Unidad Didáctica 6

Posicionamiento

Contenido

1. Introducción

Mediante CSS se puede controlar la posición de las cajas dentro del área de navegación. El uso de estas propiedades hace que se puedan conseguir efectos y presentaciones muy avanzadas e imposibles de realizar sin el uso de CSS. En este tema se verán algunas formas o modelos de posicionamiento.

2. Propiedades y tipos de posicionamiento

En CSS, la propiedad que se usa para el posicionamiento es *position*. Los valores que puede tener ésta son: *static* (posicionamiento normal o estático), *relative* (posicionamiento relativo), *absolute* (posicionamiento absoluto), *fixed* (posicionamiento fijo). El posicionamiento flotante (ver más adelante) no puede ser controlado con esta propiedad.

Esta propiedad indica cómo se posiciona un elemento, pero no lo desplaza. Para ello existen algunas propiedades, las cuales son: *top* (desplazamiento descendente), *right* (desplazamiento hacia izquierda), *bottom* (desplazamiento ascendente) y *left* (desplazamiento hacia derecha).

2.1. Posicionamiento normal

En este posicionamiento, el navegador únicamente tiene en cuenta si el elemento es bloque o línea. Las cajas se muestran una a continuación de la otra.

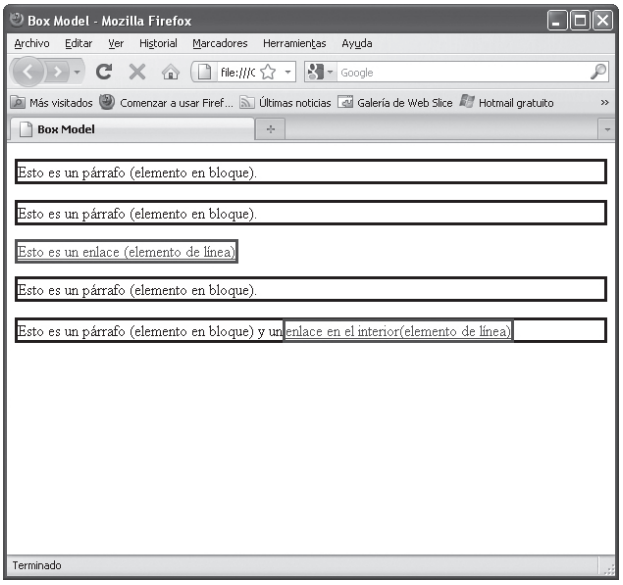
En el caso de que exista un elemento dentro de otro, el elemento "padre" se denomina contenedor y, por lo general, el contenido que se encuentra dentro de una directiva "padre", está limitado por el tamaño de ésta (aunque en algunos casos puede haber desborde de contenido). En el caso de elementos bloque, los elementos se separan entre sí mediante márgenes verticales.

Si las cajas en línea ocupan más que el espacio disponible para ellas, el resto del elemento se muestra en espacios inferiores.

En esta tipología, las propiedades *top*, *right*, *botton* y *left* son ignoradas.

Por ejemplo, observe el siguiente código:

```
P {   position: static;
      border-style: solid;
}
A {
      position: static;
      border-style: solid;
}
...
            <P>Esto es un párrafo (elemento en bloque).</P>
            <P>Esto es un párrafo (elemento en bloque).</P>
            <A href="#">Esto es un enlace (elemento de línea)</A>
            <P>Esto es un párrafo (elemento en bloque).</P>
            <P>Esto es un párrafo (elemento en bloque) y un<A
            href="#">enlace en el interior(elemento de línea)</A></P>
...
```



