

9. Trabajar con el modelo de caja

En la práctica anterior vimos el concepto de modelo de caja. A partir de este punto, existen varias propiedades relacionadas con el trabajo del modelo de caja, cada una de ellas ayudan a la colocación, transformación, posición, etc. del elemento caja. Veremos todas ellas a lo largo de la práctica.

9.1. Posicionamiento

Un elemento puede posicionarse de cuatro formas distintas:

- Posicionamiento estático.
- Posicionamiento relativo.
- Posicionamiento absoluto.
- Posicionamiento fijo.

9.1.1. Posicionamiento estático

Es el posicionamiento por defecto del elemento.

Se determina mediante el atributo **static**.

Position: static;

Con este atributo el diseñador no tiene el control. El elemento no puede posicionarse ni reposicionarse y su visibilidad no puede modificarse.

9.1.2. Posicionamiento relativo

Es el posicionamiento de un elemento respecto a su posición normal o estática.

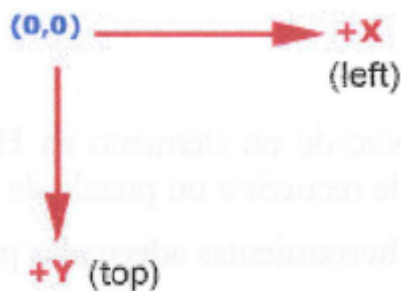
Este elemento permanece en el flujo de los datos de la página aunque está, en cierta medida, descentrado o desplazado respecto a su posición normal. El posicionamiento relativo se determina con el atributo **relative** y el desplazamiento:

Posición	Desplazamiento
position: relative;	left: <i>valor o porcentaje</i> , top: <i>valor o porcentaje</i> , right: <i>valor o porcentaje</i> , bottom: <i>valor o porcentaje</i> ,

La posición se define mediante las coordenadas horizontales (**x**) y verticales (**y**) donde:

X es la distancia respecto al borde izquierdo del elemento padre o de la ventana del navegador (eje horizontal). De este modo, **left** determina la distancia entre la parte izquierda del elemento y el borde izquierdo de la página y **right** la distancia entre la parte derecha del elemento y el borde derecho de la página.

Y es la distancia respecto al borde superior del elemento padre o de la ventana del navegador (eje vertical). De este modo, **top** determina la distancia entre el borde superior de la página y **bottom** la distancia entre el borde inferior del elemento y el borde inferior de la página.



En la práctica basta con informar una sola especificación en el eje horizontal (**left o right**) y otra sobre el eje vertical (**top o bottom**). Los valores de **top**, **right**, **bottom o left** pueden ser negativos.

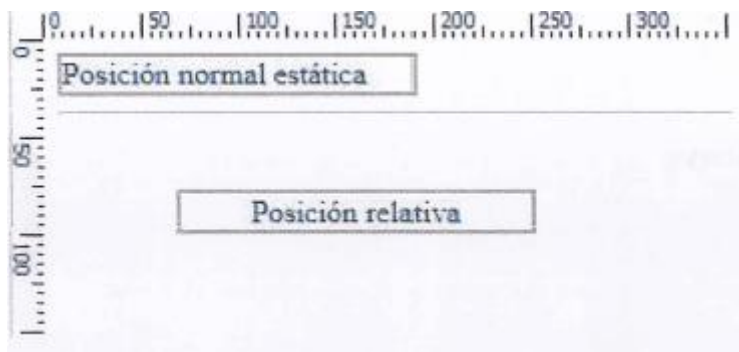
Ejemplo

Crea una página con el nombre posición.html. Escribe el siguiente código:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Trabajo modelo caja</title>
<meta charset="iso-8859-1">
<style>
  .normal{
    width: 180px;
    border: 2px solid gray;
    text-align: center;
  }
```

