

Tutorial:



Diseño Web con Estándares usando Corel y Hojas de Estilo

Para el Club Hispano Internacional de Usuarios de Corel

WWW.CORELCLUB.ORG

La gran pregunta que la mayoría se hará es Cuál es la diferencia entre crear una página web y crear una página web basada en estándares. La [World Wide Web Consortium](#) (W3C), es el organismo internacional encargado de dictar las tecnologías y pautas para la creación y elaboración de documentos basados en el web. Al crear una página bajo estándares nos aseguramos que será accesible a la mayoría de personas y que se visualizará en la mayoría de los dispositivos con acceso a web. La gran diferencia de ello es que separamos la estructura (el contenido) de la presentación.

Las ventajas de diseñar bajo estándares son principalmente:

- Hacer sitios accesibles a más personas y disponibles para mayor número de dispositivos web (lectores de pantalla, celulares, etc.)
- Reducción de código, lo que equivale a páginas que se cargan más rápido.
- Mejor y más fácil actualización y mantenimiento del sitio.
- El marcado correcto permite a los buscadores mejor indexación y por ende, mejor posicionamiento para nuestro sitio.

El método arcaico era crear un sin fin de tablas según cómo habíamos diseñado nuestro boceto, atiborrar el código html de etiquetas y terminar con un archivo kilométrico y difícilmente adaptable si queríamos modificar posiciones, colores, contenido, etc.

Qué haremos

A lo largo de este tutorial crearemos una página web de AMAZON ACTION, un ecoalbergue turístico en la selva peruana, considerando todas las pautas y también beneficios que nos da diseñar bajo estándares web.

Qué usaremos

Usaremos 3 programas: CorelDRAW, Corel PHOTO-PAINT y Bloc de notas, aunque muy recomendable en lugar de este último: HTML-kit o Araneae (ambos gratuitos).

Que... ¿no necesito un "super-programa-web"?

Si poseemos algún editor de HTML como Dreamweaver, GoLive o Front Page, bien. Aunque no es necesario.

La interrogante común es qué programa se puede usar para armar una página web. La suite de Corel no está optimizada para elaboración de sitios web pero nos ayuda invaluablemente para diseñar, probar colores, formas, tamaños, etc. Con DRAW podemos crear formas y planificar. Con PHOTO-PAINT podemos editar las imágenes y luego optimizarlas.

El marcado de las páginas web con estándares es mucho mas ligero y fácil que uno basado en tablas, de modo que nuestro enfoque será utilizar un editor sencillo (y gratuito) de html: Araenae o HTML-kit y aprender cómo insertar el contenido a la página, o sea, el texto, las imagenes, etc. El diseño y diagramación es otro cantar gracias a las CSS (hojas de estilo en cascada).

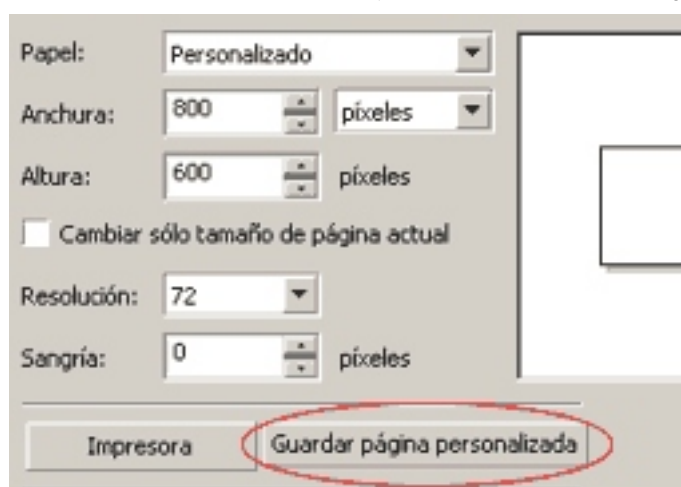
Paso I: CorelDRAW, el Arquitecto

Una vez que tenemos la idea del sitio en boceto, podemos valernos de CorelDRAW para pasar a PC aquello que hemos visualizado en papel y estipular los tamaños, formas y colores. Con DRAW tenemos la posibilidad de manipular y modificar mejor de lo que haríamos con un editor de imágenes.

Preparar la página

-Doble clic en el borde de la hoja de documento para configurar el tamaño y la resolución de la página.

-Rellenemos las casillas en el siguiente orden: **Medidas:** píxeles, **Resolución:** 72 o 96, **Anchura:** 800, **Altura:** 600. En internet las medidas se dan en píxeles, de modo que todos los elementos deberán mostrarse así. Las dimensiones del documento junto con la resolución ppp. debe imitar las condiciones de la mayoría de monitores.



Usualmente elijo la resolución de 72 para la página como también para las imágenes en PHOTO-PAINT ya que eso da menos peso a las imágenes que una resolución de 96, que es la usual en monitores Windows.

- Todavía en el cuadro *Opciones*, elegimos *Herramientas/Caja de Herramientas/Herramienta Zoom* y activamos la casilla '**Relación de 1:1**'. Esto nos permitirá dar un aumento de zoom muy real a cómo se verá en pantalla, aunque no al 100% real.

- Podemos guardar esta configuración de página para tenerla siempre disponible con el botón 'Guardar página personalizada'.

¿Los colores engañan?

CorelDRAW es un programa que por defecto está optimizado para ver los documentos como se verán al ser impresos en colores CMYK. En internet se usan colores RGB o hexadecimales los cuales son mucho más brillantes que los CMYK. Al configurar la visualización web observaremos los colores tal como se verán en internet.

- Para configurar la visualización vamos a *Herramientas/Administración de color* y en el menú desplegable inferior elegimos '**Optimizado para Web**'.



- Para trabajar con las paletas RGB y hexadecimal vamos a *Ventanas/Paletas/ RGB...* o *Colores para web* y cerramos la CMYK.

