

UT7. CSS AVANZADO

| | |
|--------------------------|---|
| 1.- Modelo de cajas..... | 2 |
| 2.- Enlaces..... | 4 |
| 3.- Listas..... | 4 |
| 4.- Tablas..... | 4 |
| 5.- Formularios..... | 6 |
| 6.- Layout..... | 8 |

1.- Modelo de Cajas

Todos los elementos HTML pueden ser considerados como cajas. En CSS, el término "modelo de caja" se usa cuando se habla sobre el diseño y el diseño. El modelo de caja CSS es esencialmente una caja que se envuelve alrededor de los elementos HTML, y se compone de: márgenes, bordes, el relleno y el contenido real. El modelo de caja nos permite colocar un borde alrededor de elementos y elementos del espacio en relación con otros elementos.

La imagen siguiente muestra el modelo de caja:



Explicación de las diferentes partes:

- **Margen** - Borra un área alrededor de la frontera. El margen no tiene un color de fondo, es completamente transparente
- **Border** - Un borde que va alrededor del relleno y contenido. La frontera se ve afectado por el color de fondo de la caja
- **Relleno** - Borra un área alrededor del contenido. El relleno se ve afectado por el color de fondo de la caja
- **Contenido** - El contenido de la caja, en la que aparecen el texto y las imágenes

Con el fin de establecer el ancho y el alto de un elemento correctamente en todos los navegadores, lo que necesita saber cómo funciona el modelo de caja.

Ancho y alto de un elemento

Importante: Al configurar las propiedades de ancho y alto de un elemento con CSS, que acaba de establecer el ancho y la altura del **área de contenido**.

Para calcular el tamaño de un elemento, también debe agregar el relleno, bordes y márgenes. La anchura total de un elemento debe calcularse así:

Ancho total elemento = ancho + padding-left + relleno derecho + borde izquierdo + margen derecho + margen izquierdo + margen derecho

La altura total de un elemento debe calcularse así:

Altura total elemento = altura + top padding + padding-bottom + top + frontera borde inferior + top + margen margen inferior

La anchura total del elemento en el ejemplo siguiente es 300px:

```
width:250px;  
padding:10px;  
border:5px solid gray;  
margin:10px;
```

Vamos a hacer los cálculos:

```
250px (anchura)  
+ 20px (izquierda + relleno derecho)  
+ 10px (izquierdo + margen derecho)  
+ 20px (margen izquierda + derecha)  
= 300px
```

Suponga que sólo 250px espacio tenía. Vamos a hacer un elemento con un ancho total de 250 píxeles:

Ejemplo

```
width:220px;  
padding:10px;  
border:5px solid gray;  
margin:0px;
```

ejemplo 1:

```
<html>
<head>
<style>
div.ex
{
width:240px;
height:80px;
border:5px solid gray;
}
.img
{
}
</style>
</head>
<body>

<div class="ex">La imagen de encima tiene un ancho de 250 px.
La anchura total de esta caja también es de 250 px y la altura es de
90 px.</div>
</body>
</html>
```

Margen y Relleno de un elemento

Los márgenes pueden especificarse de manera individual o en una regla abreviada. Además, la regla abreviada sigue permitiendo controlar cada uno de los bordes que rodean un objeto. Los valores válidos suelen

especificarse en unidades px o em (píxeles o ems). En las hojas de estilo específicas de impresión pueden utilizarse las unidades in, cm, o pt (pulgadas, centímetros o puntos).

En todos los casos, % (porcentaje) es un valor válido, pero se debe utilizar con cuidado porque estos valores se calculan como una proporción de la anchura del elemento padre, y una previsión no suficientemente precisa de valores podría tener consecuencias no deseadas. Esta dificultad se explica con más detalle cuando se habla del modelo de caja de CSS.

Exceptuando las imágenes, los elementos en línea carecen de márgenes y no aceptan valores de margen.

Ejemplo:

```
<html>
<head>
<style>
div.ex
{
width:240px;
height:80px;
border:5px solid gray;
margin: 10px 30px ;
}
.img
{
}
</style>
</head>
<body>

<div class="ex">La imagen de encima tiene un ancho de 250 px.
La anchura total de esta caja también es de 250 px y la altura es de
90 px.</div>
</body>
</html>
```

Si queremos que quede la foto y el cuadro de texto alineados utilizando márgenes el ejemplo ha de quedar así:

Ejemplo 2:

```
<html>
<head>
<style>
div.ex
{
width:240px;
height:80px;
border:5px solid gray;
margin: 10px 0px ;
}
.img
{
}
</style>
</head>
<body>

<div class="ex">La imagen de encima tiene un ancho de 250 px.
La anchura total de esta caja también es de 250 px y la altura es de
90 px.</div>
</body>
</html>
```

Añadir un borde a un elemento: propiedades de borde

Hay una propiedad de border (borde) abreviada, pero sólo es útil cuando se quiere proporcionar un borde completo y consistente alrededor de los cuatro lados de un elemento. También es posible establecer el peso (anchura), estilo y color de cualquiera de los cuatro bordes de un elemento mediante cualquier combinación significativa de las propiedades siguientes:

- border-width
- border-style
- border-color
- border-top
- border-top-width
- border-top-style

- border-top-color
- border-right
- border-right-width
- border-right-style
- border-right-color
- border-bottom
- border-bottom-width
- border-bottom-style
- border-bottom-color
- border-left
- border-left-width
- border-left-style
- border-left-color

Las propiedades border-width

Estas propiedades se comportan exactamente como sería de esperar: asignan peso explícito a uno o más bordes.

La propiedad abreviada border-width (anchura de borde) acepta valores con la misma notación que la propiedad abreviada de margin (margen), pero no acepta valores porcentuales. Puede ser que en el futuro debáis redactar una regla como la siguiente:

```
td {  
    border-width: 1px 0 0 1px;  
}
```

ejemplo

```
<html>  
<head>  
<style>
```

```
p.uno
{
border-style:solid;
border-width:5px 1px 2px 5px;
}
p.dos
{
border-style:solid;
border-width:medium;
}
p.tres
{
border-style:solid;
border-width:1px 3px 5px 7px;
}
</style>
</head>
<body>
<p class="uno">Parte del texto.</p>
<p class="dos">Parte del texto.</p>
<p class="tres">Parte del texto.</p>
<p><b>Nota: </b> La propiedad " border-width " no funciona si se
utiliza solo. Utiliza la propiedad " border-style " para establecer las
fronteras primero.</p>
</body>
</html>
```

Las propiedades border-style

Las propiedades border-style (estilo de borde) suelen aceptar cualquiera de los valores siguientes:

- dashed. La longitud de las secciones de línea y la cantidad de espacio en blanco entre ellas las determina el navegador.
- dotted. La cantidad de espacio en blanco entre los puntos (que pueden tener cualquier forma con una proporción de 1) la determina el navegador.

- **double.** La anchura proporcionada se dividirá en tercios y se asignará en el orden lleno-negativo-lleño.
- **groove.** Se representará un outset inmediatamente dentro de y junto a un inset.
- **inset.** Al borde se le aplicará una sombra para que parezca que el elemento al que se aplica está incrustado en el lienzo.
- **none.** Equivalente a especificar una -width (anchura) de cero.
- **outset.** Al borde se le aplicará una sombra para que parezca que el elemento al que se aplica sobresale del lienzo.
- **ridge.** Se representará un inset inmediatamente dentro de y junto a un outset.
- **solid.** El borde es una línea continua sin sombra.

Cuando se utiliza la propiedad abreviada **border-style** (estilo de borde), ésta acepta hasta cuatro valores que se aplican de la misma manera que los valores abreviados **margin** (margen).

La práctica de oscurecer un borde (en lugar de omitirlo) se controla con las propiedades **-color**.

Las propiedades **border-color**

Para terminar, es posible establecer cualquier color en cualquier borde concreto, tanto con una única propiedad como las que se acaban de enumerar, como con la propiedad abreviada **border-color** (color de borde).

Ejemplo 1

```
<html>  
<head>  
<style>  
p.uno
```

```
{
border-style:solid;
border-width:5px 1px 2px 5px;
border-color:red;
}
p.dos
{
border-style:solid;
border-width:medium;
border-color:green;
}
p.tres
{
border-style:solid;
border-width:1px 3px 5px 7px;
}
</style>
</head>
<body>
<p class="uno">Parte del texto.</p>
<p class="dos">Parte del texto.</p>
<p class="tres">Parte del texto.</p>
<p><b>Nota: </b> La propiedad " border-width " no funciona si se
utiliza solo. Utiliza la propiedad " border-style " para establecer las
fronteras primero.</p>
</body>
</html>
```

Ejemplo 2

```
<html>
<head>
<style>
p.uno
{
border-style:dashed;
border-width:5px 1px 2px 5px;
border-color:red;
}
p.dos
```

```
{
border-style:dotted;
border-width:medium;
border-color:green;
}
p.tres
{
border-style:double;
border-width:3px 5px 7px 9px;
border-color:orange;
}
</style>
</head>
<body>
<p class="uno">Parte del texto.</p>
<p class="dos">Parte del texto.</p>
<p class="tres">Parte del texto.</p>
<p><b>Nota: </b> La propiedad " border-width " no funciona si se
utiliza solo. Utiliza la propiedad " border-style " para establecer las
fronteras primero.</p>
</body>
</html>
```

Ejemplo 3

```
<html>
<head>
<style>
p.uno
{
border-style:groove;
border-width:5px 1px 2px 5px;
border-color:red;
}
p.dos
```

```
{
border-style:inset;
border-width:medium;
border-color:green;
}
p.tres
{
border-style:outset;
border-width:3px 5px 7px 9px;
border-color:orange;
}
</style>
</head>
<body>
<p class="uno">Parte del texto.</p>
<p class="dos">Parte del texto.</p>
<p class="tres">Parte del texto.</p>
<p><b>Nota: </b> La propiedad " border-width " no funciona si se
utiliza solo. Utiliza la propiedad " border-style " para establecer las
fronteras primero.</p>
</body>
</html>
```

Ejemplo 4

```
<html>
<head>
<style>
p.uno
{
border-style:ridge;
border-width:5px 1px 2px 5px;
border-color:red;
}
p.dos
```

```
{
border-style:groove;
border-width:medium;
border-color:green;
}
p.tres
{
border-style:ridge;
border-width:3px 5px 7px 9px;
border-color:orange;
}
</style>
</head>

<body>
<p class="uno">Parte del texto.</p>
<p class="dos">Parte del texto.</p>
<p class="tres">Parte del texto.</p>
<p><b>Nota: </b> La propiedad " border-width " no funciona si se
utiliza solo. Utiliza la propiedad " border-style " para establecer las
fronteras primero.</p>
</body>

</html>
```

propiedades padding

Encontraréis elementos con colores de fondo de tonos secundarios o destacados que necesitan canales entre el contenido y los márgenes. También puede ser que necesitéis incluir espacio entre los bordes y el texto que hay cerca.

