			Clase 9

Tópico

HTML+ CSS

Metas de comprensión

- ✓ Los alumnos comprenderán y aprenderán a implementar tablas para tabular información
- ✓ Los alumnos comprenderán de qué manera se defines los estilos y cuál es la ventaja de su uso sobre tablas
- ✓ Los alumnos implementaran CSS utilizando contenedores Divs
- ✓ Los alumnos maquetaran paginas utilizando los distintos tipos de posicionamiento de contenedores

Desempeño de Exploración

Debate sobre los conceptos expuestos en clase. Ejemplificación y relación con otras materias. Mediante la codificación de los ejercicios propuestos se harán nuevas propuestas de diseño y maquetación, lo que llevara a los alumnos a investigar nuevas definiciones o tags ya sea en forma grupal o individual.

Introducción

¡Cada clase según su contenido puede tener un numero variable de páginas, a leer, no te asustes!! La materia esta lo más actualizado posible al año 2022, de manera tal que tengas todos los temas en forma completa, de allí su extensión.

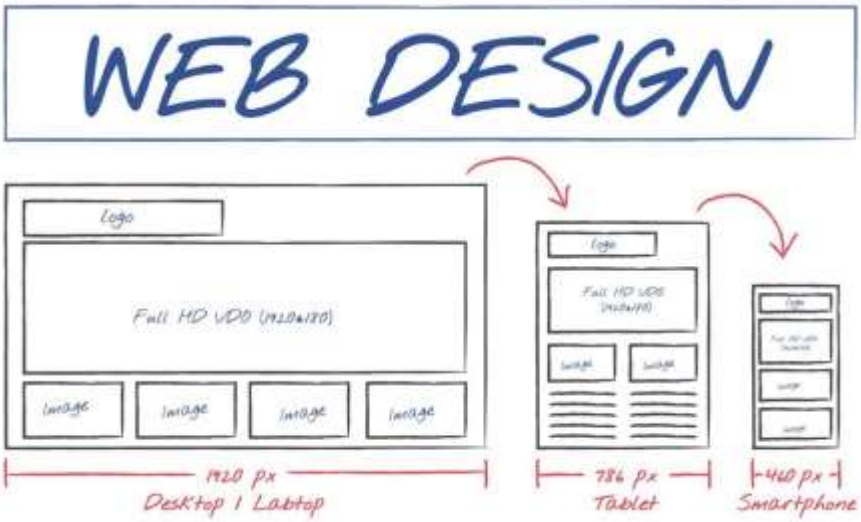
Algunas recomendaciones que te pueden ayudar a la hora de comprender el material de estudio:

- ✓ Lee varias veces la clase si fuera necesario.
- ✓ Subraya, destaca o resume los conceptos que creas principales o de importancia en cada tema.
- ✓ Puedes grabar la lectura de la clase (no necesaria puedes ser tú, puede ser un familiar, amigo etc.) para poder escucharla luego en el colectivo, recreo, relax o fin de semana etc.
- ✓ Muchas veces los números o medidas (velocidades, tamaños, resoluciones, latencia, potencia, capacidad, etc.) no son tan importantes ya que la informática evoluciona día a día y esos son bastantes cambiantes en la guerra de las empresas y fabricantes para sobresalir en el mercado, por eso no es necesario que los memorices a todos a menos que el tutor te lo indique.
- ✓ Puedes realizar grafica con cuadros,

- ✓ Puedes ampliar tu conocimiento con investigaciones adiciones en la web o viendo videos en youtube que refuercen los conceptos
- ✓ Puedes consultarle a tu tutor por cualquiera de las vías indicadas en el campus por el temario si tienes alguna duda, consulta o inquietud.
- ✓ Trata de organizar tu tiempo para la lectura y la comprensión del material
- ✓ Este texto va a acompañar a todas las clases para recordarte como leer, estudiar y comprender el material de la materia
- ✓ Cada clase en el campus podrá estar acompañada de material adicional como profundizar los temas para la mejor interpretación de los mismos con videos, encuestas, foros, actividades individuales y/o grupales.
- ✓ Busca algún compañero de estudio para poder compartir conocimiento, apuntes y metodología de estudio
- ✓ No esperes hasta último momento para realizar tus consultas, leer o ponerte al tanto con la materia

La maquetación o diagramación web se refiere a transformar un diseño gráfico en una interfaz funcional en términos de programación que entienda un navegador o dispositivo específico.

Las primeras maquetaciones se realizaban con tablas, práctica que no recomendable en la actualidad ya que existen los contenedores y las zonas semánticas de HTML5.uto. Las tablas son para presentar datos de una manera ordenada, no para maquetar una página web. Es por eso que en esta clase veremos estos dos tópicos para que puedas ver las diferencias.

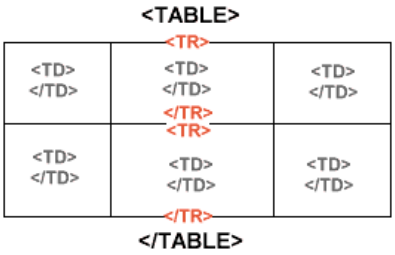


Te invito a la lectura de la clase mano a la obra!

Tablas Básicas

Una tabla en un conjunto de celdas organizadas dentro de las cuales podemos alojar distintos contenidos.

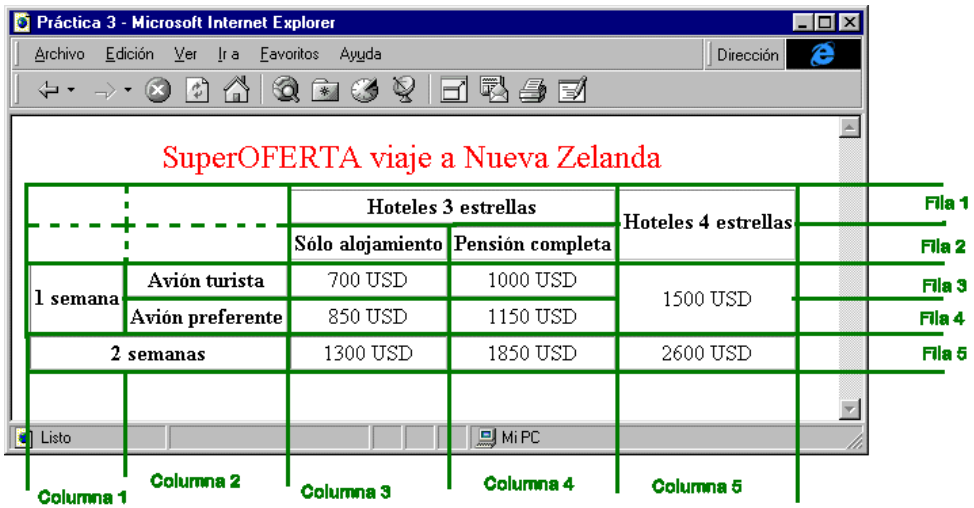
Pueden ser utilizadas principalmente para listar datos como agendas, resultados y otros datos de una forma organizada. Nada más lejos de la realidad. Una tabla nos permite organizar y distribuir los espacios de la manera más óptima. Nos puede ayudar a generar texto en columnas como los periódicos, prefijar los tamaños ocupados por distintas secciones de la página o poner de una manera sencilla un pie de foto a una imagen.



Para empezar, nada más sencillo que por el principio: las tablas son definidas por las etiquetas <***table***> y </***table***>. Dentro de estas dos etiquetas colocaremos todas las otras etiquetas, textos e imágenes que darán forma y contenido a la tabla.

Las tablas son descritas por líneas de izquierda a derecha. Cada una de estas líneas es definida por otra etiqueta y su cierre: <***tr***> y </***tr***>

Asimismo, dentro de cada línea, habrá diferentes celdas. Cada una de estas celdas será definida por otro par de etiquetas: <***td***> y </***td***>. Dentro de estas etiquetas será donde coloquemos nuestro contenido.



A continuación, se muestra el código HTML de una tabla sencilla:

```
<html>
<head><title>Ejemplo de tabla sencilla</title></head>
<body>
<h1>Listado de cursos</h1>
<table>
<tr>
  <td><strong>Curso</strong></td>
```

```
  <td><strong>Horas</strong></td>
  <td><strong>Horario</strong></td>
</tr>
<tr>
  <td>CSS</td>
  <td>20</td>
  <td>16:00 - 20:00</td>
</tr>
<tr>
  <td>HTML</td>
  <td>20</td>
  <td>16:00 - 20:00</td>
</tr>
<tr>
  <td>Dreamweaver</td>
  <td>60</td>
  <td>16:00 - 20:00</td>
</tr>
</table>
</body>
</html>
```

Listado de cursos

Curso	Horas	Horario
CSS	20	16:00 - 20:00
HTML	20	16:00 - 20:00
Dreamweaver	60	16:00 - 20:00

A continuación, veremos algunos atributos para las tablas.

align	Justifica el texto de la celda del mismo modo que si fuese el de un párrafo.
valign	Podemos elegir si queremos que el texto aparezca arriba (***top***), en el centro (***middle***) o abajo (***bottom***) de la celda.
bgcolor	Da color a la celda o línea elegida.
bordercolor	Define el color del borde.

Otros atributos que pueden ser únicamente asignados a una celda y no al conjunto de celdas de una línea son:

background	Nos permite colocar un fondo para la celda a partir de un enlace a una imagen.
height	Define la altura de la celda en *pixels* o porcentaje.
width	Define la anchura de la celda en *pixels* o porcentaje.
colspan	Expande una celda horizontalmente.
rowspan	Expande una celda verticalmente.

Height y ***width*** nos ayudan a definir las dimensiones de nuestras celdas de una forma absoluta (en pixels o puntos de pantalla) o de una forma relativa, es decir por porcentajes referidos al tamaño total de la tabla.

Los atributos ***rowspan*** y ***colspan*** son también utilizados frecuentemente. Gracias a ellos es posible expandir celdas fusionando éstas con sus vecinas. El valor que pueden tomar estas etiquetas es numérico. El número representa la cantidad de celdas fusionadas. Por lo tanto, diremos que:

- ✓ ***Rowspan*****:** indica el número de filas que ocupará la celda. Por defecto ocupa una sola fila.

- ✓ ***Colspan*:** indica el número de columnas que ocupará la celda. Por defecto ocupa una sola columna.

De esta forma si ponemos <td ***colsan***=2>, quiere decir que la celda actual se extiende en el ancho de dos celdas.