```
.relativa{
  position: relative;
  top: 30px;
  left: 60px;
  width: 180px;
  border: 2px solid gray;
  text-align: center;
}
</style>
</head>
<body>
  <div class="normal">
    Posición normal estática
  </div>
  <hr>
  <div class="relativa">
    Posición relativa
  </div>
</body>
</html>
```

Resultado:



En este ejercicio, la capa con la posición estática aparece en el orden declarado y posición por defecto, sin poder modificar la posición. Mientras que la capa con posición relativa, aparece en el orden declarado en el código **HTML** pero desplazada 60 píxeles del borde izquierdo y 30 píxeles del borde superior del elemento padre, es decir, de donde debería aparecer la capa sin la declaración de las propiedades **left** y **top**.

9.1.3. Posicionamiento absoluto

El posicionamiento absoluto crea un elemento independiente del resto del documento. Los elementos definidos con posición absoluta se retiran del flujo normal y se posicionan en el lugar exacto definido por el diseñador.

La posición se define mediante las coordenadas horizontales y verticales (x, y) donde:

X es la distancia respecto al borde izquierdo del elemento padre o de la ventana del navegador (eje horizontal). De este modo, `left` determina la distancia entre el borde izquierdo del elemento y el borde izquierdo de la página y `right`, la distancia entre el borde derecho del elemento y el borde derecho de la página.

Y es la distancia respecto al borde superior del elemento padre o de la ventana del navegador (eje vertical). De este modo, `top` determina la distancia entre el borde superior del elemento y el borde superior de la página y `bottom` la distancia entre el borde inferior del elemento y el borde inferior de la página.

El posicionamiento absoluto **tiene el riesgo de no mostrar correctamente la página en resoluciones de pantalla diferentes** a la utilizada en tiempo de diseño puesto que los elementos que se hayan posicionado mediante esta propiedad pueden superponerse sobre los elementos de flujo normal de la página.

Conviene recordar que los navegadores agregan por defecto un margen al cuerpo de la página **HTML** y que este margen por defecto varía de un navegador a otro. Por ello es prudente, en caso de posicionamiento absoluto, especificar los márgenes de la etiqueta **<BODY>**.

El posicionamiento absoluto se determina con el atributo **absolute** y el desplazamiento:

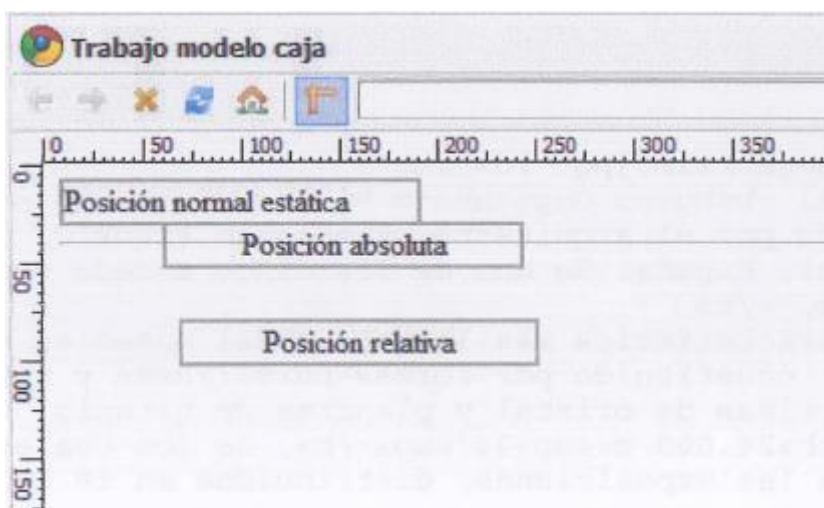
Posición	Desplazamiento
<code>position: absolute;</code>	<code>left: valor o porcentaje;</code> <code>top: valor o porcentaje;</code> <code>right: valor o porcentaje;</code> <code>bottom: valor o porcentaje;</code>

Ejemplo

Modifica la página `posición.html` para que quede como el siguiente código:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Trabajo modelo caja</title>
<meta charset="iso-8859-1">
<style>
  .normal{
    width: 180px;
    border: 2px solid gray;
    text-align: center;
  }
  .relativa{
    position: relative;
    top: 30px;
    left: 60px;
    width: 180px;
    border: 2px solid gray;
    text-align: center;
  }
  .absoluta{
    position: absolute;
    top: 30px;
    left: 60px;
    width: 180px;
    border: 2px solid gray;
    text-align: center;
  }
</style>
</head>
<body>
  <div class="normal">
    Posición normal estática
  </div>
  <hr>
  <div class="relativa">
    Posición relativa
  </div>
  <div class="absoluta">
    Posición absoluta
  </div>
</body>
</html>
```

Resultado:



En este ejercicio, pese a que la capa con posición relativa y la capa con posición absoluta tienen los mismos desplazamientos definidos, top y left, se posicionan de forma distinta. Esto es así porque la

capa relativa se posiciona a partir de su declaración en el código HTML, es decir dentro del flujo de la página y respetando el orden del resto de elementos, mientras que la capa absoluta se posiciona a partir de los bordes izquierdo y superior de la página sin tener en cuenta el resto de elementos de la página.

9.1.4. Posicionamiento fijo

El posicionamiento fijo también crea un elemento independiente del que se puede definir la posición exacta. En este caso el elemento se mantiene fijo aunque el documento se desplace o se deslice en pantalla.

El posicionamiento fijo se determina mediante el atributo fixed:

Position: fixed;

Ejemplo

Crea el documento posicion2.html con el siguiente código:

```
<!DOCTYPE html>
<html lang="es">
<head>
<title>Trabajo modelo caja</title>
<meta charset="iso-8859-1">
<style>
    .fija{
        width: 770px;
        text-align: justify;
        color: rgb(30,30,90);
        position: fixed;
        top: 380px;
        left: 20px;
    }
    .aboluta{
        width: 770px;
        text-align: justify;
        color: rgb(30,30,90);
        position: absolute;
        top: 630px;
        left: 20px;
    }
    h1{
        font-family: verdana;
        font-size: 12pt;
    }
    #img{ width: 800px;}
```



```

</style>
</head>
<body>
  <h1>Guggenheim</h1>
  
  <p class="absoluta">El <b>Museo Guggenheim Bilbao</b> es un museo de arte
  contemporáneo diseñado por el arquitecto canadiense Frank O. Gehry, y localizado
  en Bilbao (País Vasco), España. Es uno de los cinco museos de la Fundación
  Solomon R. Guggenheim. </p>
  <p class="fija">La característica más llamativa del museo es el innovador edificio
  en el que se emplaza, constituido por formas curvilíneas y retorcidas, recubiertas
  de piedra caliza, cortinas de cristal y planchas de titanio. Cuenta con una
  superficie total de <b>24.000 m<sup>2</sup></b>, de los cuales 10.540m<sup>2</sup></b>
  están reservados para las exposiciones, distribuidos en 19 galerías.</p>
</body>
</html>

```

Puedes comprobar las diferencias entre el posicionamiento absoluto y el fijo que en definición son muy parecidos. El párrafo con posición fija aparece en todo momento en pantalla, a 380 píxeles del borde superior se desplace o no la página. Mientras que el párrafo con posición absoluta se ha declarado a una distancia de 630 píxeles que inicialmente no se ve pues supera el alto de la visión de la ventana y requiere desplazar la página hacia abajo para poder ser visible.