En estos casos y en muchos otros os serán muy útiles las propiedades padding (relleno), padding-top(relleno superior), padding-right (relleno derecha), padding-bottom (relleno inferior) y padding-

left(relleno izquierda). Estas propiedades insertan espacio negativo entre los márgenes o bordes de un elemento y su contenido. Podéis ver la figura 1 anterior como claro ejemplo de la relación entre márgenes, bordes y relleno.

Ejemplo

```
<html>
<head>
<style type="text/css">
p
{
border-width: 7px 7px 7px 7px;
border-style: ridge;
border-color: pink;
padding-top: 1cm;
padding-right: 2cm;
padding-left: 3cm;
padding-bottom: 4cm;
}
</style>
</head>
<body>
<p>Este texto tiene un borde alrededor.<br>
El espacio que hay entre el texto y el borde tiene diferentes anchos y
está dado por la propiedad padding.</p>
<p> Este parrafo es una segunda caja con las mismas características
que el párrafo anterior</p>
</body>
</html>
```

2.- Enlaces

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. Utilizando las propiedades text-decoration y font-weight.

CSS también permite aplicar diferentes estilos a un mismo enlace en función de su estado. De esta forma, es posible cambiar el aspecto de un

enlace cuando por ejemplo el usuario pasa el ratón por encima o cuando el usuario pincha sobre ese enlace.

Como con los atributos `id` o `class` no es posible aplicar diferentes estilos a un mismo elemento en función de su estado, CSS introduce un nuevo concepto llamado *pseudo-clases*. En concreto, CSS define las siguientes cuatro pseudo-clases:

- `:link`, aplica estilos a los enlaces que apuntan a páginas o recursos que aún no han sido visitados por el usuario.
- `:visited`, aplica estilos a los enlaces que apuntan a recursos que han sido visitados anteriormente por el usuario. El historial de enlaces visitados se borra automáticamente cada cierto tiempo y el usuario también puede borrarlo manualmente.
- `:hover`, aplica estilos al enlace sobre el que el usuario ha posicionado el puntero del ratón.
- `:active`, aplica estilos al enlace que está pinchando el usuario. Los estilos sólo se aplican desde que el usuario pincha el botón del ratón hasta que lo suelta, por lo que suelen ser unas pocas décimas de segundo.

Como se sabe, por defecto los navegadores muestran los enlaces no visitados de color azul y subrayados y los enlaces visitados de color morado. Las pseudo-clases anteriores permiten modificar completamente ese aspecto por defecto y por eso casi todas las páginas las utilizan.

Cuando se aplican varias pseudo-clases diferentes sobre un mismo enlace, se producen colisiones entre los estilos de algunas pseudo-clases. Si se pasa por ejemplo el ratón por encima de un enlace visitado, se aplican los estilos de las pseudo-clases `:hover` y `:visited`. Si el usuario pincha sobre un enlace no visitado, se aplican las pseudo-clases `:hover`, `:link` y `:active` y así sucesivamente.

Si se definen varias pseudo-clases sobre un mismo enlace, el único orden que asegura que todos los estilos de las pseudo-clases se aplican de forma coherente es el siguiente: `:link`, `:visited`, `:hover` y `:active`.

Ejemplo

```
<html>
<head>
<style type="text/css">
a:link {color:#FF0000;}
a:visited {color:#00FF00;}
a:hover {color:#FF00FF;}
a:active {color:#0000FF;}
p
{
border-width: 7px 7px 7px 7px;
border-style: ridge;
border-color:pink;
padding-top:1cm;
padding-right:2cm;
padding-left:3cm;
padding-bottom:4cm;
}
</style>
</head>
<body>
<p> <b><a href="ejer2.html" target="_blank">ejer2.html</b></a>Este
texto tiene un borde alrededor.<br>
El espacio que hay entre el texto y el borde tiene diferentes anchos y
está dado por la propiedad padding.</p>
<p> Este parrafo es una segunda caja con las mismas características
que el párrafo anterior</p>
</body>
</html>
```

Atributos:

Color: se utiliza para poner el color del link

Text-decoration: se utiliza para poner el subrayado del link. Puede ser ninguno(none), subrayado(underline), tachado(overline), sobrelínea(line-through)

background-color: se utiliza para poner el color de fondo en los enlaces.

Ejemplo:

```
<html>
<head>
<style type="text/css">
a:link {color:#FF0000; text-decoration:line-through;background-
color:#B2FF99;}
a:visited {color:#00FF00; text-decoration: overline;background-
color:#FFFF85;}
a:hover {color:#FF00FF; text-decoration: underline;background-
color:#FF704D;}
a:active {color:#0000FF; text-decoration: overline;background-
color:#FF704D;color}
p
{
border-width: 7px 7px 7px 7px;
border-style: ridge;
border-color:pink;
padding-top:1cm;
padding-right:2cm;
padding-left:3cm;
padding-bottom:4cm;
}
</style>
</head>
<body>
<p><b><a href="ejer2.html" target="_blank">ejer2.html </b></a><br>
Este texto tiene un borde alrededor.<br>
El espacio que hay entre el texto y el borde tiene diferentes anchos y
está dado por la propiedad padding.</p>
<p> Este parrafo es una segunda caja con las mismas características
que el párrafo anterior</p>
<p> y este la tercera caja</p>
</body>
</html>
```

3.- Listas

Son igual que en html lo único que el tipo de círculo cuadrado...que queremos que aparezca en la lista se define en el style del css.

Ejemplos:

Ejemplo1:

```
<html>
<head>
  <style type="text/css">
ul.a {list-style-type:circle;}
  </style>
</head>
<body>
<UL class="a">
<LI> Mamíferos</LI>
<LI> Peces</LI>
<LI> Aves</LI>
</UL>
</body>
</html>
```

Ejemplo 2:

```
<html>
<head>
<style type="text/css">
ul.a {list-style-type:circle;}
ul.b {list-style-type:square;}
</style>
</head>
<body>
<UL class="a">
<LI> Mamíferos</LI>
<LI> Peces</LI>
  <UL class="b">
```

```
        <LI> Sardina</LI>
        <LI> Bacalao</LI>
    </UL>
<LI> Aves</LI>
</UL>
</body>
</html>
```

Ejemplo 3:

```
<html>
<head>
    <style type="text/css">
ol.a {list-style-type:lower-roman;}
    </style>
</head>
<body>
<ol class="a" >
<li>Rodriguez Pablo</li>
<li>Gonzalez Raul</li>
<li>Lopez Hector</li>
</ol>
</body>
</html>
```

Ejemplo 4:

```
<html>
<head>
    <style type="text/css">
ol.a {list-style-type:roman;}
    </style>
</head>
<body>
<ol class="a">
<li>Rodriguez Pablo</li>
<li>Gonzalez Raul</li>
<li>Lopez Hector</li>
</ol>
```

```
</body>  
</html>
```

Ejemplo 5

```
<html>  
<head>  
<style type="text/css">  
ol.a {list-style-type:upper-alpha;}  
</style>  
</head>  
<body>  
<ol class="a">  
<li>Rodriguez Pablo</li>  
<li>Gonzalez Raul</li>  
<li>Lopez Hector</li>  
</ol>  
</body>  
</html>
```

Ejemplo 6

```
<html>  
<head>  
<style type="text/css">  
ol.a {list-style-type:upper-alpha;}  
ol.b {list-style-type:lower-roman;}  
</style>  
</head>  
<body>  
<ol class="a">  
<li>Argentina</li>  
  <ol class="b">  
    <li>La Nación</li>  
    <li>Clarín</li>  
    <li>Página 12</li>
```

```
</ol>

<li>España</li>
  <ol class="b">
    <li>Abc</li>
    <li>El pais</li>
  </ol>
</ol>
</html>
```

Ejemplo 7

```
<html>
<head>
<style type="text/css">
ol.a {list-style-type:upper-alpha;}
ul.a {list-style-type:circle;}
ul.b {list-style-type:square;}
</style>
</head>
<body>
<ol class="a">
<li>Argentina</li>
  <ul class="a">
    <li> La Nación</li>
    <li>Clarín</li>
    <li>Página 12</li>
```

```
</ul>
<li>España</li>
  <ul class="b">
    <li>Abc</li>
    <li>El pais</li>
  </ul>
</ol>
</html>
```

Listas con imágenes

Ejemplo

```
<html>
<head>
  <style type="text/css">
    ol { list-style-image:url('bosque2.jpg');}
    ul.a {list-style-image:url('muelle2.jpg');}
    ul.b{list-style-image:url('floresexoticas2.jpg');}
  </style>
</head>
<body>
  <ol>
    <li>Argentina</li>
      <ul class="a">
        <li> La Nación</li>
```

```
<li>Clarín</li>
<li>Página 12</li>
</ul>

<li>España</li>
  <ul class="b">
    <li>Abc</li>
    <li>El pais</li>
  </ul>
</ol>
</html>
```

4.- Tablas

Propiedad caption-side

Nos permite ubicar el título de la tabla por encima o por debajo de la misma. La alineación horizontal del mismo se puede establecer con la propiedad **text-align**.

Ejemplo 1:

```
<html>
<head>
<title> ejemplo de tabla </title>
<style type="text/css">
caption{caption-side: bottom}
</style>
</head>
```

```
<body>
<table border="3" align=center width =30% height =30% cellpadding=10
cellspacing=8                                bordercolor=#40FF00
background="G:\multimedia\imagenes\dock.jpg" >
<caption>PLANING</caption>
<tr>
    <td>Sabado</td>
    <td>Domingo</td>
</tr>
<tr>
    <td>Ir al cine</td>
    <td> Dar un paseo </td>
</tr>
<tr>
    <td>Estudiar redes</td>
    <td> Estudiar marcas </td>
</tr>
</table>
</body>
</html>
```

Ejemplo 2:

```
<html>
<head>
<title> ejemplo de tabla </title>
<style type="text/css">
caption{caption-side: top}
</style>
</head>
<body>
<table border="3" align=center width =30% height =30% cellpadding=10
cellspacing=8                                bordercolor=#40FF00
background="G:\multimedia\imagenes\dock.jpg" >
<caption>PLANING</caption>
<tr>
    <td>Sabado</td>
    <td>Domingo</td>
</tr>
<tr>
```



```
<td>Ir al cine</td>
<td> Dar un paseo </td>
</tr>
<tr>
<td>Estudiar redes</td>
<td> Estudiar marcas </td>
</tr>
</table>
</body>
</html>
```

Propiedad border-spacing

La propiedad **border-spacing** especifica la separación entre celdas adyacentes. Si especificamos un solo valor, este actúa sobre toda la tabla. Si especificamos 2 valores el primero define la separación horizontal y el segundo la vertical. Es similar al cellpadding de html.