Algo parecido ocurre si ponemos <td ***rowspan***=3>, la celda ocupará el alto de 3 celdas normales. Veamos un ejemplo:

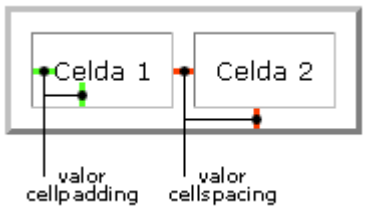
```
<!DOCTYPE HTML>
<BODY>

<TABLE CELLSPACING="2" CELLPADDING="2"
BORDER="1">
<CAPTION><FONT SIZE="+2" COLOR="#FF0000">OFERTA
viaje a Nueva Zelanda</FONT></CAPTION>

<TR>
        <Td ROWSPAN="2" COLSPAN="2"></TD>
        <TD COLSPAN="2">Hoteles 3 estrellas</TD>
        <TD ROWSPAN="2">Hoteles 4 estrellas</TD>
      </TR>
      <TR>
      <td>Sólo alojamiento</TD>
      <TD>Pensión completa</td>
</TR>

<TR>
      <TD ROWSPAN="2">1 semana</td>
      <TD>Avión turista</td>
      <TD ALIGN="center">700 USD</TD>
      <TD ALIGN="center">1000 USD</TD>
      <TD ROWSPAN="2" ALIGN="center">1500 USD</TD>
</TR>

<TR>
      <TD>Avión preferente</TD>
      <TD ALIGN="center">850 USD</TD>
      <TD ALIGN="center">1150 USD</TD>
</TR>

<TR>
      <TD COLSPAN="2">2 semanas</TD>
      <TD ALIGN="center">1300 USD</TD>
      <TD ALIGN="center">1850 USD</TD>
      <TD ALIGN="center">2600 USD</TD>
</TR>
      </TABLE>
      </BODY>
      </HTML>
```

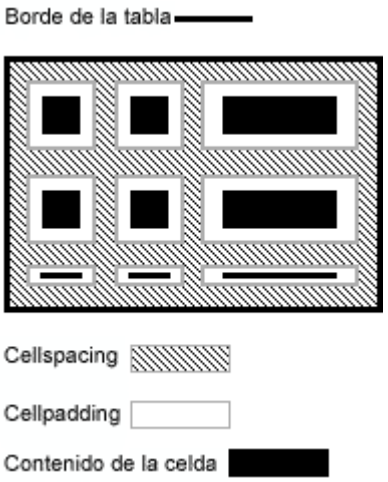
OFERTA viaje a Nueva Zelanda

		Hoteles 3 estrellas		Hoteles 4 estrellas
		Sólo alojamiento	Pensión completa	
1 semana	Avión turista	700 USD	1000 USD	1500 USD
	Avión preferente	850 USD	1150 USD	
2 semanas		1300 USD	1850 USD	2600 USD

Si queremos que la tabla tenga título, lo haremos por medio de la etiqueta ***CAPTION***. Para establecer los márgenes de las celdas utilizaremos:

- ✓ ***Cellspacing*:** Este atributo especifica cuánto espacio debería dejar el agente de usuario entre el lado izquierdo de la tabla y el lado izquierdo de la columna que está más a la izquierda, entre la parte superior de la tabla y el lado superior de la fila que está más arriba, y lo mismo para los lados derecho e inferior. El atributo también especifica la cantidad de espacio entre celdas.
- ✓ ***cellpadding* :** Este atributo especifica la cantidad de espacio entre el borde de la celda y sus contenidos. Si el valor de este atributo es una longitud en píxeles, los cuatro bordes deberían estar a esta distancia de los contenidos. Si el valor del atributo es una longitud porcentual, los bordes superior e inferior deberían estar igualmente separados del contenido según un porcentaje del espacio vertical disponible, y los bordes izquierdo y derecho deberían estar igualmente separados de los contenidos según un porcentaje del espacio horizontal disponible.





El atributo *cellspacing* especifica que las celdas deberían estar separadas entre sí y hasta el marco de la tabla por veinte píxeles. El atributo *cellpadding* específica que el margen superior de la celda y el margen inferior de la celda estarán separados de los contenidos de la celda por el 10% del espacio vertical disponible (para un total del 20%). Análogamente, el borde izquierdo de la celda y el borde derecho de la celda estarán separados de los contenidos de la celda por el 10% del espacio horizontal disponible (para un total del 20%).

```
<TABLE cellspacing="20" cellpadding="20%">
<TR> <TD>Dato1 <TD>Dato2 <TD>Dato3
</TABLE>
```

Tabla con 10 pixels de margen (CELLPADDING=10)

Fila1,Columna1	Fila1,Columna2
Fila2,Columna1	Fila2,Columna2

Tabla con margenes nulos

Fila1,Columna1	Fila1,Columna2
Fila2,Columna1	Fila2,Columna2

Si definimos una celda de un ancho 100 por ejemplo, y colocamos en la celda un contenido como una imagen que mida más de 100 píxeles, la celda crecerá en horizontal todo lo necesario para que la imagen quepa. Si el elemento, aunque más ancho, fuera divisible (como un texto) el ancho sería respetado y el texto crecería hacia abajo o lo que es lo mismo, en altura, como señalábamos en el anterior párrafo. No solo la página puede tener un fondo, también lo podemos

colocar a las tablas o las celdas, por ejemplo. Se utiliza el mismo atributo background, aunque aplicado a otras etiquetas. Por ejemplo:

<TABLE CELLPADDING=8 CELLSPACING=0 **BACKGROUND="foto.gif"**>
<TD **BACKGROUND**="images/ex.gif" WIDTH="60" HEIGHT="35">

Bordes y colores

Para mejorar el aspecto de nuestras tablas, lo primero que querremos hacer es jugar con la anchura del borde. Para ello se ha definido el parámetro border de la etiqueta <table>, que define la anchura de todos los bordes de la tabla en pixels. Si se especifica con valor 0, el borde no se visualiza. El color del borde lo controla el parámetro ***bordercolor***, que se aplica al tag <td>. Este parámetro toma valores hexadecimales o nombres de colores predefinidos, según el alfabeto inglés. La forma en que se indica un color con código hexadecimal es la misma que se utiliza para otros elementos de una página (por ejemplo, para el color de fondo).

Finalmente, también podemos modificar el color de fondo de cada celda de forma independiente, aplicando el parámetro *bgcolor* al tag <td>. Los colores se especifican de igual forma.

Con todo esto, podemos mejorar el aspecto del ejemplo anterior de la siguiente forma:

```
<table width="80%" cellspacing="0" cellpadding="5" border="4">
 <tr>
  <td width="80%" height="70" align="right" valign="bottom">Fila 1, celda 1</td>
  <td width="20%" height="70">Fila 1, celda 2</td>
 </tr>
 <tr>
  <td width="80%" bgcolor="#FFCCCC">Fila 2, celda 1</td>
  <td width="20%" bordercolor="red">Fila 2, celda 2</td>  </tr> </table>
```

Otros atributos

- ✓ summary (resumen del contenido de la tabla para sistemas acústicos)
- ✓ Frame (marco) a qué lados de la tabla deben ser añadidos bordes
- ✓ Rules (reglas) permite especificar en la etiqueta de apertura <table> las líneas horizontales o reglas para las líneas de la cuadrícula

frame

- ✓ frame="void" (void = nada)
- ✓ frame="above" (above = superior)
- ✓ frame="below" (below = inferior)
- ✓ frame="hsides" (horizontal sides = lados horizontales)
- ✓ frame="vsides" (vertical sides = lados verticales)
- ✓ frame="lhs" (left hand side = lado izquierdo)
- ✓ frame="rhs" (right hand side = lado derecho)

rules

- ✓ rules="none" (none = ningún) no son visualizadas las líneas de la cuadrícula, el borde exterior de la tabla es sin embargo visualizado.
- ✓ rules="rows" (rows = filas) son visualizadas las líneas entre todos las filas de la tabla, pero no las líneas entre las columnas de la tabla.

- ✓ rules="cols" (cols = columnas) son visualizadas todas las líneas entre todas las columnas de la tabla, sin embargo no las líneas entre las filas.
- ✓ rules="groups" (groups = grupos) son visualizadas todas las líneas entre la cabeza, cuerpo y pie de una tabla
- ✓ rules="all" (all = todo) son visualizadas todas las líneas entre todas las celdas de la tabla. Es el valor preajustado por lo que equivale a no utilizar este atributo.

Veremos a continuación un ejemplo aplicando estilos a tablas

```
<!doctype html>
<meta charset="utf-8">
<style>
caption{font-family:arial;
font-size:15px;text-align: center;
margin: 0px;font-weight: bold;
padding:10px;}

table{border-collapse: collapse;}

th{border-right: 1px solid #fff;
border-bottom: 1px solid #fff;
padding: 0.5em; background-color:#6495ed;}

thead th { background-color: #6495ed; color: #fff; }

tbody th{font-family:arial; font-weight: normal; background-color: #6495ed; color:#ff0;}
td {border: 1px solid #000; padding: .5em; background-color:#ed8f63; width:100px;
text-align:center; }

</style>
<body>

<table >
<caption>Contenido nutricional por cada 100 g de alimento. </caption>
<tr>
    <th>Alimento</th> <th>Calorías (kCal)</th> <th>Grasas (g)</th>
     <th>Proteína (g)</th> <th>Carbohidratos (g)</th>
</tr>
<tr>
      <td>Arándano</td> <td>49</td> <td>0.2</td>  <td>0.4</td> <td>12.7</td>
</tr>
<tr>
          <td>Plátano</td> <td>90</td> <td>0.3</td>    <td>1.0</td> <td>23.5</td>
</tr>

<tr>
   <td>Cereza</td> <td>46</td> <td>0.4</td> <td>0.9</td> <td>10.9</td>
</tr>

<tr>
      <td>Fresa</td> <td>37</td> <td>0.5</td> <td>0.8</td> <td>8.3</td>
</tr>
</table>
</body>
</html>
```

Contenido nutricional por cada 100 g de alimento.

Alimento	Calorías (kCal)	Grasas (g)	Proteína (g)	Carbohidratos (g)
Arándano	49	0.2	0.4	12.7
Plátano	90	0.3	1.0	23.5
Cereza	46	0.4	0.9	10.9
Fresa	37	0.5	0.8	8.3

Tablas anidadas

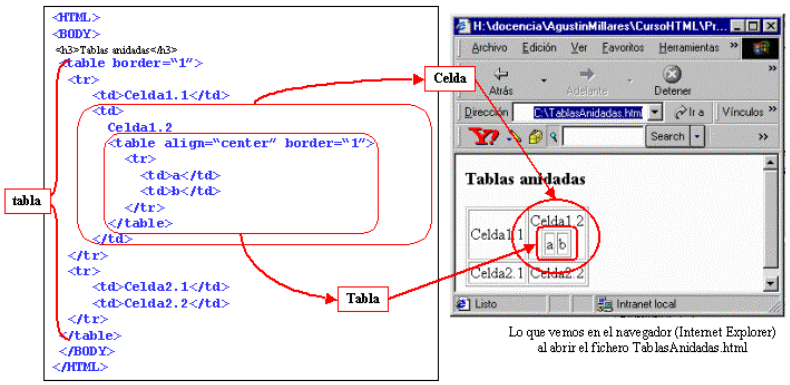
De la misma forma que podíamos incluir listas dentro de otras listas, las tablas pueden ser incluidas dentro de otras. Así, podemos incluir una tabla dentro de la celda de otra. El modo de funcionamiento sigue siendo el mismo, aunque la situación puede complicarse si el número de tablas embebidas dentro de otras es elevado.

