



Web [manual-xhtml.blogspot.com](http://manual-xhtml.blogspot.com)

## Lista de contenidos

Antes de empezar podés enterarte leyendo un poco más [acerca de este manual](#)

1. [Primer documento XHTML](#)
  - [1. Introducción](#)
  - [2. Fichero XHTML, editor y navegador web](#)
  - [3. Etiquetas](#)
  - [4. Primer documento XHTML](#)
  - [5. Ejemplo](#)
2. [Encabezados y marcas de importancia](#)
  - [Encabezados títulos](#)
  - [Elemento strong](#)

- [Elemento em](#)
- 3. [Enlaces, vínculos y anchor](#)
  - [1. Introducción](#)
  - [2. Primer link](#)
  - [3. Enlace a correo electrónico y otros recursos](#)
  - [4. Enlaces relativos y absolutos](#)
  - [5. Enlaces hacia anclas en la misma página](#)
  - [6. El atributo title](#)
  - [7. Atributos rel y rev](#)
    - [7.1. Valor nofollow para rel](#)
    - [7.2. Valor tag para rel \(Technorati\)](#)
  - [8. Código con todos los ejemplos:](#)
  - [9. Ejemplo](#)
- 4. [Citas, abreviaturas y acrónimos](#)
  - [1. Citas](#)
  - [2. Citas con <blockquote>](#)
  - [3. Citas con <cite>](#)
  - [4. Abreviaturas](#)
  - [5. Acrónimos](#)
  - [6. Código del ejemplo](#)
  - [7. Ejemplo](#)
- 5. [Imágenes](#)
  - [1. Imágenes formatos](#)
  - [2. El elemento img](#)
  - [3. Altura y ancho](#)
  - [4. Alineación de las imágenes \(con atributo align\)](#)
  - [5. Alineación de las imágenes \(con propiedad float\)](#)
    - [5.1. float: right](#)
    - [5.2. float: left](#)
- 6. [Listas](#)
  - [1. Introducción](#)
  - [2. Listas ordenadas y numeradas](#)
    - [2.1. Diferentes tipos de listas ordenadas](#)
  - [3. Listas desordenadas](#)
    - [3.1. Diferentes tipos de listas desordenadas](#)

- [4. Listas de definiciones](#)
  - [5. Listas compuestas mediante anidamiento](#)
- 7. [Texto preformateado](#)
  - [1. Elemento pre](#)
  - [2. Elemento code](#)
- 8. [Tablas](#)
  - [1. Introducción](#)
  - [2. Ejemplo sencillo](#)
  - [3. El atributo summary](#)
  - [4. El pie de la tabla \(footer\)](#)
  - [5. Elementos de una tabla gráficamente](#)
- 9. [Formularios](#)
  - [1. Introducción](#)
  - [2. Dos métodos post y get](#)
    - [2.1. Método get](#)
    - [2.2. Método post](#)
  - [3. Empezando con form](#)
  - [4. Elemento input](#)
  - [5. Campos de texto](#)
  - [6. Botones de radio](#)
  - [7. Botón de comprobación checkbox](#)
  - [8. Botones de envío y de reset](#)
  - [9. Cajas de texto](#)
  - [10. Seleccionar con select](#)
  - [11. Elemento label](#)
  - [12. Agrupando con fieldset](#)

## Apuntes sueltos

Aquí espero, en la medida de que el tiempo me lo permita, dejar algunas anotaciones sueltas sobre temas que puedan complementar el manual.

- [Elemento span, semántica y estilos](#)
- [Emacs como editor de texto XHTML e IDE para el desarrollo WEB](#)



Esta obra está; bajo una [licencia de Creative Commons](#).

Realizado por Emilio Astarita, [lee más acerca de este manual](#).



Web [manual-xhtml.blogspot.com](http://manual-xhtml.blogspot.com)

## Sobre este manual

Siempre escribí resúmenes o anotaciones personales sobre los temas que estudio. Estos apuntes me eran de gran ayuda, pero carecían de valor para los demás. Cuando emprendí el estudio del lenguaje XHTML también comencé a hacerlo, pero luego pensé que podría darle algo más de formalidad y rigurosidad para poder hacerlo público, con el fin de constituir un documento que podría llegar a ser útil a otras personas. Este manual es el resultado.

Nunca había escrito de manera formal un documento tan largo y si bien de a momentos fue tedioso, se convirtió en una gran motivación para profundizar el entendimiento en temas que de otra manera no me hubiese involucrado.

Seguramente las páginas del manual contendrán múltiples errores que espero

poder ir depurando con el tiempo, en la medida que la facultad y otras obligaciones me lo permitan. También me gustaría ampliar algunas secciones incompletas (cómo la parte dedicada a tablas) y escribir una introducción a la aplicación de hojas de estilo CSS.

El manual se publica con [una licencia Creative Commons](#), que básicamente permite la publicación completa o incompleta en otros lugares incluso con fines comerciales pero con la salvedad de que debe aclararse quien es el autor y utilizar la misma licencia del original en la redistribución. Agradecería también un aviso y una dirección de mail por cada una de estas reproducciones, para poder notificar las actualizaciones que realice.

Para escribir el manual fue imprescindible el uso de algunas herramientas que paso a enumerar:

- El sistema operativo utilizado que trae por defecto la mayoría de las herramientas es [Kubuntu](#)
- Cómo editores de texto se utilizó [emacs21](#), [Kate](#) y [Quanta](#)
- El coloreo de sintaxis se realizó con la herramienta de exportación a HTML de [Kate](#)
- Para editar imágenes [The Gimp](#), y para el gráfico de las tablas [Kivio](#)
- Para la generación de algunas listas de contenido de manera automática el navegador/editor [Amaya](#)
- Para validar documentos la herramienta on-line [W3C Markup Validation Service](#), algunas extensiones de [emacs21](#) y un uso conjunto de [tidy](#) y los editores mencionados.
- También se buscó compatibilidad con las últimas versiones de [Firefox](#) y [Konqueror](#)

Creo que no hay más que decir si quieren comunicarse conmigo pueden hacerlo en mi [correo personal](#) quitando BORRAME que es sólo una manera de evitar el spam.

[Página principal](#)

**0 aporte(s) de lectores:**

**ATENCIÓN:** Por favor intenta aportar o preguntar sobre el tema que trata la página.

Otro tipo de consultas o críticas puedes dejarlas en el [libro de visitas](#). Los aportes y preguntas están moderados simplemente para evitar spam o comentarios fuera de lugar.



Esta obra está; bajo una [licencia de Creative Commons](#).

Realizado por Emilio Astarita, [lee más acerca de este manual](#).



Web [manual-xhtml.blogspot.com](http://manual-xhtml.blogspot.com)

# Primer documento XHTML

- [1. Introducción](#)
- [2. Fichero XHTML, editor y navegador web](#)
- [3. Etiquetas](#)
- [4. Primer documento XHTML](#)
- [5. Ejemplo](#)

## 1. Introducción

XHTML significa *lenguaje de marcado de hipertexto extendido*.

Se utiliza para generar documentos y contenidos de [hipertexto](#) generalmente publicados en la WEB. Es además una reformulación del lenguaje HTML que se puede jactar de ser ahora compatible con XML.

Un lenguaje de marcado nos permite dejar indicaciones (marcas) en un documento que sirven para diferenciar distintos tipos de contenidos, estructuras o secciones. Por ejemplo hay marcas (elementos, etiquetas) que pueden "marcar" qué es un título, y su lugar en una jerarquía de



importancia, o advertir que una palabra debe estar enfatizada, o bien que una oración es además un enlace hacia otro recurso.

Todo esto, es posible de marcar sin ambigüedades con el correcto uso de XHTML.

Otra de sus virtudes es que está ideado para mantener una saludable separación entre el contenido y el diseño. Es decir, que uno no afecte al otro, y se puedan modificar independientemente.

A medida que avancen y conozcan la aplicación de CSS (hojas de estilo) a los documentos notarán tal separación de manera evidente.

El XHTML está diseñado para que sea funcional no sólo a los navegadores sino a varios dispositivos WEB (teléfonos móviles, portátiles, etc). A lo largo de este manual haremos uso del término *agente de usuario* para referirnos a estos dispositivos capaces de interpretar documentos XHTML.

Antes de seguir puedes recorrer estos enlaces en la [Wikipedia](#) para tener una introducción más amplia y conocer algo sobre la historia de estos lenguajes:

- [Lenguajes de Marcado](#)
- [SGML](#)
- [HTML](#)
- [XML](#)
- [XHTML](#)
- [CSS](#)
- [W3C](#) (World Wide Web Consortium)

**NOTA:** A lo largo de éste manual muchas palabras apuntarán a la enciclopedia libre [Wikipedia](#) y a otros sitios, en un intento de poder ofrecer mayor información o ampliar explicaciones sobre temas y conceptos que escapan al alcance de los objetivos de este manual.

## 2. Fichero XHTML, editor y navegador web

El fichero de un documento XHTML, en sí, es un archivo de texto plano generalmente con alguna de estas extensiones: **.html**, **.htm**, o bien **.xhtml**. Los agentes de usuario interpretan estos archivos y procesan el código para mostrarlos en pantalla.

Es conveniente que a medida que aparezcan ejemplos, practiquen escribiendo sus propios documentos XHTML. Pueden editar los ficheros utilizando un [editor de textos planos](#) y para poder visualizar los resultados del documento un [navegador web](#) (o cualquier otro agente de usuario que interprete XHTML).

Utilicen el editor que ustedes quieran, y al guardarlo se recomienda elegir alguna de estas extensiones, para que su sistema operativo reconozca rápidamente el tipo de fichero (**.html**, **.htm**, **.xhtml**.)

La elección del agente de usuario es menos trivial y se presta a discusión ya que por ejemplo algunos navegadores no cumplen de manera apropiada con los estándares (recomendaciones) que la W3C propone y que intenta respetar este manual.

Personalmente opto por elegir [Mozilla Firefox](#) o [Konqueror](#), puedes utilizar el que tú quieras pero asegurate de que cumpla minimamente con los estándares para que los ejemplos aquí presentados funcionen de manera apropiada.

### 3. Etiquetas

En la escritura de los documentos nos valdremos de *etiquetas* (marcas) como principal elemento de sintaxis, que para diferenciarse del contenido, están encerradas por paréntesis angulares.

Ejemplo del uso de etiquetas, con el elemento **p** (párrafo):

```
<p>Contenido del parrafo</p>
```

**<p>** marca el inicio de la sección que contendrá un párrafo, **</p>** marca el final del párrafo. Toda etiqueta abierta debe ser cerrada escribiéndose de nuevo con el agregado de una barra entre la palabra y el parentesis de angular de apertura.

Algunos elementos que no llevan contenido se nos permite escribirlos de manera tal que la apertura y el cierre se representan con una única etiqueta. El elemento **br** utilizado para el salto de línea, se puede escribir así **<br />** que es una simplificación de la forma también válida: **<br></br>**.

**NOTA:** Es importante notar el espacio que queda entre el fin de la palabra y la barra, en la versión resumida, de esta manera se conserva compatibilidad con algunos agentes de usuario que no reconocen la etiqueta sin ese espacio.

Las etiquetas de apertura pueden contener atributos que especifican características particulares de ésta. Los atributos tienen un nombre seguido de un igual y entre comillas el valor de tal atributo.

Ejemplo del elemento **a** con atributos:

```
<a href="http://www.example.com">Example</a>
```

La etiqueta es **<a>**, el atributo es **href** y el **valor** es *http://www.example.com*. Con esta etiqueta se construyen los vínculos. Tanto los elementos, cómo los atributos y valores, deben escribirse en letras minúsculas. Los valores de los atributos van obligatoriamente entre comillas (incluso cuando estos valores sean números).

Los elementos soportan anidamiento (con algunas condiciones que ya conoceremos).

Ejemplo de anidamiento con los elementos **a** e **img**:

```
<a href="http://www.example.com"></a>
```

El enlace que en el ejemplo anterior se lo habíamos aplicado a la palabra *Example* ahora se lo aplicamos a una segunda etiqueta ( **img** ) la cual tiene la finalidad de mostrar imagenes, por lo que que la etiqueta más externa (elemento **a**) convierte a la imagen en un vínculo.

## 4. Primer documento XHTML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--Este es un comentario y no sera tomado en cuenta por el navegador -->
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<title>Titulo de la pagina</title>
</head>
<body>
<p>Primer documento XHTML, es decir un, Hola mundo</p>
</body>
</html>
```

Analicemos el código línea a línea:

```
<?xml version="1.0" encoding="UTF-8"?>
```

Es una declaración XML, dónde especificamos el atributo **enconding** con el valor *UTF-8*. Este debería ser suficiente en la mayoría de los casos. Si quieres conocer el por qué, puedes investigar más sobre: la [codificación de caracteres](#) y [UTF-8](#) en la **Wikipedia**.

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

Este es el encabezado que deberían llevar todos los documentos **XHTML1.0 estrictos** acordes a

las [especificaciones de la W3C](#). Es un aviso para que el agente de usuario sepa que tipo de documento va a interpretar. Para ser más específicos diremos que al ser un documento de tipo *strict.dtd*, los elementos utilizados serán sólo los incluidos en el lenguaje XHTML 1.0. Si utilizáramos etiquetas antiguas y declaráramos el documento cómo estricto, no podríamos [validar nuestra página XHTML](#).

**NOTA:** Fijarse además que hay un salto de línea a la mitad de la etiqueta, este no es interpretado y tampoco lo es en el contenido a menos que lo explicitemos con la etiqueta `<br />`. Simplemente se incluyen saltos de línea para hacer el código más legible.

Hay otros dos tipos de documentos que mantienen compatibilidad con **HTML 4** y soporte de frames, que no trataremos en este manual, ya que prefirero explicar un uso más estricto del lenguaje.

Declaración de documentos Transitional:

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

Declaración de documentos Frameset:

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

Como dije trataremos de concentrarnos en los documentos declarados como estrictos.