Ejemplo

```
<html>
<head>
<title> ejemplo de tabla </title>
<style type="text/css">
caption{caption-side: bottom}
table{border-spacing: 10px 20px;}
</style>
</head>
<body>
<table border="3" align=center width =30% height =30%
cellpadding=10 bordercolor=#40FF00
background="G:\multimedia\imagenes\dock.jpg" >
<caption>PLANING</caption>
<tr>
<td>Sabado</td>
<td>Domingo</td>
</tr>
```

```
<tr>
  <td>Ir al cine</td>
  <td> Dar un paseo </td>
</tr>
<tr>
  <td>Estudiar redes</td>
  <td> Estudiar marcas </td>
</tr>
</table>
</body>
</html>
```

Propiedad border-collapse

Esta propiedad indica el modelo de los bordes. Tiene dos posibles valores: Collapse para plegado o separate para separado.

Ejemplo

```
<html>
<head>
<title> ejemplo de tabla </title>
<style type="text/css">
  table.plegado
  {
    border-collapse: collapse
  }
  table.separado
  {
    border-collapse: separate
  }
</style>
</head>
</html>
```

```
</style>
</head>
<body>
  <table class="plegado" border="1" align="center" >
    <tr>
      <td>Sabado</td>
      <td>Domingo</td>
    </tr>
    <tr>
      <td>Ir al cine</td>
      <td> Dar un paseo </td>
    </tr>
    <tr>
      <td>Estudiar redes</td>
      <td> Estudiar marcas </td>
    </tr>
  </table>
  <br>
  <br>
  <table class="separado" border="1" align="center" >
    <tr>
      <td>Sabado</td>
      <td>Domingo</td>
    </tr>
    <tr>
      <td>Ir al cine</td>
      <td> Dar un paseo </td>
    </tr>
    <tr>
      <td>Estudiar redes</td>
      <td> Estudiar marcas </td>
    </tr>
  </table>
</body>
</html>
```

Propiedad empty-cells

La propiedad **empty-cells** nos permite controlar la visualización de los bordes y fondos de una celda vacía. Los posibles valores para controlar las celdas vacías son: show | hide

Ejemplo

```
<html>
<head>
<title> ejemplo de tabla </title>
<style type="text/css">
    table.separado
    {
        border-collapse: separate
    }
    td.muestra{empty-cells: show}
    td.oculta{empty-cells: hide}
</style>
</head>
<body>

    <table class="separado" border="3" cellpadding=10 align=center width
=50% height =20%>
        <tr>
            <td class="oculta"> </td>
            <td>Domingo</td>
        </tr>
        <tr>
            <td>Ir al cine</td>
            <td class="oculta"> </td>
        </tr>
        <tr>
            <td>Estudiar redes</td>
            <td class="muestra"> </td>
        </tr>
    </table>
</body>
</html>
```

5.- Formularios

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

Ejemplo 1

```
<html>
<head>
<title>formulario 1</title>
<style>

.enlace {
  border: 0;
  padding: 0;
  background-color: transparent;
  color: blue;
  border-bottom: 1px solid blue;
}
</style>
</head>

<body>

<input type="button" value="Botón normal" />
<br>
<br>

<input class="enlace" type="button" value="Botón como enlace" />
</body>
</html>
```

Por defecto, los campos de texto de los formularios no incluyen ningún espacio de relleno, por lo que el texto introducido por el usuario aparece *pegado* a los bordes del cuadro de texto.

Añadiendo un pequeño **padding** a cada elemento `<input>`, se mejora notablemente el aspecto del formulario:

Ejemplo 2:

```
<html>
<head>
<title>formulario 1</title>
<style>
body {
  font-family: Arial;
```

```
    font-size:13px;
}
#divformulario {
    position:relative;
    height:124px;
    width:340px;
    margin:0 auto;
    background-color:blue;
}
#divTitulo, #lbl1, #txt1, #lbl2, #txt2, #btnEnviar {
    position:absolute;
}
#divTitulo {
    height:16px;
    width:340px;
    color:yellow;
    text-align:center;
    font-weight:bold;
    background-color:red;
}
#lbl1, #lbl2 {
    width:70px;
    background-color:lightgray;
}
#txt1, #txt2 {
    width:200px;
    border:1px solid black;
}
#lbl1 {
    top:33px;
    left:30px;
}
#txt1 {
    top:30px;
    left:104px;
}
#lbl2 {
    top:63px;
    left:30px;
}
#txt2 {
    top:60px;
    left:104px;
}
```

```
#btnEnviar {
    width:110px;
    top:90px;
    left:120px;
    cursor:pointer;
background-color:pink;
}

</style>
</head>

<body>
<div id="divformulario">
    <div id="divTitulo">ACCESO</div>
    <form id="form1" name="form1" action="#" method="post">
        <label id="lbl1" name="lbl1" for="txt1">Nombre:</label>-
        <input id="txt1" name="txt1" type="text" value="Introduce el
nombre..." />

        <label id="lbl2" name="lbl2" for="txt2">Apellidos:</label>
        <input id="txt2" name="txt2" type="text" value="Introduce los
apellidos..." />

        <input type="submit" id="btnEnviar" name="btnEnviar" />
    </form>
</div>

</body>
</html>
```

Ejemplo 3

```
<html>
<head>
<title>formulario 1</title>

</head>

<body>
<form>
```

```
<fieldset>
  <legend>Alta en el servicio</legend>

  <label for="nombre">Nombre</label>
  <input type="text" id="nombre" />
  <br>

  <label for="apellidos">Apellidos</label>
  <input type="text" id="apellidos" size="50" />
  <br>
  <label for="dni">DNI</label>
  <input type="text" id="dni" size="10" maxlength="9" />
  <br>
  <label for="contrasena">Contraseña</label>
  <input type="password" id="contrasena" />
  <br>
  <input class="btn" type="submit" value="Dar de alta" />
</fieldset>
</form>

</body>
</html>
```

En primer lugar, se muestran los elementos `<label>` como elementos de bloque, para que añadan una separación para cada campo del formulario. Además, se añade un margen superior para no mostrar juntas todas las filas del formulario:

El botón del formulario también se muestra como un elemento de bloque y se le añade un margen para darle el aspecto final deseado:

Ejemplo 4

```
<html>
<head>
<title>formulario 1</title>

</head>
<style type="text/css">
label {
  display: block;
  margin: 3px;
}
.btn {
  display: block;
```



```
margin: 2px 0;
}
</style>

<body>
<form>
<fieldset>
  <legend>Alta en el servicio</legend>

  <label for="nombre">Nombre</label>
  <input type="text" id="nombre" />

  <label for="apellidos">Apellidos</label>
  <input type="text" id="apellidos" size="50" />

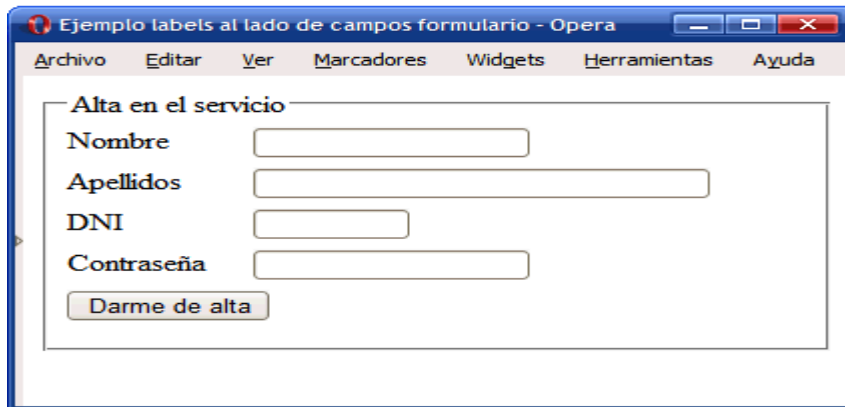
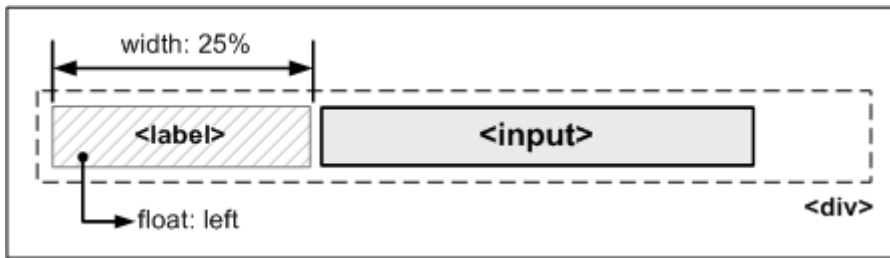
  <label for="dni">DNI</label>
  <input type="text" id="dni" size="10" maxlength="9" />

  <label for="contrasena">Contraseña</label>
  <input type="password" id="contrasena" />

  <input class="btn" type="submit" value="Dar de alta" />
</fieldset>
</form>

</body>
</html>
```

En ocasiones, es más útil mostrar todos los campos del formulario con su `<label>` alineada a la izquierda y el campo del formulario a la derecha de cada `<label>` para ello no es necesario crear una tabla y controlar la anchura de sus columnas para conseguir una alineación perfecta. Sin embargo, sí que es necesario añadir un nuevo elemento (por ejemplo un `<div>`) que encierre a cada uno de los campos del formulario (`<label>` y `<input>`). El esquema de la solución propuesta es el siguiente



Ejemplo 5

```
<html>
<head>
<title>formulario 1</title>

</head>
<style type="text/css">
div {
    margin: 5px 0;
}
div label {
    width: 15%;
    float: left;
}
</style>

<body>
<form>
<fieldset>
    <legend>Alta en el servicio</legend>

    <div>
        <label for="nombre">Nombre</label>
```

```
<input type="text" id="nombre" />
</div>

<div>
  <label for="apellidos">Apellidos</label>
  <input type="text" id="apellidos" size="50" />
</div>
<div>
  <label for="dni">DNI</label>
  <input type="text" id="dni" size="10" maxlength="9" />
</div>
<div>

  <label for="contrasena">Contraseña</label>
  <input type="password" id="contrasena" />
</div>
<div>
  <input class="btn" type="submit" value="Darme de alta" />
</div>
</fieldset>
</form>

</body>
</html>
```

Los formularios complejos con decenas de campos pueden ocupar mucho espacio en la ventana del navegador. Además del uso de pestañas para agrupar los campos relacionados en un formulario, también es posible mostrar el formulario a dos columnas, para aprovechar mejor el espacio.