Ejemplo:

Celda de la tabla principal	Tabla anidada, celda 1 / Tabla anidada, celda 2 Tabla anidada, celda 3 / Tabla anidada, celda 4

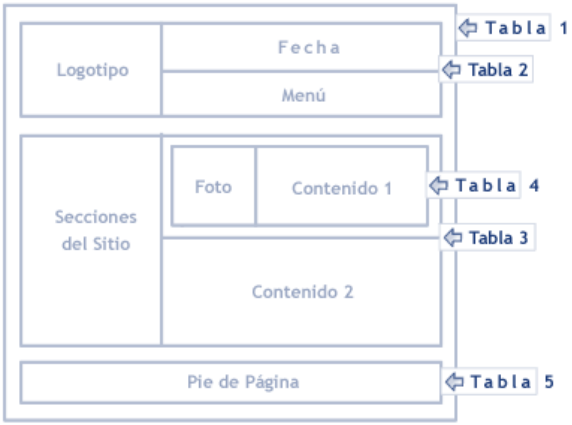
```
<table cellspacing="10" cellpadding="10" border="3">
<tr>
  <td align="center">
  Celda de la tabla principal
  </td>
  <td align="center">
    <table cellspacing="2" cellpadding="2" border="1">
    <tr>
      <td>Tabla anidada, celda 1</td>
      <td>Tabla anidada, celda 2</td>
    </tr>
    <tr>
      <td>Tabla anidada, celda 3</td>
      <td>Tabla anidada, celda 4</td>
    </tr>
    </table>
  </td>
</tr>
</table>
```

Ejemplos:



Lo que escribimos en el editor de texto (Wordpad) y guardamos como fichero TablasAnidadas.html

Lo que vemos en el navegador (Internet Explorer) al abrir el fichero TablasAnidadas.html

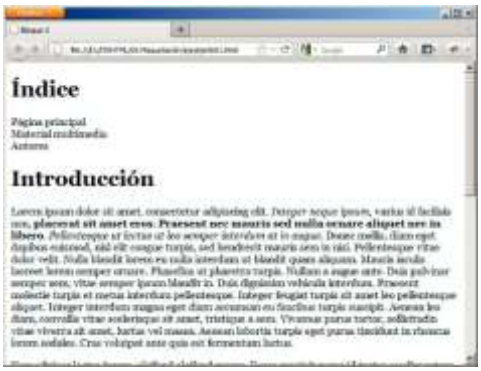


DIVs

Lo usaremos para definir bloques sin ningún tipo de significado, normalmente bloques que usaremos para maquetar correctamente la página o para agrupar elementos en principio sin querer dar un significado específico. Podemos insertar una capa a través de las etiquetas <**div**> y </**div**>, que como ya vimos, sirven para agrupar bloques de texto. A tener en cuenta para el uso de los Divs:

- ✓ A través del atributo id se le da un nombre a la capa, y a través del atributo ***style*** se establecen el resto de propiedades de la capa.
- ✓ A través de las propiedades left (izquierda) y top (superior) se establece la posición de la capa respecto a los márgenes izquierdo y superior de la página. Pueden tomar un número como valor, acompañado de px cuando haga referencia a píxeles, y acompañado de % cuando haga referencia a un porcentaje.
- ✓ Para que la capa aparezca en la posición establecida, es necesario incluir también la propiedad position con el valor *absolute*. Si no se estableciera este valor, la capa se mostraría pegada al margen izquierdo, en la posición en la que hubiera sido insertada dentro del código.
- ✓ A través de las propiedades ***width*** (anchura) y ***height*** (altura) se establece el tamaño de la capa. Pueden tomar un número como valor, acompañado de px cuando haga referencia a píxeles, y acompañado de % cuando haga referencia a un porcentaje.
- ✓ A través de la propiedad ***z-index*** puede establecerse el índice de la capa dentro de la página. Una capa podrá ser solapada por aquellas capas cuyo índice sea mayor. Siempre es un valor numérico.
- ✓ A través de la propiedad ***visibility*** puede establecerse la visibilidad de la capa. Puede tomar los valores *inherit* (se muestra la capa mientras la capa a la que pertenece también se esté mostrando), visible (muestra la capa, aunque la capa a la que pertenece no se esté viendo) y hidden (la capa está oculta).
- ✓ A través de las propiedades ***layer-background-image*** y ***background-image*** se puede establecer una imagen de fondo para la capa. La ruta y el nombre de la imagen han de aparecer entre paréntesis, después de la palabra url.
- ✓ A través de las propiedades ***layer-background-color*** y ***background-color*** se puede establecer un color de fondo para la capa. Ha de ser un número hexadecimal.
- ✓ A través de la propiedad ***overflow*** puede establecerse si se mostrará o no el contenido de la capa cuando no pueda ser visualizado en su totalidad, por ser la capa demasiado pequeña. Puede tomar los valores visibles (se muestra todo el contenido de la capa, aunque esto implique hacer que la capa sea más grande), *hidden* (no es posible visualizar el contenido de la capa que no quepa en ella), *scroll* (se muestra la barra de desplazamiento, aunque el contenido de la capa pueda ser visualizado totalmente) y auto (se muestra la barra de desplazamiento cuando sea necesario).

- ✓ A través de la propiedad ***clip*** puede establecerse el área de la capa que podrá ser visualizado. Lo que hace es recortar la capa, haciendo que partes de ella no sean visibles. Ha de especificarse la distancia de los márgenes de la capa entre paréntesis, después de la palabra url.



El uso de la etiqueta div es sencillo. Observemos este ejemplo:

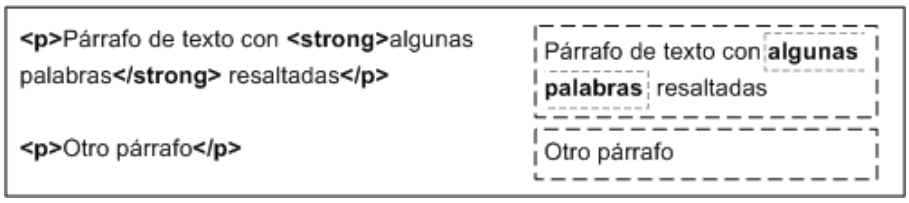
```
<div>
      <h1>Índice</h1>
       Página principal<br />
       Material multimedia<br />
       Autores<br />
</div>
```

Hemos empleado la etiqueta para crear un bloque, que hará las veces de índice de contenidos. Visualmente la etiqueta no provoca ningún cambio, pero en la estructura interna del documento hemos aplicado una división muy importante. Probemos ahora a aplicar una modificación a la apariencia de ese bloque, añadiendo un estilo de borde a la etiqueta ; quedaría así:


```
<div style="border: 2px solid rgb(204, 102, 204);">
      <h1>Índice</h1>
       Página principal<br />
       Material multimedia<br />
       Autores<br />
</div>
```

Las capas (*div*) no son más que unos recuadros, que pueden situarse en cualquier parte de la página, en los que podemos insertar contenido HTML. Dichas capas pueden ocultarse y solaparse entre sí, lo que proporciona grandes posibilidades de diseño.

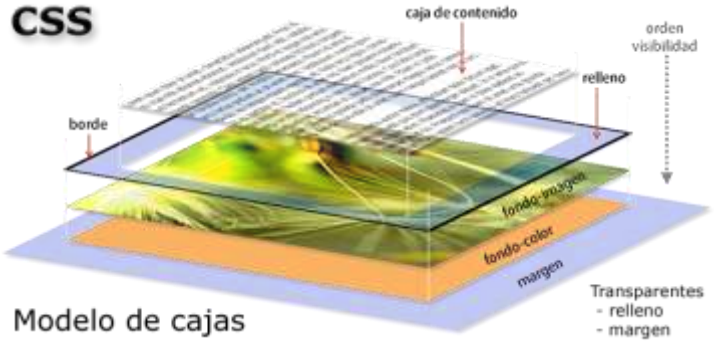
En realidad, todos los elementos de una web (párrafos, enlaces, imágenes, tablas, etc.) son cajas rectangulares. Los navegadores sitúan estas cajas de la forma que nosotros les hayamos indicado para maquetar la página.



Las partes que componen cada caja y su orden de visualización desde el punto de vista del usuario son las siguientes:

- ✓ **Contenido** (*content*): se trata del contenido HTML del elemento (las palabras de un **párrafo**, una imagen, el texto de una lista de elementos, etc.)
- ✓ **Relleno** (*padding*): espacio libre opcional existente entre el contenido y el borde.
- ✓ **Borde** (*border*): línea que encierra completamente el contenido y su relleno.
- ✓ **Imagen** de fondo (*background image*): imagen que se muestra por detrás del contenido y el espacio de relleno.
- ✓ **Color de fondo** (*background color*): color que se muestra por detrás del contenido y el espacio de relleno.

- ✓ **Margen** (*margin*): separación opcional existente entre la caja y el resto de cajas adyacentes.



Padding

Con **padding** establecemos la distancia de "**relleno**" entre el límite interior de la caja y el exterior (borde).Si queremos poner un padding de 20 píxeles para toda la caja, lo haríamos así:

padding : 20 px;

Podemos establecer un *padding* distinto para cada lado, usando los sufijos -top (superior), -bottom (inferior), left (izquierda) y right (derecha):

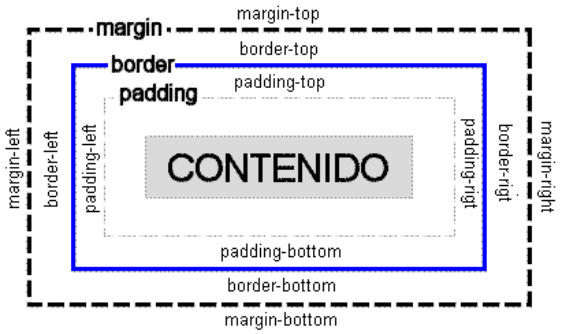
```
padding-top: 10px;
padding-bottom: 5px;
padding-left: 30px;
padding-right: 20px;
```

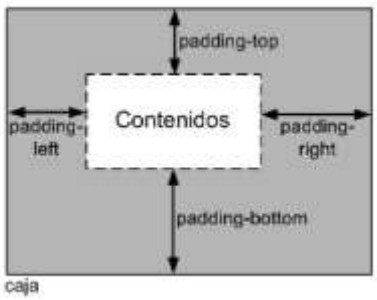
Podemos abreviar lo anterior en una sola línea, indicando primero el *padding* superior y luego siguiendo el orden de las agujas del reloj. Es decir, nos quedaría: *arriba*, *derecha*, *abajo*, *izquierda*. El ejemplo anterior se acortaría así:

padding: 10px 20px 5px 30px;

Otro atajo útil es especificar sólo dos medidas: una corresponderían al *padding* superior e inferior, y la otra al lateral. Si queremos que los paddings superior e inferior sean de 10 píxeles, y los laterales (izquierdo y derecho) de 20 píxeles, escribimos:

padding: 10px 20px;





Bordes

Si queremos que nuestra caja tenga **bordes**, lo conseguimos con la propiedad border. Tiene la siguiente sintaxis:

border: width | style | color

Como habrás supuesto, *width* especifica el grosor del borde. Normalmente es una medida en píxeles, pero también podemos utilizar las palabras *thin* (fino), *medium* (normal) y *thick* (grueso).

En cuanto a style, es el tipo de borde. Hay bastantes, pero los más comunes son: solid (línea continua), *dashed* (línea discontinua), *dotted* (línea de puntos) y *double* (línea continua doble). Por último, color indica el color del borde.

