a registration form titled "Alta en el servicio". The form contains the following elements:

- A label "Nombre" followed by a text input field.
- A label "Apellidos" followed by a text input field.
- A label "DNI" followed by a text input field.
- A label "Contraseña" followed by a text input field.
- A button labeled "Dar de alta" at the bottom of the form.

Centrar una página horizontalmente

- A medida que aumenta el tamaño y la resolución de las pantallas de ordenador, se hace más difícil diseñar páginas que se adapten al tamaño de la ventana del navegador. El principal reto que se presenta con resoluciones superiores a 1024 x 768 píxel.
- Utilizando la propiedad **margin** de CSS, es muy sencillo centrar una página web horizontalmente. La solución consiste en agrupar todos los contenidos de la página en un elemento **<div>** y asignarle a ese **<div>** unos márgenes laterales automáticos.

```
#contenedor {  
  width: 300px;  
  margin: 0 auto;  
}
```

```
<body>  
  <div id="contenedor">  
    <h1>Lorem ipsum dolor sit amet</h1>  
  </div>  
</body>
```

Centrar una página verticalmente

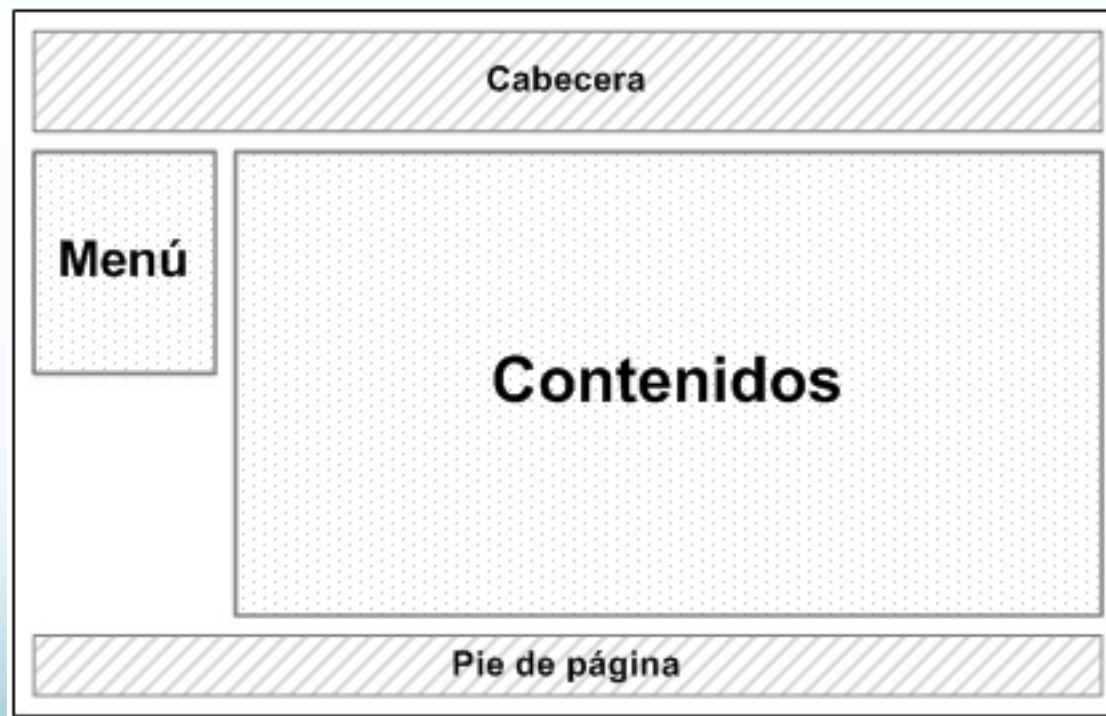
- Aunque centrar una página web horizontalmente es muy sencillo, centrarla verticalmente es mucho más complicado.
- Afortunadamente, no es muy común que una página web aparezca centrada de forma vertical.

```
#contenedor {  
  width: 300px;  
  height: 250px;  
  margin: auto;  
}
```

