

Índice: Tema 2.4 (Parte 2)

2.4 Hojas de estilo en cascada CSS

2.4.1 El lenguaje XHTML

2.4.2 Hojas de estilo CSS

2.4.3 Reglas de estilo

2.4.4 El modelo de cajas

2.4.5 Propiedades de los elementos básicos

Índice: Tema 2.4 (Parte 2)

2.4 Hojas de estilo en cascada CSS

2.4.1 El lenguaje XHTML

2.4.2 Hojas de estilo CSS

2.4.3 Reglas de estilo

2.4.4 El modelo de cajas

2.4.5 Propiedades de los elementos básicos



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.4 EL MODELO DE CAJAS

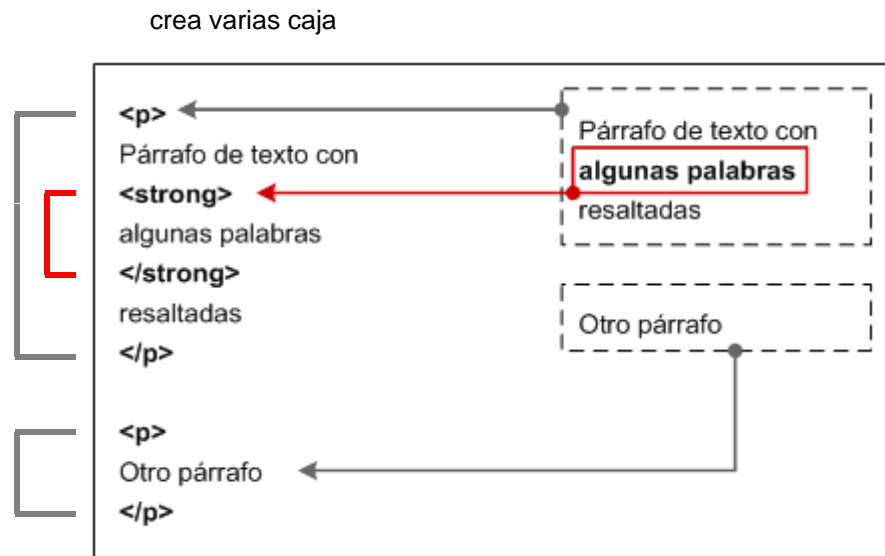


➤ El modelo de cajas

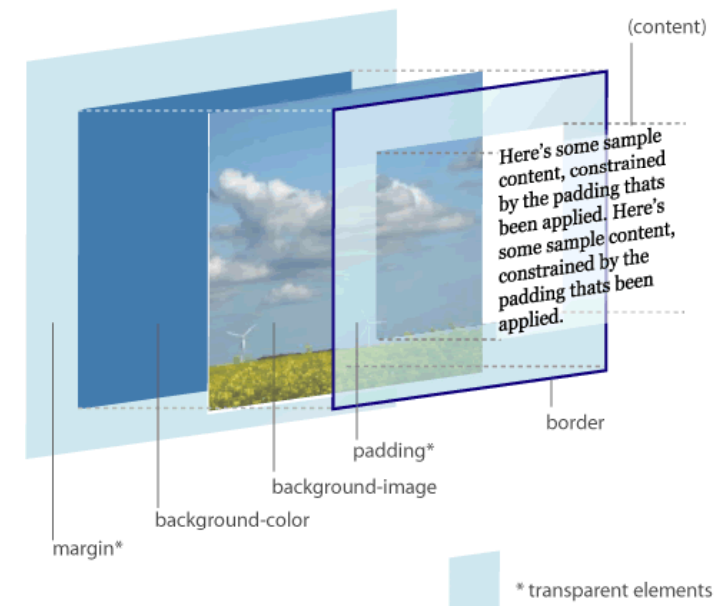
- ✓ El modelo de cajas o **box model** es seguramente la característica más importante del lenguaje de hojas de estilos CSS, ya que condiciona el diseño de todas las páginas web.
- ✓ El *box model* es la característica que hace que todos los elementos incluidos en una página HTML se representen mediante cajas rectangulares cuyo aspecto se puede controlar a través de CSS.
- ✓ El diseño de cualquier página XHTML está compuesto por cajas rectangulares. CSS permite definir la altura y anchura de cada caja, el margen existente entre cajas y el espacio de relleno interior que muestra cada caja.
- ✓ Además, CSS permite controlar la forma en la que se visualizan las cajas: se pueden ocultar, desplazar respecto de su posición original y fijarlas en una posición específica dentro del documento.

➤ Elementos del modelo de cajas

- ✓ Las cajas de una página se crean automáticamente. Cada vez que se inserta una etiqueta o elemento en la página, se crea una nueva caja rectangular que encierra los contenidos del elemento.
- ✓ El siguiente esquema muestra la creación automática de cajas:



THE CSS BOX MODEL HIERARCHY





2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.4 EL MODELO DE CAJAS



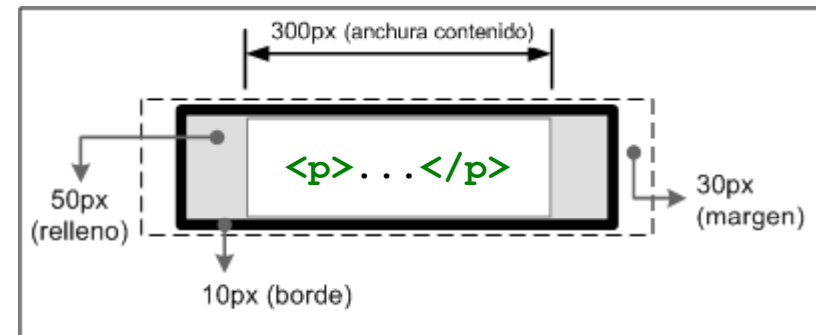
➤ Propiedades (1)

- ✓ A continuación se muestran las propiedades que controlan cada uno de los elementos de la caja relacionados con sus **dimensiones**:
 - Anchura: `width: 200px`
 - Altura: `height: 60px`
 - Margen superior: `margin-top: 2em`
 - Margen derecho: `margin-right: 2em`
 - Margen inferior: `margin-bottom: 2em`
 - Margen izquierdo: `margin-left: 2em`
 - Margen en general: `margin: sup der inf izq` (notación `{1,4}`)
- ✓ Tomando como referencia el margen (`margin`), se pueden definir propiedades similares para el relleno (`padding`), la anchura (`border-*-width`), el estilo (`border-*-style`) y el color (`border-*-color`) de los bordes.

➤ Propiedades (2)

- ✓ La anchura y altura de la caja 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. En el siguiente ejemplo se muestran los estilos CSS de un elemento **<p>** cuya anchura de caja en pantalla sería de 480:

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





2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.4 EL MODELO DE CAJAS



➤ Propiedades (3)

- ✓ 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 (padding), ya que el color de los bordes se controla directamente desde los bordes y las zonas de los márgenes siempre son transparentes.

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

- Color de fondo: `background-color: blue`
- Imagen de fondo (idf): `background-image: url(images/fondo.gif)`
- Repetición de la idf: `background-repeat: repeat (-x, -y, no-)`
- Posición de la idf: `background-position: left, top, ... ó {x% y%}`
- Comportamiento de la idf: `background-attachment: scroll ó fixed` sube y baja
- Fondo en general: `background: blue url() repeat-x fixed`

➤ Propiedades (4)

- ✓ Utilizar las propiedades básicas de las cajas para dar formato a la siguiente web:





2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.4 EL MODELO DE CAJAS



➤ Posicionamiento de las cajas (1)

- ✓ El estándar de CSS define cinco modelos diferentes para **posicionar** una caja mediante el atributo **position**:
 - **static**: Este es el modo de posicionamiento normal (por defecto).
 - **relative**: La caja se coloca con el posicionamiento normal, pero se desplaza según los atributos **left**, **right**, **top**, **bottom**.
 - **absolute**: La caja se coloca de manera absoluta con respecto a la caja que la contiene según los atributos **left**, **right**, **top**, **bottom**.
 - **fixed**: Es un tipo de posicionamiento absoluto, pero que convierte la caja en un elemento inamovible.
 - **float**: 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.



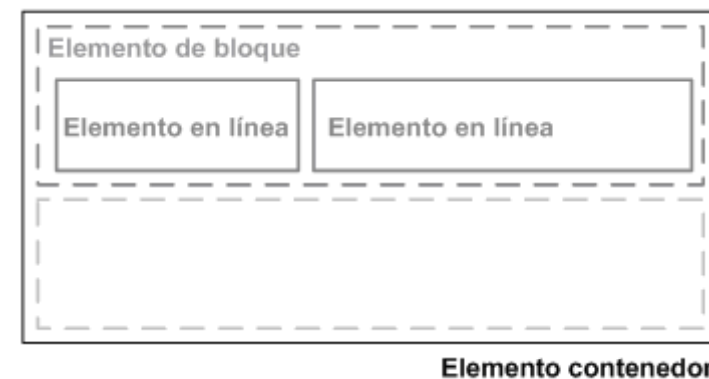
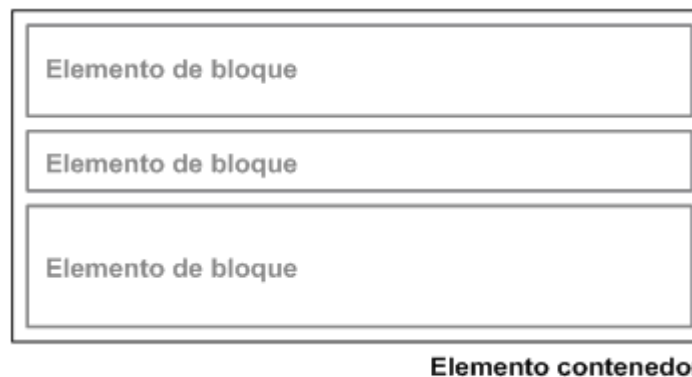
2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.4 EL MODELO DE CAJAS



➤ Posicionamiento de las cajas (2)

- ✓ El **static** o **posicionamiento normal** es el modelo que utilizan por defecto los navegadores para mostrar los elementos de las páginas.
 - Las cajas de los elementos de bloque se muestran una tras otra verticalmente comenzando en la parte superior del bloque contenedor y siguiendo el orden en el que aparecen los elementos correspondientes en el documento.
 - Las cajas de los elementos de línea se muestran una detrás de otra de forma horizontal comenzando desde la posición más a la izquierda de su elemento contenedor.