El posicionamiento absoluto se determina mediante top, left, right y bottom respecto del documento, mientras que el fijo lo hace respecto a la ventana del navegador.

9.2. Flotar y despejar

9.2.1. Flotar

El posicionamiento flotante es el más difícil de comprender pero al mismo tiempo es de los más utilizados. La mayoría de estructuras de las páginas Web complejas están diseñadas de esta manera.

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.

La propiedad CSS que permite posicionar de forma flotante una caja se denomina float.

La propiedad float permite retirar un elemento tipo caja del flujo normal de la página para situarlo lo más a la derecha o izquierda posible dentro de su elemento padre, es decir, su contenedor.

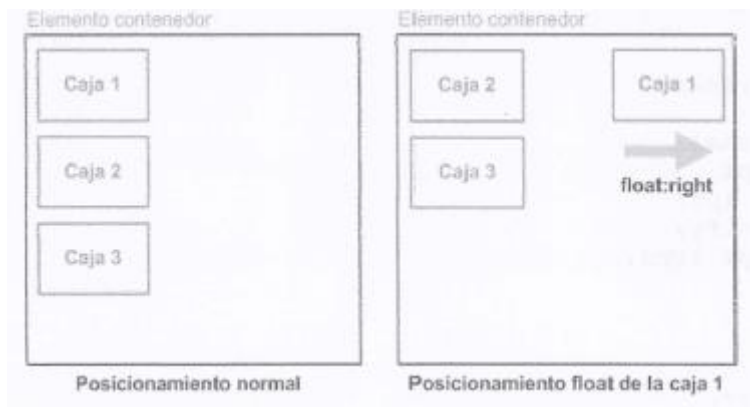
Propiedad	Valores
float:	left; o bien, right; o bien, inherit; o bien, none;

Las características de la propiedad float son:

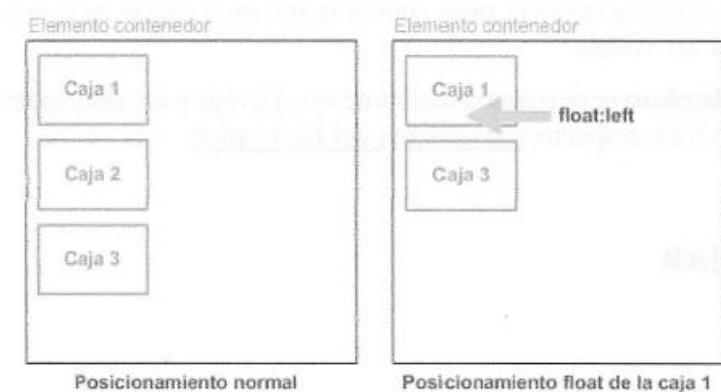
- Right: alinea a la derecha el elemento indicado, empujando a los demás elementos hacia la izquierda.
- left: alinea a la izquierda el elemento indicado, empujando a los demás elementos hacia la derecha.
- None: no especifica nada y deja la gestión al navegador.
- Inherit: para la herencia o cascada

La posición float no puede aplicarse en caso de posicionamiento absoluto.

La siguiente imagen muestra el resultado de posicionar de forma flotante hacia la derecha la caja 1:



Si en el anterior ejemplo la caja 1 se posiciona de forma flotante hacia la izquierda, el resultado es el que muestra la siguiente imagen:



La caja 1 es de tipo flotante, por lo que desaparece del flujo normal de la página y el resto de cajas ocupan su lugar. El resultado es que la caja 2 ahora se muestra dónde estaba la caja 1 y la caja 3 se muestra dónde estaba la caja 2.

Al mismo tiempo, la caja 1 se desplaza todo lo posible hacia la izquierda de la posición en la que se encontraba. El resultado es que la caja 1 se muestra encima de la nueva posición de la caja 2 ocultando su contenido.

Ejemplo

Cree un documento con el nombre posicion3.html y escriba el siguiente código:

```
<html lang="es">
<head>
<title>Trabajo modelo caja</title>
<meta charset="iso-8859-1">
<style>
  *{margin: 0px; padding: 0px;}
  #contenedor{
    margin: auto;
    width: 800px;
    height: 240px;
    border: 2px inset rgb(214,12,167);
    padding: 20px;
  }
  h1{
    font-family: verdana;
    font-size: 12pt;
    font-weight: bold;
    text-transform: uppercase;
    text-decoration: underline;
    height: 20px;
  }
  #img{
    float: right;
    border: 1px outset rgb(65,123,213);
    margin: 0px 10px;
  }
  p{
    font-family: verdana;
    font-size: 10pt;
    font-weight: normal;
    line-height: 20px;
    letter-spacing: 1px;
    text-align: justify;
    padding: 10px 0px 20px;
  }
</style>
</head>
<body>
  <div id="contenedor">
```