Ejemplo formulario 2 columnas - Opera

Archivo Editar Ver Marcadores Widgets Herramientas Ayuda

Alta en el servicio

Nombre

Apellidos

DNI

Contraseña

Datos de contacto

Telefono

Email

Dirección

Código Postal

La solución consiste en aplicar la siguiente regla CSS a los `<fieldset>` del formulario:

```
.izquierda {  
float:left;  
width: 47%;  
}
```

```
.derecha {  
float:right;  
width: 47%;  
}
```

Ejemplo 6

```
<html>  
<head>  
<title>formulario 1</title>  
</head>  
<style type="text/css">  
  .izquierda {float:left;  
width: 47%;  
  .derecha {float:right;  
width: 47%;  
}
```

```
</style>
<body>
<form>
<fieldset class="izquierda">
  <legend>Alta en el servicio</legend>
  <label for="nombre">Nombre</label>
<br>
  <input type="text" id="nombre" />
<br>
  <label for="apellidos">Apellidos</label>
<br>
  <input type="text" id="apellidos" size="50" />
<br>
  <label for="dni">DNI</label>
<br>
  <input type="text" id="dni" size="10" maxlength="9" />
<br>
  <label for="contrasena">Contraseña</label>
<br>
  <input type="password" id="contrasena" />
<br>
</fieldset>
<fieldset class="derecha">
  <legend>Datos de contacto</legend>
  <label for="telefono">Telefono</label>
<br>
  <input type="text" id="telefono" />
<br>
  <label for="email">Email</label>
<br>
  <input type="text" id="email" size="50" />
<br>
  <label for="direccion">Direccion</label>
<br>
  <input type="text" id="direccion" size="10" maxlength="9" />
<br>
  <label for="codigo postal">Codigo Postal</label>
<br>
  <input type="text" id="codigo postal" />
<br>
</fieldset>
<input class="btn" type="submit" value="Dar me de alta" />
</form>
</body>
```

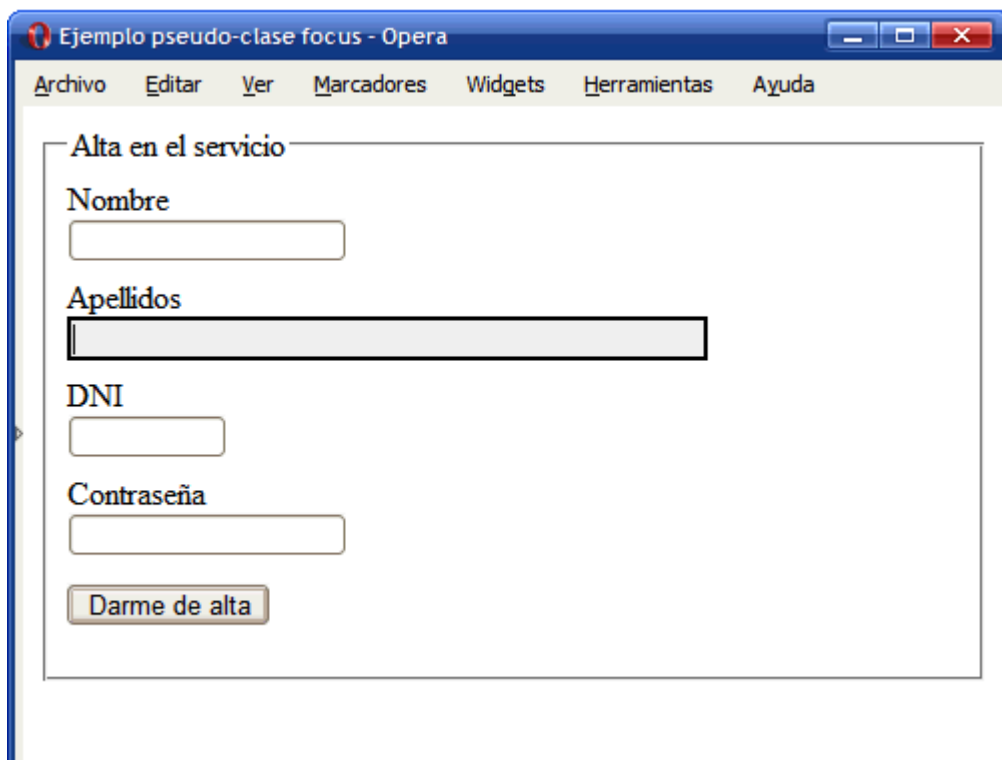
</html>

Si el formulario es muy complejo, puede ser útil agrupar los <fieldset> de cada fila mediante elementos <div>.

Resaltar el campo seleccionado

Una de las mejoras más útiles para los formularios HTML consiste en resaltar de alguna forma especial el campo en el que el usuario está introduciendo datos. Para ello, CSS define la pseudo-clase `:focus`, que permite aplicar estilos especiales al elemento que en ese momento tiene el *foco* o atención del usuario.

La siguiente imagen muestra un formulario que resalta claramente el campo en el que el usuario está introduciendo la información:



Añadiendo la pseudo-clase `:focus` después del selector normal, el navegador se encarga de aplicar esos estilos cuando el usuario activa el elemento

```
input:focus {
```

```
border: 2px solid #000;
```

```
background: #F3F3F3;
}
```

Ejemplo 7

```
<html>
<head>
<title>formulario 1</title>

</head>
<style type="text/css">
label {
    display: block;
    margin: 3px;
}
.btn {
    display: block;
    margin: 2px 0;
}
input:focus {
    border: 2px solid #000;
    background: #F3F3F3;
}
</style>

<body>
<form>
<fieldset>
    <legend>Alta en el servicio</legend>

    <label for="nombre">Nombre</label>
    <input type="text" id="nombre" />

    <label for="apellidos">Apellidos</label>
    <input type="text" id="apellidos" size="50" />

    <label for="dni">DNI</label>
```

```
<input type="text" id="dni" size="10" maxlength="9" />

<label for="contrasena">Contraseña</label>
<input type="password" id="contrasena" />

<input class="btn" type="submit" value="Dar de alta" />
</fieldset>
</form>

</body>
</html>
```

6.- Layout

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.

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 las siguientes:

- **Mantenimiento:** una página diseñada exclusivamente con CSS es mucho más fácil de mantener que una página diseñada con tablas. Cambiar el aspecto de una página creada con CSS es tan fácil como modificar unas pocas reglas en las hojas de estilos. Sin embargo, realizar la misma modificación en una página creada con tablas supone un esfuerzo muy superior y es más probable cometer errores.
- **Accesibilidad:** las páginas creadas con CSS son más accesibles que las páginas diseñadas con tablas. De hecho, los navegadores que utilizan las personas discapacitadas (en especial las personas invidentes) pueden tener dificultades con la estructura de las páginas complejas creadas con tablas HTML. No obstante, diseñar una página web exclusivamente con CSS no garantiza que la página sea accesible.

- **Velocidad de carga:** el código HTML de una página diseñada con tablas es mucho mayor que el código de la misma página diseñada exclusivamente con CSS, por lo que tarda más tiempo en descargarse. En cualquier caso, si el usuario accede al sitio con una conexión de banda ancha y la página es de un tamaño medio o reducido, las diferencias son casi imperceptibles.
- **Semántica:** aunque resulta obvio, las tablas HTML sólo se deben utilizar para mostrar datos cuya información sólo se entiende en forma de filas y columnas. Utilizar tablas para crear la estructura completa de una página es tan absurdo como utilizar por ejemplo la etiqueta `` para crear párrafos de texto.

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.

En este capítulo se muestra cómo crear algunas de las estructuras o *layouts* más habituales de los diseños web utilizando exclusivamente CSS.

Centrar una página horizontalmente

A medida que aumenta el tamaño y la resolución de las pantallas de ordenador, se hace más difícil diseñar páginas que se adapten al tamaño de la ventana del navegador. El principal reto que se presenta con resoluciones superiores a 1024 x 768 píxel, es que las líneas de texto son demasiado largas como para leerlas con comodidad. Por ese motivo, normalmente se opta por diseños con una anchura fija limitada a un valor aceptable para mantener la legibilidad del texto.

Por otra parte, los navegadores alinean por defecto las páginas web a la izquierda de la ventana. Cuando la resolución de la pantalla es muy grande, la mayoría de páginas de anchura fija alineadas a la izquierda parecen muy estrechas y provocan una sensación de vacío.

La solución más sencilla para evitar los grandes espacios en blanco consiste en crear páginas con una anchura fija adecuada y centrar la página horizontalmente respecto de la ventana del navegador. Las siguientes imágenes muestran el aspecto de una página centrada a medida que aumenta la anchura de la ventana del navegador.

UT7. CSS AVANZADO

Utilizando la propiedad `margin` de CSS, es muy sencillo centrar una página web horizontalmente. La solución consiste en agrupar todos los contenidos de la página en un elemento `<div>` y asignarle a ese `<div>` unos márgenes laterales automáticos. El `<div>` que encierra los contenidos se suele llamar `contenedor` (en inglés se denomina `wrapper` o `container`):

```
#contenedor {  
  
    width: 300px;  
  
    margin: 0 auto;  
  
}
```

```
<body>  
  
    <div id="contenedor">  
  
        ...  
  
    </div>  
  
</body>
```

Como se sabe, el valor `0 auto` significa que los márgenes superior e inferior son iguales a `0` y los márgenes laterales toman un valor de `auto`. Cuando se asignan márgenes laterales automáticos a un elemento, los navegadores centran ese elemento respecto de su elemento padre. En este ejemplo, el elemento padre del `<div>` es la propia página (el elemento `<body>`), por lo que se consigue centrar el elemento `<div>` respecto de la ventana del navegador.