```
<!--Este es un comentario y no sera tomado en cuenta por el navegador-->
```

Los comentarios comienzan con `<!--` y terminan con `-->` (los saltos de línea no los quiebran). No son tomados en cuenta por el agente de usuario, es decir no se interpretan (por lo que no se representan) y se utilizan frecuentemente para hacer aclaraciones en el código.

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
```

Aquí damos comienzo al documento XHTML, todo documento debe llevar esta etiqueta de apertura ( `<html>` ) y al finalizar debemos escribir la etiqueta de cierre que en el ejemplo se encuentra al final del código ( `</html>` ). Además aparecen 3 atributos, el primero especifica el tipo de documento, y el segundo y el tercero deben llevar como valores el código del lenguaje en el que será escrito. Para la lengua castellana este valor es **es**. Puedes averiguar los [códigos de](#)

[otras lenguas.](#)

```
<head>
<title>Titulo de la pagina</title>
</head>
```

Con la etiqueta de apertura **<head>** comienza lo que se llama el encabezado de las páginas web, aquí se dan ciertas especificaciones y otra **META** información acerca del documento que ya veremos con más detalle. Pero entre ellas está el elemento utilizado en el ejemplo, mediante las etiquetas **<title></title>**, que encierran el título de la página. Los navegadores suelen mostrarlo cómo título de la ventana. También es utilizada por los buscadores cuando presentan resultados de los documentos a sus usuarios.

```
<body>
<p>Primer documento XHTML, es decir un, Hola mundo</p>
</body>
```

El cuerpo que comienza con **<body>** contiene todo el contenido que será representado por el agente de usuario. En nuestro ejemplo sólo utilizamos una etiqueta más, **<p>** que sirve para encerrar párrafos.

**NOTA:** Cada elemento puede ser un elemento de bloque o de línea (*block* e *inline* respectivamente). **p** es un elemento de tipo *block* ya que rompe la línea en el lugar dónde es incluido y crea su propio bloque, agregando un salto de línea por encima y por debajo de sus etiquetas de apertura y cierre. Otros elementos, cómo el elemento **strong**, son *inline* ya que pueden insertarse sin romper la línea.

Cerrando el cuerpo (**</body>**) y el documento XHTML (**</html>**) conseguimos nuestra primer página.

Al probar el ejemplo sólo verán el texto del párrafo, mas o menos así:

```
Primer documento XHTML, es decir un, Hola mundo
```

## 5. Ejemplo

[VER EJEMPLO01.html](#)

**NOTA:** Haciendo click en el ejemplo consultaran el ejemplo, pero interpretado por el

navegador. Pueden descargarlo para ver su código fuente.

[Página principal](#)

**1 aporte(s) de lectores:**

[eykO](#) aporta...

añadid algo: content type... xhtml debería servirse como xml y no como texto.

[7:45 PM](#) \_

[Publicar un comentario en la entrada](#)

**ATENCIÓN:** Por favor intenta aportar o preguntar sobre el tema que trata la página. Otro tipo de consultas o criticas puedes dejarlas en el [libro de visitas](#). Los aportes y preguntas están moderados simplemente para evitar spam o comentarios fuera de lugar.



Esta obra está; bajo una [licencia de Creative Commons](#).

Realizado por Emilio Astarita, [lee más acerca de este manual](#).



Web [manual-xhtml.blogspot.com](http://manual-xhtml.blogspot.com)

# Encabezados y marcas de importancia

- [Encabezados títulos](#)
- [Elemento strong](#)
- [Elemento em](#)

## Encabezados títulos

Los encabezados o títulos se pueden obtener mediante 6 etiquetas diferentes: **h1**, **h2**, **h3**, **h4**, **h5** y **h6** que tienen una jerarquía de importancia en la manera en que

se representan por los navegadores. Esto significa que **h1** será la etiqueta del título de mayor importancia lo que se traduce en, por ejemplo, un tamaño de letra mas grande, **h2** representaría un subtítulo con letra más pequeña y así sucesivamente hasta la etiqueta **h6**.

Para dar un ejemplo escribiremos un documento parecido al anterior, pero mostrando los 6 títulos posibles de mayor a menor.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<title>Encabezados y titulos</title>
</head>
<body>
<h1>Encabezado con etiqueta h1</h1>
<h2>Encabezado con etiqueta h2</h2>
<h3>Encabezado con etiqueta h3</h3>
<h4>Encabezado con etiqueta h4</h4>
<h5>Encabezado con etiqueta h5</h5>
<h6>Encabezado con etiqueta h6</h6>
</body>
</html>
```

Al probar el ejemplo deberíamos ver algo así:

# Encabezado con etiqueta h1

## Encabezado con etiqueta h2

### Encabezado con etiqueta h3

#### Encabezado con etiqueta h4

##### Encabezado con etiqueta h5



## Encabezado con etiqueta h6

[VER EJEMPLO02.html](#).

**NOTA:** Dependiendo del navegador los tamaños pueden variar, pero la jerarquía se mantendrá en todos.

## Elemento strong

Semánticamente **strong** marca contenido al que se le quiere dar más fuerza o importancia que el resto. Los navegadores suelen formatearlo en una tipografía tipo **negrita**, otros agentes de usuario podrían indicarlo de maneras diferentes. Es un elemento de *línea*.

Ejemplo:

```
<p>Superman es oriundo de <strong>Krypton</strong>, planeta actualmente  
desaparecido (debido a la radiacion).</p>
```

Resultado:

Superman es oriundo de **Krypton**, planeta actualmente desaparecido (debido a la radiacion).

## Elemento em

El elemento **em** agrega énfasis al contenido. Al igual que **strong** es un elemento de *línea*.

Ejemplo

```
<p>Una corriente filosofica afirma que lo que progresa es la <em>tecnologia</em>,  
el hombre no.</p>
```

## Resultado:

Una corriente filosófica afirma que lo que progresa es la *tecnología*, el hombre no.

**NOTA:** Usualmente los elementos con énfasis son formateados en tipografía tipo *itálica*.

[Página principal](#)

**0 aporte(s) de lectores:**

[Publicar un comentario en la entrada](#)

**ATENCIÓN:** Por favor intenta aportar o preguntar sobre el tema que trata la página. Otro tipo de consultas o críticas puedes dejarlas en el [libro de visitas](#). Los aportes y preguntas están moderados simplemente para evitar spam o comentarios fuera de lugar.



Esta obra está; bajo una [licencia de Creative Commons](#).

Realizado por Emilio Astarita, [lee más acerca de este manual](#).



Web [manual-xhtml.blogspot.com](http://manual-xhtml.blogspot.com)

## Enlaces, vínculos y anchor

- [1. Introducción](#)
- [2. Primer link](#)
- [3. Enlace a correo electrónico y otros recursos](#)
- [4. Enlaces relativos y absolutos](#)
- [5. Enlaces hacia anclas en la misma página](#)
- [6. El atributo title](#)
- [7. Atributos rel y rev](#)
  - [7.1. Valor nofollow para rel](#)
  - [7.2. Valor tag para rel \(Technorati\)](#)
- [8. Código con todos los ejemplos:](#)
- [9. Ejemplo](#)

# 1. Introducción

Los enlaces son los que hacen que naveguemos por la Web, es decir que vayamos saltando por diferentes páginas de un mismo sitio, o de sitios externos. La etiqueta utilizada para crear un vínculo es **a** (*anchor*).

No necesariamente un enlace apuntará a una página web, puede por ejemplo apuntar a una imagen, a un servidor FTP, a cualquier otro tipo de archivo, a un correo electrónico o incluso puede apuntar a hacia otros sectores de la misma página (previamente marcando estos sectores).

## 2. Primer link

Veamos un ejemplo sencillo:

```
<a href="http://www.example.com">Example</a>
```

En el navegador lo que veremos será: [Example](http://www.example.com). La dirección *http://www.example.com* es el valor del atributo **href**, la palabra *Example*, encerrada entre **<a>** y **</a>**, es lo que suele llamarse el *anchor text* y es el contenido que será visible en el navegador.

## 3. Enlace a correo electrónico y otros recursos

Para crear un enlace hacia un correo electrónico debemos apuntar en el valor del atributo **href** hacia una dirección de correo electrónico, pero es importante agregar al principio de la dirección el *mailto:* que vendría a representar el protocolo (*http://*, *ftp://*, etc).

Ejemplo:

```
<a href="mailto:mail@example.com">Correo de ejemplo</a>
```

Como en el caso anterior, veremos en nuestro navegador: [Correo de ejemplo](mailto:mail@example.com). Si prueban el enlace seguramente su navegador disparará el cliente de correo por defecto.

Si quieres saber más sobre que **URLs** puedes poner en el valor del atributo **href** puedes leer el artículo de la **Wikipedia**, [Uniform Resource Locator](#).

## 4. Enlaces relativos y absolutos

Muchas veces para enlaces entre páginas del mismo sitio y que tengan el mismo dominio, no hace falta utilizar enlaces absolutos. Por ejemplo si estamos en la pagina [www.example.com/universidades.html](#) y queremos enlazar a [www.example.com/subdirectorio/colegios.html](#) podemos crear un enlace de la siguiente manera:

```
<a href="subdirectorio/colegios.html">Ver los colegios</a>
```

Hay que destacar que en este caso hacer enlaces absolutos (escribiendo la dirección completa) funcionará también.

**NOTA:** Algunos buscadores recomiendan utilizar enlaces absolutos siempre que sea posible, para mejorar la indexación por parte de sus motores.

## 5. Enlaces hacia anclas en la misma página

Veamos ahora cómo crear marcas (algunos las llaman anclas) y enlaces para poder navegar en la misma página, sistema muy útil para crear un índice entre documentos largos.

La marca se realiza mediante el atributo **id** de la etiqueta **a**:

```
<h2><a id="privados"/>Colegios Privados</h2>
```

Hemos establecido un ancla llamada *privados* a la altura del título *Colegios Privados* ahora nos hará falta colocar un enlace desde otro sector de la página que apunte hacia esa ancla. Para esto:

```
<a href="#privados">Ir a Colegios Privados</a>
```

Noten el agregado del carácter numeral (#). Este se incluye ya que también podríamos construir el enlace utilizando la dirección absoluta lo que implicaría que necesitamos una separación entre las anclas y la dirección de la página, ya que la ancla tiene su propia dirección absoluta, que en este caso sería: *http://www.example.com/subdirectorio/colegios.html#privados*.

**NOTA 1:** El atributo *id* para marcar anclas, se puede utilizar en cualquier etiqueta, sin embargo puede que algunos navegadores no lo reconozcan como un ancla. Por eso se suele "anclar" sobre elementos **a**.

**NOTA 2:** Es posible también usar el atributo *name* de la misma manera que *id*. La cuestión se reduce a que algunos viejos navegadores, es posible que no reconozcan el atributo *id*.

## 6. El atributo title

La etiqueta **a** tiene un atributo de importante valor semántico, llamado *title*. El atributo *title* sirve para especificar una pequeña descripción del sitio que estamos enlazando.

Los navegadores suelen desplegar un cartelito con el valor de este atributo cuando el usuario señala con el mouse el enlace. Esto es de gran ayuda a los usuarios, que decidirán si seguir el enlace o no.

Se utiliza así:

```
<a href="subdirectorio/colegios.html" title="Lista de Colegios privados de la zona sur">Colegios Privados</a>
```

## 7. Atributos rel y rev

Estos atributos sirven para enriquecer semánticamente a los enlaces. Sus objetivos son funcionalmente inversos.

El atributo **rel**, sirve para definir el tipo de relación que tiene el documento

contenedor del enlace con el que éste, está apuntando. Por ejemplo podríamos utilizarlo de la siguiente manera:

```
<a href="pagina_04.html" title="Ir a la pagina 4" rel="next">pagina 4</a>
```

El enlace anterior sería muy conveniente usarlo, por ejemplo, en la página *pagina\_03.html* ya que estaríamos indicando semánticamente que estamos enlazando con la página que continua el contenido de la actual.

Inversamente podríamos indicar en *pagina\_04.html*, mediante el elemento **link** y su atributo **rev** que *pagina\_03.html* es el documento que precede al actual, para esto le damos el valor *previus*.

Veán también una [descripción detallada del uso de de rel y rev](#).

## 7.1. Valor nofollow para rel

Un valor del atributo **rel** que ha tomado mucha popularidad en este último tiempo, es el llamado **nofollow**. Este fue propuesto por los buscadores cómo una manera de indicarles que el enlace que lo lleva puesto no debe ser considerado en los algoritmos que calculan el posicionamiento de los resultados de búsquedas. El objetivo de poder dar tal indicación, es luchar contra [spammers](#) que suelen, entre otras cosas, promocionar sus webs mediante la inserción de enlaces en sitios cómo la [Wikipedia](#) o en los comentarios de los [Weblogs](#).

Ejemplo de su uso:

```
<a href="http://www.microsoft.com" title="Sitio que vende software privativo" rel="nofollow">Microsoft</a>
```

## 7.2. Valor tag para rel (Technorati)

Mencionemos ahora otro valor para **rel** que puede ser utilizado para agregar etiquetados (tags) a nuestras páginas, y que es mayoritariamente utilizado por los [blogs](#). El valor es justamente llamado: **tag** y su uso es alentado por sitios como [Technorati](#) que cuentan con tecnología para analizar este tipo de enlaces y poder categorizar nuestras webs mediante la aplicación de etiquetas (tags).

Colocando la siguiente línea de código en una página web estaremos automáticamente dándole la oportunidad a Technorati de asociar tal página con la etiqueta *animaciones*:

```
<a href="http://www.example.com/png/animaciones" rel="tag">animaciones</a>
```

## 8. Código con todos los ejemplos:

Ahora construyamos una página completa con todos estos ejemplos:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<title>Vínculos, hipervínculos, enlaces</title>
</head>
<body>
<h1><a id="arriba" />Enlaces</h1>
<p><a href="http://www.example.com" title="Apunta hacia example.
com">Enlace absoluto</a></p>
<p><a href="ejemplo02.html" title="Ir al ejemplo numero 2"
rel="Prev">Enlace relativo</a></p>
<p><a href="#fin" title="Ir al final de la pagina mediante el ancla">Enlace
hacia ancla</a></p>
<p><a href="mailto:mail@example.com">Correo de ejemplo</a></p>
<p><a href="ftp://ftp.gnu.org" title="Enlace al ftp del sitio GNU">FTP de
GNU</a></p>
<p><a href="http://www.microsoft.com" title="Sitio que vende software
privativo" rel="nofollow">Enlace con NoFollow</a></p>
<p><a href="http://technorati.com/tag/xhtml" rel="tag">xhtml</a> </p>
<!--Forzamos varios saltos de linea para notar el uso de los enlaces mediante
anclas.-->
<br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br />
<br /><br /><br /><br />
<br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br />
```



```

<br /><br /><br /><br />
<br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br />
<br /><br /><br /><br />
</p>
<h3><a id="fin" />Sospecho que junto a mi hay un ancla.</h3>
<p><a href="#arriba" title="Volver a arriba">Volver</a></p>
</body>
</html>

```

## 9. Ejemplo

[Ver EJEMPLO03.HTML](#)

**NOTA:** Dentro de un párrafo vacío incluimos varios saltos de línea para que una de las anclas se desplace lo suficiente de la otra, para notar el efecto de navegar en la misma página.