Podemos escoger un tipo de borde diferente para cada lado con los sufijos *-top, -bottom, -left* y *-right*.

border-bottom: 1px dotted #f00;

Para eliminar el borde, simplemente ponemos que tiene de grosor 0 píxeles o que el estilo del borde es *none*. Esto es muy importante con las imágenes, ya que, si tenemos una imagen enlazando a algo, los navegadores la ponen con un reborde muy feo. Así que esto se ha convertido ya en un fijo de las hojas de estilos:

img { border: none; }

dotted

dashed

solid

double

groove

ridge

inset

outset

```
div {  border-top-width: 10px;  border-right-width: 1em;
  border-bottom-width: thick;  border-left-width: thin;}
```


```
div {  border-top-color: #CC0000;  border-right-color: blue;
border-bottom-color: #00FF00;  border-left-color: #CCC;}
```

Para establecer de forma simultánea los estilos de todos los bordes de una caja, es necesario utilizar la propiedad ***"shorthand"*** llamada ***border-style***:

```
-top-width: 6px;
  border-left-width: 8px;}
```

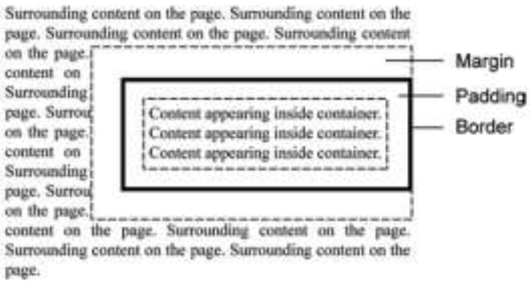
Márgenes

Los **márgenes** se controlan con la propiedad ***margin***, y es la distancia entre el borde de la caja y los elementos que la rodean. En cuanto a la forma de usarla, es igual que con la propiedad *padding*, así que la forma de escribir y los atajos es exactamente la misma. Por ejemplo, si queremos márgenes superior e inferior de 20 píxeles, y laterales de 5 píxeles:

margin: 20px 5px;

Para centrar un elemento de bloque, podemos hacer uso de auto:

margin: 0px auto;



Márgenes negativos

A todas las propiedades de márgenes se les pueden asignar valores negativos. En este caso, un margen adyacente puede ser "cancelado" a cualquier nivel. Si se aplica un valor lo suficientemente negativo a un elemento lo suficientemente grande, el elemento adyacente afectado puede acabar quedando por debajo del otro.

```
body {background-color:White; font-family:Geneva;}
#header { background-color:yellow; }
h1 { color:red; font-size:2em; }
#content {margin-top:-3em;}
```

Esto provoca el efecto visual de cambiar el elemento de manera que se superponga

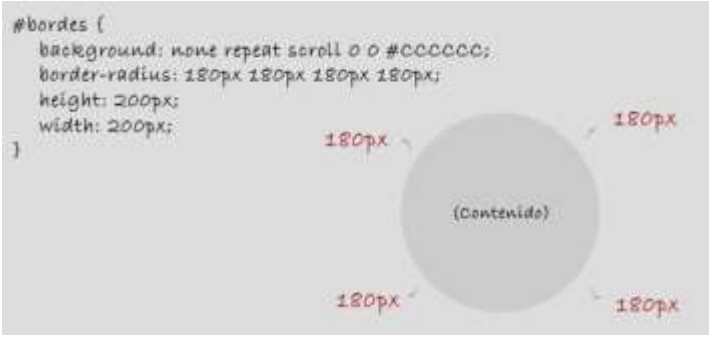
Bordes redondeados

La propiedad *border-radius* nos permite crear esquinas redondeadas sin la necesidad de imágenes o marcas adicionales como lo hacíamos anteriormente que teníamos que cortar los bordes redondeados en photoshop y crear una cantidad de div para dar un aspecto agradable a un bloque o div. Para añadir esquinas redondeadas a nuestra caja, simplemente se añade:

#ejemplo1{ border-radius: 25px;}
#ejemplo2{ border-top-left-radius: 25px; border-top-right-radius: 25px;
border-bottom-right-radius:25px; border-bottom-left-radius: 25px;}

En el código **#ejemplo1** lo que especificamos fue que al elemento que se aplique este estilo va a tener un radio de 25 px en las esquinas, también lo podríamos haber escrito de la forma en que se encuentra **#ejemplo1** pero es mejor simplificarlo como el primero.

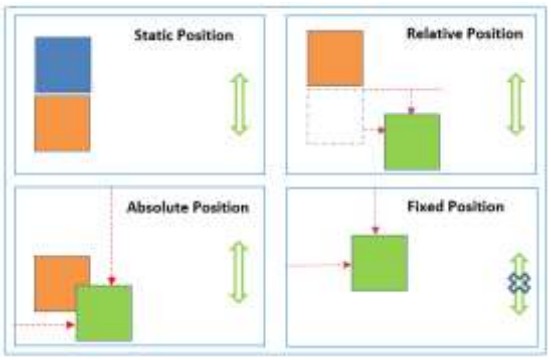
También podemos crear deformaciones de este estilo, podemos jugar con los ángulos para crear objetos que antes eran imposibles con solo css, como te muestro a continuación:

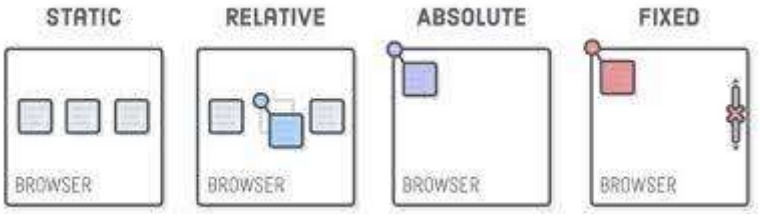


Posicionamiento

Los navegadores crean y posicionan de forma automática todas las cajas o divs que forman cada página HTML. No obstante, CSS permite al diseñador modificar la posición en la que se muestran. Podemos encontrar las siguientes posiciones o situaciones:

- ✓ **Posicionamiento normal o estático:** se trata del posicionamiento que utilizan los navegadores si no se indica lo contrario.
- ✓ **Posicionamiento relativo:** variante del posicionamiento normal que consiste en posicionar una caja según el posicionamiento normal y después desplazarla respecto de su posición original.
- ✓ **Posicionamiento absoluto:** la posición de una caja se establece de forma absoluta respecto de su elemento contenedor y el resto de elementos de la página ignoran la nueva posición del elemento.
- ✓ **Posicionamiento fijo:** variante del posicionamiento absoluto que convierte una caja en un elemento inamovible, de forma que su posición en la pantalla siempre es la misma independientemente del resto de elementos e independientemente de si el usuario sube o baja la página en la ventana del navegador.
- ✓ **Posicionamiento flotante:** se trata del modelo más especial de posicionamiento, ya que desplaza las cajas todo lo posible hacia la izquierda o hacia la derecha de la línea en la que se encuentran.





El posicionamiento de una caja se establece mediante la propiedad ***position.*** El significado de cada uno de los posibles valores de la propiedad ***position*** es el siguiente:

- ✓ ***static***: corresponde al posicionamiento normal o estático. Si se utiliza este valor, se ignoran los valores de las propiedades ***top***, ***right***, ***bottom*** y ***left***.
- ✓ ***relative***: corresponde al posicionamiento relativo. El desplazamiento de la caja se controla con las propiedades ***top***, ***right***, ***bottom*** y ***left***.
- ✓ ***absolute***: corresponde al posicionamiento absoluto. El desplazamiento de la caja también se controla con las propiedades ***top***, ***right***, ***bottom*** y ***left***, pero su interpretación es mucho más compleja, ya que el origen de coordenadas del desplazamiento depende del posicionamiento de su elemento contenedor.
- ✓ ***fixed***: corresponde al posicionamiento fijo. El desplazamiento se establece de la misma forma que en el posicionamiento absoluto, pero en este caso el elemento permanece inamovible en la pantalla.

La propiedad ***position*** no permite controlar el posicionamiento flotante, que se establece con otra propiedad llamada ***float*** .

Normalmente, cuando se posiciona una caja también es necesario desplazarla respecto de su posición original o respecto de otro origen de coordenadas. CSS define cuatro propiedades llamadas ***top***, ***right***, ***bottom*** y ***left*** para controlar el desplazamiento de las cajas posicionadas:

En el caso del posicionamiento relativo, cada una de estas propiedades indica el desplazamiento del elemento desde la posición original de su borde superior/derecho/inferior/izquierdo. Si el posicionamiento es absoluto, las propiedades indican el desplazamiento del elemento respecto del borde superior/derecho/inferior/izquierdo de su primer elemento padre posicionado. En cualquiera de los dos casos, si el desplazamiento se indica en forma de porcentaje, se refiere al porcentaje sobre la anchura (propiedades ***right*** y ***left***) o altura (propiedades ***top*** y ***bottom***) del elemento.

Posicionamiento normal o estático

El posicionamiento normal o estático es el modelo que utilizan por defecto los navegadores para mostrar los elementos de las páginas. En este modelo, sólo se tiene en cuenta si el elemento es de bloque o en línea, sus propiedades ***width*** y ***height*** y su contenido.

Los elementos de bloque forman lo que CSS denomina ***"contextos de formato de bloque"***. En este tipo de contextos, las cajas se muestran una debajo de otra comenzando desde el principio del elemento contenedor. La distancia entre las cajas se controla mediante los márgenes verticales.

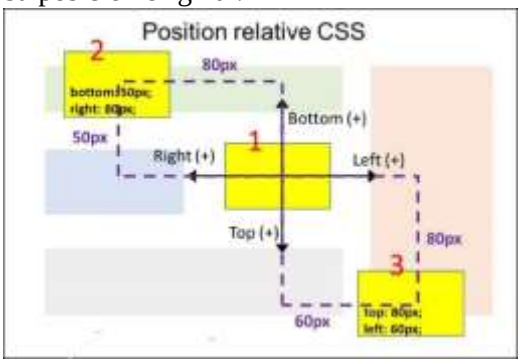
Siempre que creen una caja nueva, les aparecerá en la esquina arriba-izquierda de su contenedor. De esta forma, las cajas (divs, párrafos, etc) se ubicarán uno debajo del otro.

Si un elemento se encuentra dentro de otro, el elemento padre se llama "elemento contenedor" y determina tanto la posición como el tamaño de todas sus cajas interiores.

Posicionamiento relativo

El posicionamiento relativo desplaza una caja respecto de su posición original establecida mediante el posicionamiento normal. El desplazamiento de la caja se controla con las propiedades ***top***, ***right***, ***bottom*** y ***left***.

El valor de la propiedad ***top*** se interpreta como el desplazamiento entre el borde superior de la caja en su posición final y el borde superior de la misma caja en su posición original.



De la misma forma, el valor de las propiedades ***left***, ***right*** y ***bottom*** indica respectivamente el desplazamiento entre el borde izquierdo/derecho/inferior de la caja en su posición final y el borde izquierdo/derecho/inferior de la caja original.

Por tanto, la propiedad ***top*** se emplea para mover las cajas de forma descendente, la propiedad ***bottom*** mueve las cajas de forma ascendente, la propiedad ***left*** se utiliza para desplazar las cajas hacia la derecha y la propiedad ***right*** mueve las cajas hacia la izquierda. Este comportamiento parece poco intuitivo y es causa de errores cuando se empiezan a diseñar páginas con CSS. Si se utilizan valores negativos en las propiedades ***top***, ***right***, ***bottom*** y ***left***, su efecto es justamente el inverso.