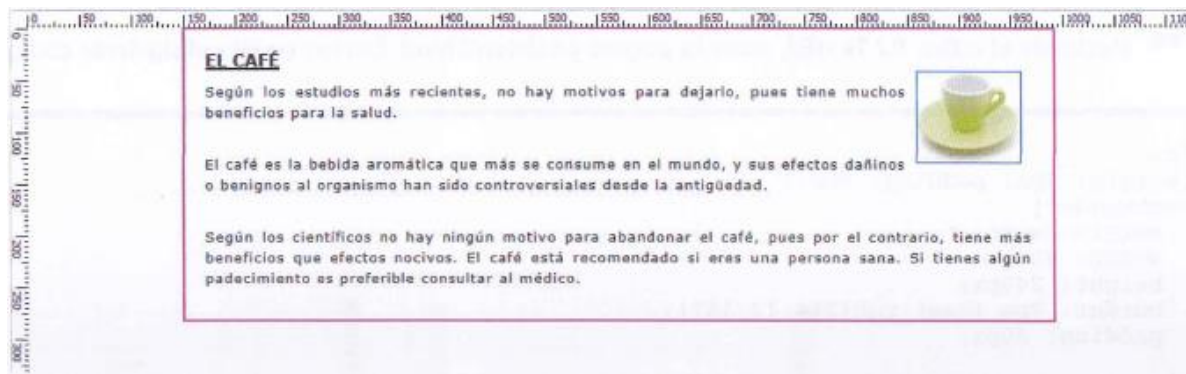
```

<h1>El café</h1>

<p>Según los estudios más recientes, no hay motivos para dejarlo, pues tiene
muchos beneficios para la salud. </p>
<p>El café es la bebida aromática que más se consume en el mundo, y sus efectos
dañinos o benignos al organismo han sido controversiales desde la antigüedad.
</p>
<p>Según los científicos no hay ningún motivo para abandonar el café, pues por
el contrario, tiene más beneficios que efectos nocivos. El café está
recomendado si eres una persona sana. Si tienes algún padecimiento es
preferible consultar al médico.</p>
</div>
</body>
</html>

```

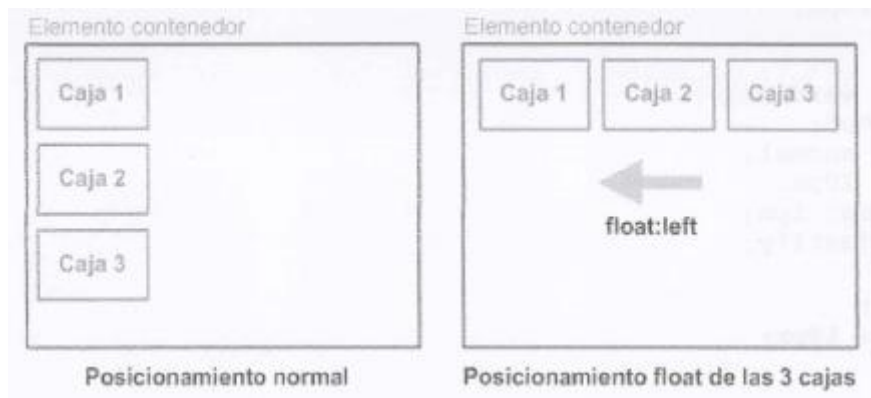
Resultado:



Observe como la imagen flotante se alinea a la derecha, siguiendo el flujo del documento, después del título. El resto de elementos que hay a continuación de la imagen, los párrafos, se colocan a continuación del título, dando la sensación de que la imagen está flotando en el código. Si la imagen no dispusiera de esta propiedad, float, se mostraría alineada a la izquierda debajo del título y todos los párrafos se desplazarían debajo de la imagen siguiendo el flujo y orden del código del documento.

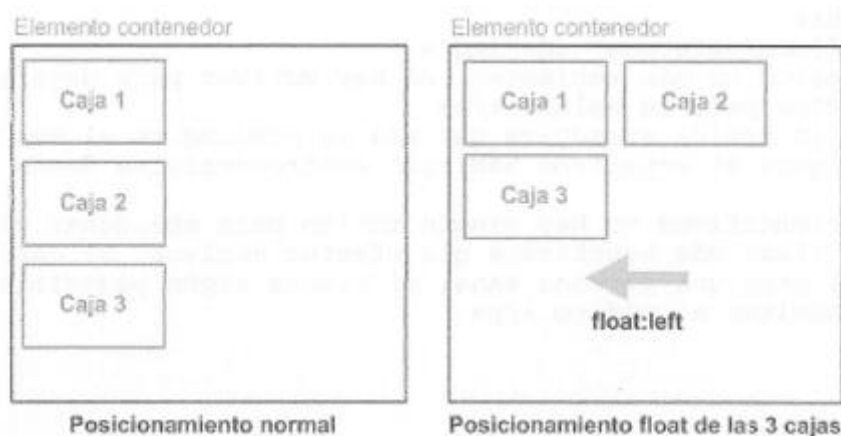
Si existen otras cajas flotantes, al posicionar de forma flotante otra caja, se tiene en cuenta el sitio disponible.

En el siguiente ejemplo se posicionan de forma flotante hacia la izquierda las tres cajas:



En el ejemplo anterior, las cajas no se superponen entre sí porque las cajas flotantes tienen en cuenta las otras cajas flotantes existentes. 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:



Las cajas flotantes influyen en la disposición de todas las demás cajas. Los elementos en línea se ajustan con respecto a las cajas flotantes adaptando su anchura al espacio libre dejado por la caja desplazada. Los elementos de bloque por el contrario no lo hacen, pero sí que adaptan sus contenidos para que no se solapen con las cajas flotantes.