2.2. Posicionamiento relativo

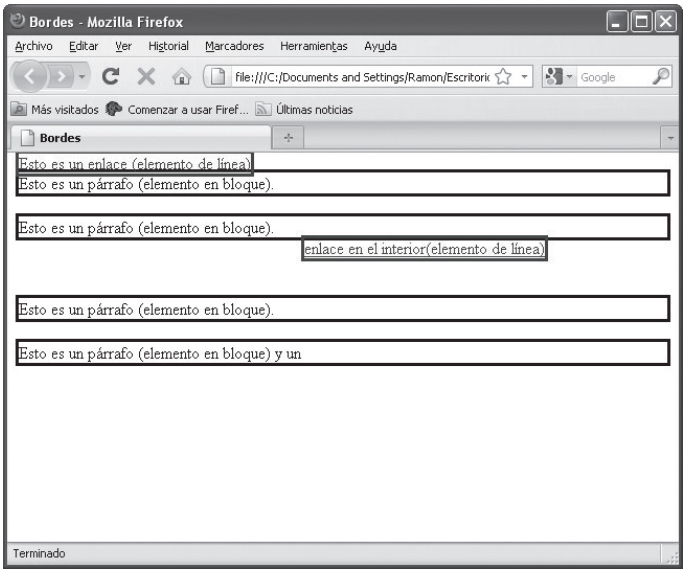
El posicionamiento relativo, desplaza la caja respecto al posicionamiento normal que debería tener. Este deplazamiento se cuantifica con las propiedades *top*, *right*, *bottom* y *left*.

El desplazamiento de una caja no afecta la posición de la demás, por lo que puede existir solapado (la mayor desventaja de este posicionamiento). Si dentro del elemento a mover existen otros elementos, éstos se posicionaran con el elemento "padre":

Si el estilo de la directiva *<A>* del ejercicio anterior se modificara de la siguiente forma:

```
...
A {
        position: relative;
        border-style: solid;
        bottom: 100px;
}
...
```

Manteniendo el resto del código, se podrá visualizar:



direction es una propiedad que permite establecer la dirección del texto de un contenido. Al ser las propiedades *left* y *right* excluyentes, cuando se les da un valor distinto de *auto* (por defecto), solo se tiene en cuenta el valor de una de estas propiedades. La que se tiene en cuenta dependerá del valor de *direction*, si su valor es *ltr* se ignora la propiedad *left* mientras que si es *rtl* se ignora la propiedad *right*.

2.3. Posicionamiento absoluto

Este posicionamiento se usa para establecer la posición de una caja de una forma precisa, la cual se indica con las propiedades *top*, *right*, *bottom* y *left*.

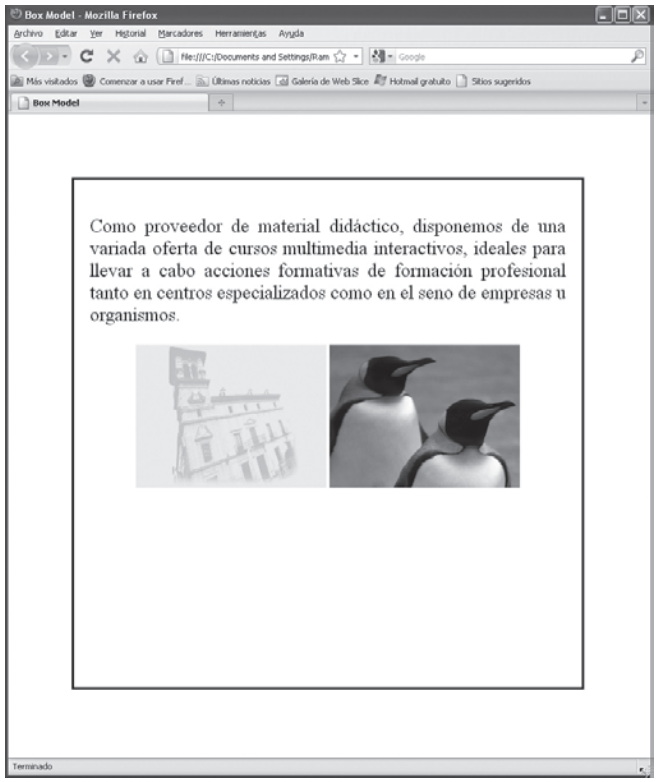
Cuando un elemento se posiciona de forma absoluta, el resto de elementos se recolocan y ocupan el espacio dejado por la caja desplazada. Al igual que en el posicionamiento relativo, pueden existir solapamientos.

Es importante tener en cuenta que la referencia que se toma en los valores de posicionamiento no es la posición que debería tener el elemento normal-

mente (como en el caso anterior), si no el origen de coordenadas (esquina superior izquierda) del primer elemento posicionado de forma diferente a *static*.

A continuación se va a ver un ejemplo. Observe el siguiente código y su correspondiente resultado en el navegador:

```
BODY {
    border-style: solid;
    text-align: center;
    padding: 20px;
    margin: 80px;
    height: 600px;
    width: 600px;
    }
P {
    text-align: justify;
    font-size: x-large;
}
...
<BODY>
        <P>Como proveedor de material
        didáctico, disponemos de una variada oferta de cursos
        multimedia interactivos, ideales para llevar a cabo
        acciones formativas de formación profesional
        tanto en centros especializados como en el seno
        de empresas u organismos.</P>
        <IMG src="imagenes/logo.jpg"/>
        <IMG src="imagenes/img01.jpg"/>
</BODY>
...
```

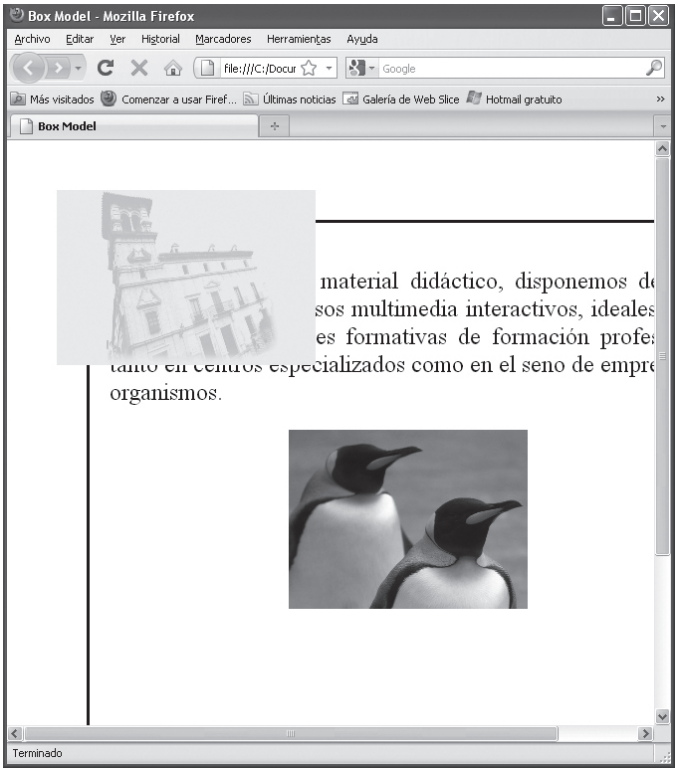


Ahora ser va a añadir el siguiente selector correspondiente a un posicionamiento absoluto:

```
.desplaza {
    position: absolute;
    top: 50px;
    left: 50px;
    }
```

Esta clase se va a aplicar a la imagen *logo.jpg*, en el correspondiente código HTML:

```
...
<IMG class="desplaza" src="imagenes/logo.jpg"/>
...
```



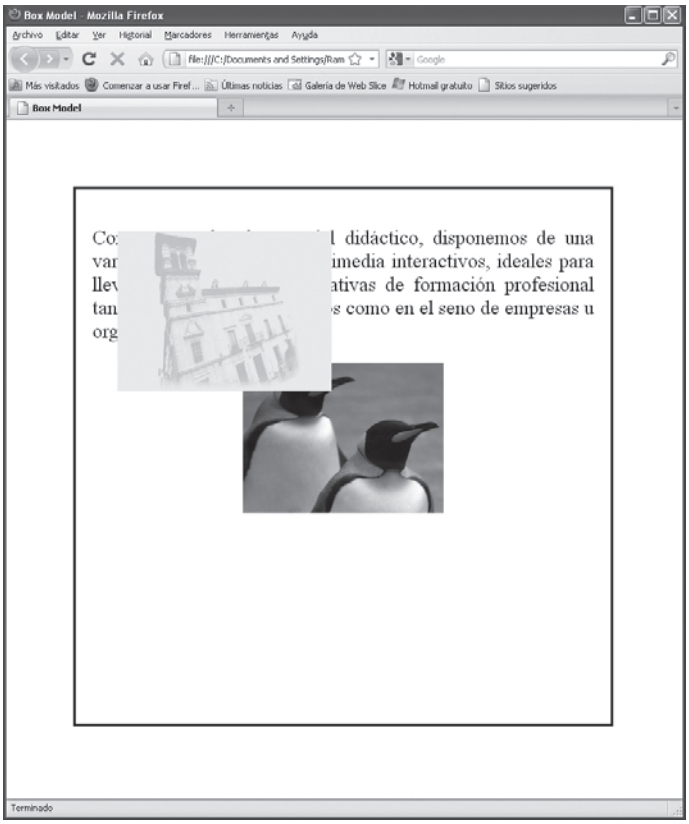
Observe la nueva posición del logo respecto al origen del área de navegación (que corresponde a la esquina superior izquierda de la pantalla) y la "recolocación" de la imagen *img01.jpg* (centrada en *<BODY>)*:

Para posicionar un elemento respecto al elemento "padre" (en el ejemplo: *<BODY>*), lo único que hay que hacer es añadir *position: relative*; al estilo asociado al elemento contenedor en cuestión, ya que deja de estar posicionado de forma *static*. En el ejemplo, basta con añadir:

```
body {
    position: relative;
    border-style: solid;
    text-align: center;
    padding: 20px;
```

```
    margin: 180px;
    height: 600px;
    width: 600px;
    }
...
```

El resultado en el navegador será:



2.4. Posicionamiento fijo

Este método de posicionamiento es idéntico al anterior. La única diferencia se puede observar cuando el usuario modifica y manipula el tamaño de la

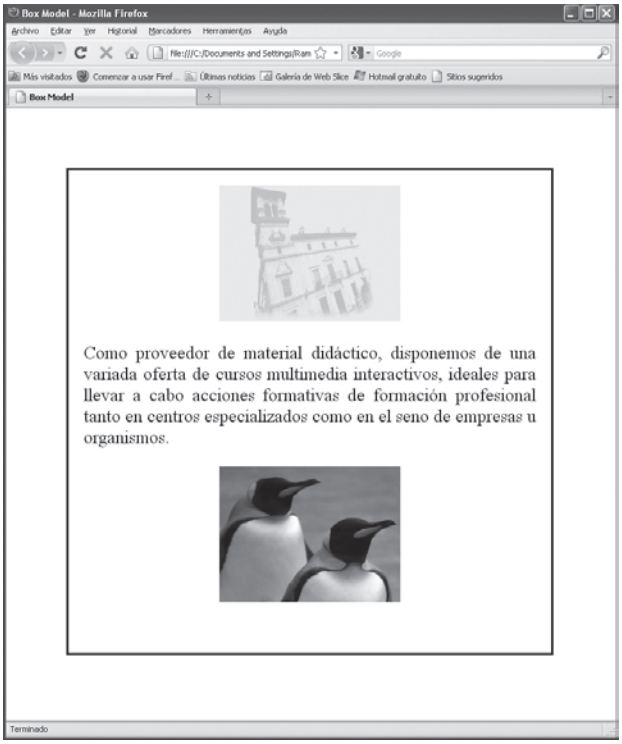
ventana de navegación: en el posicionamiento fijo las cajas no modifican su posición aunque suba o baje la página en su ventana de navegación (algo que no ocurría en el caso anterior).

2.5. Posicionamiento flotante

Es el más complejo pero el más utilizado. Cuando una caja se posiciona de forma flotante, ésta deja de pertenecer al flujo normal de la página. A la hora de posicionarse, la caja se colocará lo más a la izquierda o derecha posible.

La propiedad que permite este tipo de posicionamiento es *float* . Si su valor es *left* o *right*, la caja se desplaza hacia el punto que se encuentre lo más a la izquierda o derecha, respectivamente, posible. El resto de elementos fluyen y se adaptan al posicionamiento de la caja. El valor *none* anula este tipo de posicionamiento.

A continuación se muestra un ejemplo casi idéntico al anterior, con la diferencia de que el párrafo se localiza entre las dos imágenes:



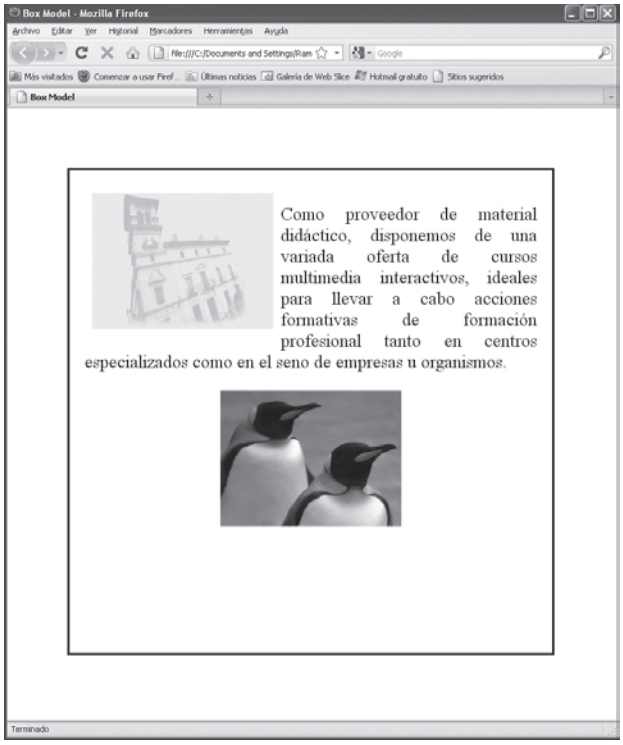
Al añadir este estilo:

```
...
.flotante {
        float: left;
        margin: 10px;
        }
...
```

Se aplica a la imagen superior:

```
...
<IMG class="flotante" src="imagenes/logo.jpg"/>
...
```

Al ejecutar el código podrá observar que la imagen pasa a ser flotante y se desplaza lo más a la izquierda posible, mientras el texto fluye y se adapta a la imagen (la propiedad *margin* del estilo sirve para que no se "junte" el texto con la imagen.

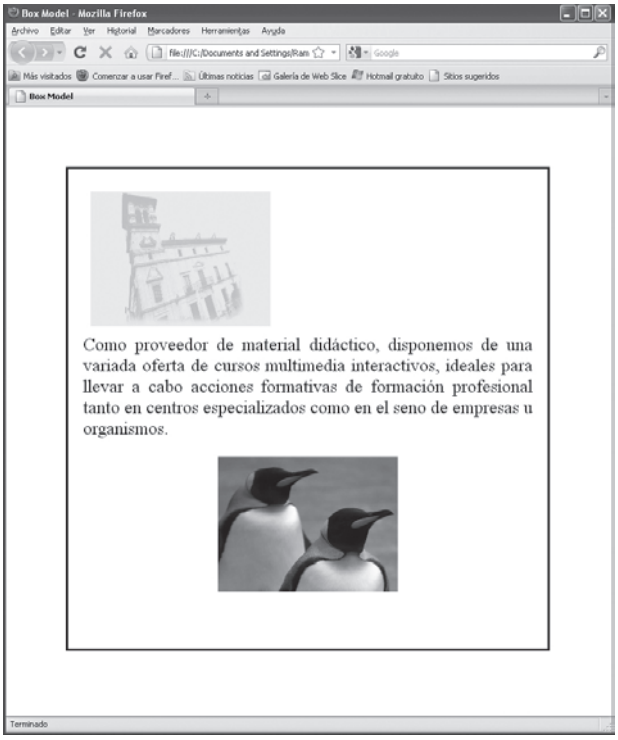


Existe una propiedad llamada *clear* que modifica el comportamiento por defecto del posicionamiento flotante, de forma que fuerza a un elemento mostrarse debajo de cualquier caja flotante. Se usa *left* o *right*, dependiendo de donde se localice el elemento flotante y con el valor *both* se tienen en cuenta ambos lados. Por ejemplo, al añadir el estilo:

```
...
P {
      clear: left;
```

```
        text-align: justify;
        font-size: x-large;
        }
...
```

De esta forma, se "obliga" a que el párrafo que entra en contacto con la caja flotante efectúe un desplazamiento descendente hasta que su borde superior quede por debajo del borde inferior del elemento flotante de la izquierda:



3. Visualización

Existen dos propiedades que consiguen controlar la visibilidad de los elementos: *display* y *visibility*.

La propiedad *display* permite ocultar completamente un elemento haciendo que desaparezca de la página. Al no encontrarse, el resto de elementos Web se redistribuyen ocupando su lugar.

Por otro lado, la propiedad *visibility* hace invisible a un elemento pero no desaparece: sigue estando ahí pero no se ve, por lo que los demás elementos no pasan a ocupar su espacio.