Modificando ligeramente el código CSS anterior se puede conseguir un diseño dinámico o *líquido* (también llamado *fluido*) que se adapta a la anchura de la ventana del navegador y permanece siempre centrado:

```
#contenedor {  
  
    width: 70%;  
  
    margin: 0 auto;  
  
}
```

Estableciendo la anchura del elemento contenedor mediante un porcentaje, su anchura se adapta de forma continua a la anchura de la ventana del navegador. De esta forma, si se reduce la anchura de la ventana del navegador, la página se verá más estrecha y si se maximiza la ventana del navegador, la página se verá más ancha.

Las siguientes imágenes muestran cómo se adapta el diseño dinámico a la anchura de la ventana del navegador, mostrando cada vez más contenidos a medida que se agranda la ventana.

Ejemplo 1

```
<html>
<head>
<style type="text/css">
body
{
background-image: url("bosque.jpg");
background-repeat:no-repeat;
}
#contenedor {
width: 300px;
margin: 0 auto;
}
```

```
.parrafo{
font-size: 16pt;
color: red;
}
.position1{
Font-weight:bold;
Font-size:70px;
Font-style:oblique;
Font-variant:small-caps;
line-height:90px;
```

```
Font-family:"comic sans ms";
}
.rojo{
color: #df0101;
}
.verde{
color: #31b404;
}
.amarillo{color: #ffcc00;
}
</style>
</head>
<body>
<div id="contenedor">
<h1 align="center"> encabezado</h1>
<div class="position1 ">
<p>La definición del estilo (negrita, 70 pixeles y tipo de letra comic
sans ms),<span class="verde">afecta a todo el</span> bloque de
etiquetas.</p>
<p>Este texto también es negrita, 70 pixeles y tipo de letra comic sans
ms. </p>
<p>al igual que esta línea. </p>
</div>
<div class="parrafo ">
<p> veamos si la definición del estilo (rojo y 16 puntos), afecta a todo
el bloque de etiquetas, hasta la etiqueta de cierre de div.</p>
<p style="color: blue">este parrafo es azul, a pesar de estar englobado
dentro de div, porque tiene su propio estilo.</p>
<p>este otro cumple con el estilo de div, porque no tiene estilo propio.
</p>
</div>
</div>
</body>
</html>
```

Ejemplo 2

```
<html>
<head>
<style type="text/css">
```

```
body
{
background-image: url("bosque.jpg");
background-repeat:no-repeat;
}
#contenedor {
width: 30%;
margin: 0 auto;
}
```

```
.parrafo{
font-size: 16pt;
color: red;
}
.position1{
Font-weight:bold;
Font-size:70px;
Font-style:oblique;
Font-variant:small-caps;
line-height:90px;
Font-family:"comic sans ms";
}
```

```
.rojo{
color: #df0101;
}
```

```
.verde{
color: #31b404;
}
```

```
.amarillo{color: #ffcc00;
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<div id="contenedor">
```

```
<h1 align="center"> encabezado</h1>
```

```
<div class="position1 ">
```

```
<p>La definición del estilo (negrita, 70 pixeles y tipo de letra comic
sans ms),<span class="verde">afecta a todo el</span> bloque de
etiquetas.</p>
```

```
<p>Este texto también es negrita, 70 pixeles y tipo de letra comic sans ms. </p>
<p>al igual que esta línea. </p>
</div>
<div class="parrafo ">
<p> veamos si la definición del estilo (rojo y 16 puntos), afecta a todo el bloque de etiquetas, hasta la etiqueta de cierre de div.</p>
<p style="color: blue">este parrafo es azul, a pesar de estar englobado dentro de div, porque tiene su propio estilo.</p>
<p>este otro cumple con el estilo de div, porque no tiene estilo propio.
</p>
</div>
</div>
</body>
</html>
```

Centrar una página verticalmente

Cuando se centra una página web de forma horizontal, sus márgenes laterales se adaptan dinámicamente de forma que la página siempre aparece en el centro de la ventana del navegador, independientemente de la anchura de la ventana. De la misma forma, cuando se centra una página web verticalmente, el objetivo es que sus contenidos aparezcan en el centro de la ventana del navegador y por tanto, que sus márgenes verticales se adapten de forma dinámica en función del tamaño de la ventana del navegador.

Aunque centrar una página web horizontalmente es muy sencillo, centrarla verticalmente es mucho más complicado. Afortunadamente, no es muy común que una página web aparezca centrada de forma vertical. El motivo es que la mayoría de páginas web son más altas que la ventana del navegador, por lo que no es posible centrarlas verticalmente.

A continuación se muestra la forma de centrar una página web respecto de la ventana del navegador, es decir, centrarla tanto horizontalmente como verticalmente.

Siguiendo el mismo razonamiento que el planteado para centrar la página horizontalmente, se podrían utilizar las siguientes reglas CSS para centrar la página respecto de la ventana del navegador:

```
#contenedor {
width: 300px;
```

```
height: 250px;
margin: auto;
}

<body>
  <div id="contenedor">
    <h1>Lorem ipsum dolor sit amet</h1>
    ...
  </div>
</body>
```

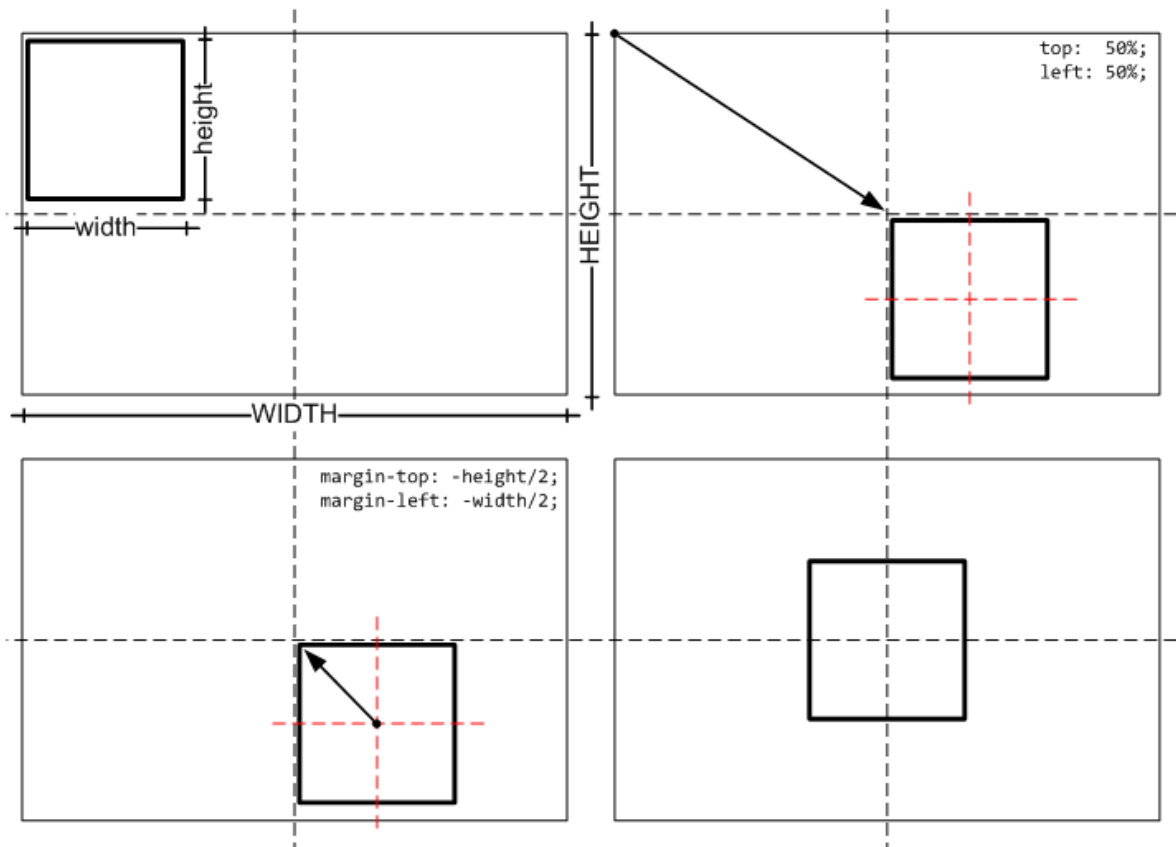
Ejemplo 1

```
<html>
<head>
  <style type="text/css">
    #contenedor {
      background-image: url("bosque.jpg");
      background-repeat: no-repeat;
      font-size: 16pt;
      color: red;
      width: 500px;
      height: 300px;
      margin: auto;
    }
  </style>
</head>
<body>
  <div id="contenedor">
    <h1 align="center"> encabezado</h1>
    <p>La definición del estilo (negrita, 70 pixeles y tipo de letra comic
    sans ms), <span class="verde">afecta a todo el</span> bloque de
    etiquetas.</p>
    <p>Este texto también es negrita, 70 pixeles y tipo de letra comic sans
    ms. </p>
    <p>al igual que esta línea. </p>
    <p> veamos si la definición del estilo (rojo y 16 puntos), afecta a todo
    el bloque de etiquetas, hasta la etiqueta de cierre de div.</p>
    <p style="color: blue">este parrafo es azul, a pesar de estar englobado
    dentro de div, porque tiene su propio estilo.</p>
    <p>este otro cumple con el estilo de div, porque no tiene estilo propio.
    </p>
  </div>
```

```
</body>  
</html>
```

Si el valor **auto** se puede utilizar para que los márgenes laterales se adapten dinámicamente, también debería ser posible utilizar el valor **auto** para los márgenes verticales. Desafortunadamente, la propiedad **margin: auto** no funciona tal y como se espera para los márgenes verticales y la página no se muestra centrada.