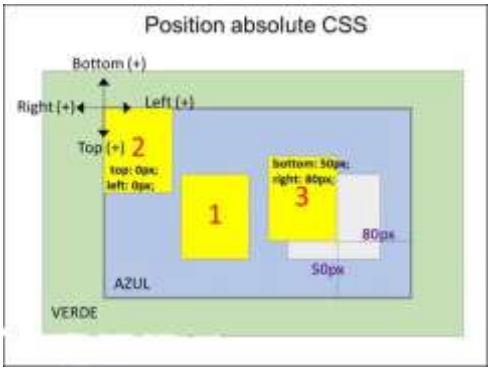
Ahora se le aplicará un posicionamiento relativo para desplazar la primera imagen hacia abajo, por ejemplo:

img.desplazada { position: relative; top: 8em;}

<img class="desplazada" src="imagenes/imagen.png">
<img src="imagenes/imagen.png">
<img src="imagenes/imagen.png">

Posicionamiento absoluto

El posicionamiento absoluto se emplea para establecer de forma exacta la posición en la que se muestra la caja de un elemento. La nueva posición de la caja se indica mediante las propiedades ***top***, ***right***, ***bottom*** y ***left***. Cuando una caja se posiciona de forma absoluta, el resto de elementos de la página se ven afectados y modifican su posición. Al igual que en el posicionamiento relativo, cuando se posiciona de forma absoluta una caja es probable que se produzcan solapamientos con otras cajas.



Este atributo es como *relative* pero mucho más poderoso, nos permite hacer romper el flujo total del elemento, no sólo de la línea donde esté, sino en todo el documento o sitio. Una vez fijado *absolute*, podemos definir top, *bottom*, *left*, *right*, *width*, *height;* a diferencia de *relative* que empieza a contar a partir de la línea donde esté, *absolute* empieza desde el inicio del sitio web, en la esquina superior izquierda. Este atributo es el más usado en la mayoría de aplicaciones y sitios web.

** ** Esta capa tiene posicionamiento absoluto.

Me permite especificar top y left para colocarla con respecto a la esquina superior izquierda.

** **

Por otra parte, el desplazamiento de una caja posicionada de forma absoluta se controla mediante las propiedades top, *right, bottom y left*. A diferencia del posicionamiento relativo, la interpretación de los valores de estas propiedades depende del elemento contenedor de la caja posicionada.

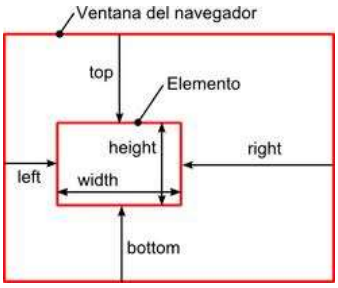
Si se quiere posicionar un elemento de forma absoluta respecto de su elemento contenedor, es imprescindible posicionar este último. Para ello, sólo es necesario añadir la propiedad position: *relative*, por lo que no es obligatorio desplazar el elemento contenedor respecto de su posición original.

Como ejemplo de posicionamiento absoluto, vamos a colocar 4 cajas en cada esquina del documento:

```
#box1 {position:absolute; top: 50px; left: 50px;        }
#box2 {position:absolute;top: 50px; right: 50px;}

#box3 {position:absolute; bottom: 50px; right: 50px;}

#box4 {position:absolute; bottom: 50px;left: 50px;}
```

Posicionamiento absoluto fijo

Es un caso particular del posicionamiento absoluto, ya que sólo se diferencian en el comportamiento de las cajas posicionadas.



Cuando una caja se posiciona de forma fija, la forma de obtener el origen de coordenadas para interpretar su desplazamiento es idéntica al posicionamiento absoluto

La principal característica de una caja posicionada de forma fija es que *su posición es inamovible dentro de la ventana del navegador*. El posicionamiento fijo hace que las cajas no modifiquen su posición ni, aunque el usuario suba o baje la página en la ventana de su navegador.

```
<!DOCTYPE html>
 <head>
  <title>Posicionamiento fijo</title>

  <style type="text/css">
          div#fijo_izquierda {position: fixed;      left: 10%; top: 50px;  width: 100px;
        height:100px;border: #663366 5px double;}

        div#fijo_derecha { position: fixed; right: 10%; top: 50px; width: 100px;
                height: 100px; border: #663366 5px double; }
        div#columna {width: 50%; margin: auto; }
        .verde, .azul {height: 500px; border: #006666 5px dashed;
                border-width: 0 5px 5px 5px;   position: relative;}
```

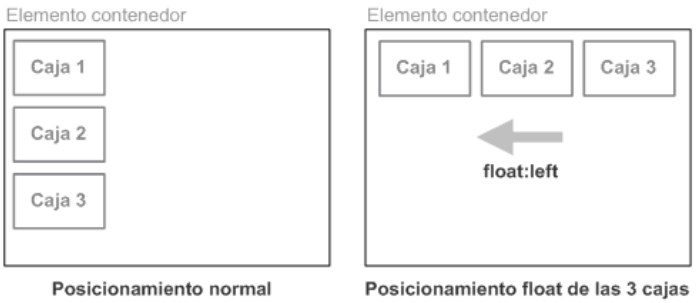
```
        .verde {background-color: #66CC99;}
        .azul {background-color:#6699FF; }
  </style>

</head>
<body>
  <div id="fijo_izquierda">Posicionamiento fijo<br />Izquierda</div>
  <div id="columna">
  <div class="azul"></div>
  <div class="verde"></div>
  <div class="azul"></div>
  <div class="verde"></div>
  <div class="azul"></div>
  <div class="verde"></div>
  <div class="azul"></div>
  <div class="verde"></div>
  <div class="azul"></div>
  <div class="verde"></div>
  <div class="azul"></div>
  <div class="verde">
    <div id="fijo_derecha">Posicionamiento fijo<br />Derecha</div></div>
  </div>
</body>
</html>
```

Posicionamiento flotante

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.

Cuando se posiciona una caja de forma flotante: La caja deja de pertenecer al flujo normal de la página, lo que significa que el resto de cajas ocupan el lugar dejado por la caja flotante. La caja flotante se posiciona lo más a la izquierda o lo más a la derecha posible de la posición en la que se encontraba originalmente.

Si se indica un valor ***left***, la caja se desplaza hasta el punto más a la izquierda posible en esa misma línea (si no existe sitio en esa línea, la caja baja una línea y se muestra lo más a la izquierda posible en esa nueva línea). El resto de elementos adyacentes se adaptan y ***fluyen*** alrededor de la caja flotante.



El valor ***right*** tiene un funcionamiento idéntico, salvo que, en este caso, la caja se desplaza hacia la derecha. El valor ***none*** permite anular el posicionamiento flotante de forma que el elemento se muestre en su posición original.

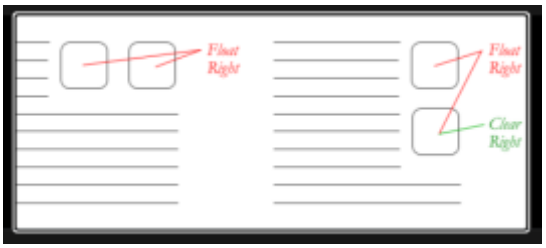
Ejemplo:

img { float: left;}

Uno de los principales motivos para la creación del posicionamiento *float* fue precisamente la posibilidad de colocar imágenes alrededor de las cuales fluye el texto.

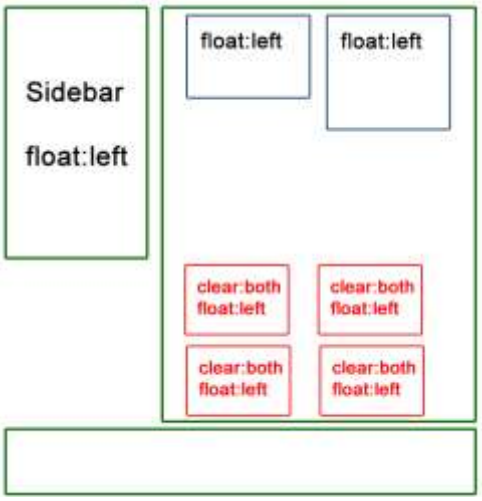
La propiedad ***clear*** permite modificar el comportamiento por defecto del posicionamiento flotante para forzar a un elemento a mostrarse debajo de cualquier caja flotante.

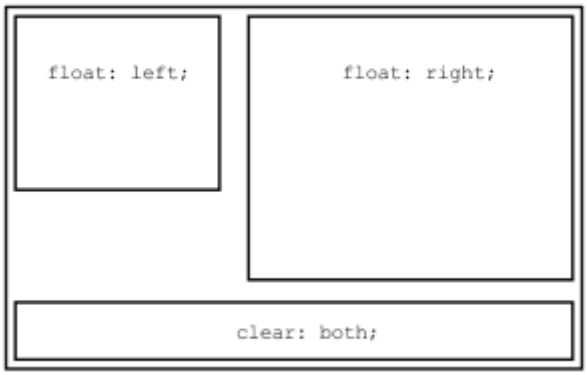
** ... **



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

El valor ***both*** despeja los lados izquierdos y derecho del elemento, ya que desplaza el elemento de forma descendente hasta que el borde superior se encuentre por debajo del borde inferior de cualquier elemento flotante hacia la izquierda o hacia la derecha.





Ejemplo:

```
#paginacion {  border: 1px solid #CCC;
  background-color: #E0E0E0;  padding: .5em; }

.derecha   { float: right; }
```

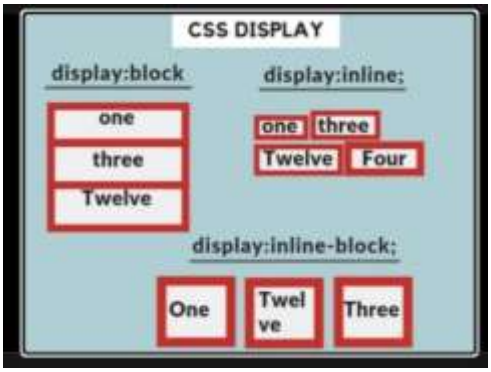
```
.izquierda { float: left;  }

<div id="paginacion">
  <span class="izquierda">&laquo; Anterior</span>
  <span class="derecha">Siguiente &raquo;</span>
</div>
```

Visualización

Las propiedades ***display*** y ***visibility*** controlan la visualización de los elementos. Las dos propiedades permiten ocultar cualquier elemento de la página.

La diferencia entre ambas propiedades es que, mientras display oculta por completo un elemento haciendo que los demás elementos ocupen su lugar, la propiedad ***visibility*** hace que los otros elementos que componen la página respeten la posición de éste y reserven su espacio.

La propiedad "display"



En realidad, la propiedad display modifica la forma en la que se visualiza un elemento. Los valores más utilizados son inline, block y none.

- ✓ ***block***: Muestra un elemento como si fuera un elemento de bloque, independientemente del tipo de elemento que se trate.
- ✓ ***inline***: Visualiza un elemento en forma de elemento en línea, independientemente del tipo de elemento que se trate.
- ✓ ***none***: Oculta un elemento y hace que desaparezca de la página.