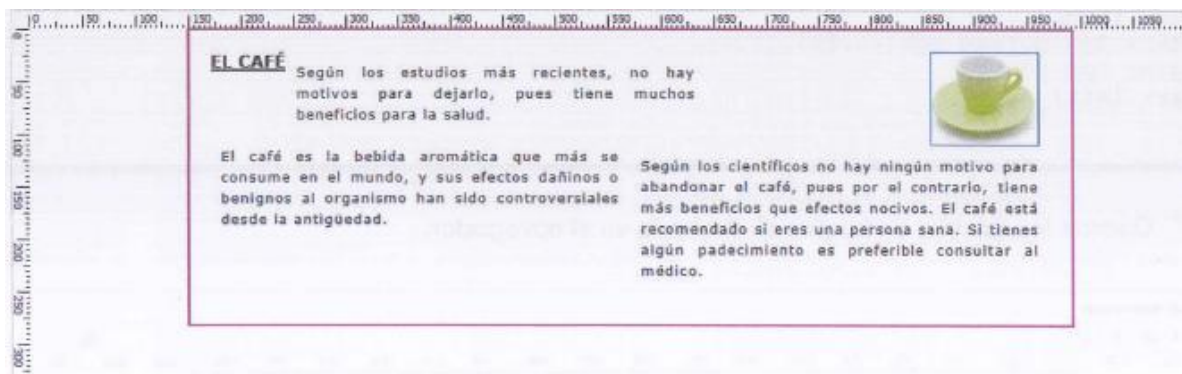
Edite la página `posicion3.html`. Escriba el siguiente código:

```

<style>
  *{margin: 0px; padding: 0px;}
  #contenedor{
    margin: auto;
    width: 800px;
    height: 240px;
    border: 2px inset rgb(214,12,167);
    padding: 20px;
  }
  h1{
    font-family: verdana;
    font-size: 12pt;
    font-weight: bold;
    text-transform: uppercase;
    text-decoration: underline;
    height: 20px;
    float: left;
  }
  #img{
    float: right;
    border: 1px outset rgb(65,123,213);
    margin: 0px 10px;
  }
  p{
    font-family: verdana;
    font-size: 10pt;
    font-weight: normal;
    line-height: 20px;
    letter-spacing: 1px;
    text-align: justify;
    float: left;
    width: 380px;
    padding: 10px 10px;
  }
</style>

```

Resultado:



Ahora todos los elementos son flotantes, observe como se distribuyen los elementos caja ocupando los espacios disponibles.

9.2.2. Despejar

La propiedad clear permite modificar el comportamiento por defecto del posicionamiento flotante para forzar a un elemento a mostrarse debajo de cualquier caja flotante. Esta propiedad anula el efecto introducido por la propiedad float, por lo que se usan conjuntamente.

La definición formal de la propiedad clear se muestra a continuación:

Propiedad	Valores
clear:	left; o bien, right; o bien, both; o bien, inherit; o bien, none;

Las características de la propiedad clear son:

- right: anula los elementos flotantes a la derecha. Si se indica el valor right, el comportamiento es análogo, salvo que en este caso se tienen en cuenta los elementos desplazados hacia la derecha.
- Left: anula los elementos flotantes a la izquierda. Si se indica el valor left, el elemento se desplaza de forma descendente hasta que pueda colocarse en una línea en la que no haya ninguna caja flotante en el lado izquierdo.
- None: este valor permite elementos flotantes.
- Both: anula los elementos flotantes de ambos lados. El valor both despeja los lados izquierdo y derecho del elemento, ya que desplaza el elemento de forma descendente hasta que el borde superior se sitúe por debajo del borde inferior de cualquier elemento flotante hacia la izquierda o hacia la derecha.
- Inherit: valor para la herencia o cascada.

La propiedad clear indica el lado del elemento HTML que no debe ser adyacente a ninguna caja posicionada de forma flotante.

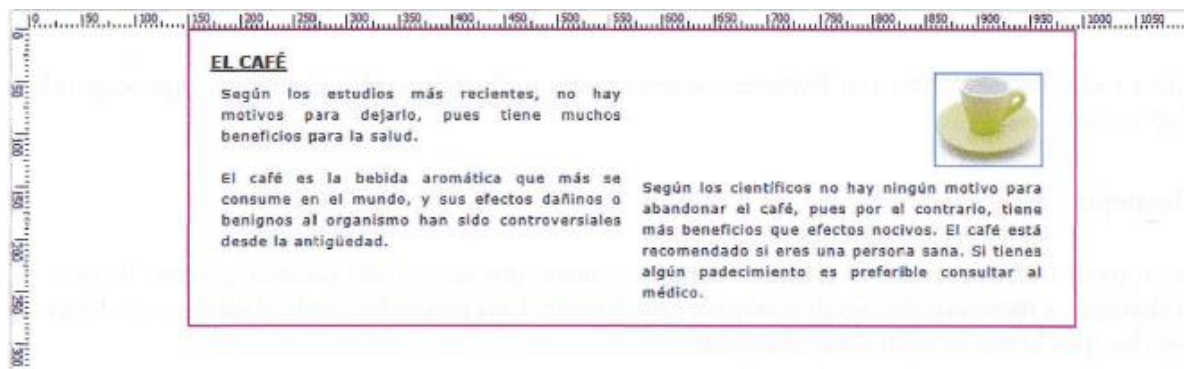
Como se verá más adelante, la propiedad clear es imprescindible cuando se crean las estructuras de las páginas Web complejas.