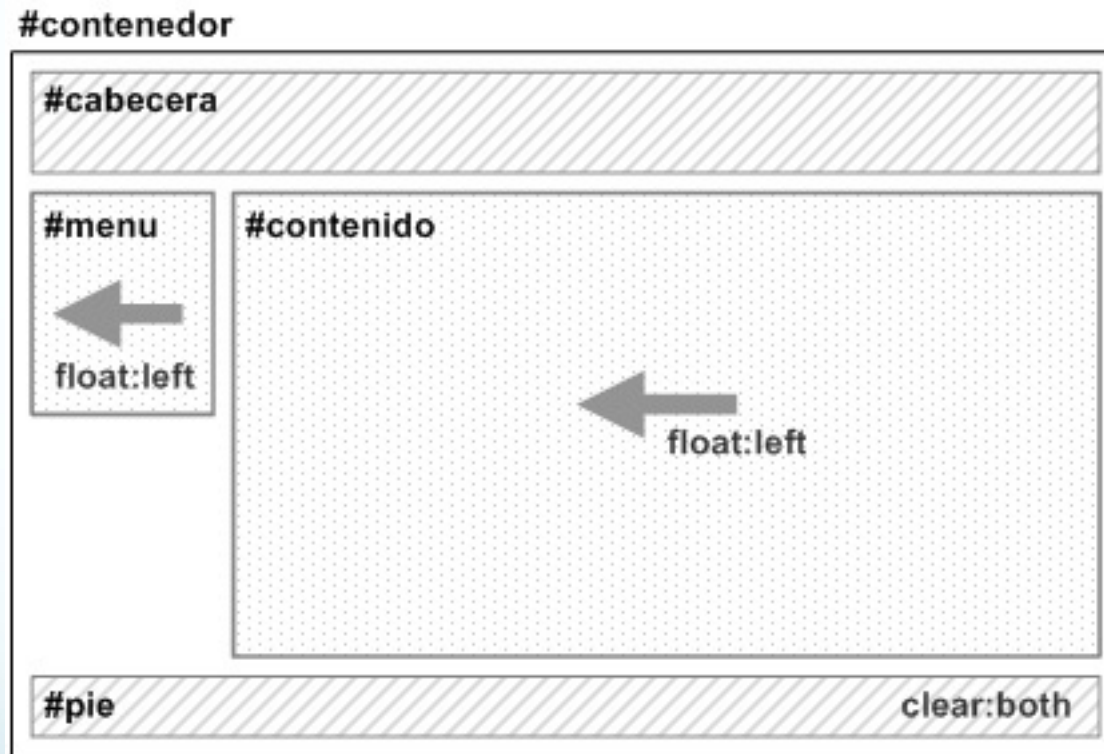
```
<body>  
  <div id="contenedor">  
    <h1>Lorem ipsum dolor sit amet</h1>  
    ...  
  </div>  
</body>
```

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:



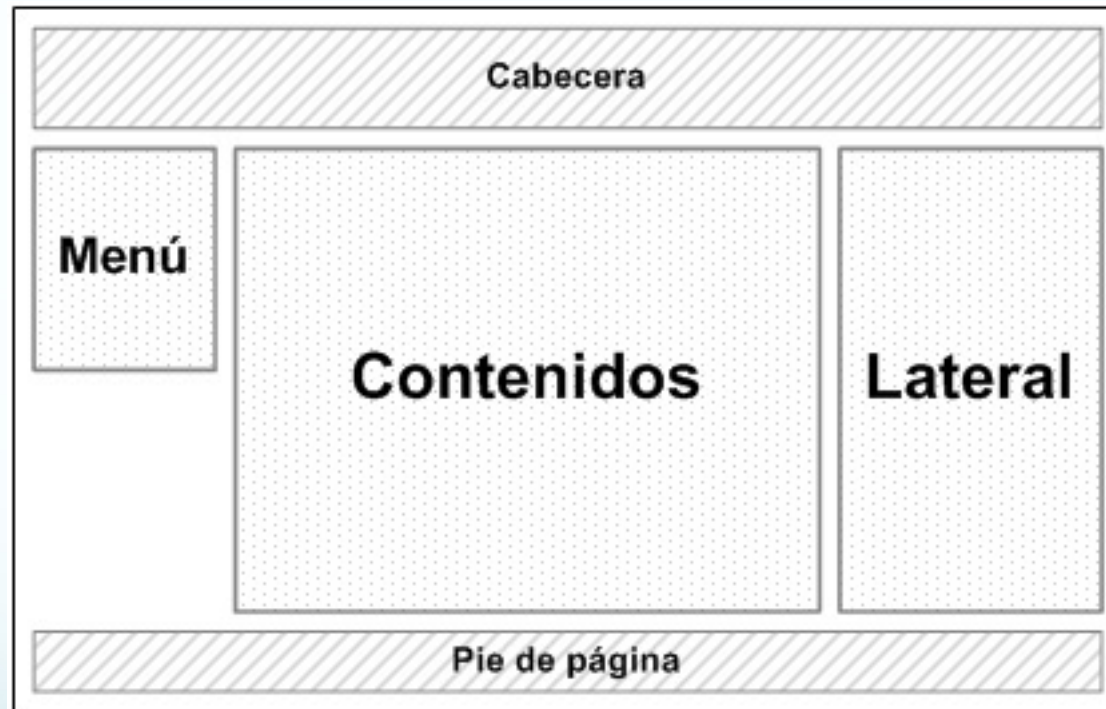
Solución (I)