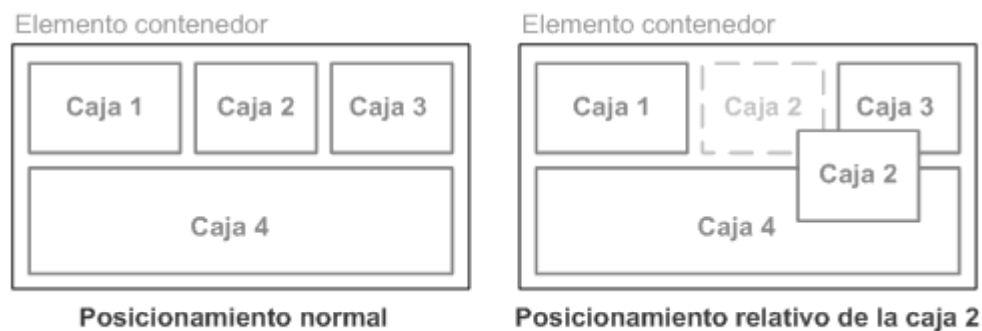
2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.4 EL MODELO DE CAJAS



➤ Posicionamiento de las cajas (3)

- ✓ El **relative** o **posicionamiento relativo** permite desplazar una caja respecto de su posición original establecida mediante el posicionamiento normal.
 - Una vez establecido **position: relative**, la propiedad **left** desplaza la caja hacia su derecha de acuerdo al **valor** establecido, la propiedad **right** la desplaza hacia su izquierda, la posición **top** desplaza la caja de forma descendente y la propiedad **bottom** desplaza la caja de forma ascendente. Además, los valores negativos tienen el efecto inverso, es decir **left=-right** y **top=-bottom**.
 - En el posicionamiento relativo, el resto de cajas no modifican su posición original.





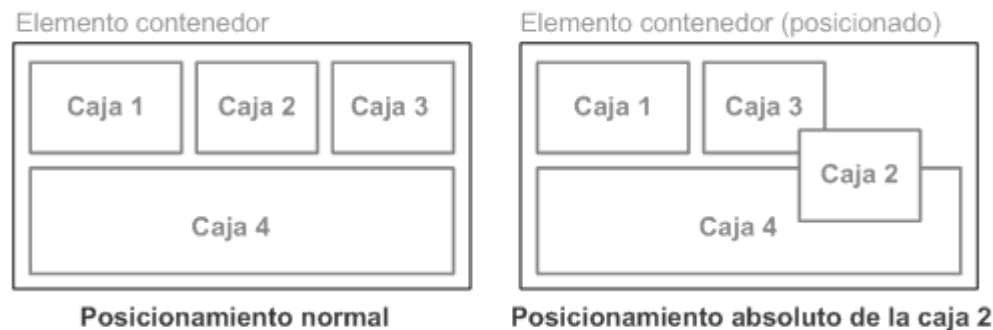
2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.4 EL MODELO DE CAJAS



➤ Posicionamiento de las cajas (4)

- ✓ El **absolute** o **posicionamiento absoluto** se emplea para establecer de forma precisa la posición en la que se muestra la caja de un elemento. Tiene dos diferencias respecto al relativo:
 - La caja se coloca de manera absoluta con respecto a la caja que la contiene. Por tanto los atributos para posicionarla dependen ahora de la posición de su elemento contenedor.
 - Cuando se posiciona una caja de forma absoluta, el resto de elementos ignoran que esa caja exista. Por este motivo, en el siguiente ejemplo, la caja 3 deja su lugar original y pasa a ocupar el hueco dejado por la caja 2.





2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.4 EL MODELO DE CAJAS



➤ Posicionamiento de las cajas (5)

- ✓ El estándar CSS considera que el **fixed** o **posicionamiento fijo** es un caso particular del posicionamiento absoluto, ya que sólo se diferencian en el comportamiento de las cajas posicionadas.
- ✓ La principal característica de una caja posicionada de forma fija es que su posición es inamovible dentro de la ventana del navegador. El posicionamiento fijo hace que las cajas no modifiquen su posición ni aunque el usuario suba o baje la página en la ventana de su navegador.
- ✓ Si la página se visualiza en un medio paginado (por ejemplo en una impresora) las cajas posicionadas de forma fija se repiten en todas las páginas. Así, esta característica puede ser útil para crear encabezados o pies de página en páginas HTML preparadas para imprimir.



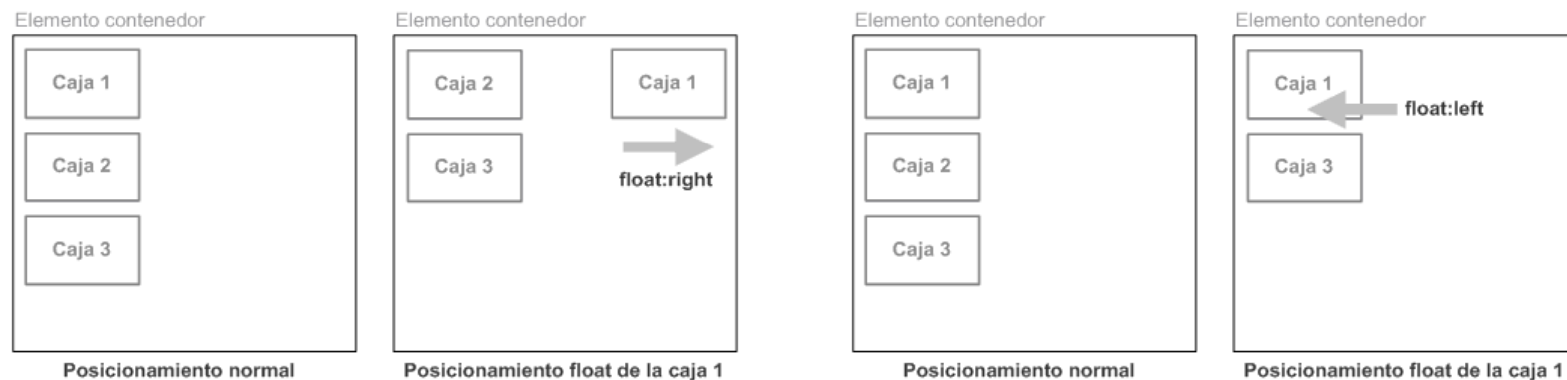
2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.4 EL MODELO DE CAJAS



➤ Posicionamiento de las cajas (6)

- ✓ El **float** o **posicionamiento flotante** es el más difícil de comprender pero al mismo tiempo es el más utilizado. Como se verá más adelante, la mayoría de estructuras de las páginas web complejas están diseñadas con el posicionamiento flotante. Cuando una caja se posiciona como flotante:
 - Se desplaza hasta la zona más a la izquierda (con **float: left**) o más a la derecha (con **float: right**) de la posición en la que originalmente se encontraba.
 - 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.





2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.4 EL MODELO DE CAJAS



➤ Posicionamiento de las cajas (7)

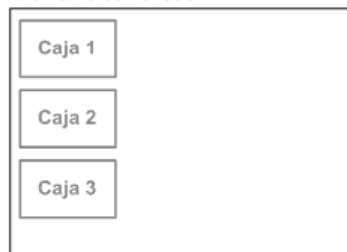
Div etiqueta que sirve para hacer una caja
a p img title form tiene su propia ca

movemes div : mueve td MMMM

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

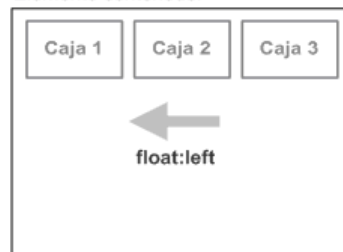
- Como la caja 1 ya estaba posicionada lo más a la izquierda posible, la caja 2 sólo puede colocarse al lado del borde derecho de la caja 1, que es el sitio más a la izquierda posible respecto de la zona en la que se encontraba.
- Si no existe sitio en la línea actual, la caja flotante baja a la línea inferior hasta que encuentra el sitio necesario para mostrarse lo más a la izquierda o lo más a la derecha posible en esa nueva línea.

Elemento contenedor



Posicionamiento normal

Elemento contenedor



Posicionamiento float de las 3 cajas

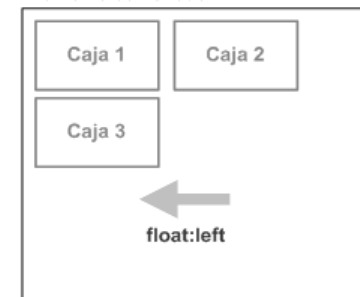
a todas clear left para poner una caja que no siga

Elemento contenedor



Posicionamiento normal

Elemento contenedor



Posicionamiento float de las 3 cajas



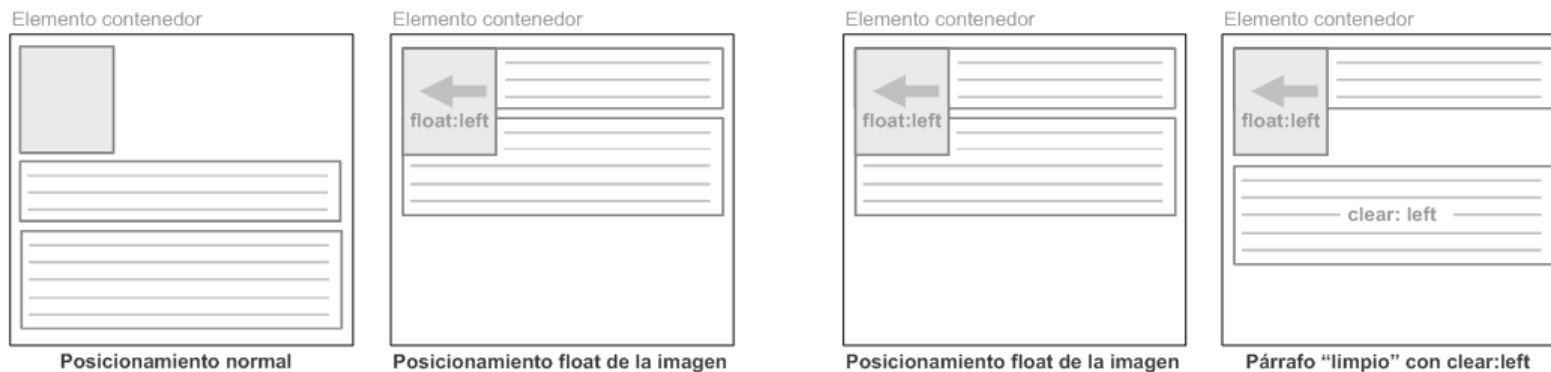
2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.4 EL MODELO DE CAJAS



