

MODELO DE CAJA TRADICIONAL

- Todo comenzó con tablas.
- Las tablas fueron los elementos que sin intención se volvieron la herramienta ideal utilizada por desarrolladores para crear y organizar cajas de contenido en la pantalla.
- Este puede ser considerado el primer modelo de caja de la web. Las cajas eran creadas expandiendo celdas y combinando filas de celdas, columnas de celdas y tablas enteras, unas sobre otras o incluso anidadas.
- Cuando los sitios webs crecieron y se volvieron más y más complejos esta práctica comenzó a presentar serios problemas relacionados con el tamaño y el mantenimiento del código necesario para crearlos.
- Usando etiquetas `<div>` y estilos CSS fue posible reemplazar la función de tablas y efectivamente separar la estructura HTML de la presentación.

ACTUALMENTE...

```
<!DOCTYPE html>
<html lang="es">
  <head>
    <meta name="description" content="Ejemplo de HTML5">
    <title>Este texto es el título del documento</title>
    <link rel="stylesheet" href="misestilos.css">
  </head>
  <body>
    <div id="agrupar">
      <header id="cabecera">
        <h1>Este es el título principal del sitio web</h1>
      </header>
      <nav id="menu">
        <ul>
          <li>principal</li>
          <li>fotos</li>
          <li>videos</li>
          <li>contacto</li>
        </ul>
      </nav>
      <section id="seccion">
        <article>
          <header>
            <hgroup>
              <h1>Título del mensaje uno</h1>
              <h2>Subtítulo del mensaje uno</h2>
            </hgroup>
            <p>publicado 10-12-2011</p>
          </header>
          Este es el texto de mi primer mensaje
```

```
        <figure>
          
          <figcaption>
            Esta es la imagen del primer mensaje
          </figcaption>
        </figure>
        <footer>
          <p>comentarios (0)</p>
        </footer>
      </article>
    </article>
    <header>
      <hgroup>
        <h1>Título del mensaje dos</h1>
        <h2>Subtítulo del mensaje dos</h2>
      </hgroup>
      <p>publicado 15-12-2011</p>
    </header>
    Este es el texto de mi segundo mensaje
    <footer>
      <p>comentarios (0)</p>
    </footer>
  </section>
  <aside id="columna">
    <blockquote>Mensaje número uno</blockquote>
    <blockquote>Mensaje número dos</blockquote>
  </aside>
  <footer id="pie">
    Derechos Reservados &copy; 2010-2011
  </footer>
</div>
</body>
</html>
```

DOS CAMBIOS

- El primero es que ahora varias etiquetas fueron identificadas con los atributos **id** y **class**. Esto significa que podemos referenciar un elemento específico desde las reglas CSS con el valor de su atributo **id** o podemos modificar varios elementos al mismo tiempo usando el valor de su atributo **class**.
- El segundo es la adición del elemento **<div>** mencionado anteriormente. Este **<div>** fue identificado con el atributo y el valor **id="agrupar"**, y es cerrado al final del cuerpo con la etiqueta de cierre **</div>**. Este elemento se encarga de agrupar todos los demás elementos permitiéndonos aplicar el modelo de caja al cuerpo y designar su posición horizontal, como veremos más adelante.



A TRABAJAR



3

UNA REGLA BASICA

- Algunos elementos por defecto tienen márgenes que son diferentes de cero y en la mayoría de los casos demasiado amplios.
- Para evitar el tener que repetir estilos constantemente, podemos utilizar el selector universal. La primera regla en nuestro archivo CSS, será asegurarnos que todo elemento tendrá un margen interno y externo de 0 pixeles. De ahora en más solo necesitaremos modificar los márgenes de los elementos que queremos que sean mayores que cero.