Sigue habiendo muchos temas a tratar sobre **enlaces**, espero con el tiempo poder seguir ampliando este documento.

[Página principal](#)

**1 aporte(s) de lectores:**

Anónimo aporta...

Hola,

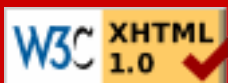
solo un saludo y agradecerte por el excelente tutorial.

Un poco corto, pero me ha sido muy util. Un abrazo.

[8:00 AM](#) \_

## [Publicar un comentario en la entrada](#)

**ATENCIÓN:** Por favor intenta aportar o preguntar sobre el tema que trata la página. Otro tipo de consultas o críticas puedes dejarlas en el [libro de visitas](#). Los aportes y preguntas están moderados simplemente para evitar spam o comentarios fuera de lugar.



Esta obra está; bajo una [licencia de Creative Commons](#).

Realizado por Emilio Astarita, [lee más acerca de este manual](#).



Web [manual-xhtml.blogspot.com](http://manual-xhtml.blogspot.com)

## Citas, abreviaturas y acrónimos

- [1. Citas](#)
- [2. Citas con <blockquote>](#)
- [3. Citas con <cite>](#)
- [4. Abreviaturas](#)
- [5. Acrónimos](#)
- [6. Código del ejemplo](#)
- [7. Ejemplo](#)

### 1. Citas

En muchos documentos citaremos texto desde otras fuentes. Para especificar que

estamos citando algún contenido desde otra fuente disponemos de dos etiquetas que se diferencian entre sí en el modo en que se presentará el contenido citado dentro del documento.

## 2. Citas con `<blockquote>`

La etiqueta `<blockquote>`, generalmente se utiliza para contener un párrafo ( `p` ) con un texto citado. El contenido citado se contendrá en un bloque que usualmente el navegador lo presentará con una tabulación mayor al resto de los párrafos.

Ejemplo:

```
<blockquote>
```

```
<p>
```

Linus Benedict Torvalds (born December 28, 1969 in Helsinki, Finland) is a Finnish software engineer best known for initiating the development of Linux.

He now acts as the project's coordinator (or Benevolent Dictator for Life).

```
</p>
```

```
<p>
```

Linus was inspired by Minix (a kernel and operating system developed by Andrew Tanenbaum) to develop a capable UNIX-like operating system that could be run on a PC. Linux now also runs on many other architectures.

```
</p>
```

```
</blockquote>
```

Resultado:

Linus Benedict Torvalds (born December 28, 1969 in Helsinki, Finland) is a Finnish software engineer best known for initiating the development of Linux. He now acts as the project's coordinator (or Benevolent Dictator for Life).

Linus was inspired by Minix (a kernel and operating system developed by Andrew Tanenbaum) to develop a capable UNIX-

like operating system that could be run on a PC. Linux now also runs on many other architectures.

La anterior cita, tendrá dos saltos de línea, por encima y por debajo, de la apertura y del cierre del elemento **blockquote** respectivamente, ya que conforma un bloque. Si queremos citar sin estos saltos de línea debemos usar el elemento **cite**.

### 3. Citas con <cite>

Veamos ahora como citar en la misma línea en la que estamos escribiendo, que suele utilizarse para pequeños trozos de texto.

Ejemplo:

Mi padre sobre Linus Benedict Torvalds alguna vez supo decir <b>cite>¿Quién es ese?</cite>

Resultado:

Mi padre sobre Linus Benedict Torvalds alguna vez supo decir *¿Quién es ese?*

Al utilizar este elemento la línea no se interrumpe como sucedería con **blockquote** que podría resultar molesto para una cita tan corta.

### 4. Abreviaturas

Al escribir la abreviatura de una palabra podemos valernos del elemento **abbr** para marcarla e indicar en el atributo **title** qué palabra estamos abreviando, los navegadores en general se encargarán de que al apoyar el puntero del mouse sobre la abreviatura se muestre un cartel con la palabra no abreviada que indicamos.

Ejemplo:

<abbr title="Telefono">Tel</abbr>

Resultado:

Tel

## 5. Acrónimos

Los acrónimos se marcan con el elemento **acronym**, tienen el atributo **title**, y generalmente, el mismo comportamiento que **abbr** en los navegadores. Ejemplo:

Mi facultad es parte de la `<acronym title="Universidad Tecnologica Nacional">UTN</acronym>`  
 Esta página sigue los estándares de la `<acronym xml:lang="en" title="World Wide Web Consortium">W3C</acronym>`

Resultado:

Mi facultad es parte de la UTN. Esta página sigue los estándares de la W3C

**NOTA:** Fijarse que en la segunda línea introducimos el valor *en* para el atributo **xml:lang**, este le indica que el acrónimo está escrito en lengua inglesa. Generalmente todas nuestras etiquetas soportan tal atributo, sin embargo no siempre es necesario escribirlo ya que las etiquetas heredan el valor especificado del elemento **<html>** (u otros elementos superiores). En nuestro caso el valor puesto en el elemento **html** era *"es"* por lo que nos vemos obligados a cambiarlo en esta etiqueta específica.

## 6. Código del ejemplo

El siguiente código es el de un documento XHTML que contiene todos los ejemplos expuestos:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
```

```

"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
<head>
<title>Vinculos, hipervinculos, enlaces</title>
</head>
<body>
<h1><a href="http://manual-xhtml.blogspot.com/" title="Ir al manual">Manual
XHTML</a></h1>
<h2>Citas abreviaturas y acrónimos</h2>
<h3>Cita:</h3>
<p>Cita con <em>blockquote</em> de la <a href="http://en.wikipedia.org/
wiki/Linus_Torvalds" title="ir a la fuente"> Wikipedia</a>:</p>
<blockquote>
<p>
Linus Benedict Torvalds (born December 28, 1969 in Helsinki, Finland) is a
Finnish software engineer best known for initiating the development of Linux.
He now acts as the project's coordinator (or Benevolent Dictator for Life).
</p>
<p>
Linus was inspired by Minix (a kernel and operating system developed by
Andrew Tanenbaum) to develop a capable UNIX-like operating system that
could be run on a PC. Linux now also runs on many other architectures.
</p>
</blockquote>
<h3>Abreviatura:</h3>
<p>Mi padre sobre Linus Benedict Torvalds alguna vez supo decir
<cite>¿Quién es ese?</cite></p>
<p><abbr title="Recursos Humanos">RRHH</abbr> <abbr
title="Telefono">Tel</abbr> 4000-0000</p>
<h3>Acronimo:</h3>
<p>Mi facultad es parte de la <acronym title="Universidad Tecnologica
Nacional">UTN</acronym></p>
</body>
</html>

```

## 7. Ejemplo

[Ver el EJEMPLO04.HTML](http://manual-xhtml.blogspot.com/2006/05/citas-abreviaturas-y-acr-nimos.html)

[Página principal](#)

## 7 aporte(s) de lectores:

Anónimo aporta...

El elemento abbr en las abreviaturas... no muestra cual es la palabra abreviada, en cambio el acronimo si... eso.. es asi o es porque te has equivocado??

[2:31 PM](#) \_

[Sr.Bit](#) aporta...

No entiendo tu pregunta. **abbr** y **acronym**, mostrarán el valor de *title* según el agente de usuario. En Firefox y Konqueror esto funciona correctamente. Si no te referías a esto explicáте mejor porque no he entendido. Saludos.

[3:32 PM](#) \_

Anónimo aporta...

Lo que sucede es que el elemento **abbr**, no se aprecia en el navegador IE. Me sucede lo mismo que la persona que efectuó el anterior comentario. Héctor.

[4:08 PM](#) \_

Anónimo aporta...

pero no funciona en IE

[4:09 PM](#) \_

[Guille](#) aporta...



pues usa un navegador decente :-) yo lo probe en firefox 1.5.0.4 y funciona muy bien..

[1:38 AM](#) \_

Anónimo aporta...

...la etiqueta **abbr** no funciona en IE por lo es importante, si quieren señalar alguna abreviatura, que utilicen alguna variante para ello (por ejemplo, la etiqueta **acronym**>, **em**.. o **span class="abbrev"** y especificar en la hoja de estilo utilizada el formato que desean para la clase .abbrev ...)

... con concuerdo con el guille... verdad que el IE explorer no es de los mejorcitos (y particularmente no acostumbro a usarlo para mis asuntos) pero es uno de los mas usados en el mundo y si desarrollas para la web tienes que tener en cuenta esto...

saludos

[10:12 PM](#) \_

[Juan José](#) aporta...

Si bien es cierto que el IE es el navegador más usado (por ahora), el Manual es de XHTML, por ese motivo, se explica el uso del abbr. Si IE no sirve para visualizar documentos XHTML, problema del IE. Si no quiere quedarse en el tiempo, Microsfot va a tener que actualizarse...

Si uno tiene que desarrollar una página en XHTML, la gente del IE se va a comenzar a quedar fuera de ciertos sitios, y, por lógica, o no va a usar esos sitios, o no va a usar IE... y si realmente le interesa acceder a la información... el IE se va a caer.

Saludos!!!! ^\_^

[2:01 PM](#) \_

[Publicar un comentario en la entrada](#)

**ATENCIÓN:** Por favor intenta aportar o preguntar sobre el tema que trata la página.

Otro tipo de consultas o críticas puedes dejarlas en el [libro de visitas](#). Los aportes y preguntas están moderados simplemente para evitar spam o comentarios fuera de lugar.



Esta obra está; bajo una [licencia de Creative Commons](#).

Realizado por Emilio Astarita, [lee más acerca de este manual](#).



Web [manual-xhtml.blogspot.com](http://manual-xhtml.blogspot.com)

## Imágenes

- [1. Imágenes formatos](#)
- [2. El elemento img](#)
- [3. Altura y ancho](#)
- [4. Alineación de las imágenes \(con atributo align\)](#)
- [5. Alineación de las imágenes \(con propiedad float\)](#)
  - [5.1. float: right](#)
  - [5.2. float: left](#)

### 1. Imágenes formatos

Además de texto, podemos incorporar al contenido de nuestros documentos

XHTML archivos de imágenes que pueden contener fotografías, dibujos, diagramas, etc.

Hay una seria discusión acerca de que formato de imágenes es conveniente utilizar para el empleo en la web, que incluye temas variados y complejos cómo: *algoritmos de compresión de imágenes*, *patentes*, *calidad*, y *compatibilidad*. Personalmente me inclino por el uso de las imágenes libres de patentes **PNG**. Perto te recomiendo que amplies el tema para que puedas decidir que tipo de imagen utilizar según el caso, en los siguientes artículos de la [Wikipedia](#):

- [Formato PNG](#)
- [Formato JPEG](#)
- [Formato GIF](#)

## 2. El elemento img

El elemento para insertar imágenes es **img**. Su principal atributo es *src* (de source). Este atributo lleva como valor la dirección (relativa o absoluta) de la ubicación de la imagen.

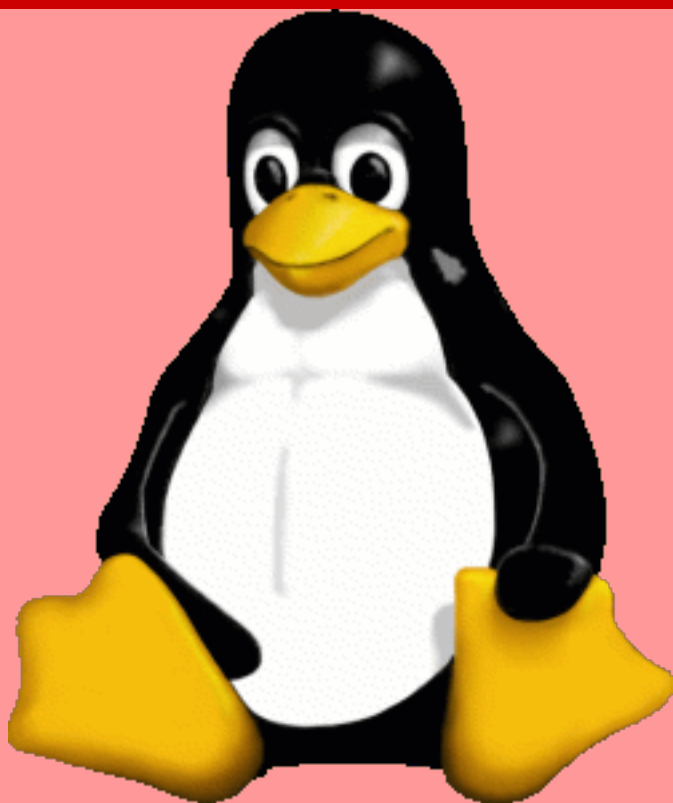
Ejemplo de uso de **img**:

```

```

Resultado:





Notar la barra antes de cerrar el parentesis angular. Como ya hemos mencionado esta es necesaria ya que en XHTML todos los elementos deben cerrarse aunque no contengan contenido, por eso utilizamos esta sintaxis para abrir y cerrar el elemento en la misma etiqueta.

Además hemos puesto el valor *Tux, el pinguino de Linux* al atributo **alt**, este debe explicar en palabras el contenido de la fotografía. Los agentes de usuario basados en texto mostrarán el valor del atributo en vez de la imagen, otros agentes de usuarios para no videntes podrían leer y reproducir con un sintetizador el valor del atributo. Es muy recomendable su uso, ya que además si la imagen por cualquier motivo no puede ser mostrada, se presentará el valor de **alt** en lugar de la imagen.

### 3. Altura y ancho

Podemos además especificar la altura y el ancho de la imagen, mediante el uso de los atributos **height** y **width** respectivamente.

En el ejemplo anterior mostramos cómo incorporar una imagen a nuestro documento sin especificar estos atributos, el problema de hacerlo de ese modo, es que el agente de usuario no podrá calcular el espacio que ocupará la imagen hasta que no termine de obtenerla. Esto podría resultar en una carga más lenta del

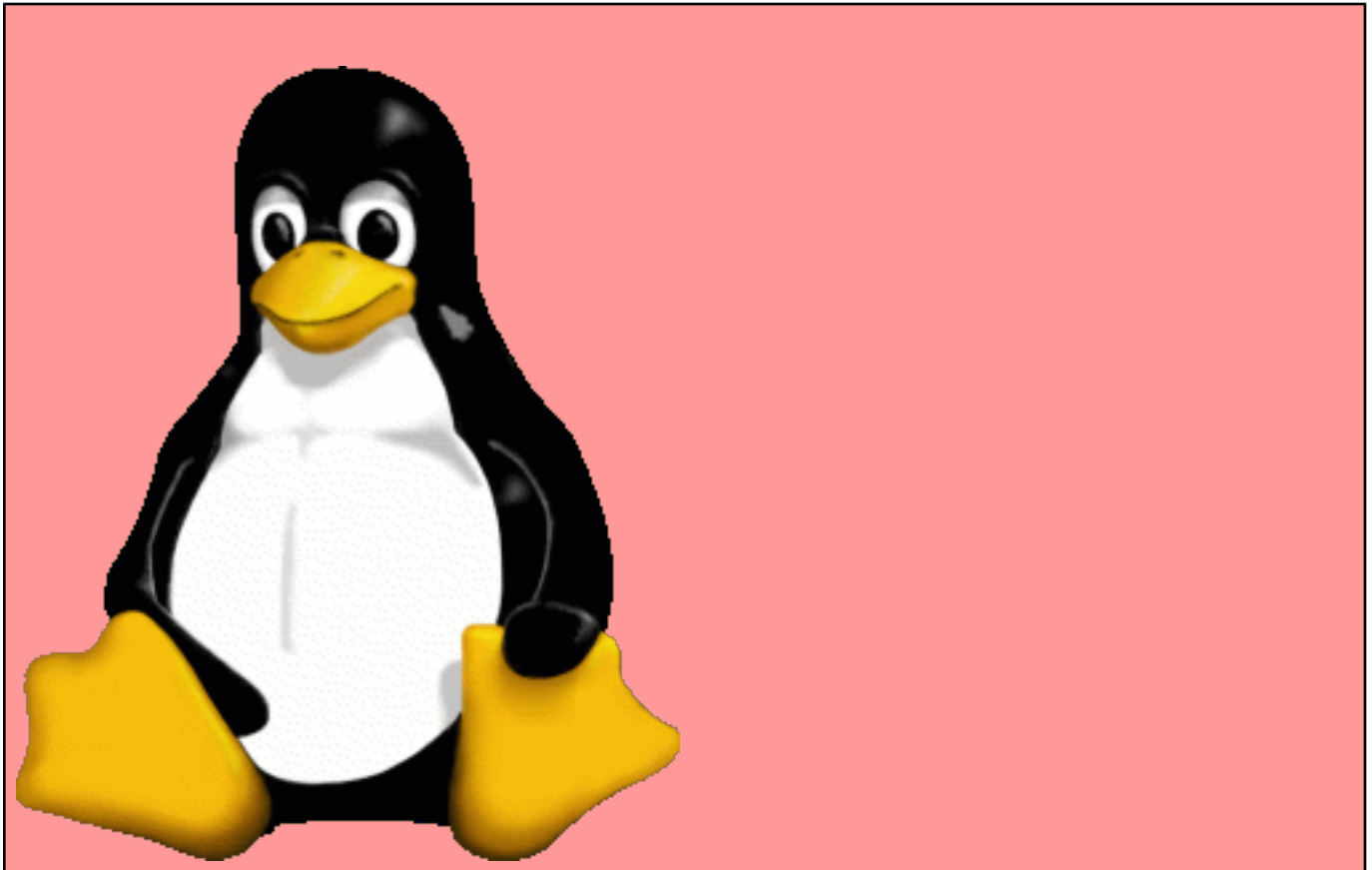
renderizado final de la página. Es muy conveniente averiguar el tamaño de nuestras imágenes en [píxeles](#) y especificarlo en los atributos **height** y **width**.

Ejemplo:

```

```

El resultado es el mismo que antes:



Otro motivo para el uso de estos atributos, podría ser que querramos un cambio en las dimensiones y no usar el tamaño de la imagen original, para tal, tengamos en cuenta las siguientes consideraciones:

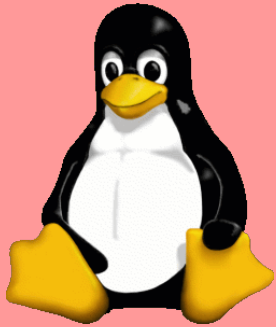
Si achicamos el ancho y la altura, lo haremos sólo en la apariencia del documento. El archivo mantendrá el tamaño original, por lo que podríamos estar gastando recursos innecesariamente. Lo conveniente sería editar y reajustar el tamaño de la imagen con algún software de edición de imágenes (por cierto [The Gimp](#) es muy bueno y es [software GNU](#)).

Ejemplo:

```

```

Resultado:



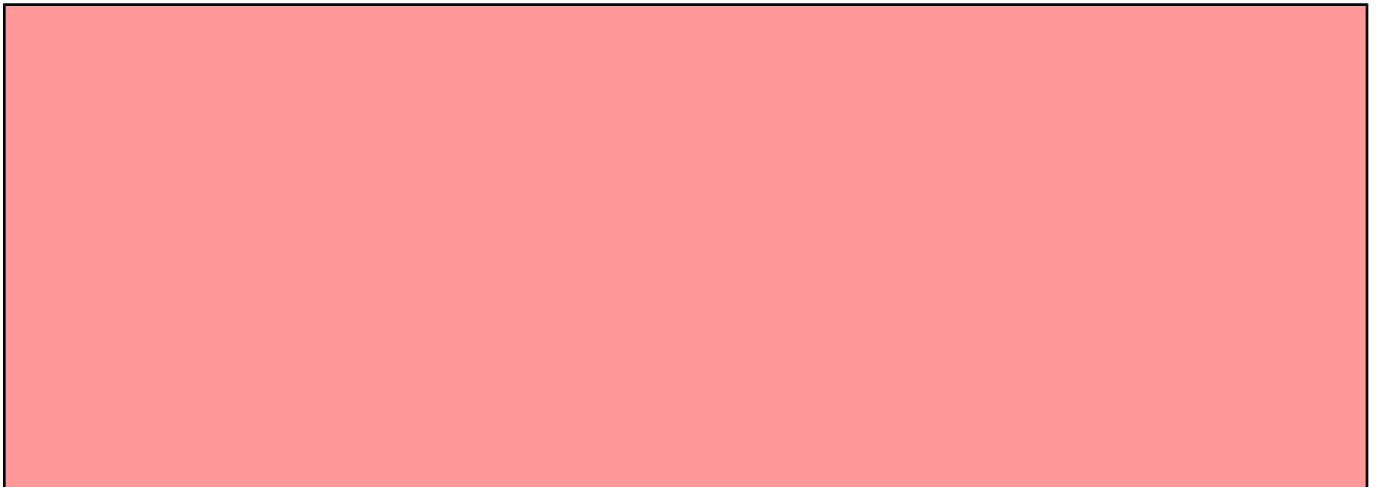
El caso inverso (aumentar las dimensiones) podría lograr un efecto no deseado que es la "pixelización" (algo así como una notable pérdida de calidad) de la imagen. De nuevo es recomendable conseguir imágenes apropiadas y no valerse de estos atributos para obtener imágenes más grandes.

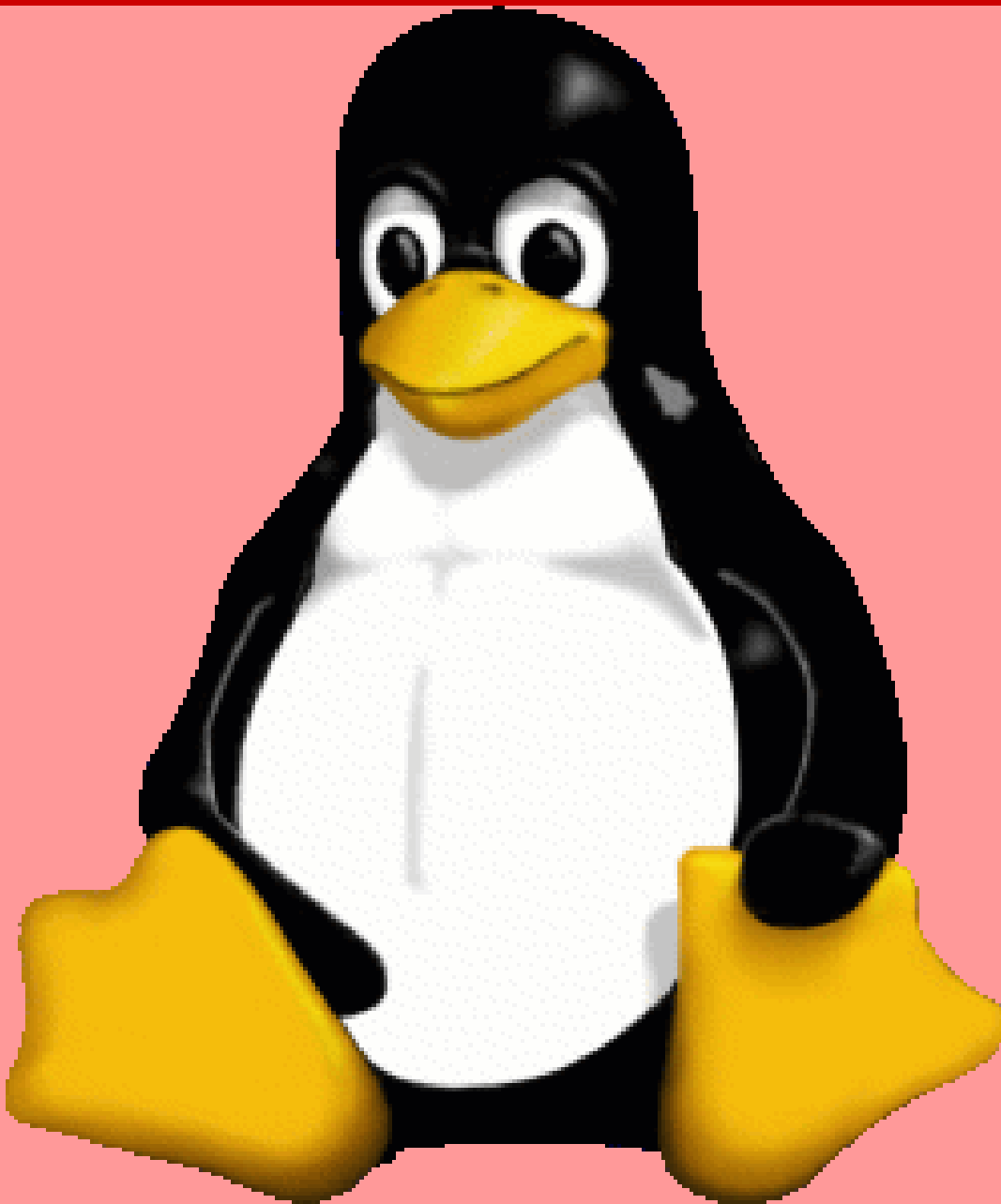
Ejemplo:

```

```

En el resultado noten los bordes "pixelados":





Observemos también que deberíamos mantener las proporciones al cambiar los valores originales de la imagen mediante `height` y `width`, si no deseamos deformar el aspecto original. Una buena idea podría ser sólo reducir o aumentar uno sólo de los valores, luego el agente de usuario calculará y asignará el valor correspondiente a esa proporción para el atributo no especificado (aunque para calcular el espacio debería obtener la imagen primero, con la consecuente pérdida de tiempo en el renderizado final del documento).

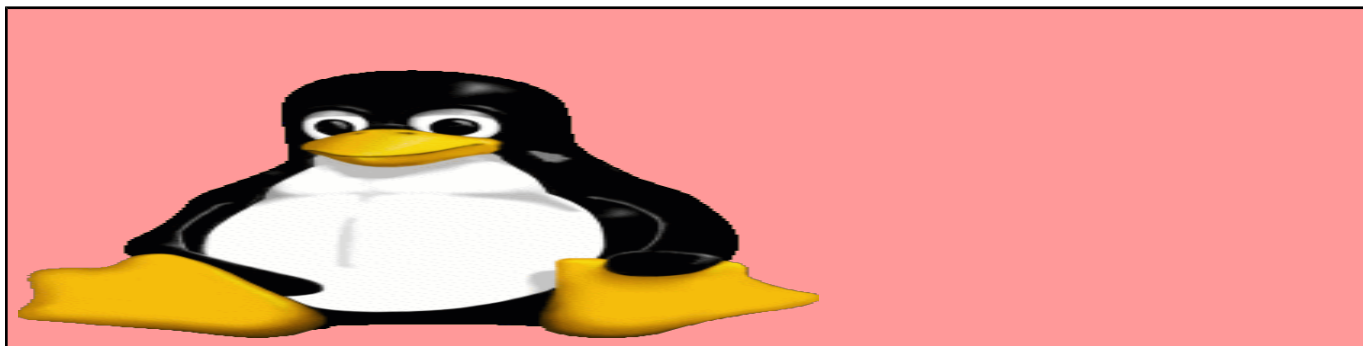
Ejemplo:

```

```



El resultado es la deformación de la imagen:

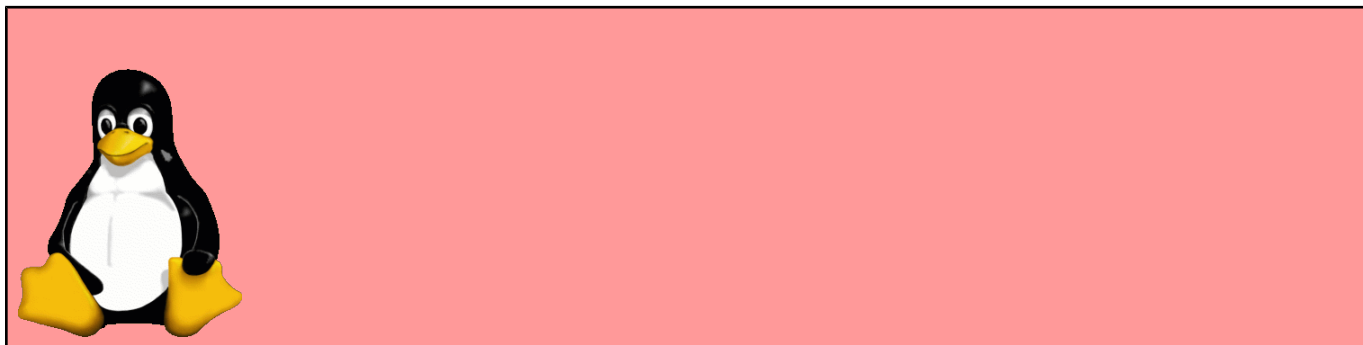


Con un valor ausente:

```

```

Ahora se mantiene la proporción ya que el valor ausente de **width** es calculado automáticamente:



## 4. Alineación de las imágenes (con atributo align)

El elemento **img** también soporta el atributo **align**, su uso no es recomendado, y se recomienda el uso de la propiedad float de **CSS**. Sin embargo esta propiedad aún no es soportada por todos los navegadores, veremos ejemplos del uso del atributo **align**.

**NOTA:** Para mantener este documento acorde al estricto uso de elementos **XHTML 1.0**, se mostrarán en otra página los usos del atributo **align**

[Uso del atributo align](#) (en otra página)

## 5. Alineación de las imágenes (con propiedad float)

Para acomodar las imágenes sin el uso del atributo **align**, podemos valernos del atributo **style** que permite especificar propiedades **CSS** de un elemento, en este caso **img**.

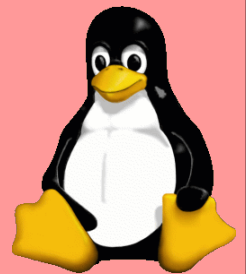
Veamos algunos ejemplos:

### 5.1. float: right

```
<p>....</p>
```

Si en lugar de los 3 puntos ponemos algo de texto, obtendríamos este resultado:

Tux es el nombre de la mascota oficial del kernel Linux. Tux es un pequeño pingüino de aspecto risueño. La idea de que la mascota de Linux fuera un pingüino provino del mismo Linus Torvalds, creador de Linux. Según se cuenta, cuando era niño le picó un pingüino, y le resultó simpática la idea de asociar un pingüino a su proyecto. Existen dos versiones sobre el origen de su nombre. La primera sugiere que el nombre surge del hecho de que los pingüinos parecen vestir un esmoquín (en inglés tuxedo). La segunda es que las letras que componen Tux provienen de las palabras Torvalds Unix.



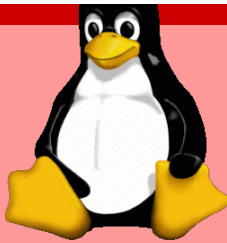
### 5.2. float: left

```
<p>....</p>
```

De nuevo un resultado parecido, pero alineada la imagen en el otro extremo:



Tux es el nombre de la mascota oficial del kernel Linux. Tux es un



pequeño pingüino de aspecto risueño. La idea de que la mascota de Linux fuera un pingüino provino del mismo Linus Torvalds, creador de Linux. Según se cuenta, cuando era niño le picó un pingüino, y le resultó simpática la idea de asociar un pingüino a su proyecto. Existen dos versiones sobre el origen de su nombre. La primera sugiere que el nombre surge del hecho de que los pingüinos parecen vestir un esmoquín (en inglés tuxedo). La segunda es que las letras que componen Tux provienen de las palabras Torvalds Unix.

[Página principal](#)

**0 aporte(s) de lectores:**

[Publicar un comentario en la entrada](#)

**ATENCIÓN:** Por favor intenta aportar o preguntar sobre el tema que trata la página.

Otro tipo de consultas o criticas puedes dejarlas en el [libro de visitas](#). Los aportes y preguntas están moderados simplemente para evitar spam o comentarios fuera de lugar.



Esta obra está; bajo una [licencia de Creative Commons](#).

Realizado por Emilio Astarita, [lee más acerca de este manual](#).



Web manual-xhtml.blogspot.  
com

## Listas

- [1. Introducción](#)
- [2. Listas ordenadas y numeradas](#)
  - [2.1. Diferentes tipos de listas ordenadas](#)
- [3. Listas desordenadas](#)
  - [3.1. Diferentes tipos de listas desordenadas](#)
- [4. Listas de definiciones](#)
- [5. Listas compuestas mediante anidamiento](#)

### 1. Introducción

Las listas ofrecen la posibilidad de presentar cierto tipo de información de una manera útil y simple. No sólo para ordenarla, sino también entre otras cosas para jerarquizarla, o numerarla. Para estos usos, **XHTML** nos permite construir diferentes tipos de listas, ya sea simples, o bien compuestas utilizando anidamiento de etiquetas.

Por ofrecer un ejemplo inmediato, resaltamos que los índices mismos de este manual utilizan listas para mostrarse.

### 2. Listas ordenadas y numeradas

Al utilizar este tipo listas, los agentes de usuario se encargarán de agregar una numeración a cada ítem que por defecto suele ser decimal, creciente y empezando en 1.

Veamos el siguiente código:

```
<ol>
```

```
<li>Correr el script llamado "borrar_casi_todo_usr_bin.sh"</li>
<li>Correr el comando, "apt-get install emacs21"</li>
<li>Correr el comando, "emacs".</li>
<li>Acostumbrarse al nuevo SO, luego conquistar el mundo.</li>
</ol>
```

Este código representa una lista ordenada numericamente, con instrucciones potenciales para conquistar el mundo.

Veamos cómo queda formateada:

1. Correr el script llamado "borrar\_casi\_todo\_usr\_bin.sh"
2. Correr el comando, "apt-get install emacs21"
3. Correr el comando, "emacs".
4. Acostumbrarse al nuevo SO, luego conquistar el mundo.

El elemento **ol** contiene los ítems de la lista que a su vez están contenidos por el elemento **li**. A cada ítem introducido se le asignará un número empezando por 1.

## 2.1. Diferentes tipos de listas ordenadas

Podemos utilizar el atributo **type** para cambiar el sistema de numeración de las listas ordenadas. Los valores soportados por **type** son: *A*, *a*, *I* (por defecto) y *I*. Sin embargo el cambio de estas propiedades son parte del diseño por lo que no sería correcto su uso en un documento estricto. Más adelante veremos que con el uso de hojas de estilo CSS, podremos usar aún más tipos, y manteniendo la correcta separación del contenido y el diseño del documento.

Por el momento pueden ver: [ejemplos del uso del atributo type](#) en otra página.

## 3. Listas desordenadas

Si cambiamos el elemento **ol** por el elemento **ul**, obtendremos un efecto parecido, sólo que ésta vez las listas no se presentarán con ningún tipo de ordenamiento o numeración, sino con un viñetado.

Por ejemplo observemos este código:

```
<ul>
<li><a href="http://www.frba.utn.edu.ar">Universidad Tecnológica de Buenos Aires</a></li>
<a><a href="http://www.uba.ar">Universidad de Buenos Aires</a></li>
<li><a href="http://www.itba.edu.ar">Instituto tecnológico de buenos aires</a></li>
</ul>
```

Que va a quedar representado así:

- [Universidad Tecnológica de Buenos Aires](http://www.frba.utn.edu.ar)
- [Universidad de Buenos Aires](http://www.uba.ar)
- [Instituto tecnológico de buenos aires](http://www.itba.edu.ar)

Cómo ven tenemos una lista construida con **ul** y dónde además cada uno de sus ítems tiene anidado un enlace. No hay ningún tipo de numeración y aunque el orden en como fueron introducidos los ítems se mantiene, no tiene

ninguna importancia (recordemos que no hablamos de representación sino de valor semántico, si el orden importara deberíamos usar **ol**).

### 3.1. Diferentes tipos de listas desordenadas

De nuevo debemos decir que especificar tipos de listas mediante el atributo **type**, no es recomendable. Por el momento veremos como hacerlo en otra página. Luego veremos de que manera es conveniente mediante la aplicación de hojas de estilo CSS.

Ver [tipos de listas desordenadas mediante type](#) en otra página.

## 4. Listas de definiciones

Con este tipo es posible construir listas de definiciones. Cada ítem de una lista de definición contiene dos secciones, la primera es el contenido (ej. una palabra, una imagen, etc) a definir, y la segunda la definición de la primera. Se construyen con el elemento **dl**.

Ejemplo de código:

```
<dl>
<dt>p</dt>
<dd>Elemento de bloque, para construir párrafos.</dd>
<dt>strong</dt>
<dd>Elemento de línea, para remarcar la importancia de un contenido.</dd>
<dt>em</dt>
<dd>Elemento de línea, para enfatizar parte del contenido.</dd>
</dl>
```

La cual se representará así:

```
p
    Elemento de bloque, para construir párrafos.
strong
    Elemento de línea, para remarcar la importancia de un contenido.
em
    Elemento de línea, para enfatizar parte del contenido.
```

Cómo sospecharán, el elemento **dt** marca el término a definir, y el elemento **dd**, marca la definición del anterior.

Debido a que no llevan ni viñetado ni numeración las listas de definición no tienen diferentes tipos, aunque sí podremos modificar su apariencia con la aplicación de hojas de estilo.

## 5. Listas compuestas mediante anidamiento

Podemos anidar varias listas a otras y de diferentes tipos sin restricciones. La sintaxis es sencilla. Supongamos que tenemos una lista ordenada, si quisiéramos anidarle otra lista desordenada, deberíamos crear un ítem en la más externa (la ordenada) que hará de título, pero antes de cerrar este ítem con la etiqueta **</li>** debemos poner el código de la lista ordenada.

Ejemplo de una lista ordenada con 3 ítems, el primero normal, el segundo con una lista de definición anidada, y el

tercero con otra lista desordenada (que a su vez tiene otro grado de anidamiento). Lo dicho:

```
<ol>
  <li>Item normal (lista ordenada)</li>
  <li>Item titulo
    <dl>
      <dt>Definición1</dt>
      <dd>Me llaman la definición 1 :).</dd>
      <dt>Definición2</dt>
      <dd>Me llaman la definición 2 :).</dd>
      <dt>Definición3</dt>
      <dd>Me llaman la definición 3 :).</dd>
    </dl>
  </li>
  <li>Item título con lista anidada.
    <ul>
      <li>Primer item (lista desordenada) segundo grado de anidamiento.</li>
      <li>Segundo item (lista desordenada) segundo grado de anidamiento.</li>
      <li>Item título de segundo grado.
        <ul>
          <li>Primer item (lista ordenada) tercer grado de anidamiento.</li>
        </ul>
      </li>
    </ul>
  </li>
</ol>
```

Y el resultado es:

1. Item normal (lista ordenada)
2. Item titulo
  - Definición1
    - Me llaman la definición 1 :).
  - Definición2
    - Me llaman la definición 2 :).
  - Definición3
    - Me llaman la definición 3 :).
3. Item título con lista anidada.
  - Primer item (lista desordenada) segundo grado de anidamiento.
  - Segundo item (lista desordenada) segundo grado de anidamiento.
  - Item título de segundo grado.
    - Primer item (lista ordenada) tercer grado de anidamiento.

[Página principal](#)

**1 aporte(s) de lectores:**

Anónimo aporta...