- ✓ La propiedad display: inline se puede utilizar en las listas (<ul>, <ol>) que se quieren mostrar horizontalmente
- ✓ display: block se emplea frecuentemente para los enlaces que forman el menú de navegación.

```
div { display:inline;}
```

La propiedad "visibility"

Visibility es una propiedad mucho más sencilla que ***display***, pues sus posibilidades son mucho más limitadas y únicamente permite hacer visibles o invisibles los elementos de una página. Los valores posibles para ***visibility*** son:

- ✓ ***visibility: normal*:** Es el valor por defecto.
- ✓ ***visibility hidden*:** convierte una caja en invisible para que no muestre sus contenidos. El resto de elementos de la página se muestran como si la caja todavía fuera visible, por lo que en el lugar donde originalmente se mostraba la caja invisible, ahora se muestra un hueco vacío
- ✓ ***visibility: collapse*:** sólo se puede utilizar en las filas, grupos de filas, columnas y grupos de columnas de una tabla. Su efecto es similar al de la propiedad display, ya que oculta completamente la fila y/o columna y se pueden mostrar otros contenidos en ese lugar. Si

se utiliza el valor *collapse* sobre cualquier otro tipo de elemento, su efecto es idéntico al valor *hidden*.

```
<head>
< style type="text/css">
h3.se_ve{visibility:visible}
h3.no_se_ve{visibility:hidden}
</style>
< /head>

< body>
<h3 class="se_ve">Este texto es visible.</h3>
<h3 class="no_se_ve">Este texto es invisible.</h3>
```

Este texto es visible.

La propiedad overflow (contenido sobrante)

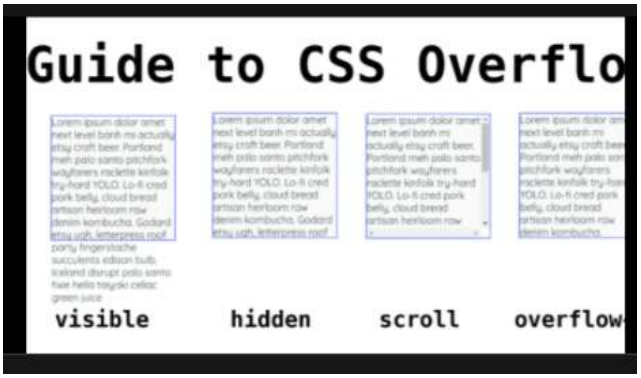
Normalmente, los contenidos de un elemento se pueden mostrar en el espacio reservado para ese elemento. Sin embargo, en algunas ocasiones el contenido de un elemento no cabe en el espacio reservado para ese elemento y se desborda.

La situación más habitual en la que el contenido sobresale de su espacio reservado es cuando se establece la anchura y/o altura de un elemento mediante la propiedad width y/o height.

CSS define la propiedad *overflow* para controlar la forma en la que se visualizan los contenidos que sobresalen (sobrantes) de un elemento.

Los posibles valores para la propiedad overflow son los siguientes:

- ✓ **overflow: *visible***: Es el **valor por defecto**.
- ✓ **overflow:** *hidden*: **el contenido sobrante se oculta** y únicamente se visualizará la parte del contenido que cabe dentro de la zona reservada para el elemento.
- ✓ **overflow: *scroll***: solamente se visualiza el contenido que cabe dentro de la zona reservada para el elemento, pero **se muestran barras de scroll que permiten visualizar el resto del contenido**.
- ✓ **overflow: *auto***: **el comportamiento depende del navegador**, aunque normalmente es el mismo que la propiedad *scroll*. Obviamente, es un comportamiento **totalmente desaconsejado**.



Ejemplo:

```
<div style="overflow: auto; width: 300px; height: 100px; background-color:#ededed;
```

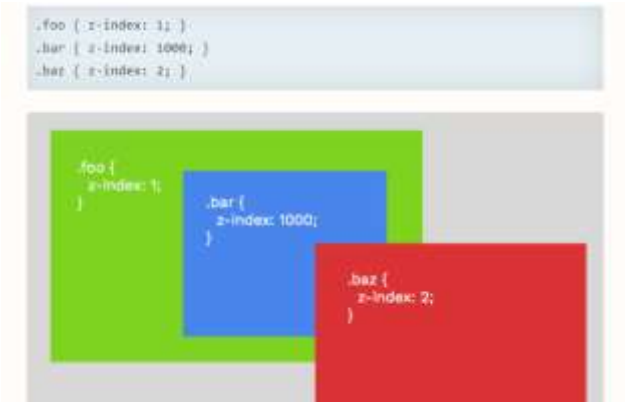
border: 1px solid #990000;">
CONTENIDO....
< /div>

Superposición de cajas

Además de posicionar una caja de forma horizontal y vertical, CSS permite controlar la posición tridimensional de las cajas posicionadas, determinando el orden de superposición de éstas. De esta forma, es posible indicar las cajas que se muestran delante o detrás de otras cajas. Es útil cuando se producen solapamientos.

Utilizando la propiedad z-index es posible crear páginas complejas con varios niveles o capas.

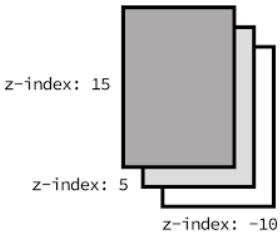
```
.overlay {
  position: absolute;
  top: 0;
  left: 0;
  height: 100%;
  width: 100%;
  background-color: rgba(0, 0, 0, 0.75);
  display: none;
  z-index: 2;
}
```

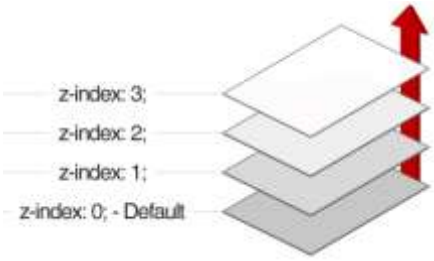


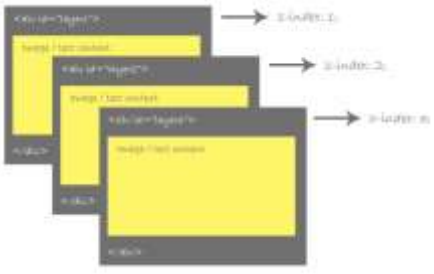
El valor más común de la propiedad ***z-index*** es un número entero. Aunque la especificación oficial permite los números negativos, en general se considera el número 0 como el nivel más bajo. Por lo tanto, cuanto más alto sea el valor más "cerca" del usuario se mostrará la capa (una caja con valor z-index:10 se mostrará por encima que otra con valor z-index:9).

La propiedad z-index sólo tiene efecto en los elementos posicionados, por lo que es obligatorio que la propiedad z-index vaya acompañada de la propiedad position.

Si debes posicionar un elemento, pero no quieres moverlo de su posición original ni afectar al resto de elementos de la página, puedes utilizar el posicionamiento relativo (position: relative).







Ejemplo:

A continuación, se aplican las reglas necesarias para posicionar las cajas de tal forma que la caja 2 quede por encima de caja 1 pero por debajo de caja 3:

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

#caja1,#caja2,#caja3 {  background-color:#f99;  font-size:1.3em;
  margin-bottom:10px;  width:200px;  height:200px; }

caja2 {  position:absolute;  background-color:#ff9;
  left:90px;  top:60px;}

</style>
</head>
<body>
   <div id="caja1">
      <p>CAJA 1</p>
   </div>
   <div id="caja2">
      <p>CAJA 2</p>
   </div>
   <div id="caja3">
      <p>CAJA 3</p>
   </div>
</body>
</html>
```

Background y sus propiedades

Css3 nos trae nuevas formas de controlar las imágenes de fondo de cajas, div o de sitios web completos, se trata de multiple background images. Que nos va a permitir colocar varias imágenes de fondo, background-size que nos va a permitir personalizar por medio de css el tamaño de la imagen y adaptarla a su contenedor y background-origin que nos va a permitir especificar el posicionamiento de una imagen de acuerdo a su área, por ejemplo. A continuación, veremos cómo se trabaja con ellos.

Múltiples imágenes de fondo

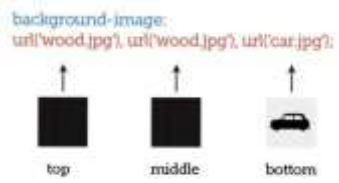
La sintaxis para esta operación es la siguiente:

background-image: url(sol.jpg), url(vaca.png), url(paisaje.jpg);

```
background-image: url(http://i.imgur.com/afC0L.jpg), url(http://www.crimsoneditor.com/images/logo.jpg);
background-repeat: no-repeat;
background-position: -10px -10px, right bottom;
```

Podemos crear composiciones compuestas por varias imágenes y ubicarlas en diferentes posiciones para lograr lo deseado. Ejemplo, vamos a ver un paisaje compuesto por 3 imágenes diferentes pero que vamos a integrar en un solo div utilizando múltiples imágenes de fondos.

```
<html>
 <head>
   <style text="text/css">
     div{ height:400px;    width: 635px;
       background-image: url(sol.png), url(vaca.png), url(paisaje.jpg);
       background-repeat: no-repeat;
       background-position: left top, right bottom, 0 0;  }
   </style>
 </head>
<body>
   <div></div>
</body>
</html>
```



Las posiciones que asignamos a cada imagen dependerán del orden en que tengamos las imágenes, desde la primera hasta la última. Podemos abreviar este código de css para aplicar múltiples fondos de imágenes de la siguiente forma.

```
#contenedor{
background: url(imagen1.png) bottom right no-repeat,
url(imagen2.png) center no-repeat,
url(imagen3.pngF) center repeat;
width: 300px;
}
```

div{ background: url(sol.png) left top no-repeat, url(vaca.png) right bottom no-repeat, url(paisaje.jpg) 0 0 no-repeat; }

Ubicación de la imagen de fondo [background-position]

Por defecto, una imagen de fondo se posiciona en la esquina superior izquierda de la pantalla. La propiedad ***background-position*** te permitirá cambiar este valor por defecto y posicionar la imagen de fondo en cualquier lugar de la pantalla que quieras.

Hay muchas formas diferentes de establecer los valores de la propiedad ***background-position***. Sin embargo, todas ellas se formatean como un conjunto de coordenadas. Por ejemplo, el valor '100px 200px' posiciona la imagen de fondo a 100 píxeles del margen izquierdo y a 200 píxeles del margen superior de la ventana del navegador.

Las coordenadas se pueden indicar como porcentajes del ancho de la pantalla, como unidades fijas (píxeles, centímetros, etc.) o puedes usar las palabras "top" (superior), "bottom" (inferior), "center" (centro), "left" (izquierda) y "right" (derecha). A continuación, vemos los siguientes ejemplos:

```
.container{
// Esta es      eje x👇  &  👇 eje y
background-position : 300px  200px;
}
```

Valor	Descripción
background-position: 2cm 2cm	La imagen se posiciona a 2 cm del margen izquierdo y a 2 cm del margen superior de la página
background-position: 50% 25%	La imagen se posiciona en el centro de la página y un 25 % del margen superior de la misma
background-position: top right	La imagen se posiciona en la esquina superior derecha de la página

El ejemplo de código siguiente posiciona la imagen de fondo en la esquina inferior derecha:

```
body {background-color: #FFCC66; background-image: url("butterfly.gif");
```

```
                background-repeat: no-repeat;  background-attachment: fixed;
                background-position: right bottom; }

        h1 {color: #990000; background-color: #FC9804;}
```

Propiedades [background-size]

Esta propiedad nos permite escalar una imagen dispuesta en el fondo. Estas longitudes se las puede especificar en píxeles, porcentajes etc. Por ejemplo, si queremos mostrar tres imágenes dentro de un div con distintos tamaños:

```
<!DOCTYPE  html>
<head>
<title>Prueba</title>

<style type="text/css">

#recuadro1{  background-image:url("logo1.png"),  url("logo1.png"),   url("foto1.jpg");
              background-position: 0% 0%,     50% 50%;
               background-size:     30px 30px, 250px 250px,  700px 450px;
               background-repeat: no-repeat;   width:700px;   height:450px;}
body {  background:white;  margin:50px;}

</style>
</head>

<body>
   <div id="recuadro1">
   </div>
</body>
</html>
```

```
.container{
// Aquí, podemos ver ancho👇  &  👇 alto
  background-size : 200px   200px;
}
```

Si queremos que una imagen tome el tamaño por defecto que tiene la imagen sin reescalar podemos utilizar la palabra clave "*auto*", por ejemplo, para mostrar la primera imagen con el tamaño original podemos escribir:

```
#recuadro1{   background-image:    url("logo1.png"),   url("logo1.png"),  url("foto1.jpg");
  background-position: 0% 0%,    50% 50%;
  background-size:  30px 30px,  250px 250px,    auto auto;
  background-repeat: no-repeat;   width:700px;   height:450px;}
```

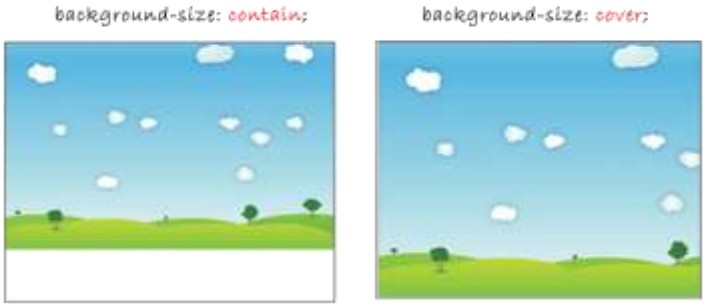
Si utilizamos como unidad porcentajes, la dimensión se basa en el elemento contenedor. Así un ancho y un alto de 50%, por ejemplo, hará que el fondo de la imagen llenar el recipiente 1/4:

```
#recuadro1{   background-image:  url("logo1.png"),  url("logo1.png"),   url("foto1.jpg");
              background-position: 0% 0%,    50% 50%;
              background-size:     50% 50%,   250px 250px,   auto auto;
              background-repeat: no-repeat;   width:700px;   height:450px;}
```

El tamaño también lo podemos definir por medio de porcentaje, como se indicó antes o bien por, cover y contain.

- ✓ ***Cover*:** Escala la imagen para el tamaño más pequeño de tal manera que su anchura y su altura puede caber dentro del área de contenido.
- ✓ ***Contain*:** Escala la imagen al tamaño más grande de tal manera que su anchura y su altura puede caber dentro del área de contenido

```
div{    height:400px;    width: 635px; background:url(paisaje.jpg) 0 0 no-repeat;
   background-size: 50% 400px;   border: 1px solid #777;}
```



Esta propiedad nos permite posicionar el *background* o imagen de fondo a relación del contenido o área de la caja contenedora, podemos hacer que sea relativa al *padding* de la caja por medio de *padding-box*, también por medio del borde de la caja por medio de *border-box* y también la podemos ubicar relativamente a el contenido de la caja por medio de content-box.

```
div{   height:300px;    width: 435px;  background:url(sol.png) 0 0 no-repeat;
          background-origin: content-box;   border: 5px solid #777;   padding: 15px;}
```

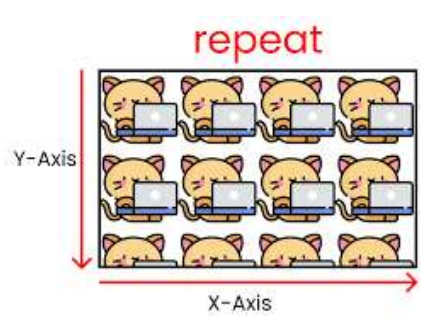
Repetir la imagen de fondo [background-repeat]

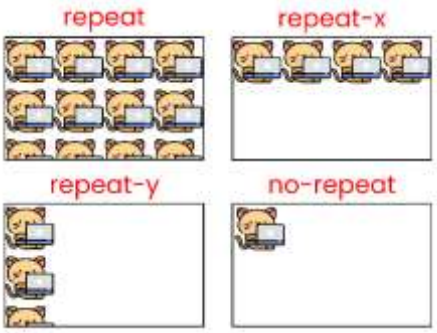
La tabla siguiente resume los cuatro valores diferentes para la propiedad ***background-repeat***.

Valor	Descripción
Background-repeat: repeat-x	La imagen se repite en el eje horizontal
background-repeat: repeat-y	La imagen se repite en el eje vertical
background-repeat: repeat	La imagen se repite en el eje horizontal y vertical
background-repeat: no-repeat	La imagen no se repite

Por ejemplo, para evitar que se repita una imagen de fondo, el código que tendríamos que usar sería el siguiente:

```
body {background-color: #FFCC66;
        background-image: url("butterfly.gif");
        background-repeat: no-repeat;}

h1 {color: #990000;  background-color: #FC9804;   }
```





Fijar la imagen de fondo [background-attachment]

La propiedad ***background-attachment*** especifica si una imagen está fija o se desplaza con el elemento contenedor.

Una imagen de fondo fija no se moverá con el texto cuando el lector se desplace por la página, mientras que una imagen de fondo no fija se desplazará con el texto de la página web.

Valor	Descripción
Background-attachment: scroll	La imagen se desplaza con la página - no está fija
Background-attachment: fixed	La imagen está fija

Por ejemplo, el siguiente código fijará la imagen de fondo.

```
body {    background-color: #FFCC66;
          background-image: url("butterfly.gif");
          background-repeat: no-repeat;
          background-attachment: fixed;}

h1 {color: #990000; background-color: #FC9804;   }
```

Combinación de propiedades [background]

Con la propiedad ***background*** se pueden comprimir varias propiedades, y así escribir una hoja de estilo de forma más abreviada, lo que facilitará su lectura. Por ejemplo, observa estas cinco líneas de código:

```
background-color: #FFCC66;
background-image: url("butterfly.gif");
background-repeat: no-repeat;
background-attachment: fixed;
background-position: right bottom;
```

Usando ***background*** se puede lograr el mismo resultado con una única línea de código:

```
background: #FFCC66 url("butterfly.gif") no-repeat fixed right bottom;
```

El orden en que deben aparecer las propiedades individuales es el siguiente:

[background-color] | [background-image] | [background-repeat]| [background-attachment] | [background-position]

Si se omite alguna propiedad, de forma automática ésta se establecerá con su valor por defecto. Por ejemplo, si se omiten las propiedades ***background-attachment*** y ***background-position*** del ejemplo anterior, quedando el código de la siguiente manera:

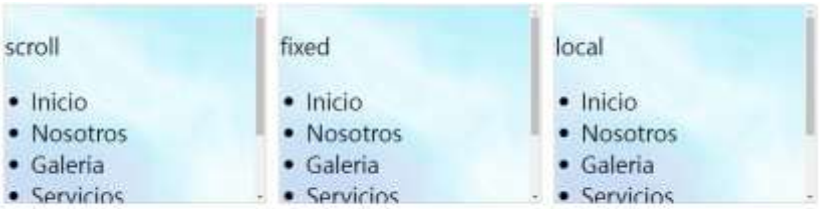
background: #FFCC66 url("butterfly.gif") no-repeat;

```
/* Color e imagen de fondo de la página mediante una propiedad shorthand (o atajo) */
        body { background: #222d2d url(./graphics/colores.gif) repeat-x 0 0; }

/* La propiedad shorthand anterior es equivalente a las siguientes propiedades*/
        body {   background-color: #222d2d;
          background-image: url("./graphics/colores.gif");
          background-repeat: repeat-x;
          background-position: 0 0;}
```

Ejemplo

```
<!doctype html>
  <meta charset="utf-8">

  <style>
   body {  height: 600px;  }

   div { width: 30%;   height: 300px;  overflow: auto;
     float: left;   margin: 0px 10px;
     background-image: url( "FOTO.jpg" );
     background-repeat: no-repeat;
     border: 1px solid #bcbcbc;  font-size: 36px; }

   .jb-scroll { background-attachment: scroll; }
   .jb-fixed {background-attachment: fixed;   }
   .jb-local {   background-attachment: local;  }
  </style>

 <body>
  <div class="jb-scroll">
   <p>scroll</p>
   <ul>
    <li>Inicio</li>
     <li>Nosotros</li>
     <li>Galeria</li>
     <li>Servicios</li>
     <li>Contacto</li>
     <li>Sucursales</li>
   </ul>
  </div>
  <div class="jb-fixed">
   <p>fixed</p>
   <ul>
    <li>Inicio</li>
     <li>Nosotros</li>
```

```
      <li>Galeria</li>
      <li>Servicios</li>
      <li>Contacto</li>
      <li>Sucursales</li>
    </ul>
  </div>
  <div class="jb-local">
    <p>local</p>
    <ul>
      <li>Inicio</li>
      <li>Nosotros</li>
      <li>Galeria</li>
      <li>Servicios</li>
      <li>Contacto</li>
      <li>Sucursales</li>
    </ul>
  </div>
 </body>
</html>
```



Alturas/anchuras máximas y mínimas

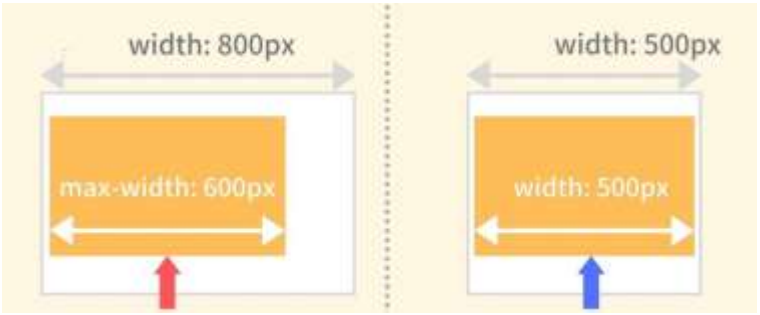
CSS define cuatro propiedades que permiten limitar la anchura y altura mínima y máxima de cualquier elemento de la página. Las propiedades son ***max-width***, ***min-width***, ***max-height*** y ***min-height***. De esta forma, para conseguir un diseño de anchura variable pero controlada, se podrían utilizar reglas CSS como la siguiente:

#contenedor { min-width: 500px; max-width: 900px;}

Las propiedades que definen la altura y anchura máxima y mínima se pueden aplicar a cualquier elemento, aunque solamente suelen utilizarse para estructurar la página. En general, las propiedades más utilizadas son ***max-width*** y ***min-width***, ya que no es muy habitual definir alturas máximas y mínimas.