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

MARGIN

- Espacio alrededor del elemento, el que se encuentra por fuera del borde de esa caja.

```
margin: 0px 5px 6px 10px;
```

```
margin: 0px;
```

```
margin: 0px 5px;
```

```
margin-top: 0px;  
margin-left: 10px;  
margin-right: 40px;  
margin-bottom: 20px;
```

PADDING

- Espacio alrededor del contenido del elemento pero dentro de sus bordes, como el espacio entre el título y el borde de la caja virtual formada por el elemento `<h1>` que contiene ese título.

```
padding-top: 0px;  
padding-left: 10px;  
padding-right: 40px;  
padding-bottom: 20px;
```

```
padding: 0px 10px 20px 5px;
```

```
padding: 0px;
```

```
padding: 0px 20px;
```

OTRA REGLA BASICA

- Tenemos que declarar desde el comienzo es la definición por defecto de elementos estructurales de HTML5.
- Algunos navegadores aún no reconocen estos elementos o los tratan como elementos inline (en línea). Necesitamos declarar los nuevos elementos HTML5 como elementos block para asegurarnos de que serán tratados como regularmente se hace con elementos **<div>**

```
header, section, footer, aside, nav, article, figure, figcaption, hgroup{  
  display: block;  
}
```

- A partir de ahora, los elementos serán posicionados uno sobre otro a menos que especifiquemos algo diferente más adelante.

CENTRANDO EL CUERPO

- Por defecto, la etiqueta **<body>** tiene un valor de ancho establecido en 100%. Esto significa que el cuerpo ocupará el ancho completo de la ventana del navegador. Por lo tanto, para centrar la página en la pantalla necesitamos centrar el contenido dentro del cuerpo. Con lo siguiente todo lo que se encuentra dentro de **<body>** será centrado en la ventana, centrando de este modo toda la página web.



```
body {  
    text-align: center;  
}
```


CREANDO LA CAJA PRINCIPAL

- Estamos referenciando por primera vez un elemento a través del valor de su atributo **id**.
- El carácter **#** le está diciendo al navegador que el elemento afectado por este conjunto de estilos tiene el atributo **id** con el valor **agrupar**.

```
#agrupar {  
  width: 960px;  
  margin: 15px auto;  
  text-align: left;  
}
```

- El primer estilo establece un valor fijo de 960 pixeles. Esta caja tendrá siempre un ancho de 960 pixeles, lo que representa un valor común para un sitio web estos días (los valores se encuentran entre 960 y 980 pixeles de ancho).

```
body {  
  text-align: center;  
}
```

```
#agrupar {  
  width: 960px;  
  margin: 15px auto;  
  text-align: left;  
}
```

- **text-align: center** solo afecta contenido inline, como textos o imágenes.
- Para elementos block, como un **<div>**, necesitamos establecer un valor específico para sus márgenes que los adapta automáticamente al tamaño de su elemento padre.

```
body {  
    text-align: center;  
}
```

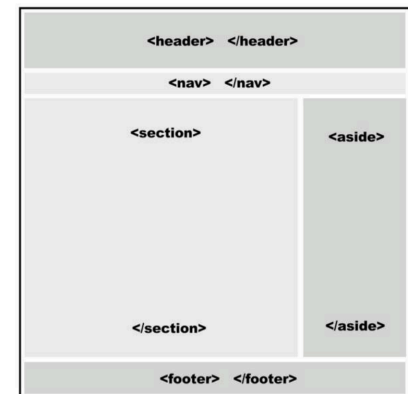
```
#agrupar {  
    width: 960px;  
    margin: 15px auto;  
    text-align: left;  
}
```

- Lo próximo que necesitamos hacer es prevenir un problema que ocurre en algunos navegadores.
- La propiedad **text-align** es hereditaria. Esto significa que todos los elementos dentro del cuerpo y su contenido serán centrados, no solo la caja principal. El estilo asignado a **<body>** será asignado a cada uno de sus hijos.
- Debemos retornar este estilo a su valor por defecto para el resto del documento. El tercer estilo logra este propósito. El resultado final es que el contenido del cuerpo es centrado pero el contenido de la caja principal (el **<div>** identificado como **agrupar**) es alineado nuevamente hacia la izquierda, por lo tanto todo el resto del código HTML dentro de esta caja hereda este estilo.

LA CABECERA

- Le otorgamos a `<header>` un fondo amarillo, un borde sólido de 1 pixel y un margen interior de 20 pixeles usando la propiedad **padding**.