Ejemplo

Edite de nuevo la página posicion3.html. Escriba el siguiente código:

```
#img{
  float: right;
  border: 1px outset rgb(65,123,213);
  margin: 0px 10px;
  clear: left;
}
```

Resultado:



Observa como el primer párrafo pierde la propiedad flotante a la izquierda, y se sitúa por debajo del título en vez del espacio comprendido entre el título y la imagen.

9.3. Display

Los elementos HTML se dividen en dos grandes tipos: **elementos block** que tienden a ocupar el espacio disponible a todo lo ancho y en caso de existir varios se sitúan unos debajo de otros y **elementos inline** que ocupan el espacio necesario dentro de una línea y en caso de existir varios se sitúan uno junto a otro en la misma línea (siempre que haya espacio). La propiedad display nos permite alterar el tipo de caja con que se muestra un elemento.

Sabemos que por ejemplo los elementos tipo título como h1 son elementos block, y por tanto de forma “natural” tienden a ocupar todo el ancho de la página. Por el contrario elementos como los links o las imágenes son elementos inline.

- **Función de la propiedad:** Permite definir el tipo de posición de caja para visualizar un elemento.
- **Valores posibles:**
 - o **inline** (el elemento se muestra en una caja inline)
 - o **block** (el elemento se muestra en una caja block).
 - o **none** (el elemento no se muestra; el efecto es como si no existiera, por lo que su espacio será ocupado por otros elementos)
 - o **list-item** (el elemento se comporta como si fuera un elemento li)
 - o **inline-block** (el elemento genera una caja block pero que se comporta como si fuera inline admitiendo otros elementos en la misma línea; el comportamiento se asemeja al de los elementos img)
 - o Otros que llevan a que el elemento simule el comportamiento de otro (inline-table, table, table-caption, table-cell, table-column, table-column-group, table-footer-group, table-header-group, table-row, table-row-group)
 - o Otros avanzados (flex, inline-flex, grid, inline-grid, run-in)
 - o **inherit** (se heredan las características del elemento padre).

La propiedad display admite numerosos valores, pero los más usados son inline, block e inline-block. Son estos tres valores los que debemos aprender ahora. El resto los iremos conociendo progresivamente a medida que nos puedan resultar necesarios o cuando profundicemos en el conocimiento de CSS. Tener en cuenta también que estas otras propiedades a que nos referimos pueden no ser reconocidas por muchos navegadores.

Ejemplo resuelto

Abre “ejemplodisplay.html” y haz los siguientes cambios:

```
*{font-family: arial;}
body {width:600px;}
div {border-style: solid;}
div div {border-style: dashed; border-color: green; margin: 5px;}
div div div {border-style: dashed; border-color: red; margin: 5px;}
h1 {border-style: dashed; border-color: green;}
h2 {border-style: dashed; border-color: red; }
li {border-style: dotted; border-width:thin; border-color:brown; margin: 3px;}
```

Con esto lo único que hemos hecho es poner bordes y márgenes para visualizar las cajas que conforman la página web.

Vemos un div en color verde correspondiente al menú (dentro del cual hay otro div en color rojo con el texto “Menú”, una línea generada por el `<hr/>` y una lista `ul` con los items de menú).

Accede al código HTML y establece como id de dicho div “menu1”.

Ahora, sin ejecutar el código, indica cuáles deberían ser los resultados obtenidos al añadir las líneas que se indican en la tabla al archivo CSS. Completa la tabla primero sin ejecutar el código. Luego, compara tu solución con la expuesta en la solución.

Fragmento de código añadido	Resultado
<code>li {display: inline;}</code>	
<code>img {display: block;}</code>	
<code>#menu1 {display:none;}</code>	
<code>h2 {margin-left:30px; display:list-item;}</code>	
<code>li {display: inline-block;}</code>	
<code>ul {display: table; } li {display: table-cell; padding:10px;}</code>	

SOLUCIÓN

Prueba a visualizar los resultados en tu navegador al añadir estos fragmentos de código y comprueba que obtienes los mismos resultados que se indican a continuación:

Fragmento de código añadido	Resultado
<code>li {display: inline;}</code>	Los elementos de ítem de menú que son <code>li</code> y que ocupaban cada uno una línea, se ponen en la misma línea. Sólo cuando no caben en una línea continúan en la siguiente.
<code>img {display: block;}</code>	Las imágenes que estaban en una sola línea unas junto a otras se sitúan cada una en una línea comportándose como elementos <code>block</code>
<code>#menu1 {display:none;}</code>	El menú (con sus cajas interiores, ítems de menú, etc.) desaparece completamente y su espacio es ocupado por el texto “Aprender a programar es...”
<code>h2 {margin-left:30px; display:list-item;}</code>	El título <code>h2</code> aparece con una viñeta como si fuera un elemento <code>li</code> de una lista

li {display: inline-block;}	A diferencia de establecer sólo inline, si un elemento salta de línea lo hace completamente por ser un bloque. Su texto no puede cortarse entre líneas como ocurriría si fuera inline.
ul {display: table; } li {display: table-cell; padding:10px;}	Hace que la lista se comporte como una tabla donde cada item es una celda.