La solución correcta para centrar verticalmente una página web se basa en el posicionamiento absoluto e implica realizar un cálculo matemático sencillo. A continuación se muestra el esquema gráfico de los cuatro pasos necesarios para centrar una página web en la ventana del navegador:



En primer lugar, se asigna una altura y una anchura al elemento que encierra todos los contenidos de la página. En la primera figura, los contenidos de la página tienen una anchura llamada **width** y una altura llamada **height** que son menores que la anchura y altura de la ventana del navegador. En el siguiente ejemplo, se supone que tanto la anchura como la altura de la página es igual a **500px**:


```
#contenedor {  
  width: 500px;  
  height: 500px;  
}  
  
<body>  
  <div id="contenedor">  
    <h1>Lorem ipsum dolor sit amet</h1>  
    ...  
  </div>  
</body>
```

Ejemplo 2

```
<html>  
<head>  
<style type="text/css">  
#contenedor {  
  background-image: url("bosque.jpg");  
  background-repeat: no-repeat;  
  font-size: 16pt;  
  color: red;  
    width: 500px;  
    height: 500px;  
}  
  </style>  
</head>  
<body>  
  <div id="contenedor">  
    <h1 align="center"> encabezado</h1>  
    <p>La definición del estilo (negrita, 70 pixeles y tipo de letra comic  
    sans ms), <span class="verde">afecta a todo el</span> bloque de  
    etiquetas.</p>  
    <p>Este texto también es negrita, 70 pixeles y tipo de letra comic sans  
    ms. </p>  
    <p>al igual que esta línea. </p>  
    <p> veamos si la definición del estilo (rojo y 16 puntos), afecta a todo  
    el bloque de etiquetas, hasta la etiqueta de cierre de div.</p>  
    <p style="color: blue">este parrafo es azul, a pesar de estar englobado  
    dentro de div, porque tiene su propio estilo.</p>  
    <p>este otro cumple con el estilo de div, porque no tiene estilo propio.  
    </p>
```

```
</div>
</body>
</html>
```

A continuación, se posiciona de forma absoluta el elemento **contenedor** y se asigna un valor de **50%** tanto a la propiedad **top** como a la propiedad **left**. El resultado es que la esquina superior izquierda del elemento **contenedor** se posiciona en el centro de la ventana del navegador:

```
#contenedor {
  width: 500px;
  height: 500px;

  position: absolute;
  top: 50%;
  left: 50%;
}
```

Ejemplo 3

```
<html>
<head>
<style type="text/css">
#contenedor {
background-image: url("bosque.jpg");
background-repeat:no-repeat;
font-size: 16pt;
color: red;
  width: 500px;
  height: 500px;
position: absolute;
top: 50%;
left: 50%;
}
</style>
</head>
<body>
<div id="contenedor">
<h1 align="center"> encabezado</h1>
<p>La definición del estilo (negrita, 70 pixeles y tipo de letra comic
sans ms),<span class="verde">afecta a todo el</span> bloque de
etiquetas.</p>
<p>Este texto también es negrita, 70 pixeles y tipo de letra comic sans
ms. </p>
<p>al igual que esta línea. </p>
```

```
<p> veamos si la definición del estilo (rojo y 16 puntos), afecta a todo
el bloque de etiquetas, hasta la etiqueta de cierre de div.</p>
<p style="color: blue">este parrafo es azul, a pesar de estar englobado
dentro de div, porque tiene su propio estilo.</p>
<p>este otro cumple con el estilo de div, porque no tiene estilo propio.
</p>
</div>
</body>
</html>
```

Si la página se debe mostrar en el centro de la ventana del navegador, es necesario desplazar hacia arriba y hacia la izquierda los contenidos de la página web. Para determinar el desplazamiento necesario, se realiza un cálculo matemático sencillo. Como se ve en la tercera figura del esquema anterior, el punto central de la página debe desplazarse hasta el centro de la ventana del navegador.

Como se desprende de la imagen anterior, la página web debe moverse hacia arriba una cantidad igual a la mitad de su altura y debe desplazarse hacia la izquierda una cantidad equivalente a la mitad de su anchura. Utilizando las propiedades `margin-top` y `margin-left` con valores negativos, la página se desplaza hasta el centro de la ventana del navegador.

```
#contenedor {
  width: 500px;
  height: 500px;

  position: absolute;
  top: 50%;
  left: 50%;

  margin-top: -250px; /* height/2 = 500px / 2 */
  margin-left: -250px; /* width/2 = 500px / 2 */
}
```

Ejemplo 4

```
<html>
<head>
<style type="text/css">
#contenedor {
background-image: url("bosque.jpg");
background-repeat:no-repeat;
font-size: 16pt;
color: red;
```

```
width: 500px;
height: 500px;
position: absolute;
top: 50%;
left: 50%;
margin-top: -250px; /* height/2 = 500px / 2 */
margin-left: -250px; /* width/2 = 500px / 2 */
}
</style>
</head>
<body>
<div id="contenedor">
<h1 align="center"> encabezado</h1>
<p>La definición del estilo (negrita, 70 pixeles y tipo de letra comic
sans ms),<span class="verde">afecta a todo el</span> bloque de
etiquetas.</p>
<p>Este texto también es negrita, 70 pixeles y tipo de letra comic sans
ms. </p>
<p>al igual que esta línea. </p>
<p> veamos si la definición del estilo (rojo y 16 puntos), afecta a todo
el bloque de etiquetas, hasta la etiqueta de cierre de div.</p>
<p style="color: blue">este parrafo es azul, a pesar de estar englobado
dentro de div, porque tiene su propio estilo.</p>
<p>este otro cumple con el estilo de div, porque no tiene estilo propio.
</p>
</div>
</body>
</html>
```

Con las reglas CSS anteriores, la página web siempre aparece centrada verticalmente y horizontalmente respecto de la ventana del navegador. El motivo es que la anchura/altura de la página son fijas (propiedades **width** y **height**), el desplazamiento necesario para centrarla también es fijo (propiedades **margin-top** y **margin-left**) y el desplazamiento hasta el centro de la ventana del navegador se calcula dinámicamente gracias al uso de porcentajes en las propiedades **top** y **left**.

Para centrar una página sólo verticalmente, se debe prescindir tanto del posicionamiento horizontal como del desplazamiento horizontal:

```
#contenedor {  
  width: 500px;  
  height: 500px;  
  
  position: absolute;  
  top: 50%;  
  
  margin-top: -250px; /* height/2 = 500px / 2 */  
}
```

Ejemplo 5

```
<html>  
<head>  
<style type="text/css">  
#contenedor {  
  background-image: url("bosque.jpg");  
  background-repeat: no-repeat;  
  font-size: 16pt;  
  color: red;  
  width: 500px;  
  height: 500px;  
  position: absolute;  
  top: 50%;  
  margin-top: -250px; /* height/2 = 500px / 2 */  
}  
</style>  
</head>  
<body>  
<div id="contenedor">  
<h1 align="center"> encabezado</h1>  
<p>La definición del estilo (negrita, 70 pixeles y tipo de letra comic  
sans ms),<span class="verde">afecta a todo el</span> bloque de  
etiquetas.</p>  
<p>Este texto también es negrita, 70 pixeles y tipo de letra comic sans  
ms. </p>  
<p>al igual que esta línea. </p>  
<p> veamos si la definición del estilo (rojo y 16 puntos), afecta a todo  
el bloque de etiquetas, hasta la etiqueta de cierre de div.</p>  
<p style="color: blue">este parrafo es azul, a pesar de estar englobado  
dentro de div, porque tiene su propio estilo.</p>  
<p>este otro cumple con el estilo de div, porque no tiene estilo propio.  
</p>
```

```
</div>  
</body>  
</html>
```

Estructura o Layout

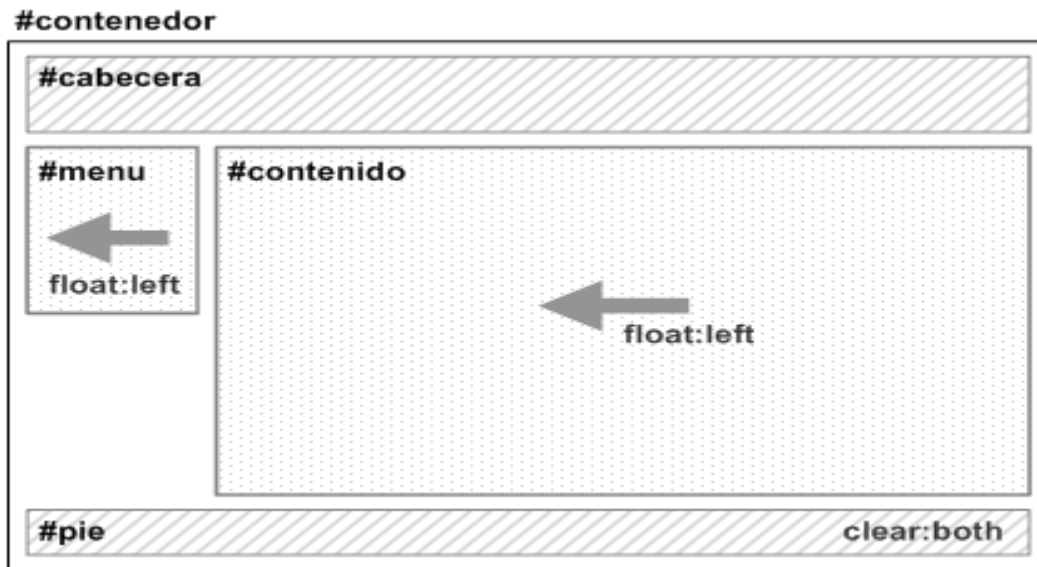
Diseño a 2 columnas con cabecera y pie de página

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 anchura del menú es de **150px** y la anchura de los contenidos es de **550px**:



Esquema del diseño a 2 columnas con cabecera y pie de página

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**.



Propiedades CSS necesarias en el diseño a dos columnas con cabecera y pie de página

El código HTML y CSS mínimos para definir la estructura de la página sin aplicar ningún estilo adicional son los siguientes:

```
#contenedor {  
  width: 700px;  
}  
#cabecera {  
}  
#menu {  
  float: left;  
  width: 150px;  
}  
#contenido {  
  float: left;  
  width: 550px;  
}  
#pie {  
  clear: both;  
}  
<body>  
<div id="contenedor">  
  <div id="cabecera">  
  </div>  
  
  <div id="menu">  
  </div>  
  
  <div id="contenido">  
  </div>  
  
  <div id="pie">  
  </div>  
</div>  
</body>
```

Ejemplo 1

```
<html>
<head>
<style type="text/css">

#contenedor
{
width: 700px;
}

#cabecera
{
color:red;
Font-weight:bold;
text-align:center;
width: 700px;
position:relative;
height: 100px;
background-image: url("cañada.jpg");
background-repeat:no-repeat;
}

#menu
{
float:left;
width: 150px;
height:300;
color:yellow;
background-image:url(bosque.jpg);
ul.a {list-style-type:upper-alpha;
}

}

#contenido
{
float: left;
width: 550px;
height:300;
background-image:url(floresexoticas.jpg);
```



```
}

#pie
{
color:green;
Font-weight:bold;
text-align:center;
background-image:url(Dock.jpg);

}

</style>
</head>

<body>
<div id="contenedor">
  <div id="cabecera">
    Esta es la cabecera de la página web
  </div>
  <div id="menu">
    <p align="center">menú</p>
    <ul class="a" >
      <li>Rodriguez Pablo</li>
      <li>Gonzalez Raul</li>
      <li>Lopez Hector</li>
    </ul>
  </div>
  <div id="contenido">
    <p style="color:pink; Font-weight:bold;" align="center">contenido</p>
    <p>Este texto también es negrita, 70 pixeles y tipo de letra comic sans ms. </p>
    <p>al igual que esta línea. </p>
    <p> veamos si la definición del estilo (rojo y 16 puntos), afecta a todo el bloque de etiquetas, hasta la etiqueta de cierre de div.</p>
    <p style="color: blue">este parrafo es azul, a pesar de estar englobado dentro de div, porque tiene su propio estilo.</p>

  </div>
  <div id="pie">
    <p> Este es el pie de página de la página web</p>
```

```
</div>  
</div>  
</body>  
</html>
```