```
<'a'><a href="http://www.uba.ar">Universidad de Buenos Aires"</a'></li'>"
```

se a colado una **a** por el **li** :P

[6:48 PM](#) \_

[Publicar un comentario en la entrada](#)

**ATENCIÓN:** Por favor intenta aportar o preguntar sobre el tema que trata la página. Otro tipo de consultas o críticas puedes dejarlas en el [libro de visitas](#). Los aportes y preguntas están moderados simplemente para evitar spam o comentarios fuera de lugar.



Esta obra está; bajo una [licencia de Creative Commons](#).

Realizado por Emilio Astarita, [lee más acerca de este manual](#).





Web [manual-xhtml.blogspot.com](http://manual-xhtml.blogspot.com)

## Texto preformateado

- [1. Elemento pre](#)
- [2. Elemento code](#)

### 1. Elemento pre

En algunas ocasiones podríamos necesitar incluir el contenido de un archivo de texto sin perder el formato, es decir conservando los saltos de línea y el espaciado original. Para esto contamos con el elemento **pre**, que permite representar texto ya preformateado.

Uno de los usos más comunes de este elemento, es la inclusión de código fuente

dentro de nuestros documentos. Los navegadores, suelen representar el contenido de la etiqueta **pre** mediante una fuente monoespaciada.

Ejemplo incluyendo código fuente:

```
<pre>
class Libro {
    public:
        Libro();
        void establecerTitulo(char *);
        void establecerPaginas(int);
        void establecerCodigo(int);
        void imprime();
    private:
        char Titulo[40];
        int Cantpaginas;
        int Codigo;
};
</pre>
```

Y el resultado sería:

```
class Libro {
    public:
        Libro();
        void establecerTitulo(char *);
        void establecerPaginas(int);
        void establecerCodigo(int);
        void imprime();
    private:
        char Titulo[40];
        int Cantpaginas;
        int Codigo;
};
```

**NOTA:** Observar que se respetan en la representación todos los espacios y tabulaciones incluidos, y que no hace falta el uso del elemento de salto de línea **br**.

## 2. Elemento code

Hay que tener en cuenta que **pre** es un elemento en *bloque*, por lo que no se puede incluir en una línea sin romperla. El elemento equivalente en *línea* es **code**, que obviamente no conserva el formato (ya que no puede romper la línea), pero sigue siendo una indicación semántica de la inclusión de código. Además los agentes de usuario suelen representar el contenido de **code** con letra monoespaciada.

Ejemplo de uso de code:

```
<p>En el ejemplo anterior incluimos el codigo fuente (en c++) de una clase llamada <strong>Libro</strong>, con las funciones miembro: <code>Libro()</code>, <code>void establecerTitulo(char *)</code> y <code>void imprime()</code> entre otras.</p>
```

El resultado de este párrafo sería:

```
En el ejemplo anterior incluimos el codigo fuente (en c++) de una clase llamada Libro, con las funciones miembro: Libro(), void establecerTitulo(char *) y void imprime() entre otras.
```

Cómo notaran incluimos varios etiquetados con **code** que no rompieron el bloque del elemento superior, párrafo (**p**).

[Página principal](#)

**0 aporte(s) de lectores:**

### [Publicar un comentario en la entrada](#)

**ATENCIÓN:** Por favor intenta aportar o preguntar sobre el tema que trata la página.

Otro tipo de consultas o criticas puedes dejarlas en el [libro de visitas](#). Los aportes y preguntas están moderados simplemente para evitar spam o comentarios fuera de lugar.



Esta obra está; bajo una [licencia de Creative Commons](#).

Realizado por Emilio Astarita, [lee más acerca de este manual](#).

MANUAL  
<xhtml />

THIS PAGE IS  
VALID  
XHTML 1.0  
TRANSITIONAL  
W3C

Web manual-xhtml.blogspot.  
com

# Tablas

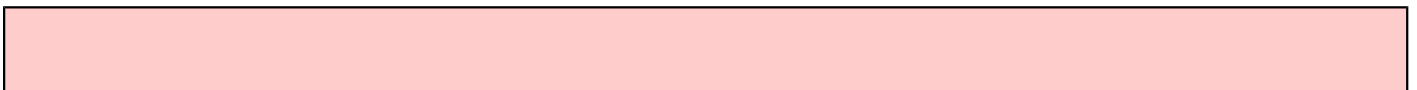
- [1. Introducción](#)
- [2. Ejemplo sencillo](#)
- [3. El atributo summary](#)
- [4. El pie de la tabla \(footer\)](#)
- [5. Elementos de una tabla gráficamente](#)

## 1. Introducción

Las tablas son una interesante herramienta para el **marcado de información tabular**, otros usos y abusos (como maquetación y diseño) se desaconsejan encarecidamente. De nuevo: **mantener por separado, el diseño del contenido**.

## 2. Ejemplo sencillo

Veamos un poco de código para empezar a hablar de tablas:



```

<table>
  <caption>Elementos XHTML</caption>
  <thead>
    <tr>
      <th>ELEMENTO</th>
      <th>TIPO</th>
      <th>SEMANTICA</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>em</td>
      <td>línea</td>
      <td>mayor énfasis</td>
    </tr>
    <tr>
      <td>strong</td>
      <td>línea</td>
      <td>mayor fuerza</td>
    </tr>
    <tr>
      <td>pre</td>
      <td>bloque</td>
      <td>texto preformateado</td>
    </tr>
  </tbody>
</table>

```

El resultado es el siguiente:

Elementos XHTML		
ELEMENTO	TIPO	SEMANTICA
em	línea	mayor énfasis
strong	línea	mayor fuerza
pre	bloque	texto preformateado

**NOTA:** cuando pruebes este ejemplo no verás bordes, aquí son mostrados mediante una propiedad **CSS** que explicaré más adelante.

El elemento **table** da comienzo a la tabla. Dentro tenemos a **caption** que encierra el título que llevará la tabla.

A partir de aquí definimos dos partes principales: **thead** dónde se ubica el encabezado, y **tbody** que encierra a las celdas del cuerpo.

**tr** es el encargado de contener filas, el encabezado tiene una sola fila, cuyas celdas de encabezado se contienen mediante tres etiquetas del elemento **th**.

Luego dentro del cuerpo (**tbody**) tenemos 3 filas más (**tr**) cuyas celdas se contienen mediante el uso del elemento **td**.

### 3. El atributo summary

El atributo *summary* se puede especificar dentro del elemento **table**, este nos permite agregar algo de semántica, al ponerle como valor un resumen del contenido de la tabla para que pueda ser reconocido por robots (ej. buscadores) u otros agentes de usuario.

Ejemplo de su uso:

```
<table summary="Elementos del lenguaje
XHTML">...
```

Los navegadores no suelen representar *summary*.

### 4. El pie de la tabla (footer)

Vimos en el ejemplo anterior como dividir la tabla en dos secciones, mediante los elementos **thead** y **tbody**.

Ahora incluyamos una tercera sección con el elemento **tfoot**, el cual determinará la sección del pie (al final de la tabla).

Utilizamos un código parecido al anterior, pero con la sección foot y otros datos.

```
<table summary="Comparativa de exploits entre IE 6 y Firefox">
  <caption>Comparativa: Exploits IE vs Firefox</caption>
  <thead>
    <tr>
      <th>Año</th>
      <th>Firefox</th>
      <th>Internet Explorer</th>
    </tr>
  </thead>
  <tbody>
    <tr>
```

```

    <td>2004</td>
    <td>15</td>
    <td>27</td>
</tr>
<tr>
    <td>2005</td>
    <td>33</td>
    <td>45</td>
</tr>
<tr>
    <td>2006</td>
    <td>37</td>
    <td>49</td>
</tr>
</tbody>
<tfoot>
    <tr>
        <th>Total</th>
        <td>85</td>
        <td>121</td>
    </tr>
</tfoot>
</table>

```

El resultado es el siguiente:

Comparativa: Exploits IE vs Firefox		
<b>Año</b>	<b>Firefox</b>	<b>Internet Explorer</b>
<b>Total</b>	85	121
2004	15	27
2005	33	45
2006	37	49

**NOTA:** Estos datos son inventados y las sumas no fueron verificadas :P.

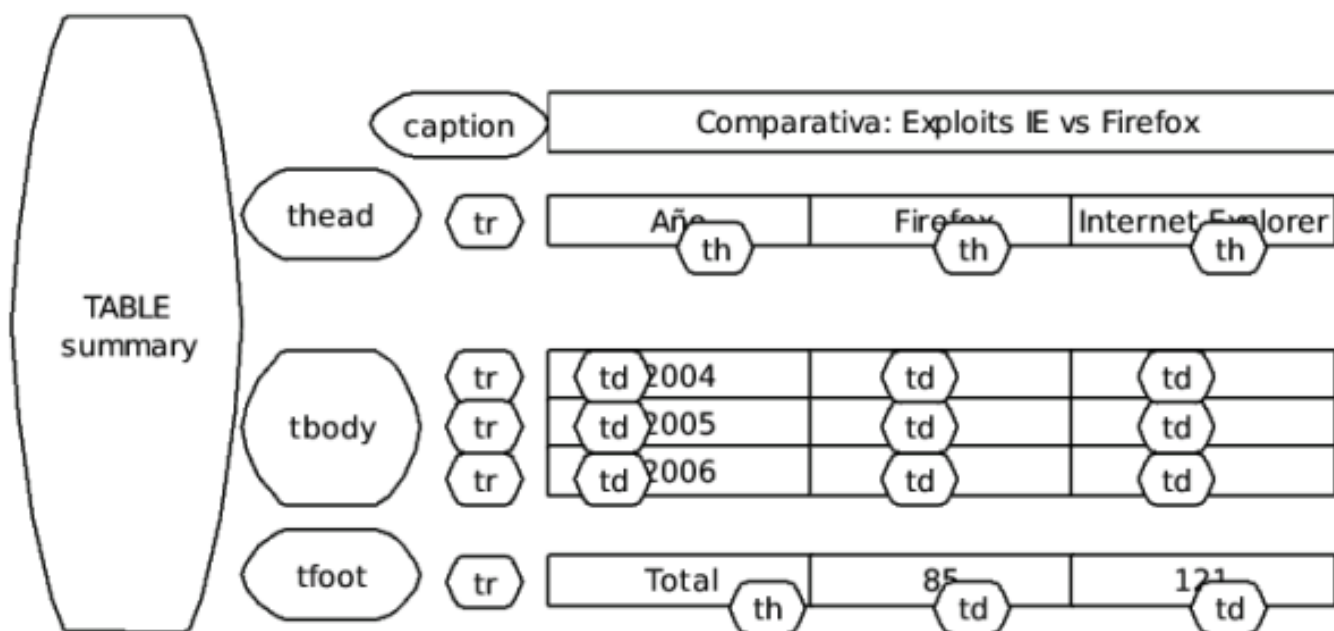
En estos ejemplos podríamos haber omitido marcar secciones (encabezado, cuerpo, pie) y la tabla tendría el mismo aspecto, sin embargo marcar estas secciones no sólo la dotarán de un valor semántico sino que también nos facilitará mucho la aplicación de estilos a estos grupos de celdas.

Notar también que en la sección **tfoot** la celda *Total* fue marcada con **th** ya que no es un dato estadístico.

## 5. Elementos de una tabla gráficamente



Quizá los maree un poco seguir la estructura de la tabla en el código, para ayudar aquí tienen un gráfico que muestra la tabla anterior visualmente y los elementos utilizados para construirla



**Nota:** gráfico creado con [Kivio](#), software gratuito para el entorno [KDE](#).

[Página principal](#)

**0 aporte(s) de lectores:**

[Publicar un comentario en la entrada](#)

**ATENCIÓN:** Por favor intenta aportar o preguntar sobre el tema que trata la página. Otro tipo de consultas o críticas puedes dejarlas en el [libro de visitas](#). Los aportes y preguntas están moderados simplemente para evitar spam o comentarios fuera de lugar.



Esta obra está; bajo una [licencia de Creative Commons](#).

Realizado por Emilio Astarita, [lee más acerca de este manual](#).



Web [manual-xhtml.blogspot.com](http://manual-xhtml.blogspot.com)

## Formularios

- [1. Introducción](#)
- [2. Dos métodos post y get](#)
  - [2.1. Método get](#)
  - [2.2. Método post](#)
- [3. Empezando con form](#)
- [4. Elemento input](#)
- [5. Campos de texto](#)
- [6. Botones de radio](#)
- [7. Botón de comprobación checkbox](#)
- [8. Botones de envío y de reset](#)
- [9. Cajas de texto](#)
- [10. Seleccionar con select](#)
- [11. Elemento label](#)
- [12. Agrupando con fieldset](#)

### 1. Introducción

Los formularios nos dan la posibilidad de recibir información de parte del usuario. Su utilización nos permitirá construir formularios de entrada de datos que podríamos recibir en nuestro e-mail o bien almacenar en una base de datos dentro de nuestro servidor, para luego por ejemplo, generar una página o una [cookie](#) de manera dinámica.

### 2. Dos métodos post y get

Hay dos métodos de envío de información mediante formularios que trataremos aquí: **post** y **get**.

## 2.1. Método get

El método **get** envía información al servidor como parte de la **URL**. El servidor analizará la **URL** pedida por el agente de usuario y separará la información relevante para su posterior procesamiento.

```
http://www.example.com/busqueda?palabra=xhtml
```

El signo **?** es el delimitador que nos permite separar la consulta de la **URL**. **busqueda** podría ser un programa CGI que recibirá la información posterior a **?** e interpretará tal cadena para crear una variable **palabra = xhtml**. En muchos casos sencillos y para enviar cadenas de pocos caracteres este método funciona bien, sin embargo para otros casos es mejor utilizar otro método de envío.

## 2.2. Método post

Supongamos que quisiéramos enviar una contraseña para loguearnos dentro de un servidor. No sería conveniente que la contraseña aparezca en la **URL** ya que otras personas podrían ver esta información.

Generalmente en este caso correspondería utilizar el método de envío **post**. Este método envía la información como parte de la petición del agente de usuario al servidor web (esta petición también incluye la **URL**, sin embargo nuestra información no formará parte de ella).

## 3. Empezando con form

El elemento principal y obligatorio de todo formulario es **form** y quizá sus dos atributos más importantes son **method** y **action**. El primero especifica el método de envío (**get** o bien **post**), y en el segundo pondremos la **URL** del recurso que contendrá la secuencia de comandos encargada de hacer algo con esa información.

Sin embargo un formulario sin otros elementos no nos sirve de mucho. Hagamos entonces un recorrido por los elementos más comunes:

## 4. Elemento input

Ahora presentaremos al elemento **input**, que al especificarle su atributo **type** nos permite construir diferentes tipos de controles.

## 5. Campos de texto

Código para construir dos campos de texto:

```
<form>
  <p>Nombre: <input type="text" name="nombre" /></p>
  <p>Apellido: <input type="text" name="apellido" /></p>
</form>
```

El valor **text** en el atributo **type** indica que será un campo de texto

Ahora si quisiéramos enviar esta información el agente de usuario armaría cadenas de texto, con pares del tipo **nombre=VALOR1** y **apellido=VALOR2** dónde **VALOR1** y **VALOR2** es la información escrita en las casillas de texto de Nombre y Apellido respectivamente, dichos valores se asocian al valor escrito en el atributo **name** del **input** de tipo **text**.

El resultado es el siguiente:

Nombre:

Apellido:

Ilustremos ahora algunas características con las que podemos dotar al **input**

```
<form>
  <p>
    E-mail:
    <input type="text" name="mail" readonly="readonly" value="example@example.
com" />
    <strong>readonly</strong>=<em>readonly</em> y <strong>value</
strong>=<em>example@example.com</em>.
  </p>
  <p>
    Contraseña:
    <input type="password" name="password" />
    <strong>type</strong>=<em>password</em>
  </p>
  <p>
    Iniciales:
    <input type="text" name="inicial" size="5" maxlength="2" />
    <strong>size</strong>=<em>3</em> y <strong>maxlength</strong>=<em>2</em>
  </p>
  <p>
    No foco:
    <input type="text" disabled="disabled" value="jo jo" />
    <strong>disabled</strong>=<em>disabled</em> y <strong>value</strong>=<em>jo
jo</em>
  </p>
</form>
```

Aquí el resultado:

E-mail: **readonly=readonly** y **value=example@example.com.**

Contraseña: **type=password**

Iniciales: **size=3** y **maxlength=2**

No foco: **disabled=disabled** y **value=jo jo**

Observen detenidamente como varía el comportamiento y la apariencia de los cuadros de texto según los diferentes atributos y valores especificados.

## 6. Botones de radio

Estos botones se suelen presentar en grupo y nos permite elegir una opción entre varias. Para esto se especifican valores iguales para el atributo **name**, mientras que para el **type** utilizamos el valor *radio*.

Piquemos código:

```
<form>
  <p>Sobre su edad:
  <input type="radio" name="edad" value="mayor" /> MAYOR
  <input type="radio" name="edad" value="menor" /> MENOR
  </p>
  <p>Su genero:
  <input type="radio" name="genero" value="hombre" /> HOMBRE
  <input type="radio" name="genero" value="mujer" /> MUJER
  </p>
</form>
```

El resultado:

Sobre su edad:    MAYOR    MENOR

Su genero:    HOMBRE    MUJER

Al especificar un valor distinto para **name**, se crea un nuevo grupo, mientras que dentro de un mismo grupo sólo podemos elegir una sola de las opciones.

## 7. Botón de comprobación checkbox

Ahora mostremos que pasa al poner en el atributo **type** el valor *checkbox*, que convierte a **input** en un *Botón de comprobación*.

Piquemos algo de código para dos **input** de tipo **checkbox**:

```
<form>
  <p>
    <input name="fuma" type="checkbox" checked="checked" value="si" /> Fumador
    <input name="deporte" type="checkbox" value="si" /> Deportista
  </p>
</form>
```

Observen el resultado:

Fumador    Deportista

Uno de los **input** lleva el atributo **checked** con el valor *checked*, de esta forma le indicamos que debe aparecer marcado por defecto ni bien se presente el formulario.

Si enviáramos esta información al servidor, se armarían pares `deporte=si` y `fuma=si` en el caso de que las dos estén marcadas.

## 8. Botones de envío y de reset

Ahora supongamos que queremos enviar esta información a algún servidor. Como dijimos existen dos métodos de envío. Según qué enviaremos y la tecnología del servidor sabremos por cuál optar. Para indicarle al agente de usuario el envío de información utilizaremos un botón de envío especificando el valor `submit` en el atributo **type**.

Probemos crear un cuadro de búsqueda para [Google](http://www.google.com/):

```
<form method="get" action="http://www.google.com/search">
  <p>
    <input name="q" type="text" />
    <input type="submit" value="Busca en Google" /></p>
</form>
```

La consulta a Google se formula de esta manera `http://www.google.com/search?q=cadena`. Para obtener `q=cadena` debemos especificar el atributo **name** con el valor `q`, luego `cadena` será lo que se introduzca en el cuadro de búsqueda. Por último el `input text` enviará toda la consulta por el método `get`.

Vean el resultado y testéenlo:

Otro botón muy utilizado, es el de tipo `reset` que se encarga de deshacer todas las modificaciones que haya hecho el usuario sobre el formulario, es decir regresa todos los elementos anidados bajo el elemento **form** a los valores originales especificados en el documento.

Prueben cambiar el texto dentro del cuadro y luego presionen **Reset**

El código anterior:

```
<form>
  <input type="text" value="Soy el original" />
  <input type="reset" />
</form>
```

## 9. Cajas de texto

Las cajas de texto se suelen utilizar para incluir una cantidad de texto mayor y de varias líneas, para la cual el **input** de `type text` no sería apropiado.

A picar código:

```
<form>
  <textarea rows="10" cols="50" >Quedo adentro del cuadro.</textarea>
</form>
```

Como resultado obtendremos:

Los atributos **rows** y **cols**, marcan la cantidad de líneas y el tamaño de ancho respectivamente, de la caja.

## 10. Seleccionar con select

Incluyendo el elemento **select** podremos armar una lista de selección múltiple, este elemento se combina con el elemento **option** para ir agregando las diferentes opciones.

Ejemplo de código:

```
<form>
  <p>
    Elija fruta:
    <select name="fruta">
      <option>Manzana</option>
      <option>Pera</option>
      <option>Kiwi</option>
      <option>Uva</option>
    </select>
  </p>
</form>
```

El resultado es el siguiente:

De nuevo **select** cuenta con el atributo **name** con el valor *fruta*.

Sí se enviara esta información a un servidor, el agente de usuario armará la cadena de caracteres con el par `fruta=Pera` que se enviará por **get** o **post**, suponiendo que se haya elegido la fruta **Pera**.



## 11. Elemento label

Este elemento nos permite asociar una cadena de texto a un control del formulario. Esto supone una cuota de semántica ya que se identifica al control con un texto y otra de accesibilidad debido a que los agentes de usuario permiten utilizar el control mediante el texto asociado.

Ejemplo de código:

```
<form>
  <p><label><input type="checkbox" /> Con label</label></p>
  <p><input type="checkbox" /> Sin label</p>
</form>
```

Observen el resultado:

Con label

Sin label

En principio parecen idénticas, pero prueben hacer clic sobre el texto *Con label*, al hacerlo la casilla de la izquierda al texto se seleccionará. En cambio al hacer clic sobre *Sin label* no ocurre nada.

Siempre que sea posible se recomienda utilizar el elemento **label**

## 12. Agrupando con fieldset

El agrupamiento es un recurso interesante para estructurar mejor nuestros formularios. El elemento **fieldset** nos da la posibilidad de agrupar diferentes secciones bajo un marco que puede llevar una descripción al anidarlo otro elemento llamado **legend**.

Escribamos algunos controles y agrupemos bajo dos grupos:

Código:

```
<form>
  <fieldset>
    <legend>Información personal</legend>
    <label>Nombre: <input type="text" /></label>
    <label>Apellido: <input type="text" /></label>
    <label><input type="checkbox" /> Soltero</label>
  </fieldset>
  <fieldset>
    <legend>Información académica</legend>
    <label>Universidad: <input type="text" /></label>
    <label><input type="checkbox" /> Título obtenido</label>
  </fieldset>
</form>
```

Resultado de lo anterior:

Información personalNombre:

Apellido:

Solter

Información académicaUniversidad:

Título obtenido

[Página principal](#)

**1 aporte(s) de lectores:**

[jam11138](#) aporta...

Le he hecho un ojo por encima y se ve muy bueno... Solo creo que en el Tema 10 (listas) pudo mencionarse la etiqueta "optgroup" que igual es XHTML Stricto.

[10:51 AM](#) \_

[Publicar un comentario en la entrada](#)

**ATENCIÓN:** Por favor intenta aportar o preguntar sobre el tema que trata la página. Otro tipo de consultas o criticas puedes dejarlas en el [libro de visitas](#). Los aportes y preguntas están moderados simplemente para evitar spam o comentarios fuera de lugar.



Esta obra está; bajo una [licencia de Creative Commons](#).

Realizado por Emilio Astarita, [lee más acerca de este manual](#).



Web [manual-xhtml.blogspot.com](http://manual-xhtml.blogspot.com)

## Elemento SPAN

### span: elemento de línea

Hemos visto elementos como **strong** o **em** utilizados para marcar fuerza o enfatizado respectivamente. Existe un elemento genérico de agrupación en línea llamado **span**. Con él podemos agrupar un conjunto de palabras por algún criterio especial que se nos ocurra. Y luego podremos aplicar a todas ellas estilos mediante clases.

La idea es sencilla veamos un ejemplo:

```
<span class="tipos_fruta">frutas de hueso</span> <span
```

```
class="ejemplo_fruta">albaricoque</span> <span
class="ejemplo_fruta">melocotón</span> <span class="tipos_fruta">fruta de
pepita</span> <span class="ejemplo_fruta">pera</span> <span
class="ejemplo_fruta">manzana</span>
```

En este sencillo y vegetal ejemplo, utilizamos span para agrupar frases según dos criterios semánticos humanos: *tipos de frutas* y *ejemplos de frutas*. A cada una de éstos elementos lo hacemos pertenecer a una clase, la cual nos facilitará la aplicación de estilos a todos los elementos que pertenezcan a esa clase.

Los agentes de usuario no suelen representar de manera diferente al texto en el renderizado de la etiqueta **span**.

Lo anterior renderizado por tu navegador se ve así:

```
frutas de hueso albaricoque melocotón fruta de pepita pera manzana
```

Por esto uno se podría preguntar cual es su utilidad, más allá de su significación semántica. Aquí entra en juego la utilización de CSS para dotar de presentación nuestros contenidos.

Los estilos podemos definirlos en un archivo de texto plano independiente de nuestro documento XHTML o bien incluirlo en el mismo marcado con el elemento **style** dentro de la sección **head**.

```
<head>

...

<style type="text/css">

    .tipos_fruta {
    color:#FF0000;
    text-decoration:underline;
    }

    .ejemplo_fruta {
    color:blue;
```

```
font-size:80%;  
}  
  
</style>  
  
...  
  
</head>
```

Tu agente de usuario interpreta y renderiza de la siguiente manera los estilos anteriores:

frutas de hueso albaricoque melocotón fruta de pepita pera manzana

Hemos logrado no sólo una agrupación semántica sino una presentación personalizada para todos los elementos agrupándolos por clases, y sin embargo hemos mantenido una separación entre el contenido y la presentación, que es uno de los objetivos fundamentales del XHTML.

[Página principal](#)

**0 aporte(s) de lectores:**

[Publicar un comentario en la entrada](#)

**ATENCIÓN:** Por favor intenta aportar o preguntar sobre el tema que trata la página.

Otro tipo de consultas o criticas puedes dejarlas en el [libro de visitas](#). Los aportes y preguntas están moderados simplemente para evitar spam o comentarios fuera de lugar.



Esta obra está; bajo una [licencia de Creative Commons](#).

Realizado por Emilio Astarita, [lee más acerca de este manual](#).

## Emacs: xHTML mode sgml-mode

- ```

emancipator@kali:~$ sudo -l
funciones y comandos
descriptores
root@kali:/#

Si estamos posicionados con el cursor dentro de alguna etiqueta
inserta de manera interactiva un atributo. La inserción
interactiva, respecta autocompletado de los atributos validos para
tal etiqueta.
Tip

root@skip-tag:backward

Salto hasta el comienzo de la anterior etiqueta abierta, si
está precedida. Supera argumentos anteriores para saltar tantas
veces como se requiera.
Tip

root@delete-tag

Elimina las etiquetas de apertura y de cierre y las elimina, si
estas tienen contenido entre ellas, este no se elimina.
Tip

root@skip-tag:forward

Mueve la posición del cursor hacia adelante hasta posicionarse
por delante de la etiqueta que cierra al primer tag abierto
anterior a la posición original del cursor.
Tip

root@emancipator@kali:~$ find /home/emancipator/.ssh -name id_rsa.pub | xargs cat > /dev/null
total used in directory 50 available 17608012
dwxv-xr-x 2 chmodo chmodo 4096 2007-02-25 21:15
dwxv-xr-x 8 chmodo chmodo 4096 2007-02-25 21:15
-rw-r--r-- 1 chmodo chmodo 17374 2007-02-25 21:12 [emancipator@kali:~$]

root@emancipator@kali:~$
```

Es un modo basado en el modo SGML, para editar documentos HTML.

Permite insertar los esqueletos más comunes utilizados al editar xHTML



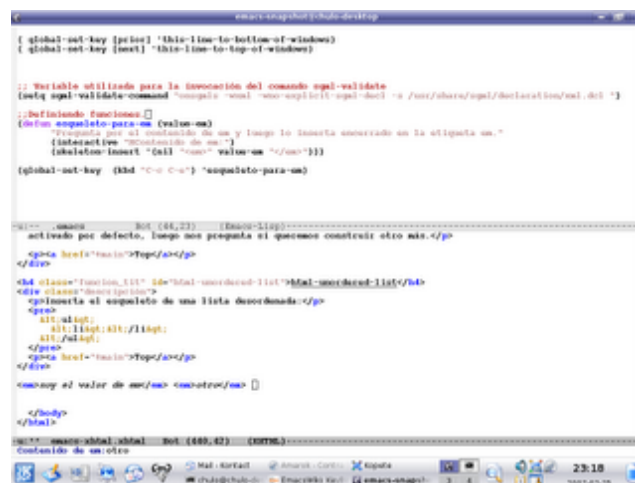
Para ver una descripción completa del modo: C-M

Esta no pretende ser una guía sobre emacs, si es la primera vez que estás por utilizarlo te recomiendo que lo abras y con C-h t (esto es presionar Control + h, soltar control y luego la letra t) leas el tutorial para principiantes, que si tienes suerte lo encontraras en tu idioma.

## Personalización

Emacs es un editor extensible por lo que facilmente podremos extender el modo, con otras funciones que no vienen construidas por defecto en este modo.

## Esqueletos



Supongamos que queremos construir un esqueleto para generar automáticamente las etiquetas de apertura y de cierre del elemento **em**. Para esto nos vamos a valer de una función del emacs-lisp conocida como `skeleton-insert`

Puedes probar el siguiente código en tu `.emacs` :

```
(defun esqueleto-para-em (value-em)
  "Pregunta por el contenido de em y luego lo inserta encerrado en la"
```