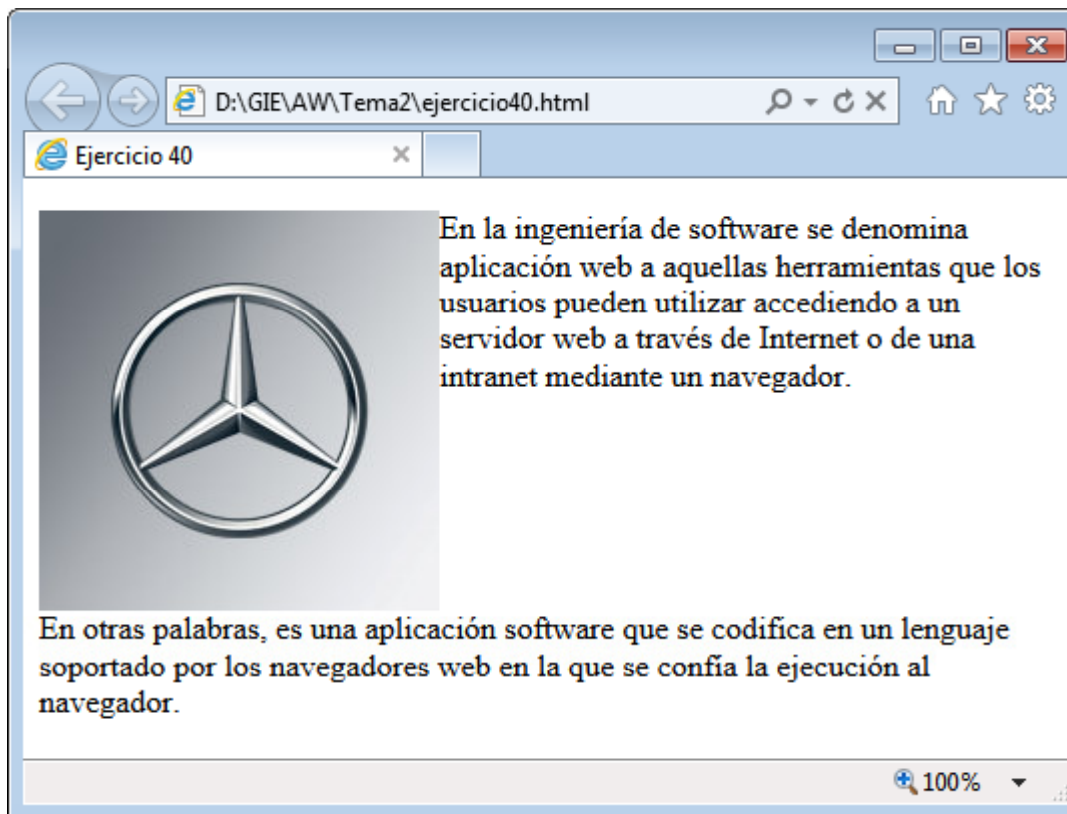
➤ Posicionamiento de las cajas (8)

- ✓ Uno de los principales motivos para la creación del posicionamiento flotante fue la posibilidad de colocar imágenes alrededor de las cuales fluye el texto. Así:
 - Los elementos que se encuentran alrededor de una caja flotante adaptan sus **contenidos** para que fluyan alrededor del elemento posicionado.
 - Si se desea que el contenido de algún elemento no fluya alrededor de una caja flotante, debe utilizarse la propiedad **clear** que permite modificar el comportamiento por defecto del posicionamiento flotante para forzar a que el elemento correspondiente se muestre bajo la caja flotante:



➤ Posicionamiento de las cajas (9)

- ✓ Utilizar la propiedad **float** de las cajas para dar formato a la siguiente web:



con float y clear



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.4 EL MODELO DE CAJAS



➤ Posicionamiento de las cajas (10)

- Con `clear: left` la caja se posiciona debajo de cualquier caja que flote a la izquierda que aparezca antes en el documento.
- Con `clear: right` la caja se posiciona debajo de cualquier caja que flote a la derecha que aparezca antes en el documento.
- Con `clear: both` la caja se posiciona debajo de cualquier caja que flote que aparezca antes en el documento.
- Como se verá más adelante, la propiedad `clear` es imprescindible cuando se crean las estructuras de las páginas web complejas.

```
#contenedor {  
    border: 1px solid #CCC;  
    background-color: #E0E0E0;  
    padding: .5em;  
}  
  
.anterior{ float: left; }  
.siguiente{ float: right; }  
.debajo{ clear: both; }
```

Elemento contenedor



Los elementos posicionados vacían de contenidos al <div>

Elemento contenedor

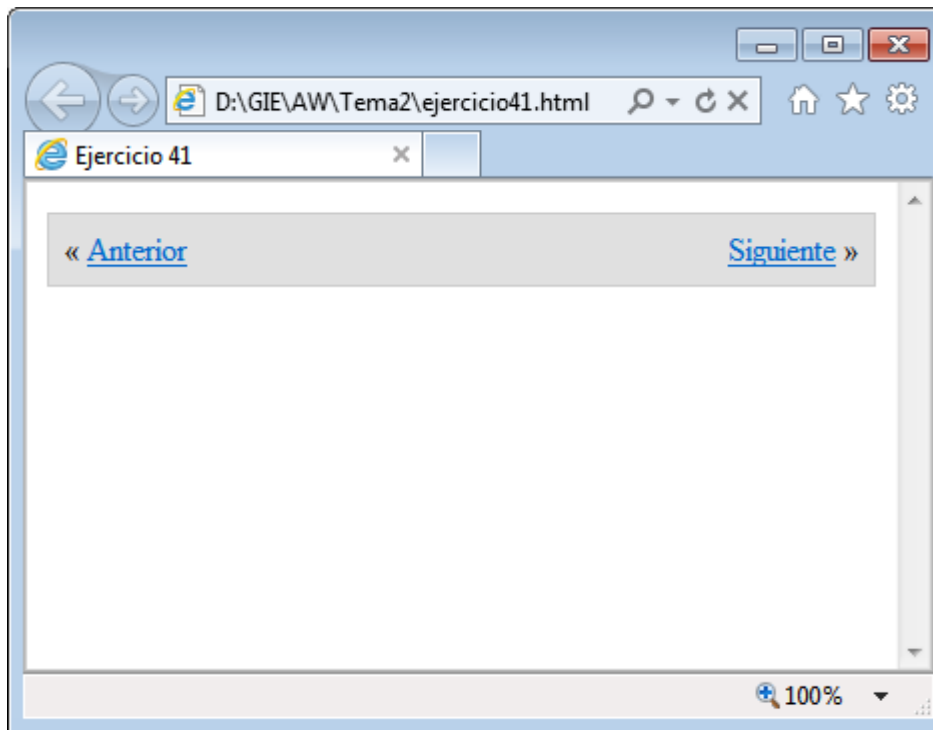


El <div> añadido "limpia" el float y fuerza la altura del <div> original



➤ Posicionamiento de las cajas (11)

- ✓ Utilizar la propiedad **clear** de las cajas para dar formato a la siguiente web:





2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.4 EL MODELO DE CAJAS



➤ Visualización (1)

- ✓ 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**.
- ✓ La propiedad **display** modifica la forma en la que se visualiza un elemento.
 - El valor **block** muestra un elemento como si fuera de bloque, y el valor **inline** visualiza un elemento en forma de elemento en línea, independientemente del tipo de elemento que se trate.

DIV normal

DIV con display:inline

[Enlace normal](#)

[Enlace con display:block](#)



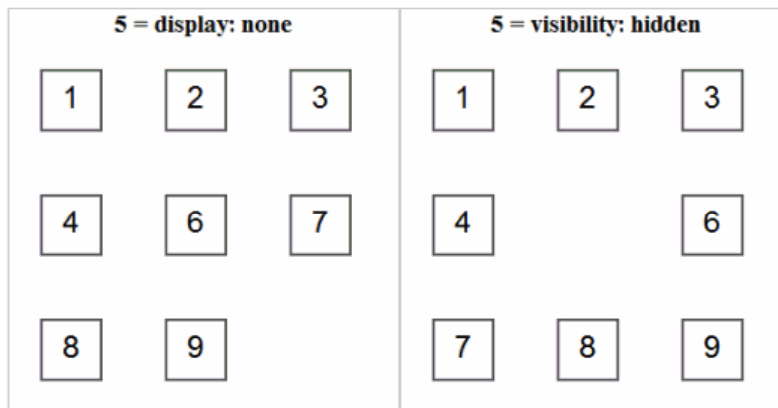
2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.4 EL MODELO DE CAJAS



➤ Visualización (2)

- El valor **none** oculta un elemento y hace que desaparezca de la página. El resto de elementos de la página se visualizan como si no existiera el elemento oculto, es decir, pueden ocupar el espacio en el que se debería visualizar el elemento.
- ✓ Por otra parte, la propiedad **visibility: hidden** 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, ya que aunque la caja no se ve, sigue ocupando sitio.