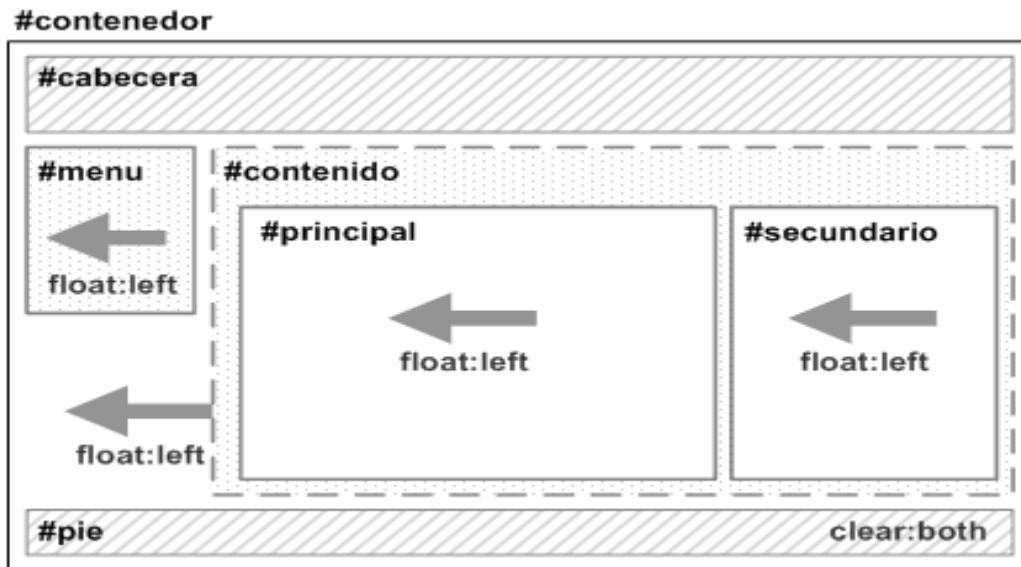
Diseño a 3 columnas con cabecera y pie de página

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:



Esquema del diseño a tres columnas con cabecera y pie de página

La solución CSS emplea la misma estrategia del diseño a dos columnas y se basa en utilizar las propiedades **float** y **clear**:



Propiedades CSS necesarias en el diseño a 3 columnas con cabecera y pie de página

El código HTML y CSS mínimos para definir la estructura de la página sin aplicar ningún estilo adicional son los siguientes:

```
#contenedor {  
}  
#cabecera {  
}  
#menu {  
  float: left;  
  width: 15%;  
}  
#contenido {  
  float: left;  
  width: 85%;  
}  
#contenido #principal {  
  float: left;  
  width: 80%;  
}  
#contenido #secundario {  
  float: left;  
  width: 20%;  
}  
  
#pie {  
  clear: both;  
}
```

```
<body>  
<div id="contenedor">  
  <div id="cabecera">  
  </div>  
  
  <div id="menu">  
  </div>  
  
  <div id="contenido">
```

```
<div id="principal">
</div>

<div id="secundario">
</div>
</div>

<div id="pie">
</div>
</div>
</body>
```

El código anterior crea una página con anchura variable que se adapta a la ventana del navegador. Para definir una página con anchura fija, solamente es necesario sustituir las anchuras en porcentajes por anchuras en píxel.

Al igual que sucedía en el diseño a dos columnas, se puede optar por posicionar todos los elementos mediante float: left o se puede utilizar float: left para el menú y la zona principal de contenidos y float: right para el contenedor de los contenidos y la zona secundaria de contenidos.

Ejemplo 1

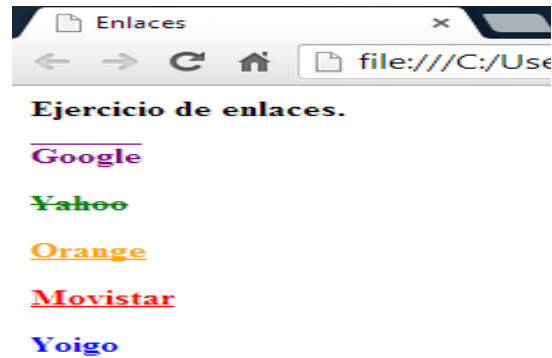
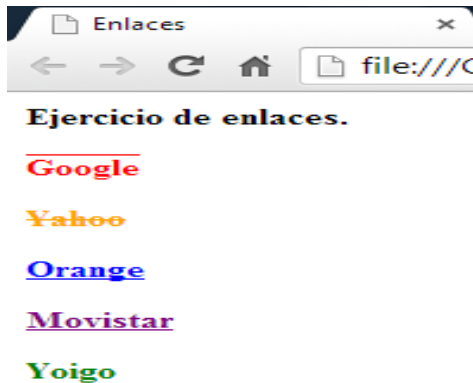
```
<html>
<head>
<style type="text/css">
#contenedor
{
width:770px;
}
#cabecera
{
width: 770px;
position: relative;
height: 100px;
color:white;
Font-weight:bold;
text-align:center;
background-image: url("cañada.jpg");
background-repeat:no-repeat;
}
#menu
{
```

```
float: left;
width: 100px;
height:300px;
color:yellow;
background-image:url(bosque.jpg);
ol.a {list-style-type:upper-alpha};
}
#contenido
{
background-image:url(floresexoticas.jpg);
float: left;
width: 670px;
height:300px;
}
#contenido #principal
{
margin: 15px 0 0 30px;
color:pink;
float: left;
width: 250px;
Font-weight:bold;
text-align:justify;
}
#contenido #secundario
{
margin: 15px 0 0 30px;
color:#BFFF00;
float: left;
width: 250px;
Font-weight:bold;
text-align:justify;
}
#pie
{
position: relative;
height: 60px;
color:black;
Font-weight:bold;
text-align:center;
background-image:url(Dock.jpg);
```

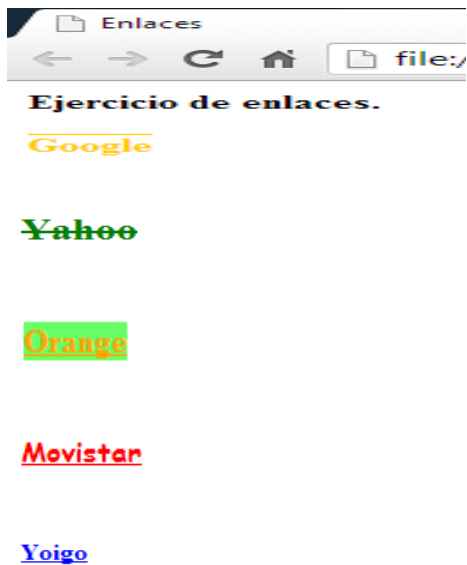
```
clear: both;
}
</style>
</head>
<body>
<div id="contenedor">
  <div id="cabecera">
    Esta es la cabecera de la página web
  </div>
  <div id="menu">
    <p align="center">menú</p>
    <ol class="a" >
      <li>Rodriguez Pablo</li>
      <li>Gonzalez Raul</li>
      <li>Lopez Hector</li>
    </ol>
  </div>
  <div id="contenido">
    <div id="principal">
      <p> este es el primer parrafo del contenido principal de la
página. Esperemos que todo salga segun lo esperado</p>
    </div>
    <div >
      <p> este es el primer parrafo del contenido secundario de la
pagina. Esperemos que todo salga segun lo esperado</p>
    </div>
  </div>
  <div id="pie">
    este es el pie
  </div>
</div>
</body>
</html>
```

Ejercicios

1º.- Definir las reglas CSS que permiten mostrar los enlaces con los siguientes estilos: en la imagen de la izquierda se ven los enlaces antes de ser visitados y en la derecha una vez visitados



Una vez que ponemos el cursor encima se deben ver de la siguiente forma:



2º.- Definir las reglas css que permiten mostrar la siguiente lista con los siguientes estilos:

Brasil

Uruguay
Argentina

- Brasil
- Uruguay
- Argentina

- Brasil
- Uruguay
- Argentina

- Brasil
- Uruguay
- Argentina

1. Brasil
2. Uruguay
3. Argentina

- i. Brasil
- ii. Uruguay
- iii. Argentina


- I. Brasil
- II. Uruguay
- III. Argentina

- a. Brasil
- b. Uruguay
- c. Argentina

- A. Brasil
- B. Uruguay
- C. Argentina

 Brasil

UT7. CSS AVANZADO

 Uruguay

 Argentina

3°.- Definir las reglas css que permiten mostrar la siguiente tabla con los siguientes estilos:

cantidad de lluvia caída en mm.

| Provincia | Enero | Febrero | Marzo |
|--------------|-------|---------|-------|
| Córdoba | 210 | 170 | 120 |
| Buenos Aires | 250 | 190 | 140 |
| Santa Fe | 175 | 140 | 120 |

4°.- Definir las reglas css que permiten mostrar el siguiente formulario con los siguientes estilos:

Ingrese nombre:

Ingrese mail:

Comentarios:

5°.-Definir las reglas css que permiten mostrar la siguiente pagina con cabecera, pie y dos columnas.

Página de periódico a dos columnas

¿Sabes cuál es la forma más barata de calentar tu hogar?

Un reciente estudio realizado por Pwc «El mix térmico en España a 2030» ha analizado cuál es el gasto medio anual de calentar un hogar. Como muestra ha tomado una vivienda de 90 metros cuadrados con un consumo medio de ocho megavatios/año. El resultado, según la energía utilizada, es el siguiente:

- Gas natural: 760-928 euros/año
- Gas propano: 848-1.040 euros/año
- Gasóleo: 1.112-1.224 euros/año
- Biomasa: 1.168-2.104 euros/año
- Electricidad: 1.960-2.168

Las 10 claves que te ayudarán a ahorrar 4000 euros en tu coche

Más de 4.000 euros al año y un aumento de la probabilidad de sufrir una avería grave. Eso es lo que puede costar el mal mantenimiento de un vehículo, según el último estudio realizado por LeasePlan Lab con el fin de concienciar a los conductores de la importancia de someter a examen sus vehículos cada cierto tiempo.