Podemos elegir el color en modo RGB en la ventana acoplable *Color* y para saber su equivalente hexadecimal, en la misma ventana elegimos *Colores Web* pero en la modalidad '*deslizadores de color*'



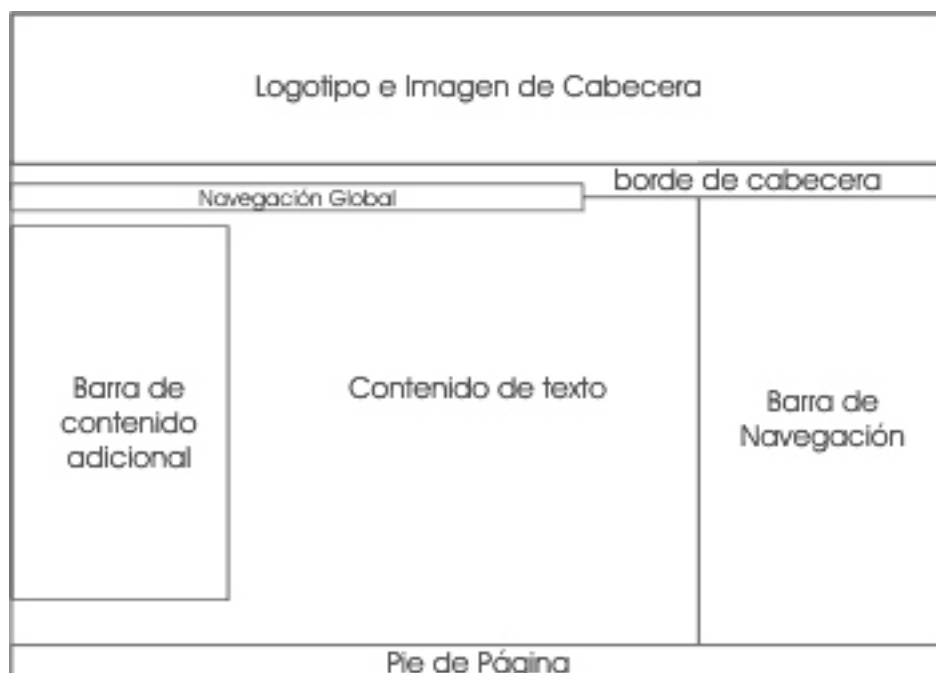
Un color hexadecimal no es un modo de color como el RGB o CMYK sino un valor que se le da a cada color para representarlo en la web. Un valor hexadecimal se compone de 6 elementos entre numeros (0-9) o letras (A-F) precedidos por una almohadilla, como por ejemplo #FFFFFF (color blanco).

Existen 216 colores seguros para web con su respectivo valor hexadecimal, aunque todos los colores RGB tienen su equivalente. Los 216 colores de Colores para web se utilizaban para garantizar una visualización correcta en monitores antiguos. Diseñando para los monitores actuales no necesitamos ceñirnos a esos escasos 216 colores.

Una manera eficaz de probar colores y contrastes en cada elemento es usar la ventana acoplable Color y seleccionar RGB, así podremos arrastrar y cambiar el color interactivamente en cada elemento. Es importante anotar el valor hexadecimal elegido para copiarlo luego en la pagina web.

El tamaño es importante..

Como este es un tutorial web, nos referimos a las dimensiones que tendrán los elementos de nuestra página que unido a los colores y el contenido dará la tónica adecuada al sitio.



Empezaremos creando los bloques grandes de la página, según este boceto:

- Podemos valernos de las líneas guía para encajar nuestros elementos o con la opción de encajar en objetos.

- La creación de los elementos debe ser una aproximación y mejora del boceto original en papel. La gran ventaja es que con CorelDRAW podemos probar los tamaños, los colores, la diagramación, etc.

*Si aún no tenemos las imágenes listas,

podemos reemplazarlas por bloques de colores aproximados a los que tendrá cada imagen ó en todo caso sustituirlas por imágenes de prueba.

Al terminar tenemos una versión muy cercana a lo que será la página real en html:



¿Pero DRAW no tiene un comando para hacer webs?

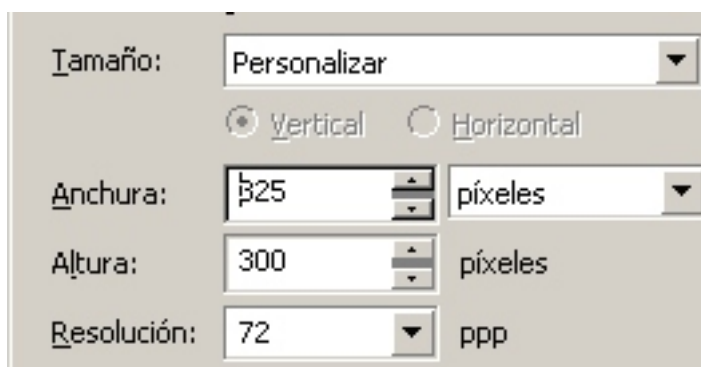
Teóricamente, sí: Archivo/Publicar en web/html. Según dicen, convierte el archivo CDR a HTML, pero su calidad deja muchísimo que desear si buscamos resultados al menos decentes. Además el comando está plagado de errores y no es nada raro que al usarlo, se cuelgue el programa.

La forma más eficaz de crear webs es usando un programa editor de html ya sea a nivel de código o visual (WYSIWYG) que permite ver directamente el resultado final.

Paso 2: PHOTO-PAINT, lo que DRAW no puede

En este paso nos ocuparemos de trabajar nuestras imágenes, para darle la apariencia necesaria, a dividirlas y a optimizarlas

- El primer paso es preparar y crear todas las imágenes a una resolución de 72. Aunque también podemos usar 96, a menor resolución, menos peso. El modo de color, sin duda, RGB.



La mesa de operaciones

La mayoría de imágenes debemos exportarlas y trabajarlas en PHOTO-PAINT ya que podemos darles realces que CorelDRAW no puede hacer. El logotipo de abajo fue creado en DRAW (izquierda) pero le dimos un poco de tridimensionalidad y algún efecto en PHOTO-PAINT (derecha).



- Una opción es copiar de DRAW a PHOTO-PAINT y otra exportar desde DRAW como archivo CPT. Esta última nos da la posibilidad de convertir un documento CDR con sus respectivas capas para poder editarlas independientemente como objetos de PHOTO-PAINT.

- Para ello colocamos cada elemento de DRAW en capas independientes y luego al exportar activamos la casilla correspondiente:



Hora de partir la torta

En ocasiones necesitaremos dividir una imagen en varias secciones, ya sea para optimizar o separarla como archivos independientes. En nuestro caso tenemos la imagen de cabecera que necesita 2 divisiones para optimizarlas de manera individual: el logo necesita una optimización menor que la imagen lateral.

-Para ello nos valemos de la **Herramienta Segmentación de imagen** en la Caja de herramientas y sus opciones en la barra de Propiedades.

Podemos así segmentar la imagen en varias partes y darle a cada una un nombre y una optimización distinta por medio de la ventana acoplable Segmentación de imagen para luego optimizar.



Nombre de División

Elegir formato

Personalizar optimización

La opción *Nombre* es muy importante porque identifica el nombre que tendrá esa sección de la imagen y que se exportará como archivo independiente. *URL*, *ALT* y *Destino* sólo se aplican para los archivos que se exportarán como html por medio de tablas. *Formato* nos permite elegir cómo se va optimizar el area, además de guardar optimizaciones personalizadas con la ayuda del botón *Avanzada*.

Optimizar: el arte de bajar de peso (sin píldoras)

Optimizar una imagen consiste en convertirla a un formato para web que pueda comprimirla lo suficiente para aligerar su peso sin que se deteriore visiblemente. A mayor compresión, menor calidad de imagen y viceversa.

Los formatos válidos para bitmaps en internet son:

GIF: Para imágenes de colores planos sin muchas gradaciones, como logotipos, dibujos, etc. Aceptan hasta 256 colores y tienen soporte para transparencia.

JPG: Imágenes de abundantes colores y gradaciones (como fotografías), tienen buen nivel de compresión pero no soportan transparencia. Este formato, igual que GIF, es de 'compresión con pérdida'

PNG: Es un archivo de compresión sin pérdida. Soporta gran cantidad de colores, transparencia, pero su tamaño suele ser mayor que otros. Útil para imágenes pequeñas pero con mucho detalle. Tiene 2 variantes: el **PNG-8** que es muy similar al GIF pero más eficiente y el **PNG-24** (color verdadero).

Todo en un solo paquete

Aunque PHOTO-PAINT no posee la destreza y la eficacia de otros programas orientados netamente a la web, tampoco se amilana a la hora de optimizar imágenes.

- Si tenemos que optimizar toda la imagen en un solo formato (jpg, gif o png) podemos utilizar directamente el **Optimizador de imágenes** (*Archivo/Optimizador de imágenes*), pero si vamos a usar diferentes niveles o formatos de optimización a cada parte exportándolos como archivos independientes, lo más recomendable es usar la herramienta Segmentación de imagen.

- En ambos casos tenemos la opción de un menú desplegable donde podemos elegir el formato necesario. Para personalizar una optimización recurrimos al botón "Avanzado" disponible en ambas modalidades.



La personalización de la optimización es diferente dependiendo del formato elegido.

En caso de JPG tenemos 2 opciones de "Método de codificación". Si queremos que nuestra imagen aparezca paulatinamente mientras se carga la página usamos el método "Progresivo", eficaz para imágenes pesadas. La elección estándar es "Optimizar".

Si elegimos GIF, las opciones serán aplicar o no transparencia, si queremos que el fondo de la imagen no se visualice. Luego, por medio de un nuevo cuadro de diálogo, tendremos que elegir el número de colores que tendrá el archivo. El peso de la imagen será proporcional a la cantidad de colores incluido.

¿Y acaso no es mejor Optimizar desde CorelDRAW?

Aunque DRAW también posee los comandos respectivos, hacerlo en PHOTO-PAINT es mejor por las siguientes razones:

- DRAW no visualiza las imágenes al 100% de su tamaño real en pantalla sino aproximadamente.
- El comando 'Optimizar imágenes web' de DRAW no exporta los elementos independientemente sino toda la página.
- El comando 'Optimizar imágenes web' de DRAW no respeta el tamaño estipulado; casi siempre es mucho menor.
- El comando Exportar como GIF o JPG de DRAW añade un borde blanco de 1 px a los elementos.
- No se pueden añadir realces o efectos bitmaps a los objetos.



Calidad Exportación

Una vez optimizadas las imágenes necesitamos exportarlas a una carpeta que hayamos elegido dentro de la carpeta del sitio. Así será fácil colocar en el código xhtml la ruta respectiva, por ejemplo:

Ubicación: mi_sitio/imagenes_de_mi_sitio/mi_imagen.jpg



Para evitar problemas en la transferencia al servidor de internet todos nuestros carpetas y archivos deben EXCLUIR: acentos, eñes, caracteres raros, espacios y todo debe ir en minúsculas.

- La exportación la podemos hacer de 2 formas: con el comando **Optimizador de imágenes** o **Exportar para Web**. Si previamente hemos usado la herramienta Segmentación de imagen y tenemos varias divisiones, el Optimizador muestra en la parte superior dos opciones que se muestran en la imagen de arriba.

Si optamos por mantener segmentos, veremos el cuadro Exportación como figura a la derecha.

Solo las imágenes: cada archivo se exportará con el nombre que le dimos al segmentarlo.

Html e imágenes: se exportarán éstas junto con el código html para ensamblar el archivo segmentado. Esta no es la mejor elección, porque crea tablas para el armado.



Con las imágenes listas, ya estamos preparados para crear el armazón de la página.

¿Y qué de las 'imágenes cambiantes' de Corel?

Se entiende que esas famosas imágenes se usan como rollovers de Java Script (cambian al pasar el cursor encima) pero el programa añade como 5 kilos de código entre java script y tablas para hacer visible la gracia. Por nuestra salud, es mejor olvidarnos de las 'imágenes cambiantes' ya que tenemos mejores formas para crear el efecto con vínculos de texto o imágenes gracias a las Hojas de estilo. Lo veremos en el paso 4.

Paso 3: Esqueleto, músculos y tendones

Codificar html a la usanza antigua: con tablas y etiquetas desfasadas, es -qué duda cabe- una tarea titánica y aburrida. Actualmente trabajando bajo los estándares web el código se reduce en forma impresionante, porque no necesitamos atiborrarlo de esos elementos arcaicos.

El xhtml (el sucesor del html) sólo alberga el contenido de la página y no los atributos de su presentación. De modo que podemos codificar manualmente sin miedo valiéndonos de un editor de código html.

Entre los programas para editar código html hay una oferta impresionante. Si tenemos Dreamweaver, Golive, u otro, perfecto pero no es algo que nos deba quitar el sueño. Escribir código bajo estándares reduce abismalmente la labor, así que codificar manualmente es saludable y hasta divertido. Entre los **EDITORES HTML GRATUITOS** de nivel profesional destacan:

HTML-Kit: Potentísimo editor de páginas web, trabaja únicamente a nivel de código. Limpia el código para hacerlo compatible con estándares web. (Español sólo en menús)

First Page 2006: Algunos lo consideran el Dreamweaver gratuito, muy avanzado y con una interfaz super cómoda. Trabaja en vista código, diseño y previsualización. (Inglés)

NVU: Buen editor de páginas, con ventanas de diseño, código y previsualización. (Español)

Amaya: Creado por el W3C, es un híbrido entre navegador y editor de páginas web. Como no podía ser de otro modo, tiene pleno respaldo para estándares web. (Español)

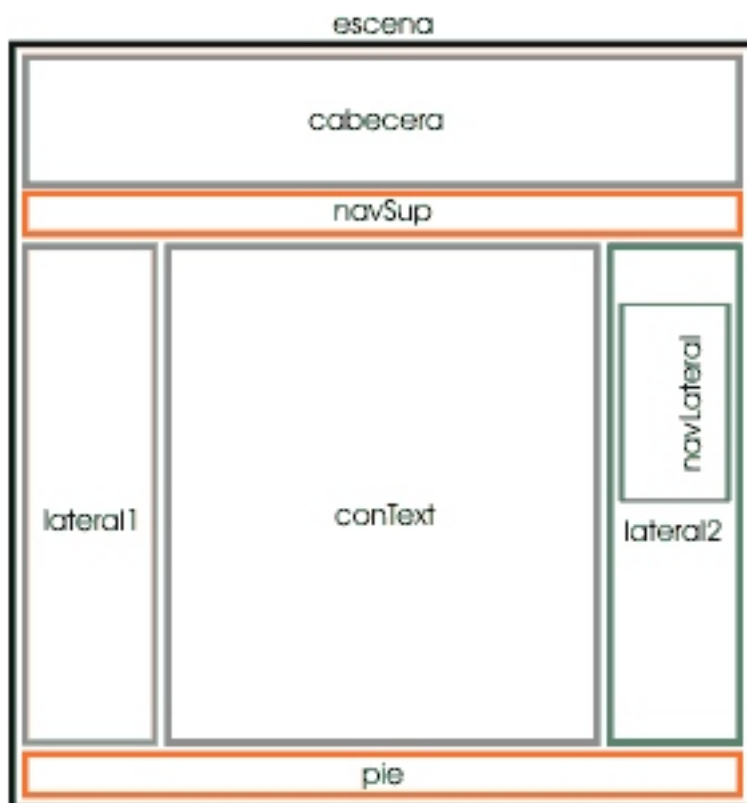
Araneae: Sencillo editor de código muy práctico y mil veces más eficiente que el bloc de notas. Colorea el código y crea plantillas facilitando la edición html. (Español)

Mi abuelita diseñaba con tablas

Las tablas html han sido, durante años la salvación de muchos diseñadores web. Pero ahora estos elementos están desfasados porque añaden demasiado código y no permiten modificar la estructura tan fácilmente. Aunque los estándares no prohíben usarlas, se recomienda hacerlo sólo para mostrar datos tabulados. Su remplazante ideal: las etiquetas DIV, que son una especie de contenedores o bloques donde podemos almacenar los elementos que deseemos.

-A cada elemento de nuestra página lo dividimos en bloques. Según nuestro diseño en CorelDraw nuestros bloques serán los siguientes.

A cada uno tenemos que darle nombres descriptivos. Podemos aplicar bloques dentro de bloques y así sucesivamente.



-Estos bloques se representarán cada uno por medio de capas que utilizarán su respectivo nombre con una etiqueta `<div>` de apertura y una de cierre `</div>`. El div inicial esta acompañado de un Id, es decir, identificador del bloque para que podamos invocarlo por medio de las CSS. Ejemplo:

```
<div id="escena">
.....el contenido de la página...
</div>
```

El Código...¡Ay, mamita!

El código xhtml no es un ser maléfico e indescifrable que muchos alguna vez imaginamos.

El hhtml consiste en una serie de etiquetas básicas, presentes invariablemente en todos los archivos xhtml y otras etiquetas que se usan de acuerdo a nuestras necesidades. El marcado base empieza con la declaración *DOCTYPE* o DTD. Con ello le indicamos a los llamados "Agentes de usuario", como dispositivos (pc, navegadores, teléfonos móviles, lectores de pantalla, etc.) que nuestro documento se ciñe a los estándares web. Es algo así como:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
```

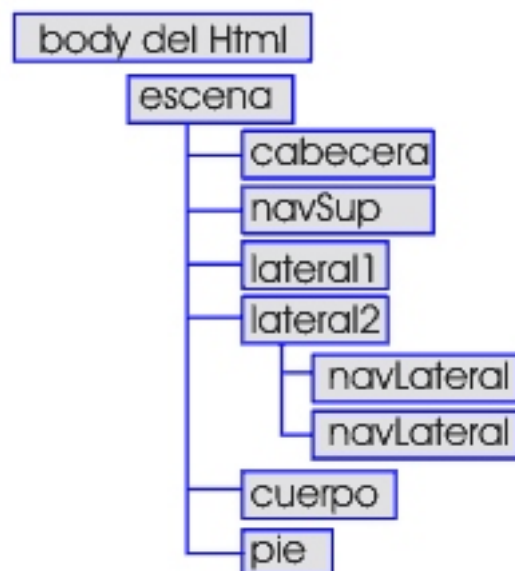
Entonces llegamos al punto que el archivo base será así:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<title>Titulo de Página</title>
<meta http-equiv="Content-Type"
content="text/html; charset=iso-8859-1" />
</head>
<body>
...el contenido de la página....
</body>
</html>
```

Todo lo que se abre, se cierra

Dentro de las etiquetas body de apertura (`<body>`) y de cierre (`</body>`) ubicaremos nuestras etiquetas `<div>` según lo planeado en esta jerarquía que muestra la imagen derecha:

Mandando a la tierra del olvido las aburridas y geriátricas tablas, el código con su jerarquía será tan simple como esto:



```
<body>

<div id="escena">
  <div id="cabecera">Contenido de cabecera</div>
  <div id="navSup">Contenido de navSup</div>
  <div id="lateral1">Contendio de la lateral1</div>
  <div id="lateral2">Contendio de la lateral2</div>
    <ul id="navLateral">contenido de navLateral</div>
    <ul id="navLateral2">contenido de navLateral2</div>
  <div id="conText">Contenido del texto</div>
  <div id="pie">Contenido de pie</div>
</div>

</body>
```

- Nunca debemos olvidar cerrar una etiqueta, ya sea esta DIV u otra de xhtml para evitar errores en la página. Cuando aumenta nuestro contenido, a veces resulta bastante difícil ver claramente cuál es la etiqueta de cierre de un bloque DIV determinado. Por esto, una muy buena práctica es poner comentario al cerrar la etiqueta DIV, por ejemplo:

```
<div id="lateral2">
  ...Contenido de lateral muy extenso y con varios bloques anidados...
</div><!--fin de #lateral2-->
```

¿Pero es HTML o XHTML?

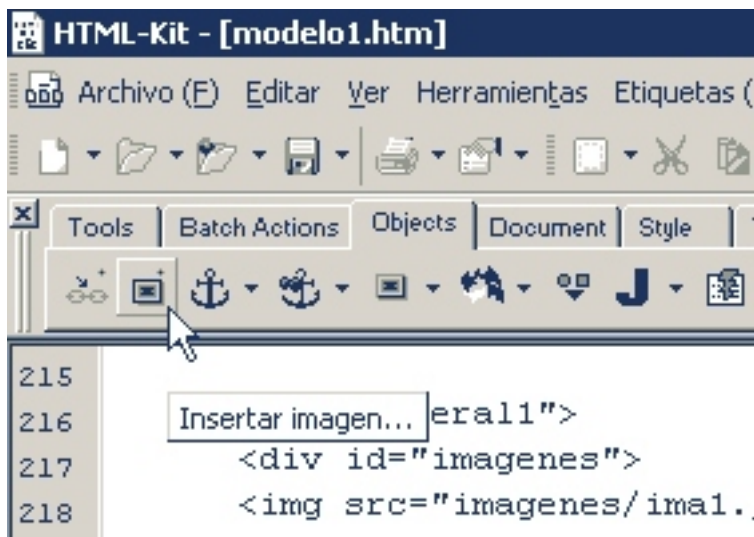
El XHTML es el reemplazante del html, y está basado en el XML, un lenguaje que tiene por finalidad describir datos y no mostrarlos. El XHTML es html mejorado, y la base de los sitios con estándares, permitiendo que cualquier dispositivo pueda mostrar nuestras páginas y que éstas sean accesibles al mayor número de personas.

[Esta es una guía breve sobre el XHTML.](#)

Llenando el carrito del 'Super'

Ahora podemos insertar el contenido en su bloque DIV correspondiente. El texto lo podemos copiar y pegar de otra aplicación. Para las imágenes nos valemos de el comando 'Insertar imagen' de HTML-KIT:

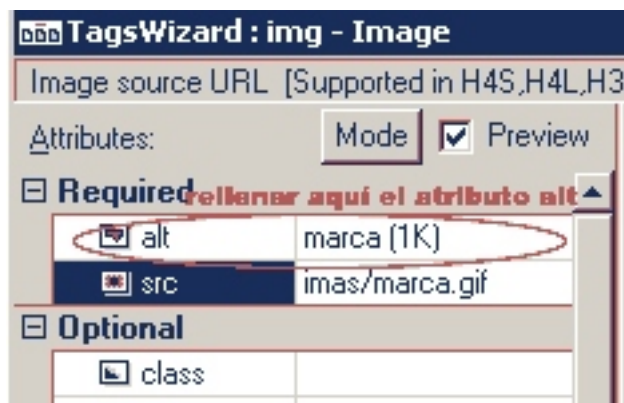
- Para cumplir uno de los requisitos de Accesibilidad web, es importante incluir la etiqueta **ALT** para todas las imágenes que insertemos. Si usamos HTML-KIT, podemos hacerlo en la ventana que nos aparece al insertar imagen.



Una vez insertada nuestra imagen con su atributo ALT, el código quedará así:

```

```



La importancia de llamarse ALT

Definir todas las etiquetas ALT para nuestras imágenes no es snobismo de webmaster sino un importante requisito de accesibilidad. Por ejemplo, si un usuario invidente utiliza un lector de pantalla para escuchar el contenido web, el dispositivo recurrirá a la etiqueta ALT para describirle su contenido.

[Para conocer más sobre Accesibilidad](#)

- Todos los párrafos deben estar contenidos entre etiquetas <p> y </p> para el marcado correcto, mientras que el encabezado, los títulos y subtítulos deben marcarse con etiquetas h1, h2, h3, h4 h5 o h6 según necesitemos para establecer jerarquías de contenido textual.

```
<h1>Este será el encabezado</h1>
```

- Para la barra de navegación superior y lateral crearemos listas sin ordenar. Un ejemplo de lista en xhtml sería:

```
<ul>
<li>>China</li>
<li>>Japón</li>
<li>>Korea</li>
</ul>
```

El resultado:

- China
- Japón
- Korea

El bloque de la barra de navegación superior, contiene una lista, por eso lo llamamos así

```
<ul id="navSup">
```

Así quedará nuestro código de la barra de navegación superior con sus enlaces:

```
<ul id="navSup">
<li><a href="index.htm">Portada</a></li>
<li><a href="albergue.htm">Nuestro Albergue</a></li>
<li><a href="proyectos.htm">Proyectos</a></li>
<li><a href="tarifas.htm">Tarifas</a></li>
<li><a href="expediciones.htm">Expediciones</a></li>
<li><a href="mapas.htm">Mapas</a></li>
<li><a href="contac.htm">Contáctenos</a></li>
</ul>
```


Este será nuestro código completo con todo el "carrito lleno". Y no hemos necesitado los favores de "San Dreamweaver" ni del señor Frontapage, ni de ningún otro santo.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Página Modelo</title>
</head>

<body>
<div id="escena">
  <div id="cabecera">
    
  </div>

  <ul id="navSup">
    <li><a href="index.htm">Portada</a></li>
    <li><a href="albergue.htm">Albergue</a></li>
    <li><a href="proyectos.htm">Proyectos</a></li>
    <li><a href="tarifas.htm">Tarifas</a></li>
    <li><a href="expacts.htm">Expediciones & Actividades</a></li>
    <li><a href="mapas.htm">Mapas</a></li>
    <li><a href="contac.htm">Contáctenos</a></li>
  </ul>

  <div id="lateral1">
    El contenido opcional de 'lateral 1'
  </div>

  <div id="lateral2">
<h2>Expediciones y Actividades</h2>
<ul id="navLateral">
  <li><a href="proyectos.htm">Expedición Nocturnas</a></li>
  <li><a href="expacts.htm">Expedición de Aves</a></li>
  <li><a href="workshop.htm">Expedición Acuática</a></li>
  <li><a href="workshop.htm">Expedición Científica</a></li>
  <li><a href="workshop.htm">Otras Actividades</a></li>
</ul>
<h2>Tarifas</h2>
<ul id="navLatearl2">
  <li><a href="tarifas.htm">Tarifas</a></li>
  <li><a href="info.htm">Useful Info</a></li>
</ul>
</div>

  <div id="conText">
    <h1>Expedición Acuática</h1>
    <h3>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut sodales tincidunt
mauris. In posuere porttitor nisl. Donec commodo egestas turpis. Nullam faucibus
```

```
vulputate neque.Quisque ac sapien.Mauris at odio. Sed turpis dolor, volutpat eget, aliquam id, ultricies ut,nunc. Duis nisi tellus,pulvinar et, vulputate vel, consequat et, nunc. Phasellusvolutpat nisl non metus.Nulla accumsan interdum quam. Quisque eget leo. Vestibulum placeratest adipiscing nulla.Aenean lobortis tempor dui. Aliquam enim purus, ultrices non, imperdiet eu, porta sit amet, elit.</h3>
```

```
<p>Cras sit amet risus. Donec odio. Donec mollis, dui sit amet consequat scelerisque, dolor tellusadipiscing leo, a ultrices eros eros vestibulum neque. Vivamus accumsan, purus non tempus aliquamquam ligula vehicula quam, in ultricies dui nibh non eros. Aliquam nonummy nisl et arcu.Quisque eget elit. Aliquam risus nunc, tempor ullamcorper, congue ut, cursus vitae purus.Mauris dictum dolor at velit. Nulla aliquam pede id dolor. Integer egestas, orci quis pulvinarlacinia, felis odio scelerisque neque, sit amet feugiat massa tortor vel ante. Nulla sapien neque, aliquet venenatis, vestibulum id, tortor. Mauris magna metus, lobortis vitae, dignissim nec,condimentum vitae, lorem. Cras consequat. Donec vel augue vitae erat mollis convallis.</p>
```

```
<p>Aenean libero urna, pharetra in, interdum eget, iaculis eget, nibh. Proin ac metus quis metusconsequat rhoncus. Fusce id arcu. Nulla iaculis purus vel tellus. Morbi vitae est.Nullam posuere tincidunt velit. Phasellus pulvinar, tellus ut fermentum volutpat, tellus feliselementum leo, et pretium est lorem in pede. Vivamus feugiat. Nulla lobortis. Integer porttitorrisus semper ligula. Nullam sollicitudin. Quisque porta mauris eu sapien. Donec ullamcorperalesuada leo. Duis vel justo. Vestibulum euismod sem in erat. Sed tincidunt nulla non tellus.Praesent non leo. Nulla augue justo, volutpat quis, dapibus sed, imperdiet sed, tortor.
```

```
</p>  
</div>
```

```
<div id="pie">  
  <p>AMAZON ACTION Ecologic & Research Center Lodge</p>  
</div>
```

```
</div>
```

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

El archivo xhtml con todas sus etiquetas correspondientes se verá muy plano en un navegador; pero nos garantiza que cualquier lector de pantalla o dispositivo con acceso a internet podrá interpretarlo correctamente.

Nuestra página guardada con extensión **.html** y visualizada en un navegador estará feucha, porque necesita el maquillaje y el piling que sólo CSS le puede dar.

Paso 4: La belleza viene con estilo

Llegó la hora de la diversión. En este último paso vamos usar las Hojas de Estilo para darle a nuestra página las características que tendrá: colores, tamaños, ubicaciones, etc. según el boceto que creamos en CorelDraw en el paso I.

Un poco de teoría no hace daño

Las Hojas de Estilo en Cascada o Cascading Style Sheet (**CSS**) son archivos que contienen información sobre la presentación o apariencia que tendrá un documento en cualquier dispositivo. Gracias a los estilos podemos separar la presentación del contenido, una de las principales características del diseño web con estándares. La gran ventaja de las CSS es que modificando el archivo correspondiente actualizamos 1, 20, 50, 1000 o más páginas web que dependan de él en una sola vez.

Para vincular una hoja de estilo externa a nuestra página, basta escribir en el código xhtml, entre las etiquetas <head> y </head> una línea como la siguiente:

```
<head>
  <title>Título</title>
  <link rel="stylesheet" type="text/css" href="miHojadeEstilo.css" />
</head>
```

Aunque no es una práctica muy eficaz - preferible usarla sólo en fase de prueba -, también podemos incrustar una hoja de estilo en la página; basta con añadir una etiqueta de apertura: <style type="text/css"> y una de cierre: </style> y dentro de ellas añadir el contenido normal del archivo css.

Las Hojas de estilo constan de los siguientes elementos:

Selector: Es el elemento al que se aplica la regla de estilo. Puede ser una etiqueta h1 (encabezado), p (un párrafo), img (imagen), un div, una tabla, etc.

Propiedad: CSS consta de muchas propiedades que establecen tamaños, colores, posiciones, orden, etc. de los elementos.

Valor: Consiste en los atributos que tendrá la propiedad. Por ejemplo: el ancho será de tanto, el fondo será de tal color, la fuente será tal, etc. Así tendríamos:

```
#escena {
width: 760px;
}
```

que establece que la etiqueta div llamada "escena" tendrá un ancho (una de sus propiedades) de 760 píxeles (el valor).

Una regla de estilo puede consistir en una o muchas propiedades.

La propiedad y el valor juntos forman la declaración.

Un archivo CSS consta únicamente de una lista de selectores con sus respectivas declaraciones.

Solamente usaremos algunas propiedades. Para ver todas las propiedades que existen en CSS ir a la [Lista Completa de Propiedades CSS](#).

Tipos, clases e ids

Los selectores de CSS pueden ser principalmente:

Selectores de Tipo- tiene influencia sobre los elementos de un determinado tipo como etiquetas h1, h2, img, etc. Cambian la apariencia de todas las etiquetas elegidas. Ejemplo:

```
h1 {  
font-family: Arial, Helvetica, san-serif;  
}
```

mostrará todo el texto de encabezado (h1) con el juego de fuentes elegido.

Selectores de Clase- Permite seleccionar y aplicar estilos a elementos concretos sin que afecte a otros de su mismo tipo. Por ejemplo, en el código xhtml se ha incluido el selector de clase "resalte":

```
<p class="resalte">Este es un párrafo que no tiene mucho que decir.</p>
```

Y en la hoja de estilo correspondiente, la regla de estilo que viene precedida por un punto indica que únicamente ese párrafo en concreto debe ser de color rojo:

```
.resalte {  
color: #f00;  
}
```

Selectores id- Con ellos podemos invocar elementos únicos en nuestra página, como bloques de contenido que no se repiten, etc. Mientras que los selectores de clase recurren al punto, los id van precedidos de una almohadilla (#). Si este fuera el código xhtml:

```
<div id="lateral2"> ...contenido de lateral2... </div>
```

la regla de estilo para establecer su anchura y altura podría ser:

```
#lateral2 {  
width: 130px;  
height: 380px;  
}
```

El más tonto de la clase

Aunque actualmente la gran mayoría de Navegadores tratan de ceñirse a los estándares web, por ahora, el gran problema de diseñar con CSS es que no todos ellos interpretan exactamente igual las hojas de estilo. Concretamente el navegador más popular (si, por supuesto: Internet Explorer) tiene problemas para su correcta interpretación y no admite algunas nuevas propiedades CSS, por eso es bueno probar nuestros avances en los 3 navegadores más usados: Firefox, Opera y bueno, ni modo, IE.

CSS: las chicas Superpoderosas

Cuando digo chicas, aludo a algunas formas de hacer que nuestras CSS sean más compactas, pequeñas y fáciles de modificar; entre ellas están las siguientes:

-Los valores hexadecimales normalmente contienen una almohadilla y 6 dígitos y/o letras. Por ejemplo (#FF0033). Cuando los pares son iguales, podemos representarlo así: #f03. Y para hacer todo más fácil, escribirlo en minúsculas.

- Las Propiedades margin, padding y border deben considerar sus 4 valores posibles: top, right, bottom y left. En ese orden, un ejemplo para algún elemento sería:

```
{  
margin: 0; /*todos los márgenes están en 0 */  
padding: 10px 4px 0 6px; /*relleno superior: 10px, derecho: 4px, inferior: 0  
e izquierdo: 6px */  
border: 2px 4px; /*los bordes superior e inferior tienen un grosor  
de 2px y el derecho e izquierdo de 4px */
```

```
}
```

-Si dos o más selectores comparten las mismas propiedades, podemos agruparlos separándolos con una coma (,) e invocar las propiedades una sola vez para todos ellos:

```
#escena, #laterall, #pie {  
border: 1px solid #693;  
color: #ddd;  
}
```

- Usando selectores descendentes para evitar el uso innecesario y abusivo de selectores class:

```
#conText h3 em {  
color: #f33;  
}
```

Esta regla aplicará color rojo solamente a las palabras en cursiva (em) del encabezado h3 que está dentro del bloque 'conText'.

Una posición envidiable

Ahora es cuándo. Aplicaremos nuestros estilos según el modelo realizado en CorelDraw.

- En una hoja en blanco del editor html que hallamos elegido (ver editores html gratuitos) redactaremos la hoja de estilo. Le damos un nombre distintivo como 'amazon.css' y la guardamos en nuestra carpeta del sitio

Como vimos en la Parte I, la jerarquía del sitio es ésta. Entonces, iremos creando las reglas de estilo en ese orden.

- El body es la base del documento y sus atributos se heredarán a todos sus elementos (hijos). Establecemos el juego de fuentes, el tamaño de ellas, el color de fondo y centramos todo el contenido.

```
body {  
font-family: Verdana, Geneva, Helvetica, sans-serif;  
font-size: 11px;  
background-color: #fff;  
text-align: center;  
}
```

La teoría de relatividad

Los valores más usados de la propiedad position son relative y absolute. La primera mantiene el objeto dentro del flujo normal del documento (uno tras otro), mientras que absolute 'quita' el elemento del flujo normal para ubicarlo independientemente.

En nuestro caso usaremos el bloque 'escena' con posición relativa, lo que nos garantiza que todos los elementos se conservarán englobados dentro de este bloque padre. Y los elementos que tengan posición absoluta o relativa -en su interior- se ubicarán en relación a él.

Aquí tenemos unos [apuntes más detallados sobre posicionamiento CSS](#).

- El bloque "escena" es el contenedor de todos los elementos. Lo ubicamos en el centro con 'margin', con un ancho de 760 píxeles y un borde verde de 1 píxel de grosor. Como hemos centrado todo con el body, tenemos que volver a alinear los elementos a la izquierda con la propiedad 'text-align'.

```
#escena {  
width: 760px;  
background-color: #fff;  
position: relative;  
margin: 0 auto;
```



```
border: 1px solid #688e23;
text-align: left;
}
```

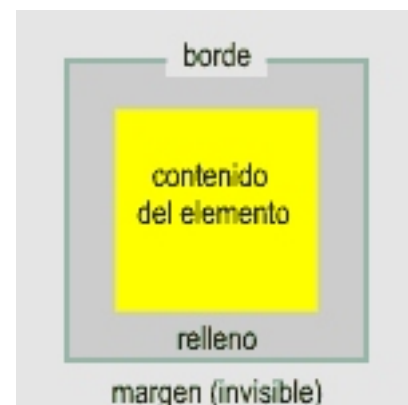
- Vamos a crear los estilos básicos para cada bloque y luego iremos completando cada regla de estilo a medida que chequeamos el resultado. Así, establecemos tamaños de fuentes para títulos y subtítulos. Damos ancho, altura, color de fondo y bordes a los elementos para visualizarlos.

```
* {
margin: 0;
padding: 0;
}
h1, h2 {
font-family: Verdana, Geneva, Helvetica, sans-serif;
font-size: 16px;
}
h3 {
font-family: Verdana, Geneva, Helvetica, sans-serif;
font-size: 12px;
}
#cabecera {
width: auto;
height: 125px;
background-color: #693;
border-bottom: 38px solid #693;
}
ul#navSup {
background-color: #f90;
}
#lateral1 {
background-color: #fff;
border: 1px solid #ddd;
}
#lateral2 {
background-color: #cc3;
}
#conText {
color: #999;
}
#pie {
background-color: #f90;
}
```

Una Caja no tan boba

Ya tenemos nuestros elementos visualmente definidos. Para ubicarlos correctamente usaremos 3 propiedades, a parte de position.

Los elementos de bloque están 'revestidos' por decirlo de alguna forma, de varios niveles o zonas que podemos modificar como propiedades. De dentro hacia afuera tenemos: relleno (**padding**), borde (**border**) y margen (**margin**). La imagen de la derecha es lo que se denomina *modelo de caja*.



Para garantizar la compatibilidad de nuestras CSS en todos los navegadores es preferible empezar estableciendo los márgenes y rellenos en 0. Esto se puede hacer en cada elemento o hacerlo con un selector universal como lo hemos hecho en nuestro caso:

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

Aunque este selector no es muy usado para navegadores antiguos, si diseñamos para la versión 6+ de IE y los otros, nos garantiza que margin y padding estarán predeterminadamente con sus valores en 0.

- En seguida veamos las reglas de estilo que se añaden a cada selector, además de las que ya existían. 'navSup' y 'lateral1' los colocamos absolutamente, quitándolos así del flujo y usando las propiedades top, left y right los posicionamos basandonos en su contenedor padre, es decir, 'escena'. Por ahora ambos se montan uno sobre otro, pero no hay problema, ya lo arreglaremos luego.

```
ul#navSup {  
  position: absolute;  
  top: 155px;  
  left: 0;  
}  
#lateral1 {  
  width: 140px;  
  height: 400px;  
  position: absolute;  
  top: 200px;  
  left: 0;  
  border: 1px solid #ddd;  
}
```

Por otro lado, a 'lateral2', que anidará la barra de navegación lateral, lo hacemos flotar a la derecha. Le damos altura al igual que a 'pie'. Ubicamos el contenido de texto de este último con text-align y padding superior. Le aplicamos color blanco y grosor (font-weight)

```
#lateral2 {  
  width: 160px;  
  height: 480px;  
  float: right;  
  padding-top: 20px;  
}  
#pie {
```

```
background-color: #f90;
width: auto;
height: 22px;
text-align: center;
padding-top: 4px;
color: #fff;
font-weight: bold;
}
```

La cosa va mejorando, pero todavía nos falta ordenar y posicionar los elementos.

¡Qué listas son las listas! (con CSS, claro)

Con CSS es posible reemplazar los menus de navegación repletos de tablas y celdas con simples listas a las que podemos dar la apariencia que se nos antoje y así ceñirnos a los estándares web.

- En nuestro código xhtml tenemos dos etiquetas casi iguales `<ul id="navLateral">` y `<ul id="navLateral2">` que están dentro del div 'lateral2'. Estas son las listas de navegación lateral que trabajaremos ahora. Como son dos listas que tendrán la misma apariencia, podemos agrupar los selectores con una coma. Primero formateamos los títulos que preceden a cada lista (h2) dándoles color tamaño y márgenes, luego ubicamos con padding la posición de 'navLateral' y seguidamente establecemos la distancia vertical (line-height) que habrá entre cada elemento de lista (li). Para terminar, establecemos las características que tendrán los elementos de lista: negrita (font-weight: bold;), eliminamos el subrayado predeterminado que tienen al ser vínculos (text-decoration: none;) y le damos color verde. Cuando pase el cursor encima del vínculo (a:hover), éste cambiará a verde claro.

```
#lateral2 h2 {
font-size: 11px;
color: #000;
margin: 7px 0 4px 10px;
}
ul#navLateral, ul#navLateral2 {
list-style: none;
padding: 5px 5px 5px 12px;
}
ul#navLateral li, ul#navLateral2 li {
line-height: 18px;
}
ul#navLateral li a, ul#navLateral2 li a {
font-weight: bold;
text-decoration: none;
color: #693;
}
ul#navLateral li a:hover, ul#navLateral2 li a:hover {
color: #cf0;
}
```

Texto en la Ciudad

Para aplicar el texto, en el código xhtml hemos definido etiquetas h1, h2, h3 y p. Es importante hacerlo así para establecer una jerarquía de contenido textual que todos los dispositivos con acceso a web interpretarán correctamente.

¿Necesario repetirlo?

Al establecer el juego de fuentes y el tamaño de ellas en el body, tenemos que redefinir para los encabezados y otros, pues sucede que los navegadores dan la opción al usuario de reemplazar etiquetas (h1, h2, p, etc.) o ellos mismos lo hacen si éstas no tienen un estilo especificado en concreto. De modo que entre los primeros selectores de nuestro CSS ya especificamos tamaños y familias para h1, h2 y h3

-El bloque 'conText' que alberga -cómo no- el texto, mantiene su posición predeterminada (static). Le daremos un padding izquierdo de 168 px para 'hacer creer' que flota al lado de 'lateral1'; hacia arriba un relleno de 35 px y hacia abajo 40 px. Su anchura es de 400 px.

A continuación especificamos características para el h1 de 'conText' como color y espaciado entre letras (letter-spacing). El h3 y los párrafos de 'conText' tienen características parecidas: se les ha dado margen superior para separarlos del elemento que los precede. Al final se les ha dado justificado y espaciado entre líneas (line-height).

```
#conText {
width: 400px;
padding: 35px 0 40px 168px;
color: #999;
}
#conText h1 {
color: #f90;
letter-spacing: 1px;
}
#conText h3 {
margin: 2em 0 0 0;
text-align: justify;
line-height: 18px;
}
#conText p {
margin: 2em 0 0 0;
text-align: justify;
line-height: 18px;
}
```

Tu navegas, yo navego

Para finalizar nuestra página modelo (¡ufffff!) daremos forma y brillo a la barra de navegación principal ('navSup') que hasta ahora está apilada abajo de 'lateral1'. A diferencia de 'navLateral', esta barra tiene una posición horizontal y requiere unos pasos adicionales.

- Al conjunto de lista le aplicamos el tamaño de fuente, el color de fondo y lo ubicamos absolutamente en relación al contenedor principal 'escena' con left y top. Seguidamente a la lista y a sus elementos los ubicamos uno seguido de otro (horizontal) con float y eliminamos las viñetas de las listas.

```
ul#navSup {  
  font-size: 10px;  
  background-color: #f93;  
  position: absolute;  
  left: 0;  
  top: 152px;  
}  
ul#navSup li {  
  float: left;  
  list-style: none;  
}
```

- A cada elemento de lista, le damos las características de vínculo que tendrá: color, grosor y estilo de borde. Reiteramos el float y damos un relleno de 3 px. Lo demás se aplica igual que la lista anterior.

```
ul#navSup li a {  
  border-color: #cf0;  
  border-width: 2px 2px 2px 0;  
  border-style: solid;  
  float: left;  
  padding: 3px;  
  color: #000;  
  text-decoration: none;  
}  
ul#navSup li a:hover {  
  color: #fff;  
  background-color: #c93;  
}
```

¿Algún programita que nos ayude con las CSS?

Hay una infinidad de programas dedicados a facilitarnos el trabajo con las hojas de estilo. Al principio usaba TopStyle, o el editor de Dreamweaver, pero con el tiempo descubrí que más rápido resultaba escribir directamente las reglas de estilo directamente en mi archivo css. Con la práctica se vuelve de lo más sencillo. Podemos ayudarnos con cualquier editor html en vista código pues la mayoría de ellos tienen listadas las propiedades CSS y los selectores.

Para seguir viviendo

Dejo algunos enlaces interesantes que son de gran ayuda para continuar y mejorar nuestro aprendizaje en los estándares web y las hojas de estilo.

La World Wide Web Consortium: La web oficial de W3C, el organismo que regula los estándares web. (en inglés).

[W3C-España](#): La oficina del W3C en España.

[HTML con clase](#): Amigable tutorial para aprender html y xhtml.

[Tierra de Nómadas](#): Excelente sitio sobre xhtml, CSS, accesibilidad, etc.

[Web Design Group](#): Guía muy completa para aprender CSS.

[CSS Beauty](#): Sitios diseñados con CSS que inspiran. (en inglés).

[Amazon Action](#): Este es el sitio real que inspiró este tutorial, pero totalmente modificado.

Tutorial: Diseño Web con Corel y Hojas de Estilo

2007© Daniel Laime

Derechos reservados. Prohibida su reproducción sin autorización previa del autor.