```
p {
  width: auto;
  min-width: 150px;
  max-width: 600px;
}
```

```
/* ancho completo pantalla */
.site, .site-header {
  max-width:100%;
}


.site, .site-header {
  max-width:100%; /* ancho completo */
}
```



Conclusión

Concluimos nuestra clase viendo las ventajas y la potencia que tiene CSS sobre la estructura HTML para el desarrollo Front de tu sitio.

No olvides leer la clase tantas veces como sea necesario para la comprensión de los temas, recuerda que mientras más programes más experiencia y dominio tendrás sobre el código.

¡A programar …..!

Ejercitación

1. Codifica los siguientes ejemplos, estos te permitirán adquirir habilidad en maquetado.

Ejercicio 1

Animales en peligro de extinción			
Nombre	Cabezas	Previsión 2010	Previsión 2020
Ballena	6000	4000	1500
Oso Pardo	50	0	
Lince	10		
Tigre	300	210	

Ejercicio 2

Climas de América del Sur			
Parte de arriba de América del Sur. Países como:	Venezuela Colombia Ecuador Perú	Parte de abajo de América del Sur. Países como:	Argentina Chile Uruguay Paraguay
Bosque tropical, clima de sabana, clima marítimo con inviernos secos.		Climas marítimos con veranos secos, con inviernos secos, climas frios, clima de estepa, clima desértico.	

Ejercicios 3

DIFERENCIAS ENTRE EL PERRO Y EL HOMBRE			
DIFERENCIAS	PERRO		HOMBRE
	PEQUEÑO	GRANDE	
Duración crecimiento	10 meses	18 a 24 meses	16 años
Tiempo de gestación	58 a 63 días		9 meses
Duración de vida del pelo/cabello	1 año		2 a 7 años

Ejercicio 4

Ejercicio 5

Una tabla de muestra con celdas fusionadas

	Media		Ojos Rojos
	altura	peso	
Machos	1.9	0.003	40%
Hembras	1.7	0.002	43%

Subject	View	Edit	Delete
Accounting			
Animation			
Anthropology			
Archaeology			
Architecture			
Art			

Ejercicio 6

Resultados Globales

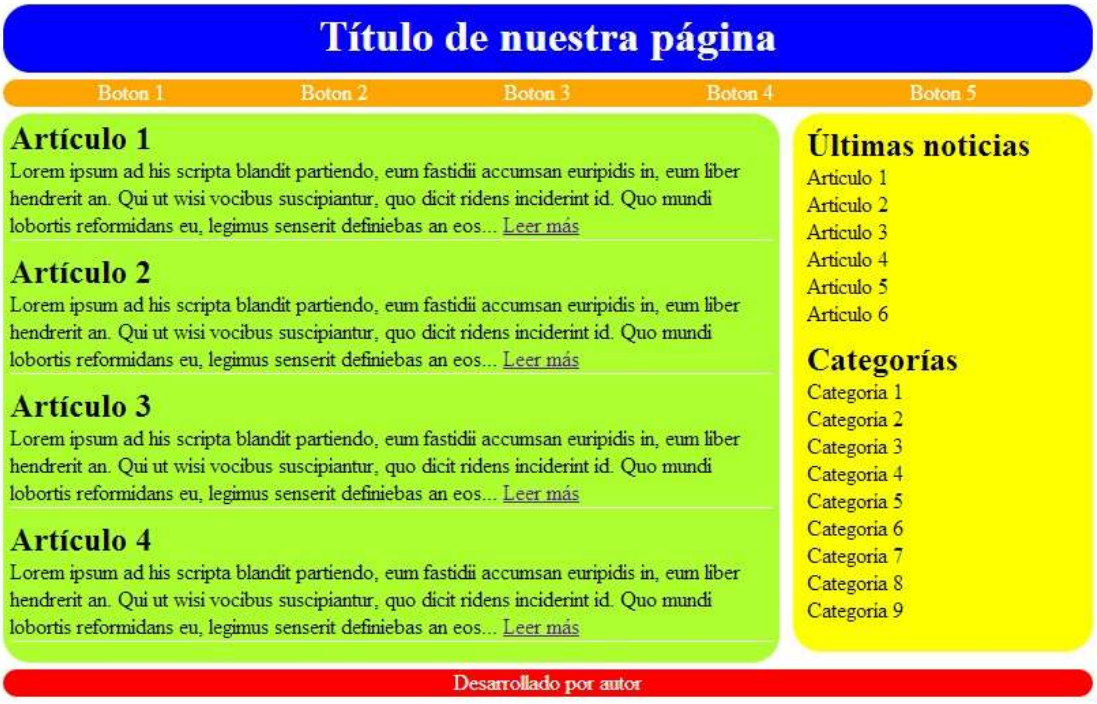
Curso 2019/2010			
1º ESO - A		1º ESO - A	
Aprobados	Suspensos	Aprobados	Suspensos
68%	57%	32%	43%

Ejercicio 7

Lunes	Martes	Miercoles	Jueves	Viernes
Mates	Dibujo			Historia
Fisica	Descanso	Descanso	Descanso	
	Gimnasia	..futbolín..	Gimnasia	Mates

Utilizando Zonas semánticas de HTML5 o contenedores *Divs*, maqueta la siguiente estructura (Puedes utilizar cualquier temática para el desarrollo de las paginas, utiliza texto e imágenes a tu gusto, pero recuerda que debes mantener la maquetación tal cual el ejemplo expuesto a continuación.)

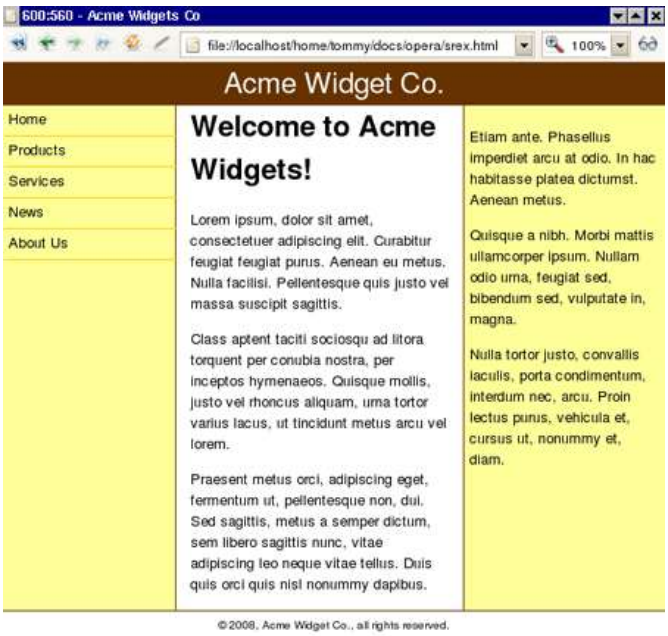
Ejercicio 8



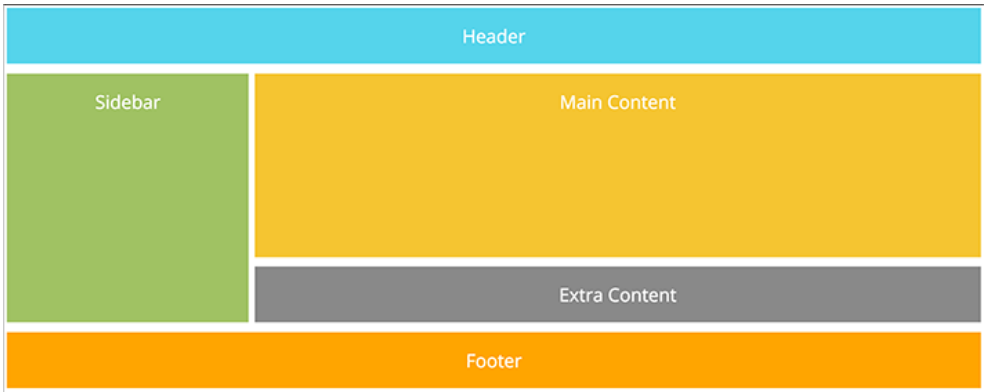
Ejercicio 9

Ejercicio 11

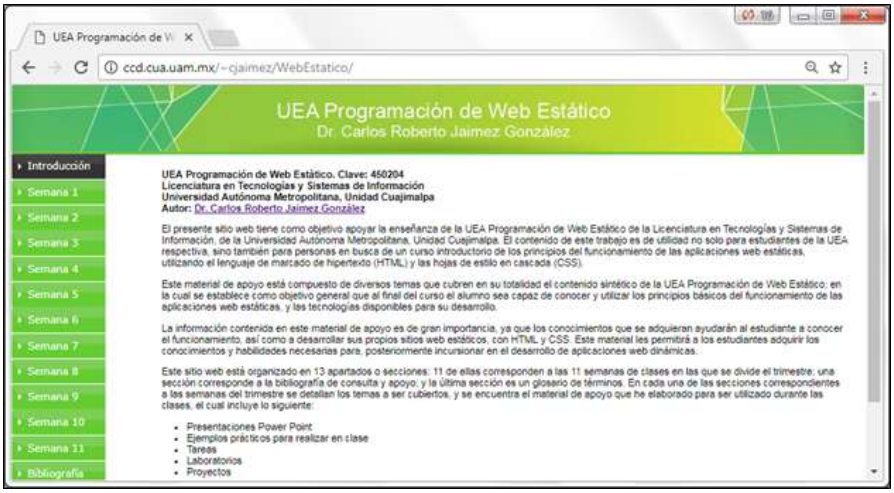
Ejercicio 10



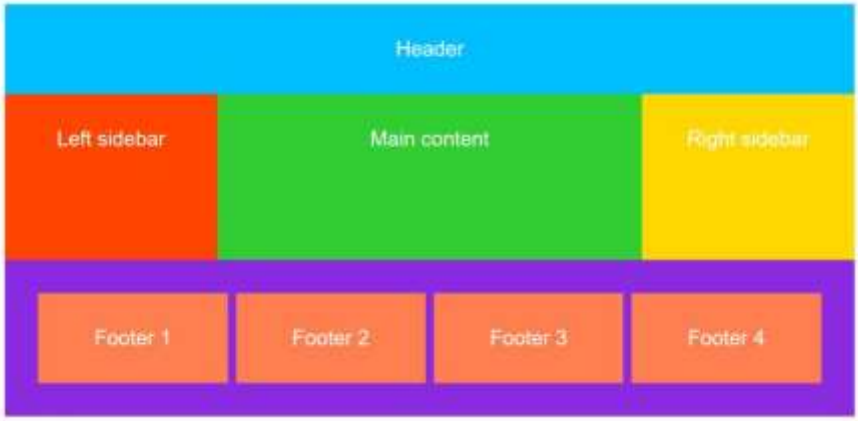
Ejercicio 12



Ejercicio 13



Ejercicio 14.



Ejercicio 15

Autoevaluación

En base a los conocimientos adquiridos, con sus propias palabras responde las siguientes preguntas:

1. Ejemplifica los atributos de bordes para las tablas (rules y frames)
2. ¿Cómo se controla el espacio entre la tabla y los bordes de las celdas?
3. ¿Cómo pueden establecerse anchos diferentes para cada columna?
4. Investiga cual es el uso que se les da a las tablas en HTML5
5. ¿Cómo se coloca una imagen en el fondo de una celda de una tabla?
6. Que hace background-repeat : space ; background-repeat : round ;?