A continuación se van a estudiar más detenidamente estas propiedades.

3.1. Display

display permite operaciones mucho más avanzadas que la simple ocultación de elementos, ya que puede modificar la forma en la que se visualizan.

Los valores que más se utilizan son: *inline*, *block* y *none*. El valor *inline* visualiza un elemento como si fuera en línea mientras que *block* permite la visualización del elemento como bloque. *none* permite hace que el elemento desaparezca de la página y los demás elementos ocupan su lugar (como si no estuviera).

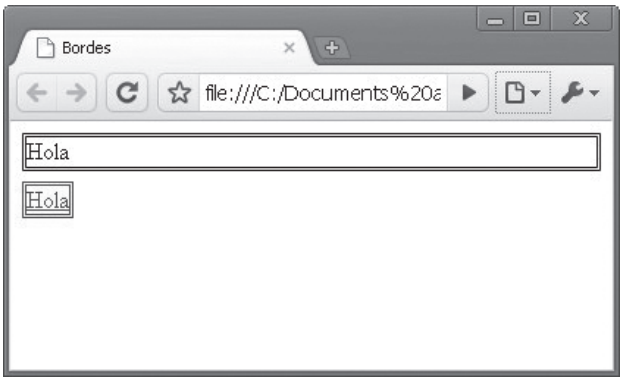
El siguiente ejemplo muestra, un bloque <*SECTION*> y un enlace con borde doble. Como puede ver <*SECTION*> ocupa toda la línea (bloque) mientras que el enlace (<A>) se extiende sólo lo necesario (línea).

Nota

Las directivas <SECTION>, <FOOTER>, <NAV> etc. son nuevas en HTML 5 (ver Módulo anterior) y no son soportadas en algunos navegadores actuales (en el momento de la elaboración del libro no son reconocidas por Firefox y Opera pero sí por Chrome y Safari). No obstante, estas etiquetas son idénticas a <DIV> (reconocida por todos los navegadores) y la única diferencia es puramente semántica.

```
SECTION {
        border-style: double;
        margin-bottom: 10px;
        }


A       {
        border-style: double;
        }
...
                <SECTION>Hola</SECTION>
                <A href="#">Hola</A>
...
```

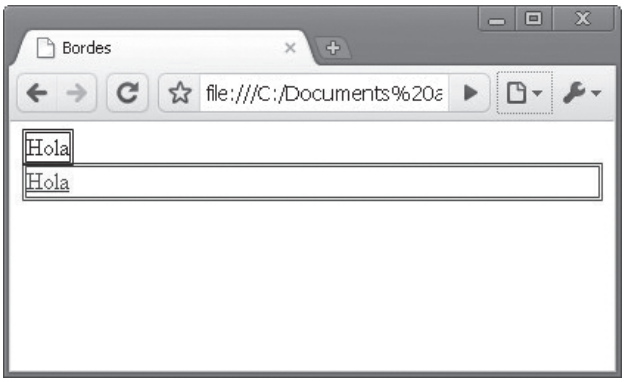


A continuación se va utilizar la propiedad *display* para modificar la visualización de (bloque a línea y viceversa), únicamente cambiando el código CSS por el siguiente:

```
SECTION {
        border-style: double;
        margin-bottom: 10px;
        display: inline;
        }
A       {
```

```
border-style: double;
display: block;
}
```

El resultado es:



Observe que ahora *<SECTION>* se visualiza como elemento de línea y el enlace como un bloque. Observe que, la propiedad *margin-bottom* de *<SECTION>* deja de tener efecto (como estudió en el tema anterior, al igual que *margin-to*p no funciona en elementos de línea).

3.2. Visibility

Esta propiedad es mucho más simple que la anterior ya que únicamente permite hacer visible o invisible a los elementos.

El valor *hidden*, la caja se vuelve invisible y deja de mostrar sus contenido aunque el lugar que ocupaba sigue apareciendo como un hueco vacío.

El valor *collapse* solo se puede usar en tablas ya que permite ocultar filas y/o columnas y mostrar otros contenidos en su lugar.