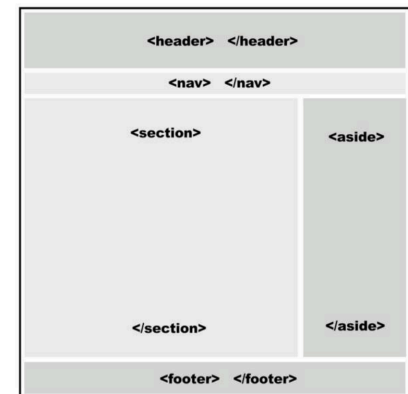
```
#cabecera {  
  background: #FFFB9;  
  border: 1px solid #999999;  
  padding: 20px;  
}
```



BARRA DE NAVEGACIÓN

- `<nav>` será el menu que tendra el propósito de proporcionar ayuda para la navegación.
- Este menú será una simple barra ubicada debajo de la cabecera.
- `<nav>` es un elemento block por lo que será ubicado debajo del elemento previo, su ancho por defecto será 100% por lo que será tan ancho como su padre (el `<div>` principal), y será tan alto como su contenido y los márgenes predeterminados. Por lo tanto, lo único que nos queda por hacer es mejorar su aspecto en pantalla.
- Esto último lo logramos agregando un fondo gris y un pequeño margen interno para separar las opciones del menú del borde del elemento:

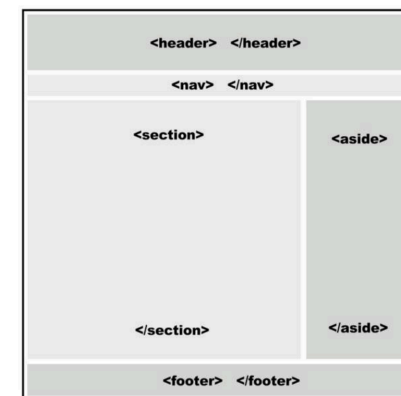
```
#menu {  
  background: #CCCCCC;  
  padding: 5px 15px;  
}  
#menu li {  
  display: inline-block;  
  list-style: none;  
  padding: 5px;  
  font: bold 14px verdana, sans-serif;  
}
```



BARRA DE NAVEGACIÓN

- Los ítems de una lista son posicionados unos sobre otros.
- Para cambiar este comportamiento y colocar cada opción del menú una al lado de la otra, referenciamos los elementos `` dentro de este elemento `<nav>` en particular usando el selector `#menu li`, y luego asignamos a todos ellos el estilo **display: inline-block** para convertirlos en lo que se llama cajas inline.
- A diferencia de los elementos block, los elementos afectados por el parámetro **inline-block** estandarizado en CSS3 no generan ningún salto de línea pero nos permiten tratarlos como elementos block y así declarar un valor de ancho determinado. Este parámetro también ajusta el tamaño del elemento de acuerdo con su contenido cuando el valor del ancho no fue especificado.
- En esta última regla también eliminamos el pequeño gráfico generado por defecto por los navegadores delante de cada opción del listado utilizando la propiedad **list-style**.

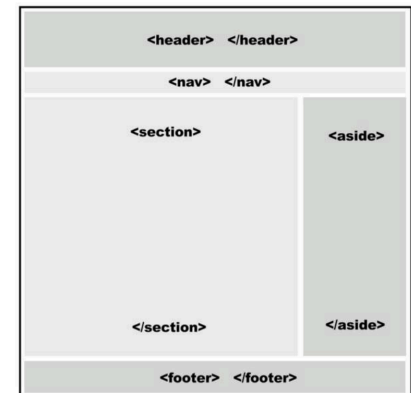
```
#menu {  
  background: #CCCCCC;  
  padding: 5px 15px;  
}  
#menu li {  
  display: inline-block;  
  list-style: none;  
  padding: 5px;  
  font: bold 14px verdana, sans-serif;  
}
```



SECTION Y ASIDE

- **float** es una de las propiedades más ampliamente utilizadas para aplicar el Modelo de Caja.
- Hace que el elemento flote hacia un lado o al otro en el espacio disponible. Los elementos afectados por **float** actúan como elementos block.
- Los elementos son movidos a izquierda o derecha en el área disponible, tanto como sea posible, respondiendo al valor de **float**.

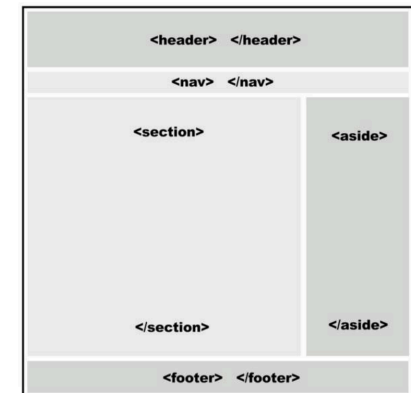
```
#seccion {  
  float: left;  
  width: 660px;  
  margin: 20px;  
}  
#columna {  
  float: left;  
  width: 220px;  
  margin: 20px 0px;  
  padding: 20px;  
  background: #CCCCCC;  
}
```



SECTION Y ASIDE

- Afectado por estos valores, el contenido del elemento **<section>** estará situado a la izquierda de la pantalla con un tamaño de 660 pixeles, más 40 pixeles de margen, ocupando un espacio total de 700 pixeles de ancho.
- La propiedad **float** del elemento **<aside>** también tiene el valor **left** (izquierda). Esto significa que la caja generada será movida al espacio disponible a su izquierda. Debido a que la caja previa creada por el elemento **<section>** fue también movida a la izquierda de la pantalla, ahora el espacio disponible será solo el que esta caja dejó libre. La nueva caja quedará ubicada en la misma línea que la primera pero a su derecha, ocupando el espacio restante en la línea, creando la segunda columna de nuestro diseño.

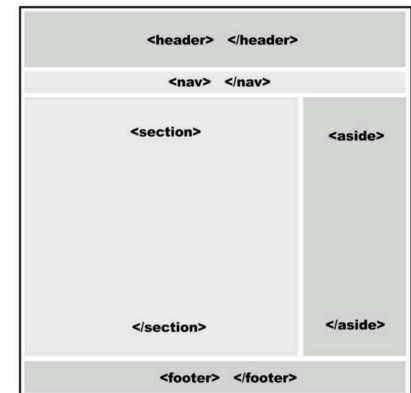
```
#seccion {  
    float: left;  
    width: 660px;  
    margin: 20px;  
}  
#columna {  
    float: left;  
    width: 220px;  
    margin: 20px 0px;  
    padding: 20px;  
    background: #CCCCCC;  
}
```



SECTION Y ASIDE

- El tamaño declarado para esta segunda caja fue de 220 pixeles. También agregamos un fondo gris y configuramos un margen interno de 20 pixeles. Como resultado final, el ancho de esta caja será de 220 pixeles más 40 pixeles agregados por la propiedad **padding** (los márgenes de los lados fueron declarados a **0px**).

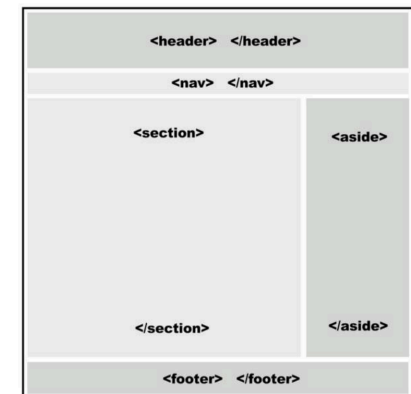
```
#seccion {  
  float: left;  
  width: 660px;  
  margin: 20px;  
}  
#columna {  
  float: left;  
  width: 220px;  
  margin: 20px 0px;  
  padding: 20px;  
  background: #CCCCCC;  
}
```



FOOTER

- Declaramos un borde de 2 píxeles en la parte superior de `<footer>`, un margen interno (**padding**) de 20 píxeles, y centra el texto dentro del elemento.
- Se restaura el normal flujo del documento con la propiedad **clear**. Esta propiedad simplemente restaura las condiciones normales del área ocupada por el elemento. El valor usualmente utilizado es **both**, el cual significa que ambos lados del elemento serán restaurados y el elemento seguirá el flujo normal (este elemento ya no es flotante como los anteriores). Esto, para un elemento block, quiere decir que será posicionado debajo del último elemento, en una nueva línea.

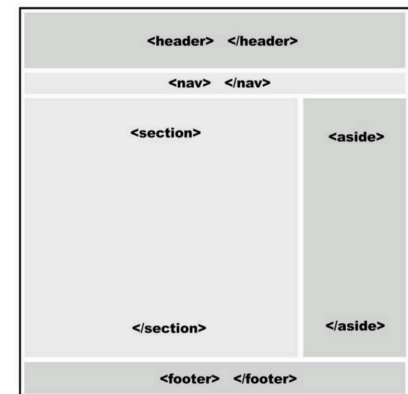
```
#pie {  
  clear: both;  
  text-align: center;  
  padding: 20px;  
  border-top: 2px solid #999999;  
}
```



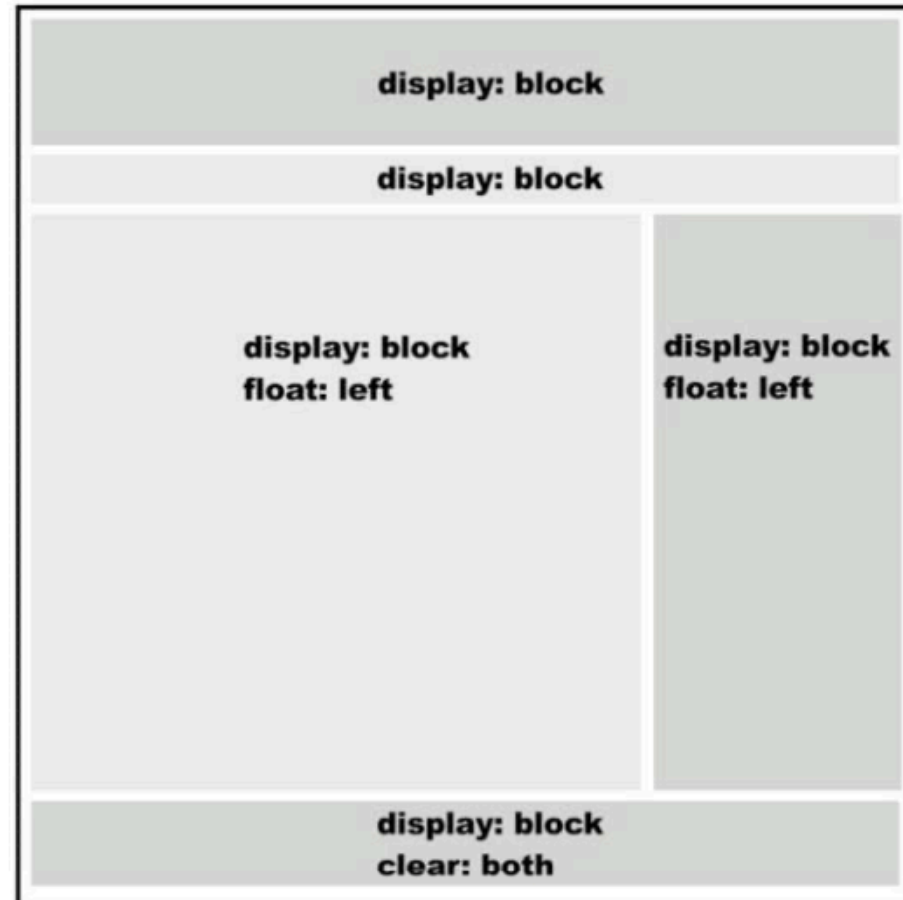
FOOTER

- La propiedad **clear** también empuja los elementos verticalmente, haciendo que las cajas flotantes ocupen un área real en la pantalla. Sin esta propiedad, el navegador presenta el documento en pantalla como si los elementos flotantes no existieran y las cajas se superponen.

```
#pie {  
  clear: both;  
  text-align: center;  
  padding: 20px;  
  border-top: 2px solid #999999;  
}
```



FINALMENTE ...



COMO SE VERÁ?



3

ULTIMOS DETALLES

- A todos los elementos `<article>` le aplicamos un color de fondo, un borde sólido de 1 pixel, un margen interno y un margen inferior.
- Cada elemento `<article>` cuenta también con un elemento `<footer>` que muestra el número de comentarios recibidos, le aplicamos ese estilo para alinear a la derecha el texto dentro de los elementos `<footer>` de cada `<article>`.
- Y al elemento `<figcaption>` le aplicamos estilos de fuente.

```
article {  
    background: #FFFBCC; border: 1px solid #999999; padding: 20px; margin-bottom: 15px;  
}  
article footer {  
    text-align: right;  
}  
figcaption {  
    font: italic 14px verdana, sans-serif;  
}
```

COMO SE VERÁ?



3