Hay una curiosidad. En la imagen donde vimos las cajas podemos observar que el elemento h2 tiene un espacio libre por arriba y por debajo, que corresponde a un margin-top y margin-bottom automático que aplica el navegador por defecto para este tipo de elementos. Si escribimos `h2 {display: inline;}` comprobamos no sólo que la caja deja de ocupar todo el ancho disponible sino que los márgenes desaparecen ¿Por qué? Porque en los elementos inline no son aplicables (o se ignoran si están establecidas) ciertas propiedades como width, height, margin y float. De este modo, al cambiar el formato de caja no sólo estamos afectando a que los elementos ocupen una línea a todo lo ancho o no, sino también otras propiedades que pueden quedar desactivadas cuando se cambia el tipo de caja usando la propiedad display. Tener en cuenta también que los elementos inline sólo pueden contener otros elementos inline o texto en su interior.

Otra curiosidad. Posiblemente no obtengas nada coherente, pero comprueba qué ocurre si como css utilizas esta única declaración `{font-family: arial; display: inline-block;}`

Ejemplo de menú

Abre la página “ejemplomenu.html”. Trata de entender cómo se han creado los elementos de menú.

Básicamente se ha creado un menú (utilizando ``), que está siempre centrado automáticamente en cualquier resolución, excepto cuando ya no entre en pantalla, que habría que modificar su posicionamiento a través de CSS media queries, las cuales aprenderemos en la siguiente práctica.

Trucos:

- El `<div>` del logo tiene un `margin:auto` para el centrado automático y una anchura asignada. Para centrar lo que hay en su interior, la imagen del cangrejo y el `<h1>`, utilizo la propiedad `text-align:center`.
- Los elementos de la lista ``, tienen la propiedad `float:left`, para que se posicionen horizontalmente.
- La lista ``, tiene la propiedad `list-style:none` para quitar el punto que llevan delante los `` y la propiedad `display:inline-block` que hace que la `` sea tratada como texto. Esto será la clave para centrar la lista automáticamente, usando la propiedad `text-align:center` en el `<div>` o elemento que la contiene, en este caso el elemento nav.

Si hay algo que no entiendas, como la propiedad `box-sizing: border-box;`, búscala en internet.

Ejercicios

La finalidad de estos ejercicios es que practiquéis las propiedades vistas:

`float:left;`

`display:block;`

`display:inline-block;`

`display:none;`

`clear:both;`

`vertical-align;`

`margin:auto;`

`margin,padding,width,height,`

Tema de bordes: `border-width,style,radius;`

Además, por supuesto, de las etiquetas propias de html para estructurar una página (header,footer,aside,section,article)

Ejercicio 1

A partir de ej1.html haz los siguientes cambios:

- a. El primer artículo tendrá la imagen “responsive.png” **en el lado derecho**.
 - i. El tamaño de la imagen será de 300x200 px.
 - ii. Queremos guardar una distancia con respecto al párrafo de 10px.
- b. El segundo artículo tendrá la imagen “webDesign.jpg” **en el lado izquierdo**.
 - i. El tamaño de la imagen será de 300x200 px.
 - ii. Queremos guardar una distancia con respecto al párrafo de 10px.
- c. Los márgenes de la página serán de:
 - i. 50px para las partes superior e inferior.
 - ii. 100px para los lados izquierdo y derecho.

Este es mi primer artículo con su foto a la derecha

Una Web Responsive es aquella que es capaz de adaptarse a cualquier dispositivo donde se visualice. Las estadísticas muestran el creciente tráfico web que se genera desde dispositivos móviles, pero además Google penaliza ya en su buscador a aquellas webs que no son responsables. Si tu sitio web está obsoleto, es el momento de ponerle en manos de nuestro equipo para diseñar una web responsive con las mejores garantías.

Una Web Responsive es aquella que es capaz de adaptarse a cualquier dispositivo donde se visualice. Las estadísticas muestran el creciente tráfico web que se genera desde dispositivos móviles, pero además Google penaliza ya en su buscador a aquellas webs que no son responsables. Si tu sitio web está obsoleto, es el momento de ponerle en manos de nuestro equipo para diseñar una web responsive con las mejores garantías.

Una Web Responsive es aquella que es capaz de adaptarse a cualquier dispositivo donde se visualice. Las estadísticas muestran el creciente tráfico web que se genera desde dispositivos móviles, pero además Google penaliza ya en su buscador a aquellas webs que no son responsables. Si tu sitio web está obsoleto, es el momento de ponerle en manos de nuestro equipo para diseñar una web responsive con las mejores garantías.

Una Web Responsive es aquella que es capaz de adaptarse a cualquier dispositivo donde se visualice. Las estadísticas muestran el creciente tráfico web que se genera desde dispositivos móviles, pero además Google penaliza ya en su buscador a aquellas webs que no son responsables. Si tu sitio web está obsoleto, es el momento de ponerle en manos de nuestro equipo para diseñar una web responsive con las mejores garantías.



Este es mi primer artículo con su foto a la izquierda



Este término hace referencia al diseño web multidispositivo. Hoy en día los usuarios acceden a los sitios web desde diferentes dispositivos, ya sean ordenadores de sobremesa, portátiles, iPad, tablets Android o smartphones. Cada uno de estos dispositivos muestra el sitio web de una forma diferente, si esto ocurre así estamos ante una Web Responsive, es decir un sitio web capaz de adaptarse al dispositivo en el que se está visualizando.

Este término hace referencia al diseño web multidispositivo. Hoy en día los usuarios acceden a los sitios web desde diferentes dispositivos, ya sean ordenadores de sobremesa, portátiles, iPad, tablets Android o smartphones. Cada uno de estos dispositivos muestra el sitio web de una forma diferente, si esto ocurre así estamos ante una Web Responsive, es decir un sitio web capaz de adaptarse al dispositivo en el que se está visualizando.