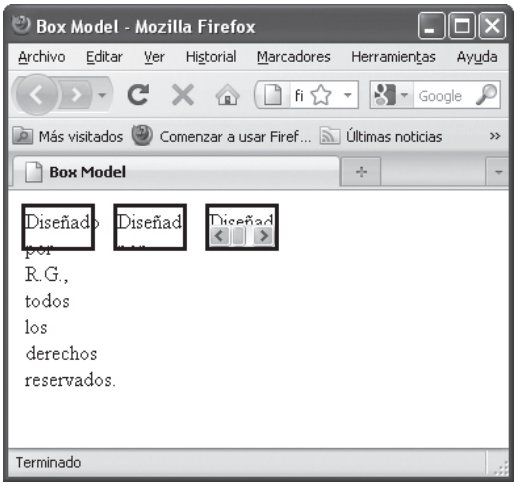
3.3. Otras propiedades de visualización

Existen otras propiedades CSS muy usadas que permiten modificar la visualización de los elementos. Estas son: *overflow* y *z-index*.

3.3.1 *overflow*

CSS permite controlar el "desbordamiento" de contenido con la propiedad *overflow*. Los valores que puede tener son:

- *visible*: Es el comportamiento por defecto y hace que el contenido sobresalga cuando no quepa (ver imagen anterior).
- *hidden*: El contenido que sobresale desaparece y no se vusualiza
- *scroll*: El contenido que se visualiza es el que cabe en la región contenedora. También aparecen unas barras de *scroll* que permiten visualizar el resto del contenido
- *auto*: El comportamiento depende del navegador.



3.3.2 *z-index*

Con la propiedad *z-index* se puede controlar el orden de superposición en caso de que se solapen dos o varias cajas. Lo más normal es que el valor de esta propiedad sea un número entero y que se considere el valor 0 como el más bajo (o el de menos prioridad a la hora de visualizarse). A medida que aumenta este valor, también lo hace la prioridad o nivel de muestreo de la caja, por ejemplo, una caja con *z-indez=8* se mostrará sobre una de *z-index=3*).

Por defecto las cajas se posicionan por orden de declaración en el código HTML correspondiente (abajo del todo se sitúa la primera mientras que la última en declararse es la que se ve por encima de todas las demás).

Observe estas tres cajas posicionadas de forma absoluta:

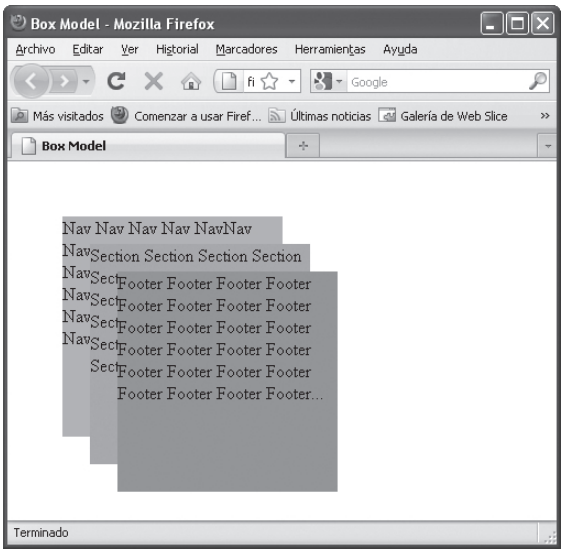
```
nav {
    position: absolute;
    top: 50px;
    left: 50px;
    width: 200px;
    height: 200px;
    background-color: #AADE55;
    }

section {
    position: absolute;
    top: 75px;
    left: 75px;
    width: 200px;
    height: 200px;
    background-color: #DEAA55;
    }

footer {
    position: absolute;
    top: 100px;
    left: 100px;
```

```
        width: 200px;
        height: 200px;
        background-color: #55AADE;
        }

...
                <NAV>Nav Nav  Nav Nav Nav...</NAV>
                <SECTION>Section Section Section...</SECTION>
                <FOOTER>Footer Footer Footer...</FOOTER>
...
```



Como puede observar, las cajas se sitúan en orden de declaración HTML. A continuación se va a invertir el orden de disposición con la propiedad *z-index* añadiendo las siguientes líneas al código CSS:

```
nav {
        ...
        z-index: 3;
        ...
```

```
    }

section {
    …
    z-index: 2;
    …
    }

footer {
…
    z-index: 1;
    …
    }
```

Si se ejecuta el documento *.html*, en el navegador se verá lo siguiente: