

MANUAL DE HTML CSS

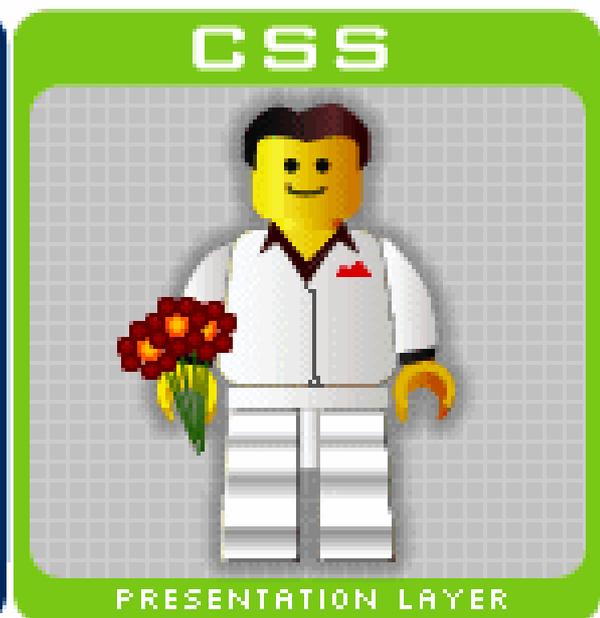
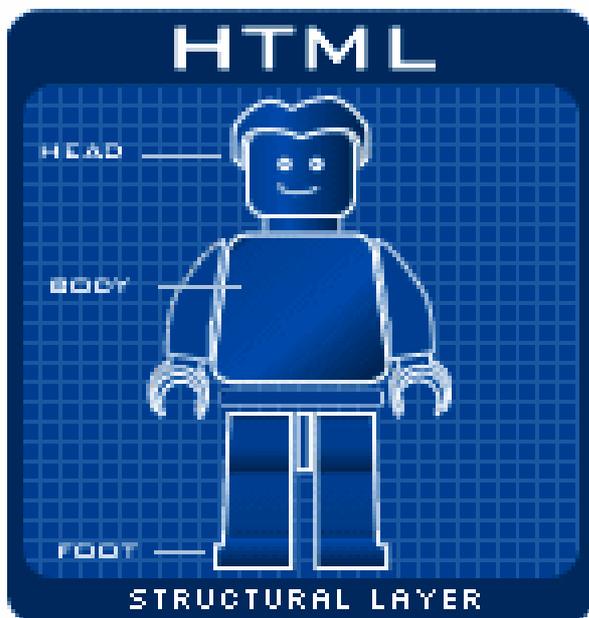


TABLA DE CONTENIDO

GNU Free Documentation License

Licencia de Documentación Libre GNU

Sección 1. Los Principios de HTML

- 1. 1 ¿Qué es HTML?
- 1. 2 Herramientas
- 1. 3 Archivos HTML
- 1. 4 Sintaxis de las etiquetas
- 1. 5 Comenzando
- 1. 6 Etiqueta <HTML>
- 1. 7 Etiquetas <HEAD> y <TITLE>
- 1. 8 Etiqueta <BODY>
- 1. 9 Colores
- 1. 10 Etiqueta <P>
- 1. 11 Comentarios
- 1. 12 Caracteres especiales

Sección 2. Aún mas principios

- 2. 1 Encabezados
- 2. 2 Texto preformateado
- 2. 3 Alineación del texto
- 2. 4 Fuentes
- 2. 5 Dando énfasis
- 2. 6 Trucos molestos
- 2. 7 Líneas horizontales
- 2. 8 Listas
- 2. 9 Saltos de línea.

Sección 3. Vinculando

- 3. 1 Etiqueta <A>
- 3. 2 Vínculos dentro de una página
- 3. 3 Vínculos con otra página
- 3. 4 Vínculos con otros sitios
- 3. 5 Atributo TARGET
- 3. 6 Correo electrónico
- 3. 7 Grupos de noticias
- 3. 8 Sitios FTP
- 3. 9 Archivos
- 3. 10 Ejemplos

Sección 4. Imágenes y píxeles

- 4. 1 Entendiendo los píxeles
- 4. 2 GIF
- 4. 3 PNG
- 4. 4 JPG
- 4. 5 Mas sobre <BODY>
- 4. 6 Etiqueta
- 4. 7 Thumbnails
- 4. 8 Ejemplos

Sección 5. Etiqueta <META>

- 5. 1 Etiqueta <META>
- 5. 2 Sobre HTTP-EQUIV
- 5. 3 Máquinas de búsqueda
- 5. 4 Ejemplos

Sección 6. Accesorios

- 6. 0. Características del sitio
- 6. 1 Galerías de imágenes
- 6. 2 Programas de intercambio de banners
- 6. 3 Buscando un terreno
- 6. 4 Dándonos a conocer
- 6. 5 Apoyando nuestra causa
- 6. 6 Algunas palabras sobre la publicidad

- 6. 7 Estadísticas y contadores
- Sección 7. Tablas
 - 7. 1 Introducción
 - 7. 2 Etiqueta <TABLE>
 - 7. 3 Etiquetas <TR> y <TD>
 - 7. 4 Etiquetas <TH> y <CAPTION>
 - 7. 5 Ejemplos
- Sección 8. Formularios
 - 8. 1 Introducción
 - 8. 2 Etiqueta <FORM>
 - 8. 3 Etiqueta <INPUT>
 - 8. 4 Campos de texto
 - 8. 5 Campos de contraseña
 - 8. 6 Áreas de texto
 - 8. 7 Casillas de verificación
 - 8. 8 Botones de opción
 - 8. 9 Menús
 - 8. 10 Botones
 - 8. 11 Campos de archivos
 - 8. 12 Ejemplo
- Sección 9. Mapas
 - 9. 1 Introducción
 - 9. 2 Etiqueta <MAP>
 - 9. 3 Imágenes como mapas
 - 9. 4 Etiqueta <AREA>
 - 9. 5 Ejemplo
- Sección 10. Frames
 - 10. 1 Introducción a los frames
 - 10. 2 Etiqueta <FRAMESET>
 - 10. 3 Etiqueta <FRAME>
 - 10. 4 Frames anidados
 - 10. 5 Etiqueta <IFRAME>
 - 10. 6 Contenido Alternativo
- Sección 11. Scripts, Applets y demás I
 - 11. 1 Introducción
 - 11. 2 Cliente vs. Servidor
 - 11. 3 JavaScript
 - 11. 4 VBScript
 - 11. 5 Java
 - 11. 6 PHP
 - 11. 7 ASP
 - 11. 8 CGI
 - 11. 9 Mapas del lado del servidor
 - 11. 11 Etiqueta <SCRIPT>
 - 11. 12 Eventos
 - 11. 13 Etiqueta <NOSCRIPT>
- Sección 12. Introducción a las hojas de estilo
 - 12. 1 ¿Que son las hojas de estilo?
 - 12. 2 CSS y XSL
 - 12. 3 Reglas
- Sección 13. Agregando Estilos I
 - 13. 1 Agregando en HTML
 - 13. 2 Links a hojas de estilo externas
 - 13. 3 Importar hojas de estilo
 - 13. 4 Estilos de línea en HTML
 - 13. 5 Agrupamiento
 - 13. 6 Clases
 - 13. 7 Selectores contextuales
 - 13. 8 Comentarios

- 13. 9 Ejemplo
- Sección 14. Agregando Estilos II
 - 14. 1 Árbol de documento
 - 14. 2 Herencia
 - 14. 3 Cascada
 - 14. 4 Reglas de orden de cascada
 - 14. 5 Selectores hijos
 - 14. 6 Selectores por atributos
 - 14. 7 Selectores por ID
 - 14. 8 Seudo clases
 - 14. 9 Medidas
 - 14. 10 Colores
 - 14. 11 Problemas y navegadores
- Sección 15. Propiedades de las fuentes
 - 15. 1 font-family
 - 15. 2 font-size
 - 15. 3 font-style
 - 15. 4 font-variant
 - 15. 5 font-weight
 - 15. 6 font
- Sección 16. Propiedades de texto
 - 16. 1 text-align
 - 16. 2 text-decoration
 - 16. 3 text-indent
 - 16. 4 letter-spacing
 - 16. 5 text-transform
 - 16. 6 word-spacing
- Sección 17. La caja
 - 17. 1 La caja (box)
 - 17. 2 Dimensiones de la caja
 - 17. 3 Propiedades del colchón
 - 17. 4 Propiedades del borde
 - 17. 5 Propiedades de los márgenes
 - 17. 6 Ejemplos
- Sección 18. Integración de interfaz
 - 18. 1 Introducción
 - 18. 2 Cambio de estilos
 - 18. 3 Colores
 - 18. 4 Fuentes
 - 18. 5 Cursores
 - 18. 6 Ejemplos
- Sección 19. Propiedades de color y fondo
 - 19. 1 color
 - 19. 2 background-color
 - 19. 3 background-image
 - 19. 4 background-attachment
 - 19. 5 background-position
 - 19. 6 background-repeat
 - 19. 7 background

Licencia de Documentación Libre GNU

This is an unofficial translation of the GNU Free Documentation License into spanish. It was not published by the Free Software Foundation, and does not legally state the distribution terms for software that uses the GNU FDL--only the original English text of the GNU FDL does that. However, we hope that this translation will help spanish speakers understand the GNU FDL better.

Esta es una traducción no oficial de la GNU Free Document License (GFDL), versión 1.1 (de Marzo de 2.000), que cubre manuales y documentación. Esta traducción no tiene ningún valor legal, ni ha sido comprobada de acuerdo a la legislación de ningún país en particular. Se incluye sólo con propósitos educativos. Para efectos legales por favor remítase al original en inglés.

Esta traducción fue realizada en Mayo de 2000 por Igor Támara ikks@bigfoot.com y Pablo Reyes reyes_pablo@hotmail.com. Fue revisada en Enero de 2001 por Vladimir Támara vtamara@gnu.org y actualizada por Luis Miguel Arteaga Mejía.

Version 1.1, Marzo 2000

Copyright (C) 2000 Free Software Foundation, Inc.

59 Temple Place, Suite 330, Boston, MA 02111-1307, USA

Se permite la copia y distribución de copias literales de este documento de licencia, pero no se permiten cambios.

0. PREÁMBULO

El propósito de esta licencia es hacer que un manual, libro de texto, u otro documento escrito sea libre en el sentido de libertad: para asegurar a todo el mundo la libertad efectiva de copiarlo y redistribuirlo, con o sin modificaciones, bien de manera comercial o no comercial. En segundo término, esta licencia preserva para el autor o para quien publica una manera de obtener reconocimiento por su trabajo, al tiempo que no es considerado responsable de las modificaciones realizadas por terceros. Esta licencia es una especie de "copyleft" que significa que los trabajos derivados del documento deben a su vez ser libres en el mismo sentido. Esta licencia complementa la Licencia Pública General GNU, que es una licencia de copyleft diseñada para el software libre. Hemos diseñado esta Licencia para usarla en manuales de software libre, ya que el software libre necesita documentación libre: un programa libre debe venir con los manuales que ofrezcan la mismas libertades que da el software. Pero esta licencia no se limita a manuales de software; puede ser usada para cualquier trabajo textual, sin tener en cuenta su temática o si se publica como libro impreso. Recomendamos esta licencia principalmente para trabajos cuyo propósito sea instructivo o de referencia.

1. APLICABILIDAD Y DEFINICIONES

Esta Licencia se aplica a cualquier manual u otro trabajo que contenga una nota del propietario de los derechos de reproducción que indique que puede ser distribuido bajo los términos de esta Licencia. El "Documento", en adelante, se refiere a cualquiera de dichos manuales o trabajos. Cualquier miembro del público es un licenciatario, y será denominado como "Usted". Una "Versión Modificada" del Documento designa cualquier trabajo que contenga el Documento o una porción del mismo, ya sea una copia literal o con modificaciones y/o traducciones a otro idioma. Una "Sección Secundaria" es un apéndice titulado o una sección preliminar al prólogo del Documento que tiene que ver exclusivamente con la relación de quien publica o los autores del Documento con el tema general del Documento (o asuntos relacionados) y cuyo contenido no entra directamente en tal tema general. (Por ejemplo, si el Documento es en parte un texto

de matemáticas, una Sección Secundaria puede no explicar matemáticas.) La relación puede ser un asunto de conexión histórica, o de posición legal, comercial, filosófica, ética o política con el tema o con materias relacionadas.

Las "Secciones Invariantes" son ciertas Secciones Secundarias cuyos títulos son denominados como Secciones Invariantes, en la nota que indica que el documento es liberado bajo esta Licencia.

Los "Textos de Cubierta" son ciertos pasajes cortos de texto que se listan, como Textos de Portada o Textos de Contra Portada, en la nota que indica que el documento es liberado bajo esta Licencia.

Una copia "Transparente" del Documento significa una copia para lectura en máquina, representada en un formato cuya especificación está disponible al público general, cuyos contenidos pueden ser vistos y editados directamente con editores de texto genéricos o (para imágenes compuestas por píxeles) con programas genéricos para gráficas o (para dibujos) algún editor de dibujos ampliamente disponible, y que sea adecuado para exportar a formateadores de texto o para traducción automática a una variedad de formatos adecuados para ingresar a formateadores de texto. Una copia hecha en un formato que de otra forma sería Transparente pero cuyo formato ha sido diseñado para impedir o dificultar subsecuentes modificaciones por parte de los lectores no es Transparente. Una copia que no es "Transparente" es llamada "Opaca".

Los ejemplos de formatos adecuados para copias Transparentes incluyen ASCII plano sin formato, formato de Texinfo, formato de LaTeX, `@acronym{SGML}` o `@acronym{XML}` que usen un DTD disponible ampliamente, y `@acronym{HTML}` simple que siga los estándares y esté diseñado para modificaciones humanas. Los formatos Opacos incluyen PostScript, `@acronym{PDF}`, formatos propietarios que pueden ser leídos y editados únicamente con procesadores de palabras propietarios, `@acronym{SGML}` o `@acronym{XML}` para los cuáles los `@acronym{DTD}` y/o herramientas de procesamiento no están disponibles generalmente, y el `@acronym{HTML}` generado en una máquina, producido por algún procesador de palabras solo con propósitos de presentación.

La "Portada" significa, para un libro impreso, la portada misma más las páginas siguientes necesarias para mantener, legiblemente, el material que esta Licencia requiere que aparezca en la portada. Para trabajos en formatos que no tienen Portada como tal, "Portada" significa el texto cerca a la aparición más prominente del título del trabajo, precediendo el comienzo del cuerpo del texto.

2. COPIA LITERAL

Usted puede copiar y distribuir el Documento en cualquier medio, sea en forma comercial o no comercial, siempre y cuando esta Licencia, las notas de derecho de autor, y la nota de licencia que indica que esta Licencia se aplica al Documento se reproduzcan en todas las copias, y que usted no adicione ninguna otra condición sobre las expuestas en esta Licencia. No puede usar medidas técnicas para obstruir o controlar la lectura o copia posterior de las copias que usted haga o distribuya. Sin embargo, usted puede aceptar compensación a cambio de las copias. Si distribuye un número suficientemente grande de copias también deberá seguir las condiciones de la sección 3. Usted también puede prestar copias, bajo las mismas condiciones establecidas anteriormente, y puede exhibir copias públicamente.

3. COPIADO EN CANTIDADES

Si publica copias impresas del Documento que sobrepasen las 100, y la nota de Licencia del Documento exige Textos de Cubierta, debe incluir las copias con cubiertas que lleven en forma clara y legible, todos esos textos de Cubierta: Textos de Portada en la portada, y Textos de Contra Portada en la contra portada. Ambas cubiertas deben identificarlo a usted clara y legiblemente como quien publica tales copias. La portada debe presentar el título completo con todas las palabras del título igualmente prominentes y visibles. Usted puede adicionar otro material en las cubiertas. Las copias con cambios limitados a las cubiertas, siempre que

preserven el título del Documento y satisfagan estas condiciones, puede considerarse como copia literal. Si los textos requeridos para la cubierta son muy voluminosos para que ajusten legiblemente, debe colocar los primeros listados (tantos como sea razonable colocar) en la cubierta real, y continuar con el resto en páginas adyacentes. Si publica o distribuye copias Opacas del Documento cuya cantidad exceda las 100, debe incluir una copia Transparente que pueda ser leída por una máquina con cada copia Opaca, o indicar en o con cada copia Opaca una dirección en una red de computadores publicamente accesible que contenga una copia completa y Transparente del Documento, libre de material adicional, a la cual el público general de la red tenga acceso para bajar anónimamente sin cargo, usando protocolos de redes públicos y estándares. Si usted hace uso de la última opción, deberá tomar medidas razonablemente prudentes, cuando comience la distribución de las copias Opacas en cantidad, para asegurar que esta copia Transparente permanecerá accesible en el sitio indicado por lo menos un año después de su última distribución al público de copias Opacas de esa edición (directamente o a través de sus agentes o distribuidores). Se solicita, aunque no es requisito, que contacte a los autores del Documento antes de redistribuir cualquier gran número de copias, para permitirle la oportunidad de que le provean una versión actualizada del Documento.

4. MODIFICACIONES

Usted puede copiar y distribuir una Versión Modificada del Documento bajo las condiciones de las secciones 2 y 3 anteriores, siempre que usted libere la Versión Modificada bajo esta misma Licencia, con la Versión Modificada asumiendo el rol del Documento, por lo tanto licenciando la distribución y modificación de la Versión Modificada a quienquiera que posea una copia de este. En adición, debe hacer lo siguiente en la Versión Modificada:

A. Uso en la Portada (y en las cubiertas, si hay alguna) de un título distinto al del Documento, y de versiones anteriores (que deberían, si hay alguna, estar listados en la sección de Historia del Documento). Puede usar el mismo título que versiones anteriores del original siempre que quién publicó la primera versión lo permita.

B. Listar en la Portada, como autores, una o más personas o entidades responsables por la autoría o las modificaciones en la Versión Modificada, junto con por lo menos cinco de los autores principales del Documento (Todos sus autores principales, si hay menos de cinco).

C. Establecer en la Portada del nombre de quién publica la Versión Modificada, como quien publica.

D. Preservar todas las notas de derechos de reproducción del Documento.

E. Adyacente a las otras notas de derecho de reproducción, adicionar una nota de derecho de reproducción de acuerdo a sus modificaciones.

F. Incluir, inmediatamente después de la nota de derecho de reproducción, una nota de licencia dando el permiso público para usar la Versión Modificada bajo los términos de esta Licencia, de la forma mostrada más adelante en el Addendum.

G. Preservar en esa nota de licencia el listado completo de Secciones Invariantes y de los Textos de las Cubiertas que sean requeridos como se especifique en la nota de Licencia del Documento.

H. Incluir una copia sin modificación de esta Licencia.

I. Preservar la sección con título "Historia", y su título, y adicionar a esta una sección estableciendo al menos el título, el año, los nuevos autores, y quién publicó la Versión Modificada como reza en la Portada. Si no hay una sección titulada "Historia" en el Documento,

crear una estableciendo el título, el año, los autores y quien publicó el Documento como reza en la Portada, añadiendo además un artículo describiendo la Versión Modificada como se estableció en la oración anterior.

J. Preservar la localización en red, si hay, dada en el Documento para acceso público a una copia Transparente del Documento, así como las otras direcciones de red dadas en el Documento para versiones anteriores en las cuáles estuviese basado. Estas pueden ubicarse en la sección "Historia". Se puede omitir la ubicación en red para un trabajo publicado por lo menos 4 años antes que el Documento mismo, o si quien publicó originalmente la versión a la que se refiere da permiso.

K. En cualquier sección titulada "Agradecimientos" o "Dedicatorias", preservar el título de la sección, y preservar en la sección toda la sustancia y el tono de los agradecimientos y/o dedicatorias de cada contribuyente que estén incluidas.

L. Preservar todas las Secciones Invariantes del Documento, sin alterar su texto ni sus títulos. Números de sección o el equivalente no son considerados parte de los títulos de la sección.

M. Borrar cualquier sección titulada "Aprobaciones". Una tal sección no pueden estar incluida en las Versiones Modificadas.

N. No retitular ninguna sección existente como "Aprobaciones" o conflictuar con título de alguna Sección Invariante.

Si la Versión Modificada incluye secciones o apéndices nuevos o preliminares al prólogo que califican como Secciones Secundarias y contienen material no copiado del Documento, puede opcionalmente designar algunas o todas esas secciones como invariantes. Para hacerlo, adicione sus títulos a la lista de Secciones Invariantes en la nota de licencia de la Versión Modificada. Tales títulos deben ser distintos de cualquier otro título de sección. Puede adicionar una sección titulada "Aprobaciones", siempre que contenga únicamente aprobaciones de su Versión Modificada por varias fuentes--por ejemplo, observaciones de peritos o que el texto ha sido aprobado por una organización como un estándar. Puede adicionar un pasaje de hasta cinco palabras como un Texto de Portada, y un pasaje de hasta 25 palabras como un texto de Contra Portada, al final de la lista de Textos de Cubierta en la Versión Modificada. Solamente un pasaje de Texto de Portada y un Texto de Contra Portada puede ser adicionado por (o a manera de arreglos hechos por) cualquier entidad. Si el Documento ya incluye un texto de cubierta para la misma cubierta, previamente adicionado por usted o por arreglo hecho por la misma entidad, a nombre de la cual usted actúa, no puede adicionar otra; pero puede reemplazar el anterior, con permiso explícito de quien previamente publicó y agregó tal texto.

El(los) autor(es) y quien(es) publica(n) el Documento no dan con esta Licencia permiso para usar sus nombres para publicidad o para asegurar o implicar aprobación de cualquier Versión Modificada.

5. COMBINANDO DOCUMENTOS

Puede combinar el Documento con otros documentos liberados bajo esta Licencia, bajo los términos definidos en la sección 4 anterior para versiones modificadas, siempre que incluya en la combinación todas las Secciones Invariantes de todos los documentos originales, sin modificar, y las liste como Secciones Invariantes de su trabajo combinado en la respectiva nota de licencia. El trabajo combinado necesita contener solamente una copia de esta Licencia, y múltiples Secciones Invariantes idénticas pueden ser reemplazadas por una sola copia. Si hay múltiples Secciones Invariantes con el mismo nombre pero con contenidos diferentes, haga el título de cada una de estas secciones único adicionándole al final de este, entre paréntesis, el nombre del autor o de quien publicó originalmente esa sección, si es conocido, o si no, un número único. Haga el mismo ajuste a los títulos de sección en la lista de Secciones

Invariantes en la nota de licencia del trabajo combinado. En la combinación, debe combinar cualquier sección titulada "Historia" de los varios documentos originales, formando una sección titulada "Historia"; de la misma forma combine cualquier sección titulada "Agradecimientos", y cualquier sección titulada "Dedicatorias". Debe borrar todas las secciones tituladas "Aprobaciones."

6. COLECCIONES DE DOCUMENTOS

Usted puede hacer una colección que consista del Documento y otros documentos liberados bajo esta Licencia, y reemplazar las copias individuales de esta Licencia en los varios documentos con una sola copia que esté incluida en la colección, siempre que siga las reglas de esta Licencia para una copia literal de cada uno de los documentos en cualquiera de todos los aspectos. Usted puede extraer un solo documento de una de tales colecciones, y distribuirlo individualmente bajo esta Licencia, siempre que inserte una copia de esta Licencia en el documento extraído, y siga esta Licencia en todos los otros aspectos concernientes a la copia literal de tal documento.

7. AGREGACIÓN CON TRABAJOS INDEPENDIENTES

Una recopilación del Documento o de sus derivados con otros documentos o trabajos separados e independientes, en cualquier tipo de distribución o medio de almacenamiento, no cuenta como un todo como una Versión Modificada del Documento, siempre que no se clamen derechos de reproducción por la compilación. Tal recopilación es llamada un "agregado", y esta Licencia no aplica a los otros trabajos auto-contenidos y por lo tanto compilados con el Documento, o a cuenta de haber sido compilados, si no son ellos mismos trabajos derivados del Documento. Si el requerimiento de la sección 3 del Texto de la Cubierta es aplicable a estas copias del Documento, entonces si el Documento es menor que un cuarto del agregado entero, Los Textos de la Cubierta del Documento pueden ser colocados en cubiertas que enmarquen solamente el Documento entre el agregado. De otra forma deben aparecer en cubiertas enmarcando todo el agregado.

8. TRADUCCIÓN

La traducción es considerada como una clase de modificación, así que puede distribuir traducciones del Documento bajo los términos de la sección 4. Reemplazar las Secciones Invariantes con traducciones requiere permiso especial de los propietarios de los derechos de reproducción, pero usted puede incluir traducciones de algunas o todas las Secciones Invariantes además de las versiones originales de las Secciones Invariantes. Puede incluir una traducción de esta Licencia siempre que incluya también la versión original en inglés de esta Licencia. En caso de un desacuerdo entre la traducción y la versión original en Inglés de esta Licencia, la versión original en Inglés prevalecerá.

9. TERMINACIÓN

Usted no puede copiar, modificar, sublicenciar, o distribuir el Documento excepto como lo permite expresamente esta Licencia. Cualquier otro intento de copia, modificación, sublicenciamiento o distribución del Documento es nulo, y terminarán automáticamente sus derechos bajo esta Licencia. Sin embargo, los terceros que hayan recibido copias, o derechos, de su parte bajo esta Licencia no tendrán por terminadas sus licencias siempre que tales terceros permenezcan en total conformidad.

10. REVISIONES FUTURAS DE ESTA LICENCIA

La Free Software Foundation puede publicar nuevas y revisadas versiones de la GNU Free Documentation License de tiempo en tiempo. Tales versiones nuevas serán similares en espíritu a la presente versión, pero pueden diferir en detalles para solucionar problemas o intereses. Vea <http://www.gnu.org/copyleft/>. Cada versión de la Licencia tiene un número de versión que la distingue. Si el Documento especifica que una versión numerada particularmente de esta licencia o "cualquier versión posterior" se aplica a este, tiene la opción de seguir los términos y condiciones de esa versión especificada o de cualquiera versión posterior que hubiera sido publicada (no como un borrador) por la Free Software Foundation. Si el Documento no especifica un número de versión de esta Licencia, puede escoger cualquier versión que haya sido publicada (no como un borrador) por la Free Software Foundation.

COMO USAR ESTA LICENCIA PARA SUS DOCUMENTOS

Para usar esta licencia en un documento que usted haya escrito, incluya una copia de la Licencia en el documento y ponga el siguiente derecho de reproducción y notas de licencia justo después del título de la página:

Copyright (C) año su nombre.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being list their titles, with the Front-Cover Texts being list, and with the Back-Cover Texts being list. A copy of the license is included in the section entitled "GNU Free Documentation License".

Si no tiene Secciones Invariantes, escriba "with no Invariant Sections" en vez de decir cuales son invariantes. Si no tiene Textos de Portada, escriba "no Front-Cover Texts" en vez de "Front-Cover Texts being list"; y de la misma forma para los Textos de Contra Portada.

Si su documento contiene ejemplos de código de programa no triviales, le recomendamos liberar estos ejemplos en paralelo bajo una licencia de software libre de su elección, tal como la GNU General Public License, para permitir su uso en software libre.

Sección 1. Los Principios de HTML

Contenido:

- [1. 1 ¿Qué es HTML?](#)
- [1. 2 Herramientas](#)
- [1. 3 Archivos HTML](#)
- [1. 4 Sintaxis de las etiquetas](#)
- [1. 5 Comenzando](#)
- [1. 6 Etiqueta <HTML>](#)
- [1. 7 Etiquetas <HEAD> y <TITLE>](#)
- [1. 8 Etiqueta <BODY>](#)
- [1. 9 Colores](#)
- [1. 10 Etiqueta <P>](#)
- [1. 11 Comentarios](#)
- [1. 12 Caracteres especiales](#)

1. 1 ¿Qué es HTML?

HTML o Hypertext Markup Language. La traducción que le doy al español es "Lenguaje Etiquetado de Hipertexto". Probablemente no es la traducción exacta pero es muy significativa. Vayamos desentrañando cada uno de estos conceptos.

Hipertexto (HyperText)

Navegar en Internet es como navegar en el océano, si no llevamos una buena brújula o si no sabemos el lugar al que queremos ir, lo mas seguro es que nos perderemos en este océano digital.

Al seguir un vinculo, lo hacemos para aumentar una información, ver una imagen o reproducir un sonido o video. Esta capacidad de ir uniendo páginas con otras páginas es lo que le da a Internet su dinamismo.

Como todos sabemos la información es la esencia de Internet, o mejor dicho, la transferencia de esa información. En si, el concepto del Hipertexto fue uno de los mas importantes detonantes de lo que ahora llamamos World Wide Web que podría ser definido llanamente como un conjunto cuasi infinito de vínculos.

Una definición mas general que podríamos dar de Hipertexto es la siguiente:

"Es la organización de las unidades de información interconectadas entre si".

Supongamos por ejemplo una página o sitio, con varios vínculos incrustados en ella. La lectura de dicha página se puede hacer de manera lineal, olvidando que los vínculos existen, o podemos llevar una lectura no lineal siguiendo cada uno de los vínculos. Así nuestra definición podría ser de la siguiente forma:

"Es la organización no lineal de las unidades de información interconectadas entre si"

Si quisiéramos ser puristas, podríamos decir que el hipertexto solo se refiere a texto y al hablar de vínculos a objetos multimedia podríamos hablar de hipermedia. En fin, que cada quien escoja la suya.

Una nota final, el termino hipertexto fue usado por primera vez por Ted Nelson, inventor de Xanadu, un antecesor del Web.

Etiquetado (Markup)

Imaginemos esta escena: estamos sentados frente a nuestra computadora escribiendo un documento en nuestro procesador de texto favorito. Conforme vamos escribiendo vamos decidiendo el tamaño de las letras de los títulos y del texto normal, también definimos colores, tipos de letras o creamos tablas o incluso insertamos un dibujo a grafica.

Ahora imaginemos esta escena: Vamos en el transporte colectivo tratando de escribir ese mismo documento y nuestro procesador de texto es papel y lápiz. Así que, si queremos indicar la forma en que se vera el documento, tendremos que anotar las características de ese titulo o de esa imagen que debe ir aquí y centrada. La mejor forma seria inventar una forma de escribir esas características... a ver que les parece esta

<P> para indicar un párrafo nuevo

 para indicar negritas

<H1> para indicar un titulo

 para indicar que aquí va una imagen

Estas, en si, son etiquetas utilizadas en HTML para indicar en donde van las cosas y cuales son sus características.

Así que las etiquetas son solo cadenas de caracteres especiales que se insertan en el documento para indicarle a nuestro navegador como se vera o imprimirá una página en particular.

Lenguaje (Language)

En si HTML no es un lenguaje de programación; es solo un Lenguaje Etiquetado de Hipertexto.

Sin embargo se pueden encontrar algunas analogías como por ejemplo decir que las etiquetas son palabras reservadas, etc. Es mejor tomar a HTML como un formato que como un lenguaje de programación... evitaremos pasar algunas vergüenzas :-).

Por ultimo, cabe mencionar que HTML no es el único lenguaje, o formato, que maneja etiquetas, por ejemplo, también existe el XML.

1. 2 Herramientas

Después de una larga introducción (que probablemente muchos de ustedes ni leyeron) entramos de lleno a esto del HTML. Antes de comenzar tenemos que saber cuales son las herramientas que necesitaremos para aprender HTML.

En si lo único que necesitamos es un simple editor de textos como el Notepad de Windows o Pico en Linux. Cualquier otro editor que encontremos por ahí también puede que sirva. Por otra parte cabe mencionar que usar un procesador de textos no es una buena idea (como por ejemplo el Word) pues este tiende a modificar el texto que vamos escribiendo. En pocas palabras necesitamos un editor que guarde en formato ASCII.

¿Para que aprendo HTML si ya hago páginas muy bonitas con el Dreamweaver o con el Front Page?

No es que critiquemos a los programas WYSIWYG (What You See Is What You Get), de hecho este tutorial esta hecho en uno de estos. Pero hay muchísimas etiquetas que no son manejadas por estos programas y por lo tanto perdemos una parte importante de características adicionales. Por otra parte muchas veces es casi imposible obtener el resultado deseado y es entonces necesario meterse a descifrar el código.

Además del editor de textos necesitaremos un navegador, de preferencia mas, pero con uno es suficiente para empezar. Muchas veces es necesario probar la página HTML en navegadores distintos para ver si en los dos se ve igual.

Para comenzar a editar un sitio entero podremos utilizar una cantidad indefinida de programas de todo tipo, estos son algunos de los que yo he utilizado en estos años pero recuerden, no son necesarios para aprender HTML:

Edición de Imágenes: Para crear imágenes.

Corel Draw
Paint Shop Pro
Fireworks
The Gimp

Edición de Textos: Para la ortografía :-)

Corel Word Perfect
Star Office
Microsoft Office

1. 3 Archivos HTML

Un archivo HTML puede ser leído, como ya lo dijimos, en un editor de texto como el Notepad.

Cuando creamos un archivo HTML debemos de guardar el archivo con las extensiones .htm o .html. No creo que tenga que explicar como hacer esto.

1. 4 Sintaxis de la etiquetas

La forma de escribir una etiqueta básica es <ETIQUETA>; además la gran mayoría de estas se deben "cerrar" de esta manera: </ETIQUETA>. En este tutorial se muestran las etiquetas de cierre para todas aquellas etiquetas que lo requieren. Si en algún tema no se habla de una etiqueta de cierre, entonces damos por entendido que no es necesaria.

Los atributos de una etiqueta definen como va a afectar a lo que se encuentre dentro de ella. Existen varios atributos y estos se incluyen de la siguiente manera dentro de una etiqueta: <ETIQUETA ATRIBUTO>. Muchas etiquetas tienen varios atributos que pueden ser utilizados en la etiqueta de la siguiente manera: <ETIQUETA ATRIBUTO1 ATRIBUTO2>. En este tutorial se muestran

varios atributos para las diferentes etiquetas. Si en algún tema no se habla de un atributo, entonces damos por entendido que no es necesario especificar atributo alguno.

Para cerrar una etiqueta con atributos se utiliza solo la etiqueta de cierre, esto es sin atributos.

El valor de un atributo dentro de una etiqueta se expresara de la siguiente manera <ETIQUETA ATRIBUTO="VALOR">. Muchos atributos no requieren de un valor específico. Si en algún tema se muestra, en un ejemplo, solo el nombre del atributo y no un valor, entonces damos por entendido que no es necesario.

Por último, las etiquetas pueden estar escritas en minúsculas o mayúsculas esto es, la etiqueta <ETIQUETA> es igual a la etiqueta <etiqueta>.

1.5 Comenzando

Este es un pequeño ejemplo que nos servirá para entender los temas siguientes. No duden en copiar y pegar en el notepad, salvarlo y correrlo en su navegador favorito (Esto se aplica para todos los ejemplos del tutorial). Por cierto el texto fue tomado del libro "Historia del tiempo" de Stephen W. Hawking:

```
<HTML>
<HEAD>
<TITLE>Fragmento de &quot;Historia del tiempo&quot;
de Stephen W. Hawking</TITLE>
</HEAD>

<!--Mi primer documento HTML-->
<BODY BGCOLOR="#FFFFFF">
<P>La teoría de la relatividad general de Einstein,
por sí sola, predijo que el espacio-tiempo comenzaría
en la singularidad del big bang y que iría hacia un
final, bien en la singularidad del big crunch [“gran
crujido”, “implosión”] (si el
universo entero se colapsase de nuevo) o bien en una
singularidad dentro de un agujero negro (si una
región local, como una estrella, fuese a colapsarse).
Cualquier materia que cayese en el agujero negro
sería destruida en la singularidad, y solamente el
efecto gravitatorio de su masa continuaría
sintiendo afuera. Por otra parte, teniendo en
cuenta los efectos cuánticos parece que la masa o
energía de la materia tendrá que ser
devuelta finalmente al resto del universo, y que el agujero
negro, junto con cualquier singularidad dentro de
él, se evaporará y por último
desaparecerá. ¿Podrá la
mecánica cuántica tener un efecto
igualmente espectacular sobre las singularidades del big
bang y del big crunch? ¿Ocurre
realmente durante las etapas muy tempranas o muy
```

```
tard&iacute;as del universo, cuando los campos
gravitatorios son tan fuertes que los efectos
cu&aacute;nticos no pueden ser ignorados? &iquest;Tiene
de hecho el universo un principio y un final? Y si es
as&iacute;, &iquest;c&oacute;mo son?</P>
</BODY>
</HTML>
```

¿Interesantes preguntas no creen?

1. 6 Etiqueta <HTML>

La etiqueta <HTML> define un documento de HTML. Dentro de esta etiqueta se encuentran las etiquetas <HEAD> y <BODY> con las respectivas etiquetas que se encuentran en ellas.

Etiqueta de cierre: </HTML>

Sintaxis:

```
<HTML>
Otras etiquetas
</HTML>
```

1. 7 Etiquetas <HEAD> y <TITLE>

En la etiqueta <HEAD> se encuentra toda aquella información no visible sobre la página; cabe aclarar que el título de la hoja se encuentra dentro esta etiqueta y es visible en la barra de título del navegador.

Etiqueta de cierre: </HEAD>

Sintaxis:

```
<HTML>
<HEAD>
<TITLE>Titulo de mi página </TITLE>
Información no visible y título de la página
</HEAD>
</HTML>
```

El título de la página se especifica con la etiqueta <TITLE>

Etiqueta de cierre: </TITLE>

1. 8 Etiqueta <BODY>

En la etiqueta <BODY> se encuentra toda aquella información visible en la pantalla del navegador. Se encuentran dentro de esta etiqueta las imágenes, texto, sonidos, vínculos etc.

Etiqueta de cierre: </BODY>

Sintaxis:

```
<HTML>  
<HEAD>  
Información no visible y título de la página  
</HEAD>  
<BODY>  
Información visible en pantalla  
</BODY>  
</HTML>
```

Atributos:

BACKGROUND: Ver la sección 4.5.

BGCOLOR: Muestra un color de fondo para la página. La sintaxis es de la siguiente manera:
<BODY BGCOLOR="#RRVVAA">

BGPROPERTIES: Ver la sección 4.5.

TEXT: Especifica el color del texto en todo el documento. Su sintaxis es:
<BODY TEXT="RRVVAA">
Ver sección 1.9.

LINK: Especifica el color de los vínculos no visitados en todo el documento. Su sintaxis es:
<BODY LINK="RRVVAA">

VLINK: Especifica el color de los vínculos visitados en todo el documento. Su sintaxis es:
<BODY VLINK="RRVVAA">

ALINK: Especifica el color de los vínculos cuando se hace click sobre ellos. Su sintaxis es:
<BODY ALINK="RRVVAA">

Se especificó un valor de atributo de la forma RRVVAA, lo cual se explica en la siguiente sección.

1.9 Colores

En muchas etiquetas es necesario utilizar códigos hexagesimales para definir los colores a utilizar. Un color se define como la cantidad de rojo, verde y azul. La forma de escribir el código de un color dentro de una etiqueta es la siguiente:

#RRVVAA

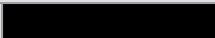
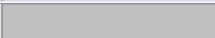
En donde RR representa al color rojo en código hexagesimal, VV representa al color verde en código hexagesimal y AA representa al color azul en código hexagesimal.

Por ejemplo, cierto color naranja tiene como código hexagesimal FF6600. FF representa la saturación de rojo que en decimales es 255 (máximo posible), 66 representa la saturación de verde que en decimales es 102 y 00 representa la saturación de azul que en decimales es 0.



Este es el color naranja del ejemplo.

Mostramos a continuación algunos de los colores mas utilizados junto con su código hexagesimal. De cualquier manera es útil tener una calculadora para hacer las conversiones de colores.

Color	Código	Ejemplo
Rojo	FF0000	
Azul	0000FF	
Verde	00FF00	
Blanco	FFFFFF	
Negro	000000	
Amarillo	FFFF00	
Gris	C0C0C0	
Violeta	4F2F4F	

1. 10 Etiqueta <P>

La etiqueta <P> crea un párrafo con el texto contenido dentro de ella.

Etiqueta de cierre: </P>, aunque puede omitirse.

Sintaxis:

<P>Párrafo</P>

Atributos:

ALIGN: Permite indicar la alineación del párrafo. Los posibles valores pueden ser LEFT (izquierda), CENTER (centro), RIGHT (derecha).

<P ALIGN="CENTER">Este es un ejemplo de texto en el centro</P>

1. 11 Comentarios

Muchas veces necesitaremos hacer comentarios no visibles en pantalla, por ejemplo, cuando queremos poner una indicación del lugar específico en que empieza un formulario o un script o simplemente poner el nombre del autor. Para estos comentarios existen 2 etiquetas.

Etiqueta <!--...-->

Permite hacer comentarios no visibles en pantalla. Es utilizada también por JavaScript para que los navegadores que no lo soportan ignoren el script.

Sintaxis:

<!--comentario no visible en pantalla-->

Etiqueta <COMMENT>

Utilizada por grupos de diseñadores para llevar un registro de las modificaciones hechas a la página.

Etiqueta de cierre: </COMMENT>

1. 12 Caracteres especiales

En HTML existen ciertos caracteres que no pueden ser escritos como tales en el documento HTML. Si deseamos utilizar alguno de estos caracteres en nuestra página debemos utilizar el código correspondiente. Presentamos los mas comunes:

Caracter	Se escribe	Caracter	Se escribe
Espacio en blanco	 	±	±
¡	¡	´	´
¶	¶	¨	¨
©	©	»	»
«	«	®	®
¿	¿	á	á
é	é	í	í
ó	ó	ú	ú
ü	ü	<	<
>	>	ñ	ñ
"	"	&	&

Fin de la Sección 1.

Sección 2. Aún mas principios

Contenido:

[2. 1 Encabezados](#)

[2. 2 Texto preformateado](#)

[2. 3 Alineación del texto](#)

[2. 4 Fuentes](#)

[2. 5 Dando énfasis](#)

[2. 6 Trucos molestos](#)

[2. 7 Líneas Horizontales](#)

[2. 8 Listas](#)

[2. 9 Saltos de línea](#)

2. 1 Encabezados

Los encabezados que podremos utilizar están definidos en 6 niveles de encabezamiento; La etiqueta <Hn> representa un encabezado de nivel n, donde n puede ser un valor entre 1 y 6, siendo 1 el nivel mas alto de encabezamiento y 6 el nivel mas bajo de encabezamiento.

<H1>Encabezamiento de nivel 1</H1>

Encabezamiento de nivel 1

<H2>Encabezamiento de nivel 2</H2>

Encabezamiento de nivel 2

<H3>Encabezamiento de nivel 3</H3>

Encabezamiento de nivel 3

<H4>Encabezamiento de nivel 4</H4>

Encabezamiento de nivel 4

<H5>Encabezamiento de nivel 5</H5>

Encabezamiento de nivel 5

<H6>Encabezamiento de nivel 6</H6>

Encabezamiento de nivel 6

Realmente sencillo, ¿no? Cabe hacer notar que los encabezados presentan al texto con una fuente predeterminada.

2. 2 Texto preformateado

El texto contenido dentro de la etiqueta <PRE> aparecerá tal como lo hace en el archivo fuente del código HTML. Esto resulta de utilidad pues podemos hacer que cualquier navegador respete los espacios y saltos de línea.

<PRE>Aristóteles reconoce, en su tratado "Las partes de los animales", que entre todos los animales, el hombre es el que posee el cerebro de mayor tamaño, en proporción al cuerpo y su función es "nada menos que la preservación del cuerpo entero". Sin embargo, Aristóteles pensaba que esta importante función era llevada a cabo por el cerebro sólo porque este órgano era capaz de compensar un exceso de calor del corazón, verdadero asiento del alma y de los sentidos, de tal manera que el cerebro funcionaría como contrapeso del corazón, moderando su actividad. La razón

de esto, dice Aristóteles, es que el cerebro es frío mientras que el corazón es caliente, por lo cual aquél es capaz de enfriar a éste y mantener un equilibrio que no se encuentre ni en un extremo de calor ni en un frío excesivo. Para que el cerebro pueda llevar a cabo esta importante función moderadora del corazón, la composición del cerebro es húmeda, y está formada por agua y tierra. [...]

Fragmento tomado de "Las células de la mente" de Ricardo Tapia.

2.3 Alineación del texto

Ya antes habíamos hablado de la etiqueta `<P>` y de su atributo `ALIGN` que permite alinear el texto de la página. Los valores para este atributo son `LEFT`, `RIGHT` y `CENTER`. Como podemos darnos cuenta, no es posible justificar el texto... bueno no será posible hasta que entremos a CSS.

2.4 Fuentes

La etiqueta `` permitirá asignar la fuente, color y tamaño del texto contenido en ella.

```
<P><FONT FACE="ARIAL", COURIER, HELVETICA" COLOR="#FF00CC" SIZE="3">Hola Mundo</FONT></P>
```

Hola Mundo

FACE

El atributo `FACE` permite establecer la fuente a utilizar.

Es posible utilizar hasta tres nombres de fuentes separadas por comas. La necesidad de hacer esto es debido a que, en el momento de cargar el documento en el navegador, este busca la primera fuente dentro del sistema; si no la encuentra continúa con la segunda, etc. Debemos de tener cuidado de solo utilizar fuentes "estándar", que puedan ser encontradas en cualquier sistema. Las más comunes son:

- Arial
- Courier
- Helvetica
- Sans-Serif
- Times New Roman
- Times
- Verdana

Y claro no debemos utilizar las siguientes

- Ricolino Machine
- Super Fancy Font
- Star Wars

Creo que el punto esta claro.

SIZE

Hablando sobre el tamaño, solo hace falta decir que los posibles valores que SIZE puede tomar son:

1, 2, 3, 4, 5, 6, 7: Tamaños fijos.

+1, +2, +3, +4, +5, +6, +7: Incrementos Relativos

-1, -2, -3, -4, -5, -6, -7: Decrementos Positivos

Cabe hacer notar que los tamaños pueden ser no interpretados como deseamos por el navegador.

Tamaño 1 Tamaño 2 Tamaño 3 Tamaño 4 Tamaño 5 Tamaño 6 Tamaño 7

Tamaño +1 Tamaño +2 Tamaño +3 Tamaño +4
Tamaño +5 Tamaño +6 Tamaño
+7

Tamaño -1 Tamaño -2 Tamaño -3 Tamaño -4 Tamaño -5 Tamaño -6 Tamaño -7

Recientemente se ha decidido que el uso de la etiqueta y su atributos pueden ser dañinos para la salud de nuestro documento, ya que es posible obtener un resultado mejor si utilizamos CSS. Sin embargo, la gente sigue resistiéndose.

Por ultimo creo que el atributo COLOR no debe ser un problema, pues ya sabemos lo que #RRVVAA significa.

2. 5 Dando énfasis

En esta sección veremos algunas etiquetas que pueden ser de utilidad para mejorar el aspecto.

Etiqueta <ADDRESS>

Esta etiqueta provoca que el texto contenido en ella figure como un rotulo en una carta convencional.

Etiqueta de cierre: </ADDRESS>

Etiqueta

Esta etiqueta provoca que el texto contenido en ella aparezca en negritas.

Etiqueta de cierre:

Etiqueta <DFN>

Esta etiqueta muestra el texto como una definición.

Etiqueta de cierre: </DFN>

Etiqueta

Esta etiqueta provoca un énfasis en el texto contenido en ella.

Etiqueta de cierre:

Etiqueta <I>

Esta etiqueta provoca que el texto contenido aparezca en cursivas.

Etiqueta de cierre: </I>

Etiqueta <KBD>

Esta etiqueta provoca que el texto contenido aparezca en una fuente en negritas con un ancho fijo.

Etiqueta de cierre: </KBD>

Etiqueta <S>

Esta etiqueta tacha el texto contenido en ella.

Etiqueta de cierre: </S>

Etiqueta <SUB>

Esta etiqueta provoca que el texto contenido en ella aparezca como subíndice.

Etiqueta de cierre: </SUB>

Etiqueta <SUP>

Esta etiqueta provoca que el texto contenido en ella aparezca como superíndice.

Etiqueta de cierre: </SUP>

Etiqueta <U>

Esta etiqueta provoca que el texto contenido en ella aparezca subrayado.

Etiqueta de cierre: </U>

2. 6 Trucos molestos

En esta sección aprenderemos como crear sentimientos de odio y repulsión en nuestro visitante... para que vean el alcance de HTML. Probablemente a algunos les parecerán realmente divertidas y no estarán de acuerdo conmigo pero, recuerden, todo con moderación.

Etiqueta <BLINK>

La etiqueta <BLINK> ocasiona un efecto de parpadeo en el texto comprendido en ella:

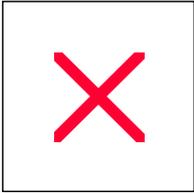
¿Realmente molesto no?

Nota: Solo Netscape lo incorpora. Si usamos un navegador diferente, veremos el texto estático.

Etiqueta de cierre: </BLINK>

Etiqueta <MARQUEE>

La etiqueta <MARQUEE> provoca que el texto dentro de ella se desplace:



Etiqueta de cierre: `</MARQUEE>`

A continuación se muestran los atributos:

BGCOLOR: Da un color de fondo a la marquesina:

`<MARQUEE BGCOLOR="#RRVVAA">`

DIRECTION: Especifica hacia donde será el desplazamiento:

`<MARQUEE DIRECTION="LEFT">` Hacia la izquierda.

`<MARQUEE DIRECTION="RIGHT">` Hacia la derecha.

ALIGN: Alinea el texto en la parte superior, media o inferior de la marquesina:

`<MARQUEE ALIGN="TOP">` Parte superior.

`<MARQUEE ALIGN="MIDDLE">` Parte media.

`<MARQUEE ALIGN="BOTTOM">` Parte inferior.

BEHAVOIR: Indica el tipo de movimiento que tendrá el texto dentro de la marquesina:

`<MARQUEE BEHAVOIR="SLIDE">` El texto se detendrá.

`<MARQUEE BEHAVOIR="ALTERNATE">` Alternará el movimiento de lado a lado.

`<MARQUEE BEHAVOIR="SCROLL">` El texto se desplazará rápidamente.

HEIGHT: Especifica la altura de la marquesina en píxeles o porcentaje:

`<MARQUEE HEIGHT="100">` En píxeles.

`<MARQUEE HEIGHT="100%">` En porcentaje.

WIDHT: Especifica el ancho de la marquesina en píxeles o porcentaje:

`<MARQUEE WIDHT="100">` En píxeles.

`<MARQUEE WIDHT="100%">` En porcentaje.

LOOP: Determina el número de veces que se desplazará el texto dentro de una marquesina:

`<MARQUEE LOOP="10">` El texto se desplazará 10 veces.

`<MARQUEE LOOP="INFINITE">` El texto se desplazará indefinidamente.

`<MARQUEE LOOP="-1">` El texto se desplazará indefinidamente.

SCROLLDELAY: Tiempo del desplazamiento en milisegundos:

`<MARQUEE SCROLLDELAY="2">` El tiempo es de 2 milisegundos.

SCROLLAMOUNT: Determina el espacio, en píxeles, entre el texto desplazado.

`<MARQUEE SCROLLAMOUNT="20">`

2. 7 Líneas Horizontales

Las líneas horizontales son de gran ayuda para separar secciones de texto o incluso como decoración.

Etiqueta <HR>

La etiqueta <HR> permite dibujar una línea horizontal en el documento de la siguiente manera:



Atributos:

NOSHADE: Elimina el efecto tridimensional a la línea.

<HR NOSHADE>



WIDTH: Permite especificar el ancho de la línea en píxeles o porcentaje:

<HR WIDTH="200"> ó <HR WIDTH="50%">



HEIGHT: Permite especificar el alto de la línea en píxeles.

<HR HEIGHT="10">



ALIGN: Ya vimos este atributo.



En estos atributos se usan valores de tamaño de la línea en píxeles y porcentaje. El porcentaje se refiere al tamaño de la línea respecto a la ventana que contiene al navegador. Personalmente recomiendo el uso de porcentajes.

2. 8 Listas

Listas básicas

Para definir una lista básica utilizaremos dos etiquetas. La etiqueta y la etiqueta .

Etiqueta

Esta etiqueta indicara al navegador de que debe crear una lista básica.

Etiqueta de cierre:

Etiqueta

Dentro de la etiqueta debemos de utilizar la etiqueta para definir los elementos de una lista.

Etiqueta de cierre:

Elemento 1 de la lista

Elemento 2 de la lista

Elemento 3 de la lista

- Elemento 1 de la lista
- Elemento 2 de la lista
- Elemento 3 de la lista

Tipos de viñetas en una lista básica

Para cambiar la forma de las viñetas en una lista básica debemos utilizar el atributo TYPE de la etiqueta . Usando el atributo en esta etiqueta ocasionará que todos los elementos en ella presenten la viñeta que definamos.

<UL TYPE="CIRCLE"> muestra una viñeta con forma de círculo.

<UL TYPE="DISC"> muestra una viñeta con forma de disco.

<UL TYPE="SQUARE"> muestra una viñeta con forma de cuadrado.

- Viñeta circular
- Viñeta de disco
- Viñeta cuadrada

Nota: Algunos navegadores no muestran las viñetas como deberían

Listas numeradas

Para definir una lista numerada utilizaremos dos etiquetas. La etiqueta y la etiqueta .

Etiqueta

Esta etiqueta indicará al navegador de que debe crear una lista numerada.

Etiqueta de cierre: ``

El Atributo START de la etiqueta `` es utilizado para indicar el numero en el que debe de empezar la lista.

Etiqueta ``

Dentro de la etiqueta `` debemos utilizar la etiqueta `` para definir los elementos de una lista.

Etiqueta de cierre: ``

``

`Elemento 1 de la lista`

`Elemento 2 de la lista`

`Elemento 3 de la lista`

``

1. Elemento 1 de la lista
2. Elemento 2 de la lista
3. Elemento 3 de la lista

Tipos de numeración en una lista numerada

Para cambiar la forma en que aparecerá la numeración en una lista numerada debemos utilizar el atributo TYPE de la etiqueta ``.

`<OL TYPE="1">` Para números enteros (1,2,3,4,...).

`<OL TYPE="A">` Para letras mayúsculas.

`<OL TYPE="a">` Para letras minúsculas.

`<OL TYPE="I">` Para números romanos en mayúsculas.

`<OL TYPE="i">` Para números romanos en minúsculas.

Listas de definición

Para crear una lista de definición utilizaremos tres etiquetas. La etiqueta `<DL>`, la etiqueta `<DT>` y la etiqueta `<DD>`

Etiqueta `<DL>`

Esta etiqueta indicará al navegador de que debe crear una lista de definición.

Etiqueta de cierre: </DL>

Etiqueta <DT>

Dentro de la etiqueta <DL> debemos de utilizar la etiqueta <DT> para indicar el termino a definir.

Etiqueta de cierre: </DT>

Etiqueta <DD>

Dentro de la etiqueta <DL> debemos de utilizar la etiqueta <DD> para indicar la definición correspondiente del termino a definir.

Etiqueta de cierre: </DD>

<DL>

<DT>Primer termino a definir</DT>

<DD>Primera definición</DD>

<DT>Segundo termino a definir</DT>

<DD>Segunda definición</DD>

<DL>

Primer elemento a definir

Primera definición

Segundo elemento a definir

Segunda definición

Listas anidadas

Anidar una lista es incluir otra lista dentro de otra. Cualquier lista puede ser anidada dentro de otra. A continuación se anida una lista básica en una lista numerada.

 Elemento 1 de la lista

Sub-elemento a

Sub-elemento b

Elemento 2 de la lista

Elemento 3 de la lista

1. Elemento 1 de la lista
 - o Sub-elemento a
 - o Sub-elemento b
2. Elemento 2 de la lista
3. Elemento 3 de la lista

2. 9 Saltos de línea

La etiqueta
 es utilizada para insertar un salto de línea, esto es, sin espacios entre dos líneas de texto.

Sintaxis:

Línea de texto

Línea de texto

Línea de texto

No existe etiqueta de cierre.

Mas adelante veremos los atributos relacionados a esta etiqueta

Fin de la sección 2.

Sección 3. Vinculando

[3. 1 Etiqueta <A>](#)

[3. 2 Vínculos dentro de una página](#)

[3. 3 Vínculos con otra página](#)

[3. 4 Vínculos con otros sitios](#)

[3. 5 Atributo TARGET](#)

[3. 6 Correo electrónico](#)

[3. 7 Grupos de noticias](#)

[3. 8 Sitios FTP](#)

[3. 9 Archivos](#)

[3. 10 Ejemplos](#)

3. 1 Etiqueta <A>

La etiqueta <A> es la que da la propiedad de hipertexto a una página HTML. Los vínculos entre varias páginas están dados por esta etiqueta. Etiquetas como (que se vera en la siguiente sección) ó <P>, entre otras, pueden estar contenidas dentro de <A> para crear vínculos con imágenes o texto respectivamente. Las palabras vínculo e hipervínculo son sinónimos. También se utiliza la palabra "link" para nombrar los vínculos.

Etiqueta de cierre:

<P>Texto</P> hace del párrafo un hipervínculo.

 hace de la imagen un hipervínculo.

Observemos que el atributo HREF a sido utilizado en la sintaxis por considerarlo parte esencial de la etiqueta <A>. Cuando se hable de vínculos dentro de una página se observará que en ciertas circunstancias el atributo HREF será sustituido por el atributo NAME para crear vínculos dentro de una pagina.

HREF Se utiliza para especificar el tipo de hipervínculo (páginas, otros sitios, sitios FTP, grupos de noticias, etc) y la ubicación o dirección de dichos recursos.

3. 2 Vínculos dentro de una página

Muchas veces tendremos la necesidad de mostrar un documento muy extenso en nuestro sitio. Para ayudar al visitante a navegar por este documento podemos crear vínculos a partes del documento en el mismo.

Para crear este tipo de vínculos tendremos que definir el lugar al que queremos ir con el atributo NAME de la etiqueta <A> y el vínculo a ese lugar con el atributo HREF de la etiqueta <A>.

EXPLICACIÓN: Si nosotros queremos que en el final del documento aparezca un vínculo que nos lleve al principio, tendremos que dar un nombre al principio del documento (NAME) y después hacer un vínculo a el en la parte final (HREF).

Texto en el principio del documento da al texto que se encuentra en el principio del documento el nombre "principal".

Ir al principio hace del texto "Ir al principio" un vínculo que lleva al principio del documento que tiene la etiqueta anterior. El uso de esta técnica se puede observar consistentemente en las distintas secciones del tutorial en donde un índice al principio del documento lleva a las distintas subsecciones. El símbolo # es indispensable.

3. 3 Vínculos con otra página

Para crear un vínculo desde una página hacia otra dentro del mismo sitio, debemos utilizar el atributo HREF de la etiqueta A. A continuación se explicara la forma de trabajar con directorios y subdirectorios.

`` crea un vínculo hacia la página "INDEX.HTM" que se encuentra en el mismo directorio que la página que contiene el vínculo.

`` crea un vínculo hacia la página "INDEX.HTM" que se encuentra en un directorio anterior a la página que contiene el vínculo.

`` crea un vínculo hacia la página "INDEX.HTM" que se encuentra en el subdirectorio "DIRECTORIO".

3. 4 Vínculos con otros sitios

Para crear un vínculo desde una página hacia un sitio, debemos utilizar el atributo HREF de la etiqueta `<A>`. Este capítulo puede resultar útil si nos confundimos con la sintaxis de directorios y subdirectorios de nuestra página; Si es así tendríamos que poner toda la dirección de la página, comenzando desde `http://`

Sintaxis:

`` crea un vínculo hacia el sitio "cualquier sitio". Hay que hacer notar que las URL's se escriben en minúsculas.

3. 5 El atributo TARGET

El atributo TARGET junto con sus valores nos permitirán trabajar con los vínculos y las ventanas o frames (marcos) en los que se cargaran. Los valores predeterminados para el atributo TARGET son `_BLANK`, `_PARENT`, `_TOP` y `_SELF`. Aquí solo veremos `_BLANK` y `_SELF` dejando los otros dos valores por definir en los capítulos de frames.

`_SELF` es el valor predeterminado para el atributo TARGET. Sirve para que el vínculo se cargue en la misma ventana del navegador en que la cual esta la página que hace la referencia al vínculo. En si es redundante utilizar el atributo TARGET con el valor `_SELF` cuando estamos en una página sin frames.

Por otra parte, el valor `_BLANK` resulta muy útil cuando deseamos cargar el vínculo en una ventana nueva del navegador.

`` abre el sitio en una ventana nueva del navegador.

3. 6 Correo Electrónico

Si deseamos crear un vínculo de correo electrónico debemos dar al atributo HREF el valor MAILTO. Al dar click sobre este vínculo, se instruirá al navegador para que abra el programa de correo electrónico con la dirección utilizada en la casilla "TO". Si deseamos que en la casilla "SUBJECT" aparezca un título determinado por nosotros debemos utilizar el atributo TITLE de la etiqueta <A>.

 abre el programa de correo electrónico con la dirección en la casilla "TO".

 abre el programa de correo electrónico con la dirección en la casilla "TO" y con un título determinado en la casilla "SUBJECT".

Nota: La dirección de correo electrónico debe de ser escrita toda en minúsculas.

3. 7 Grupos de noticias

Los grupos de noticias son lugares en Internet en donde se pueden publicar mensajes sobre un tema particular así como leer otros dejados por otros usuarios. Los navegadores tienen la capacidad para ver ese tipo de lugares aunque se pueden utilizar programas especializados. Nosotros nos centraremos en la primera parte.

3. 8 Sitios FTP

FTP son las siglas para File Transfer Protocol (Protocolo de Transferencia de Archivos). Un sitio de FTP es aquel en donde los usuarios pueden bajar archivos o subirlos si es que cuentan con la clave de acceso. Hay muchos sitios de FTP que son públicos y permiten que cualquier usuario entre a ellos y baje un archivo de su interés. Los navegadores tienen la capacidad para ver ese tipo de directorios así que podemos utilizar un navegador, aunque es mejor utilizar un programa de FTP. Nosotros nos centraremos en la primera parte.

Cuando un usuario haga click en el vínculo a un sitio FTP, el navegador mostrara los archivos disponibles para el público como si estuviéramos viendo una carpeta de nuestra computadora.

3. 9 Vínculos con archivos

Hasta ahora hemos visto como vincular a archivos HTM o HTML, pero muchas veces necesitaremos que el visitante vea un archivo de otro tipo como TXT, WAV, MIDI, etc. o incluso que lo baje a su computadora. De hecho, cualquier archivo puede estar vinculado desde un documento. El visitante al hacer click sobre un archivo vinculado verá un cuadro de dialogo preguntando sobre que hacer con ese archivo: bajarlo o abrirlo. Este tipo de vínculos se hacen de la

misma manera que hemos visto en capítulos anteriores, ya que solo cambia el nombre y tipo de archivo.

3. 10 Ejemplos

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Página Personal</TITLE>
```

```
</HEAD>
```

```
<BODY BGCOLOR="#FFFFFF">
```

```
<H2 ALIGN="center">Esta es la página personal de un simple humano con gustos y aficiones.</H2>
```

```
<HR WIDTH="80%" SIZE="1" ALIGN="center" NOSHADE><!--una separacion-->
```

```
<H4>Esta es la sección de vínculos.</H4>
```

```
<P><A HREF=" ../index.html">Aquí; esta la página principal</A><BR>
```

```
<A HREF="indice.htm">El mejor tutorial de HTML jamás escrito</A>. :-)<BR>
```

```
<A HREF="http://www.linux.com" TARGET="_blank">Linux.com en un ventana nueva</A><BR>
```

```
<A HREF="#inicio">Este link lleva al principio de la sección 3</A><BR>
```

```
<A HREF="seccion-2.htm#inicio">Este link lleva al final de la sección 2</A><BR>
```

```
<A HREF="seccion-1.htm" target="_blank">Este link lleva a la sección 1, en una ventana nueva</A><BR>
```

```
<A HREF="ftp://ftp.xfree86.org/pub/XFree86">Descargar XFree86</A><BR>
```

```
<A HREF=" ../trieb.jpg">Este soy yo.</A></P>
```

```
<P><A HREF="mailto:trieb@netscape.net">Y claro si quieren quejarse manden un correo.</A></P>
```

```
<HR WIDTH="80%" SIZE="1" ALIGN="center" NOSHADE><!--una separacion-->
```

```
</BODY>
```

```
</HTML>
```

Resultado:

Esta es la página personal de un simple humano con gustos y aficiones.

Esta es la sección de vínculos.

[Aquí esta la pagina principal](#)

[El mejor tutorial de HTML jamás escrito . :-\)](#)

[Linux.com en un ventana nueva](#)

[Este link lleva al principio de la sección 3](#)

[Este link lleva al final de la sección 2](#)

[Este link lleva a la sección 1, en una ventana nueva](#)

[Descargar XFree86](#)

[Este soy yo.](#)

[Y claro si quieren quejarse manden un correo.](#)

Fin de la sección 3

Sección 4. Imágenes y píxeles

[4. 1 Entendiendo los píxeles](#)

[4. 2 GIF](#)

[4. 3 PNG](#)

[4. 4 JPG](#)

[4. 5 Mas sobre <BODY>](#)

[4. 6 Etiqueta](#)

[4. 7 Thumbnails](#)

[4.8 Ejemplos](#)

El navegar nos introduce a un mundo multimedia. Encontramos imágenes, sonidos, videos, y toda clase de alardes tecnológicos relacionados pero lo mas común son las imágenes. La posibilidad de incrustar una imagen dentro de un documento HTML nos permitirá hacer mas vistosa nuestra página. Desgraciadamente el no conocer ciertas características de las imágenes convertirán nuestra página en una pesadilla para el visitante.

Comencemos pues entendiendo de que están hechas las imágenes.

4.1 Entendiendo los píxeles

Como todos sabemos, una imagen es una representación visual de algo: personas, paisajes o cualquier otra cosa que queramos. Las imágenes que usaremos en nuestro sitio tendrán que haber sido almacenadas en forma electrónica, dentro de una unidad de almacenamiento (Disco Duro, Floppy, etc).

Las imágenes pueden ser categorizadas en: gráficos de trama (raster graphics) y gráficos vectoriales (vector graphics). Por ahora solo nos enfocaremos a los gráficos de trama.

Gráficos de Trama

Son imágenes digitales mejor conocidas como mapas de bits (bitmap). Dentro de un sistema de coordenadas xy que representan una región de visualización (por ejemplo el monitor), los diferentes puntos existentes pueden ser iluminados de diferentes colores. Se le llama mapa de bits pues la información que contiene el grafico es asociada inmediatamente con la malla (grid) formada por los ejes x y y.

Debido a la iluminación de estos puntos, al tratar de modificar este tipo de gráficos se tienen perdidas de información. Por ejemplo cuando tratamos de ampliar una imagen, estos puntos se expanden también, al reducir una imagen, estos puntos se amontonan entre si. Las imágenes siguientes son de utilidad para ver lo que sucede al aumentar y disminuir un grafico de trama.

Los puntos de los que hemos estado hablando son los famosos píxeles.

4. 2 GIF

GIF son las siglas para Graphic Interchange Format (Formato de Intercambio Gráfico). GIF es considerado un estándar en Internet.

GIF es un grafico de trama en 2D codificado de forma binaria y utilizando el algoritmo de compresión LZW. Existen dos versiones, GIF 87a y GIF 89a. La diferencia entre estas dos consiste en que GIF 89a permite crear una animación contenida dentro de un solo archivo y la creación de GIF's entrelazados.

Un GIF Animado es tan solo un conjunto de imágenes presentadas en un orden específico.

Debido a que el algoritmo LZW pertenece a UNISYS se creo el formato PNG y se considera que se convertirá en el nuevo estándar de gráficos en Internet. El hecho de que pertenezca a UNISYS no implica que los usuarios comunes y corrientes no puedan hacer uso de ellos, el problema reside en que para la creación de programas que trabajen con este tipo de archivos es necesaria una licencia.

Por ultimo, este formato de imágenes permite distribuir datos de imagen hasta 8 bits

4. 3 PNG

PNG son las siglas para Portable Network Graphics.

Como dijimos anteriormente nace como el sustituto perfecto para GIF con características realmente interesantes. En esta ocasión la información concerniente a PNG podrá ser encontrada en la página principal de PNG:

<http://www.libpng.org/pub/png/>

4. 4 JPG

Si hemos sido bastante observadores y curiosos nos habremos dado cuenta que este tipo de imágenes se ven mejor que las imágenes GIF. Las imágenes JPEG pueden tener una calidad cercana a los 24 bits por lo cual son excelentes si deseamos presentar mas de 256 colores.

Otra cualidad de las imágenes JPEG es que pueden ser comprimidas. La compresión es una técnica usada para reducir el tamaño de un archivo de datos. Sabiendo esto podemos decir que las imágenes JPEG son imágenes de gran calidad y menor tamaño.

JPEG son las siglas en inglés para Joint Picture Experts Group (Grupo Conjunto de Expertos Fotográficos). Estas imágenes son las mas utilizadas en Internet después de las imágenes GIF.

4. 5 Mas sobre <BODY>

Si deseamos que nuestra página tenga una imagen de fondo debemos utilizar el atributo BACKGROUND de la etiqueta <BODY>. Este atributo presentará la imagen seleccionada como un fondo. La imagen se repite hasta llenar el espacio de la página.

```
<BODY BACKGROUND="IMAGEN.JPG">
```

Al desplazar la página hacia arriba o abajo, el fondo también se moverá, para evitar que la imagen de fondo se desplace debemos utilizar el atributo BGPROPERTIES de la etiqueta <BODY>. Este atributo junto con BACKGROUND, presentará la imagen de fondo fija.

```
<BODY BACKGROUND="IMAGEN.JPG" BGPROPERTIES="FIXED">
```

<BODY BACKGROUND="IMAGEN.JPG">: presenta como fondo la imagen "IMAGEN.JPG" que está en el mismo directorio que la página que lleva el fondo.

<BODY BACKGROUND="../IMAGEN.JPG">: presenta como fondo la imagen "IMAGEN.JPG" que está en un directorio anterior al que se encuentra la página que lleva el fondo.

<BODY BACKGROUND="DIRECTORIO/IMAGEN.JPG">: presenta como fondo la imagen "IMAGEN.JPG" que se encuentra en el subdirectorio "directorio".

<BODY BACKGROUND="http://www.cualquiersitio.com/IMAGEN.JPG">: presenta como fondo la imagen "IMAGEN.JPG" que se encuentra en la URL <http://www.cualquiersitio.com>.

4. 6 Etiqueta

: Imagen a exhibir.

Para determinar la imagen a utilizar así como su tamaño, etc, tenemos que usar los siguientes atributos.

SRC

Especifica la ubicación de una imagen:

: muestra la imagen "IMAGEN.JPG" que se encuentra en el mismo directorio que la página que la contiene.

: muestra la imagen "IMAGEN.JPG" que se encuentra en un directorio anterior al de la página que la contiene.

: muestra la imagen "IMAGEN.JPG" que se encuentra en el subdirectorio "directorio" que a su vez se encuentra en el mismo directorio en donde se encuentra la página que contiene la imagen.

: muestra la imagen "IMAGEN.JPG" que se encuentra en la URL <http://www.cualquiersitio.com>

ALT

Para mostrar un texto alternativo en el marco de las imágenes debemos utilizar el atributo ALT de la etiqueta . Este atributo es muy útil para aquellos buscadores que solo presentan texto. También sirve para hacer una descripción de las imágenes que no se han cargado. Para ver como funcionan coloca tu ratón sobre alguna de las imágenes de navegación al final de esta página.

ALIGN

La alineación de las imágenes se produce con el atributo ALIGN de la etiqueta . Esta alineación de las imágenes es con respecto al texto, es decir, podemos modificar la forma en que el texto aparece alrededor de las imágenes. Los valores posibles para este atributo son: TOP, MIDDLE, BOTTOM, LEFT y RIGHT.

: alinea el texto en la parte superior a un lado de la imagen. La siguiente línea de texto aparecerá abajo de la imagen.

: alinea el texto en la parte media a un lado de la imagen. La siguiente línea de texto aparecerá abajo de la imagen.

: alinea el texto en la parte inferior a un lado de la imagen. La siguiente línea de texto aparecerá abajo de la imagen.

: coloca la imagen en el margen izquierdo y provoca que el texto a un lado de ella la rodee sin pasar abajo de la imagen.

: coloca la imagen en el margen derecho y provoca que el texto a un lado de ella la rodee sin pasar abajo de la imagen.

HEIGHT y WIDTH

Una forma bastante "artificial" de dimensionar las imágenes es utilizando los atributos HEIGHT (altura) y WIDTH (ancho) de la etiqueta . Otro uso de estos atributos es especificar el tamaño real de las imágenes para que el navegador reserve el espacio que ocupará la imagen. Estos atributos están dados en píxeles.

BORDER

Muestra un borde alrededor de la imagen, en píxeles

: la imagen no muestra borde.

VSPACE y HSPACE

Para dejar espacios en blanco alrededor de una imagen debemos utilizar los atributos HSPACE y VSPACE de la etiqueta . Los valores de estos atributos están dados en píxeles.

: Deja un espacio de 30 píxeles a la derecha y a la izquierda de la imagen.

: Deja un espacio de 30 píxeles arriba y abajo de la imagen.

LOWSRC

El atributo LOWSRC de la etiqueta permite cargar una imagen de baja resolución antes de cargar la imagen definitiva con mayor resolución. Este atributo es muy útil cuando deseamos mostrar imágenes muy grandes.

USEMAP

Este atributo será estudiado en la sección de mapas.

Vínculos

En la sección anterior hablamos de vínculos de texto. Ahora veremos los vínculos con imágenes:

Para hacer de una imagen un hipervínculo debemos utilizar la etiqueta <A> y sus atributos antes de la etiqueta IMG y sus atributos.

: hace de la imagen "IMAGEN.JPG" un hipervínculo que lleva a la página index.htm

4. 7 Thumbnails

La finalidad de este capítulo es dar una pequeña recomendación para trabajar con imágenes grandes.

Muchas veces necesitaremos mostrar alguna imagen de tamaño considerable. Según aumente el tamaño de la imagen aumentará también el tamaño del archivo, por lo tanto el poner una imagen grande en una página puede causar que la velocidad para bajar una página disminuya.

Para el usuario resulta sumamente molesto el tener que esperar para ver una imagen sumamente grande o, ya cuando baje, tener que desplazarse horizontal o verticalmente para poder verla completa, por lo tanto debemos evitar en lo posible imágenes grandes. Si es inevitable, lo mejor que podemos hacer es crear un vínculo de texto a esa imagen aunque debemos ser muy explícitos para dar a entender de que se trata esa imagen.

Otra forma mas difundida es el uso de thumbnails. Un thumbnail es una pequeña imagen que sirve como vínculo para llevar a la misma imagen en su tamaño normal. La forma de crear thumbnails es reduciendo, en un programa para gráficos, la imagen de gran tamaño.

Utilizar thumbnails evitará que el rendimiento de una página disminuya y por ende también evitará que el visitante se moleste y huya de nuestra página pues el decidirá si desea ver esa imagen o no.

Un comentario final: Por ningún motivo se deben utilizar los atributos HEIGHT y WIDTH de la etiqueta para crear thumbnails.

4. 8 Ejemplos

Estos son los vínculos para las imágenes utilizadas en este ejemplo:

[paper002.jpg](#)

[undercons.gif](#)

[durazno.gif](#)

[limon.gif](#)

[tomate.gif](#)

[uvas.gif](#)
[tyr_a.gif](#)
[ants.gif](#)
[trieb.gif](#)

<HTML>

<HEAD>

<TITLE>Mi p´gina</TITLE>

</HEAD>

<BODY background=" ../imagenes/paper002.jpg">

<CENTER>

<HR width="80%" align="center" size="10">

</CENTER>

<H1 ALIGN="CENTER">Bienvenido a mi p´gina</H1>

<PRE>Como podr´n darse cuenta esta pagina est´ aun en construcci´n.
Sin embargo aquí est´n unas frutas y lo que pienso de ellas. Pongan
su rat´n sobre la imagen de alguna de ellas para leer los que pienso:</PRE>

<P> </P>

<P>

</P>

<P>

</P>

<P> </P>

<P>

Yo alguna vez estude Ingeniería Química en la UAM Azcapotzalco pero después de un tiempo descubrí que lo mí eran las computadoras. Ahora estudio Matemáticas Aplicadas y Computación en la E.N.E.P. Acatlán.

De cualquier forma, la imagen es la molécula de tirosina... Algo quedo de mis tiempos de estudiante de ingeniería química.

</P>

<P> </P>

<P>Este es un GIF animado y tiene un borde:</P>

<CENTER>

<P></P>

<P> </P>

<P>Si quieren regresar a mi página hagan click aquí:</P>

<P></P>

</CENTER>

</BODY>

</HTML>

Este es el resultado



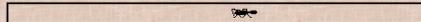
Bienvenido a mi página

Como podrán darse cuenta esta pagina está aun en construcción.
Sin embargo aquí están unas frutas y lo que pienso de ellas. Pongan
su ratón sobre la imagen de alguna de ellas para leer lo que pienso:



Yo alguna vez estude Ingeniería Química en la UAM Azcapotzalco pero después de un tiempo descubrí que lo mío eran las computadoras. Ahora estudio Matemáticas Aplicadas y Computación en la E.N.E.P. Acatlán.
De cualquier forma, la imagen es la molécula de tirosina... Algo quedo de mis tiempos de estudiante de ingeniería química.

Este es un GIF animado y tiene un borde:



Fin de la Sección 4.

Sección 5. Etiqueta <META>

[5. 1 Etiqueta <META>](#)

[5. 2 Sobre HTTP-EQUIV](#)

[5. 3 Máquinas de búsqueda](#)

[5. 4 Ejemplos](#)

Hasta ahora hemos visto como crear una página sencilla. Nosotros sabemos que la finalidad será colocarla en un servidor y mostrarla al mundo. Las ultimas dos secciones de esta primera parte nos ayudaran a conseguir eso. En esta sección trataremos la etiqueta <META> que es la llave que abre la puerta de nuestra página o sitio a nuestros visitantes y las máquinas de búsqueda que, a la larga, atraerán visitantes.

5. 1 Etiqueta <META>

<META> permite al autor de un documento HTML especificar información fuera del contenido. Esta información recibe el nombre de meta data o meta datos. Los atributos de <META> son

NAME: Nombre de la propiedad

CONTENT: Valor de la propiedad

HTTP-EQUIV: Encabezados HTTP

A diferencia de las etiquetas vistas hasta ahora en donde los atributos tenían propiedades especificas, los atributos de <META> reciben propiedades especificadas por el autor.

El atributo NAME declara la propiedad y el atributo CONTENT especifica el valor del atributo. Por ejemplo

```
<META NAME="Autor" CONTENT="Jorge L. Rivas">
```

La cantidad de meta datos depende del autor pero podemos listar algunas que se repiten:

Author: Autor.

Keywords: Palabras clave que identifiquen el contenido.

Description: Una breve descripción de la página.

No necesariamente deben de ser en ingles pero se utilizan como estándar pues algunas máquinas de búsqueda tal vez no las entenderán en otro idioma. Podríamos crear una lista de meta datos especificando diferentes lenguajes para las palabras clave, por ejemplo, usando el atributo LANG:

```
<!--Ingles:-->
```

```
<META NAME="keyword" lang="en-us" content="home, page, linux, software">
```

```
<!--Español:-->
```

```
<META NAME="keyword" lang="es" content="pagina, personal, linux, software">
```

```
<!--Cualquier cosa que se les ocurra pero descriptiva-->
```

5. 2 Sobre HTTP-EQUIV

El Atributo HTTP-EQUIV nos permitirá:

1. Refrescar la pantalla después de un tiempo determinado:

```
<META HTTP-EQUIV="refresh" CONTENT="10">: Espera 10 segundos antes de refrescar la pantalla
```

2. Redireccionar a nuestros visitantes a otra página:

```
<META HTTP-EQUIV="refresh" CONTENT="10;URL=http://www.avantel.net/~trieb/algunapagina.html">: Espera 10 segundos antes de redireccionar a http://www.avantel.net/~trieb/algunapagina.html
```

3. Indicar la caducidad de una página.

```
<META HTTP-EQUIV="expires" CONTENT="Fri, 23 Jan 2001 23:11:00 GMT">: Al indicar la caducidad de una página podemos obligar al navegador a que cargue una copia mas reciente que la que se encuentra en cache.
```

5. 3 Máquinas de Búsqueda

Cuando necesitamos información lo primero que hacemos es ir a Altavista, Yahoo, Google o cualquier otra que conozcamos, escribimos alguna palabra clave y de pronto nos vemos inundados en una cantidad gigantesca de resultados pero... ¿alguna vez nos hemos puesto a pensar en como Altavista puede desplegar tantos resultados? ¿De donde salieron tantas páginas?

Las máquinas de búsqueda tienen tres partes fundamentales:

- Una Araña cuya labor es andar por toda la red buscando vínculos, cuando los encuentra los sigue, lee la información que encuentra y la manda al
- Catalogo que es una colección de páginas o sitios. Este catalogo es entonces objeto de búsqueda por el
- Buscador que es el que se encarga de recibir todas las peticiones de búsqueda, compararlas con el catalogo y presentarlas a quien hizo la petición.

Ejemplos de máquinas de búsqueda son:

- Altavista
- Google
- Lycos
- Excite
- Infoseek

Otra opción a las máquinas de búsqueda son los directorios. Los directorios es una colección de vínculos ordenados por tema.

Ejemplos de directorios:

- Yahoo

En la próxima sección veremos como agregar nuestra página a las máquinas de búsqueda y directorios.

5. 4Ejemplos

Primer Documento:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Esta página se ha mudado</TITLE>
```

```
<META http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

```
<META NAME="author" CONTENT="Jorge L. Rivas">
```

```
<META HTTP-EQUIV="refresh" CONTENT="10;URL=ejemplosecc4.htm">
```

```
</HEAD>
```

```
<BODY BGCOLOR="#FFFFFF">
```

```
<H1 ALIGN="center"><U>Esta página se ha mudado. Usted será redireccionado  
después de 10 segundos</U></H1>
```

```
<H3 ALIGN="center">Si algo falla siga el vinculo: <A  
href="ejemplosecc4.htm">ejemplosecc4.html</A></H3>
```

```
</BODY>
```

```
</HTML>
```

Segundo Documento (sección anterior):

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Mi página</TITLE>
```

```
<META name="keywords" content="durazno, limon, uvas, tomate, hormigas">
```

```
<META name="keywords" content="matematicas, quimica">
```

```
<META name="description" content="Una simple página personal">
```

```
<META name="Author" content="Jorge L. Rivas">
```

```
</HEAD>
```

```
<BODY background="paper002.jpg">
```

```
<CENTER>
```

```
<IMG src="../imagenes/undercons.gif" width="261" height="78" alt="En construcción">
```

```
<HR width="80%" align="center" size="10">
```

```
</CENTER>
```

```
<H1 ALIGN="CENTER"><FONT FACE="Arial, Helvetica, sans-serif">Bienvenido a mi  
página</FONT></H1>
```

```
<PRE>Como podrán darse cuenta esta página está aun en construcción.  
Sin embargo aquí están unas frutas y lo que pienso de ellas. Pongan  
su ratón sobre la imagen de alguna de ellas para leer los que pienso:</PRE>
```

```
<P>&nbsp;</P>
```

```
<P>
```

```
<IMG src="../imagenes/durazno.gif" width="64" height="65" vspace="20" hspace="20" alt="Sin  
comentarios">
```

```
<IMG src="../imagenes/limon.gif" width="73" height="65" vspace="20" hspace="20" alt="En agua  
de limon pero nunca solo">
```

```
</P>
```

<P>

</P>

<P> </P>

<P>

Yo alguna vez estude Ingeniería Química en la UAM Azcapotzalco pero después de un tiempo descubrí que lo mío eran las computadoras. Ahora estudio Matemáticas Aplicadas y Computación en la E.N.E.P. Acatlán.

De cualquier forma, la imagen es la molécula de tirosina... Algo quedo de mis tiempos de estudiante de ingeniería química.

</P>

<P> </P>

<P>Este es un GIF animado y tiene un borde:</P>

<CENTER>

<P></P>

<P> </P>

<P>Si quieren regresar a mi página hagan click aquí:</P>

<P></P>

</CENTER>

</BODY>

</HTML>

Fin de la secci3n 5

- [6. 0 Características del Sitio](#)
- [6. 1 Galerías de imágenes](#)
- [6. 2 Programas de intercambio de banners](#)
- [6. 3 Buscando un terreno](#)
- [6. 4 Dándonos a conocer](#)
- [6. 5 Apoyando nuestra causa](#)
- [6. 6 Algunas palabras sobre la publicidad](#)
- [6. 7 Estadísticas y contadores](#)

6. 0 Características del Sitio

Ahora que ya contamos con las bases necesarias para crear un sitio sencillo, lo siguiente es comenzar a hacerlo. Probablemente ya escogimos un tema adecuado; puede ser nuestra pagina personal con información sobre nosotros (tal vez nuestro currículum) o imágenes de nuestras vacaciones, la pagina de nuestro negocio o simplemente un lugar en donde mostrar información que creamos importante.

Sin importar que tipo de sitio estemos construyendo habrán algunas características necesarias para hacerlo agradable al visitante.

Tener claro el tipo de sitio a crear

A veces uno comienza el proceso de creación de un sitio sin tener claro cuales serán sus objetivos. El sitio de una empresa que ha sido construido sin una planeación previa tendera al fracaso, por ejemplo, la navegación puede ser deficiente, la información de nuestros productos o servicios puede ser confusa o simplemente la pagina principal es realmente lenta. Al planear nuestro sitio debemos hacernos las siguientes preguntas:

1. ¿Qué deseo lograr con este sitio?
2. ¿A quien va dirigida?
3. ¿Qué cantidad de información presentare?
4. ¿Es esta información realmente necesaria?
5. ¿Cuento con esa información o tendré que hacer un estudio para obtenerla?
6. ¿Cuento con la tecnología necesaria para colocar mi sitio?
7. ¿Cuenta el cliente potencial con la tecnología necesaria para visitar mi sitio?

Estas son algunos ejemplos de preguntas pero podemos cuestionarnos mas o menos según sea el caso.

Creo que una clasificación posible para los tipos de sitios podría ser la siguiente

1. Sitios Personales: El tema principal es el ego de alguien. La información ahí mostrada es, la mayoría de las veces, banal. Sin embargo existen algunas paginas personales que contienen información de utilidad. No se dan ejemplos para no herir los sentimientos de alguien.
2. Sitios de información: Sitios en los cuales se puede encontrar información sobre un tema o temas en específico. Ejemplos de estos sitios son las maquinas de búsqueda, tutoriales, libros y revistas en línea, enciclopedias, periódicos, etc.

3. Sitios Empresariales: Su objetivo principal es mostrar al público un producto y servicio, ventas electrónicas etc.

Cabe hacer notar que la clasificación no es exacta pues la mayoría de los sitios se encuentran en 2 o mas categorías. De todas maneras la clasificación puede resultar útil al momento de pensar en nuestro futuro sitio.

Consistencia

Si en nuestra pagina tenemos un menú de navegación, seria buena idea incluirlo en todas las paginas del sitio, así ayudamos a crear una imagen de sitio. Los colores y tipos de letra deberán ser similares, muchas diferencias entre paginas de un mismo sitio pueden confundir al visitante al grado de perderse.

Espacio

Un libro esta dividido en párrafos y tiene márgenes... estas ideas las debemos utilizar también al construir nuestro sitio. ¿Han notado alguna vez lo molesto que resulta tener que mover la barra de desplazamiento para leer el texto que parece continuar infinitamente hacia la derecha?

Procuremos no amontonar la información en un solo espacio, dejemos algunos espacios en blanco para que la vista no se canse; muchos accesorios en una sola pagina pueden abrumar a nuestro visitante.

Ortografía

¿Que se puede decir sobre esto? Si no somos muy buenos en el arte de escribir sin errores ortográficos por lo menos podemos usar un procesador de texto o pedir a alguien que nos ayude a revisar. Por experiencia propia lo digo.

Atracción

Solo puedo decir que el amor llega por los ojos.

Tamaño

Nuestros visitantes no son cautivos. Si algo no les gusta se van rápidamente sin decir porque.

Se dice que un sitio tiene tan solo 10 segundos para llamar la atención del visitante, para mostrar algo de su interés. La velocidad de nuestra pagina principal debe de ser aceptable aun para gente con módems de 14.4. Para evitar lentitud nuestras imágenes deben de ser pequeñas y no debemos incluir tecnologías extrañas al navegador (¿quien no se ha topado con una pagina principal que

muestra un mensaje: "cargando" o algo similar?). Si desean ver la velocidad con que debe de mostrarse una pagina principal, visiten Yahoo.

Recuerden, la mayoría de nuestros visitantes llegaran a nuestro sitio por error.

Simplicidad

¿Que hay detrás de esos sitios que alardean de poseer las tecnologías mas modernas? Un tiempo de espera mayor al acceder una pagina. La simplicidad hará que no nos preocupemos tanto por el tamaño de nuestra pagina. Probablemente un fondo blanco sea mejor que una imagen de fondo o, una barra de navegación con imágenes pequeñas sea mejor que un menú gigantesco hecho en flash o java.

La simplicidad también nos evitara los problemas de moda pues lo simple nunca pasa de moda.

Facilidad

Cuando me refiero a facilidad me refiero a la sensación de un visitante no experto. Debemos evitar tamaños de letra pequeños pues, no todos los visitantes tienen tan buena vista como nosotros, escoger bien los colores de letra y fondo (no escoger letras amarillas en un fondo blanco por ejemplo), escoger letras de fácil lectura como times o arial. Evitar eliminar los subrayados de los vínculos pues aunque se ven bien, mucha gente no sabe donde dar click. Si es posible agregar un buscador de información. No queremos hacer sufrir al visitante.

Estandarización

La estandarización significa poder acceder a nuestro sitio a través de cualquier navegador (Netscape, IE, Mozilla, Mosaic, Lynx). Debemos siempre de pensar que nuestro visitante tiene la peor computadora del universo, con programas viejos... así nunca fallaremos.

Sobre las imágenes podemos decir que es mejor presentar imágenes JPEG, GIF o PNG y evitar formatos extraños.

Procuremos también proporcionar una versión de solo texto para aquella gente "extraña" (yo a veces soy de este tipo de gente) que navega con solo texto.

6. 1 Galerías de imágenes

Aquí encontraran galerías de imágenes para incluir en su sitio. Antes de pegar 200 de ellas en su sitio, pregúntense si existe realmente necesidad. Lo que a uno le puede gustar, a otros los puede molestar.

- [Mediatheque](#)
- [GeneralSearch - Image Gallery](#)

- [Intergrafx](#)

6. 2 Programas de intercambio de banners

Una manera de promocionar nuestro sitio es a través de los programas de intercambio de banners. Como siempre, debemos de ser cuidadosos a la hora de incluir un banner pues puede aumentar el tiempo de descarga de nuestro sitio.

- [MS-Links Exchange](#)
- [LinkExchange](#)

6. 3 Buscando un terreno

Web Hosting Gratuito

El único problema es que la mayoría de los sitios que ofrece estos servicios nos obligan a mostrar publicidad en nuestro sitio. Antes de buscar uno de estos proveedores debemos de hablar con nuestros ISP (Proveedor de servicios de Internet) para ver si tenemos un espacio disponible para una pagina y evitarnos la molesta publicidad.

- [Geocities](#)
- [Starmedia](#)
- [NBCI](#)

Web Hosting

Existen distintas compañías de Web Hosting en México y en el extranjero con buenos precios, pero antes de comprar revisemos unas 10 o 20 de ellas para encontrar las mejores.

Para hacer una buena elección de nuestro proveedor de Web Hosting debemos de tener en cuenta:

1. Tipo de Servidor: Un servidor UNIX a diferencia de uno basado en Windows NT será mas estable y probablemente mas económico. Sin embargo al hacer la elección tendremos en cuenta nuestras necesidades.
2. Cuentas de correo y espacio en disco: ¿El plan que nos interesa cuenta con suficientes cuentas de correo y el espacio suficiente para nuestro sitio? El sitio de una empresa pequeña puede necesitar tan solo unos 5 o 10 Mb de espacio en disco y cuentas de correo para mandos altos y medios.
3. Precio: Los precios pueden ir desde tan solo unos 5 dólares mensuales hasta los 100 dólares mensuales dependiendo de las características del paquete. Generalmente los paquetes mas caros incluyen una cantidad importante de cuentas de correo, espacio en disco, encriptación SSL de 128 bits, PHP , soporte para base de datos y carrito de compras, todo esto para transacciones seguras a través de Internet.

4. Trafico: Esto es, la cantidad de visitantes que pueden visitar nuestro sitio al mes. Aproximadamente 1Gb de transferencia al mes nos permitirá recibir unos 75,000 visitantes.
5. Capacidades de Script: Incluye un directorio CGI-BIN, etc.
6. Tipo de líneas: T1, T3 etc.
7. ¿Ofrece acceso a Internet?: Si ofrece acceso a Internet debemos eliminarla como opción pues sus líneas pueden estar saturadas por una cantidad enorme de gente tratando de conectarse a Internet.
8. Soporte Técnico

Dominios

Un dominio, en pocas palabras, es la que será la dirección de nuestro sitio, por ejemplo: www.misitio.com y es necesario para el Web Hosting no gratuito. Para Información pueden visitar www.networksolutions.com

6. 4 Dándonos a conocer

En la sección 5 hablamos sobre la etiqueta <META> y su relación con las maquinas de búsqueda. La mayoría de estas nos permitirá agregar nuestro sitio de una manera sencilla. Este proceso puede ser lento por lo que es preferible agregar nuestro sitio solo en aquellas maquinas de búsqueda famosas. Otra opción es contratar un servicio que permitirá agregar nuestro sitio a una cantidad importante de maquinas de búsqueda.

Agregar sitio en:

- [Altavista](#)
- [Yahoo](#)
- [Yahoo México](#)
- [Lycos](#)

6. 5 Apoyando nuestra causa

A los humanos nos gusta ser parte de algo. Nuestras pasiones por cierto tipo de temas pueden ser expresadas de forma sutil dentro de nuestro sitio. Si no tienes ninguna causa que apoyar estas son algunas interesantes:

- [The GNU Project and the Free Software Foundation](#)
- [Electronic Frontier Foundation](#)

6. 6 Algunas palabras sobre la publicidad

Es cierto que muchas compañías nos harán pagos por incluir un banner publicitario en nuestro sitio, sin embargo, estos pagos son mínimos por lo que si queremos ganar una buena cantidad de dinero tendremos que incrementar el trafico.

La publicidad agrega algo de estilo a nuestro sitio por eso debemos de tener mucho cuidado con los tipos de productos y servicios que anunciemos en nuestro sitio así como la cantidad de publicidad que aparecerá pues una gran cantidad de banners en una sola pagina afectaran la velocidad y a la larga, nuestro trafico.

6. 7 Estadísticas y contadores

Es necesario contar con estadísticas confiables de las visitas que nuestro sitio recibe. Estas estadísticas nos pueden decir de que país son la mayoría de nuestros visitantes, las horas de mayor trafico así como las paginas mas visitadas en nuestro sitio. Contar con esta información nos permitirá crear un perfil de nuestros visitantes, sus gustos e intereses y, basado en ello, mejorar la forma en que la información que buscan se presenta ante ellos.

A diferencia de las estadísticas, los contadores solo nos permiten saber la cantidad de veces que se ha visitado la pagina que los contiene y no la cantidad de gente que ha visitado la pagina.

[Nedstat](#)

Fin de la sección 6

Sección 7. Tablas

[7. 1 Introducción](#)

[7. 2 Etiqueta <TABLE>](#)

[7. 3 Etiquetas <TR> y <TD>](#)

[7. 4 Etiquetas <TH> y <CAPTION>](#)

[7. 5 Ejemplos](#)

7. 1 Introducción

Muchas veces resulta practico ordenar cierta información dentro de celdas que pueden ser localizadas por medio del numero de columna y renglón en que se encuentra. La siguiente tabla muestra las calificaciones hipotéticas de ciertos alumnos hipotéticos:

Alumno	Calificación 1	Calificación 2	Promedio
Sutanito	10	8	9
Perenganito	5	10	7.5
Sutanita	4	0	2
Perenganita	9	9	9
Calificaciones hipotéticas			

Si hemos llegado hasta este punto del tutorial significa que hemos estudiado y puesto en practica nuestros nuevos conocimientos... y tal vez nos hayamos sentido defraudados por no obtener los resultados deseados. Las tablas nos permitirán dar un mejor formato a nuestro sitio. La mayoría de los sitios comerciales utilizan tablas para disponer de cierta forma la información de forma rápida. El siguiente vinculo muestra un supuesto sitio que utiliza tablas para obtener un diseño sencillo pero profesional.

[Supuesto sitio comercial.](#)

A pesar de las ventajas que nos brindan las tablas para obtener resultados al momento de dar formato a un documento, esto no significa que sea la opción mas sencilla. De hecho el ejemplo del supuesto sitio comercial nos permite ver la complejidad a la cual pueden llegar ciertos documentos HTML. Probablemente la opción mas sencilla sea el uso de hojas de estilo. De esto se hablara en secciones posteriores, por lo pronto sigamos con las tablas, al fin de cuenta, todo se complicara en la medida que nosotros queramos y no hay nada que temer.

Ejemplo de una tabla sencilla. 5 renglones, 5 columnas				

7. 2 Etiqueta <TABLE>

La etiqueta <TABLE> nos permite crear una tabla. Todo lo que este dentro de estas etiquetas será considerado como parte de la tabla. La etiqueta de cierre es </TABLE>.

Los posibles atributos de la etiqueta <TABLE> son:

ALIGN

Permite especificar la posición de la tabla respecto al documento. Los posibles valores de este atributo son LEFT (izquierda), CENTER (centro) y RIGHT (derecha).

BGCOLOR

Simple. Da un color de fondo a la tabla

BORDER

Especifica el tamaño del borde de la tabla, se mide en píxeles.

BORDERCOLOR

Asigna un color al borde.

CELLSPACING

Especifica la cantidad de espacio entre celdas.

Ejemplo con CELLSPACING = 1		

Ejemplo con CELLSPACING = 10		

CELLPADDING

Especifica la cantidad de espacio entre el borde de una celda y su contenido

Renglón 1 Columna 1	Renglón 1 Columna 2	Renglón 1 Columna 3
Renglón 2 Columna 1	Renglón 2 Columna 2	Renglón 2 Columna 3
Renglón 3 Columna 1	Renglón 3 Columna 2	Renglón 3 Columna 3
Ejemplo con CELLPADDING = 1		

Renglón 1 Columna 1	Renglón 1 Columna 2	Renglón 1 Columna 3
Renglón 2 Columna 1	Renglón 2 Columna 2	Renglón 2 Columna 3
Renglón 3 Columna 1	Renglón 3 Columna 2	Renglón 3 Columna 3
Ejemplo con CELLSPADDING = 10		

WIDTH

Permite especificar el ancho de una tabla respecto al documento. El ancho puede ser en píxeles o en porcentaje. Dicho porcentaje se refiere a la ventana del navegador. Esta ultima forma es la recomendada para evitar que el visitante tenga que desplazarse horizontalmente.

Otros atributos que pueden o no servir de un navegador a otro son:

BORDERDARK

Determina el color para la parte oscura del borde de la tabla.

BORDERLIGHT

Determina el color para la parte clara del borde de la tabla.

BACKGROUND

Permite especificar una imagen como fondo de la tabla.

Específicamente, estos 3 funcionan en IExplorer 5. Y en Netscape no funcionaron con la versión 6. Por este sencillo hecho, es recomendable no utilizarlos pues al construir un sitio debemos de pensar que no todos tienen el mismo navegador que nosotros. Seamos neutrales.

7. 3 Etiquetas <TR> y <TD>

Una vez que se formo el cascaron de la tabla comenzaremos a crear nuestras celdas. Para esto utilizaremos las etiquetas <TR> y <TD>. En ambos casos la etiqueta de cierre es opcional.

La etiqueta <TR> nos permite crear los distintos renglones de nuestra tabla. Por ejemplo, una tabla de 5 renglones podría escribirse así:

```
<TABLE>
<TR> Renglón 1
<TR> Renglón 2
<TR> Renglón 3
<TR> Renglón 4
<TR> Renglón 5
</TABLE>
```

Sin embargo, para crear las columnas (y así crear las celdas) necesitaremos la etiqueta <TD>, así que nuestro ejemplo sería más correcto de la siguiente forma:

```
<TABLE BORDER="2">
<TR><TD>Renglón 1
<TR><TD>Renglón 2
<TR><TD>Renglón 3
<TR><TD>Renglón 4
<TR><TD>Renglón 5
</TABLE>
```

Renglón 1
Renglón 2
Renglón 3
Renglón 4
Renglón 5

así generamos una tabla de 5 renglones y una columna.

Los atributos para la etiqueta <TR> son:

BGCOLOR

Especifica el color de fondo de todo el renglón.

```
<TABLE BORDER="2">
<TR BGCOLOR="#6666FF"><TD>Renglón 1
<TR><TD>Renglón 2
<TR><TD>Renglón 3
<TR><TD>Renglón 4
<TR><TD>Renglón 5
</TABLE>
```

Renglón 1
Renglón 2
Renglón 3
Renglón 4
Renglón 5

ALIGN

Alinea el texto dentro de una celda. Los valores son LEFT (izquierda), CENTER (centro) y RIGHT (derecha).

```
<TABLE BORDER="2">
<TR><TD>Esta es la columna 1
<TR ALIGN="LEFT"><TD>LEFT
<TR ALIGN="CENTER"><TD>CENTER
<TR ALIGN="RIGHT"><TD>RIGHT
</TABLE>
```

Esta es la columna 1
LEFT
CENTER
RIGHT

Los atributos para la etiqueta <TD> son:

BGCOLOR

Especifica el color de fondo de la celda.

```
<TABLE BORDER="2">
<TR BGCOLOR="#FF0000"><TD>Renglón 1
<TR><TD BGCOLOR="#0000FF">Renglón 2
<TR><TD>Renglón 3
<TR><TD>Renglón 4
<TR><TD>Renglón 5
</TABLE>
```

Renglón 1
Renglón 2
Renglón 3
Renglón 4
Renglón 5

ALIGN

Similar a la anterior definición de ALIGN.

COLSPAN

Indica la cantidad de celdas adyacentes a la derecha que se combinaran como una sola.

```
<TABLE BORDER="2">
<TR><TD> 1 <TD> 2
<TR><TD> 3 <TD> 4
<TR><TD COLSPAN="2"> Ejemplo de COLSPAN
<TR><TD> 5 <TD> 6
<TR><TD> 7 <TD> 8
</TABLE>
```

1	2
3	4
Ejemplo de COLSPAN	
5	6
7	8

ROWSPAN

Indica la cantidad de celdas adyacentes y abajo que se combinaran como una sola.

```
<TABLE BORDER="2">
<TR><TD ROWSPAN ="3"> Ejemplo de ROWSPAN <TD> 1
<TR><TD> 2
<TR><TD> 3
<TR><TD> 4 <TD> 5
<TR><TD> 6 <TD> 7
</TABLE>
```

Ejemplo de	1
ROWSPAN	2
	3
4	5
6	7

NOWRAP

Evita que el texto sea ajustado dentro de una celda.

```
<TABLE>
<TR><TD> 1 <TD> 2
<TR><TD> 3 <TD> 4
<TR><TD COLSPAN="2"> Ejemplo de COLSPAN y NOWRAP
<TR><TD> 5 <TD> 6
<TR><TD> 7 <TD> 8
</TABLE>
```

1	2
---	---

3	4
Ejemplo de COLSPAN y NOWRAP	
5	6
7	8

WIDTH

Especifica la anchura que tendrá la celda con respecto a la tabla. Se puede especificar el tamaño en píxeles y en porcentaje. El ultimo es el recomendado.

```
<TABLE WIDTH="50%" BORDER="2">
<TR ALIGN="CENTER"><TD WIDTH="15%">15%<TD WIDTH="60%">60%<TD
WIDTH="25%">25%
<TR ALIGN="CENTER"><TD>1<TD>2<TD>3
<TR ALIGN="CENTER"><TD>4<TD>5<TD>6
<TR ALIGN="CENTER"><TD>7<TD>8<TD>9
</TABLE>
```

15%	60%	25%
1	2	3
4	5	6
7	8	9

Este ejemplo crea una tabla que ocupa la mitad del espacio disponible de la ventana del navegador. Las columnas son de tamaño 15%, 60% y 25% respectivamente.

HEIGHT

Nos permite especificar la altura de las celdas.

```
<TABLE WIDTH="50%" BORDER="2">
<TR ALIGN="CENTER"><TD HEIGHT="50" WIDTH="15%">15%<TD
WIDTH="60%">60%<TD WIDTH="25%">25%
<TR ALIGN="CENTER"><TD>1<TD>2<TD>3
<TR ALIGN="CENTER"><TD>4<TD>5<TD>6
<TR ALIGN="CENTER"><TD>7<TD>8<TD>9
</TABLE>
```

15%	60%	25%
1	2	3
4	5	6
7	8	9

En este ejemplo las unidades recomendadas son los píxeles, sin embargo debemos de ser precavidos al usar este atributo.

VALIGN

Permite especificar la posición vertical de los contenidos de una celda. Los valores que puede tomar son: TOP (arriba), MIDDLE (en medio), BOTTOM (abajo) y BASELINE.

```
<TABLE BORDER="2">
<TR>
<TD VALIGN="TOP">
<P>TOP
<P>&nbsp;
<P>&nbsp;
<TD VALIGN="MIDDLE">MIDDLE
<TD VALIGN="BOTTOM">BOTTOM
</TABLE>
```

TOP		
	MIDDLE	
		BOTTOM

7. 4 Etiquetas <TH> y <CAPTION>

La etiqueta <TH> nos permitirá exhibir texto como un título para la columna. Esta etiqueta muestra el texto contenido dentro de ella en negritas y centrado. Los atributos de esta etiqueta son los mismos que los de <TD>.

```
<TABLE BORDER="1">
<TR><TD ALIGN="CENTER">-<TH>Columna 1<TH>Columna 2<TH>Columna 3
<TR ALIGN="CENTER"><TD>Renglón 1<TD>1, 1<TD>1, 2<TD>1, 3
<TR ALIGN="CENTER"><TD>Renglón 2<TD>2, 1<TD>2, 2<TD>2, 3
<TR ALIGN="CENTER"><TD>Renglón 3<TD>3, 1<TD>3, 2<TD>3, 3
</TABLE>
```

-	Columna 1	Columna 2	Columna 3
Renglón 1	1, 1	1, 2	1, 3
Renglón 2	2, 1	2, 2	2, 3
Renglón 3	3, 1	3, 2	3, 3

La etiqueta <CAPTION> nos permite escribir una pequeña descripción o título para la tabla. La etiqueta de cierre es </CAPTION> y es necesaria. Esta etiqueta debe estar siempre después de la etiqueta <TABLE> para que surta efecto.

```

<TABLE BORDER="1">
<CAPTION ALIGN="BOTTOM">Ejemplo de <TH> y <CAPTION>; </CAPTION>
<TR><TD ALIGN="CENTER">-<TH>Columna 1<TH>Columna 2<TH>Columna 3
<TR ALIGN="CENTER"><TD>Renglón 1<TD>1, 1<TD>1, 2<TD>1, 3
<TR ALIGN="CENTER"><TD>Renglón 2<TD>2, 1<TD>2, 2<TD>2, 3
<TR ALIGN="CENTER"><TD>Renglón 3<TD>3, 1<TD>3, 2<TD>3, 3
</TABLE>

```

-	Columna 1	Columna 2	Columna 3
Renglón 1	1, 1	1, 2	1, 3
Renglón 2	2, 1	2, 2	2, 3
Renglón 3	3, 1	3, 2	3, 3
Ejemplo de <TH> y <CAPTION>			

El atributo ALIGN de esta etiqueta permite especificar la posición de la descripción o título. Los posibles valores pueden ser TOP (arriba), BOTTOM (abajo), LEFT (izquierda) y RIGHT (Derecha).

7.5 Ejemplos

Supuesto Sitio Comercial

Para desarrollar el sitio se utilizarán las siguientes imágenes. ¿Se sabían el truco del pixel?

- [pixel.gif](#) (salvenla con el botón derecho del mouse a menos de que tengan buen pulso)
- [eriol_slashdot.jpg](#)

El código del [Supuesto Sitio Comercial](#) es el siguiente:

```

<HTML>
<HEAD>
<TITLE>Supuesto Sitio Comercial.com</TITLE>
</HEAD>

<BODY bgcolor="#FFFFFF" leftmargin="0" topmargin="0" marginwidth="0" marginheight="0">

<TABLE width="100%" border="0" cellspacing="0" cellpadding="0">
<TR bgcolor="006699">

<TD bgcolor="006699" width="90%">

<H1><FONT color="#FFFFFF" face="Arial"> Supuesto Sitio Comercial</FONT><FONT
color="#000000">.com</FONT></H1>

```

<TD>

<TR bgcolor="#000000">

<TD>

<TABLE border="0" cellspacing="3" cellpadding="0">

<TR>

<TD><U>P´gina
Principal</U>

<TD> |

<TD><U>Servicios</U>

<TD> |

<TD><U>Productos</U>

<TD> |

<TD><U>FAQ</U>

</TABLE>

<TD align="center"><FONT color="#CCCCCC" face="Arial"
size="2"><U>Ayuda</U>

</TABLE>

<P align="center">

<U>Publicidad</U>

<TABLE width="90%" border="0" cellspacing="0" cellpadding="0" align="center">

<TR>

<TD bgcolor="#CCCCCC" width="1">

<TD width="34%" height="300">

<!--Las siguientes lineas permiten dar el espacio vertical para que-->

<!-- se formen las lineas de division-->

<P> </P>

<TD bgcolor="#CCCCCC" width="1">

<TD width="33%"> </TD>

<TD bgcolor="#CCCCCC" width="1">

<TD width="33%"> </TD>

<TD bgcolor="#CCCCCC" width="1">

</TABLE>

<P> </P>

<HR align="center" size="1" width="90%" noshade>

<P align="center">

|
<U>P´gina Principal</U>

|
<U>Servicios</U>

|
<U>Produtos</U>

|
<U>FAQ</U>

|
<U>Politica de Privacidad</U>

|

Correo al webmaster: <U><FONT
color="#0000FF">webmaster@supuestositiocomercial.com</U>

© 2001. Supuesto Sitio Comercial.

Ciudad de México

</BODY>

</HTML>

Fin de la sección 7

Sección 8. Formularios

[8.1 Introducción](#)

[8.2 Etiqueta <FORM>](#)

[8.3 Etiqueta <INPUT>](#)

[8.4 Campos de texto](#)

[8.5 Campos de contraseña](#)

[8.6 Áreas de texto](#)

[8. 7 Casillas de verificación](#)

[8. 8 Botones de opción](#)

[8. 9 Menús](#)

[8. 10 Botones](#)

[8. 11 Campos de archivos](#)

[8. 12 Ejemplo](#)

8. 1 Introducción

Un formulario es una manera de reunir información del público que visita nuestra página.

Existen diferentes formas de procesar la información de un formulario:

PHP

PHP es un lenguaje muy parecido a C y Bash, que nos permite escribir scripts. Las ventajas que tiene sobre otro tipo de lenguajes es que es software libre (libre como en libertad), además de que está muy difundido en Internet. PHP puede estar dentro del código HTML y nos puede ayudar, por ejemplo, a enviar o recibir información desde una base de datos (i.e. MySQL) vía un formulario HTML. Actualmente las versiones más difundidas de PHP son las versiones 3 y 4. Para más información sobre PHP consulten: <http://www.php.net>

Tal vez después escriba algo sobre PHP.

Correo Electrónico

La forma más sencilla de procesar un formulario. La información se envía por medio del correo electrónico. Este método será del que hablemos en esta sección pues los otros dos necesitan conocimientos un poco más avanzados.

Además de estas dos, existen otras formas como vía CGI (Common Gateway Interface) y ASP (Active Server Page) de Microsoft.

8. 2 Etiqueta <FORM>

La etiqueta <FORM> definirá un formulario con los diversos elementos que puede contener. De estos elementos se hablará más tarde. La etiqueta de cierre es </FORM>.

Los atributos posibles en la etiqueta <FORM> son:

ACTION

Especifica la acción a tomar después de que el usuario ha apretado el botón "enviar" de un formulario. En nuestro caso, la acción a tomar será enviar la información vía correo electrónico a la dirección de correo electrónico especificada, para esto utilizamos el valor MAILTO:

```
<FORM ACTION="MAILTO:alguien@algunlugar.com ">
```

METHOD

Especifica la forma en que se enviarán los datos del formulario:

```
<FORM ACTION="MAILTO:alguien@algunlugar.com" METHOD="POST">
```

ENCTYPE

Especifica el formato que tendrán los datos del formulario cuando se utiliza el valor POST:

```
<FORM ACTION=" MAILTO:alguien@algunlugar.com " METHOD="POST"  
ENCTYPE="TEXT/PLAIN">
```

Es decir, texto sin formato.

8. 3 Etiqueta <INPUT>

La etiqueta <INPUT> nos permitirá crear cada uno de los elementos que queramos utilizar dentro de un formulario. Por el momento solo veremos los atributos de esta etiqueta.

Los atributos pueden ser:

TYPE

Define el tipo de elemento que se puede utilizar. Los valores para este atributo pueden ser: CHECKBOX, PASSWORD, RADIO, RESET, SUBMIT y TEXT los cuales se verán mas adelante.

```
<INPUT TYPE="TEXT">  
<INPUT TYPE="PASSWORD">  
<INPUT TYPE="RADIO">  
<INPUT TYPE="CHECKBOX">  
<INPUT TYPE="RESET">  
<INPUT TYPE="SUBMIT">
```

NAME

Da un nombre específico a cada elemento del formulario. Al enviar el formulario nosotros recibiremos los datos organizados de la forma NAME = VALUE. El nombre para cada elemento debe ser único y descriptivo:

```
<INPUT TYPE="..." NAME="nombre">
```

VALUE

Permite mostrar un valor predeterminado en el elemento de formulario:

```
<INPUT TYPE="..." NAME="..." VALUE="valor">
```

8. 4 Campos de texto

Los campos de texto son rectángulos en donde el usuario puede escribir una cadena corta de texto. Los campos de texto se crean con el valor TEXT para el atributo TYPE de la etiqueta <INPUT>:

```
<INPUT TYPE="TEXT" NAME="..." VALUE="...">
```

Ejemplo de un campo de texto.

Otros atributos que pueden ser utilizados para crear campos de texto son:

MAXLENGTH

Define la cantidad de caracteres que pueden ser introducidos en el campo de texto:

```
<INPUT TYPE="TEXT" MAXLENGTH="5">
```

En este campo solo se pueden introducir 5 caracteres.

SIZE

Define el tamaño horizontal del campo de texto:

```
<INPUT TYPE="TEXT" SIZE="5">
```

Campo de tamaño 5.

VALUE

Este atributo permite mostrar un texto predeterminado por nosotros en el campo de texto:

```
<INPUT TYPE="TEXT" VALUE="Hola Mundo">
```

Hola Mundo

Campo de texto con un valor predeterminado.

8. 5 Campos de contraseña

Al igual que los campos de texto, los campos de contraseña son rectángulos en donde el usuario puede escribir una cadena corta de texto; la única diferencia es que el texto en este campo es reemplazado por asteriscos.

```
<INPUT TYPE="PASSWORD" NAME="..." VALUE="...">
```

Ejemplo de campo de contraseña.

Otros atributos que pueden ser utilizados para crear campos de contraseña son:

MAXLENGTH

Define la cantidad de caracteres que pueden ser introducidos en el campo de contraseña.

```
<INPUT TYPE="TEXT" MAXLENGTH="5">
```

En este campo solo se pueden introducir 5 caracteres.

SIZE

Define el tamaño horizontal del campo de contraseña.

```
<INPUT TYPE="TEXT" SIZE="5">
```

Campo de tamaño 5.

VALUE

Este atributo permite mostrar una contraseña predeterminada por nosotros en el campo de contraseña.

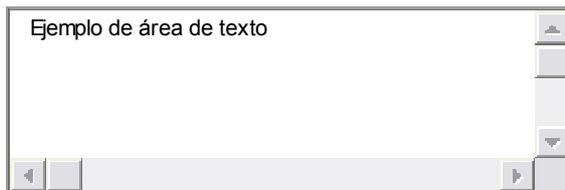
```
<INPUT TYPE="TEXT" VALUE="Hola Mundo">
```

Campo de contraseña con valor predeterminado.

8. 6 Áreas de texto

Si nosotros necesitamos que el visitante nos mande una cadena de texto algo extensa necesitaremos utilizar la etiqueta `<TEXTAREA>`. Esta etiqueta nos permitirá crear un cuadro de texto con barras de desplazamiento. Esta etiqueta también se incluye dentro de la etiqueta `<FORM>`. La etiqueta de cierre es `</TEXTAREA>`.

```
<TEXTAREA ROWS="..." COLS="..." NAME="...">  
</TEXTAREA>
```

A screenshot of a web browser showing a text area input field. The text area contains the text "Ejemplo de área de texto". The text area has a light gray border and a vertical scrollbar on the right side. The text is centered within the text area.

Los atributos posibles en la etiqueta `<TEXTAREA>` son:

ROWS

Define la altura del área de texto. Este atributo no se debe olvidar.

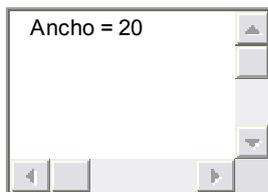
```
<TEXTAREA ROWS="15">
```



COLS

Define el ancho del área de texto. Este atributo no se debe olvidar.

```
<TEXTAREA COLS="20">
```



NAME

Nombra al área de texto.

```
<TEXTAREA NAME="area de texto">
```

READONLY

Nos permite crear un área de texto de solo lectura, es decir, algún texto dentro de ella no podrá ser modificado por el usuario.

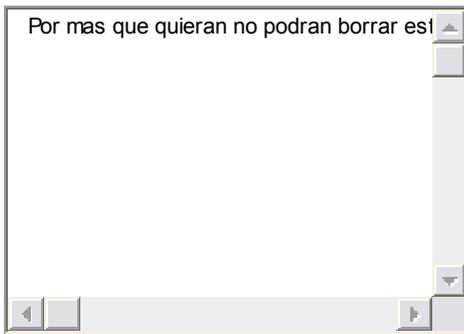
Ejemplo

```
<TEXTAREA name="sololectura" rows="10" cols="40" readonly>
```

Por mas que quieran no podrán borrar esto...

```
</TEXTAREA>
```

Resultado:



8. 7 Casillas de verificación

Una casilla de verificación es un pequeño cuadrado que puede ser seleccionado. Se crea con el valor CHECKBOX del atributo TYPE en la etiqueta <INPUT>.

```
<INPUT TYPE="CHECKBOX" NAME="..." VALUE="...">
```

Ejemplo de una casilla de verificación

Otros atributos que pueden ser utilizados para crear casillas de verificación son:

CHECKED

Muestra la casilla de verificación como marcada.

```
<INPUT TYPE="CHECKBOX" CHECKED>
```

Casilla de verificación marcada

8 . 8 Botones de opción

Un botón de opción permite crear un pequeño círculo que puede o no estar seleccionado. Para el correcto funcionamiento de este tipo de campos, se deben tener por lo menos dos botones de opción que tendrán el mismo nombre.

```
<INPUT TYPE="RADIO" NAME="..." VALUE="...">
```

Ejemplo de botones de Opcion.

Otros atributos que pueden ser utilizados para crear botones de opción son:

CHECKED

Muestra el botón de opción como marcado.

```
<INPUT TYPE="RADIO" CHECKED>
```



El botón de enmedio está seleccionado.

8.9 Menús

Para crear un menú desplegable que permita escoger una o varias opciones necesitaremos dos etiquetas; la etiqueta `<SELECT>` y la etiqueta `<OPTION>`.

La etiqueta `<SELECT>` define un menú desplegable. Su etiqueta de cierre es `</SELECT>`

Los atributos posibles en la etiqueta `<SELECT>` son:

NAME

Nombra al menú.

```
<SELECT NAME="nombre del menú">
```

SIZE

Determina la cantidad de opciones que aparecen desplegadas. El valor predeterminado es 1. Este atributo es útil cuando trabajamos con menús en donde se pueden hacer varias selecciones.

```
<SELECT SIZE="2">
```

MULTIPLE

Permite crear menús en donde se pueden hacer varias selecciones a la vez. La mayoría de la gente no sabe que en este tipo de menús se puede seleccionar más de una opción, así que su uso no es muy recomendable.

```
<SELECT MULTIPLE>
```

Por otra parte, la etiqueta `<OPTION>` define a cada uno de los elementos que aparecerán en el menú desplegable.

```
<SELECT NAME="...">  
<OPTION VALUE="...">Opción 1  
<OPTION VALUE="...">Opción 2  
<OPTION VALUE="...">Opción 3  
</SELECT>
```

Los atributos posibles en la etiqueta <OPTION> son:

SELECTED

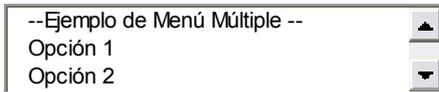
El elemento se mostrará seleccionado

<OPTION SELECTED>Opción 1

VALUE

Define el valor que se recibirá de la forma NAME = VALUE. NAME estará dado por el nombre que hayamos dado al menú en la etiqueta <SELECT>.

<OPTION VALUE="uno">Opción 1



Las distintas selecciones se pueden hacer apretando la tecla CTRL.

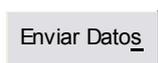
8. 10 Botones

Existen dos tipos de botones básicos que al ser presionados generarán una acción predeterminada.

Botón enviar

Este botón procesará el formulario de la forma que hayamos especificado en la etiqueta <FORM> con el atributo ACTION. Se crea con el valor SUBMIT del atributo TYPE en la etiqueta <INPUT>. En este campo solo necesitará el atributo VALUE para rotular el botón.

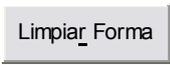
<INPUT TYPE="SUBMIT" VALUE="Enviar datos">



Botón limpiar

Este botón limpiará todos los datos en un formulario. Se crea con el valor RESET del atributo TYPE en la etiqueta <INPUT>. En este campo solo necesitará el atributo VALUE para rotular el botón.

<INPUT TYPE="RESET" VALUE="Limpiar forma">



8. 11 Campos de Archivos

Por ultimo, todos hemos visto aquellas formas en que podemos indicar un archivo para enviar a través de la forma. No crean que necesitan JavaScript para poder hacerlo. Esto lo podemos lograr fácilmente con el atributo TYPE y el valor FILE de la etiqueta <INPUT>.

<INPUT TYPE="FILE" NAME="...">

Los atributos de la etiqueta <INPUT> para este caso son los mismos que vimos anteriormente.

8. 12 Ejemplo

Nombre	<input type="text"/>
Apellido	<input type="text"/>
Correo	<input type="text"/>
Contraseña	<input type="password"/>
¿Que te parece el tutorial?	<input checked="" type="checkbox"/> Excelente <input type="checkbox"/> Mas o Menos <input type="checkbox"/> Pesimo
¿De estas secciones cual o cuales te gustaron mas?	<input type="checkbox"/> Seccion 1 <input type="checkbox"/> Seccion2 <input type="checkbox"/> Seccion 3
Continente:	<input type="text"/>

Enviar

Borrar

Este es el código:

```
<CENTER>
```

```
<FORM ACTION="MAILTO:triebjl@netscape.net" METHOD="post" ENCTYPE="text/plain">
```

```
<TABLE width="50%" border="1">
```

```
<TR>
```

```
<TD>Nombre</TD> <TD><INPUT type="text" name="nombre" size="25"
maxlength="50"></TD>
```

```
</TR>
```

```
<TR>
```

```
<TD>Apellido</TD> <TD><INPUT type="text" name="apellido" size="25"
maxlength="50"></TD>
```

```
</TR>
```

```
<TR>
```

```
<TD>Correo</TD> <TD><INPUT type="text" name="correo" size="30"
maxlength="100"></TD>
```

```
</TR>
```

```
<TR>
```

```
<TD>Contrase&ntilde;a</TD> <TD><INPUT type="password" name="contrase&ntilde;a"
size="8" maxlength="8"></TD>
```

```
</TR>
```

```
<TR>
```

```
<TD>&iquest;Que te parece el tutorial?</TD>
```

```
<TD><INPUT type="radio" name="opinion" value="excelente" checked>Excelente
```

```
<INPUT type="radio" name="opinion" value="masomenos">Mas o Menos
```

```
<INPUT type="radio" name="opinion" value="pesimo">Pesimo</TD>
```

```
</TR>
```

```

<TR>

<TD>&iquest;De estas secciones cual o cuales te gustaron mas?</TD>
<TD><P><INPUT type="checkbox" name="seccion1" value="si">Seccion 1</P>
<P><INPUT type="checkbox" name="seccion2" value="si">Seccion2</P>
<P><INPUT type="checkbox" name="seccion3" value="si">Seccion 3</P></TD>

</TR>

<TR>

<TD>Continente:</TD>

<TD><SELECT name="continente">

<OPTION value="america">America</OPTION>
<OPTION value="europa">Europa</OPTION>
<OPTION value="africa">Africa</OPTION>
<OPTION value="asia">Asia</OPTION>
<OPTION value="oceania">Oceania</OPTION>

</SELECT></TD>

</TR>

</TABLE>

<P><INPUT type="submit" name="enviar" value="Enviar">
<INPUT type="reset" name="borrar" value="Borrar">
</FORM>

</CENTER>

```

Fin de la Sección 8.

Sección 9. Mapas

[9.1 Introducción](#)

[9.2 Etiqueta <MAP>](#)

[9.3 Imágenes como mapas](#)

[9.4 Etiqueta <AREA>](#)

[9.5 Ejemplo](#)

9.1 Introducción

Si nosotros deseamos crear varios vínculos en una sola imagen debemos crear un mapa. Un mapa es una imagen con varias regiones activas que nos servirán como vínculos a otros documentos.

Existen dos tipos de mapas: los mapas del lado del servidor y los mapas del lado del cliente.

Mapas del lado del servidor

Estos mapas requieren para su funcionamiento un script de CGI (Common Gateway Interface). Cuando el usuario hace click en el mapa, el script manda las coordenadas al servidor. Estas coordenadas están vinculadas con un documento.

Mapas del lado del cliente

Estos mapas, a diferencia de los anteriores, no necesitan scripts para su funcionamiento pues la definición del mapa se incluye en el código HTML. Cuando el usuario hace click en el mapa, la maquina del usuario calcula las coordenadas en donde se hizo click y manda al usuario al documento solicitado. La etiqueta `<MAP>` es la responsable de crear un mapa del lado del cliente.

Para crear un mapa del lado del cliente tendremos que saber las coordenadas de la zona activa que funcionará como vínculo. Para saber las coordenadas a utilizar podemos utilizar cualquier software relacionado con imágenes o incluso conseguir un programa especializado para crear mapas.

9. 2 Etiqueta `<MAP>`

Como dijimos antes, la etiqueta `<MAP>` definirá varias zonas activas como vínculos en una sola imagen. Su etiqueta de cierre es `</MAP>`.

Los atributos de esta etiqueta pueden ser:

NAME

Nombra al mapa para poder ser utilizado con una imagen. Este atributo no puede ser omitido.

```
<MAP NAME="MAPA1">
```

9. 3 Imágenes como mapas

Una vez que hemos definido el mapa que utilizaremos con una imagen debemos de utilizar el atributo USEMAP de la etiqueta `` para que dicha imagen sea un mapa.

Sintaxis:

```
<IMG SRC="IMAGEN.JPG" USEMAP="#MAPA1">
```

El valor del atributo USEMAP es el mismo valor que dimos al mapa con el atributo NAME de la etiqueta `<MAP>`. No olvidemos poner el símbolo # antes del nombre del mapa a utilizar.

9. 4 Etiqueta <AREA>

La etiqueta <AREA> definirá cada una de las zonas activas en un mapa. Esta etiqueta estará siempre dentro de la etiqueta <MAP>.

```
<MAP NAME="MAPA1">  
<AREA SHAPE="..." COORDS="..." HREF="...">  
</MAP>
```

Los atributos de esta etiqueta son:

SHAPE

Permite especificar el tipo de región. Las regiones pueden ser rectangulares, circulares y poligonales. Los valores de este atributo son: RECT, CIRCLE y POLYGON.

COORDS

Define el valor de las coordenadas para las regiones activas, es decir los puntos que nos permitirán delimitar una región.

Regiones rectangulares

Para que una zona activa tenga forma rectangular debemos utilizar el atributo SHAPE de la etiqueta <AREA> con el valor RECT. Para indicar las coordenadas de esta zona debemos de utilizar las coordenadas XY de la esquina superior izquierda y las coordenadas XY de la esquina inferior derecha. Para indicar las coordenadas utilizamos el atributo COORDS de la etiqueta <AREA>

Sintaxis:

```
<AREA SHAPE="RECT" COORDS="X1, Y1, X2, Y2">
```

Las coordenadas se separan con comas y corresponden a la esquina superior izquierda (X1, Y1) e inferior derecha (X2, Y2) respectivamente.

Regiones circulares

Para que una zona activa tenga forma circular debemos utilizar el atributo SHAPE de la etiqueta <AREA> con el valor CIRCLE. Para indicar las coordenadas de esta zona debemos de utilizar las coordenadas XY del centro y el valor del radio R. Para indicar las coordenadas utilizamos el atributo COORDS de la etiqueta <AREA>.

Sintaxis:

```
<AREA SHAPE="CIRCLE" COORDS="X, Y, R">
```

Las coordenadas se separan con comas. Los dos primeros valores corresponden al centro del círculo (X, Y) y R corresponde a la longitud del radio del círculo. Este debe ser el orden para escribirlas.

Regiones poligonales

Para que una zona activa tenga forma poligonal debemos utilizar el atributo SHAPE de la etiqueta <AREA> con el valor POLYGON. Para indicar las coordenadas de esta zona debemos de utilizar las coordenadas XY de cada ángulo en el polígono. Para indicar las coordenadas utilizamos el atributo COORDS de la etiqueta <AREA>.

Sintaxis:

```
<AREA SHAPE="CIRCLE" COORDS="X1, Y1, X2, Y2, X3, Y3">
```

Las coordenadas se separan con comas. Para evitar errores debemos escribir las coordenadas de ángulos sucesivos.

HREF

Para que cada zona activa en el mapa sea un vínculo, debemos utilizar el atributo HREF de la etiqueta <AREA> tal y como se vio con la etiqueta <A>.

Sintaxis:

```
<AREA SHAPE="..." COORDS="..." HREF="INDEX.HTML" TARGET="_blank">
```

Los valores del atributo TARGET pueden ser revisados en la sección 3.

ALT

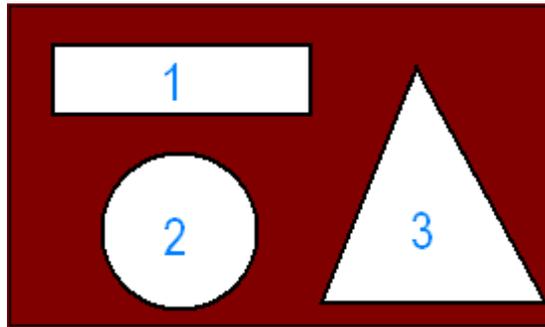
Para que cada zona activa en el mapa presente un texto alternativo, debemos utilizar el atributo ALT de la etiqueta <AREA> tal y como se vio con la etiqueta .

Sintaxis:

```
<AREA SHAPE="..." COORDS="..." HREF="..." TARGET="..." ALT="Texto alternativo">
```

9.5 Ejemplo

Este es un mapa que nos permite navegar en las secciones 1, 2 y 3 del tutorial. La imagen es realmente simple pero basta como ejemplo.



Este es el código:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Ejemplo de Mapas</TITLE>
```

```
</HEAD>
```

```
<BODY BGCOLOR="#FFFFFF">
```

```
<DIV align="center">
```

```
<IMG src="../../imagenes/mapa1.png" width="275" height="164" usemap="#indice" border="0">
```

```
<MAP name="indice">
```

```
<AREA shape="rect" coords="23,21,153,57" href="seccion-1.htm" alt="Sección 1">
```

```
<AREA shape="circle" coords="87,115,38" href="seccion-2.htm" alt="Sección 2">
```

```
<AREA shape="poly" coords="205,36,159,150,266,149" href="seccion-3.htm" alt="Sección 3 (Nueva Ventana)" target="_blank">
```

```
</MAP>
```

```
</DIV>
```

```
</BODY>
```

```
</HTML>
```

Fin de la Sección 9.

Sección 10. Frames

[10. 1 Introducción a los frames](#)

[10. 2 Etiqueta <FRAMESET>](#)

[10. 3 Etiqueta <FRAME>](#)

[10. 4 Frames anidados](#)

[10. 5 Etiqueta <IFRAME>](#)

[10. 6 Contenido Alternativo](#)

10. 1 Introducción a los frames

Frame significa marco y la creación de frames es una de las características mas notables del HTML. El uso de frames permite tener en una sola ventana varios documentos HTML abiertos, cada uno de ellos con sus propias características. Por lo tanto un frame no es otra cosa mas que una división de la ventana del navegador.

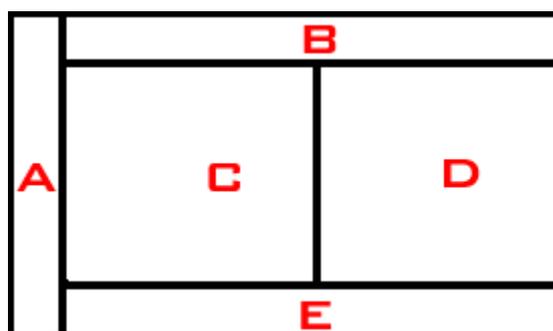
Antes de comenzar la explicación de cómo crear frames debemos de tener en cuenta los distintos elementos que conforman un documento con frames:

Documento "Maestro"

Es en el que se definirán cada una de las separaciones de la ventana del navegador. El documento maestro no tiene etiqueta <BODY> pues se remplaza con la etiqueta <FRAMESET> que se vera en la sección 10.2.

Documentos de los frames

Dependiendo de la cantidad de frames que hayamos creado, deberemos crear igual numero de documentos. Estos documentos aparecerán en cada una de las separaciones creadas. Ejemplo:



En la imagen el documento maestro especifica las separaciones que contendrán a los documentos de los frames: A, B, C, D y E.

Una consideración final: el vincular documentos desde los frames puede resultar catastrófico si no utilizamos el atributo TARGET de la etiqueta <A>. Se hablara de esto mas adelante.

10. 2 Etiqueta <FRAMESET>

En esta sección hablaremos de cómo crear el documento maestro.

La etiqueta <FRAMESET> es básica para la creación de páginas con frames. La etiqueta <FRAMESET> sustituirá a la etiqueta <BODY> en el documento de definición de frames o documento maestro. Como se vio anteriormente este documento maestro solo contiene la información necesaria para dividir la ventana del navegador en frames y no el contenido de tales. La etiqueta de cierre es </FRAMESET>.

Los atributos de esta etiqueta son:

COLS

Nos permite crear divisiones rectangulares en forma de columna.

Ejemplo:

```
<FRAMESET COLS="25%, 25%, 25%, 25%">
```

Resultado:



ROWS

Nos permite crear divisiones rectangulares en forma de renglón

Ejemplo:

```
<FRAMESET ROWS="25%, 25%, 25%, 25%">
```

Resultado:

25%
25%
25%
25%

BORDER

Este atributo especifica el tamaño del borde que divide los distintos documentos en píxeles, por lo tanto, si deseamos quitar el borde debemos dar a este atributo el valor cero.

Ejemplo:

```
<FRAMESET ROWS="25%, 25%, 25%, 25%" BORDER="0">
```

BORDERCOLOR

Nos permite especificar el color que tendrán los bordes.

Ejemplo:

```
<FRAMESET ROWS="25%, 25%, 25%, 25%" BORDERCOLOR="#FF6600">
```

Al ver los atributos COLS y ROWS, especificamos los tamaños de los frames en porcentaje, sin embargo existen otras dos formas:

Dimensiones Absolutas

Consiste en poner el valor en píxeles que tendrá la ventana:

Ejemplo:

```
<FRAMESET COLS="300,300">
```

 Divide la ventana en dos columnas de 300 píxeles de ancho cada una.

```
<FRAMESET ROWS="200,200">
```

 Divide la ventana en dos renglones de 200 píxeles de alto cada uno.

Solo puedo decir: cuidado con este tipo de dimensiones.

Dimensiones Relativas

Semejantes a las dimensiones por porcentaje

Ejemplo:

<FRAMESET COLS="4*,6*"> Divide la ventana en dos columnas de 40 y 60% respectivamente.
<FRAMESET ROWS="4*,6*"> Divide la ventana en dos renglones de 25% y 75% respectivamente.

También es posible combinar los distintos tipos de dimensiones:

<FRAMESET COLS="200,*"> Divide la ventana en dos columnas; una con 200 píxeles de ancho y la otra con el espacio restante.
<FRAMESET ROWS="50,*"> Divide la ventana en dos renglones; uno con 50 píxeles de alto y el otro con el espacio restante.
<FRAMESET COLS="25%,*"> Divide la ventana en dos columnas; una del 25% de tamaño y la otra con el espacio restante.
<FRAMESET ROWS="35%,*"> Divide la ventana en dos renglones; uno del 35% de tamaño y el otro con el espacio restante.

Hasta ahora hemos visto como dividir la pantalla, en la sección 10. 4 veremos como rellenar esos espacios con documentos HTML pero antes veremos como anidar frames.

10. 3 Etiqueta <FRAME>

La etiqueta <FRAME> nos permitirá especificar el documento que se encontrará dentro de un frame así como su comportamiento al estar contenido en el. La etiqueta de cierre es </FRAME>.

Los atributos de <FRAME> son:

SRC

Especifica el documento que estará dentro del frame.

Ejemplo:

```
<HTML>
<HEAD>
<TITLE>Frames</TITLE>
</HEAD>
<FRAMESET COLS="20%, 40%, 40%">
<FRAME SRC="doc1.htm">
<FRAME SRC="doc2.htm">
<FRAME SRC="doc3.htm">
</FRAMESET>
</HTML>
```

Tendremos 3 columnas; en la primera tendremos a doc1.htm, en la segunda a doc2.htm y en la tercera a doc3.htm.

NAME

Este atributo da un nombre a cada frame:

```
<HTML>
<HEAD>
<TITLE>Frames</TITLE>
</HEAD>
<FRAMESET COLS="20%, *">
<FRAME SRC="doc1.htm" name="col1">
<FRAME SRC="doc2.htm" name="col2">
</FRAMESET>
</HTML>
```

Tendremos 2 columnas; en la primera tendremos a doc1.htm y en la segunda a doc2.htm.

Si nosotros queremos que al hacer click en doc1.htm nos dirija a doc3.htm, pero con la condición de que este vinculo abra en el frame 2 (col2 según el nombre que le dimos) entonces en alguna parte de doc1 deberemos de tener algo similar a esto:

```
<A HREF="doc3.htm" TARGET="col2">...
```

SCROLLING

Nos permite indicar si se colocaran barras de desplazamiento dentro del frame. Los posibles valores para este atributo son: AUTO (el navegador determina si son necesarias), NO (no las coloca aunque sean necesarias) y YES (las coloca aunque no sean necesarias)

```
<FRAME SRC="doc1.htm" name="col3" SCROLLING="no">
```

MARGINWIDTH

Especifica el espacio que habrá entre el contenido del frame y sus márgenes derecho e izquierdo. El valor se especifica en píxeles y debe de ser mayor a cero.

```
<FRAME SRC="doc1.htm" MARGINWIDTH="20">
```

MARGINHEIGHT

Especifica el espacio que habrá entre el contenido del frame y sus márgenes superior e inferior. El valor se especifica en píxeles y debe de ser mayor a cero.

```
<FRAME SRC="doc1.htm" MARGINHEIGHT="10">
```

NORESIZE

Atributo booleano que evita que un frame sea redimensionado por el usuario.

```
<FRAME SRC="doc1.htm" name="col1" NORESIZE>
```

FRAMEBORDER

Los valores posibles para este atributo son 0 (cero) y 1. El primer valor indica al navegador que no debe de dibujar un borde entre el frame y otro adyacente. El borde se dibujara de todas formas si no tenemos cuidado en usar este atributo con los frames adyacentes. El valor 1 es el valor por default e indica que se dibujara el borde.

```
<FRAME SRC="doc1.htm" name="col1" FRAMEBORDER="0" NORESIZE >  
<FRAME SRC="doc2.htm" name="col2" FRAMEBORDER="0" NORESIZE >
```

Ejemplo de la sección

Este es el código del documento maestro (nombre = ej1_index.html):

```
<HTML>  
<HEAD>  
<TITLE>Documento Maestro</TITLE>  
</HEAD>  
<FRAMESET COLS="25%, *" BORDER="0">  
<FRAME SRC="ej1_frame1.html" NAME="indice" SCROLLING="YES" NORESIZE  
FRAMEBORDER="0" MARGINWIDTH="25" MARGINHEIGHT="10">  
<FRAME SRC="ej1_frame2a.html" NAME="principal" SCROLLING="NO" NORESIZE  
FRAMEBORDER="0" MARGINWIDTH="20" MARGINHEIGHT="100">  
</FRAMESET>  
</HTML>
```

Este es el código del frame 1 (nombre = ej1_frame1.html):

```
<HTML>  
<HEAD>  
<TITLE>Principal 1</TITLE>  
</HEAD>  
<BODY BGCOLOR="#FFFFFF">  
<H2 ALIGN="center">Frame 1<BR>  
(ej1_frame1.html)  
</H2>  
<HR ALIGN="center" WIDTH="90%" SIZE="4">  
<P><A HREF="ej1_frame2b.html" TARGET="principal">Abrir vinculo</A></P>  
</BODY>  
</HTML>
```

Este es el código del frame 2 (nombre = ej1_frame2a.html):

```
<HTML>  
<HEAD>  
<TITLE>Indice</TITLE>  
</HEAD>  
< BODY BGCOLOR="#FFFFFF">  
<H2 ALIGN="center">Frame 2<BR>
```

```
(ej1_frame2a.html)
</H2>
<HR ALIGN="center" WIDTH="90%" SIZE="4">
<P>Antes de abrir el vinculo</P>
</BODY>
</HTML>
```

Este es el código del frame 2, despues del click (nombre = ej1_frame2b.html):

```
<HTML>
<HEAD>
<TITLE>Principal 2</TITLE>
</HEAD>
<BODY BGCOLOR="#FFFFFF">
<H2 ALIGN="center">Frame 2<BR>
(ej1_frame2b.html)
</H2>
<HR ALIGN="center" WIDTH="90%" SIZE="4">
<P>Despues de seguir el v&iacute;nculo.</P>
<P><A HREF="ej1_frame2a.html" TARGET="_self">Regresar al documento anterior</A></P>
</BODY>
</HTML>
```

[Por último, este es el resultado.](#)



Espero que hayan notado el uso del valor `_self` para TARGET. Este tema se vio en la sección 3.

10. 4 Frames anidados

Supongamos que queremos crear algo similar a esto:



Como vemos este documento tiene dos columnas y en la segunda columna se tiene 3 renglones por lo que la forma de crearlo es la siguiente:

```
<FRAMESET COLS="25%, *">
```

```
<FRAME SRC="..." >
```

```
<FRAMESET ROWS="20%, 30%, *">
```

```
<FRAME SRC="..." >
```

```
<FRAME SRC="..." >
```

```
<FRAME SRC="..." >
```

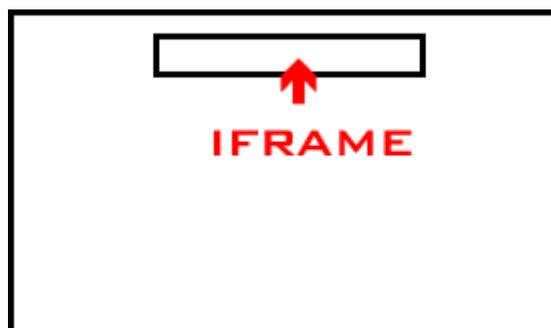
```
</FRAMESET>
```

```
</FRAMESET>
```

Simplemente se van creando las nuevas separaciones (dentro de separaciones) agregando mas etiquetas <FRAMESET> con sus respectivas etiquetas <FRAMES>. Esto es anidar frames.

10. 5 Etiqueta <IFRAME>

¿Qué pasaría si nosotros quisiéramos crear un frame de la siguiente forma?



La forma de lograrlo es utilizando la etiqueta <IFRAME>. Los atributos son casi los mismos que para <FRAME> exceptuando el atributo NORESIZE. Además podemos usar los siguientes atributos:

WIDTH

Especifica el ancho que tendrá nuestro frame.

HEIGHT

Especifica el alto que tendrá nuestro frame.

ALIGN

Permite alinear el frame respecto al documento. Los valores que acepta son LEFT (izquierda) y RIGHT (derecha). Debido a que el valor CENTER no centra el frame, es mejor alinear el frame utilizando cualquier otro método, por ejemplo:

```
<HTML>
<HEAD>
<TITLE>frames</TITLE>
</HEAD>
<BODY>
<DIV ALIGN="CENTER">
<IFRAME src="index.html" width="100" height="100">
Su navegador no soporta el uso de frames.
</IFRAME>
</DIV>
</BODY>
</HTML>
```

[Este es el resultado](#)



El texto dentro de <IFRAME> y </IFRAME> es el contenido que se mostrara si el navegador de nuestro visitante no soporta el uso de frames.

Otra forma de agregar documentos HTML dentro de otros es con la etiqueta <OBJECT>, tema que se discutirá en la sección siguiente.

10. 6 Contenido Alternativo

Como vimos en la sección anterior, el contenido alternativo para navegadores que no soportan el uso de <IFRAME> se especifica dentro de la misma etiqueta. Cuando usemos frames normales (no <IFRAME>) el contenido alternativo se debe de especificar con la etiqueta <NOFRAMES>:

```
<HTML>
<HEAD>
<TITLE>Frames</TITLE>
</HEAD>
<FRAMESET COLS="20%, 40%, 40%">
<FRAME SRC="doc1.htm">
<FRAME SRC="doc2.htm">
<FRAME SRC="doc3.htm">
</FRAMESET>
<NOFRAMES>
<H1>Esta es la versión sin frames del documento ....</H1>
</NOFRAMES>
</HTML>
```

(Por fin el) Fin de la Sección 10.

Sección 11. Scripts, Applets y demás I

[11. 1 Introducción](#)

[11. 2 Cliente vs. Servidor](#)

[11. 3 JavaScript](#)

[11. 4 VBScript](#)

[11. 5 Java](#)

[11. 6 PHP](#)

[11. 7 ASP](#)

[11. 8 CGI](#)

[11. 9 Mapas del lado del servidor](#)

[11. 10 Scripts](#)

[11. 11 Etiqueta <SCRIPT>](#)

[11. 12 Eventos](#)

[11. 13 Etiqueta <NOSCRIPT>](#)

11. 1 Introducción

Al navegar por Internet nos podemos dar cuenta de la cantidad de tecnologías disponibles que tienen como único fin la presentación de información. Si somos asiduos navegantes de Internet, palabras como Java, JavaScript, PHP, ASP, CGI, Plug-in, etc, nos son comunes e intuitivamente tenemos conocimiento de su utilidad. El único fin de esta sección es presentar estas tecnologías de

una forma sencilla, evitando entrar en demasiados detalles; al fin de cuentas este es un tutorial de HTML y CSS.

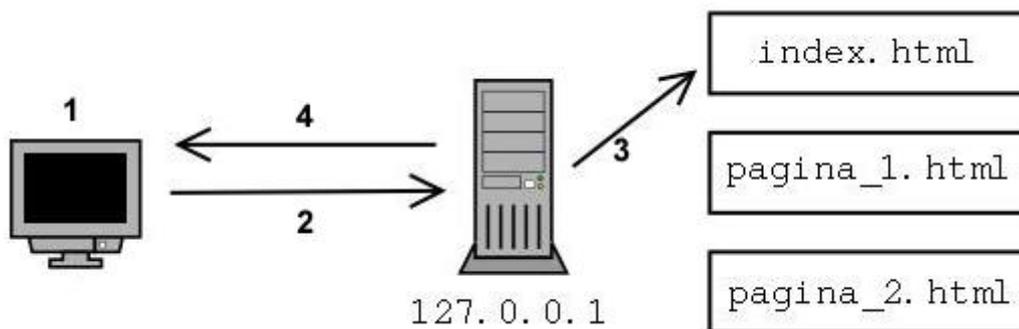
Estas tecnologías nos permitirán crear sitios dinámicos. Con ellas podremos manejar bases de datos, presentar información personalizada a cada visitante, crear un efecto visual que con HTML sería imposible de hacer o cualquier otra cosa que se nos ocurra.

11. 2 Cliente vs. Servidor

En realidad no es una pelea, mas bien es una relación de pareja. Antes que nada debemos de dar una definición para cada uno de los términos.

Servidor

Al oír esta palabra lo primero que pensamos es en una maquina extremadamente potente conectada a una red distribuyendo páginas web o cualquier otra cosa que nos imaginemos. Puede ser, pero en realidad a lo que nos queremos referir como servidor es a un programa que se dedica a escuchar. Veamos el caso de un servidor web de una manera extremadamente sencilla:



1. Una persona en la computadora de su casa (que puede ser el mismo equipo en el que se encuentra el servidor) llama a la página `http://127.0.0.1/index.html`
2. El navegador manda la petición al servidor que está en 127.0.0.1
3. El servidor recibe la petición y regresa la página `index.html` al navegador del usuario
4. El navegador se encarga de ir traduciendo el código de `index.html` y presentándolo en su ventana

En pocas palabras, el servidor es un programa que se dedica a escuchar posibles peticiones y, si encuentra alguna, regresa el documento requerido.

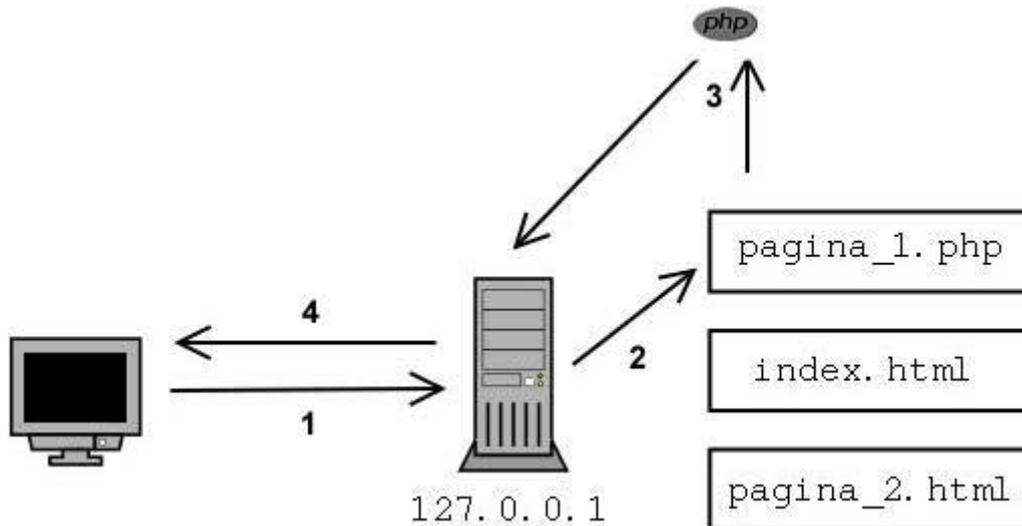
Cliente

Como imaginan, en este caso, el cliente es el navegador. Es el encargado de realizar las peticiones al servidor.

Esta sección en general trata de pequeños programas que podremos incluir en nuestras páginas HTML. Sin embargo existe una diferencia que nos permitirá clasificar nuestros pequeños programas.

Programas del lado del servidor

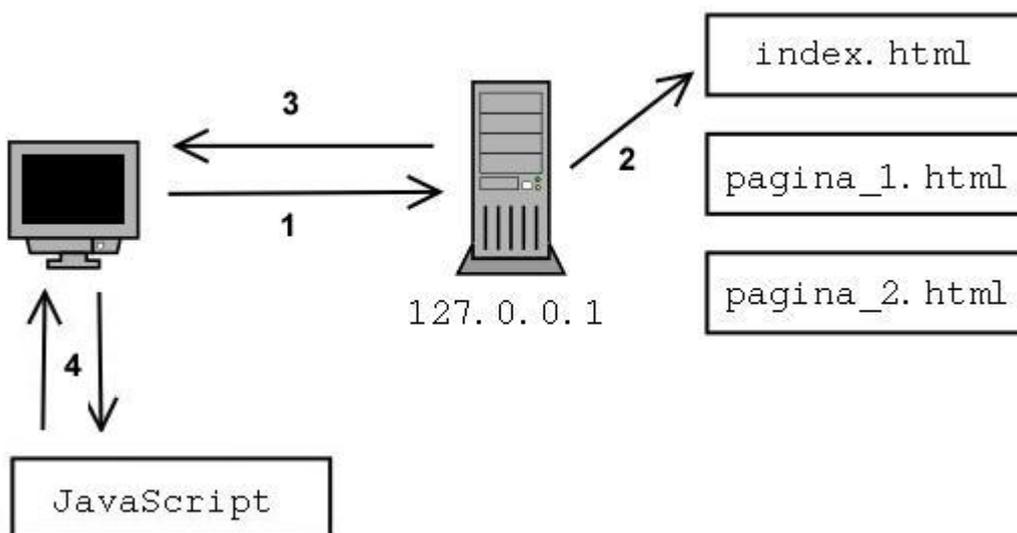
Son aquellos que son ejecutados en el servidor después de la petición pero antes de regresar el documento al cliente.



1. El cliente hace una petición al servidor.
2. El servidor recibe la petición y "observa" que hay que ejecutar cierto programa.
3. Ejecuta el programa y envía los resultados junto con el documento requerido.
4. El navegador interpreta los datos que recibe y despliega el documento al usuario.

Programas del lado del cliente

Son aquellos que son ejecutados una vez que son recibidos por el navegador.



1. El cliente hace una petición al servidor
2. El servidor regresa el documento requerido al cliente.
3. El cliente comienza a presentar el documento y encuentra que hay un código adicional

4. Si conoce como tratarlo lo ejecuta.
5. Si no conoce como tratarlo lo trata como texto sin formato

Por ultimo, las explicaciones aquí dadas han sido simplificadas en extremo pues este puede ser un proceso mas complicado.

11.3 JavaScript

Desarrollador: Netscape

URL: <http://www.netscape.net>

Es un lenguaje que nos permite escribir scripts que se ejecutaran del lado del cliente o del servidor. En ambos casos se incrusta en las páginas HTML para agregar funcionalidad e interacción. Es muy común que la gente confunda Java y JavaScript pero nosotros no debemos de caer en ese error. Mas adelante se hablara de Java y pronto las cosas se aclararan aun mas.

Como mencionamos, JavaScript funciona del lado del cliente y del servidor. En el caso del cliente, es el navegador el que se encarga de interpretar el código e irlo ejecutando según vaya apareciendo; por otra parte, cuando funciona del lado del servidor, las páginas HTML son compiladas en archivos ejecutables. Cuando un documento es requerido, el servidor ejecuta dichos archivos para generar las páginas HTML dinámicamente.

Para mas información, referirse a la [Documentación Oficial de JavaScript](#).

Ahora pasemos a algunos pequeños trucos que nos pueden servir al diseñar nuestra página, como siempre, todo con medida.

Hola Mundo

```
<HTML>
<HEAD>
<TITLE>Hola Mundo</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<INPUT TYPE="button" VALUE="¿Hola?" onClick="abrir_dialogo('Hola Mundo!')">
</BODY>
</HTML>
```

Resultado:

Abrir Ventana de navegador

```
<HTML>
<HEAD>
<TITLE>Abrir Ventana</TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
function abrir_ventana(url,nombre,caract) {
window.open(url,nombre,caract);
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<INPUT TYPE="button" VALUE="Abrir Ventana" NAME="Button"
onClick="abrir_ventana('../prop/eriol_slashdot.jpg', 'ventanita', 'toolbar=yes, location=yes,
status=yes, menubar=yes, scrollbars=yes, resizable=yes, width=330, height=60')">
</BODY>
</HTML>
```

Resultado

Notas

toolbar: indica si habrá o no barra de herramientas.

location: indica si habrá o no barra de dirección.

status: indica si habrá o no barra de estado.

scrollbars: indica si habrá o no barras de desplazamiento.

Recursos

<http://javascript.internet.com/>

<http://javasite.com/>

<http://www.org.org.mx/manuales/javascript/>

11. 4 VBScript

Desarrollador: Microsoft

URL: <http://www.microsoft.com>

Microsoft, en su continuo intento de reinventar la rueda, creo el VBScript como una alternativa a JavaScript de Netscape. Es parte de Visual Basic pero la diferencia estriba en que VBScript necesita un navegador para poder funcionar.

Ejemplo:

```
<HTML>
<HEAD>
```

```
<TITLE>Hola Mundo</TITLE>
<SCRIPT TYPE="text/vbscript">
<!--
Sub dialogo(cadena)
Alert cadena
End Sub
'-->
</SCRIPT>
</HEAD>

<BODY>
<INPUT TYPE="button" VALUE="¿Hola?" onClick="dialogo('Hola Mundo!')">
</BODY>
</HTML>
```

Resultado:

Recursos

<http://www.intranetjournal.com/corner/wrox/progref/vbt/>

11. 5 Java

Desarrollador: Sun Microsystems

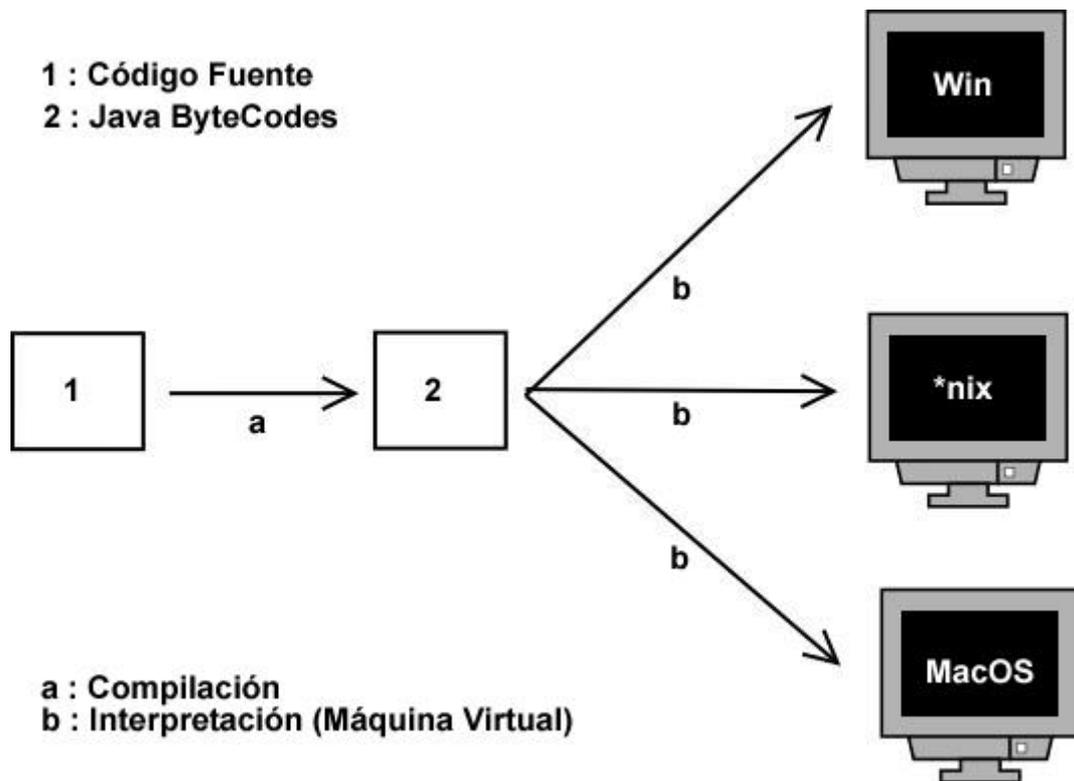
URL: <http://www.sun.com> | <http://java.sun.com>

Java es un lenguaje de programación orientado a objetos, realmente muy parecido a C++ aunque probablemente un poco mas sencillo.

La característica mas peculiar de Java es su capacidad de ejecutarse en cualquier plataforma (Windows, *nix, Mac) lo que lo hace perfecto para aplicaciones de Internet. Un buen ejemplo de esto es el [Mensajero de Yahoo](#).

Una vez que se tiene el código fuente de un programa en Java, este se compila lo que hace que el código se traduce a un lenguaje intermedio llamado "Java bytecodes" que es en si es la parte independiente a la plataforma. Cuando el programa es requerido, estos "bytecodes" son interpretados por la Máquina Virtual de Java y presentados. Cabe hacer notar que Java puede funcionar del lado del cliente (applets en el caso del navegador) o del lado del servidor (servlets).

Sobre como utilizar Applets de Java en HTML tratara la sección 11b.



Recursos

<http://developer.java.sun.com/developer/>

11. 6 PHP

Desarrollador: Apache Software Foundation

URL: <http://www.apache.org> | <http://www.php.net>

PHP son las siglas de una versión anterior llamada "Personal Home Page Tools". Es un lenguaje de programación de scripts que trabaja del lado del servidor. Para aquellos que conocen algo de programación, se puede decir que es una mezcla de Perl y C. Las versiones mas comunes son la 3 y la 4.

Últimamente, PHP, ha ganado muchos adeptos debido a la facilidad que ofrece para la construcción de sitios web dinámicos como por ejemplo, sitios de compras en línea.

PHP se incrusta dentro de las páginas HTML, lo que nos permite una edición sencilla, sin necesidad de compilación:

```

<HTML>
<HEAD>
<TITLE>Ejemplo con PHP</TITLE>
</HEAD>
<BODY>
<?php
  
```

```
$cadena = "Hola Mundo";  
echo $cadena;  
?>  
</BODY>  
</HTML>
```

En este ejemplo lo único que obtenemos es "Hola Mundo".

Por ultimo, en vez de llamar a nuestros archivos .HTML o .HTM los deberemos de llamar .PHP.

Recursos

<http://www.devshed.com>
<http://www.webmonkey.com>
<http://www.phpwizard.com>
<http://www.phpnuke.com>

11. 7 ASP

Desarrollador: Microsoft

URL: <http://www.microsoft.com>

ASP son las siglas para Active Server Pages. Es un lenguaje de programación de scripts del lado del servidor que permite combinar scripts, HTML y componentes Active X para la creación de sitios dinámicos. Por ultimo, ASP y PHP son muy parecidos en cuanto a las ventajas que podemos obtener de ellos sobre CGI.

Un buen artículo sobre ASP puede encontrarse en [Maestros del Web](#)

Recursos

<http://msdn.microsoft.com/workshop/server/>

11. 8 CGI

CGI son las siglas para Common Gateway Interface. En realidad no es un lenguaje de programación pues un CGI puede estar escrito en lenguajes tales como C/C++, Perl, Bash, e incluso Java. Si no es un lenguaje de programación... ¿entonces que es? CGI es una parte del Protocolo de Transferencia de Hipertexto (HTTP). Es un método para pasar datos entre el servidor y la aplicación. Obviamente se maneja del lado del servidor.

Veamos un ejemplo sencillo en Perl.

```
#!/usr/local/bin/perl  
# hola.pl  
print "Content-type: text/html\n\n";  
print "Hola Mundo!\n";
```

Este archivo, hola.pl, debe de estar en el directorio llamado CGI-BIN de nuestro servidor.

CGI es muy usado debido a que nos puede servir para el procesamiento de formas, por ejemplo, de seguro alguna vez hemos visto algo así:

```
<FORM ACTION="procesa.cgi" METHOD="POST">
```

Recursos

<http://www.perl.com/reference/query.cgi?section=cgi&x=21&y=6>

<http://www.perlarchive.com/>

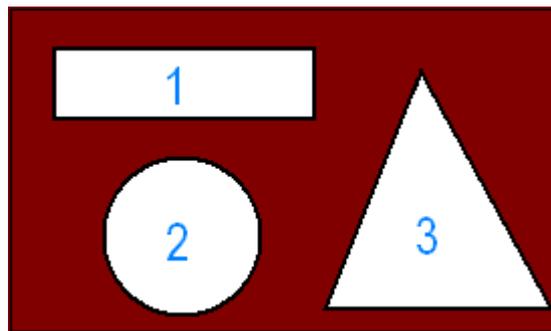
11. 9 Mapas del lado del servidor

En realidad esta sección no trata de cómo hacer un mapa del lado del servidor, si no mas bien de la parte concerniente al HTML. ¿Porque decidí poner este tema en esta sección y no en la de mapas? Porque hasta esta sección se han tratado algunos temas importantes que no se habían tratado en la sección de mapas.

En realidad lo único que se tiene que explicar es el atributo booleano ISMAP de la etiqueta . Cuando esta presente indica que la imagen será tratada como un mapa del lado del servidor.

```
<A HREF="http://www.misitio.com/cgi-bin/busca_coords">  
<IMG SRC="mapa1.png" ISMAP>  
</A>
```

Es necesario que se encuentre dentro de <A> para que todo esto funcione.



Podemos ver en la barra de estado como son capturadas las coordenadas del puntero del ratón sobre el mapa, algo similar a:

<http://www.avantel.net/~trieb/?120,130>

Al hacer click sobre el mapa, las coordenadas son enviadas al servidor quien busca en este caso el cgi busca_coords que se encargara de mostrar la página vinculada a esas coordenadas.

Un buen tutorial sobre mapas del lado del servidor puede ser encontrado en esta dirección:
<http://lightsphere.com/dev/ismaps/>

11. 10 Scripts

Hasta el momento hemos utilizado la palabra script sin dar una definición formal de ella. A continuación se da una descripción de script vista desde el punto de vista de HTML.

Script: Es una secuencia de instrucciones lógicas que acompañan a un documento HTML.

Los scripts nos permitirán evaluar un documento y modificarlo dinámicamente , procesar un formulario, validarlo, etc.

A pesar de la facilidad de uso, los lenguajes de scripts son muy limitados en sus capacidades y mas lentos que un programa compilado pues necesitan un programa que vaya interpretando cada una de las instrucciones.

Los scripts pueden acompañar a un documento HTML de dos formas, ya sea dentro del código HTML o en un archivo adicional. Además los scripts pueden ser ejecutados en el momento en que el documento HTML es descargado o cada vez que un comportamiento específico ocurre.

Hemos visto que lenguajes como JavaScript, VBScript o PHP nos permiten escribir scripts y las diferencias entre ellos son las características propias del lenguaje, la complejidad y si corren del lado del cliente o del lado del servidor. En las siguientes secciones hablaremos de los lenguajes de scripts que corren del lado del cliente tales como JavaScript y VBScript, aunque no son los únicos.

11. 11 Etiqueta <SCRIPT>

Esta etiqueta nos permite incrustar un script dentro de un documento HTML. Su etiqueta de cierre es </SCRIPT>.

Los atributos de esta etiqueta son:

SRC

Nos permite especificar un nombre de archivo de script externo.

LANGUAGE

Permite especificar el lenguaje de scripts a utilizar:

```
<SCRIPT LANGUAGE="JavaScript">  
<SCRIPT LANGUAGE="VBScript">
```

Debido a que no hay estándares para estos valores, es mejor utilizar el atributo TYPE para indicar el lenguaje a utilizar.

TYPE

Similar a la anterior. Esta es la forma recomendada

```
<SCRIPT TYPE="text/javascript">  
<SCRIPT TYPE="text/vbscript">
```

DEFER

DEFER es un atributo booleano que permite especificar al navegador que el script no generara contenido en el documento.

Como sabemos, mucha gente utiliza navegadores antiguos y estos no tienen la capacidad de entender los scripts por lo tanto resulta útil esconder los contenidos del script de la siguiente forma:

```
<SCRIPT LANGUAGE="JavaScript">  
<!--  
[...]  
!-->  
</SCRIPT>
```

```
<SCRIPT type="text/vbscript">  
<!--  
[...]  
'-->  
</SCRIPT>
```

En el primer ejemplo se coloca // antes de --> para esconder esa línea a Javascript. En el segundo ejemplo se utiliza ' para obtener el mismo resultado. Estos son los caracteres de comentario de JavaScript y VBScript respectivamente.

11. 12 Eventos

En ejemplos anteriores se vio una especie de atributo que nos permitía buscar un comportamiento (evento) del usuario hacia la página (i.e. hacer click a un botón de la forma). A este tipo de atributos se les conoce como controladores de eventos. Regresemos a uno de los ejemplos de JavaScript:

```
[...]  
<BODY>  
<INPUT TYPE="button" VALUE="¿Hola?" onclick="abrir_dialogo('Hola Mundo!')">  
</BODY>  
[...]
```

En este caso onclick es un manejador de eventos. Cuando se hace click se llama a la función abrir_dialogo, definida anteriormente.

Los controladores de eventos son:

onload

Este evento ocurre cuando el navegador termina de cargar el documento o, en su caso, todos los frames. Puede ser utilizado con <BODY> o <FRAMESET>.

Ejemplo

```
<HTML>
<HEAD>
<TITLE>onload</TITLE>
<SCRIPT TYPE="text/javascript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
</SCRIPT>
</HEAD>
<BODY onload="abrir_dialogo('Hola Mundo')">
<CENTER>
<H1>Ejemplo de onload</H1>
<HR width="90%">
</CENTER>
</BODY>
</HTML>
```

[Resultado](#)

onunload

Este evento ocurre cuando el navegador pasa a otro documento, justo en el momento en que inicia a cargar el nuevo. Puede ser utilizado con <BODY> o <FRAMESET>.

Ejemplo

```
<HTML>
<HEAD>
<TITLE>onunload</TITLE>
<SCRIPT TYPE="text/javascript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
</SCRIPT>
</HEAD>
<BODY onunload="abrir_dialogo('Hola Mundo!')">
<CENTER>
<H1>Ejemplo de onunload</H1>
```

```
<HR width="90%">
<A HREF="http://www.yahoo.com">Seguir este vinculo para ver resultado</A>
</CENTER>
</BODY>
</HTML>
```

Resultado

onclick

Este evento ocurre cuando se da click sobre algún elemento. Puede ser utilizado con casi todas las etiquetas.

Ejemplo

```
<HTML>
<HEAD>
<TITLE>onclick</TITLE>
<SCRIPT TYPE="text/javascript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<H1>Ejemplo de onclick</H1>
<HR width="90%">
<INPUT TYPE="checkbox" onclick="abrir_dialogo('Hola Mundo!')">
Dar click a la casilla de verificaci&oacute;n para ver ejemplo.
</CENTER>
</BODY>
</HTML>
```

Resultado

ondblclick

Este evento ocurre cuando se da click sobre algún elemento. Puede ser utilizado con casi todas las etiquetas.

Ejemplo

```
<HTML>
<HEAD>
<TITLE>ondblclick</TITLE>
```

```
<SCRIPT TYPE="text/javascript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<H1>Ejemplo de ondblclick</H1>
<HR width="90%">
<INPUT TYPE="radio" ondblclick="abrir_dialogo('Hola Mundo!')">
Dar doble click
</CENTER>
</BODY>
</HTML>
```

[Resultado](#)

onmousedown

Este evento es similar a onclick.

Ejemplo

```
<HTML>
<HEAD>
<TITLE>onmousedown</TITLE>
<SCRIPT TYPE="text/javascript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<H1>Ejemplo de onmousedown</H1>
<HR WIDTH="90%">
<INPUT TYPE="radio" onmousedown="abrir_dialogo('Hola Mundo!')">
Dar click
</CENTER>
</BODY>
</HTML>
```

[Resultado](#)

onmousemove

Este evento ocurre mientras el ratón este en movimiento.

Ejemplo

```
<HTML>
<HEAD>
<TITLE>onmousemove</TITLE>
<SCRIPT TYPE="text/javascript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
</SCRIPT>
</HEAD>
<BODY >
<CENTER>
<H1>Ejemplo de onmousemove</H1>
<HR WIDTH="90%">
<P>Cada vez que se ponga el raton en el area de texto saldra un mensaje. Para
que ya no salga pongan el raton fuera el area de texto y den &quot;Enter&quot;
para cerrar el dialogo.</P>
<P>
<TEXTAREA cols="40" rows="20" onmousemove="abrir_dialogo('Hola
mundo!')"></TEXTAREA>
</P>
</CENTER>
</BODY>
</HTML>
```

[Resultado](#)

onmouseout

Este evento ocurre cuando el cursor del ratón se aleja de algún elemento.

Ejemplo

```
<HTML>
<HEAD>
<TITLE>onmouseout</TITLE>
<SCRIPT TYPE="text/javascript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
```

```

</SCRIPT>
</HEAD>
<BODY >
<CENTER>
<H1>Ejemplo de onmouseout</H1>
<HR WIDTH="90%">
<P>Cada vez que se ponga el raton en el area de texto y despues lo pongan fuera
de el saldra un mensaje. Para que ya no salga pongan el raton fuera el area
de texto y den &quot;Enter&quot; para cerrar el dialogo.</P>
<P>
<TEXTAREA cols="40" rows="20" onmouseout="abrir_dialogo('Hola mundo!')"></TEXTAREA>
</P>
</CENTER>
</BODY>
</HTML>

```

Resultado

onmouseover

Este evento ocurre cuando el cursor se coloca sobre algún elemento.

Ejemplo

```

<HTML>
<HEAD>
<TITLE>onmouseover</TITLE>
<SCRIPT TYPE="text/javascript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<H1>Ejemplo de onmouseover</H1>
<HR WIDTH="90%">
<A HREF="http://www.yahoo.com" onmouseover="abrir_dialogo('Hola Mundo!')">Pongan su
ratón sobre este vínculo</A>
</CENTER>
</BODY>
</HTML>

```

Resultado

onmouseup

Este evento ocurre cuando el botón del ratón es soltado sobre algún elemento.

Ejemplo

```
<HTML>
<HEAD>
<TITLE>onmouseup</TITLE>
<SCRIPT TYPE="text/javascript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
</SCRIPT>
</HEAD>
<BODY >
<CENTER>
<H1>Ejemplo de onmouseup</H1>
<HR WIDTH="90%">
<P>Den click en el cuadro 1 y sin soltar el boton del mouse vayan al 2, ahi
suelten el boton</P>
<P>
<TEXTAREA>1</TEXTAREA>
<TEXTAREA onmouseup="abrir_dialogo('Hola Mundo')">2</TEXTAREA>
</CENTER>
</BODY>
</HTML>
```

Resultado

onfocus

Este evento ocurre cuando algún elemento esta seleccionado, es decir, está enfocado. Puede ser utilizado con <A>, <INPUT>, <SELECT>, <TEXTAREA> y <BUTTON>.

Ejemplo

```
<HTML>
<HEAD>
<TITLE>onfocus</TITLE>
<SCRIPT TYPE="text/javascript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<H1>Ejemplo de onfocus</H1>
```

```
<HR WIDTH="90%">
<TEXTAREA name="nombre" onfocus="abrir_dialogo('Hola Mundo!')">Dame Click! (asi gano
enfoco)</TEXTAREA>
</CENTER>
</BODY>
</HTML>
```

Resultado

onblur

Este evento ocurre cuando algún elemento pierde el enfoque. Puede ser utilizado con las mismas etiquetas que onfocus.

Ejemplo

```
<HTML>
<HEAD>
<TITLE>onblur</TITLE>
<SCRIPT TYPE="text/javascript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<H1>Ejemplo de onblur</H1>
<HR WIDTH="90%">
<TEXTAREA name="nombre" onblur="abrir_dialogo('Hola Mundo!')" rows="5">Dame Click y
despues da click en cualquier otro lugar (asi pierdo el enfoque)</TEXTAREA>
</CENTER>
</BODY>
</HTML>
```

Resultado

onkeydown

Este evento ocurre cuando se presiona una tecla sobre un elemento.

Ejemplo

```
<HTML>
<HEAD>
<TITLE>onkeydown</TITLE>
```

```

<SCRIPT TYPE="text/javascript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<H1>Ejemplo de onkeydown</H1>
<HR WIDTH="90%">
<P>Escribir aqu&iacute;;
<TEXTAREA name="nombre" onkeydown="abrir_dialogo('Hola Mundo!')"></TEXTAREA>
</CENTER>
</BODY>
</HTML>

```

Resultado

onkeypress

Este evento ocurre cuando se presiona y suelta una tecla sobre un elemento.

Ejemplo

```

<HTML>
<HEAD>
<TITLE>onkeypress</TITLE>
<SCRIPT TYPE="text/javascript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<H1>Ejemplo de onkeypress</H1>
<HR WIDTH="90%">
<P>Escribir aqu&iacute;;
<TEXTAREA name="nombre" onkeypress="abrir_dialogo('Hola Mundo!')"></TEXTAREA>
</CENTER>
</BODY>
</HTML>

```

Resultado

onkeyup

Este evento ocurre cuando se suelta una tecla sobre un elemento.

Ejemplo

```
<HTML>
<HEAD>
<TITLE>onkeyup</TITLE>
<SCRIPT TYPE="text/javascript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<H1>Ejemplo de onkeyup</H1>
<HR WIDTH="90%">
<P>Escribir aqu&iacute;:
<TEXTAREA name="nombre" onkeyup="abrir_dialogo('Hola Mundo!')"></TEXTAREA>
</CENTER>
</BODY>
</HTML>
```

Resultado

onsubmit

Este evento ocurre cuando se presiona el botón de envío de un formulario. Solo puede ser utilizado con <FORM>.

Ejemplo

```
<HTML>
<HEAD>
<TITLE>onsubmit</TITLE>
<SCRIPT TYPE="text/javascript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<H1>Ejemplo de onsubmit</H1>
```

```

<HR WIDTH="90%">
<FORM METHOD="post" ACTION="mailto:triebjl@netscape.net" ENCTYPE="text/plain"
onsubmit="abrir_dialogo('Esta completamente seguro?')">
<P>Nombre:
<INPUT TYPE="text" NAME="nombre">
<P>Comentarios sobre el tutorial:<BR>
<TEXTAREA NAME="comentarios"></TEXTAREA>
<P>
<INPUT TYPE="submit" VALUE="Enviar">
</FORM>
</CENTER>
</BODY>
</HTML>

```

Resultado

onreset

Este evento ocurre cuando se presiona el botón de borrar de un formulario. Solo puede ser utilizado con <FORM>.

Ejemplo

```

<HTML>
<HEAD>
<TITLE>onreset</TITLE>
<SCRIPT TYPE="text/javascript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<H1>Ejemplo de onreset</H1>
<HR width="90%">
<FORM METHOD="post" ACTION="mailto:triebjl@netscape.net" ENCTYPE="text/plain"
onreset="abrir_dialogo('Ponga mas atencion la proxima vez!')">
<P>Nombre:
<INPUT TYPE="text" NAME="nombre">
<P>Comentarios sobre el tutorial:<BR>
<TEXTAREA NAME="comentarios"></TEXTAREA>
<P>
<INPUT TYPE="submit" VALUE="Enviar">
<INPUT TYPE="reset" VALUE="Borrar">
</P>
</FORM>
<P>&nbsp;</P>
</CENTER>

```

```
</BODY>
</HTML>
```

[Resultado](#)

onselect

Este evento ocurre cuando se selecciona texto dentro de un campo de texto.

Ejemplo

```
<HTML>
<HEAD>
<TITLE>onselect</TITLE>
<SCRIPT TYPE="text/javascript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<H1>Ejemplo de onselect</H1>
<HR WIDTH="90%">
<TEXTAREA name="nombre" onselect="abrir_dialogo('Hola
Mundo!')">SELECCIONAME</TEXTAREA>
</CENTER>
</BODY>
</HTML>
```

[Resultado](#)

onchange

Este evento ocurre cuando un elemento (como campos de texto) pierden el enfoque y su valor se modifico después de que fueron enfocados.

Ejemplo

```
<HTML>
<HEAD>
<TITLE>onchange</TITLE>
<SCRIPT TYPE="text/javascript">
<!--
function abrir_dialogo(cadena) {
alert(cadena);
}
//-->
```

```
</SCRIPT>
</HEAD>
<BODY>
<CENTER>
<H1>Ejemplo de onchange</H1>
<HR WIDTH="90%">
<TEXTAREA name="nombre" onchange="abrir_dialogo('Hola
Mundo!')">MODIFICAME</TEXTAREA>
</CENTER>
</BODY>
</HTML>
```

[Resultado](#)

11. 13 Etiqueta <NOSCRIPT>

Esta etiqueta nos permite proveer un contenido alternativo en el caso que el navegador no pueda ejecutar el script. La etiqueta de cierre es </NOSCRIPT>.

```
<SCRIPT type="text/javascript">
<--
[...]
// -->
</SCRIPT>
<NOSCRIPT>
<P>Su navegador no acepta JavaScript
</NOSCRIPT>
```

Fin de la Sección 11.

Sección 12. Introducción a las hojas de estilo

[12. 1 ¿Que son las hojas de estilo?](#)

[12. 2 CSS y XSL](#)

[12. 3 Reglas](#)

Después de planear nuestro sitio web, de haber escogido los colores, creado las imágenes y tener la idea general de como queremos ver ese sitio, comienzan los problemas. Veamos algunos de esos problemas que tienen como consecuencia el detrimento de la calidad de nuestro sitio:

Imágenes:

Alguna vez tendremos la necesidad de tener una pequeña imagen en la parte inferior de nuestra pagina. ¿Como hacemos para que, sin importar lo que haga el visitante, se quede ahí? Es sabido por

ejemplo que el navegador IExplorer permite ver en pantalla completa por lo que nuestra pagina se alarga, así que todas las imágenes en la parte inferior ya no estarán en el fin de la pagina.

Texto:

¿Porque en HTML solo existen 7 tamaños de letra si en nuestra computadora podemos especificar un tamaño en píxeles (px) como se nos antoje? ¿Que pasa si yo necesito una letra mas grande de la que me puede ofrecer la etiqueta <H1>?

¿Que pasaría si yo pusiera un texto en negritas pero quisiera poner un texto que resaltara mas? Definitivamente hay mas problemas y mencionarlos no serviría de mucho pues solo nos recordarían aquellos momentos en que los sufrimos en carne propia; lo que buscamos son soluciones.

12. 1 ¿Que son las hojas de estilo?

Las hojas de estilo sirven precisamente para definir la forma en que se presentaran documentos en la pantalla. Nada mas y nada menos. Aunque esta definición suene un tanto burda, al comenzar a usarlas sentiremos como se abre un nuevo mundo de infinitas posibilidades... y si aun siguen un tanto escépticos sigan leyendo este tutorial.

12. 2 CSS y XSL

Existen dos lenguajes para las hojas de estilo que son el CSS y el XSL. ¿Cual es la diferencia? CSS puede ser usado con HTML y XSL no. Este es un tutorial de CSS por lo que no abordaremos el XSL, pero se puede encontrar bastante información sobre XSL en [The World Wide Web Consortium](#).

12. 3 Reglas

Aquí es en donde comenzamos el aprendizaje; no hay porque asustarse pues CSS es un lenguaje extremadamente sencillo. A continuación se muestra el código de una pagina a la que se aplico un estilo y seria una buena idea el que abriéramos notepad y escribiéramos lo siguiente:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Hoja de Estilo</TITLE>
```

```
</HEAD>
```

```
<STYLE TYPE="text/css">
```

```
P { font-size: 30 px; font-family: arial; background: #FF6600 }
```

```
</STYLE>
```

```
<BODY>
```

```
<P>Hola Mundo.</P>
```

```
<BODY>
```

```
</HTML>
```

[Este es el resultado.](#)



Hola Mundo.

Realmente fácil ¿no? ¿Como habrías creado el fondo naranja con solo HTML? Esta bien, pudiste hacer una tabla pero ¿No tardarías mas creando todo ese código? Esa es una de las ventajas principales de CSS, hacer mas en menos tiempo. Desentrañar este código no implica demasiados problemas, pues solo vemos dos elementos nuevos: La etiqueta `<STYLE>` que se vera en un capitulo posterior y una regla de CSS:

```
P { font-size: 30 px; font-family: arial; background: #FF6600 }
```

Una regla de CSS consiste de dos partes que son el selector y la declaración.

Un selector no es mas que una etiqueta HTML sin encerrar entre `<` y `>`.

En el ejemplo el selector es `P` Una declaración ira encerrada entre `{` y `}`. Consta de dos partes que son la propiedad (`font-size`, `font-family` y `background` en este ejemplo) y el valor (`30 px`, `arial` y `#FF6600` respectivamente en el ejemplo).

Cabe mencionar que existen unas 50 propiedades que permitirán especificar la presentación de un documento. Este tutorial fue creado para abarcar la mayor parte de ellas así como facilitar la navegación entre ellas. Ahora que ya sabemos como escribir CSS, pasemos a la forma de incluirlo en el código HTML.

Fin de la Sección 12.

Sección 13. Agregando Estilos

[13. 1 Agregando en HTML](#)

[13. 2 Links a hojas de estilo externas](#)

- [13. 3 Importar hojas de estilo](#)
- [13. 4 Estilos de línea en HTML](#)
- [13. 5 Agrupamiento](#)
- [13. 6 Clases](#)
- [13. 7 Selectores contextuales](#)
- [13. 8 Comentarios](#)
- [13. 9 Ejemplo](#)

13. 1 Agregando en HTML

Este método fue el utilizado en el ejemplo del capítulo anterior, en donde utilizamos la etiqueta `<STYLE>` y su atributo `TYPE` con el valor `text/css`. Este método resulta práctico para agregar estilos cuando estos son únicos para cada página. Es recomendable usar la etiqueta de comentario para evitar que navegadores que no soporten CSS hagan caso omiso del código, tal y como se utiliza en JavaScript.

```
<STYLE TYPE="text/css">
<!--
[...]
-->
</STYLE>
```

Nota: el símbolo [...] no forma parte del código, solo indica que el código continúa.

Veamos un documento típico en HTML:

```
<HTML>
<HEAD>
<TITLE>Mi página</TITLE>
</HEAD>
<BODY>
<H1 ALIGN="center">Bienvenidos todos a mi página</H1>
<HR>
<B>
<FONT SIZE="3" COLOR="#3366CC" FACE="Arial">
<P>Un párrafo cualquiera .
<P>Otro párrafo cualquiera.
</FONT>
</B>
</BODY>
</HTML>
```

Ahora utilicemos CSS dentro de HTML:

```
<HTML>
<HEAD>
<TITLE>Mi página</TITLE>
<STYLE type="text/css">
P { color: #3366CC; font-family: Arial; font-weight: bold;}
H1 { text-align: center;}
</STYLE>
```

```
</HEAD>
<BODY>
<H1>Bienvenidos todos a mi página</H1>
<HR>
<P>Un párrafo cualquiera .
<P>Otro párrafo cualquiera.
</BODY>
</HTML>
```

Así, cada vez que coloquemos una etiqueta <P> o una <H1> estaremos utilizando los estilos definidos y nos evitaremos poner código adicional.

En este ejemplo los selectores son P y H1; las propiedades son color, font-family y font-weight.

13. 2 Links a hojas de estilo externas

El problema con el método anterior es que si deseamos cambiar los estilos, tendremos que ir cambiando todas las definiciones en cada una de las hojas. Una forma de evitar esto es crear un documento de estilos y aplicarlo a todas las paginas de nuestro sitio. Es realmente sencillo y altamente recomendable pues al cambiar este documento CSS los cambios se verán en cada uno de los documentos HTML a los que se aplico este método. Siguiendo con el ejemplo anterior:

```
<HTML>
<HEAD>
<TITLE>Mi página</TITLE>
<LINK REL="stylesheet" TYPE="text/css" HREF="estilos.css">
</HEAD>
<BODY>
<H1>Bienvenidos todos a mi página</H1>
<HR>
<P>Un párrafo cualquiera .
<P>Otro párrafo cualquiera.
</BODY>
</HTML>
```

El vinculo es al archivo estilos.css el cual se ve de la siguiente forma:

```
P { color: #3366CC; font-family: Arial; font-weight: bold}
H1 { text-align: center}
```

13. 3 Importar hojas de estilo

El problema con el método anterior es que no puede ser utilizado junto con otros métodos. Para evitar esto utilizaremos el método de importar, que es casi igual al anterior:

```
<HTML>
<HEAD>
<TITLE>Mi página</TITLE>
```

```

<STYLE type="text/css">
<!--
@import "estilos.css";

I { font-family: "Times New Roman"; font-style: italic; font-weight: lighter}

-->
</STYLE>
</HEAD>
<BODY>
<H1>Bienvenidos todos a mi página</H1>
<HR>
<P>Un párrafo cualquiera .
<P>Otro párrafo cualquiera.

</BODY>
</HTML>

```

Así agregamos los estilos definidos para P y H1 del ejemplo anterior y definimos un estilo para I dentro del código HTML.

13. 4 Estilos de línea en HTML

Este método consiste en agregar estilos dentro del código HTML de la siguiente manera:

```

<HTML>
<HEAD>
<TITLE>Mi página</TITLE>
</HEAD>
<BODY>
<H1 STYLE="text-align: center">Bienvenidos todos a mi página</H1>
<HR>
</BODY>
</HTML>

```

Lo único que estamos haciendo es redefinir una etiqueta HTML por lo que si queremos utilizar otra vez ese estilo tendremos que definir la etiqueta cuando sea necesario.

Hemos visto 4 métodos de agregar estilos a documentos HTML, cada uno tiene desventajas y ventajas sobre el otro así que el decidir usar uno u otro es decisión del webmaster :-)

13. 5 Agrupamiento

Para reducir el tamaño de las hojas de estilo es posible agrupar varios selectores separándolos por medio de una coma:

```
H1, H2, H3 { font-color: blue }
```

Nota: también las declaraciones pueden ser agrupadas.

13. 6 Clases

Sabemos que toda etiqueta de HTML puede ser utilizada como selector. Sin embargo nosotros podemos crear tres distintos tipos de clases para un selector de la siguiente manera (con cualquiera de los 4 métodos vistos anteriormente):

```
H1.uno { color: #FF6600 }
H1.dos {color: #FF6666 }
H1.tres {color:#FF66FF }
```

La manera de utilizar estas tres clases de H1 se muestra a continuación:

```
<H1 CLASS=uno>Primer encabezado H1.</H1>
<H1 CLASS=dos>Segundo encabezado H1.</H1>
<H1 CLASS=tres>Tercer encabezado H1.</H1>
```

El nombre de las clases puede ser cualquiera que se nos venga a la mente. Estas clases fueron creadas específicamente para el selector H1 pero si quisiéramos utilizar alguna de estas clases para otro selector debemos omitir, precisamente, cualquier selector:

```
*.uno { color: #FF6600 }
```

o simplemente:

```
.uno { color: #FF6600 }
```

Por ultimo, cualquier elemento dentro de BODY puede ser vinculado a una clase.

```
<H1 CLASS="uno">...
<P CLASS="uno">...
<I CLASS="uno">...
<DIV CLASS="uno">...
```

13. 7 Selectores contextuales

Para explicar mejor este tema, ¿que mejor que un ejemplo?

```
<HTML>
[... ]
<STYLE TYPE="text/css">
<!--
P { color: #000000 }
I { color: #FF0000 }
-->
</STYLE>
[... ]
```

¿Que sucederá con este código? Todo el texto en cursivas será rojo sin importar si esta entre P o no. Si nosotros quisiéramos que solo el texto en cursivas dentro de P sea rojo debemos utilizar la siguiente sintaxis:

```
P I { color: #FF0000 }
```

Esto indicara al navegador que solo el texto en cursivas dentro de P será de color rojo.

Así en el siguiente código:

```
<HTML>
<HEAD>
<TITLE>Tutorial</TITLE>
<STYLE TYPE="text/css">
<!--
P I { color: #FF0000 }
H1 { color: #000000 }
-->
</STYLE>
<BODY>
<H1><I>Bienvenido</I></H1>
<P>Este es el tutorial de <I>CSS</I>
</BODY>
</HTML>
```

El encabezado de nivel uno aparecerá en cursivas y de color negro; por otra parte, CSS aparecerá en color rojo y en cursivas.

Nota: es común caer en el error de separar los selectores por medio de comas, sintaxis que sirve para agrupar distintos selectores, tema que se vio anteriormente.

13. 8 Comentarios

Por ultimo, para indicar un comentario existe la siguiente sintaxis similar a la de C :

```
P { color: #FF0000 } /*este es un comentario */
```

13. 9 Ejemplo

A continuación se muestra el contenido del archivo estilos.css utilizado en las distintas secciones del tutorial:

```
.normal { font-family: Arial, Helvetica, sans-serif; font-size: 10pt}
```

```
.definicion { font-family: Arial, Helvetica, sans-serif; font-size: 10pt; font-style: italic}
```

```
.etiqueta { font-family: "Courier New", Courier, mono; font-size: 10pt; font-weight: bold}
```

```
.titulos1 { font-family: Arial, Helvetica, sans-serif; font-size: 14pt; text-decoration:  
underline; text-align: center}
```

```
.titulos2 { font-family: Arial, Helvetica, sans-serif; font-size: 12pt; color: #006699; font-weight:  
bold}
```

```
.titulos3 { font-family: Arial, Helvetica, sans-serif; font-size: 12pt; text-decoration: underline}
```

La forma en que es agregada esta hoja de estilos, así como el resultado, puede ser observada en el código de las distintas secciones del tutorial.

Fin de la sección 13

Sección 14 Agregando Estilos II

[14. 1 Árbol de documento](#)

[14. 2 Herencia](#)

[14. 3 Cascada](#)

[14. 4 Reglas de orden de cascada](#)

[14. 5 Selectores hijos](#)

[14. 6 Selectores por atributos](#)

[14. 7 Selectores por ID](#)

[14. 8 Seudo clases](#)

[14. 9 Medidas](#)

[14. 10 Colores](#)

[14. 11 Problemas y navegadores](#)

14. 1 Árbol de documento

Hasta ahora, hemos trabajado con la parte de fácil entendimiento de CSS. Es ahora, en esta sección, que ahondaremos más en los principios que necesitan una explicación un poco más detallada.

Para empezar, regresaremos un poco al tema de HTML con el siguiente ejemplo:

```
<HTML>  
<HEAD>  
<TITLE>Mi página</TITLE>  
</HEAD>  
<BODY>  
<H1>Bienvenidos todos a mi página</H1>  
<HR>  
<P>Un parrafo cualquiera .
```

```
<P>Otro párrafo cualquiera.  
<UL>  
<LI>Elemento 1 de la lista 1</LI>  
<LI>Elemento 2 de la lista 1</LI>  
<LI>Elemento 3 de la lista 1</LI>  
</UL>  
<OL>  
<LI>Elemento 1 de la lista 2</LI>  
<LI>Elemento 2 de la lista 2</LI>  
<LI>Elemento 3 de la lista 2</LI>  
</OL>  
</BODY>  
</HTML>
```

Por supuesto que este código no representa ningún problema pues con solo mirarlo una vez podemos describir el comportamiento que obtendremos en el navegador. Sin embargo de lo que se trata aquí no es sobre dicho comportamiento si no mas bien sobre la estructura del documento. Un documento HTML puede describirse como un árbol.

Árbol: Grafo no orientado conexo sin ciclo. Esta compuestos por nodos o módulos conectados por líneas llamadas ramas.

En un árbol se tiene un nodo denominado raíz u origen y que carece de un antecesor. Cualquier otro nodo tiene un antecesor único en el nivel anterior y un sucesor o mas, incluso ninguno en el nivel siguiente. Ejemplo de árbol: un sistema de archivos.

Ahora veamos otras definiciones, derivadas de la anterior, que nos serán utiles mas adelante:

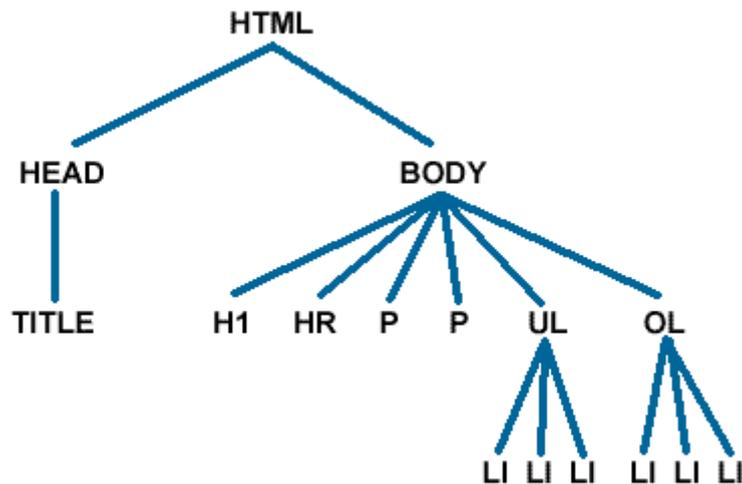
Hijo: Un elemento A se considera hijo de B si y solo si B es el padre de A (el nodo en el nivel inmediatamente anterior a A).

Descendiente: Un elemento A se considera descendiente de B si

- a) A es hijo de B
- b) A es hijo de C que es descendiente de B.

Antecesor: Un elemento A se considera ancestro de B si y solo si B es un descendiente de A.

Y para los que no entendieron nada, aquí esta el árbol del documento HTML del ejemplo.



En este caso, la raíz es HTML y sus hijos son HEAD y BODY. HEAD tiene un hijo que es TITLE y BODY tiene a H1, HR, P, P, UL y OL como hijos que a su vez son descendientes de HTML.

14.2 Herencia

¿Recuerdan el uso de selectores contextuales? Si no, aquí está el mismo ejemplo de ese tema:

```

<HTML>
<HEAD>
<TITLE>Tutorial</TITLE>
<STYLE TYPE="text/css">
P I { color: #FF0000 }
H1 { color: #000000 }
</STYLE>
<BODY>
<H1><I>Bienvenido</I></H1>
<P>Este es el tutorial de <I>CSS</I>
</BODY>
</HTML>
  
```

Nosotros habíamos dicho que "CSS" aparecería en cursivas y de color rojo si y solo si en nuestro código la etiqueta <I> aparecía dentro de <P>. Ahora podemos decir sin equivocarnos que I es un hijo de P. Es por eso que a los selectores contextuales también se les conoce como selectores descendientes. Debido a que pueden existir varios descendientes, declaraciones del tipo:

```

P * I
BODY * I *
  
```

son validas.

¿Qué pasa con el siguiente código?

```
<HTML>
<HEAD>
<TITLE>Tutorial</TITLE>
<STYLE TYPE="text/css">
P { color: #FF0000 }
</STYLE>
<BODY>
<P>Este es el tutorial de <U>CSS</U>
</BODY>
</HTML>
```

El texto se mostrara en color rojo y "CSS" aparecerá del mismo color pero subrayado. ¿Cómo sabe el navegador que "CSS" debe ser de color rojo?

La respuesta es sencilla: Herencia. A través de la herencia, cada antecesor transmite sus propiedades a todos sus descendientes; en el ejemplo, U es hijo de P por lo cual U lleva implícita la propiedad color:#FF0000. Si tenemos:

```
BODY { color: #FF0000 }
```

Entonces todos los descendientes de BODY tendrán implícita la propiedad color:#FF0000.

14.3 Cascada

Existen distintas fuentes de donde pueden provenir las hojas de estilo:

Autor: El creador de la pagina. Sabemos que el autor puede utilizar hojas de estilo, ya sea vinculándolas, importándolas o incluyéndolas dentro del código HTML.

Usuario: Nosotros, como simples usuarios también podemos especificar un estilo para un documento en particular.

Agente de Usuario (Agent User): En pocas palabras, se refiere al programa con el que estemos visualizando (por ejemplo el navegador). Los navegadores toman en cuenta las especificaciones de CSS para poder representar el código fuente de forma visual, es decir, especifican el significado de los selectores y propiedades.

Esto es a lo que conocemos como cascada. Estas fuentes trabajan conjuntamente para la aplicación de estilos. Sin embargo una fuente de autor tiene mas peso que una fuente de usuario y esta a su vez tiene mas peso que una fuente de agente de usuario.

14.4 Reglas de orden de cascada

Observemos el siguiente código

```
<HTML>
<HEAD>
```

```
<TITLE>Alguna Página</TITLE>
<STYLE TYPE="text/css">
@import url (estilos.css);
H1 { color: #FF0000 }
</STYLE>
</HEAD>
<BODY>
<H1 STYLE="color: #00FF00">
[... ]
</HTML>
```

Como vemos, se esta importando el archivo "estilos.css" con el siguiente código:

```
H1 { color: #0000FF }
```

Sabiendo esto ¿De que color serán los encabezados de nivel uno?

Debido a que los tres métodos usados aplican un estilo distinto a H1, no sabríamos cual seria el resultado final a menos de que existiera una jerarquía para los distintos métodos de aplicar los estilos. Dicha jerarquía existe.

La jerarquía, en orden de importancia, es la siguiente

1. Estilos de línea en HTML.
2. Estilos Agregados en HTML.
3. Links a hojas de estilo externas.
4. Hojas de estilo importadas.
5. Estilos de usuario.
6. Estilos predeterminados del agente de usuario.

Ahora sabemos que el encabezado de nivel uno será color verde.

Por ultimo si tenemos algo similar a esto:

```
H1 { color: #FF0000 }
H1 { color: #00FF00 }
```

el encabezado de nivel 1 será de color verde pues la ultima declaración será la que se tomara en cuenta.

14. 5 Selectores hijos

Un selector hijo se escribe de la forma:

```
S1 > S2 {...}
```

donde S2 es hijo de S1. Esto permite aplicar estilos solo a aquellos elementos hijos de S1.

Veamos un ejemplo:

UL > LI { font-size: 1cm }

En este ejemplo se aplicara el estilo a todos los elementos LI que sean hijos de UL. Así, en el siguiente ejemplo:

```
<HTML>
<HEAD>
<TITLE>Mi página</TITLE>
<STYLE TYPE="text/css">
P { color: #3366CC; font-family: Arial; font-weight: bold}
H1 { text-align: center}
UL > LI { font-size: 1cm }
</STYLE>
</HEAD>
<BODY>
<H1>Bienvenidos todos a mi página</H1>
<HR>
<P>Un párrafo cualquiera .
<P>Otro párrafo cualquiera.
<UL>
<LI>Elemento 1 de la lista 1</LI>
<LI>Elemento 2 de la lista 1</LI>
<LI>Elemento 3 de la lista 1</LI>
</UL>
<OL>
<LI>Elemento 1 de la lista 2</LI>
<LI>Elemento 2 de la lista 2</LI>
<LI>Elemento 3 de la lista 2</LI>
</OL>
</BODY>
</HTML>
```

Solo los elementos LI en una lista sin orden mostraran el estilo.

14. 6 Selectores por atributos

Supongamos que nosotros queremos aplicar un estilo a P pero solo si tiene el atributo ALIGN. Para esto nos sirven los selectores por atributos.

Existen 4 formas de utilizar este tipo de selectores de las cuales solo veremos las 2 primeras.

1. [ATRIBUTO]

Aplica el estilo solo si el elemento contiene el atributo especificado:

P[ALIGN] { font-size: 1cm } : Aplica el estilo a P pero solo si tiene el atributo ALIGN.

*[ALIGN] { font-size: 1cm } : Aplica el estilo a cualquier elemento que contenga el atributo ALIGN.

P[ALIGN][SIZE]: Aplica el estilo a P pero solo si tiene los atributos ALIGN y SIZE.

2. [ATRIBUTO = VALOR]

Aplica el estilo solo si el elemento contiene el atributo y el valor especificados:

P[ALIGN=CENTER] { font-size: 1cm } : Aplica el estilo a P pero solo si tiene el atributo ALIGN y el valor CENTER.

*[ALIGN=CENTER] { font-size: 1cm } : Aplica el estilo a cualquier elemento que contenga el atributo ALIGN y el valor CENTER.

P[ALIGN=CENTER][SIZE]: Aplica el estilo a P pero solo si tiene los atributos SIZE y ALIGN y el valor CENTER para ALIGN.

Hasta ahora hemos utilizado los atributos vistos en este tutorial, sin embargo el siguiente código también es válido:

```
<HTML>
<HEAD>
<TITLE>Mi página</TITLE>
<STYLE TYPE="text/css">
*[NOMBRE=ANA] { color: #006699; font-family: Arial; font-weight: bold font-size: 4pt }
P[NOMBRE=PEPE] { color: #ff0000; font-family: Arial; font-weight: bold }
P[NOMBRE=JORGE] { color: #00ff00; font-family: Arial; font-weight: bold }
P[NOMBRE=JUAN] { color: #0000ff; font-family: Arial; font-weight: bold }
</STYLE>
</HEAD>
<BODY>
<H1 NOMBRE="ANA">Ana</H1>
<P NOMBRE="PEPE">Pepe
<P NOMBRE="JORGE">Jorge
<P NOMBRE="JUAN">Juan
</BODY>
</HTML>
```

14. 7 Selectores por ID

En el ejemplo anterior se utilizó un atributo creado por nosotros llamado NOMBRE. Sin embargo, a todos los atributos pudimos darle el valor ANA o JORGE. Esto no nos puede servir para identificar distintos elementos... pero el atributo ID sí pues, ningún atributo puede tener el mismo valor para ID que otro. Ejemplo

```
<HTML>
<HEAD>
<TITLE>Mi pagina</TITLE>
</HEAD>
<BODY>
<P ID="NOMBRE">Nombre.
<P ID="EDAD">Edad.
</BODY>
</HTML>
```

Otro uso que podemos dar a los identificadores es, como todos imaginan de seguro, la aplicación de estilos, esto se hace de la forma:

```
SELECTOR#ID { ... }
```

El siguiente ejemplo aplica el estilo a P con ID=nombre

```
<HTML>
<HEAD>
<TITLE>Mi pagina</TITLE>
<STYLE type="text/css">
P#NOMBRE { color: #FF0000 }
</STYLE>
</HEAD>
<BODY>
<P ID="NOMBRE">Nombre.
<P ID="EDAD">Edad.
</BODY>
</HTML>
```

Y el siguiente ejemplo aplica el estilo a cualquier selector con ID=nombre

```
<HTML>
<HEAD>
<TITLE>Mi pagina</TITLE>
<STYLE TYPE="text/css">
*#NOMBRE { color: #FF0000 }
</STYLE>
</HEAD>
<BODY>
<P ID=NOMBRE>Nombre.
<P ID=EDAD>Edad.
</BODY>
</HTML>
```

14. 8 Seudo clases

Las seudo clases nos permiten clasificar elementos por características distintas a sus atributos, contenido o nombre. La parte interesante de estas seudo clases es que pueden ser dinámicas, es

decir, un elemento puede adquirir o perder una pseudo clase según el visitante interactúe con el documento.

:first-child

Nos permite aplicar estilos al primer hijo de un elemento. En pocas palabras, el primer elemento después del elemento padre.

```
<HTML>
<HEAD>
<TITLE>Mi página</TITLE>
<STYLE TYPE="text/css">
P { color: #3366CC; font-family: Arial; font-weight: bold}
H1 { text-align: center}
UL > LI:first-child { font-size: 1cm }
</STYLE>
</HEAD>
<BODY>
<H1>Bienvenidos todos a mi página</H1>
<HR>
<P>Un párrafo cualquiera .
<P>Otro párrafo cualquiera.
<UL>
<LI>Elemento 1 de la lista 1</LI>
<LI>Elemento 2 de la lista 1</LI>
<LI>Elemento 3 de la lista 1</LI>
</UL>
<OL>
<LI>Elemento 1 de la lista 2</LI>
<LI>Elemento 2 de la lista 2</LI>
<LI>Elemento 3 de la lista 2</LI>
</OL>
</BODY>
</HTML>
```

Cuando visitamos una página hecha por todo un "caballero webmaster", podemos observar que los vínculos no visitados se ven [así](#) y los visitados se ven [así](#). Las siguientes 2 pseudo clases nos permitirán jugar con esto.

:link

Aplica el estilo a un vínculo no visitado.

:visited

Aplica el estilo a un vínculo visitado.

Ejemplo:

```
<HTML>
<HEAD>
<TITLE>Mi página</TITLE>
```

```

<STYLE TYPE="text/css">
A:LINK { color: #FFFFFF; background: #006699 }
A:VISITED { color: #FF0000; background #FFFFFF }
</STYLE>
</HEAD>
<P><A href="http://www.avantel.net/~trieb">Trieb </A>
</BODY>
</HTML>

```

Ahora veamos las pseudo clases dinámicas:

:active

Nos sirve para aplicar un estilo cuando un elemento esta activo:

```

<HTML>
<HEAD>
<TITLE>Mi página</TITLE>
<STYLE TYPE="text/css">
A:ACTIVE { color: #FFFFFF; background: #006699 }
</STYLE>
</HEAD>
<P><A href="http://www.avantel.net/trieb">Trieb </A>
</BODY>
</HTML>

```

:hover

A diferencia de la anterior pseudo clase, :hover nos permitirá cambiar estilos cuando el elemento no esta activo pero si seleccionado, por ejemplo, cuando pasamos el ratón por encima del vinculo:

```

<HTML>
<HEAD>
<TITLE>Mi pagina</TITLE>
<STYLE TYPE="text/css">
A:hover { color: #FFFFFF; background: #006699 }
P#NOMBRE { color: #FF0000 }
</STYLE>
</HEAD>
<BODY>
<P id="trieb"><A href="http://www.avantel.net/~trieb">trieb</A>
</BODY>
</HTML>

```

14. 9 Medidas

Al estudiar HTML encontramos dos formas de medición, por porcentaje y por píxeles. En CSS encontramos mas formas de medición:

- centímetros (cm)

- milímetros (mm)
- pulgadas (in): 1 in = 2.54 cm
- puntos (pt): 1pt = 1 / 72 in
- picas (pc): 1 pc = 12 pt

Dependiendo de nuestras necesidades será el tipo de medida a utilizar. Ejemplo:

```
H1 { font-size: 5cm }
H1 { font-size: 5mm }
H1 { font-size: 5in }
H1 { font-size: 5pt }
H1 { font-size: 5pc }
```

En cuanto al uso de porcentajes, estos porcentajes son relativos a otro valor por ejemplo:

```
BODY { font-size: 5cm }
BODY H1 { font-size: 90% }
```

El tamaño de la fuente será el 90% de 5 cm.

14. 10 Colores

Al igual que en HTML, en CSS podemos usar el formato hexagesimal extendido, es decir #RRVVAA. Además de este formato, también podemos usar los siguientes:

Formato Hexagesimal Compacto

Similar al anterior pero de la forma:

#RVA

Notación Funcional Entera:

Este formato es de la forma:

rgb(RRR, VVV, AAA)

los valores RRR (cantidad de rojo), VVV (cantidad de verde) y AAA (cantidad de azul), son valores enteros en el rango 0 - 255.

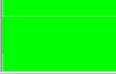
Notación Funcional Porcentual:

Este formato es de la forma:

rgb(RRR%, VVV%, AAA%)

los valores RRR (por ciento de rojo), VVV (por ciento de verde) y AAA (por ciento de azul), son valores enteros en el rango 0 - 100%

La siguiente tabla muestra algunos colores escritos en distintos formatos:

Color	Hexagesimal Extendido	Hexagesimal Compacto	Funcional Entera	Funcional Porcentual	Ejemplo
Rojo	FF0000	F00	255, 0, 0	100%, 0%, 0%	
Azul	0000FF	00F	0, 0, 255	0%, 0%, 100%	
Verde	00FF00	0F0	0, 255, 0	0%, 100%, 0%	
Blanco	FFFFFF	FFF	255, 255, 255	100%, 100%, 100%	
Negro	000000	000	0,0,0	0%, 0%, 0%	
Amarillo	FFFF00	FF0	255, 255, 0	100%, 100%, 0%	

14. 11 Problemas y navegadores

Tal vez mientras estamos aprendiendo CSS, hemos observado que partes de lo aquí descrito no funciona. Esto se debe a que algunos navegadores (no voy a decir cual para que no se enojen) no respetan las especificaciones para las hojas de estilo. Por lo tanto, cuando apliquemos estilos debemos de ver que funciona y que no, que navegador funciona y cual no.

Si han llegado hasta aquí, felicidades, ya saben CSS en un 90%. Lo que sigue ahora es aprender mas propiedades que de las pocas que hemos estado usando. Podemos decir que de aquí en adelante lo demas es de bajada.

Fin de la sección 14

Sección 15. Propiedades de las fuentes

[15. 1 font-family](#)

[15. 2 font-size](#)

[15. 3 font-style](#)

[15. 4 font-variant](#)

[15. 5 font-weight](#)

[15. 6 font](#)

15. 1 font-family

En HTML para indicar la fuente que se deberá aplicar en nuestro texto debemos utilizar algo similar a:

```
<P><FONT FACE="ARIAL">Algún texto</FONT>
```

Para indicar el tipo de fuente cuando usamos hojas de estilo utilizaremos la propiedad font-family de la siguiente manera:

```
P { font-family: Arial }
```

Recordemos que en HTML se pueden utilizar varias fuentes que serán utilizadas por el navegador en el caso de que alguna de ellas no se encuentre en la maquina del visitante:

```
<P><FONT FACE="ARIAL, COURIER">Algún texto</FONT></P>
```

Utilizando la propiedad font-family:

```
P { font-family: Arial, Courier }
```

Cabe recordar que los nombres de las fuentes varían de una plataforma a otra. Para evitar que los nombres de las fuentes resulten confusos a la maquina de nuestro visitante, existe la posibilidad de llamar a las fuentes por nombres genéricos:

```
P { font-family: Serif }
```

Esto no significa que debamos omitir el nombre de la fuente que nosotros queramos utilizar, pero se recomienda que uno de los nombres de las fuentes sea genérico:

```
P { font-family: Arial, Sans-serif }
```

En el ejemplo anterior se utiliza el nombre genérico Sans-serif cuyo resultado podría ser Arial. Los nombres genéricos aceptados junto con un ejemplo de una fuente de esa familia son:

Serif (Times)

Sans-serif (Arial)

Cursive (Comic sans)

Fantasy (Western)

Monospace (Courier)

¿Que pasa si el nombre de la fuente que queremos utilizar tiene mas de una palabra? La solución consiste en encerrar en comillas el nombre de esa fuente:

```
P { font family: Arial, "Times New Roman", sans-serif }
```

¿Que pasa si deseamos utilizar un nombre largo dentro de la etiqueta <STYLE>? He aquí la solución:

```
<P STYLE="font-family: arial, 'cualquier fuente', sans-serif">
```

Nótese el uso de ' para encerrar el nombre de la fuente.

15. 2 font-size

Los tamaños de las fuentes pueden ser especificados de cuatro formas:

Tamaño absoluto

Acepta los valores xx-small, x-small, small, medium, large, x-large, xx-large.

Ejemplo:

```
<P STYLE="font-size: xx-small">xx-small</P>
```

```
<P STYLE="font-size: x-small">x-small</P>
```

```
<P STYLE="font-size: small">small</P>
```

```
<P STYLE="font-size: medium">medium</P>
```

```
<P STYLE="font-size: large">large</P>
```

```
<P STYLE="font-size: x-large">x-large</P>
```

```
<P STYLE="font-size: xx-large">xx-large</P>
```

Resultado:

xx-small ... x-small ... small ... medium ... large ... X-large ... **XX-**
large...

Tamaño relativo

Acepta los valores larger y smaller.

Ejemplo

```
<P STYLE="font-size: larger">larger</P>
```

```
<P STYLE="font-size: smaller">smaller</P>
```

Resultado:

larger... smaller...

Tamaño por medida

Es la forma en que habíamos venido trabajando.

Ejemplo:

Usando Píxeles como unidad:

```
<P STYLE="font-size: 10px">10 px</P>
```

```
<P STYLE="font-size: 20px">20 px</P>
```

```
<P STYLE="font-size: 30px">30 px</P>
```

```
<P STYLE="font-size: 40px">40 px</P>
```

```
<P STYLE="font-size: 50px">50 px</P>
```

10px ... 20px ... 30px ... 40px ... 50px...

Usando centímetros como unidad:

```
<P STYLE="font-size: 1cm">1 cm</P>
```

```
<P STYLE="font-size: 2cm">2 cm</P>
```

```
<P STYLE="font-size: 3cm">3 cm</P>
```

```
<P STYLE="font-size: 4cm">4 cm</P>
```

```
<P STYLE="font-size: 5cm">5 cm</P>
```

1 cm...2 cm...3 cm...

4 cm...

5 cm...

Usando milímetros como unidad:

<P STYLE="font-size: 1mm">1 mm</P>

<P STYLE="font-size: 2mm">2 mm</P>

<P STYLE="font-size: 3mm">3 mm</P>

<P STYLE="font-size: 4mm">4 mm</P>

<P STYLE="font-size: 5mm">5 mm</P>

1mm... 2mm... 3mm... 4mm... 5mm...

Usando pulgadas como unidad:

<P STYLE="font-size: 0.5in">0.5 in</P>

<P STYLE="font-size: 1in">1 in</P>

<P STYLE="font-size: 1.5in">1.5 in</P>

0.5in... 1in...

1.5in...

Usando puntos como unidad:

```
<P STYLE="font-size: 10pt">10 pt</P>  
<P STYLE="font-size: 20pt">20 pt</P>  
<P STYLE="font-size: 30pt">30 pt</P>  
<P STYLE="font-size: 40pt">40 pt</P>  
<P STYLE="font-size: 50pt">50 pt</P>
```

10pt... 20pt... 30pt... 40pt... 50pt...

Usando picas como unidad:

```
<P STYLE="font-size: 0.5pc">0.5 pc</P>  
<P STYLE="font-size: 1pc">1 pc</P>  
<P STYLE="font-size: 1.5pc">1.5 pc</P>  
<P STYLE="font-size: 2pc">2 pc</P>  
<P STYLE="font-size: 2.5pc">2.5 pc</P>
```

0.5pc... 1pc... 1.5pc... 2pc... 2.5pc...

Tamaño por porcentaje

La otra forma que sabemos:

```
<HTML>  
<HEAD>  
<TITLE>Ejemplo</TITLE>  
<STYLE>  
BODY { font-size: 50px }  
</STYLE>
```

```
</HEAD>
<BODY>
<P STYLE="font-size: 100%">100%</P>
<P STYLE="font-size: 80%">80%</P>
<P STYLE="font-size: 60%">60%</P>
<P STYLE="font-size: 40%">40%</P>
<P STYLE="font-size: 20%">20%</P>
</BODY>
</HTML>
```

En este ejemplo se especifica el tamaño de letra para BODY, así, se calcula el 20, 40, 60, 80 y 100 por ciento de 50px. Esto se debe a que BODY es el padre de P y este tiene el valor implícito de 50 px.

15. 3 font-style

Esta propiedad nos permitirá mostrar el texto en cursivas. Los valores para esta propiedad son:

normal

No muestra la fuente en cursivas.

italic

Muestra la fuente en cursivas.

oblique

Similar a italic.

Ejemplo:

```
<P STYLE="font-style: normal">Ejemplo de normal</P>
<P STYLE="font-style: italic">Ejemplo de italic</P>
<P STYLE="font-style: oblique">Ejemplo de oblique</P>
```

Resultado:

Ejemplo de normal

Ejemplo de italic

Ejemplo de oblique

15. 4 font-variant

Los valores que acepta esta propiedad son normal y small-caps. Al utilizar el valor small-caps veremos el texto al que se le aplico el estilo en mayúsculas pero de tamaño similar a las minúsculas.

Ejemplo

```
<HTML>
<HEAD>
<TITLE>Ejemplo</TITLE>
<STYLE type="text/css">
P I { font-style: normal; font-variant: small-caps }
</STYLE>
</HEAD>
<P>MAYUSCULAS ... <I>small-caps ... </I>minúsculas ... <I>small-caps</I>
</BODY>
</HTML>
```

15.5 font-weight

Los valores permitidos para esta propiedad son:

100, 200, 300, 400, 500, 600, 700, 800, 900

Los valores son crecientes, por ejemplo, un valor de 400 se vera mas destacado que uno con valor 200 (por lo menos en teoría).

Ejemplo:

```
<P STYLE="font-weight: 100">100</P>
<P STYLE="font-weight: 200">200</P>
<P STYLE="font-weight: 300">300</P>
<P STYLE="font-weight: 400">400</P>
<P STYLE="font-weight: 500">500</P>
<P STYLE="font-weight: 600">600</P>
<P STYLE="font-weight: 700">700</P>
<P STYLE="font-weight: 800">800</P>
<P STYLE="font-weight: 900">900</P>
```

Resultado:

100... 200... 300... 400... 500... **600... 700... 800... 900...**

Además de estos valores numéricos podemos utilizar:

normal

Texto normal. Similar al valor numérico 400.

```
P { font-weight: normal }
```

bold

Pone en negritas algún texto. Similar al valor numérico 700.

```
P { font-weight: bold }
```

bolder

Similar a aumentar en 100 el valor numérico.

```
P { font-weight: bolder }
```

lighter

Similar a disminuir en 100 el valor numérico.

```
P { font-weight: lighter }
```

En esta propiedad no se permite el uso de porcentajes como valores.

Ejemplo:

```
<P STYLE="font-weight: normal">normal</P>  
<P STYLE="font-weight: bold">bold</P>  
<P STYLE="font-weight: bolder">bolder</P>  
<P STYLE="font-weight: lighter">lighter</P>
```

Resultado:

normal... **bold...** **bolder...** lighter...

15.6 font

Veamos el siguiente ejemplo:

```
<HTML>  
<HEAD>  
<TITLE>Ejemplo</TITLE>  
<STYLE type="text/css">  
H1 {
```

```
font-family: "Times New Roman", Serif ;  
font-size: 50px ;  
font-style: normal ;  
font-weight: 700 ;
```

```

}
P {

font-family: Arial, Sans-Serif;
font-size: 14px;
font-style: italic;
font-variant: normal;
font-weight: 400;

}
</STYLE>
</HEAD>
<BODY bgcolor="#FFFFFF">
<H1>Encabezado</H1>
<P>Algún párrafo</P>
</BODY>
</HTML>

```

La forma en que escribimos las propiedades para H1 y P resulta un poco larga. Para evitarnos algunas líneas de código podemos usar la propiedad font de la siguiente manera:

```

<HTML>
<HEAD>
<TITLE>Ejemplo</TITLE>
<STYLE type="text/css">
H1 { font: 700 50px "Times New Roman", Serif }
P { font: italic 14px Arial, Sans-Serif }
</STYLE>
</HEAD>
<BODY bgcolor="#FFFFFF">
<H1>Encabezado</H1>
<P>Algún párrafo</P>
</BODY>
</HTML>

```

El orden importa por lo cual se muestra la forma de escribir los valores:

```
font: || style || variant || weight || size || family ||
```

Como vemos, los valores para font son los mismos utilizados con las propiedades vistas anteriormente.

Fin de la sección 15

- [16. 1 text-align](#)
- [16. 2 text-decoration](#)
- [16. 3 text-indent](#)
- [16. 4 letter-spacing](#)
- [16. 5 text-transform](#)
- [16. 6 word-spacing](#)

16. 1 text-align

Esta propiedad nos permite alinear nuestro texto de la misma forma que con el atributo ALIGN. Lo nuevo e interesante es que ahora si tenemos la posibilidad para justificar nuestros párrafos.

Los valores para esta propiedad son left (izquierda), center (centro), right (derecha), justify (justificado).

El siguiente texto nos ayudara a mostrar los resultados de un texto justificado. Fue tomado del libro "Los dragones del edén" de Carl Sagan:

El mundo es viejísimo y el ser humano sumamente joven. Los acontecimientos relevantes de nuestras vidas se miden en años o fracciones de tiempo aún más pequeñas, en tanto que la duración de una vida humana se reduce a unos pocos decenios, el linaje familiar a unos cuantos siglos y los hechos que registra la historia a unos milenios. Y, sin embargo, se extiende a nuestras espaldas un fantástico panorama temporal que se pierde en un pasado remotísimo del que apenas sabemos nada. En primer lugar porque no poseemos testimonios escritos, y en segundo lugar porque resulta muy difícil hacerse una idea de la inmensidad de los períodos involucrados.

16. 2 text-decoration

Los posibles valores para esta propiedad son:

none

Ninguna decoración

`<P STYLE="text-decoration: none">`Ejemplo de none

underline

Similar a lo obtenido con la etiqueta `<U>` , es decir, subrayado.

`<P STYLE="text-decoration: underline">`Ejemplo de underline

Resultado:

Ejemplo de underline

overline

Dibuja una línea en la parte superior.

`<P STYLE="text-decoration: overline">Ejemplo de overline`

Resultado:

Ejemplo de overline

line-through

Similar a lo obtenido con la etiqueta `<S>` , es decir, tachado.

`<P STYLE="text-decoration: line-through">Ejemplo de line-through`

Resultado:

~~Ejemplo de line-through~~

blink

¿Recuerdan el truco molesto del parpadeo? Bueno pues aquí hay otra forma de lograrlo:

`<P STYLE="text-decoration: blink">Ejemplo de blink (Netscape)`

Resultado:

Ejemplo de blink (Netscape)

16. 3 text-indent

Nos permite dar un sangrado. Los valores pueden ser en píxeles, pulgadas, porcentajes etc.

Ejemplo:

`<P STYLE="text-indent: 10px">Línea 1</P>`

`<P STYLE="text-indent: 20px">Línea 2</P>`

`<P STYLE="text-indent: 30px">Línea 3</P>`

`<P STYLE="text-indent: 40px">Línea 4</P>`

`<P STYLE="text-indent: 50px">Línea 5</P>`

Resultado:

Línea 1

Línea 2

Línea 3

Línea 4

Línea 5

16. 4 letter-spacing

Nos permite especificar el tamaño del espacio entre las letras de una palabra. El valor es especificado en píxeles, centímetros, porcentajes etc. Para volver a tener el espaciado normal usamos el valor normal.

Ejemplo:

```
<P STYLE="letter-spacing: normal">letter-spacing: normal</P>  
<P STYLE="letter-spacing: 5px">letter-spacing: 5px</P>  
<P STYLE="letter-spacing: 15px">letter-spacing: 15px</P>  
<P STYLE="letter-spacing: 20px">letter-spacing: 20px</P>  
<P STYLE="letter-spacing: 25px">letter-spacing: 25px</P>
```

Resultado:

letter-spacing: normal

l e t t e r - s p a c i n g : 5 p x

l e t t e r - s p a c i n g : 1 5 p x

l e t t e r - s p a c i n g :
2 0 p x

l e t t e r - s p a c i n g :
2 5 p x

16. 5 text-transform

Permite especificar si un texto estará en minúsculas o mayúsculas. Los valores para esta propiedad son:

none

No aplica ningún estilo.

capitalize

Pone el primer carácter de cada palabra en mayúsculas.

uppercase

Pone todos los caracteres de cada palabra en mayúsculas.

lowercase

Pone todos los caracteres de cada palabra en minúsculas.

Ejemplo:

```
<P STYLE="text-transform: none">Ejemplo de none</P>  
<P STYLE="text-transform: capitalize">Ejemplo de capitalize</P>  
<P STYLE="text-transform: uppercase">Ejemplo de uppercase</P>  
<P STYLE="text-transform: lowercase">Ejemplo de lowercase</P>
```

Resultado:

Ejemplo de none

Ejemplo de capitalize

EJEMPLO DE UPPERCASE

Ejemplo de lowercase

16. 6 word-spacing

Permite especificar el espacio en blanco que habrá entre palabras. Los valores para esta propiedad son píxeles, milímetros, porcentajes, etc. Para volver a tener el espaciado normal usamos el valor normal.

Ejemplo

```
<P STYLE="word-spacing: normal">word-spacing: normal</P>  
<P STYLE="word-spacing: 5px">Ejemplo word-spacing: 5px</P>  
<P STYLE="word-spacing: 10px">Ejemplo word-spacing: 10px</P>  
<P STYLE="word-spacing: 15px">Ejemplo word-spacing: 15px</P>  
<P STYLE="word-spacing: 20px">Ejemplo word-spacing: 20px</P>
```

Resultado:

word-spacing: normal

Ejemplo word-spacing: 5px

Ejemplo word-spacing: 10px

Ejemplo word-spacing: 15px

Ejemplo word-spacing: 20px

Fin de la sección 16

Sección 17. La caja

[17. 1 La caja \(box\)](#)

[17. 2 Dimensiones de la caja](#)

[17. 3 Propiedades del colchón](#)

[17. 4 Propiedades del borde](#)

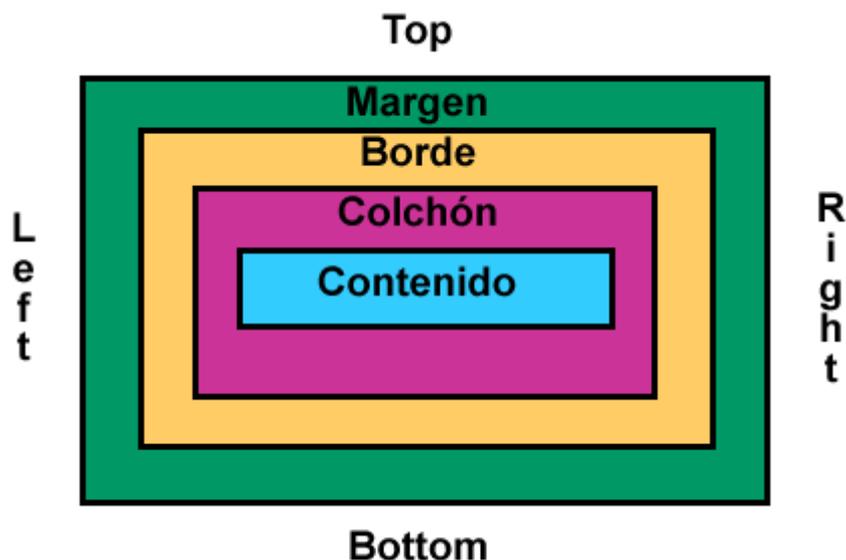
[17. 5 Propiedades de los márgenes](#)

[17. 6 Ejemplos](#)

17. 1 La caja (box)

En la sección 14 hablamos sobre el árbol de documento. A cada uno de los elementos dentro de este árbol les corresponde un espacio específico, llámenlo "espacio vital", "esfera de influencia o, menos rebuscado: "caja".

Antes que nada debemos de decir que cada una de las cajas tiene 4 partes que son el contenido, el colchón (padding), el borde (border) y el margen (margin). Cada una de estas partes tiene una frontera o limite que la separa de las demás, así, podemos hablar de un limite de contenido, limite de colchón, limite de borde y limite de margen.



Nosotros podremos dar distintas características a las distintas partes de la caja para obtener efectos impresionantes que no podríamos crear con simple HTML. ¿No me creen? Bueno, pues sigan leyendo.

Contenido

Si tenemos:

```
<P STYLE="text-decoration: underline">Hola Mundo
```

entonces la cadena "Hola Mundo" es el contenido de la caja.

Colchón

Es el espacio que se encuentra entre el contenido y el margen. Al usar la propiedad background-color para especificar el color de fondo de un elemento, tanto el contenido como el colchón tomaran este color.

Borde

Es una línea que separa el colchón de los márgenes. Cuando no se indica su tamaño se utiliza el valor por defecto lo que significa un borde no visible.

Margen

Es el espacio en blanco que existe alrededor de una caja, ningún otro elemento se colocara dentro de estos limites. Cabe destacar que el margen siempre será transparente.

17. 2 Dimensiones de la caja

Antes de comenzar hagamos que se vea un elemento con su caja correspondiente:

```
<STYLE TYPE="text/css">
P {
height: 400px;
width: 400px;
border-style: solid
}
</STYLE>
</HEAD>

<BODY>
<P>Hola Mundo
</BODY>
</HTML>
```

Esto es lo que obtenemos



Hola Mundo

Con height especificamos el alto de la caja y con width el ancho para P. De la otra propiedad se hablara mas adelante (aunque no creo que tengan mucho problema en descifrar para que sirve).

Cabe destacar que el ancho esta dado por la suma de los márgenes laterales (izquierdo y derecho), el ancho del colchon y el ancho del contenido. De manera similar, la altura esta dada por la suma de los márgenes superior e inferior, la altura del colchon y la altura del contenido.

17.3 Propiedades del colchón

Las propiedades del colchón permiten especificar el tamaño de esta área. Al igual que todas las propiedades que trabajan con medidas, nosotros podemos especificar el tamaño en distintas unidades, incluso en porcentaje. Para mayor información sobre medidas ver la sección 14.

padding-top: especifica el tamaño de la parte superior del colchón.

padding-right: especifica el tamaño de la parte derecha del colchón.

padding-bottom: especifica el tamaño de la parte inferior del colchón.

padding-left: especifica el tamaño de la parte izquierda del colchón.

Este es un ejemplo de cómo luce nuestro texto sin utilizar las propiedades del colchón. Las propiedades border-width y border-style se verán mas adelante.

```
P {  
  
width: 150px;  
border-width: 5px 5px 5px 5px;  
border-style: solid;  
  
}
```

Resultado:



Hola Mundo

```
P {  
  
width: 150px;  
border-width: 5px 5px 5px 5px;  
border-style: solid;  
padding-top: 50px;  
  
}
```

Resultado:



Hola Mundo

```
P {  
  
width: 150px;  
border-width: 5px 5px 5px 5px;  
border-style: solid;  
padding-left: 20px;  
  
}
```

Resultado:

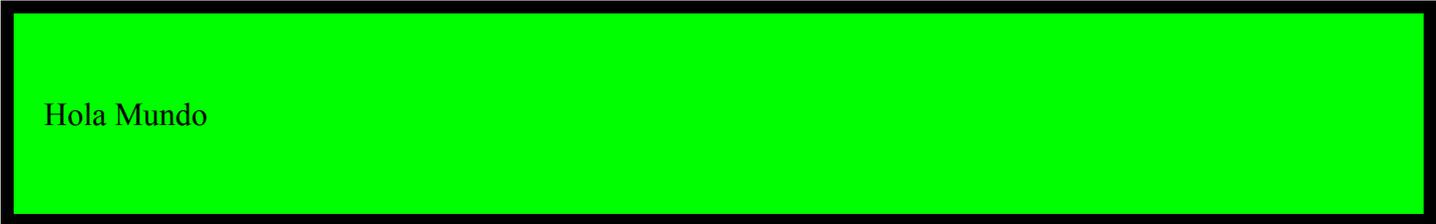


Hola Mundo

Todos juntos:

```
P {  
  
width: 200px;  
border-width: 5px 5px 5px 5px;  
border-style: solid;  
padding-top: 100px;  
padding-right: 150px;  
padding-bottom: 100px;  
padding-left: 10px;  
background-color: #0f0;  
  
}
```

Resultado:



Hola Mundo

padding

Esta propiedad nos permitirá ahorrar unas líneas de código. El ejemplo anterior quedaría:

```
P {  
  
width: 200px;  
border-width: 5px 5px 5px 5px;  
border-style: solid;  
padding: 100px 150px 100px 10px;  
background-color: #0f0;  
  
}
```

es decir, el orden para colocar los valores es top (arriba), right (derecha), bottom (abajo), left (izquierda). Tal como las manecillas del reloj.

¿Que hace el siguiente código?

```
P {  
  
width: 200px;  
border-width: 5px 5px 5px 5px;  
border-style: solid;  
padding: 100px 50px;  
background-color: #069;  
  
}
```

Cuando en padding no encontramos cuatro valores entonces los valores son tomados de la siguiente forma:

Un valor: top, right bottom y left toman el valor.

Dos valores: top y bottom toman el primer valor y right y left el segundo valor.

Tres valores: top toma el primer valor, left y right el segundo y bottom el tercero.

Este es el resultado del ejemplo anterior.



Hola Mundo

17. 4 Propiedades del borde

Hasta ahora hemos utilizado propiedades de los bordes sin meternos para nada en el tema. En el ultimo ejemplo teníamos la siguiente línea:

```
border-width: 5px 5px 5px 5px;
```

border-width nos permite especificar el tamaño de los bordes en el orden de manecillas de reloj.

Esta propiedad es la forma compacta de especificar el tamaño de borde pero para aquellos que les guste complicarse la vida mostramos la forma larga:

border-top-width: tamaño del borde superior.

border-right-width: tamaño del borde derecho.

border-bottom-width: tamaño del borde inferior.

border-left-width: tamaño del borde izquierdo.

En la forma compacta se aplica la misma regla para cuando encontramos menos de 4 valores.

La parte mas interesente de estas propiedades es que existen medidas especiales para indicar el tamaño del borde. Además de poder utilizar las medidas que ya conocemos, podemos utilizar las siguientes medidas:

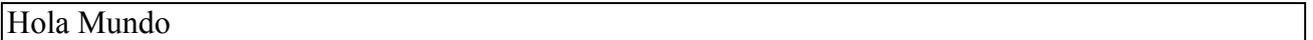
thin

Para un borde delgado.

```
P {
```

```
border-width: thin;  
border-style: solid;
```

```
}
```



Hola Mundo

medium

Para un borde medio.

```
P {
```

```
border-width: medium  
border-style: solid;
```

```
}
```



Hola Mundo

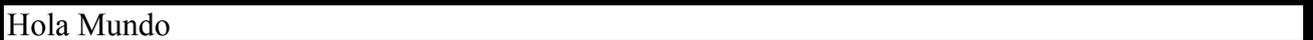
thick

Para un borde grueso.

```
P {
```

```
border-width: thick;  
border-style: solid;
```

```
}
```



Hola Mundo

Y esto es todo respecto a bordes... si es que no queremos bordes de colores o bordes especiales.

Veamos el siguiente ejemplo:

```
P {
```

```
width: 2cm;
border-width: 10px;
border-style: solid;
border-color: #000 #f00 #0f0 #00f;
}
```



Hola Mundo

Ya se habrán dado cuenta que de nuevo utilizamos la forma compacta que igual utiliza la forma de manecillas de reloj. De preferir complicarnos la vida debemos de utilizar:

```
border-top-color: color del borde superior.
border-right-color: color del borde derecho.
border-bottom-color: color del borde inferior.
border-left-color: color del borde izquierdo.
```

En la forma compacta se aplica la misma regla para cuando encontramos menos de 4 valores.

Además de las notaciones que ya conocemos (vistas en la sección 14) para especificar el color, también podemos usar el valor `transparent` para esta propiedad. Este valor hace que el borde sea transparente, lo cual no implica que no exista o que no tenga dimensiones.

Nota: MS IExplorer no reconoce el valor `transparent`.

Dicho esto tan solo nos queda hablar de los estilos de borde que podremos utilizar. Para esto utilizaremos la propiedad `border-style` que otra vez es la forma compacta y sigue las mismas reglas que las anteriores propiedades.

Las propiedades no compactas son: `border-top-style`, `border-right-style`, `border-bottom-style` y `border-left-style`, las cuales, creo, no necesitan explicación.

Los valores que pueden tomar las propiedades de estilo de borde son:

`none`

Sin borde.

```
P {
```

```
width: 2cm;
border-width: 10px;
border-style: none;
border-color: #000 #069;
```

```
}
```

dashed

Líneas discontinuas.

P {

width: 5cm;
border-width: 10px;
border-style: dashed;
border-color: #000 #069;

}



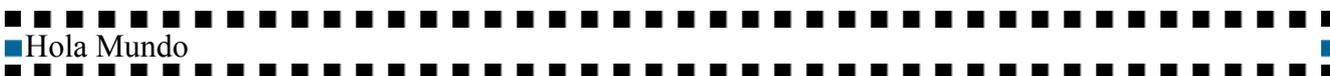
dotted

Punteado.

P {

width: 5cm;
border-width: 10px;
border-style: dotted;
border-color: #000 #069;

}



double

Dos líneas continuas.

P {

width: 5cm;
border-width: 10px;
border-style: double;
border-color: #000 #069;

}



groove

Lo opuesto a ridge.

```
P {  
  
width: 5cm;  
border-width: 10px;  
border-style: groove;  
border-color: #000 #069;  
  
}
```



inset

Hace que la caja se vea en bajorrelieve.

```
P {  
  
width: 5cm;  
border-width: 10px;  
border-style: inset;  
border-color: #000 #069;  
  
}
```



outset

Hace que la caja se vea en altorrelieve.

```
P {  
  
width: 5cm;  
border-width: 10px;  
border-style: outset;  
border-color: #000 #069;  
  
}
```



ridge

Lo opuesto a groove.

```
P {  
  
width: 5cm;  
border-width: 10px;
```

```
border-style: ridge;
border-color: #000 #069;

}
```



solid

Una línea continua.

```
P {
width: 5cm;
border-width: 10px;
border-style: solid;
border-color: #000 #069;
}
```



Nota: el porque de las definiciones de groove y ridge se debe a que aun no encuentro palabras para describir el efecto. Cualquier idea enviarla a mi correo.

17. 5 Propiedades de los márgenes

Las propiedades de márgenes tan solo nos permitirán especificar el tamaño del margen, el cual dijimos que es siempre transparente.

La forma compacta es margin y cumple las mismas reglas vistas anteriormente:

```
P {
width: 5in;
border-width: thick;
border-style: double;
border-color: #000 #069 transparent;
margin: 20px 20px 20px 100px;
}
```



Para la forma extendida tenemos las propiedades margin-top, margin-right, margin-bottom y margin-left.

A manera de ejercicio... ¿Qué creen que haga el siguiente código?

```
P {  
  
width: 5in;  
background-color: #f96;  
margin-left: -15px;  
/* Si, es un signo menos */  
padding-left: 15px;  
  
}
```

17. 6 Ejemplos

```
<HTML>  
<HEAD>  
<TITLE>Recursi&oacute;n</TITLE>  
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">  
<STYLE TYPE="TEXT/CSS">
```

```
BODY {  
  
background-color: #006;  
color: #FFF;  
font-family: sans-serif;  
font-size: 13px;  
text-align: justify;  
margin: 1cm 2cm 4cm 1cm;  
  
}
```

```
DIV {  
  
border-color: #09F;  
border-width: thin;  
border-style: groove;  
padding: 20px;  
  
}
```

```
.titulo {  
  
color: #FF0;  
font-size: 24px;  
letter-spacing: 20px;  
  
}
```

```
*#resaltar {
```

```
color: #f96;
font-weight: bold;
}
```

```
</STYLE>
</HEAD>
```

```
<BODY>
<P CLASS="titulo">Recursi&oacute;n
<DIV>
```

<P>La recursión es una técnica algorítica donde una función se llama a si misma para completar una tarea.

<P>Para aplicar soluciones recursivas se debe dividir el problema en pequeñas partes, invocar a la función recursivamente y combinar las soluciones de las partes en una sola que será la solución final al problema.

<P>Una parte importante de una solución recursiva es la condición de paro. Cuando las condiciones son conseguidas, un algoritmo recursivo deja de llamarse a si mismo y comienza a regresar valores.

```
</DIV>
```

```
<P CLASS="titulo">Recursi&oacute;n vs Iteraci&oacute;n
```

```
<DIV>
```

<P>Tanto la iteración y la recursión se basan en una estructura de control; para el caso de la iteración se trata de una estructura de repetición y para la recursión se trata de una estructura de selección. Ambas también involucran una prueba de terminación y si esta prueba es equivocada pueden ocurrir de forma indefinida.

<P>Los contras de la recursión son que esta, al invocar de manera repetida al mecanismo tiende a sobrecargar las llamadas de función lo que resulta costoso tanto en tiempo de procesador como en espacio en memoria. Cada llamada recursiva genera otra copia de las variables de la función.

<P>Los pros de la recursión son que permite dar mas naturalidad a ciertos problemas lo que resulta en un programa fácil de comprender y depurar. Otra razón es que la solución iterativa puede no ser tan evidente.

```
</DIV>
</BODY>
</HTML>
```

[Este es el resultado](#)

R e c u r s i ó n

La recursión es una técnica **algorítmica** donde una función se llama a si misma para completar una tarea.

Para aplicar soluciones recursivas se debe **dividir el problema** en pequeñas partes, invocar a la función recursivamente y **combinar** las soluciones de las partes en una sola que será la solución final al problema.

Una parte importante de una solución recursiva es la **condición de paro**. Cuando las condiciones son conseguidas, un algoritmo recursivo deja de llamarse a si mismo y comienza a regresar valores.

R e c u r s i ó n v s I t e r a c i ó n

Tanto la iteración y la recursión se basan en una **estructura de control**, para el caso de la iteración se trata de una **estructura de repetición** y para la recursión se trata de una **estructura de selección**. Ambas también involucran una prueba de terminación y si esta prueba es equivocada pueden ocurrir de forma indefinida.

Los contras de la recursión son que esta, al invocar de manera repetida al mecanismo tiende a sobrecargar las llamadas de función lo que resulta costoso tanto en tiempo de procesador como en espacio en memoria. Cada llamada recursiva genera otra copia de las variables de la función.

Los pros de la recursión son que permite dar mas naturalidad a ciertos problemas lo que resulta en un programa fácil de comprender y depurar. Otra razón es que la solución iterativa puede no ser tan evidente.

Fin de la Sección 17.

Sección 18. Integración de interfaz

[18. 1 Introducción](#)

[18. 2 Colores](#)

[18. 3 Fuentes](#)

[18. 4 Cursores](#)

18. 1 Introducción

Muchas veces, cuando diseñamos un sitio, no nos ponemos a pensar en lo fácil o difícil de navegar que es nuestro sitio. No toda la gente tiene tan buena vista como nosotros así que si en realidad deseamos que nuestro sitio sea amigable con todos nuestros visitantes debemos de dar opciones para aquellas personas que así lo deseen. El uso de colores o fuentes predefinidos en el sistema del usuario es una manera sencilla de tomar en cuenta las necesidades visuales de nuestros visitantes. De este tema hablaremos en esta sección.

18. 2 Colores

Los colores específicos de sistema que podemos utilizar con las distintas propiedades vistas hasta el momento son:

ActiveBorder: borde de la ventana activa

ActiveCaption: fondo de la barra de títulos activa

AppWorkspace:

Background: fondo del escritorio

ButtonFace:
ButtonHighlight: elementos en 3 dimensiones seleccionados
ButtonShadow: sombra de elementos en 3 dimensiones
ButtonText: texto de los botones
CaptionText: texto de la barra de títulos inactiva
GrayText: elementos desactivados
Highlight: elementos seleccionados
HighlightText: texto seleccionados
InactiveBorder: borde de la ventana inactiva
InactiveCaption: fondo de la barra de títulos inactiva
InactiveCaptionText: texto de la barra de títulos inactiva
InfoBackground: fondo de la información (tooltips)
InfoText: texto de la información (tooltips)
Menu: fondo de menús
MenuText: texto de menús
Scrollbar: barra de desplazamiento
ThreeDDarkShadow: sombra oscura para elementos en 3 dimensiones
ThreeDFace:
ThreeDHighlight: elementos en 3 dimensiones seleccionados
ThreeDLightShadow: sombra clara para elementos en 3 dimensiones
ThreeDShadow: sombra clara para elementos en 3 dimensiones
Window: fondo ventana
WindowFrame: marco de la ventana
WindowText: texto de la ventana

[Ejemplos de uso](#)

18. 3 Fuentes

En la sección 15. 6 hablamos de la propiedad font y vimos que era una forma condensada para el especificar el estilo, tamaño, familia, etc. de una fuente. Además de los valores vistos en esa sección, existen otros que nos permiten utilizar las fuentes que el usuario a definido en su sistema.

Estos valores son: caption, icon, menu, message-box, small-caption, status-bar.

<P STYLE="font: caption">Ejemplo de uso de estos valores</P>

Ejemplo de caption

Ejemplo de menu

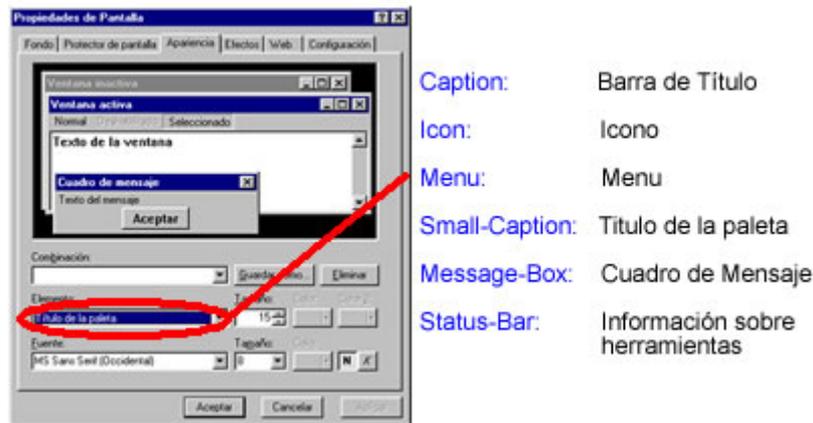
Ejemplo de icon

Ejemplo de message-box

Ejemplo de small-caption

Ejemplo de status-bar

En el caso de Windows, el usuario especifica las fuentes dentro de las propiedades de pantalla. A continuación se muestra la relación que hay entre estas propiedades y el respectivo elemento del sistema.



En cuanto a la compatibilidad, Netscape soporta todos estos valores pero Iexplorer parece no soportar los valores message-box, small-caption ni status-bar.

Nota: En el caso de *nix, Mozilla no establece ninguna diferencia entre estos valores pero si los despliega de manera diferente al texto al que no se le han aplicado estilos. En mi caso, por ejemplo, el texto con estilos, con estos valores, se despliega como Sans-Serif de 13 px y el texto sin estilos se despliega como Serif de 16 px.

18. 4 Cursores

Así como hemos sido capaces de especificar nuestras fuentes y colores, también podemos especificar el cursor que se desplegara en ciertos elementos de nuestra pagina. Para esto utilizaremos la propiedad cursor. Esta propiedad puede utilizar los valores: auto, crosshair, default, pointer, move, e-resize, ne-resize, nw-resize, n-resize, se-resize, sw-resize, s-resize, w-resize, text, wait y help. A continuación se muestra un ejemplo en la siguiente tabla (pasen su ratón por cada una de las celdas):

auto	crosshair	default	pointer	move	text	wait	help
e-resize	ne-resize	nw-resize	n-resize	se-resize	sw-resize	s-resize	w-resize

En cada una de las celdas aplicamos algo similar a esto:

```
<TD STYLE="cursor: ne-resize">ne-resize
```

Fin de la Sección 18.

Sección 19. Propiedades de color y fondo

[19.1 color](#)

[19.2 background-color](#)

[19.3 background-image](#)

[19.4 background-attachment](#)

[19.5 background-position](#)

[19.6 background-repeat](#)

[19.7 background](#)

19.1 color

En secciones anteriores hemos usado esta propiedad así que el resultado de aplicarla no es desconocido. Esta propiedad nos permite cambiar el color (foreground) de un elemento. Por ejemplo:

```
<HTML>
<HEAD>
<TITLE>Ejemplo</TITLE>
<STYLE type="text/css">
BODY { color: #FF0000}
P { color: #990000 }
P I { color: #0000FF }
</STYLE>
</HEAD>
<P>Ejemplo del uso de <I>color</I>
<DIV>Algo mas...</DIV>
</BODY>
</HTML>
```

19.2 background-color

Esta propiedad, a diferencia de la anterior nos permite cambiar el color de fondo de un elemento. Regresando al ejemplo anterior:

```
<HTML>
<HEAD>
<TITLE>Ejemplo</TITLE>
<STYLE type="text/css">
BODY { color: #FF0000; background-color: #CCCCFF }
P { color: #990000;}
P I { color: #0000FF; background-color: #FFFFFF }
</STYLE>
</HEAD>
<P>Ejemplo del uso de <I>color</I>
<DIV>Algo mas...</DIV>
</BODY>
</HTML>
```

```

HR { background-color: #00FF00;}
</STYLE>
</HEAD>
<P>Ejemplo del uso de <I>color</I> y <I>background-color</I></P>
<DIV>Algo mas...</DIV>
<HR size="10">
</BODY>
</HTML>

```

Además de los valores que conocemos, se puede usar el valor transparent (transparente). Esto es, el elemento toma el color de fondo según en donde se encuentre:

```

<HTML>
<HEAD>
<TITLE>Ejemplo</TITLE>
<STYLE type="text/css">
I { background-color: transparent }
</STYLE>
</HEAD>
<P STYLE="background-color: #0000FF"><I>Párrafo 1, Color Azul</I></P>
<P STYLE="background-color: #00FF00"><I>Párrafo 2, Color Verde</I></P>
<P STYLE="background-color: #FF0000"><I>Párrafo 3, Color Rojo</I></P>
<P>
</BODY>
</HTML>

```

19. 3 background-image

Esta propiedad nos permitirá usar una imagen de fondo para cualquier elemento.

Los posibles valores para esta propiedad pueden ser:

none

Sin imagen de fondo. Ejemplo:

```
BODY { background-image: none }
```

Dirección:

Permite especificar en donde se encuentra la imagen.

[Ejemplo](#)

19. 4 background-attachment

Los posibles valores para esta propiedad son:

fixed

La imagen de fondo no se mueve junto con el documento.

Ejemplo:

```
<STYLE type="text/css">  
BODY { background-image: url(../imagenes/topicgnome.gif); background-attachment: fixed }  
</STYLE>
```

[Resultado](#)

scroll

La imagen de fondo se mueve junto con el documento. Este es el valor por defecto.

Ejemplo:

```
<STYLE type="text/css">  
BODY { background-image: url(../imagenes/topicgnome.gif); background-attachment: scroll }  
</STYLE>
```

[Resultado](#)

19. 5 background-position

Con esta propiedad podremos indicar la posición de una imagen de fondo si es que se ha especificado alguna. La posición se puede especificar con medidas tradicionales (píxeles, centímetros, etc) pero es recomendable hacer uso de porcentajes. La siguiente imagen muestra la disposición de la pantalla:

0%, 0%

50%, 0%

100%, 0%

0%, 50%

50%, 50%

100%, 50%

0%, 100%

50%, 100%

100%, 100%

La declaración es de la forma

background-position: X% Y%

donde el primer valor (X%) es para la posición horizontal y el segundo valor (Y%) para la posición vertical. Así, para que la imagen este en la esquina superior izquierda, usamos los valores 0% y 0% de la siguiente manera:

```
<STYLE type="text/css">
BODY { background-image: url(../imagenes/eriol_slashdot.jpg); background-repeat: no-repeat;
background-position: 0% 0% }
</STYLE>
```

Resultado

Para que la imagen este en el centro (horizontal y vertical) usamos los valores 50% y 50% de la siguiente manera:

```
<STYLE type="text/css">
BODY { background-image: url(../imagenes/eriol_slashdot.jpg); background-repeat: no-repeat;
background-position: 50% 50% }
</STYLE>
```

Resultado

Para que la imagen este en la esquina inferior derecha usamos los valores 100% y 100% de la siguiente manera:

```
<STYLE type="text/css">
BODY { background-image: url(../imagenes/eriol_slashdot.jpg); background-repeat: no-repeat;
background-position: 100% 100% }
</STYLE>
```

Resultado

La propiedad background-repeat se vera mas adelante.

19. 6 background-repeat

Nos permite indicar el comportamiento que tendrá la imagen de fondo en cuanto a la repetición que habrá de la misma. Los valores para esta propiedad son:

repeat

La imagen se repite horizontal y verticalmente.

Ejemplo:

```
<STYLE type="text/css">  
BODY { background-image: url(../imagenes/topicgnome.gif); background-repeat: repeat }  
</STYLE>
```

[Resultado](#)

repeat-x

La imagen se repite horizontalmente.

Ejemplo:

```
<STYLE type="text/css">  
BODY { background-image: url(../imagenes/topicgnome.gif); background-repeat: repeat-x }  
</STYLE>
```

[Resultado](#)

repeat-y

La imagen se repite verticalmente.

Ejemplo:

```
<STYLE type="text/css">  
BODY { background-image: url(../imagenes/topicgnome.gif); background-repeat: repeat-y }  
</STYLE>
```

[Resultado](#)

no-repeat

Solo se muestra una vez; sin repetición.

Ejemplo:

```
<STYLE type="text/css">
BODY { background-image: url(../imagenes/topicgnome.gif); background-repeat: no-repeat }
</STYLE>
```

Resultado

19. 7 background

Esta propiedad nos permite agrupar todas las propiedades anteriores para salvarnos líneas de código. Es similar a los visto en la propiedad font por lo que solo necesitamos saber el orden en que deberemos escribir los valores:

```
background : || color || image || repeat || attachment || position ||
```

Ejemplo:

```
<HTML>
<HEAD>
<TITLE>Ejemplo</TITLE>
<STYLE type="text/css">
BODY {
background-color: #000000;
background-image: url(../imagenes/eriol_slashdot.jpg);
background-repeat: no-repeat;
background-position: 50% 50% }
</STYLE>
</HEAD>
<BODY bgcolor="#FFFFFF">
</BODY>
</HTML>
```

Se escribe mas fácil:

```
<HTML>
<HEAD>
<TITLE>Ejemplo</TITLE>
<STYLE type="text/css">
BODY { background: #000000 url(../imagenes/eriol_slashdot.jpg) no-repeat 50% 50% }
P { color: #FFFFFF}
</STYLE>
</HEAD>
<BODY bgcolor="#FFFFFF">
</BODY>
</HTML>
```

Fin de la sección 19