Realizar las revisiones recomendadas por el fabricante y las propias del automóvil son sólo un seguro de vida para quienes se ponen al volante de los vehículos, sino que para muchas empresas puede traducirse en un aumento de la movilidad de su flota con el consecuente incremento de su productividad.

Con un correcto mantenimiento se consigue menor consumo de combustible, mayor seguridad y movilidad, ya que se evitan las averías fortuitas y se minimizan los riesgos de sufrir una avería grave que inmovilice el vehículo en el taller durante diez días. Ese es el tiempo medio que la compañía estima que tiene que estar un vehículo reparándose tras la rotura, por ejemplo, de un motor.

«Saltarse las revisiones recomendadas por el fabricante aumenta claramente la probabilidad de sufrir una avería grave, que puede suponer hasta 10 días de estancia en taller, frente a un día al año que supone de media acudir al taller para realizar el mantenimiento preventivo. Esto sin duda, repercute directamente, no sólo en la productividad, sino en más costes producidos por la inmovilización, el tiempo extra de vehículo de sustitución, etc.», asegura David Henche, director de Marketing y Comunicación de LeasePlan.

Recomendaciones para los conductores

1.Revisar los niveles del vehículo periódicamente. La falta de agua o aceite puede provocar problemas en el sistema de refrigeración, ya que sin lubricante se produce una mayor fricción entre las piezas, que se calientan y pueden llegar a romperse. Este tipo de avería tiene un coste mucho mayor que reemplazar sólo el aceite

euros/año
Es decir, el uso del gas natural supone el mayor ahorro y la electricidad la opción más cara. Este hecho puede hacer que en muchos hogares se planteen hacer un cambio del uso de la energía empleada. Pero, ¿cuánto costaría? Los expertos consultados en la Asociación Española de Gas Sedigas aseguran que no existe un dato concreto y preciso, porque dependerá del tipo de instalación que se quiera modificar y a qué tipo de energía. Existen muchas calderas que son adaptables — para uso de gas natural— con la sustitución de algunas piezas, y algunos radiadores podrían, igualmente, valer. Es decir, en estos casos, el coste del "reciclaje" sería menor. Por otro lado, existen instalaciones muy antiguas que sería casi imposible reciclarlas.

2.El sistema de suspensión del vehículo experimenta un envejecimiento paulatino que muchas veces el conductor no es capaz de controlar, por eso conviene realizar un test de amortiguación cada 50.000 Km en el taller donde se realiza la revisión. Unos amortiguadores en mal estado aumentan de forma drástica las probabilidades de sufrir una colisión.

3.Comprobar el estado de las correas de distribución es otra de las operaciones básicas de mantenimiento. Entre las recomendaciones del fabricante figura la revisión o sustitución de la misma. En caso de rotura, el motor del automóvil puede quedar inutilizado.

4.Revisar el sistema de frenado. Esto puede llevarse a cabo tanto durante la conducción, estando pendientes del tacto del pedal, como a través de la información que muestra la centralita del vehículo o la información sonora en caso de no disponer de aviso de testigo.

5.Alejar la batería de temperaturas extremas, pues es lo que más daño causa a esta pieza, fundamental en el sistema eléctrico de los automóviles.

6.Cuidar los neumáticos, ya que son el único punto de unión entre el coche y el asfalto, por lo que se encargan de transmitir la potencia del motor sobre el suelo para mover el vehículo. Dada su importancia conviene estar especialmente pendientes de que la presión de inflado es correcta y la profundidad del dibujo, suficiente. Hay que tener en cuenta que circular con menos de 2 mm de dibujo no es aconsejable y con menos de 1,6 mm es además temerario.

7.Ojo al repostar diesel o gasolina, equivocarse de combustible es más común de lo que muchos piensan. Equivocarse a la hora de repostar es la negligencia más común, según un estudio elaborado por LeasePlan Lab (41% de las negligencias se deben a este motivo).

8.Además del filtro de aceite, otros filtros que se suelen cambiar varias veces a lo largo de la vida útil de un automóvil que son los de polen, aire y combustible. En todos los casos, el momento adecuado de la sustitución viene indicado en la hoja del mantenimiento.

9.En caso de percibir cualquier ruido o malfuncionamiento en el vehículo, se debe acudir al taller para evitar que daños que inicialmente son fáciles de arreglar, se conviertan en problemas importantes.

10.Por último, es importante respetar el plan de mantenimiento establecido por el fabricante, que se encuentra en el libro de usuario que se entrega con el vehículo. Pasarse 500 kilómetros de los indicados puede derivar en una avería grave.

fin de la página

Contenido de la cabecera:

Página de periódico a dos columnas

Contenido de la primera columna:

¿Sabes cuál es la forma más barata de calentar tu hogar?

Un reciente estudio realizado por Pwc «El mix térmico en España a 2030» ha analizado cuál es el gasto medio anual de calentar un hogar. Como muestra ha tomado una vivienda de 90 metros cuadrados con un consumo medio de ocho megavatios/año. El resultado, según la energía utilizada, es el siguiente:

- Gas natural : 760-928 euros/año
- Gas propano: 848-1.040 euros/año
- Gasóleo: 1.112-1.224 euros/año
- Biomasa: 1.168-2.104 euros/año
- Electricidad: 1.960-2.168 euros/año

Es decir, el uso del gas natural supone el mayor ahorro y la electricidad la opción más cara. Este hecho puede hacer que en muchos hogares se planteen hacer un cambio del uso de la energía empleada. Pero, ¿cuánto costaría? Los expertos consultados en la Asociación Española de Gas Sedigas aseguran que no existe un dato concreto y preciso, porque dependerá del tipo de instalación que se quiera modificar y a qué tipo de energía. Existen muchas calderas que son adaptables —para uso de gas natural— con la sustitución de algunas piezas, y algunos radiadores podrían, igualmente, valer. Es decir, en estos casos, el coste del "reciclaje" sería menor. Por otro lado, existen instalaciones muy antiguas que sería casi imposible reciclarlas.

Contenido de la segunda columna:

Las 10 claves que te ayudarán a ahorrar 4000 euros en tu coche

Más de 4.000 euros al año y un aumento de la probabilidad de sufrir una avería grave. Eso es lo que puede costar el mal mantenimiento de un vehículo, según el último estudio realizado por LeasePlan Lab con el fin de concienciar a los conductores de la importancia de someter a examen sus vehículos cada cierto tiempo.

Realizar las revisiones recomendadas por el fabricante y las propias del automóvil no son sólo un seguro de vida para quienes se ponen al volante de los vehículos, sino que para muchas empresas puede traducirse en un aumento de la movilidad de su flota con el consecuente incremento de su productividad.

Con un correcto mantenimiento se consigue menor consumo de combustible, mayor seguridad y movilidad, ya que se evitan las averías fortuitas y se minimizan los riesgos de sufrir una avería grave que inmovilice el vehículo en el taller durante diez días. Ese es el tiempo medio que la compañía estima que tiene que estar un vehículo reparándose tras la rotura, por ejemplo, de un motor.

«Saltarse las revisiones recomendadas por el fabricante aumenta claramente la probabilidad de sufrir una avería grave, que puede suponer hasta 10 días de estancia en taller, frente un día al año que supone de media acudir al taller para realizar el mantenimiento preventivo. Esto sin duda, repercute directamente, no sólo en la productividad, sino en más costes producidos por la inmovilización, el tiempo extra de vehículo de sustitución, etc.», asegura David Henche, director de Marketing y Comunicación de LeasePlan.

Recomendaciones para los conductores:

1. Revisar los niveles del vehículo periódicamente. La falta de agua o aceite puede provocar problemas en el sistema de refrigeración, ya que sin lubricante se produce una mayor fricción entre las piezas, que se calientan y pueden llegar a romperse. Este tipo de avería tiene un coste mucho mayor que reemplazar sólo el aceite.
2. El sistema de suspensión del vehículo experimenta un envejecimiento paulatino que muchas veces el conductor no es capaz de controlar, por eso conviene realizar un test de amortiguación cada 50.000 Km en el taller donde se realiza la revisión. Unos amortiguadores en mal estado aumentan de forma drástica las probabilidades de sufrir una colisión.
3. Comprobar el estado de las correas de distribución es otra de las operaciones básicas de mantenimiento. Entre las recomendaciones del fabricante figura la revisión o sustitución de la misma. En caso de rotura, el motor del automóvil puede quedar inutilizado.
4. Revisar el sistema de frenado. Esto puede llevarse a cabo tanto durante la conducción, estando pendientes del tacto del pedal, como a través de la información que muestra la centralita del vehículo o la información sonora en caso de no disponer de aviso de testigo.
5. Alejar la batería de temperaturas extremas, pues es lo que más daño causa a esta pieza, fundamental en el sistema eléctrico de los automóviles.
6. Cuidar los neumáticos, ya que son el único punto de unión entre el coche y el asfalto, por lo que se encargan de transmitir la potencia del motor sobre el suelo para mover el vehículo. Dada su importancia conviene estar especialmente pendientes de que la presión de inflado es correcta y la profundidad del dibujo, suficiente. Hay que tener en cuenta que circular con

menos de 2 mm de dibujo no es aconsejable y con menos de 1,6 mm es además temerario.

7. Ojo al repostar diesel o gasolina, equivocarse de combustible es más común de lo que muchos piensan. Equivocarse a la hora de repostar es la negligencia más común, según un estudio elaborado por LeasePlan Lab (41% de las negligencias se deben a este motivo).

8. Además del filtro de aceite, otros filtros que se suelen cambiar varias veces a lo largo de la vida útil de un automóvil que son los de polen, aire y combustible. En todos los casos, el momento adecuado de la sustitución viene indicado en la hoja del mantenimiento.

9. En caso de percibir cualquier ruido o malfuncionamiento en el vehículo, se debe acudir al taller para evitar que daños que inicialmente son fáciles de arreglar, se conviertan en problemas importantes.

10. Por último, es importante respetar el plan de mantenimiento establecido por el fabricante, que se encuentra en el libro de usuario que se entrega con el vehículo. Pasarse 500 kilómetros de los indicados puede derivar en una avería grave.

Contenido del pie de página:

Fin de la página