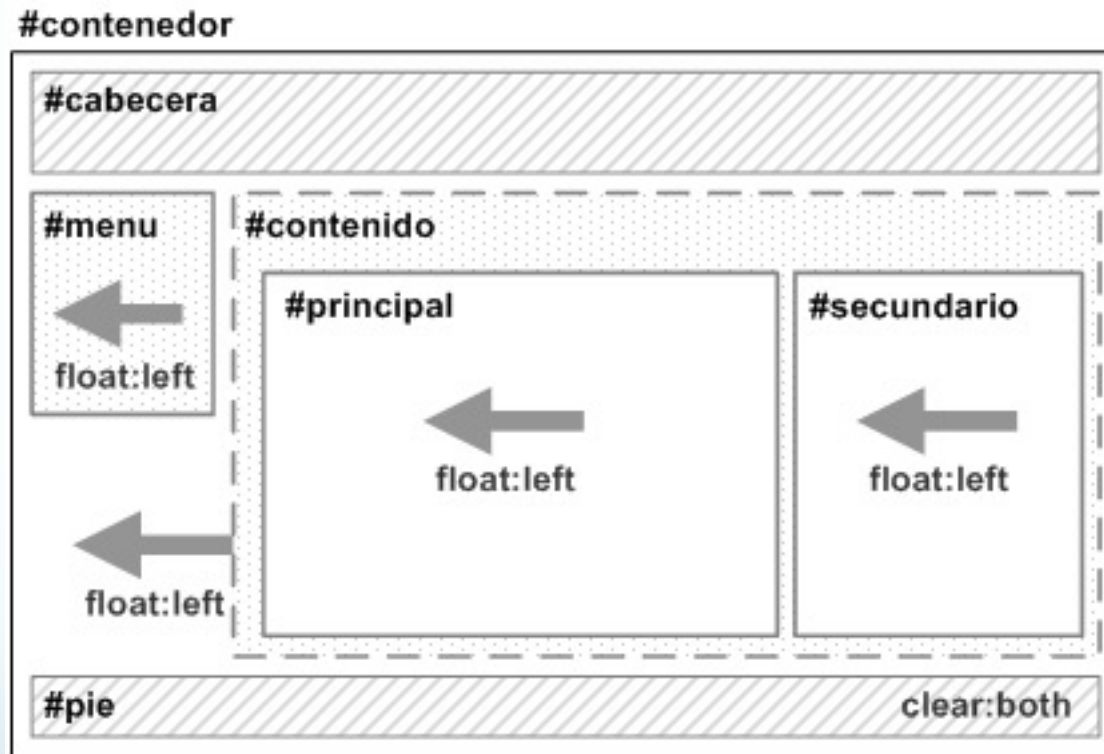
Solución (II)

```
#contenedor {  
    width: 700px;  
}  
#cabecera {  
}  
#menu {  
    float: left;  
    width: 150px; /* se puede aplicar porcentajes 15% */  
}  
#contenido {  
    float: left;  
    width: 550px; /* se puede aplicar porcentajes 85% */  
}  
#pie {  
    clear: both;  
}
```

Diseño a 3 columnas con cabecera y pie de página



Solución (I)



Solución (II)

```
#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;  
}
```

Bibliografía

[1] Javier Eguiluz; Introducción a CSS;
<http://librosweb.es/libro/css/>

[2] Jorge A. Tejedor; Asignatura: Tecnologías de Desarrollo para Clientes Ligeros; Master en Ingeniería Web; Madrid; 2013.