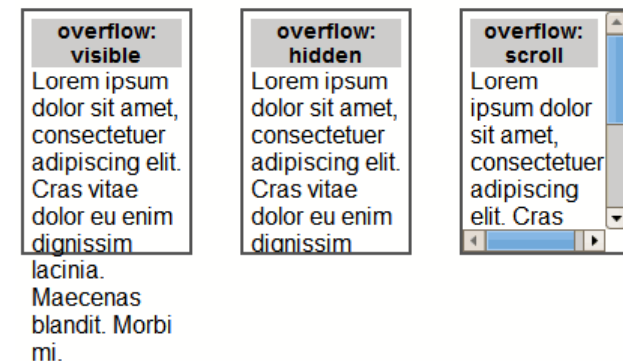
2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.4 EL MODELO DE CAJAS



➤ Visualización (3)

- ✓ CSS define la propiedad **overflow** para controlar la forma en la que se visualizan los contenidos que sobresalen de sus elementos.
 - **visible**: el contenido no se corta y se muestra sobresaliendo la zona reservada para visualizar el elemento. Este es el comportamiento por defecto.
 - **hidden**: el contenido sobrante se oculta y sólo se visualiza la parte del contenido que cabe dentro de la zona reservada para el elemento.
 - **scroll**: solamente se visualiza el contenido que cabe dentro de la zona reservada para el elemento, pero también se muestran barras de scroll que permiten visualizar el resto del contenido.
 - **auto**: el comportamiento depende del navegador, aunque normalmente es el mismo que la propiedad scroll.





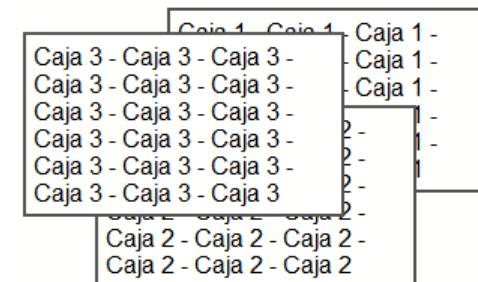
2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.4 EL MODELO DE CAJAS



➤ Visualización (4)

- ✓ Además de posicionar una caja de forma horizontal y vertical, 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.
- ✓ La posición tridimensional de un elemento se establece sobre un tercer eje llamado “Z” y se controla mediante la propiedad **z-index**. Utilizando esta propiedad es posible crear páginas complejas con varios niveles o capas.
- ✓ Cuanto más alto sea el valor numérico, más cerca del usuario se muestra la caja.
- ✓ La propiedad **z-index** sólo tiene efecto en los elementos posicionados, por lo que es obligatorio que la propiedad **z-index** vaya acompañada de la propiedad **position**.





2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.4 EL MODELO DE CAJAS

EJERCICIO 42



➤ Visualización (5)

- ✓ Utilizar la propiedad **z-index** de las cajas para dar formato a la siguiente web:





2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.4 EL MODELO DE CAJAS



➤ Diseño de Layouts (1)

- ✓ El diseño de las páginas web habituales se divide en bloques: cabecera, menú, contenidos y pie de página.
- ✓ Visualmente, los bloques se disponen en varias filas y columnas. Por este motivo, hace varios años la estructura de las páginas HTML se definía mediante tablas o marcos.
- ✓ El desarrollo de CSS ha permitido que se puedan realizar los mismos diseños sin utilizar tablas HTML. Las principales ventajas de diseñar la estructura de las páginas web con CSS en vez de con tablas HTML son el mantenimiento, la accesibilidad, la velocidad de carga y la semántica.
- ✓ Por estos motivos, la estructura basada en tablas ha dado paso a la estructura basada exclusivamente en CSS. Aunque crear la estructura de las páginas sólo con CSS presenta en ocasiones retos importantes, en general es más sencilla y flexible.

➤ Diseño de Layouts (2): Centrado horizontal

- ✓ El **centrado horizontal** del diseño de una web se suele utilizar para evitar las situaciones siguientes:
 - Con resoluciones superiores a 1024 x 768 píxel, las líneas de texto son demasiado largas como para leerlas con comodidad.
 - Aunque se opte por un diseño de anchura fija, los navegadores alinean por defecto las páginas web a la izquierda de la ventana, provocando una sensación de vacío a la derecha.

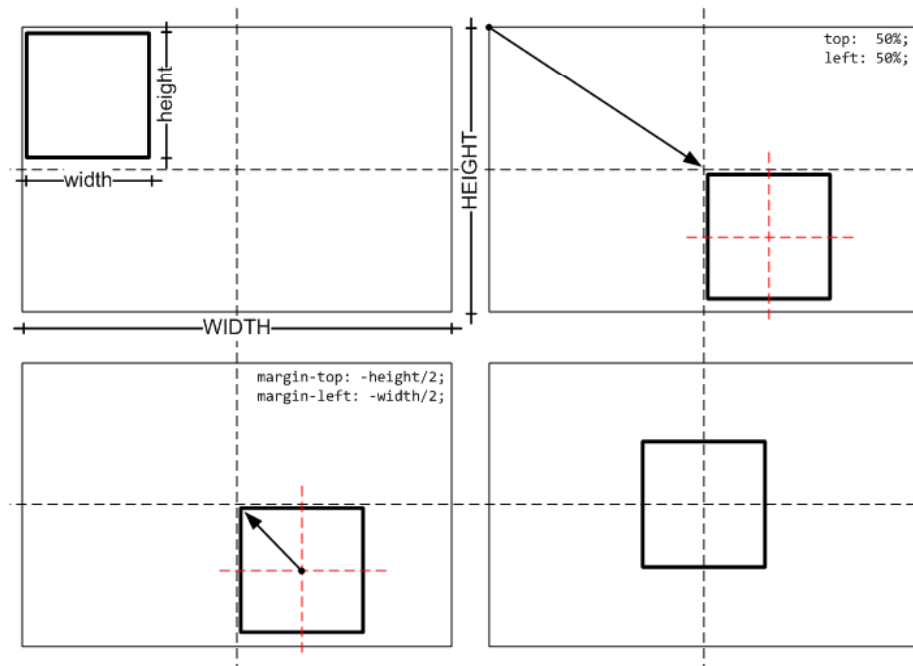
```
#contenedor {  
    width: 300px; (ó 70%)  
    margin: 0 auto 0 auto;  
}  
  
<body>  
    <div id="contenedor">  
        <h1>Lorem ipsum...</h1>  
        <p>Lorem ...</p>  
    </div>  
</body>
```



➤ Diseño de Layouts (3): Centrado vertical

- ✓ El **centrado vertical** del diseño de una web no suele utilizarse dado que el contenido de una web suele ser mayor que la altura del navegador:
 - A continuación se muestra la forma de centrar una página web respecto de la ventana del navegador, es decir, centrarla tanto horizontalmente como verticalmente.

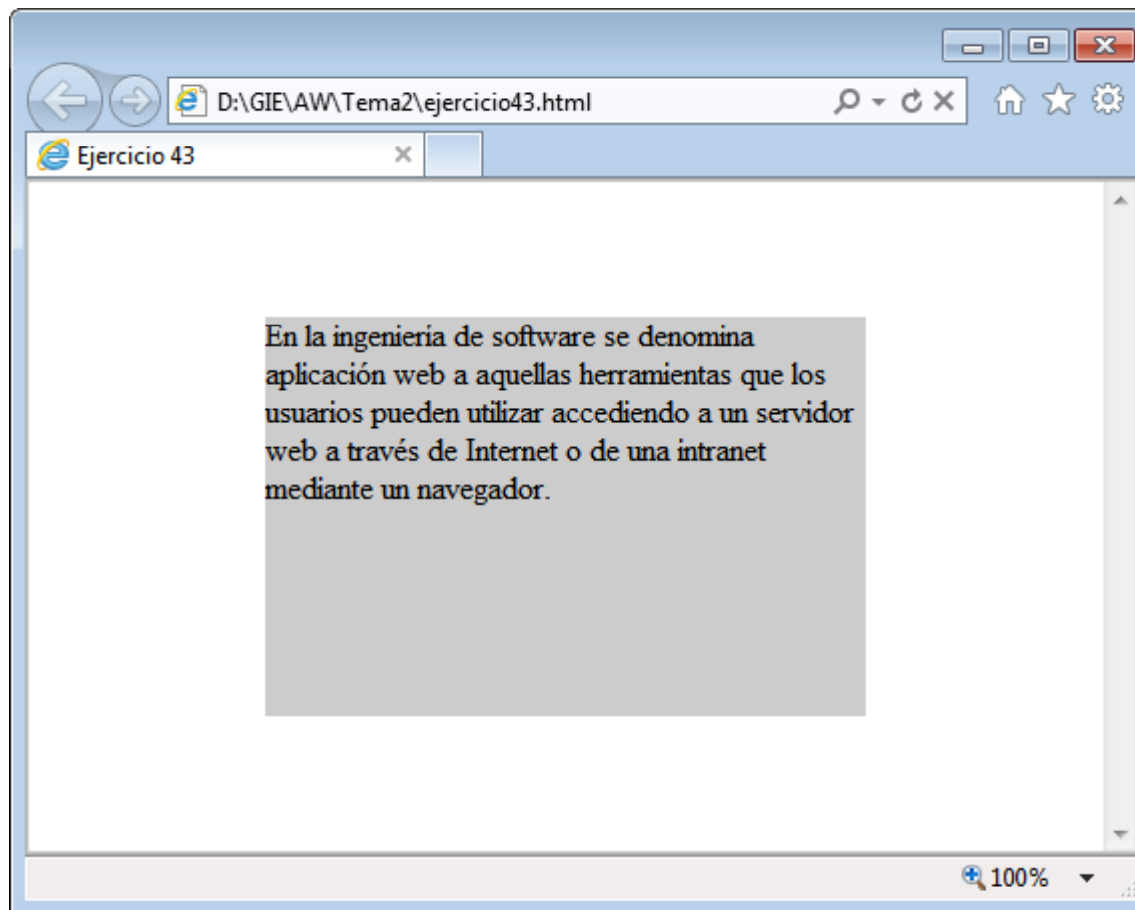
```
#contenedor {  
    width: 300px;  
    height: 200px;  
  
    position: absolute;  
    top: 50%;  
    left: 50%;  
  
    margin-left: -150px;  
    margin-top: -100px;  
}
```