Este término hace referencia al diseño web multidispositivo. Hoy en día los usuarios acceden a los sitios web desde diferentes dispositivos, ya sean ordenadores de sobremesa, portátiles, iPad, tablets Android o smartphones. Cada uno de estos dispositivos muestra el sitio web de una forma diferente, si esto ocurre así estamos ante una Web Responsive, es decir un sitio web capaz de adaptarse al dispositivo en el que se está visualizando.

Este término hace referencia al diseño web multidispositivo. Hoy en día los usuarios acceden a los sitios web desde diferentes dispositivos, ya sean ordenadores de sobremesa, portátiles, iPad, tablets Android o smartphones. Cada uno de estos dispositivos muestra el sitio web de una forma diferente, si esto ocurre así estamos ante una Web Responsive, es decir un sitio web capaz de adaptarse al dispositivo en el que se está visualizando.

Este es mi primer artículo sin foto

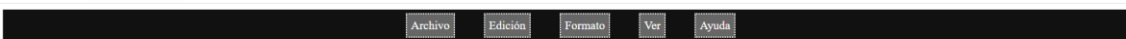
También debemos tener en cuenta los elementos de navegación de la página, el menú de navegación no se debe mostrar de la misma forma en la versión de escritorio que en la versión móvil. La razón es, por un lado y de nuevo, el tamaño de la pantalla, y por otro, que para hacer click, en teléfonos táctiles, utilizamos nuestros dedos y no un puntero. En un Smartphone la pantalla es bastante más pequeña que la de un ordenador de escritorio, un portátil o una tablet. Esta diferencia de tamaños nos obliga a que, por ejemplo, si un menú de navegación se le diseñó de forma horizontal, en este caso se mostrar de forma vertical. Además, como utilizamos nuestros dedos tenemos que diseñar botones lo suficientemente grandes para que puedan hacer click con facilidad. En relación al contenido hay que señalar además que cuanto más pequeña es la pantalla más relevancia adquieren los contenidos en comparación con los elementos de navegación. Es decir, los elementos de navegación cambian de posición para dejar paso al contenido, el usuario debe encontrarse con contenido útil cuando entra en la versión móvil de un sitio web.

También debemos tener en cuenta los elementos de navegación de la página, el menú de navegación no se debe mostrar de la misma forma en la versión de escritorio que en la versión móvil. La razón es, por un lado y de nuevo, el tamaño de la pantalla, y por otro,

group-therapy-con...zip webDesign.jpg responsive.png Mostrar todo

Ejercicio 2

A partir de ej2.html haz los cambios oportunos para que tenga este aspecto:



Ampliado:



Especificaciones:

- No se debe tocar nada en el html.
- Los colores utilizados son: #111111 y #666666.
- Márgenes interiores y exteriores de 5px.
- Margen derecho de cada elemento de menú: 30px

Ejercicio 3

Crea una página web que luzca similar a la imagen siguiente. Ten en cuenta que tienes que hacer uso de las propiedades vistas en los ejercicios:

Inicio de sesión

Inicio de sesión

[Olvidé la pass](#)

Ejercicio 4

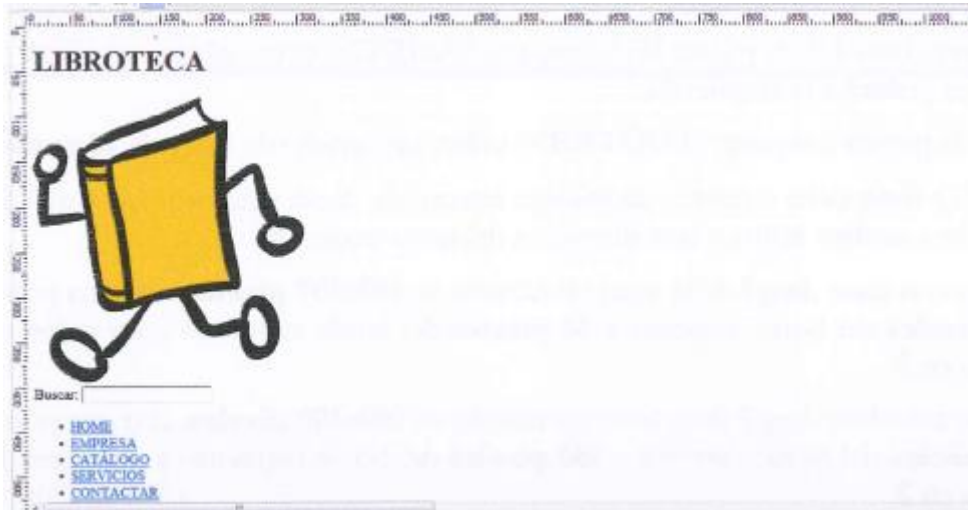
Crear una estructura de página como la que se muestra a continuación. Utiliza la imagen “fondo.jpg”:



Ejercicio final

Antes de nada, abrir librería.html y realizar las modificaciones pertinentes para que cumpla con las especificaciones de html5 en cuanto a nombres de las etiquetas (header, footer, aside, section, article,...)

En este ejercicio se trata de aplicar una serie de estilos a la página, que es una página dedicada a la venta y noticias de libros de lectura. Deberá crear una hoja de estilos externa para trabajar con el comportamiento del modelo caja visto.



Realice las modificaciones para que luzca como en la imagen siguiente:

