CENTRO INTEGRADO DE FORMACIÓN PROFESIONAL Nº 1 - CUENCA

Modelo de caja tradicional

Con elementos ** ** y CSS podemos crear cajas en la pantalla, posicionar estas cajas a un lado o a otro y darles un tamaño, color o borde específico entre otras características. CSS provee propiedades específicas que nos permiten organizar las cajas acordes a nuestros deseos. Estas propiedades son lo suficientemente poderosas como para crear un modelo de caja que se transformó en lo que hoy conocemos como Modelo de Caja Tradicional.

Plantilla

Este modelo necesita agrupar cajas juntas para ordenarlas horizontalmente. Debido a que el contenido completo del cuerpo es creado a partir de cajas, debemos agregar un elemento ** ** para agruparlas, centrarlas y darles un tamaño específico. La nueva plantilla lucirá de este modo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="description" content="Ejemplo de HTML5">
  <meta name="keywords" content="HTML5, CSS3, JavaScript">
  <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>         <time datetime="2022-09-28" pubdate>publicado
28-09-2022
```

```
        </time>
      </header>
      Este es el texto de mi primer mensaje
      <figure>
        <img
src="https://upload.wikimedia.org/wikipedia/commons/thumb/1/19/Hanging_hous
es_in_Cuenca_Spain.jpg/450px-Hanging_houses_in_Cuenca_Spain.jpg">
        <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>         <time datetime="2022-09-29" pubdate>publicado
29-09-2022
        </time>
      </header>
      Este es el texto de mi segundo mensaje
      <footer>
        <p>comentarios (0)</p>
      </footer>
    </article>
  </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>
```

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 cambio realizado es la adición del elemento ** ** mencionado anteriormente. Este ** ** fue identificado con el atributo y el valor **id="agrupar"**, y es cerrado al final del cuerpo con la etiqueta de cierre ** **. 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.

Selector universal *

Comencemos con algunas reglas básicas que nos ayudarán a proveer consistencia al diseño:

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

Listado 2-26. *Regla CSS general.*

Algunos elementos por defecto tienen márgenes que son diferentes de cero y en la mayoría de los casos demasiado amplios. La primera regla en nuestro archivo CSS, nos asegura que todo elemento tendrá un margen interno y externo de 0 pixeles.

> **Conceptos básicos:** Recuerde que en HTML cada elemento es considerado como una caja. El margen (**`margin`**) es en realidad el espacio alrededor del elemento, el que se encuentra por fuera del borde de esa caja (el estilo **`padding`**, por otro lado, es el 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).

Nueva jerarquía para cabeceras

En nuestra plantilla usamos elementos **`<h1>`** y **`<h2>`** para declarar títulos y subtítulos de diferentes secciones del documento.

```
h1 {
  font: bold 20px verdana, sans-serif;
} h2 {
  font: bold 14px verdana, sans-serif; }
```

>.

La propiedad **`font`**, asignada a los elementos **`<h1>`** y **`<h2>`**, nos permite declarar todos los estilos para el texto en una sola línea. Las propiedades que pueden ser declaradas usando **`font`** son: **`font-style`**, **`font-variant`**, **`font-weight`**, **`font-size/line-height`**, y **`font-family`** en este orden. Con estas reglas estamos cambiando el grosor, tamaño y tipo de letra del texto dentro de los elementos **`<h1>`** y **`<h2>`** a los valores que deseamos.

Declarando nuevos elementos HTML5

Otra regla básica que debemos 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>`** y de este modo construir nuestro modelo de caja:

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

Centrando el cuerpo

El primer elemento que es parte del modelo de caja es siempre **`<body>`**. Vamos a centrar su contenido y así logramos que la capa “agrupar” de su interior aparezca centrada.

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

Creando la caja principal

Siguiendo con el diseño de nuestra plantilla, debemos especificar una tamaño o tamaño máximo para el contenido del cuerpo. Agregamos un elemento ** ** a la plantilla para agrupar todas las cajas dentro del cuerpo. Este ** ** será considerado la caja principal para la construcción de nuestro modelo de caja (este es el propósito por el que lo agregamos). De este modo, modificando el tamaño de este elemento lo hacemos al mismo tiempo para todos los demás:

```
#agrupar {   width: 980px;
margin: 15px auto;
  text-align: left;
}
```

La regla en el Listado 2-30 está 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**.

El estilo **margin: 15px auto** asigna 15 pixeles al margen superior e inferior del elemento ** ** que está afectando y declara como automático el tamaño de los márgenes de izquierda y derecha (los dos valores declarados son usados para definir los cuatro márgenes). De esta manera, habremos generado un espacio de 15 pixeles en la parte superior e inferior del cuerpo y los espacios a los laterales (margen izquierdo y derecho) serán calculados automáticamente de acuerdo al tamaño del cuerpo del documento y el elemento ** **, efectivamente centrando el contenido en pantalla.

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. (**text-align: left**) logra este propósito.

Ejercicio

Utilizando la propiedad max-width, modifica el estilo para que el ancho de la capa “agrupar” sea el 75% de la pantalla, pero que su ancho máximo sea de 980 píxeles.

La cabecera

En nuestra plantilla, **<header>** fue identificado con el atributo **id** y el valor **cabecera**.

Como ya mencionamos, cada elemento *block*, así como el cuerpo, por defecto tiene un valor de ancho del 100%. Esto significa que el elemento ocupará todo el espacio horizontal disponible.

```
#cabecera {   background: #FFFBB9;
border: 1px solid #999999;
  padding: 20px; }
```

Listado 2-31. *Agregando estilos para <header>.*

Debido a que **<header>** ocupará todo el espacio horizontal disponible en la caja principal y será tratado como un elemento *block* le otorgamos a **<header>** un fondo amarillo, un borde sólido de 1 pixel y un margen interior de 20 pixeles usando la propiedad **padding**.

Barra de navegación

<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 ** ** principal), y (también por defecto) 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; }
```

Listado 2-32. *Agregando estilos para <nav>.*

Conceptos básicos: La propiedad **padding** trabaja exactamente como **margin**. Cuatro valores pueden ser especificados: superior, derecho, inferior, izquierdo, en este orden. Si solo declaramos un valor, el mismo será asignado para todos los espacios alrededor del contenido del elemento. Si en cambio especificamos dos valores, entonces el primero será asignado como margen interno de la parte superior e inferior del contenido y el segundo valor será asignado al margen interno de los lados, izquierdo y derecho.

Dentro de la barra de navegación hay una lista creada con las etiquetas **<ul>** y **<li>**. Por defecto, 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 **#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..

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**.

Section y aside

Usando la propiedad **float** podemos posicionar estas cajas del lado izquierdo o derecho de acuerdo a nuestras necesidades. Los elementos que utilizamos en nuestra plantilla HTML para crear estas cajas son **<section>** y **<aside>**, cada uno identificado con el atributo **id** y los valores **seccion** y **columna** respectivamente.

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

La propiedad de CSS **float** es una de las propiedades más ampliamente utilizadas para aplicar el Modelo de Caja Tradicional. Hace que el elemento flote hacia un lado o al otro en el espacio disponible. Los elementos afectados por **float** actúan como elementos *block* (con la diferencia de que son ubicados

de acuerdo al valor de esta propiedad y no el flujo normal del documento). Los elementos son movidos a izquierda o derecha en el área disponible, tanto como sea posible, respondiendo al valor de **float**.

La propiedad **float** mueve la caja al espacio disponible del lado especificado por su valor, **width** asigna un tamaño horizontal y **margin**, por supuesto, declara el margen del elemento.

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.

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**).

> **Conceptos básicos:** El tamaño de un elemento y sus márgenes son agregados para obtener el valor real ocupado en pantalla. Si tenemos un elemento de 200 pixeles de ancho y un margen de 10 pixeles a cada lado, el área real ocupada por el elemento será de 220 pixeles. El total de 20 pixeles del margen es agregado a los 200 pixeles del elemento y el valor final es representado en la pantalla. Lo mismo pasa con las propiedades **padding** y **border**. Cada vez que agregamos un borde a un elemento o creamos un espacio entre el contenido y el borde usando **padding** esos valores serán agregados al ancho del elemento para obtener el valor real cuando el elemento es mostrado en pantalla. Este valor real es calculado con la fórmula: **tamaño + márgenes + márgenes internos + bordes**.

Ejercicio

Utilizando la propiedad max-width, modifica el estilo para que el ancho de las capas <section> y <aside> sea el mismo que en el ejercicio pero que entre ellas se repartan un porcentaje equivalente al 100%..

Footer

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

declara un borde de 2 pixeles en la parte superior de **<footer>**, un margen interno (**padding**) de 20 pixeles, y centra el texto dentro del elemento. A sí mismo, restaura el normal flujo del documento con la propiedad **clear**. Esta propiedad simplemente restaura las condiciones normales del área ocupada por el elemento, no permitiéndole posicionarse adyacente a una caja flotante. 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.

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.

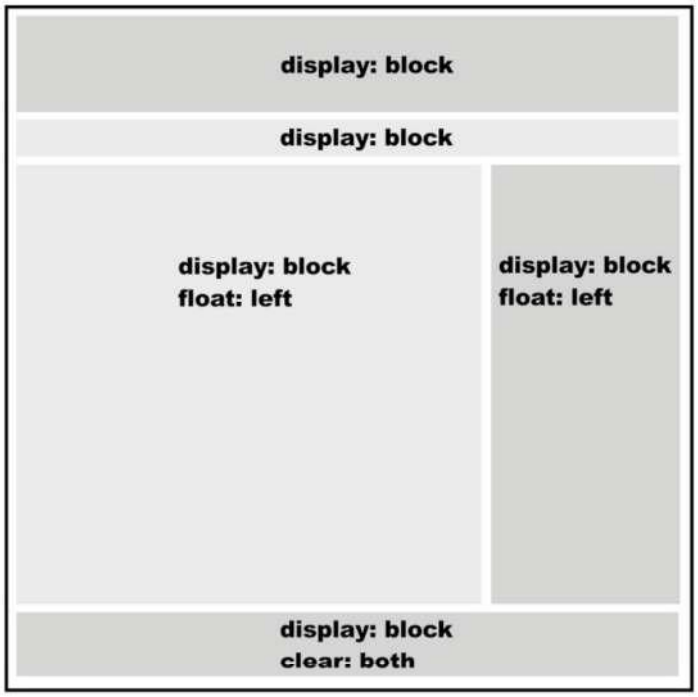

Figura 2-2. Representación visual del modelo de caja tradicional.

Cuando tenemos cajas posicionadas una al lado de la otra en el Modelo de Caja Tradicional siempre necesitamos crear un elemento con el estilo **`clear: both`** para poder seguir agregando otras cajas debajo de un modo natural. La Figura 2-2 muestra una representación visual de este modelo con los estilos básicos para lograr la correcta disposición en pantalla.

Últimos toques

Lo único que nos queda por hacer es trabajar en el diseño del contenido. Para esto, solo necesitamos configurar los pocos elementos HTML5 restantes:

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

Referencia todos los elementos **`<article>`** y les otorga algunos estilos básicos (color de fondo, un borde sólido de 1 pixel, margen interno y margen inferior). El margen inferior de 15 pixeles tiene el propósito de separar un elemento **`<article>`** del siguiente verticalmente.

Cada elemento **`<article>`** cuenta también con un elemento **`<footer>`** que muestra el número de comentarios recibidos. Para referenciar un elemento **`<footer>`** dentro de un elemento **`<article>`**, usamos el selector **`article footer`**

Cambiamos el color de cada elemento **`<time>`** y diferenciamos la descripción de la imagen (insertada con el elemento **`<figcaption>`**) del resto del texto usando una tipo de letra diferente.