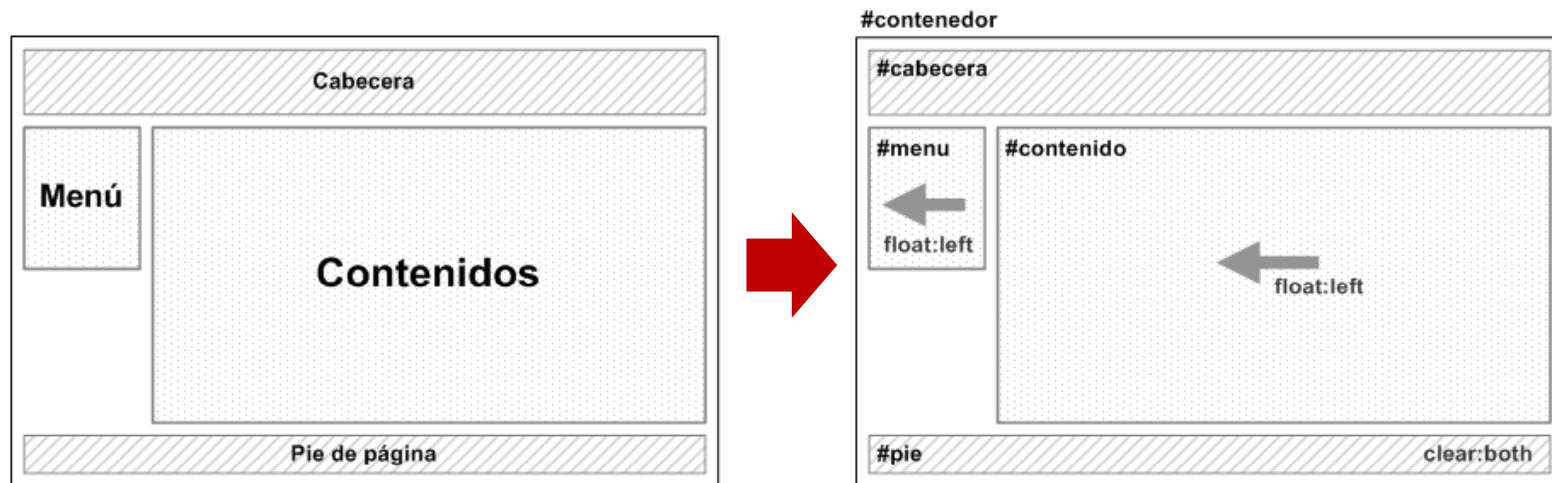
➤ Diseño de Layouts (4): Ejemplo de centrado

- ✓ Utilizar las propiedades de centrado para dar formato a la siguiente web:



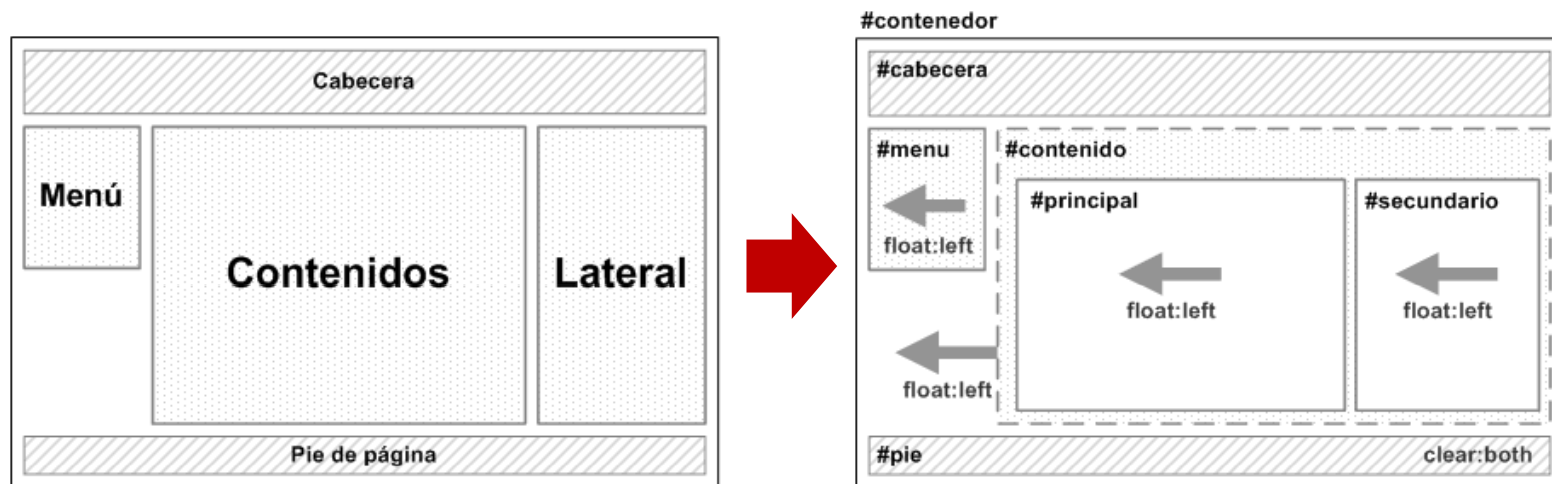
➤ Diseño de Layouts (5): Diseño a dos columnas

- ✓ 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 del menú en 150px y la de los contenidos en 550px:
 - La solución CSS se basa en el uso de la propiedad **float** para los elementos posicionados como el menú y los contenidos, y el uso de la propiedad **clear** en el pie de página para evitar los solapamientos ocasionados por los elementos posicionados con **float**.



➤ Diseño de Layouts (6): Diseño a tres columnas

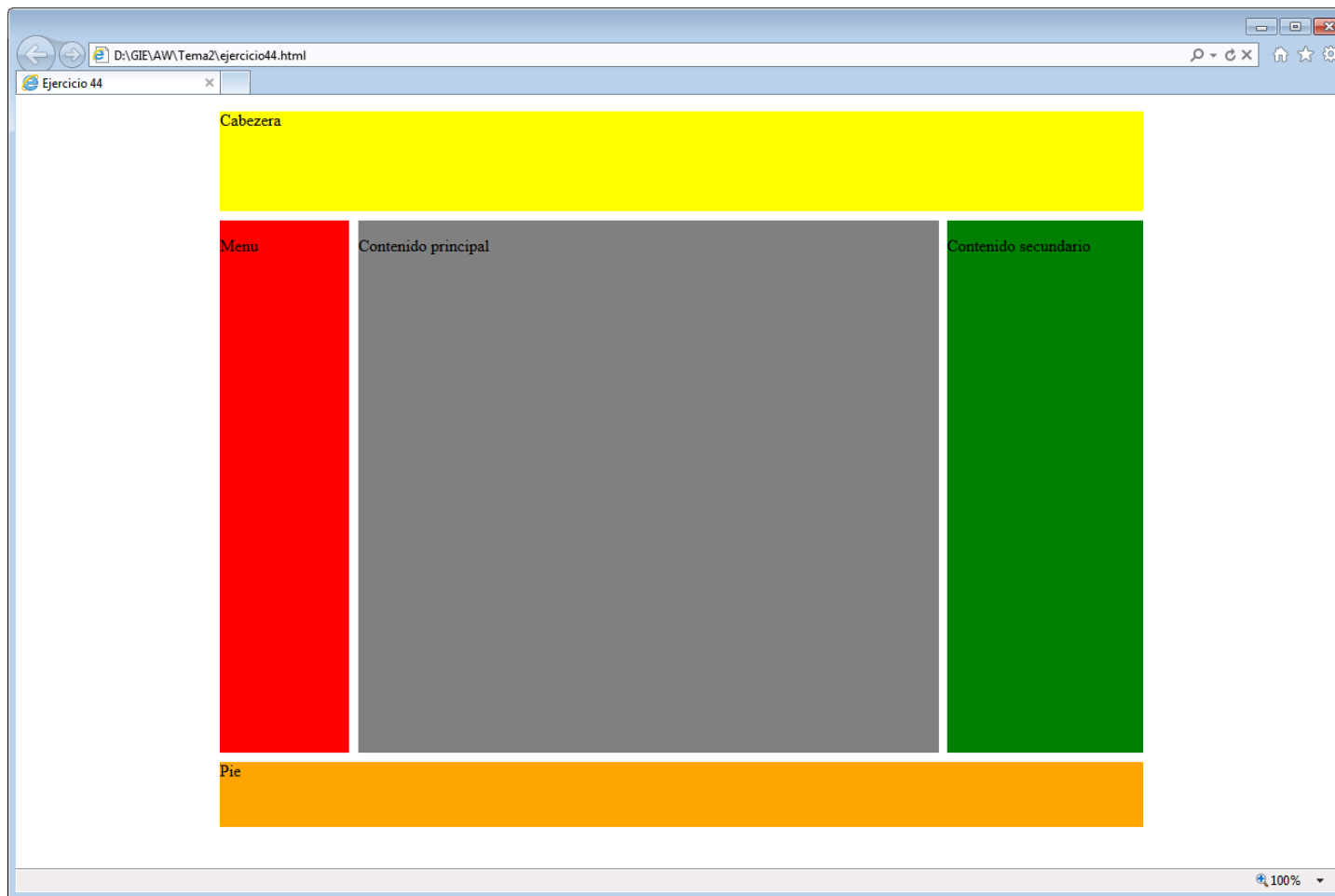
- ✓ Además del diseño a dos columnas, el diseño más utilizado es el de **tres columnas** con cabecera y pie de página.
 - En este caso, los contenidos se dividen en dos zonas diferenciadas: zona principal de contenidos y zona lateral de contenidos auxiliares.
 - La solución CSS emplea la misma estrategia del diseño a dos columnas y se basa en utilizar las propiedades **float** y **clear**.





➤ Diseño de Layouts (7): Diseño a tres columnas

- ✓ Generar un diseño a tres columnas similar al siguiente:



Índice: Tema 2.4 (Parte 2)

2.4 Hojas de estilo en cascada CSS

2.4.1 El lenguaje XHTML

2.4.2 Hojas de estilo CSS

2.4.3 Reglas de estilo

2.4.4 El modelo de cajas

2.4.5 Propiedades de los elementos básicos



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.5 PROPIEDADES DE LOS ELEMENTOS BÁSICOS



➤ Propiedades de texto (1)

✓ CSS define numerosas propiedades para modificar la apariencia del **texto**:

- Color: `color: blue`
- Tipo de letra: `font-family: Arial`
- Tamaño de letra: `font-size: 12pt`
- Anchura de letra: `font-weight: bold`
- Estilo de letra: `font-style: italic`
- Estilo alternativo de letra: `font-variant: small-caps`
- Todas las propiedades: `font: blue, Arial, 12pt, etc...`
- Alineado de texto: `text-align: left`
- Interlineado: `line-height: 6pt`
- Decoración de texto: `text-decoration: underline`
- Transformación de texto: `text-transform: uppercase`
- Alineación vertical: `vertical-align: top`
- Tabulación de texto: `text-indent: 12pt`
- Espaciado entre letras: `letter-spacing: normal`
- Espaciado entre palabras: `word-spacing: inherit`
- Uso de espacios en blanco: `white-spacing: nowrap`



2.4 HOJAS DE ESTILO EN CASCADA CSS

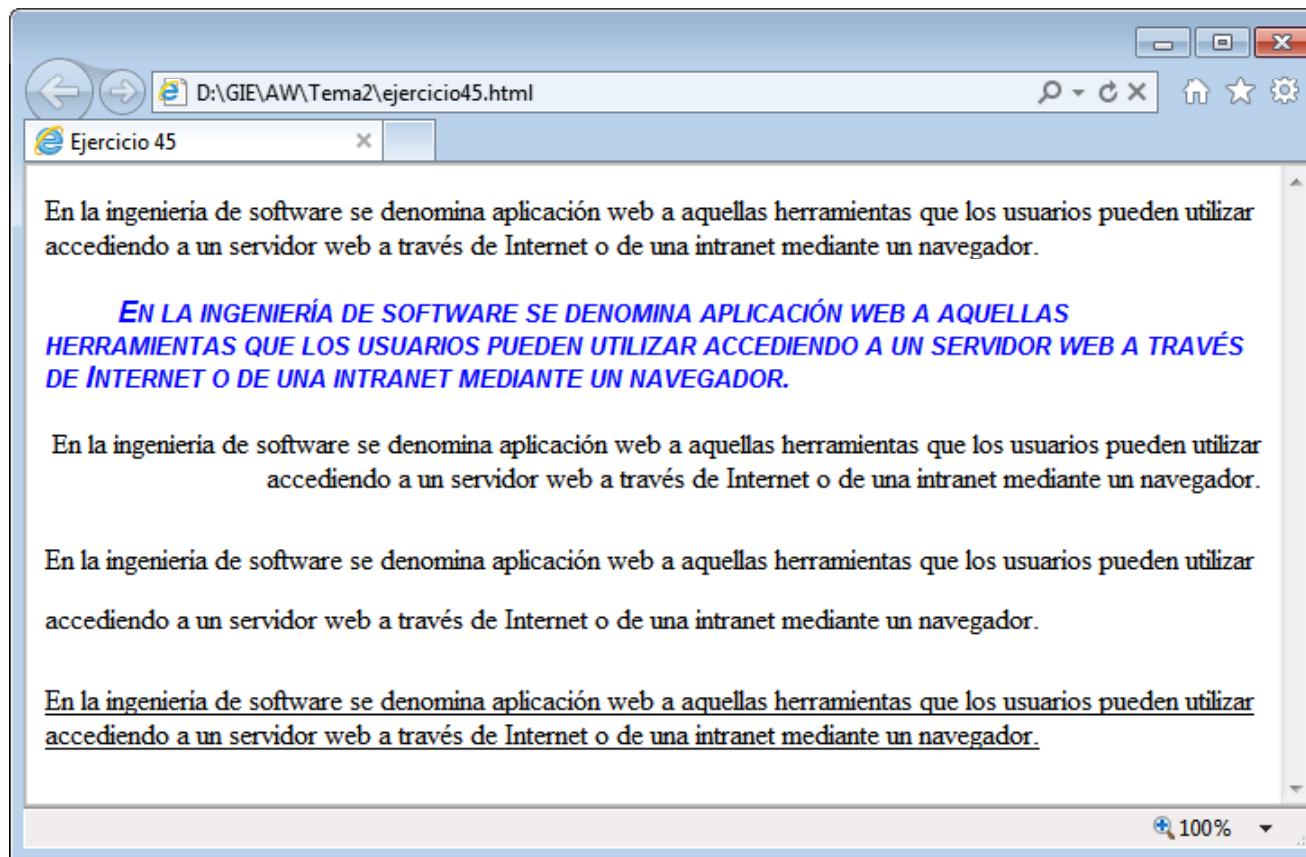
2.4.5 PROPIEDADES DE LOS ELEMENTOS BÁSICOS

EJERCICIO 45



➤ Propiedades de texto (2)

- ✓ Utilizar las propiedades de texto para generar las reglas con las que dar formato a los siguientes párrafos:





2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.5 PROPIEDADES DE LOS ELEMENTOS BÁSICOS

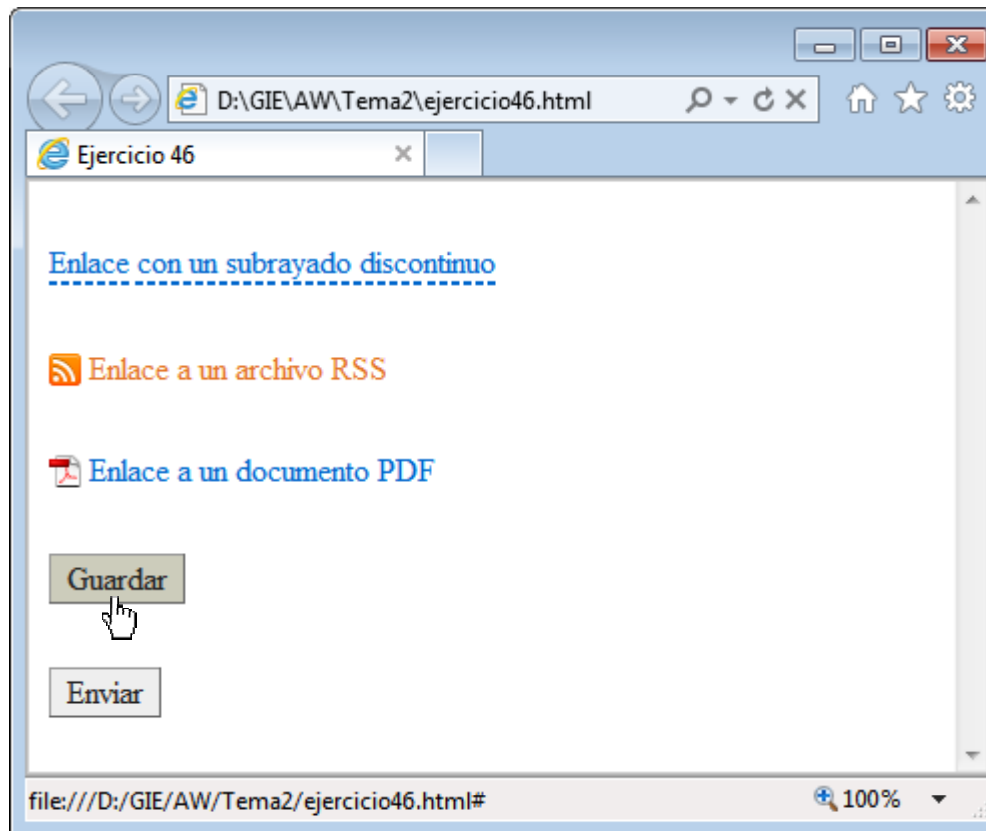


➤ Propiedades de enlaces (1)

- ✓ 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. Así, se tienen por ejemplo las siguientes propiedades para modificar el estilo original de los enlaces:
 - Utilizando la propiedad **border-bottom** es posible añadir cualquier tipo de subrayado para los enlaces de las páginas.
 - Utilizando la propiedad **background** se puede incluir un pequeño icono al lado de un enlace para indicar el tipo de contenido que enlaza.
 - Las propiedades definidas por CSS permiten mostrar los enlaces con el aspecto de un **botón**, lo que puede ser útil en formularios en los que no se utilizan elementos específicos para crear botones.
 - CSS también permite aplicar diferentes estilos a un mismo enlace en función de su estado mediante el uso de **pseudo-clases** (botón activado o no).

➤ Propiedades de enlaces (2)

- ✓ Utilizar las propiedades de los enlaces para generar las reglas con las que dar formato a los siguientes enlaces:





2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.5 PROPIEDADES DE LOS ELEMENTOS BÁSICOS



➤ Propiedades de imágenes (1)

- ✓ CSS y CSS3 contemplan diversas propiedades que pueden ayudar a mejorar el aspecto de las imágenes originales. Así, se tienen por ejemplo las siguientes:

- Las dimensiones de la imagen en pantalla se pueden modificar a través de su altura y anchura:

```
img { width: 100px; }
```

- Se pueden aplicar las propiedades de los bordes para poner las esquinas redondeadas:

```
img { border-radius: 25px; }
```

- El nivel de transparencia de una imagen se puede controlar a través de la opacidad:

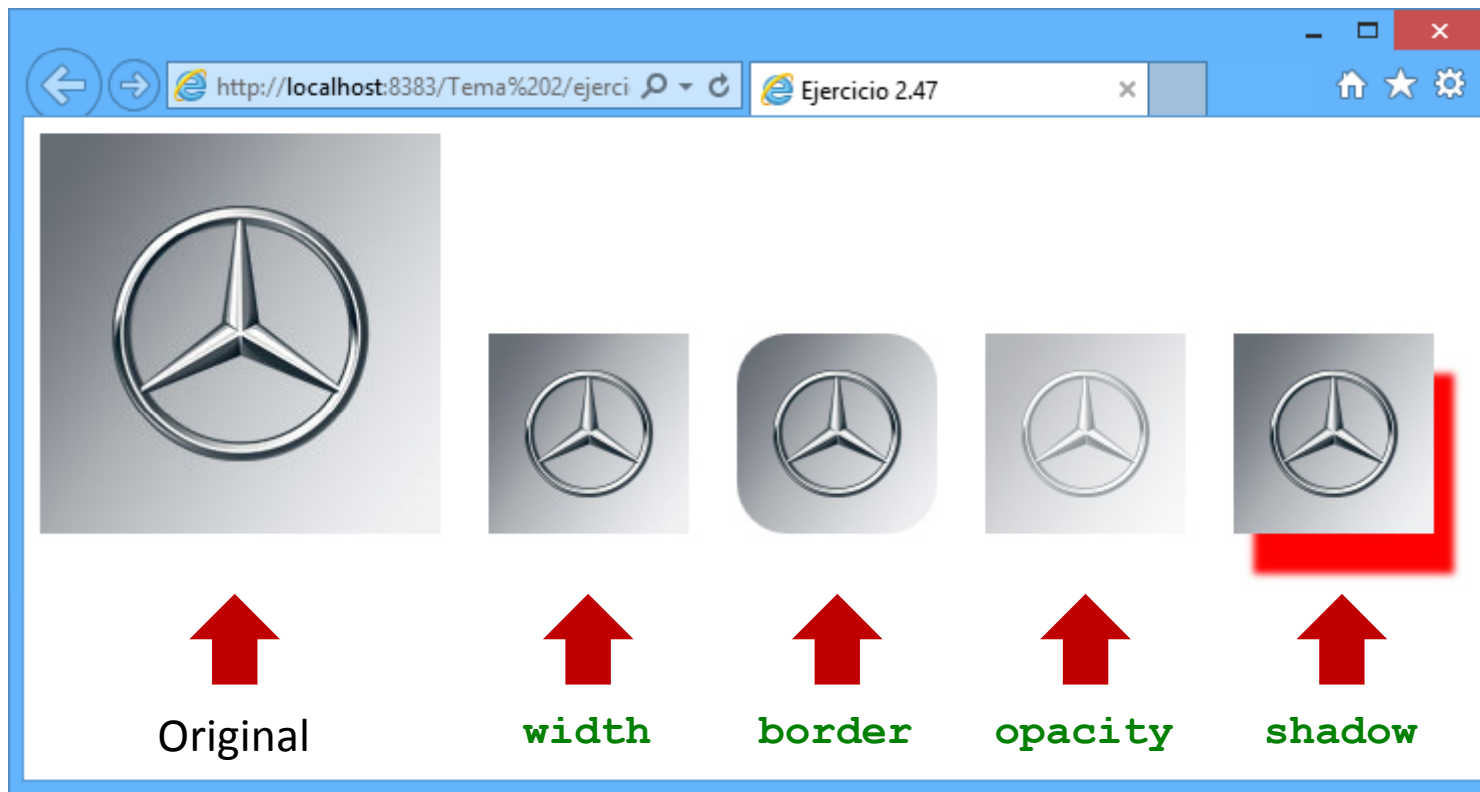
```
img { opacity: 0.5; }
```

- La propiedad **box-shadow** de CSS3 permite añadir un efecto de sombra sobre una imagen:

```
img { border-radius: 10px 20px 5px red; }
```

➤ Propiedades de imágenes (2)

- ✓ Utilizar las propiedades de imágenes para generar las reglas con las que dar formato a la siguiente web:





2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.5 PROPIEDADES DE LOS ELEMENTOS BÁSICOS



➤ Propiedades de listas (1)

- ✓ CSS define numerosas propiedades para modificar la apariencia de las **listas**:

- Tipo de viñeta: `list-style-type: disc`
- Posición de la viñeta: `list-style-position: inside`
- Imagen de la viñeta: `list-style-image: <url>`
- Estilo global de la lista: `list-style: disc, inside, <url>`

- ✓ Las listas HTML se suelen emplear, además de para su función natural, para la creación de **menús de navegación verticales y horizontales**. Para verlo, supóngase la siguiente lista:

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

- [Elemento 1](#)
- [Elemento 2](#)
- [Elemento 3](#)
- [Elemento 4](#)





➤ Propiedades de listas (2)

- ✓ El proceso de transformación de la lista en un **menú vertical** requiere de los siguientes pasos:
 - Definir la anchura del menú.
 - Eliminar las viñetas automáticas y todos los márgenes y espaciados aplicados por defecto.
 - Añadir un borde al menú de navegación y establecer el color de fondo y los bordes de cada elemento del menú.
 - Aplicar estilos a los enlaces mostrándolos 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>



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.5 PROPIEDADES DE LOS ELEMENTOS BÁSICOS



➤ Propiedades de listas (3)

- ✓ A partir de un menú vertical sencillo, es posible crear un **menú horizontal** sencillo aplicando las propiedades CSS conocidas hasta el momento.
- ✓ El proceso de transformación del menú vertical en un menú horizontal consta de los siguientes pasos:
 - Eliminar la anchura y el borde de `` y aplicar las propiedades `float` y `clear`.
 - Establecer el valor de la propiedad `float` de los `` del menú a `left`.
 - Modificar el padding y los bordes de los enlaces que forman el menú.
 - Añadir un borde izquierdo en el elemento `` para homogeneizar el aspecto de los elementos del menú.
- ✓ Algunos ejemplos de menús horizontales pueden encontrarse en la página:
 - <http://exploding-boy.com/images/cssmenus/menus.html>



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.5 PROPIEDADES DE LOS ELEMENTOS BÁSICOS

EJERCICIO 48



➤ Propiedades de listas (4)

- ✓ Probar los estilos de las páginas sugeridas para generar diferentes menús verticales y horizontales:

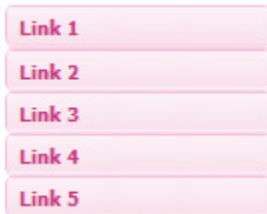
Tab Menu 11



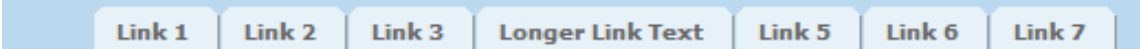
Tab Menu 12



Tab Menu 13



Tab Menu



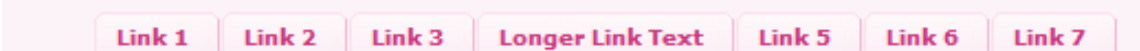
Tab Menu B



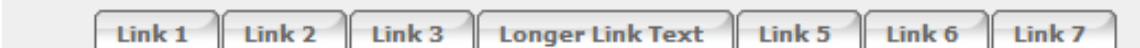
Tab Menu C



Tab Menu D



Tab Menu F





2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.5 PROPIEDADES DE LOS ELEMENTOS BÁSICOS

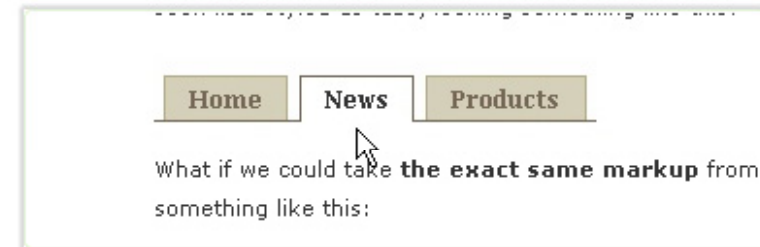


➤ Propiedades de listas (5): Menús avanzados

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



Drop-Down Menus, Horizontal Style // ALA



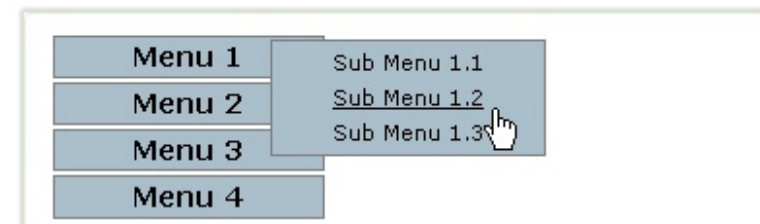
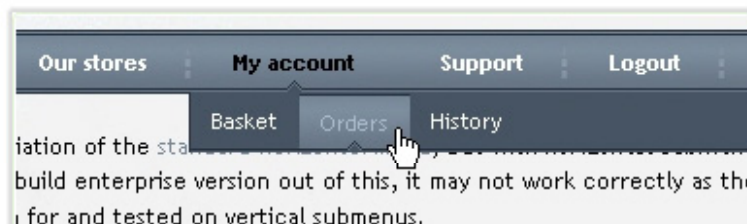
Sliding Doors // ALA



Taming Lists // ALA



Bulletproof Slants





2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.5 PROPIEDADES DE LOS ELEMENTOS BÁSICOS



➤ Propiedades de tablas (1)

- ✓ Cuando se aplican bordes a las celdas de una **tabla**, el aspecto por defecto con el que se muestra en un navegador es el siguiente:

| 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 |

- ✓ El estándar CSS define dos modelos diferentes para el tratamiento de los bordes de las celdas:

- Fusión de bordes:

border-collapse: collapse

| 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 |

- Espaciado entre bordes:

border-spacing: 10px

| 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 |

➤ Propiedades de tablas (2)

- ✓ CSS define otras propiedades específicas para el control del aspecto de las tablas:
 - Una de ellas es el tratamiento que reciben las celdas vacías de una tabla, que se controla mediante la propiedad `empty-cells` (`show`, `hide`). Esta propiedad sólo se aplica cuando el modelo de bordes de la tabla (`border-collapse`) es de tipo `separate`.
 - Se puede establecer un título para la tabla usando el elemento `<caption>` dentro de `<table>`. La propiedad `caption-side` (`top`, `bottom`) permite controlar la posición del título de la tabla (que por defecto se muestra encima de la tabla).

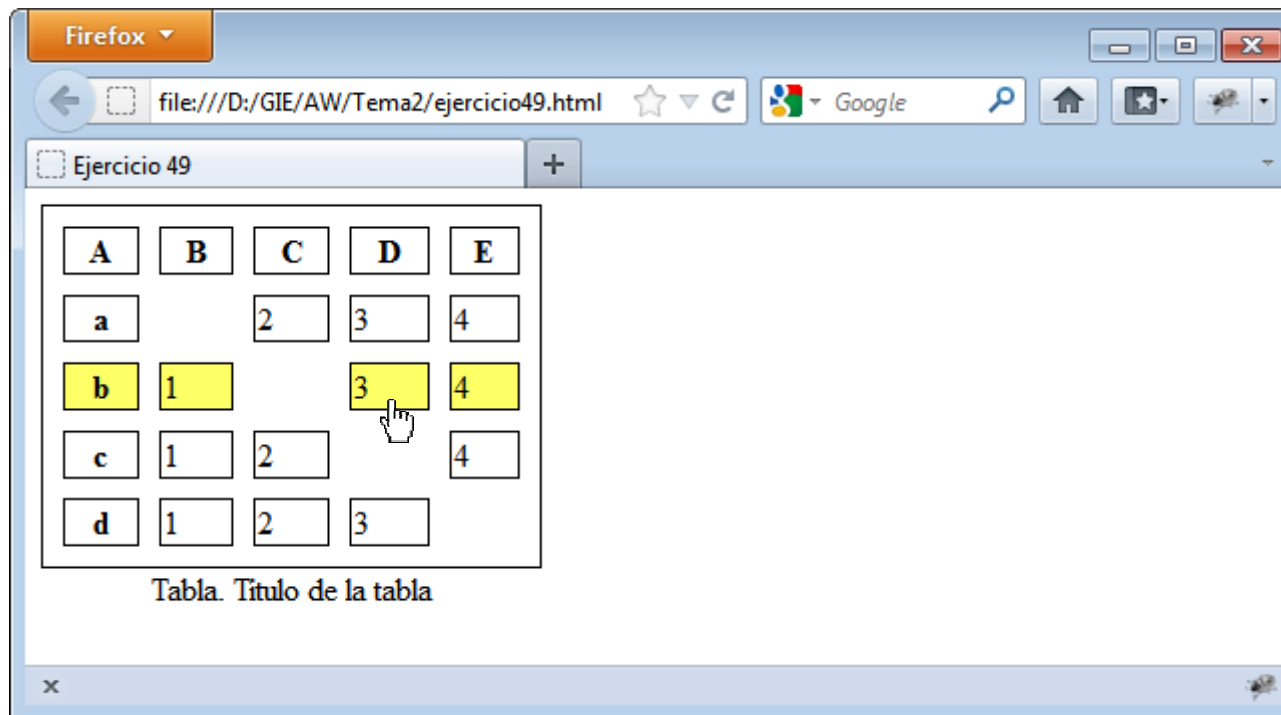
```
.tabla {  
    width: 250px;  
    border: 1px solid #000;  
    border-collapse: separate;  
    empty-cells: hide;  
    caption-side: bottom;  
}  
.tabla th, .tabla td {  
    border: 1px solid #000;  
}
```

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

Tabla. Título de la tabla

➤ Propiedades de tablas (3)

- ✓ Las pseudo-clases también funcionan con los elementos `<tr>` de las tablas.
- ✓ Por ejemplo, se puede utilizar dicha pseudo-clase como selector de las filas, para que estas cambien de color al pasar el ratón por encima:



➤ Propiedades de formularios (1)

- ✓ Así como el estilo por defecto de los enlaces se puede modificar para que se muestren como botones de **formulario**, los botones de formulario también se pueden modificar para que parezcan enlaces.

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

```
<input type="button" value="Boton normal" />
```

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



Botón normal

Botón como enlace

Formulario sin padding en los input

Nombre

lorem ipsum

Contraseña

Formulario con padding en los input

Nombre

lorem ipsum

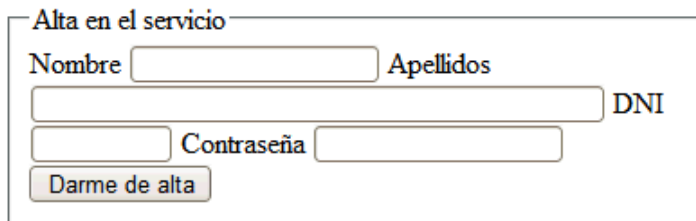
Contraseña



- ✓ Añadiendo un pequeño **padding** a cada elemento **<input>**, se mejora notablemente el aspecto del formulario.

➤ Propiedades de formularios (2)

- ✓ Utilizando la propiedad **display** se pueden convertir los elementos de línea en elementos de bloque para conseguir el siguiente aspecto sin insertar **
**:

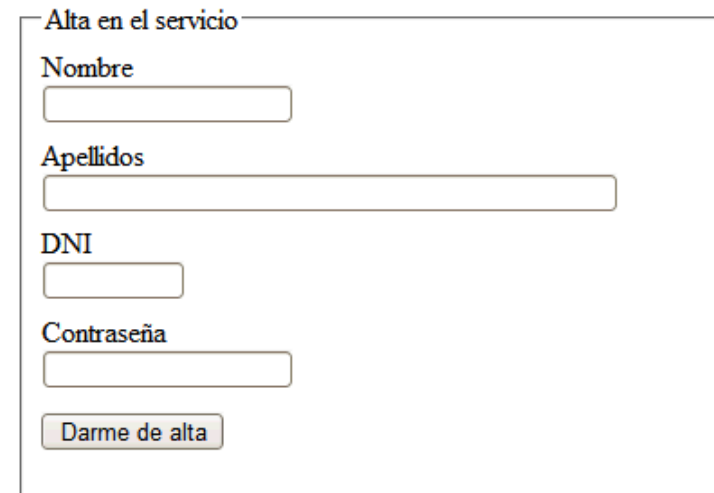


Alta en el servicio

Nombre Apellidos

DNI

Contraseña



Alta en el servicio

Nombre

Apellidos

DNI

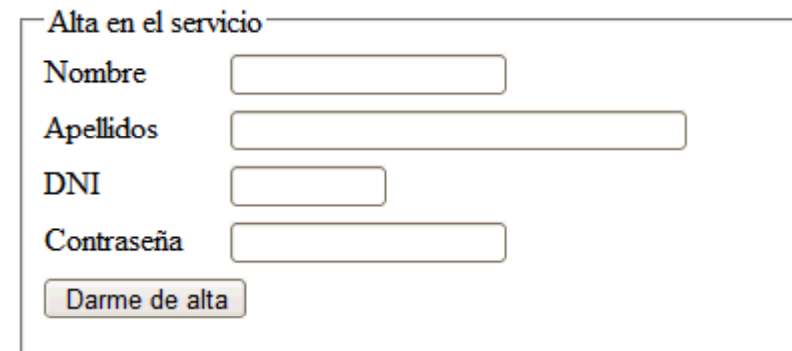
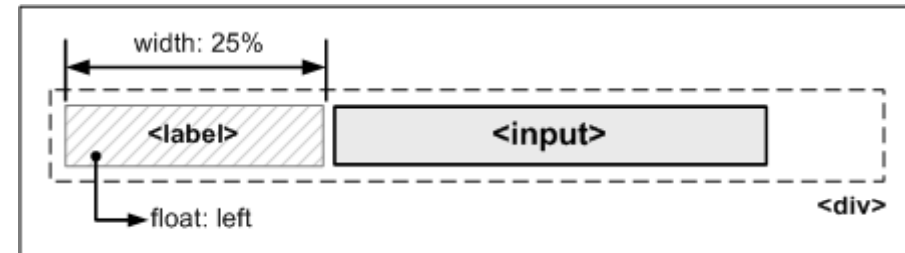
Contraseña

```
.btn, label {  
    display: block;  
    margin: 1em 0 0 0;  
}  
  
<label for="nombre">Nombre</label>  
<input type="text" id="nombre" />  
...  
<input class="btn" type="submit" value="Dar de alta" />
```


➤ Propiedades de formularios (3)

- ✓ Insertando cada pareja `<label>`-`<input>` dentro de un `<div>` se pueden alinear las etiqueta con los campos del formulario:

```
div {  
    margin: 0.4em;  
}  
  
div label {  
    float: left;  
    width: 25%;  
}
```



Alta en el servicio

Nombre

Apellidos

DNI

Contraseña

➤ Propiedades de formularios (4)

- ✓ Haciendo flotantes las cajas de los **<fieldset>** de los formularios pueden conseguirse diseños a dos (o más) columnas para simplificar la visualización de formularios complejos con decenas de opciones que pueden agruparse:

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



| Alta en el servicio | Datos de contacto |
|--|---------------------------------------|
| Nombre <input type="text"/> | Telefono <input type="text"/> |
| Apellidos <input type="text"/> | Email <input type="text"/> |
| DNI <input type="text"/> | Dirección <input type="text"/> |
| Contraseña <input type="password"/> | Código Postal <input type="text"/> |
| <input type="button" value="Dar de alta"/> | |

- ✓ Finalmente, la pseudo-clase **:focus** puede utilizarse como selector en un **input** para resaltar el campo de un formulario cuando el usuario activa el elemento.



2.4 HOJAS DE ESTILO EN CASCADA CSS

2.4.5 PROPIEDADES DE LOS ELEMENTOS BÁSICOS

EJERCICIO 50



➤ Propiedades de formularios (5)

- ✓ Utilizar las propiedades de los formularios para generar las reglas con las que dar formato a los siguientes formularios:

Firefox

file:///D:/GIE/AW/Tema2/ejercicio50.html

Ejercicio 50

Alta en el servicio

Nombre

Apellidos

DNI

Contraseña

Dar de alta

Datos de contacto

Teléfono

Email

Dirección

Código postal