```
etiqueta em."
  (interactive "MContenido de em:")
  (skeleton-insert '(nil "<em>" value-em "</em>")))
```

Ahora hemos definido una función `esqueleto-para-em`, que al ejecutarse nos preguntará en el minibuffer por el string que debe encerrar las etiquetas de apertura y cierre de `em`, para luego insertar todo junto.

Opcionalmente podrías optar definirle una secuencia de teclas para que sea más fácil invocarlo, por ejemplo yo utilizo `C-c C-e`, debajo tu `.emacs` agregas:

```
(global-set-key (kbd "C-c C-e") 'esqueleto-para-em)
```

## Generador de tablas

Un ejemplo muy útil y apenas un poco más avanzado es armarnos un generador de tablas. La construcción de tablas HTML es algo tediosa, por lo que no viene nada mal implementar una función en nuestra librería para librarnos de este trabajo.

Nuestra función recibirá nos preguntará por dos parametros filas y columnas, a lo que nosotros responderemos en el minibuffer con un número entero positivo.

Veamos como definirla:

```
(defun construye-tabla (filas columnas)
  "Dadas las cantidad de filas y columnas construye una tabla"
  (interactive "nFilas: \nnColumnas: ")
  (insert "<table border=\"1\">\n<caption>Tabla</caption>\n")
  (loop for i from 0 below filas by 1 do
    (insert "<tr>\n")
    (loop for j from 0 below columnas by 1 do
      (insert "<td></td>\n"))
    (insert "</tr>\n"))
  (insert "</table>"))
```

La función `construye-tabla` recibe dos argumentos, luego inserta la apertura de la etiqueta **table** un **caption** y entra en el primer *loop* que es el encargado de generar las filas. Luego comienza un *loop* interno que es el que construye las columnas, tantas como el argumento recibido. A la salida de los loops simplemente queda por cerrar la tabla.

Con muy poco trabajo y en sólo 10 líneas hemos definido un generador de tablas básico pero muy útil.

Sólo por deporte expondré una salida:

```
Filas: 3
Columnas: 2
```

Tabla

|     |     |
|-----|-----|
| 1-1 | 1-2 |
| 2-1 | 2-2 |
| 3-1 | 3-2 |

Emacs mola :)

## Validación



Emacs nos permite validar nuestros documentos invocando un validador externo en un buffer asociado a un shell. Luego en ese buffer obtendremos la salida del validador indicando, si los hay, los errores que cometimos, con la combinación de teclas `C-x o` podemos cambiar a tal buffer y al dar `RET` sobre un error específico nos llevará automáticamente a la posición dónde el error se ha encontrado.

Especialmente útil para cumplir las recomendaciones del [World Wide Web consortium](http://www.w3.org/).

Para correr el validador basta con:

`M-x sgml-validate`

**NOTA:** Es posible que nuestra distribución no traiga por defecto el validador externo por línea de comando, en Ubuntu el paquete que lo contiene se llama `opensp`. Para instalarlo vale con:

```
aptitude install opensp
```

También debemos definir el comando específico que se invocará para validar, esto lo podemos hacer en nuestro `.emacs`:

```
;; Variable utilizada para la invocación del comando sgml-validate
(setq sgml-validate-command
  "onsgmls -wxml -wno-explicit-sgml-decl -s /usr/share/sgml/declaration/
xml.dcl ")
```



## Key bindings

| Secuencia de teclas | Función                                |
|---------------------|----------------------------------------|
| C-c C-a             | <a href="#">sgml-attributes</a>        |
| C-c C-b             | <a href="#">sgml-skip-tag-backward</a> |
| C-c C-d             | <a href="#">sgml-delete-tag</a>        |
| C-c C-f             | <a href="#">sgml-skip-tag-forward</a>  |
| C-c TAB             | <a href="#">sgml-tags-invisible</a>    |
| C-c C-j             | <a href="#">html-line</a>              |
| C-c RET             | <a href="#">html-paragraph</a>         |
| C-c C-n             | <a href="#">sgml-name-char</a>         |
| C-c C-s             | <a href="#">html-autoview-mode</a>     |
| C-c C-t             | <a href="#">sgml-tag</a>               |
| C-c C-v             | <a href="#">browse-url-of-buffer</a>   |
| C-c /               | <a href="#">sgml-close-tag</a>         |
| C-c 1               | <a href="#">sgml-headline-1</a>        |
| C-c 2               | <a href="#">sgml-headline-2</a>        |
| C-c 3               | <a href="#">sgml-headline-3</a>        |
| C-c 4               | <a href="#">sgml-headline-4</a>        |
| C-c 5               | <a href="#">sgml-headline-5</a>        |
| C-c 6               | <a href="#">sgml-headline-6</a>        |
| C-c 8               | <a href="#">sgml-name-8bit-mode</a>    |
| C-c ?               | <a href="#">sgml-tag-help</a>          |
| C-c DEL             | <a href="#">sgml-delete-tag</a>        |
| C-c left            | <a href="#">sgml-skip-tag-backward</a> |
| C-c right           | <a href="#">sgml-skip-tag-forward</a>  |
| C-c C-c -           | <a href="#">html-horizontal-rule</a>   |
| C-c C-c c           | <a href="#">html-checkboxes</a>        |
| C-c C-c h           | <a href="#">html-href-anchor</a>       |

|           |                                     |
|-----------|-------------------------------------|
| C-c C-c i | <a href="#">html-image</a>          |
| C-c C-c l | <a href="#">html-list-item</a>      |
| C-c C-c n | <a href="#">html-name-anchor</a>    |
| C-c C-c o | <a href="#">html-ordered-list</a>   |
| C-c C-c r | <a href="#">html-radio-buttons</a>  |
| C-c C-c u | <a href="#">html-unordered-list</a> |

## Funciones y comandos

### Descripciones

#### **sgml-attributes**

Si estamos posicionados con el cursor adentro de algún tag inserta de manera interactiva un atributo. La inserción interactiva, soporta autocompletado de los atributos validos para tal etiqueta.

[Top](#)

#### **sgml-skip-tag-backward**

Saltea hasta el comienzo de la anterior etiqueta abierta, si está presente. Soporta argumentos numéricos para saltar tantas veces como se requiera.

[Top](#)

#### **sgml-delete-tag**

Busca las etiquetas de apertura y de cierre y las elimina, si estas tienen contenido entre ellas, este no se elimina.

[Top](#)

#### **sgml-skip-tag-forward**

Mueve la posición del cursor hacia adelante hasta posicionarse por delante de la etiqueta que cierra el primer tag abierto anterior a la posición original del cursor.

[Top](#)

#### **sgml-tags-invisible**

Oculto los tags, y deja sólo el contenido formateandolo y estructurandolo según las etiquetas, muy útil para observarlo más humanamente. Invicando el comando nuevamente volvemos a hacer visibles las etiquetas.

[Top](#)

#### **html-line**

Inserta un salto de línea forzado con el elemento **br**

[Top](#)

## html-paragraph

Inserta un párrafo utilizando el elemento **p**.

[Top](#)

## sgml-name-char

Ciertos caracteres como `<` o `>`, no pueden ser insertados directamente debido a que pueden ser interpretados como la apertura de una etiqueta, este comando nos permite introducir tales caracteres directamente y escribir su entidad correspondiente.

[Top](#)

## html-autoview-mode

Cuando tenemos activada esta función, cada vez que salvemos se nos abrirá directamente el navegador por defecto para mostrarnos los cambios salvados.

[Top](#)

## sgml-tag

Permite insertar un tag, interactivamente, opcionalmente nos pregunta de manera interactiva por los argumentos que debe llegar, con presionar RET, obviamos los argumentos.

[Top](#)

## browse-url-of-buffer

Envia la url del buffer al navegador por defecto para visualizar nuestro documento.

[Top](#)

## sgml-close-tag

Busca el tag de apertura más inmediatamente anterior al cursor y escribe la etiqueta de cierre correspondiente.

[Top](#)

## sgml-headline-1

Inserta la etiqueta de apertura y cierre del elemento **h1**

[Top](#)

## sgml-headline-2

Inserta la etiqueta de apertura y cierre del elemento **h2**

[Top](#)

### **sgml-headline-3**

Inserta la etiqueta de apertura y cierre del elemento **h3**

[Top](#)

### **sgml-headline-4**

Inserta la etiqueta de apertura y cierre del elemento **h4**

[Top](#)

### **sgml-headline-5**

Inserta la etiqueta de apertura y cierre del elemento **h5**

[Top](#)

### **sgml-headline-6**

Inserta la etiqueta de apertura y cierre del elemento **h6**

[Top](#)

### **sgml-name-8bit-mode**

Cuando no trabajamos con caracteres ASCII tales como letras acentuadas, u otros, podemos activar este modo que reemplazará acentos y Ñs por la entidad equivalente. Sólo funciona para inputs Latin-1.

[Top](#)

### **sgml-tag-help**

Posicionados sobre un tag, nos brinda en el mini-buffer una breve descripción de tal tag.

[Top](#)

### **html-horizontal-rule**

[Top](#)

### **html-checkboxes**

Nos pregunta nombre, valor, y si debe estar tildado por defecto, luego inserta el esqueleto.

[Top](#)

### **html-href-anchor**

Nos pregunta por una URL y luego inserta el esqueleto de un enlace para que podamos poner la palabra mediante la cual queremos enlazar.

[Top](#)

## html-image

Escribe el esqueleto para la inserción de una imagen, nos pregunta por el valor del atributo *src* y luego deja posicionado el cursor para completar el valor del atributo *alt*

[Top](#)

## html-list-item

Simplemente agrega un item (elemento **li**) a la lista.

[Top](#)

## html-name-anchor

Invoca a un comando interactivo para la construcción de un anchor, sólo pide el valor de un nombre y luego construye un anchor, con los atributos id, y name completados con el valor preguntado.

[Top](#)

## html-ordered-list

Inserta el esqueleto de una lista ordenada:

```
<ol>
  <li></li>
</ol>
```

[Top](#)

## html-radio-buttons

Inserta el esqueleto para radio-buttons de formularios, de manera interactiva. Nos pregunta por el nombre el valor y si debe estar activado por defecto, luego nos pregunta si queremos construir otro más.

[Top](#)

## html-unordered-list

Inserta el esqueleto de una lista desordenada:

```
<ul>
  <li></li>
</ul>
```

[Top](#)

Etiquetas: [emacs](#)

[Página principal](#)

**0 aporte(s) de lectores:**

[Publicar un comentario en la entrada](#)

**ATENCIÓN:** Por favor intenta aportar o preguntar sobre el tema que trata la página. Otro tipo de consultas o críticas puedes dejarlas en el [libro de visitas](#). Los aportes y preguntas están moderados simplemente para evitar spam o comentarios fuera de lugar.



Esta obra está; bajo una [licencia de Creative Commons](#).

Realizado por Emilio Astarita, [lee más acerca de este manual](#).





Sign in to use Blogger

But first, where do you blog?

**New Blogger**  
Using your  
Google account

**Old Blogger**  
Using your  
Blogger account



## Check out the new Blogger!

We're out of beta and ready to go.

There's new stuff like labels, drag-and-drop template editing, and privacy controls. [Take a tour](#)

We're already moving some lucky people, but you can cut the line and switch now. New accounts get the new Blogger, of course.

[Switch Now](#)

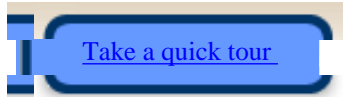
Need more info? See our [help with switching](#) guide.

## What's a blog?

- [Publish thoughts](#)
- [Get feedback](#)
- [Post photos](#)
- [Go mobile](#)



A **blog** is your easy-to-use web site, where you can quickly post thoughts, interact with people, and more. All for **FREE**.



## Create a blog in 3 easy steps:

1. Create an account
2. Name your blog
3. Choose a template

[Create your blog now](#)

## Explore blogs

**Blogs  
Updated at  
3:11 AM**

- [.....](#)
- [maschenfee](#)
- [reporting tool](#)
- [portosun günlü•ü](#)
- [Health](#)

### Blogs of Note

- [The Vixtrot Kid](#)
- [Thingology](#)
- [soul sides](#)
- [ikea hacker](#)
- [Swing, Jazz and Blues](#)
- [Soyapi Mumba's Blog](#)
- [Cricket Zone with Krish Srikkanth](#)
- [PuppetVision](#)
- [unimaginable inscape](#)
- [Obviously](#)

### Random Blogs

[Next Blog >>](#)

Google

Search blogs from across the web with [Google Blog Search](#).



---

[Home](#) | [About](#) | [Buzz](#) | [Help](#) |  
[Language](#) | [Developers](#) | [Gear](#) |  
[Privacy](#) | Copyright © 1999 - 2007



## Creative Commons

Esta página está disponible en los siguientes idiomas:

Afrikaans ••••• Català Dansk Deutsch English English (CA) English (GB) English (US) Esperanto Castellano Castellano (AR) Español (CL) Castellano (MX) Euskara Suomeksi français français (CA) Galego •••• hrvatski Magyar Italiano •• •• Melayu Nederlands Sesotho sa Leboa polski Português Português (PT) svenska slovenski jezik ••• •• (••) IsiZulu



## Atribución-CompartirDerivadasIgual 2.5 Argentina



### Usted es libre de:



- copiar, distribuir, exhibir, y ejecutar la obra



- hacer obras derivadas

### Bajo las siguientes condiciones:

- **Atribución.** Usted debe atribuir la obra en la forma especificada por el autor o el licenciente.

- **Compartir Obras Derivadas Igual.** Si usted altera, transforma, o crea sobre esta obra, sólo podrá distribuir la obra derivada resultante bajo una licencia idéntica a ésta.

- 
- Ante cualquier reutilización o distribución, usted debe dejar claro a los otros los términos de la licencia de esta obra.
- Cualquiera de estas condiciones puede dispensarse si usted obtiene permiso del titular de los derechos de autor.
- Nada en esta licencia menoscaba o restringe los derechos morales del autor.

**Sus usos legítimos u otros derechos no son afectados de ninguna manera por lo dispuesto precedentemente.**

[Limitación de Responsabilidad](#)



Este es un resumen legible-por-humanos del [Código Legal \(la licencia completa\)](#).

[Aprenda como distribuir su obra usando esta licencia](#)