

# Lenguajes de programación HTML y CSS

## Parte 1: Conceptos básicos de HTML 5

### ¿Cómo crear páginas web?

¿Nunca has oído hablar de HTML, o sólo vagamente del mismo?

No te asustes, las explicaciones vienen desde el primer capítulo ... la práctica sigue justo después.

Empezamos por la presentación de cómo funcionan los sitios web, entonces vamos a descargar todos los programas (gratis) para trabajar bien.

Al final de esta sección, sabrás insertar texto, enlaces e imágenes.

¿Cómo puedo crear sitios web?

El primer capítulo de esta documentación para principiantes, te enseñará a crear un sitio.

Vamos a pasar algún tiempo juntos, todo dependerá de lo rápido que aprendas. Si lees esta documentación regularmente y a buen ritmo, terminarás en una o dos semanas. Pero si necesitas un poco más de tiempo, no te preocupes: la principal es que vayas a tu propio ritmo, con preferencia en pasar un buen rato.

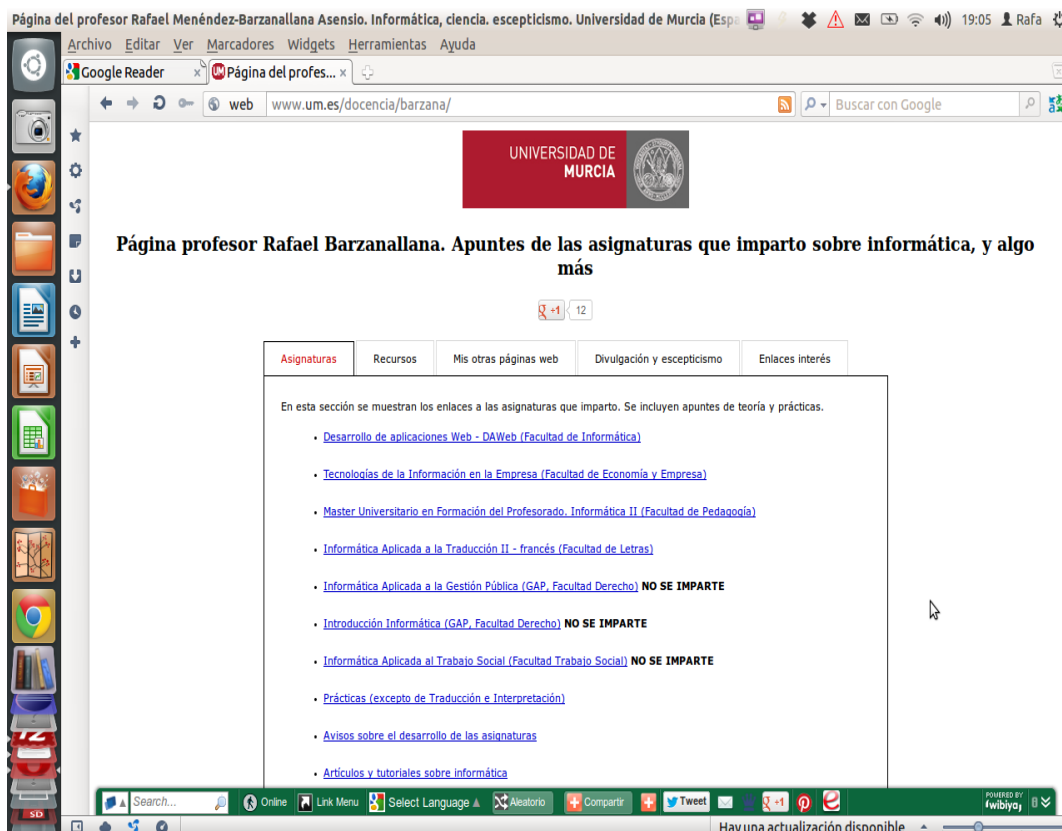
Te sugiero que comiences con las preguntas más sencillas, pero también las más importantes:

### Funcionamiento de sitios web

¿Cómo funcionan los sitios web? No tengas miedo de hacer preguntas, incluso si piensas que son absurdas. Es muy importante hablar un poco antes de entrar de lleno en el diseño del sitio. Estoy seguro de que visitas sitios web a diario. Para ello, se ejecuta un programa llamado navegador web, (Chrome, Firefox, Opera, ...) haciendo clic en uno de los iconos que se muestran en la siguiente figura y que es habitual que estén disponibles en la mayoría de ordenadores.



Con el navegador, se puede visitar cualquier sitio web. Por ejemplo, un explorador muestra sitios como Wikipedia, un periódico de la ciudad en que resido o mi página en la Universidad de Murcia.



Hoy en día, cualquier persona sabe acceder a la Web ... pero ¿quién sabe realmente cómo funciona internet? ¿Cómo crear páginas web como esta?

Has oído hablar de HTML y CSS, ¿tiene esto que ver con el funcionamiento de sitios web? ¡Por supuesto! Estos son los lenguajes de programación que permiten crear sitios web. Todos los sitios web se basan en estos lenguajes, que son la base de los sitios de internet. HTML fue inventado por Tim Berners-Lee en 1991, quien todavía sigue de cerca la evolución de la Web. Creó el *World Wide Web Consortium* (W3C), que define las nuevas versiones de internet relacionadas con los lenguajes de programación. También ha creado más recientemente, la

**Fundación World Wide Web**, que hace el seguimiento y análisis de la evolución de la Web.

Mucha gente confunde (erróneamente) internet y la Web. Debes saber que la Web es parte de internet. Es un gran paquete que incluye, entre otros: la Web, correos electrónicos, mensajería instantánea, etc. Tim Berners-Lee no es el inventor de internet, es "solo" inventor de la Web.

HTML y CSS son la base del funcionamiento de casi todos los sitios web. Cuando visitas un sitio mediante un navegador, debes saber que, detrás de las escenas, las "ruedas" se activan para permitir que el sitio web se muestre. El ordenador se basa en lo que se indica en HTML y CSS para saber lo que ha de mostrar, la figura siguiente indica parte del código de la página mostrada previamente.

```

Código fuente: Página del profesor Rafael Menéndez-Barzanallana Asensio. Informática, ciencia, escepticismo, Universidad...
Archivo Editar Ver Marcadores Widgets Herramientas Ayuda
Google Reader
Página del profes... Código fuente: P...
Guardar Aplicar cambios
<!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">
<head>
<title>Página del profesor Rafael Menéndez-Barzanallana Asensio. Informática, ciencia, escepticismo, Universidad de Murcia (España)</title>
<meta name="Author" content="Rafael Menéndez-Barzanallana Asensio" />
<base href="http://www.un.es/docencia/barzana/" />
<script type="text/javascript" src="http://www.un.es/docencia/barzana/js/jquery-1.7.2.js"></script>
<script type="text/javascript" src="http://clivick.com/assets/js/jquery.tools.js"></script>
<link rel="stylesheet" media="all" type="text/css" href="css/tabbed_pages.css" />
<script src="js/tabbed_pages.js" type="text/javascript"></script>
<meta name="verify-v1" content="2lBkq0HJfYpW000PTj0u0x000M+Hv0Z00" />
<title>Tutorial y apuntes de Informática. Escepticismo, pensamiento crítico, magia, física, divulgación. Rafael Barzanallana. Departamento Informática y Sistemas
IBSI. Universidad de Murcia (España)</title>
<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-15" />
<meta http-equiv="Cache-Control" content="no-cache">
<meta http-equiv="Expires" content="0">
<meta name="Robots" content="index,follow">
<meta name="Description" content="Apuntes de las asignaturas:
Máster Universitario en Formación del Profesorado de Educación Secundaria Obligatoria y Bachillerato, Formación Profesional, Enseñanzas de Idiomas y Enseñanzas
Artísticas
Desarrollo aplicaciones web
Informática Aplicada a la Traducción II (francés)
Informática Aplicada a la Gestión Pública
Introducción a la Informática
Informática Aplicada al Trabajo Social
Sistemas Informáticos para la Gestión Económica y Financiera.
Escepticismo
Pensamiento crítico
Física
Divulgación
Religión">
<meta http-equiv="content-language" content="ES-ES">
<meta name="Rating" content="General">
<meta name="author" content="Universidad Murcia - Rafael Barzanallana">
<meta name="keywords" content="apuntes, tutorial, informática, desarrollo aplicaciones web, máster formación profesorado, trabajo social, gestión y administración
pública, escepticismo, escepticos, pensamiento crítico, física, divulgación, religión, magia">
<meta name="revisit-after" content="2">
<meta name="generator" content="Basta + 3.5">
<meta http-equiv="distribution" content="Global">
<link rel="alternate" type="application/rss+xml" title="RSS" href="http://www.un.es/docencia/barzana/rdt.xml" />

```

HTML y CSS son dos "lenguajes" que necesitas saber para elaborar sitios web. Es el navegador web el que hará la traducción (intérprete) entre los lenguajes de programación y lo que se ve en la pantalla. Puedes estar preguntándote ¿por qué se precisa saber dos lenguajes para crear sitios web? Voy a responder ya.

## HTML y CSS: dos lenguajes para la creación de un sitio web

Para crear un sitio web, debes dar instrucciones al ordenador. No basta con escribir el texto que se ha incluido en el sitio (como lo haría en un procesador de textos), también debe indicar dónde colocar este texto, insertar imágenes, crear enlaces entre páginas, etc.

## Las funciones de HTML y CSS

Para explicar al ordenador lo que quieres hacer, tendrás que utilizar un lenguaje de programación que comprendas. Y es aquí donde las cosas se ponen difíciles, porque vas a tener que aprender dos.

¿Por qué se crearon dos? Uno habría bastado, ¿no? Manejar dos lenguajes será el doble de complejo y el doble de tiempo para aprender ... pero no es el caso. Lo aseguro, si hay dos lenguajes es, en cambio, para facilitar las cosas. Vamos a tratar con dos lenguajes que son complementarios, ya que tienen funciones diferentes:

**HTML** (*HyperText Markup Language*): apareció por primera vez en 1991 en el lanzamiento de la Web. Su función es la gestión y organización del contenido. Así que en HTML puedes escribir lo que deseas mostrar en la página: texto, enlaces, imágenes ...

Se podría decir: "Este es mi título, este es mi menú, aquí está el texto principal de la página, aquí hay una visualización de la imagen, etc. ".

**CSS** (*Cascading Style Sheets*, también conocidas como hojas de estilo): el papel de CSS es gestionar la apariencia de la página web (diseño, posicionamiento, colores, tamaño de texto ...). Este lenguaje ha complementado el código HTML desde 1996.

También puedes haber oído hablar de XHTML. Esta es una variante de HTML que es más rigurosa y que es un poco más complicada de manejar.

En pocas palabras, el HTML apareció por primera vez en 1991. A principios de 2000, la organización W3C lanzó el XHTML indicando que sería el futuro ..., pero no se difundió como se esperaba. En 2009 XHTML salió de la W3C y decidió regresar a HTML para hacerlo evolucionar.

Hay mucha confusión en torno a estos lenguajes, ya que son muy similares. No es uno mejor que el otro, hay dos maneras de hacer las cosas. Aquí vamos a trabajar con la última versión de HTML (HTML5) que ahora es el lenguaje del futuro, que todos estáis invitados a utilizar.

Se puede crear un sitio web únicamente en HTML, pero no va a quedar muy estético por la forma como aparecerá la información. Esta es la razón por la que CSS siempre lo completa.

Para hacerse una idea, la figura siguiente muestra cómo se ve la misma página sin CSS y con CSS.



HTML define el contenido, CCS permite organizar el contenido y definir la presentación: el color, la imagen de fondo, los márgenes, el tamaño del texto ... Como te puedes imaginar, CSS necesita una página HTML para funcionar. Es por eso que lo primero que se aprende son los conceptos básicos de HTML antes de lograr el cuidado de la decoración en CSS.

Las primeras páginas no serán las más agradables estéticamente, pero a quién le importa. Esto no durará mucho tiempo.

## Diferentes versiones de HTML y CSS

Con el tiempo, HTML y CSS han evolucionado. La primera versión de HTML (HTML 1.0) ni siquiera ofrecía la posibilidad de mostrar las imágenes.

Una muy breve historia de estos lenguajes para conocimiento general:

HTML 1: es la primera versión creada por Tim Berners-Lee en 1991.

HTML 2: la segunda versión de HTML apareció en 1994 y se terminó en 1996 con la aparición de HTML 3.0, esta es la versión que en realidad plantea las bases de las siguientes versiones de HTML. Las reglas y el funcionamiento de esta versión están dadas por el W3C (mientras que la primera versión fue creada por una persona).

HTML 3: apareció en 1996, esta nueva versión de HTML, añade muchas posibilidades al lenguaje como tablas, *applets*, *scripts*, posicionamiento de texto alrededor de imágenes, etc.

HTML 4: Esta es la versión más común de HTML (en concreto, es HTML 4.01). Apareció por primera vez en 1998 y propone el uso de marcos (que dividen una página web en varias partes), tablas más complejas, mejoras en las formas, etc. Más importante aún, esta versión permite por primera vez utilizar hojas de estilo del famoso CSS.

HTML 5: es la última versión. Aún no está muy extendida, llama mucho la atención porque trae muchas mejoras como la posibilidad de incluir fácilmente vídeos, mejorar el contenido, nuevas características para los formularios, etc. Esta es la versión que vamos a describir en esta documentación. EL estándar estará completado en el año 2014.

CSS 1: en 1996, apareció la primera versión de CSS. En ella se establecen las bases de este lenguaje que permiten mejoras de presentación en páginas web, tales como colores, márgenes, fuentes, etc.

CSS 2: apareció en 1999 y fue completado por CSS 2.1, esta nueva versión de CSS, añade numerosas opciones. Ahora se pueden utilizar técnicas de posicionamiento muy precisas que permiten ver los elementos en el lugar deseado en la página.

CSS 3: esta es la última versión, que agrega características muy esperadas, tales como bordes redondeados, degradados, sombras, etc.

De HTML5 y CSS3 aún no se han finalizado totalmente las versiones estandarizadas de W3C. Sin embargo, incluso si puede haber cambios menores en estos lenguajes, te recomiendo que comiences ahora con estas nuevas versiones. Sus contribuciones son muchas y vale la pena. Además, muchos sitios web profesionales se construyen hoy en día en las versiones recientes.

### **Editor de textos (programas)**

¿Qué software necesito para crear mi sitio web? ¿Tendré que romper mi hucha para comprar un software muy complejo que me llevará meses entender?

En realidad, hay muchos programas dedicados a la creación de sitios web. Pero, aseguro, que no tendrás que gastar un solo euro. ¿Por qué ir por un *software* de pago y complicado, si ya tienes en casa todo lo que necesitas?

Para iniciarse basta solo con ... el bloc de notas, o equivalente en otros sistemas operativos distintos de Windows. Sin embargo, actualmente hay software más potente y en realidad ya nadie usa el bloc de notas para escribir programas.

Podemos clasificar estos programas de creación de sitios web en dos categorías:

**WYSIWYG** (lo que ves es lo que obtienes - se trata de programas muy fáciles de usar, se

pueden crear sitios web sin tener que aprender el lenguaje en particular. Entre los más conocidos destaca Dreamweaver, su principal inconveniente es a menudo la bastante mala calidad del código HTML y CSS que se genera de forma automática mediante estas herramientas. Un diseñador de sitios web de buena calidad, tarde o temprano precisará conocimientos de HTML y CSS. No es recomendable el uso de estas herramientas.

**Editores de texto:** se trata de programas dedicados a la escritura de código. Uno por lo general, puede utilizarlos para varios lenguajes, no sólo HTML y CSS. Han demostrado ser poderosos aliados para los diseñadores sitios de web. Invitamos a usar un editor de texto en este curso.

## Algunos consejos, si estás frente a un ordenador con Windows, Mac OS X o Linux

### Windows

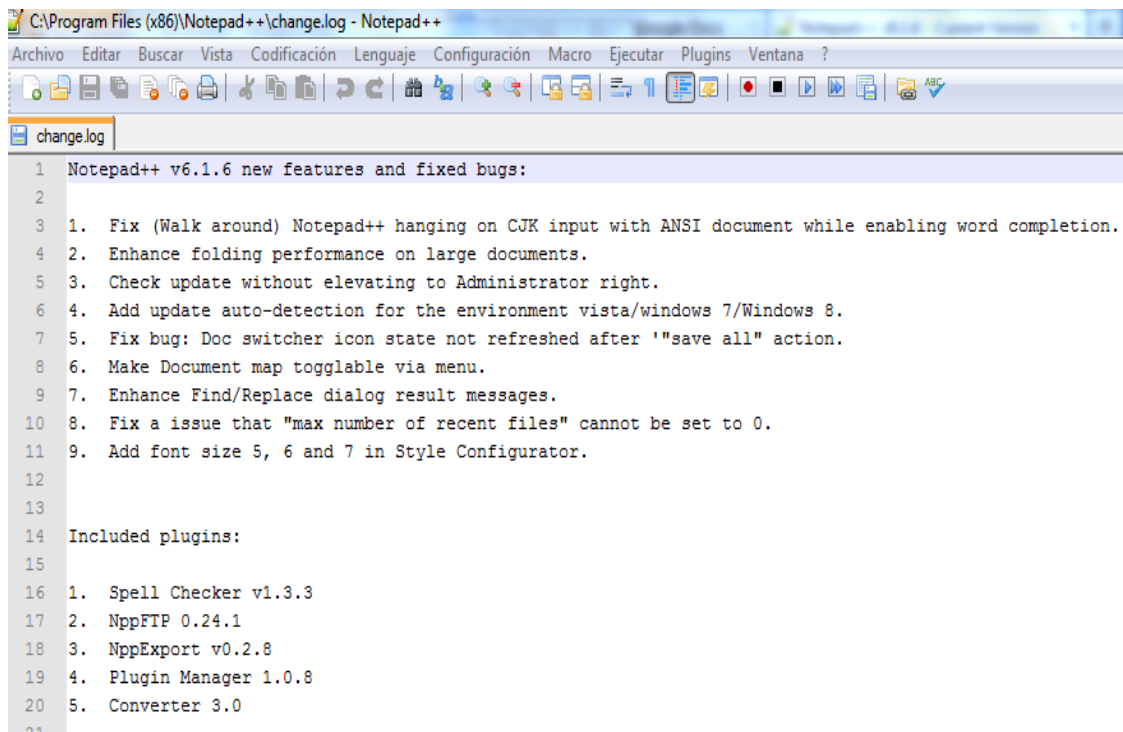
Hay muchos editores de textos, aquí no puedo presentar todos. Sin embargo, te invito a mirar Notepad ++, uno de los más usados entre los disponibles para Windows. Este software es sencillo y gratuito.

Para descargarlo (recuerda sólo hay versión para Windows), accede a Notepad ++ en el sitio <http://notepad-plus-plus.org/>. Aparecerá la opción *download* y accediendo a ella, aparecerá la pantalla mostrada a continuación, desde la que se iniciará el proceso de instalación como cualquier programa de Windows.



Una vez instalado se ejecuta accediendo en el escritorio al icono Notepad++, apareciendo el programa tal como se muestra a continuación:





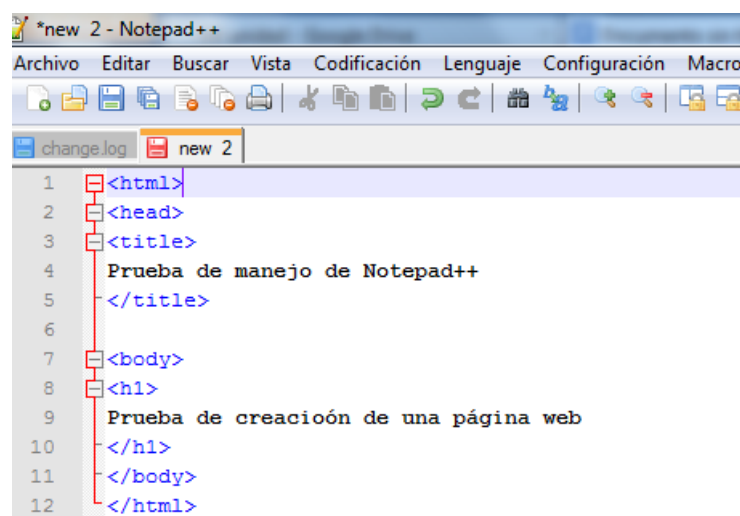
```

C:\Program Files (x86)\Notepad++\change.log - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Macro  Ejecutar  Plugins  Ventana  ?
change.log
1  Notepad++ v6.1.6 new features and fixed bugs:
2
3  1. Fix (Walk around) Notepad++ hanging on CJK input with ANSI document while enabling word completion.
4  2. Enhance folding performance on large documents.
5  3. Check update without elevating to Administrator right.
6  4. Add update auto-detection for the environment vista/windows 7/Windows 8.
7  5. Fix bug: Doc switcher icon state not refreshed after "save all" action.
8  6. Make Document map toggable via menu.
9  7. Enhance Find/Replace dialog result messages.
10 8. Fix a issue that "max number of recent files" cannot be set to 0.
11 9. Add font size 5, 6 and 7 in Style Configurator.
12
13
14 Included plugins:
15
16 1. Spell Checker v1.3.3
17 2. NppFTP 0.24.1
18 3. NppExport v0.2.8
19 4. Plugin Manager 1.0.8
20 5. Converter 3.0
21

```

Sugiero que hagas la siguiente operación: ir a [menú>lenguaje>HTML](#). Esto permitirá que el *software* sepa que vas a escribir código HTML, pues este lenguaje es uno de entre los muchos de los que permite Notepad++.

Cuando se utiliza el *software*, será el color de tu código (figura siguiente), el que permitirá identificaciones más fácilmente.



```

*new 2 - Notepad++
Archivo  Editar  Buscar  Vista  Codificación  Lenguaje  Configuración  Macro
change.log  new 2
1  <html>
2  <head>
3  <title>
4  Prueba de manejo de Notepad++
5  </title>
6
7  <body>
8  <h1>
9  Prueba de creación de una página web
10 </h1>
11 </body>
12 </html>

```

Por ahora, no te preocupes por lo que significa todo este galimatías. Solo quería dar una visión



general del *software*.

Hay otros editores de programas disponibles para Windows además de Notepad ++ Si no te conviene, puedes intentar el uso de:

Aptana  
Alleycode  
jEdit;  
PSPad;  
ConTEXT;

... y muchos otros si buscas "Editor de textos" en la Web.

## Mac OS X

Puedes probar uno de los siguientes programas:

jEdit;  
Smultron;  
TextWrangler.

## Linux

Los editores de texto son abundantes en Linux. Algunos de ellos están instalados por defecto, otros se pueden descargar fácilmente a través del centro de descargas (Ubuntu en particular) o mediante el uso de comandos como *apt-get*.

Algún software que puedes probar:

Aptana (recomendado)  
Quanta +  
gEdit;  
Kate;  
vim;  
Emacs;  
jEdit.

## Navegadores. ¿Por qué es importante el navegador?

El navegador es el programa que nos permite visualizar los sitios web. Como ya he explicado anteriormente, el trabajo del navegador es leer el código HTML y CSS (principalmente) para mostrar una salida en pantalla. Si el CSS indica al navegador: "Los títulos son rojos ", el

navegador mostrará los títulos en rojo.

El papel del explorador es esencial. Uno no lo haría, pero un navegador es un programa extremadamente complejo. De hecho, la comprensión del código HTML y CSS es una hazaña. El principal problema, que rápidamente te das cuenta es que los diferentes navegadores no muestran exactamente el mismo sitio de la misma manera. Has de tener el hábito de comprobar con regularidad que el sitio trabaja correctamente en la mayoría de los navegadores.

## Los navegadores en ordenadores tipo PC

Es aconsejable instalar varios navegadores en el ordenador para asegurarse de que el sitio funciona correctamente en cada uno de ellos. En general, recomiendo probar el sitio web sobre una base regular por lo menos Google Chrome, Mozilla Firefox, Opera e Internet Explorer .

La siguiente figura muestra una visión general del resultado producido por algunos de los principales navegadores en la página de Google.



## Entender las diferencias entre navegadores






Como se ha visto antes, los navegadores no siempre muestran el mismo sitio exactamente de la misma manera. ¿Por qué? Esto es porque los navegadores no siempre saben las últimas características de HTML y CSS. Por ejemplo, Internet Explorer (como es habitual en los productos Microsoft) siempre ha quedado a la zaga en algunas de las características de CSS (y paradójicamente, fue también por delante de algunas otras). Para complicar las cosas, coexisten múltiples versiones del navegador:

Firefox 2, Firefox 3.5, Firefox 3.6, Firefox 4, ... Firefox 18;  
 Internet Explorer 6, Internet Explorer 7, Internet Explorer 8, Internet Explorer 9;  
 Chrome 8, Chrome 9, Chrome 10, .... Chrome 25;  
 etc.

Cada versión es compatible con las nuevas características, pero si los usuarios no actualizan su navegador, la situación se convierte en un problema para los creadores de sitios web.

Chrome y Firefox han resuelto en gran medida el problema mediante la creación de actualizaciones automáticas, sin intervención del usuario. En Internet Explorer, los usuarios están aún menos incentivados para actualizar sus navegadores pues requieren actualizar a las últimas versiones del sistema operativo Windows (Internet Explorer 9 no está disponible para Windows XP, por ejemplo).

Sitios como *When can I use* <http://caniuse.com/> incluyen una lista de características compatibles con CSS de las diferentes versiones de cada navegador, como se indica en la siguiente figura.

Browser Rendering Engine	 Firefox Gecko					 Safari Webkit			 Chrome Webkit		 Internet Explorer Trident			 Opera Presto	
Version	3.6	4	5	6	7b-8a	4	5	5.1	9	10-14	6-8	9	10a3	10.6-11	11.1-11.5
Animations	✗	✗	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗	✗
Background Gradients	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗	⚠
Background Size	✓	✓	✓	✓	✓	⚠	✓	✓	✓	✓	✗	✓	✓	✓	✓
Border Image	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✗	✓	✓
Border Radius	⚠	⚠	⚠	⚠	⚠	⚠	✓	✓	✓	✓	✗	✓	✓	✓	✓
Box Shadow	✓	✓	✓	✓	✓	⚠	✓	✓	⚠	✓	✗	✓	✓	✓	✓
Columns	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗	✓
Font Face	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	⚠	✓	✓	✓	✓
HSLa	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
Hyphens	✗	✗	✗	✓	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗
Multiple Backgrounds	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓
Opacity	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✓	✓	✓

## Características gestionadas por los diferentes navegadores

Como puedes ver, es ... complicado. La mayor parte de las preocupaciones más a menudo provienen de las versiones anteriores de Internet Explorer (IE6, IE7, IE8). En este caso has de verificar cómo el sitio se presenta como las versiones anteriores ... ¡Espera lo inesperado! Comprueba especialmente que se muestre el sitio sin errores, sin tratar de obtener exactamente la misma representación en las versiones anteriores de estos navegadores.

Hay un programa para Windows llamado IETester (<http://www.my-debugbar.com/wiki/IETester>). Se puede consultar el sitio debido a las diferentes versiones de Internet Explorer. Ten en cuenta que este programa es relativamente inestable (se bloquea a menudo), pero tiene el mérito de existir.

## Los navegadores en el teléfono móvil celular

En la mayoría de los navegadores que he presentado, debemos saber que existen variantes de estos navegadores diseñados para teléfonos móviles celulares, especialmente teléfonos inteligentes y tabletas.

Más y más gente consulta los sitios web en el móvil o tableta, por lo que es necesario conocer el mínimo funcionamiento de los navegadores móviles. De hecho, no te sientas desorientado: la mayoría de los navegadores en los teléfonos inteligentes están en el mismo equipo, son una versión más ligera adecuada para móvil. Depende del tipo de teléfono, principalmente del sistema operativo.

**iPhone.** En el iPhone de Apple, se utiliza el navegador Safari Mobile. Esta es una versión ligera y sin embargo muy completa.

**Android.** Los Android se benefician de Mobile Chrome. De nuevo, esta es una versión adaptada al móvil.

**Windows Phone.** Internet Explorer Mobile. El principio es el mismo que para navegadores anteriores: se dedica a la versión móvil.

**Blackberry.** Los Blackberry son la excepción porque tienen su propio navegador (no hay equivalente en el ordenador). Sin embargo, las versiones más recientes de este navegador se basan en un núcleo común con Safari y Chrome (es el motor de renderizado Webkit). Por lo tanto, la pantalla es en general similar a la propuesta por Safari y Chrome.

Los navegadores móviles admiten la mayoría de las últimas características de HTML y CSS. La actualización automática ofrece garantías de que los usuarios de teléfonos celulares móviles están actualizados a las últimas versiones.

Entiende que existen diferencias entre estos navegadores en diferentes móviles, y es aconsejable probar el sitio en estos dispositivos también. En particular, la pantalla es mucho más estrecha, por lo que se verificará que tu sitio se muestra correctamente.

Las pantallas táctiles de las tabletas están equipadas con el mismo navegador, la pantalla es más grande. De este modo, por ejemplo el iPad se ofrece con Mobile Safari.

## En resumen

- La Web fue inventada por Tim Berners-Lee en la década de 1990.

- . Para crear sitios web, usamos dos lenguajes de programación:

HTML: Puedes escribir y organizar el contenido de la página (párrafos, encabezados ...);  
CSS permite dar formato a la página (color, tamaño ...).

- Ha habido varias versiones de HTML y CSS. Las últimas versiones son HTML5 y CSS3.
- El navegador web es un programa que muestra los sitios Web. Se lee el código HTML y CSS para saber lo que debe mostrar.
- Hay varios navegadores web: Google Chrome, Mozilla Firefox, Internet Explorer, Safari, Opera ...
- Cada uno muestra la página web de forma ligeramente diferente de otros navegadores.
- En este curso vamos a aprender a usar HTML y CSS. Vamos a trabajar en un programa llamado "editor de texto" (Aptana, Notepad ++, jEdit, vim ...).

## Tu primera página web en HTML

Todo lo necesario es que hayas instalado algún *software*. Debes tener un editor de textos para crear el sitio (como Notepad ++ o Aptana) y probar varios navegadores (Mozilla Firefox, Google Chrome, Opera ...).

En este capítulo vamos a empezar a practicar. Vamos a descubrir los fundamentos de HTML y editar nuestra primera página web. Así que, por supuesto, no esperamos lograr incluso una página web excepcional en este segundo capítulo, pero ten paciencia ... que vendrá.

### Crear una página web con el editor

Vamos a entrar en posición. Como he dicho, vamos a crear nuestro sitio en un editor de textos. El software tiene un objetivo simple: que puedas escribir texto.

Al seguir este curso, voy a trabajar en principio con Notepad++ y más adelante con Aptana. Ya vimos previamente como se empezaba a trabajar con Notepad++.

Bueno, ¿qué hacemos ahora? ¿Qué está escrito en esa hoja en blanco? Vamos a hacer una pequeña prueba. Invito a escribir, puedes escribir la misma frase que yo o la que quieras, el objetivo es escribir algo.

Ahora, grabar el archivo. Para ello, es muy simple: como en todos los programas, tiene un menú **Archivo > Guardar** (o Ctrl S). Un cuadro de diálogo preguntará dónde deseas guardar el archivo y con qué nombre. Guardar en el lugar donde se desee. Da el nombre del archivo que quieras, que termine en .html, pon `prueba.html`, por ejemplo.

Recomiendo crear una carpeta nueva con los documentos que contienen los archivos de tu sitio. Por mi parte, he creado una carpeta de prueba donde puse mi archivo prueba.html.

Ahora abre el explorador de archivos en la carpeta donde guardaste la página. Visualiza el archivo que acabas de crear. El archivo en el explorador de Windows se visualiza:



El icono que representa el archivo depende del navegador web por defecto. En este caso, el icono es el de Opera, mi navegador por defecto bajo Windows, pero el archivo puede tener otro. Tales iconos aparecen cuando el navegador principal es Firefox o Chrome, por ejemplo. Simplemente haz doble clic en este archivo y ... el navegador se abre y, en la siguiente figura,

se muestra el texto que escribiste.



El texto aparece en la misma línea, cuando desees que se escriba en dos líneas diferentes se necesita algo más. De hecho, para crear una página web no es suficiente con teclear el texto como has hecho. Además de este texto, se deben escribir de lo que se llaman etiquetas (*tags*), que le darán instrucciones al ordenador como "ir a la línea", "mostrar una imagen ", etc.

### Etiquetas y sus atributos

Es muy fácil. Obviamente, no basta escribir "sólo" texto en el editor, también darás instrucciones al ordenador. En HTML, se utiliza para eso las etiquetas.

### Etiquetas

Las páginas HTML se llena con lo que se denominan etiquetas. Estos son invisibles en la pantalla para los visitantes, pero permiten que el equipo entienda lo que se desea mostrar. Las etiquetas son fáciles de detectar. Están rodeadas de "galones", es decir, los símbolos < y >, de esta manera:

<etiqueta>

¿Para qué sirven? Indican la naturaleza del texto que encierran. Se refieren, por ejemplo: "Este es el Título de la página", "Esta es una imagen", "Este es un párrafo de texto", etc.

Hay dos tipos de etiquetas: las etiquetas huérfanas y las etiquetas de dos en dos

### Etiquetas de dos en dos

Se abren, contienen texto, y se cierran. Lo que aparece como:

<title> Este es un título </ title>

Distinguimos una etiqueta de apertura (<title>) y una etiqueta de cierre (</ title>) que indica que el título termina. Todo lo que no se encuentra entre estas dos etiquetas no es un título



Esto no es un título de <title> Este es un título </ title> Esto es sin título

## Etiquetas huérfanas

Estas son las etiquetas que más se utilizan para insertar un elemento en un lugar específico (por ejemplo, una imagen). No es necesario delimitar el principio y el fin de la imagen, sólo significa que el equipo "Inserte una imagen aquí."

Una etiqueta huérfana se escribe así:

```
<image />
```

Nota que al final / no es obligatorio en HTML5. Podrías escribir <image>. No obstante, para evitar el confundirse con el primer tipo de etiqueta, los diseñadores recomiendan este complemento barra diagonal (/) al final de las etiquetas huérfanas.

## Atributos

Los atributos son unas etiquetas referentes a opciones. Se complementan para proporcionar información adicional. El atributo se coloca después del nombre de la etiqueta y tiene el mayor valor, así:

```
<etiqueta atributo="valor">
```

¿De qué sirve? Toma la etiqueta <image /> que acabamos de ver. Sólo que no se utiliza mucho. Se podría agregar un atributo que especifique el nombre de la imagen a mostrar:

```
<image nom="foto.jpg" />
```

El ordenador entonces entiende que debe mostrar la imagen contenida en el archivo foto.jpg.

En el caso del funcionamiento de etiquetas "pares", se ponen los atributos en la etiqueta de apertura y no en la etiqueta de cerrado. Por ejemplo, este código indica que la cita es de Neil Armstrong y del 21 de julio de 1969:

```
<quote autor="Neil Armstrong" date="21/07/1969"> Este es un pequeño paso para el hombre, un salto gigante para la la humanidad.</ quote>
```

Las etiquetas tienen nombres reales en Inglés, que descubrirás más adelante.

## Estructura básica de una página HTML5

Volviendo a nuestro editor de texto. Os invito a escribir o copiar y pegar el código fuente mostrado a continuación. Este código corresponde a una página web en HTML5:

Código HTML:

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title> Página de prueba HTML5. Rafael Barzanallana </ title>
</ head>
    <body>
    </body>
</html>
```

Debes poner espacios al principio de algunas líneas para desplazar las etiquetas. No es obligatorio y no tienen impacto en la visualización en pantalla, pero hace que el código fuente sea más legible. Esto se llama sangría. En el editor, basta con pulsar la tecla de tabulación (Tab) para obtener el mismo resultado.

Te darás cuenta que las etiquetas se abren y cierran en un orden específico. Por ejemplo, el `<html>` es el primero que se abre y es también el último que se cierra (al final del código con `</html>`). Las etiquetas deben ser cerradas en el orden inverso de su apertura. Ejemplo:

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

Correcto. Una etiqueta que está abierta en el interior de otra también debe estar cerrada la interior.

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

etiquetas incorrectas, se entremezclan.

¿No podrías dar una explicación de todas las etiquetas que acabas de copiar en el editor? Por supuesto, no tengas miedo de ver todas estas etiquetas a la vez, voy a explicar su papel.

## Tipo documento

```
<!DOCTYPE html>
```

La primera línea es el tipo de documento. Es esencial porque es lo que indica que se trata de una página web HTML. En realidad no es una etiqueta como cualquier otra (pues comienza con un signo de exclamación), se puede considerar que es sólo la excepción que confirma la regla.

Esta línea fue alguna una vez un tipo de documento muy complejo. Era imposible retenerla en la cabeza.

Para XHTML 1.0, se tenía que escribir:

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

Como parte de HTML5, se decidió simplificarla, para el deleite de los desarrolladores. Cuando se ve una etiqueta de tipo corto (<!DOCTYPE html>), significa que la página está escrita en HTML5.

## Etiqueta <html>

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

Este es el código de la etiqueta principal. Abarca todo el contenido de una página. Como puedes ver, la etiqueta de cierre </html> se encuentra al final de código.

## Encabezado <head> y cuerpo <body>

Una página web consta de dos partes:

La cabecera <head>: esta sección proporciona información general sobre la página, como su título, la codificación (por gestión de caracteres especiales), etc. Esta sección suele ser bastante corta. La información en la cabecera no se muestran en la página, se trata de información general para el equipo. Es sin embargo, muy importante.

El cuerpo <body>, aquí es donde se encuentra la parte principal de la página. Todo lo que escribas aquí se mostrará en la pantalla. Este es el interior del cuerpo donde vamos a escribir la mayor parte de nuestro código.

Por el momento, el cuerpo está vacío (ver más adelante). Considera la posibilidad por contra de otras etiquetas en la cabecera.

## Codificación (*charset*)

```
<meta charset="utf-8" />
```

Esta etiqueta especifica la codificación utilizada en el archivo .html. Sin entrar en detalles, ya que esto puede convertirse rápidamente en complicado, la codificación indica cómo se guarda el archivo. Determina la forma en que los caracteres especiales se mostrarán (acentos, ideogramas chinos y caracteres japoneses, árabes, etc.).

Existen varias técnicas de codificación con nombres extraños, el lenguaje utilizado es según la norma ISO-8859-15, 775 OEM, Windows 1253 ... Sin embargo, uno debe ser utilizado en la actualidad tanto como sea posible: UTF-8. Este método de codificación muestra sin ningún tipo de problemas prácticamente todos los símbolos de todas las lenguas de nuestro planeta. Es por eso que pongo UTF-8 en esta etiqueta.

También se requiere que el archivo se guarde correctamente en UTF-8. Este es el caso con mayor frecuencia en Linux por defecto, pero, en Windows, debes decírselo al software en general.

Por si no lo haces para cada nuevo archivo, sugiero que vayas en Notepad++ al menú [Configuración>Preferencias> Archivo nuevo/Carpeta predeterminada](#). y selecciona UTF-8 sin BOM en la lista. Si tienes un problema de visualización en la página web, hay un problema con codificación. Asegúrate de que la etiqueta indica UTF-8 y que el archivo se guarda en UTF-8 (el editor de texto es capaz de decirle, Notepad++ lo hace en el menú de codificación).

## Título principal de la página

```
<title>
```

Este es el título de la página, probablemente el más importante. Cada página debe tener un título que describe lo que contiene. Es aconsejable mantener el título lo suficientemente corto (menos de 100 caracteres en general).

El título no aparece en la página visualizada, pero sí en la parte superior del mismo (a menudo en la pestaña del navegador). Guarda la página web y ábrela en tu navegador. Verás que el título aparece en la parte superior del navegador. Debes saber que el título también aparece en los resultados de búsqueda, por lo tanto la elección del título adecuado es importante.

## Comentarios

Hemos aprendido a crear nuestra primera página HTML real en este capítulo. Antes de concluir, me gustaría introducir el principio de los comentarios.

Es un texto que simplemente sirve como nota. No se muestra, no es leído por el ordenador, no hace alterar la visualización de la página. En pocas palabras, ¿es inútil?, bueno, si.

Esto es para que el programador y la gente que lee el código fuente de la página, pueda usar los comentarios para dejar información sobre el funcionamiento de la página.

¿Qué interés? Esto te ayudará a recordar cómo está elaborada tu página si regresas al código fuente después de un largo tiempo de ausencia. No te rías, nos sucede a todos los desarrolladores.

### Insertar un comentario

Un comentario es una etiqueta HTML con una forma muy especial:

```
<- Esto es un comentario ->
```

Puedes ponerlo donde quieras dentro del código fuente: no tiene ningún impacto en la página, pero se puede utilizar para ayudar a identificar en el código fuente (sobre todo si es largo).

```
<!DOCTYPE html>
<html>

<head>
    <- Encabezado ->
    <meta charset="utf-8" />
    <title> Página de prueba HTML5. Rafael Barzanallana </ title>
</head>
<- Fin de la cabecera ->

<body>
</ body>

</ html>
```

Cualquiera puede ver tus comentarios ... y todo el código.

Un punto importante: que todos pueden ver el código HTML de la página cuando se publique en una web. Simplemente haciendo clic botón derecho sobre la página y seleccionando "Ver código fuente de la página" (el texto indicado puede cambiar dependiendo del navegador). Se puede probar esta manipulación en cualquier sitio web, que sea funcional, 100% garantizado.

Esto se explica fácilmente: el navegador debe obtener el código HTML para lo que deseas

mostrar. El código HTML para todos los sitios es público.

¿La moraleja de la historia? Todo el mundo puede ver el código HTML y no puedes evitarlo. Por lo tanto, no pongas la información confidencial como contraseñas, en los comentarios ... y cuida de tu código fuente.

Cuando nos fijamos en el código de algunos sitios web, no tengas miedo si piensas que no cumple las mismas reglas que se presentan en esta documentación. Todos los sitios no están escritos en HTML5 (ni mucho menos) y, a veces, algunos desarrolladores escriben su código muy mal, no siempre son ejemplos a seguir.

## Resumen

- Usamos el editor de texto (Notepad + +, Aptana, jEdit, vim ...) para crear un archivo con la extensión html (por ej.: prueba.html). Esta será nuestra página web.
- Este archivo se puede abrir en el navegador del sistema operativo simplemente haciendo doble *clic* en él.
- Dentro del archivo, vamos a escribir el contenido de nuestra página, con las etiquetas html.
- Las etiquetas vienen en muchas formas:

`<etiqueta> </ etiqueta>` Se abren y cierran para delimitar el contenido (inicio y final de un título, por ejemplo).

`<etiqueta/>` etiquetas huérfanas (sin inserciones de una sola copia), que te permiten insertar un elemento en un lugar específico (por ejemplo, una imagen).

- Las etiquetas están a veces acompañada de atributos para proporcionar orientación adicional (por ejemplo: `<image name = "foto.jpg" />`).
- Una página web se compone de dos secciones principales: un encabezado (`<head>`) y cuerpo (`<body>`).
- Uno puede ver el código fuente de cualquier página web haciendo *clic* derecho y seleccionar: Ver la página de código fuente.

## Organizar el texto

La página en blanco es válida, pero tu sitio web es probable que tenga éxito limitado si lo dejas así. Vamos a descubrir muchas etiquetas HTML en este capítulo. Algunas han existido desde la primera versión de HTML, otras han sido recientemente introducidas en HTML5.

Vamos a ver, sucesivamente, en este capítulo:

- cómo escribir párrafos;
- cómo estructurar la página con los títulos;
- la forma de dar importancia a ciertas palabras del texto;
- cómo organizar la información en forma de lista.

### Párrafos

La mayoría de las veces, el escribir texto en una página web se realiza dentro de los párrafos HTML

HTML ofrece la etiqueta `<p>` para delimitar los párrafos.

```
<p> Hola y bienvenidos a mi web </ p>
<p> significa "comienzo del párrafo";
</p> significa "final del párrafo."
```

Como ya he dicho anteriormente se escribe el contenido de la página web entre la etiqueta `<body>` y la `</ body>`. Así que es suficiente poner nuestra sección de entre estas dos etiquetas y finalmente tendremos nuestra primera página web real con texto.

Toma exactamente el mismo código del capítulo anterior, y puedes añadir un párrafo:

```
<! DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title> párrafos </ title>
</ head>
<body>
  <p> Hola y bienvenidos a mi web </ p>
</body>
</html>
```



Inténtalo, verás el resultado. Esto es un buen comienzo.

Pero no se detiene allí el camino. Ahora vamos a ver algo un poco especial en HTML: el salto de línea. Suena simple y sin embargo, no funciona realmente como un procesador de textos ...

## Salto de línea

En HTML, si se pulsa la tecla *Enter*, no inserta una nueva línea como estás acostumbrado. Prueba este código:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title> Saltos de línea de prueba </ title>
</ head>

<body>
<p> ¡Hola y bienvenidos a mi sitio!
Esta es mi primera prueba, ser indulgentes.
Por favor, estoy aprendiendo poco a poco cómo funciona.
  Por ahora está un poco vacío, pero volver en tres días en los que
  habré aprendido unas cuantas cosas más, te aseguro que te sorprenderán
</ p>
</ body>
</ html>
```

Como era de esperar, sin embargo, hay una buena manera de hacer los saltos de línea en HTML. De hecho, si se quiere escribir un segundo párrafo, sólo hay que utilizar un `<p>` de segunda etiqueta.

El código HTML debe ser llenado con las etiquetas de párrafo final. Por ejemplo:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title> párrafos </ title>
</ head>
<body>
  <p> ¡Hola y bienvenidos a mi sitio!
  Esta es mi primera prueba
  Por favor, estoy aprendiendo poco a poco cómo funciona. </p>
```

```

    <p>Por ahora está un poco vacío, vuelve en tres días en los que he
    habré aprendido unas cuantas cosas más, te aseguro que te sorprenderás
  </ p>
</ body>
</ html>

```

El resultado lo podrás comprobar usando Notepad++ o cualquier otro editor.

Si sólo quiero ir a una línea de un párrafo y que salte una línea, sin pasar a un nuevo párrafo. Se dispone de `<br>` que es una etiqueta huérfana que se utiliza para decir que tenemos que saltar una línea.

```

<html>
<head>
  <meta charset="utf-8" />
  <title> Saltos de línea </ title>
</ head>

<body>
<p> ¡Hola y bienvenidos a mi sitio! <br />
Esta es mi primera prueba así que perdona.
Por favor, estoy aprendiendo poco a poco cómo funciona. </ p>
<p> Actualmente es un poco vacío, pero volver en tres
días en los que habré aprendido unas cuantas cosas más, te aseguro que
Te sorprenderás </p>
</ body>
</ html>

```

Teóricamente, puedes poner varias `<br />` en una fila para lograr saltos de varias líneas, pero se considera una mala práctica que hace que el código sea difícil de mantener. Para mover un texto con más precisión, se necesita el uso de CSS, el lenguaje que complementa al HTML del que se hablará más adelante.

Entonces, ¿qué está incluido?

```

</ p>: para organizar el texto en párrafos;
<br />: para ir a la línea.

```

Ahora que podemos escribir párrafos, vamos a ver cómo podemos crear títulos.

## Títulos

Cuando el contenido de la página se amplía con varios párrafos, será difícil para los visitantes

identificarlos. Aquí es donde los títulos son útiles.

En HTML, se pueden utilizar seis niveles diferentes de títulos. Quiero indicar que puedes decir: "Este es un título muy importante", "Este es un título un poco menos importante", "Este es un título mucho menos importante", etc. Tienes seis etiquetas diferentes de título: `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`

`<h1>`: significa "como muy importante". En general, se utiliza para visualizar el título de la página al principio.

`<h2>`: significa "tan importante".

`<h3>`: el mismo, es como algo menos importante (se podría decir, "subtítulo").

`<h4>`: como mucho menos importante.

`<h5>`: el título no es importante.

`<h6>`: título realmente, pero realmente no es importante en absoluto.

Precaución: No deben confundirse con la etiqueta del título. La etiqueta del título muestra la barra de título de la página, como hemos visto.

No te dejes engañar por todas esas etiquetas. De hecho, seis niveles de encabezados, es mucho. En la práctica, personalmente, yo sólo uso `<h1>` y `<h2>` `<h3>`, muy rara vez los otros (no se suelen necesitar seis niveles diferentes de titulares).

Un navegador muestra el título "muy importante" en tamaño muy grande, el título de "un poco menos importante" en tamaño algo menor, etc.

No elijas tu etiqueta del título de acuerdo con el tamaño que se aplica al texto. Es esencial para una buena estructura de la página, iniciar con un encabezado de nivel 1 (`<h1>`), entonces un encabezado de nivel 2 (`<h2>`), etc. No debe haber subtítulos sin título principal.

Si deseas cambiar el tamaño del texto, vamos a aprender a hacer esto con CSS un poco más adelante.

Trata de crear una página web con títulos para ver lo que sucede:

```
<html>
<head>
<meta charset="utf-8" />
<title> Niveles de títulos </ title>
</ head>

<body>
<h1>Título súper importante </ h1>
<h2> Título importante </ h2>
```

```

<h3> Título un poco menos importante</ h3>
<h4> Título no demasiado importante </ h4>
<h5> Título no importante </ h5>
<h6> Título que no es realmente importante </ h6>
</ body>
</ html>

```

Vamos, yo te doy un ejemplo del uso en un sitio web (verás que no estoy utilizando todas las etiquetas):

```

<html>
<head>
<meta charset="utf-8" />
<title> Presentación del sitio de Rafael Barzanallana </ title>
</head>

<body>
<h1> Bienvenido al Sitio de Rafael Barzanallana </ h1>

<p> Hola y bienvenidos a mi sitio.<br />
¿Qué es? el sitio de Rafael Barzanallana, </ p>

<h2> Cursos para principiantes </ h2>

<p> El sitio de Rafael Barzanallana ofrece el cursos (tutoriales)
para principiantes: no se precisan conocimiento previo para seguirlos.</p>
<p> Podrás aprender, aunque no sepas nada, a crear una página Web </ p>

<h2> Una comunidad activa</ h2>

<p> ¿Tienes problemas, un componente del curso que no
entiendes ¿Necesitas ayuda para crear tu sitio web? <br/>

¡Visita los foros! Descubrirás que no estás solo en este caso, y sin duda
encuentras a alguien que te ayude a solucionar el problema.</ p>
</body>
</html>

```

En estos momentos la página web va tomando forma. Sí, pero quiero centrar mi título, escribir y subrayar en rojo. Vamos a hacer todo esto cuando sepamos CSS (en la segunda parte del curso). Debes saber que <h1> no quiere decir "Times New Roman de 16 puntos", pero si "título importante".

Con CSS, puedes decir "quiero que mis títulos más importantes se centren, con texto color rojo y subrayados", en HTML, solo estructuramos nuestra página. Existen diferentes formas para resaltar el texto en tu página.

### Pon un poco de valor

Para poner un poco de valor en tu texto, debes utilizar la etiqueta `<em> </em>`. El funcionamiento es muy simple: incluir las palabras a destacar con estas etiquetas. A partir de un pequeño ejemplo anterior pongo algunas palabras destacadas:

```
<html>
<head>
  <meta charset="utf-8" />
  <title> Énfasis </ title>
</ head>

<body>
  <p> ¡Hola y bienvenidos a mi sitio! <br />
  Esta es mi primera prueba .
  Por favor ten paciencia, estoy <em>aprendiendo poco a poco<em>
  cómo funciona. </p>
</ body>
</ html>
```

Como podrás ver si ejecutas el código, utilizar la etiqueta `<em>` tiene el efecto de poner el texto en cursiva. De hecho, el explorador elige la forma de mostrar las palabras. Se dice que las palabras son muy importantes y para destacar esta información, cambia el aspecto del texto a cursiva.

### Texto importante

Para poner un texto como importante utiliza la etiqueta `<strong>` que significa "fuerte" o "importante" si así lo prefieres. Se utiliza exactamente de la misma manera que `<em>`:

```
<html>
<head>
  <meta charset="utf-8" />
  <title> Importante </ title>
</ head>

<body>
  <p> ¡Hola y bienvenidos a mi sitio! <br />
  Esta es mi primera prueba .
```

```

        Por favor ten paciencia, estoy <strong>aprendiendo poco a
        poco</strong> cómo funciona. </p>
    </ body>
</ html>

```

Seguramente verás que el texto aparece en negrita. De nuevo, la negrita es sólo una consecuencia. El navegador ha elegido para mostrar en negrita las palabras importantes que hacen que se destaque más.

La etiqueta <strong> no significa poner en negrita, sino "importante". Podemos decidir más tarde, con CSS, como poner las palabras "importantes" de una manera diferente a la negrita, si lo deseas.

### Marcar el texto

La etiqueta <mark> pone de manifiesto visualmente una parte del texto. El extracto no se considera necesariamente importante, pero se desea que se distinga claramente del resto de texto. Esto puede ser útil para resaltar un texto a buscar con posterioridad en el sitio, por ejemplo.

```

<meta charset="utf-8" />
<html>
<head>
    <title> Texto marcado </ title>
</head>

<body>
    <p> ¡Hola y bienvenidos a mi sitio! <br />
    Esta es mi primera prueba, <mark> sea indulgente </ mark>
    Por favor, estoy aprendiendo poco a poco cómo funciona. </p>
</ body>
</ html>

```

De forma predeterminada, el efecto de <mark> es resaltar el texto. Podemos cambiar la visualización de CSS (decidir resaltar, otro color, enmarcar el texto, etc.). Es el mismo principio que lo que dije antes de las etiquetas: indican el significado de las palabras y no la forma en que deben aparecer.

### Recuerda: HTML para el fondo, CSS para la forma

Yo tal vez parezca un poco pesado, pero es muy importante que entendamos bien, porque los principiantes a menudo cometen el mismo error en esta etapa. Vieron las etiquetas <em>, <strong>, <mark> ... y dicen: "muy bien, he descubierto cómo poner en cursiva, negrita, y la

forma de resaltar texto en HTML".

Y sin embargo ... esto no es así ¿por qué son estas etiquetas? Ya sé que vas a decir "Sí, pero cuando uso `<strong>` el texto aparece en negrita, por lo que es en negrita. Y sin embargo es un error pensar que esta etiqueta se utiliza para eso.

El papel de las etiquetas es para indicar el sentido del texto. Por lo tanto, `<strong>` dice al ordenador "Este texto es importante." Eso es todo. Y para mostrar que el texto es importante, el equipo decide ponerlo en negrita (pero puede ser que lo escriba en rojo).

La mayoría de los navegadores muestran el texto importante en negrita, pero nada les obliga. No entiendo. ¿Cuál es el punto en que el equipo sabe que un texto es importante? No es lo suficientemente inteligente como para comprender. Piensa otra vez. Muchos programas analizan el código fuente de páginas web, comenzando con los robots de los motores de búsqueda. Estos robots buscan en la web, leyendo el código HTML para todos los sitios. Este es el caso de las arañas de Google y Yahoo, por ejemplo. Las palabras clave "importantes" tienden a ser más valiosas para ellos, así que si alguien hace una búsqueda con estas palabras, es más probable que aparezcan a través de tu sitio.

Por supuesto, esta es una explicación aproximada y no pienses que el uso de la etiqueta `<strong>` en absoluto, va a mejorar tu aparición en un buscador. Basta con confiar en los ordenadores: ellos entienden lo que un texto "importante" y pueden utilizar esta información.

Entonces, ¿cómo hacer que la negrita, aparezca en rojo? Todo esto se hace con CSS.

Recuerda:

- HTML define la sustancia (el contenido lógico de los elementos);
- CSS define la forma (apariencia).

Veremos CSS más adelante, por ahora nos centramos en HTML y etiquetas, que cada una tiene un significado en particular.



## Listas

Las listas a menudo permiten estructurar mejor nuestro texto y ordenar nuestra información. Vamos a descubrir aquí dos tipos de listas:

- listas no ordenadas o listas de viñetas y
- listas ordenadas o listas numeradas o enumeraciones.

### Lista no ordenada

Una lista no ordenada se parece a esto:

- Fresas
- Frambuesas
- Cerezas

Este es un sistema que nos permite crear una lista de elementos sin ninguna noción de orden (no hay una "primera" o "última").

Crear una lista sin orden es muy simple. Sólo tienes que utilizar la etiqueta `<ul>` que se cierra con `</ul>`.

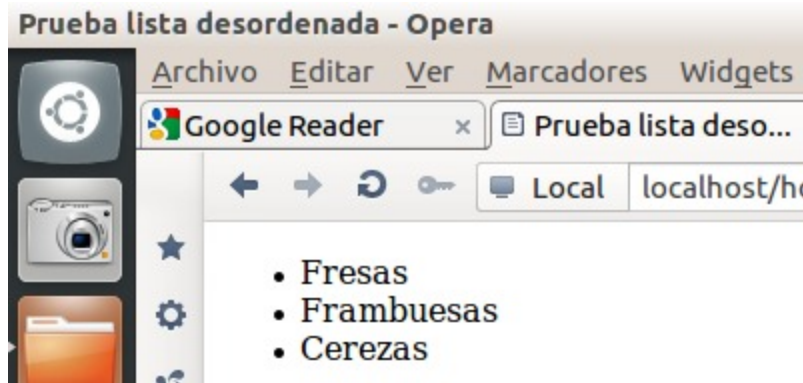
Así que empieza a escribir lo siguiente:

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

Y ahora, esto es lo que vamos a hacer: escribimos cada elemento de la lista entre dos etiquetas `<li></li>`. Cada una de estas etiquetas debe estar entre `<ul>` y `</ul>`. Se entiende de inmediato con este ejemplo:

```
<ul>  
<li> Fresas </li>  
<li> Frambuesas </li>  
<li> Cerezas </li>  
</ul>
```

El resultado es el siguiente:



Ten en cuenta que la lista debe estar colocada dentro de `<body> </body>`. A partir de ahora, no pongo todo el código de página para que sea legible.

Ten en cuenta estas dos etiquetas:

- `<ul> </ul>` definen toda la lista;
- `<li> </li>` definen un elemento de la lista.

Se pueden poner tantos elementos como se desee en la lista con viñetas, no se limita a tres elementos como se ha visto en el ejemplo.

Y ahora que sabes cómo crear una lista con viñetas. No es tan difícil entender cómo anidas las etiquetas.

Para aquellos que necesiten hacer listas complejas, sabemos que se pueden anidar las listas no ordenadas (crear una lista con viñetas en una lista con viñetas). Si deseas hacer eso, abre una segunda etiqueta `<ul>` dentro de un elemento `<li> </li>`.

Si cierras las etiquetas en el orden correcto, no habrá problemas. Sin embargo, esta técnica es un poco difícil de dominar.

## Lista ordenada

Una lista ordenada funciona del mismo modo, sólo hay que realizar un cambio de etiqueta: el bloque `<ul> </ul>` debe ser reemplazado por el bloque `<ol> </ol>`.

Dentro de la lista, nada ha cambiado: todavía utiliza etiquetas `<li> </li>` para delimitar los elementos.

El orden en el que colocar los elementos en la lista es importante. La primera `<li> </li>` será el elemento nº 1, el segundo será el nº 2, etc ...

Como es muy intuitivo, admira la simplicidad de este ejemplo (los resultados en la figura posterior):

```
<h1> Mi día de vagancia</ h1>
```

```
<ol>
```

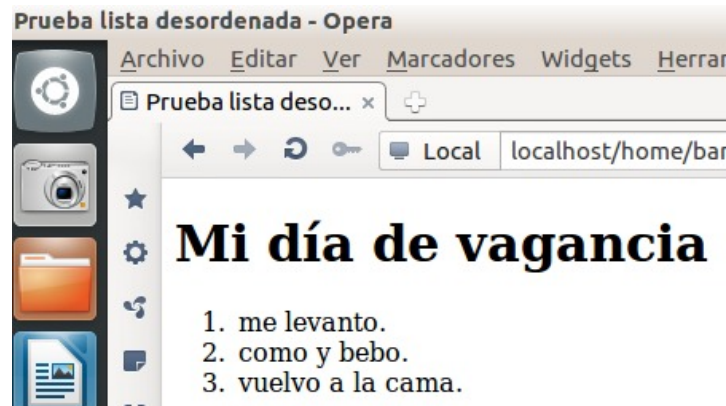
```
<li> me levanto. </ li>
```

```
<li> como y bebo. </ li>
```

```
<li> vuelvo a la cama. </ li>
```

```
</ ol>
```

En comparación con el ejemplo anterior, todo lo que tenías que cambiar es la etiqueta <ol>.



Para más información, hay un tercer tipo de lista, mucho más rara: una lista de definiciones. Se trata de la etiquetas <dl> (para delimitar la lista), <dt> (para definir un término) y <dd> (para delimitar la definición de este término).

## Resumen

- HTML tiene muchas etiquetas que nos permiten organizar el texto de nuestra página. Estas etiquetas dan indicaciones tales como "Este es un párrafo", "Este es un título", etc.
- Los párrafos se definen por la etiqueta </ p> y los saltos de línea por la etiqueta <br />.
- Hay seis niveles de encabezados, de <h1> </ h1> a <h6> </ h6> utilizarlos dependiendo del tamaño del título.

- Podemos enfatizar ciertas palabras con etiquetas `<strong>`, `<mark>` y `<em>`.
- Para crear listas, hay que usar la etiqueta `<ul>` (con viñetas, sin orden) u `<ol>` (lista ordenada). Los elementos se insertan con una etiqueta `<li>` para cada elemento.

## Crear enlaces

En el capítulo anterior, hemos aprendido a crear una página HTML simple. Aunque no era realmente atractiva, era una página HTML real de todos modos.

Como sabes, una página web consta de varias páginas. ¿Cómo pasar de una página a otra? Con vínculos, por supuesto. En este apartado nos limitaremos a aprender a crear vínculos entre nuestras páginas.

Supongo que todos saben lo que es un enlace: se trata de un texto sobre el que puedes hacer clic para ir a otra página.

Uno puede hacer un enlace desde una página b.html a la a.html, pero también se puede enlazar a otro sitio “externo” (por ejemplo, <http://www.barzanallana.es>). En ambos casos, se ve que la operación es la misma.

### Enlace a otro sitio

Es fácil de reconocer los enlaces de la página: aparecen en pantalla escritos de una manera diferente (por defecto, en azul y subrayado) y un cursor aparece al hacer *clic* en él.

Sugiero probar el código de enlace que conduce a mi página web personal, como la figura siguiente.



Para el enlace, la etiqueta que utilizamos es muy sencilla de recordar: `<a>`. Sin embargo, se debe agregar un atributo, `href`, para indicar la página del enlace a que debería llevar.

Por ejemplo, en el código de abajo hay un enlace que conduce a [barzanallana.es](http://www.barzanallana.es) situada en <http://www.barzanallana.es>. El código completo de la captura de pantalla mostrada es:

```

<head>
<title>Prueba de enlace al sitio de Rafael Barzanallana</title>
<meta name="GENERATOR" content="Quanta Plus">
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
</head>

<body>
<h1> Acceso a un sitio interesante</h1>

    <p>Buenos días</p>
    <p>¿Deseas visitar un sitio sobre <a href="http://www.barzanallana.es">
escepticismo</a>?</p>
    <p>Es útil para no ser engañado</p>

</body>
</html>

```

Por defecto, el enlace aparece resaltado en azul. Si ya has abierto previamente la página, el enlace se muestra en color morado. Veremos cómo esta apariencia cambia cuando se estudie CSS.

Si deseas crear un vínculo a otro sitio, sólo tenemos que copiar la dirección (llamada URL) en `http://`. Nótese que algunos enlaces a veces empiezan con prefijos `https://` (sitio seguro) o de otro tipo (`ftp://` ...).

Los enlaces que acabamos de ver se llaman enlaces absolutos, ya que muestra la dirección completa. Ahora veremos que podemos escribir los enlaces de una manera algo diferente, que será útil para hacer enlaces entre las páginas de nuestro sitio.

### Un enlace a otra página del mismo sitio

Hemos aprendido a crear enlaces a sitios ya existentes. Pero estoy seguro de que te gustaría hacer conexiones entre distintas páginas del mismo sitio.

Sí, exactamente, ¿cómo puedo hacer un enlace a otra página de mi sitio? No sé su dirección en `http://` ..., acabo de empezar a crear mi página web aquí, almacenándola en el disco duro del ordenador. En efecto, por el momento, vas a crear tu sitio en tu ordenador. Tu eres el único que puede verlo sin embargo, "una dirección web", comienza con `http://` como la mayoría de los sitios. Afortunadamente, esto no nos impide trabajar.

Dos páginas (archivos HTML) ubicadas en la misma carpeta o directorio.

Para empezar, vamos a crear dos archivos correspondientes a dos páginas HTML diferentes.

Como no estoy muy inspirado, los llamo pagina2.html y pagina1.html. Así que vamos a tener estos dos archivos en el disco en la misma carpeta.

¿Cómo establecer un vínculo desde la página 1 a la 2 sin http:// en la dirección? De hecho, es fácil: si ambos archivos se encuentran en la misma carpeta, basta con escribir el enlace como el nombre del archivo de destino al que quieres llevar. Por ejemplo: `<a href="pagina2.html">`. Se dice que es un vínculo relativo.

El código que vamos a utilizar para enlazar los archivos en HTML, pagina1.html y pagina2.html, es para la página1, por ejemplo:

```
<p>Buenos días. ¿Deseas consultar la <a href="pagina2.html">página 2 </a>?</p>
```

y el código en la 2, por ejemplo:

```
<h1>¡Bienvenido a la página 2!</h1>
```

Dos páginas (archivos HTML) en carpetas diferentes

Las cosas se ponen un poco difíciles si las páginas se encuentran en carpetas diferentes. Idealmente, no deben estar demasiado lejos una de la otra (en una subcarpeta, por ejemplo). Supongamos que la pagina2.html se encuentra en una subcarpeta llamada contenido.

En este caso, el vínculo debe ser escrito así:

```
<a href="contenido/pagina2.html">
```

Si hubiera varias subcarpetas anidadas, se debería escribir, por ejemplo, lo siguiente:

```
<a href="contenido/privado/pagina2.html">
```

Si el archivo de destino en el enlace se coloca en una carpeta que es "superior" en el árbol, debes escribir dos puntos (no :) como se indica:

```
<a href="../pagina2.html">
```

Los enlaces relativos no son muy complicados de utilizar una vez que se capta la idea. Basta con mirar a qué "nivel de la carpeta" está el archivo de destino para aprender a escribir tu enlace.

## Enlace a un ancla

Un ancla es una especie de lugar marcado que se puede poner en las páginas HTML cuando



son muy largas. De hecho, vale la pena poner enlaces a sitios posteriores en la misma página para que el visitante pueda ir directamente a la la parte de interés.

Para crear un ancla, sólo tienes que añadir el atributo de identidad para una etiqueta que a su vez servirá como punto de referencia. Esto puede ser cualquier etiqueta, por ejemplo un título.

Utiliza el atributo `id` para nombrar el ancla. Esto será útil entonces para hacer un enlace a ese ancla. Por ejemplo:

```
<h2 id="mi-ancla"> Título </ h2>
```

Luego, basta con crear un enlace, como de costumbre, pero esta vez el atributo `href` contendrá una almohadilla (`#`) seguida del nombre del ancla. Ejemplo:

```
<href="#mi-ancla"> Ir al ancla </ a>
```

Normalmente, si haces clic en el enlace, te llevará más abajo en la misma página (siempre que la página tenga el texto lo suficiente largo como para mover las barras de desplazamiento de forma automática).

Una página de muestra con una gran cantidad de texto y uso de anclas:

```
<h1>Mi gran página, sobre los fraudes de las pseudomedicinas</ h1>
<p>Ir directamente a la sección que trata sobre: <br />
<a href="#homeopatia"> Homeopatía</ a> <br />
<a href="#acupuntura"> Acupuntura</ a> <br />
<a href="#quirop practica"> Quiropráctica </ a> <br />
</ p>
<h2 id="homeopatía"> </ h2>
<p> ... (Hay un montón de texto) ... </ p>
<h2 id="acupuntura"> acupuntura</ h2>
<p> ... (Hay un montón de texto) ... </ p>
<h2 id="quirop practica"> quiropráctica</ h2>
<p> ... (Hay un montón de texto) ... </ p>
```

Si no sucede nada al hacer clic en los enlaces es que no hay suficiente texto. En este caso, puedes agregar a la página “blablabla” repetidas veces para aumentar la cantidad de texto, o reducir el tamaño de la ventana del navegador para mostrar las barras de desplazamiento en el lateral.

El atributo `id` se utiliza para dar un nombre "único" a una etiqueta, para hacer uso de puntos de referencia. Aquí se utiliza para enlazar a un ancla, pero en CSS, será muy útil para "identificar"

una etiqueta específica. Sin embargo, no crees id con espacios o caracteres especiales, sólo tienes que utilizar, en la medida de lo posible letras y números para que el valor sea reconocido por todos los navegadores.

### Enlace a un ancla situado en otra página

La idea es hacer un enlace que abre otra página y que lleve directamente a un ancla situado más abajo en esa página. En la práctica esto es bastante simple de hacer: basta con escribir el nombre de la página, seguido de un signo almohadilla (#), seguido por el nombre del ancla.

Por ejemplo: `<a href="anclas.html#parte2"> ...` te llevará a la página `anclas.html`, directamente en el ancla, denominado `parte2`.

### Estudios de casos prácticos de la utilización de enlaces

Voy a tratar de mostrar aquí algunos casos prácticos de uso de los enlaces. Por ejemplo, ¿sabes que es muy fácil hacer las conexiones que inician una descarga? ¿Crear un nuevo correo electrónico? ¿Quién abrirá una nueva ventana? Vamos a ver todo aquí.

- Un enlace que muestra una “burbuja” de información al pasar el ratón por encima.

Se puede usar el atributo de título que muestra una información sobre herramientas cuando se apunta a la relación. Este atributo es opcional. Se obtendrá un resultado como el siguiente gráfico.



La descripción puede ser útil para informar a los visitantes antes de que él hubiera hecho clic en el enlace, incluso.

Cómo reproducir este resultado:

```
Hola <p>. ¿ Te gustaría visitar el <a href = "http://www.escepticismo.es" title =  
"Escepticismo y pensamiento crítico"> sitio Escepticismo</ a> </ p>
```

- Un enlace que abre una nueva ventana

Es posible "forzar" abrir un enlace en una nueva ventana. Para ello, vamos a añadir `target =`

"\_blank" en <a>:

```
Hola <p>. ¿ Te gustaría visitar el <a href = "http://www.escepticismo.es" target =
"_blank"> Sitio Escepticismo</ a>
<br />
El sitio se abrirá en otra ventana. </ p>
```

Dependiendo de la configuración del navegador, la página se abrirá en una nueva ventana o pestaña. No se puede elegir entre abrir una nueva ventana o pestaña nueva.

Sin embargo no es recomendable abusar de esta técnica, ya que interfiere con la navegación. El propio visitante puede decidir si desea abrir el enlace en una nueva ventana. Si se pulsa *Shift* + *clic* el enlace se abra en una nueva ventana o *Ctrl* + *clic* para abrir en una pestaña nueva.

- Un enlace para enviar un e-mail

Si quieres que tus visitantes puedan enviar un correo electrónico, puedes utilizar los enlaces mailto. Nada de cambios en la etiqueta, sólo tienes que cambiar el valor del atributo href de esta manera:

```
<p><a href="mailto:ii@barzanallana.es">Enviar un correo electrónico</a></p>
```

Si haces *clic* en el enlace, aparece un mensaje en blanco que se abre una nueva ventana para ser enviado a su correo electrónico.

- Enlace para descargar un archivo

Muchos se estarán preguntando cómo va esto de descargar un archivo. De hecho, se debe proceder exactamente como si fuera un enlace a una página web, pero esta vez indicando el del nombre del archivo para descargar.

Por ejemplo, supongamos que deseas descargar apuntes.rar. Sólo tienes que colocar este archivo en la misma carpeta de tu página de internet (o una subcarpeta) y poner un enlace a este archivo:

```
<p><a href="apuntes.rar"> Descarga los apuntes</ a> </ p>
```

El navegador, al ver que no es una página web para mostrar, iniciará el proceso de descarga, cuando haces clic en el enlace.

## Resumen

- Los enlaces que permiten de cambiar de página aparecen, por defecto, escritos en azul y

subrayados.

- Para insertar un enlace se utiliza etiqueta `<a>` con el atributo `href` para especificar la dirección de la página de destino. Ejemplo: `<a href = "http://www.barzanallana.es">`.
- Uno puede hacer un enlace a otra página del mismo sitio, simplemente introduciendo el nombre del archivo: `<a href = "pagina2.html">`.
- Los enlaces también pueden llevar a otros lugares en la misma página. Tenemos que crear un ancla con el atributo `id` para "Marca" en algún lugar de la página, a continuación, vincular al ancla de esta manera: `<a href="#ancla">`.

## imágenes

Insertar una imagen en una página web es, es increíblemente fácil. Hay diferentes formatos de imágenes que se pueden utilizar en los sitios web, y no debemos elegir al azar. De hecho, las imágenes son a veces muy grandes para descargar, lo que ralentiza el tiempo de carga de la página (mucho más que el texto).

Para asegurarse de que tus páginas se pueden leer y ser rápidas de descargar, sigue activamente mi consejo.

### Los diversos formatos de imagen

¿Sabes lo que es un formato de imagen? Cuando tienes una imagen a tu alcance, se puede guardar en varios formatos diferentes. El peso (en KB o incluso MB de la imagen será mayor o menor dependiendo del formato elegido y la calidad de la imagen).

Por ejemplo, el programa de dibujo Paint (aunque esto está lejos de ser el mejor) puede elegir entre varios formatos al guardar una imagen. Hay algunos programas que nos permiten la elección entre una amplia variedad de formatos estándares.

Algunos formatos son más adecuados que otros, dependiendo del tipo de imagen (fotografía, dibujo, cine ...). Nuestro objetivo aquí es dar una vuelta por diferentes formatos utilizados en la Web para conocerlos y saben elegir el que se adaptará mejor a tus imágenes. Ten la seguridad, que no hay muchos formatos diferentes, por lo que no será muy largo.

Todas las imágenes publicadas en internet tienen en común que se comprimen. Esto significa que el ordenador hace cálculos para reducirlas y por lo tanto sean más rápidas de cargar.

### El formato JPEG

Las imágenes en formato JPEG (*Joint Photographic Expert Group*) están muy extendidas en la Web. Este formato está diseñado para reducir el peso de imágenes (es decir el tamaño del archivo asociado), que puede comprender más de 16 millones de colores diferentes. La figura siguiente es una imagen guardada como un archivo JPEG.



Las imágenes JPEG se guardan con la extensión. jpg o jpeg.

Se ha de considerar que JPEG deteriora un poco la calidad de la imagen, de una manera generalmente imperceptibles. Eso es lo que lo hace tan efectivo para reducir el tamaño de las fotografías.

Cuando se trata de una imagen, por lo general no puedes detectar la pérdida de calidad. Por contra, si no es una imagen, puede ver la imagen un poco borrosa. En este caso, debes utilizar el formato PNG.

## **PNG**

El formato PNG (*Portable Network Graphics*) es el más nuevo de todos. Este formato es adecuado para la mayoría de los gráficos (estoy tentado a decir "cualquier cosa que no es una imagen"). PNG tiene dos grandes ventajas: se puede hacer transparente y no afecta a la calidad de la imagen.

El PNG se inventó para competir con otro formato, el GIF, del que había que pagar derechos de propiedad por usarlo. En efecto, PNG ha evolucionado y se ha convertido en el formato más potente para grabar la mayoría de las imágenes.

PNG se presenta en dos versiones, dependiendo del número de colores que la imagen debe incluir:

PNG de 8 bits: 256 colores;

PNG de 24 bits: 16 millones de colores (como una imagen JPEG).

Una versión antigua de Internet Explorer (IE6) no puede mostrar correctamente imágenes de 24-bit PNG transparentes. Este explorador tiende a ser menos y menos utilizado, pero aún así

mantener esta información en mente.

De hecho, si el formato PNG de 24-bit puede mostrar tantos colores como una imagen JPEG, y además se puede hacer transparente, sin afectar la calidad de la imagen ... ¿cuál es la ventaja de JPEG?

La compresión JPEG es la más poderosa para las fotos. Una foto guardada como JPEG se cargará mucho más rápida que si se ha guardado como PNG. Sugiero reservar el formato JPEG para las fotos.

## **GIF**

Este formato es bastante viejo, estaba siendo ampliamente utilizado (y sigue siendo ampliamente utilizado por costumbre). Hoy en día, PNG es en general, mucho mejor que las imágenes GIF, son generalmente más ligeros y la transparencia es mejor. Recomienda utilizar PNG si es posible.

El formato GIF está limitado a 256 colores (mientras que el PNG hasta millones de colores). Sin embargo, el GIF mantiene una cierta ventaja sobre el PNG: que no pueden ser animados.

## **WebP**

Es un formato gráfico en forma de contenedor, que soporta tanto compresión con pérdida como sin ella . Está siendo desarrollado por Google, basándose en tecnología adquirida con la compra de On2 Technologies. Como derivado del formato de vídeo VP8, es un proyecto hermano del formato WebM , y está liberado bajo la licencia BSD.

## **Hay un formato adecuado para cada imagen**

En resumen, se ha de considerar el formato de acuerdo con la imagen que se disponga:

Una foto: usar JPEG

Gráficos con pocos colores (menos de 256): Usar PNG de 8-bits o, posiblemente, GIF.

Gráficos con muchos colores: Utilizar un archivo PNG de 24 bits.

Imagen en movimiento: utilizar un GIF animado.

## **Errores que deben evitarse**

- Desterrar otros formatos

Otros formatos no mencionados aquí, como formato de mapa de bits (BMP) están prohibidos porque a menudo no están comprimidos, por lo tanto, son demasiado grandes. No están del todo adaptados a la Web. Podemos ponerlos en un sitio pero la carga es muy tediosa.

- Seleccionar cuidadosamente el nombre de la imagen

Si quieres evitar problemas, toma el hábito de guardar los archivos con los nombres en minúsculas, sin espacios o guiones, por ejemplo: mifoto.png.

Se pueden sustituir los espacios con el carácter de subrayado (" \_ ")

## Insertar una imagen

Ahora, de vuelta al código HTML para descubrir la manera de colocar las imágenes en nuestras páginas.

¿Cuál es la etiqueta famosa que nos permitirá insertar una imagen? Esta es `<img />`. Se trata de un tipo de etiqueta huérfana (como `<br/>`). Eso significa que no es necesario escribirla en pares como la mayor parte de etiquetas que se han visto hasta aquí. En efecto, no es necesario definir una parte del texto, sólo queremos insertar una imagen en un lugar específico.

La etiqueta debe ir acompañada de dos atributos necesarios:

- `src`: le permite especificar el lugar donde se ubica la imagen que desees insertar. Se puede configurar una ruta absoluta (por ejemplo, `http://www.um.es/logo.jpg`) o relativa a la ruta (esto se suele hacer). Así que si la imagen se encuentra en una subcarpeta de imágenes, por ejemplo habría que escribir: `src = "imagenes/um.png"`
- `alt`: significa "texto alternativo". Siempre se debe especificar un texto alternativo a las imágenes, es decir, un texto breve que describe lo que contiene la imagen. Este texto se muestra en lugar de la imagen si no puede ser descargada (esto puede suceder), o en los navegadores de las personas con discapacidad visual, que por desgracia no se puede "ver" la imagen. También ayuda a los motores de búsqueda para las búsquedas de imágenes. Para `um.png`, lo haría por ejemplo: `alt = "Universidad de Murcia"`.

Las imágenes deben estar obligatoriamente dentro de un párrafo (`</p>`). Un ejemplo de inserción de la imagen:

Código: HTML

```
<p>
Aquí está una foto que tomé durante mis últimas vacaciones en la playa: <br />

</p>
```



</p>

Pasa el ratón sobre la imagen para ver la descripción, aparece "Foto de la playa de Luarca". En pocas palabras, la inserción de la imagen es algo muy fácil siempre que sepamos cómo indicar donde está la imagen, al igual que se ha aprendido a hacer con los enlaces.

El mayor "problema" (si se le puede llamar un problema) es elegir la relación de aspecto correcta.

## Imagen con enlace

Si la imagen es muy grande, es aconsejable mostrar en tu sitio una miniatura de la imagen. A continuación, se agrega un enlace en esta miniatura por lo que los visitantes puedan ver la imagen en tamaño original.

Hay muchos programas para la creación de miniaturas de las imágenes. Por ejemplo Easy Thumbnails. Lo usaré para obtener dos versiones de una imagen, como la figura siguiente: la miniatura y la imagen original.



El origen y la imagen en miniatura las pongo en una carpeta llamada img, por ejemplo. Que muestra la versión de montagne\_mini y montagne, puedes hacer un enlace a montagne.jpg que es la imagen ampliada que se muestra al hacer pulsar en la miniatura.

El código HTML que uso para esto:

<p>

¿Quieres ver la imagen en su tamaño original? Clic por encima <br />

<a href="img/montagne.jpg"> <img src = "img/montagne\_mini.jpg" alt = "Foto de montañas nevadas" title = "Pulsa para agrandar" /> </ a>

</ p>

A veces, algunos navegadores optan por mostrar un color azul (o color púrpura) muy poco

estético alrededor de la imagen en la que puedes hacer clic. Afortunadamente, podemos eliminar este marco, gracias a CSS.

## Las figuras

Al leer esta documentación, ya te has encontrado varias veces con figuras. Estas son los elementos que vienen a enriquecer el texto para complementar la información de la página.

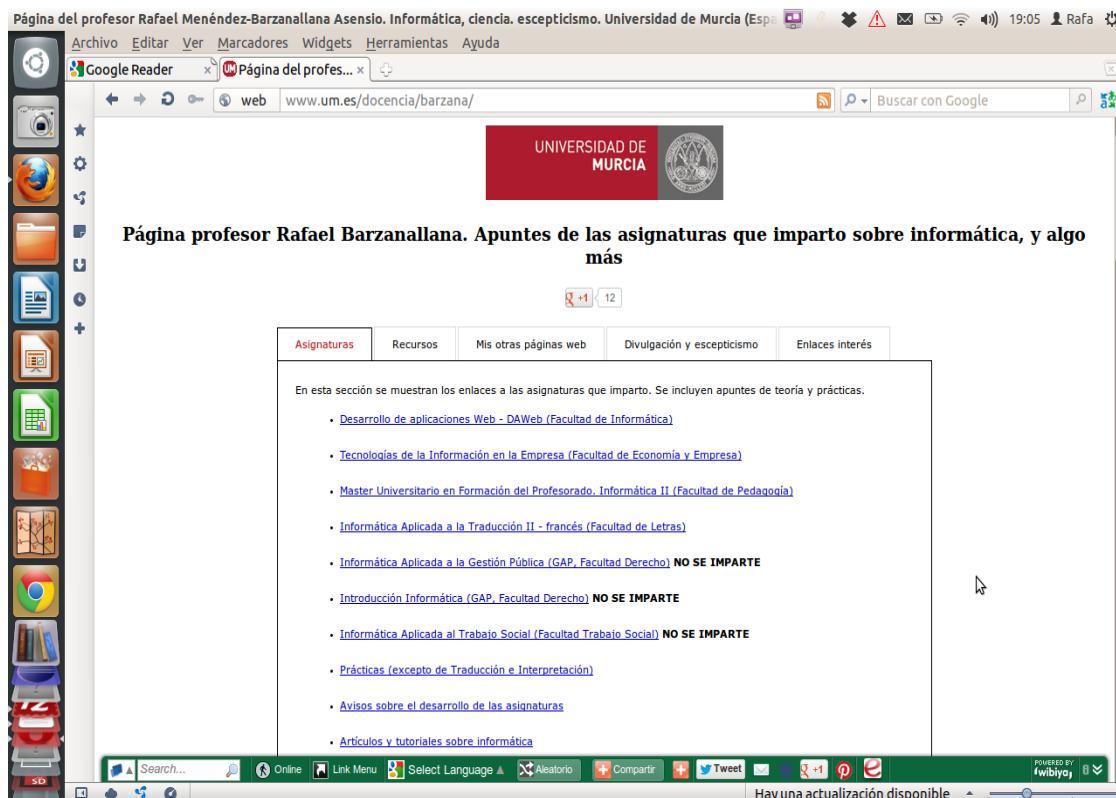
Las figuras pueden ser de diferentes tipos:

imágenes, código fuente, citaciones, etc.

En resumen, todo lo que sirve para ilustrar el texto es una figura. Vamos a ocuparnos aquí de las imágenes, al contrario de lo que era de esperar, las cifras no son necesariamente imágenes: un código fuente también ilustra el texto.

## Creación de una figura

Recordemos el ejemplo de captura de pantalla del primer capítulo, representado en la figura siguiente.



En HTML5, se dispone de la etiqueta <figure>. Cómo se podría utilizar:

Código: HTML

```
<figure>
  
</figure>
```

Una figura suele ir acompañada de una leyenda. Para añadir un título, utiliza la etiqueta <figcaption> dentro de la etiqueta <figure>, de esta manera:

Código: HTML

```
<figure>
  
  <figcaption> Escudo de la universidad de Murcia</figcaption>
</figure>
```

## Entender el papel de las figuras

Anteriormente en este capítulo, dije que las imágenes deben estar ubicadas en los párrafos (que se encuentra en el interior de una etiqueta <p>). No es del todo cierto. Si haces una foto del rostro, la imagen puede estar localizado fuera de un párrafo.

Código: HTML

```
<p> ¿Conoces el programa Quanta +? Podemos hacer los sitios web con esta
aplicación</p>
<figure>
  
  <figcaption> Software Quanta + </figcaption>
</figure>
```

Realmente no cambian. ¿Cuándo debo poner mi imagen en un párrafo y cuando debo colocarla en una figura? Buena pregunta. Todo depende de lo que contiene la imagen con el texto:

- Si no proporciona ninguna información (es sólo una ilustración para decorar): colocar la imagen en un párrafo.
- Si ofrece información: colocar la imagen en una figura.

La etiqueta <figure> tiene un papel principalmente semántico. Esto significa que se le dice al

ordenador que la imagen tiene sentido.

Es importante entender el texto. Esto puede permitir que un programa recupere todas las figuras del texto y hacer referencia a ellas en una tabla de figuras, por ejemplo.

Por último, ten en cuenta que una figura puede tener varias imágenes. Un caso en el que esto se justifica:

Código: HTML

```
<figure>  
  <img src = "img/opera.png" alt = "Logo de Opera"/>  
    
    
  <figcaption> Logos de diferentes navegadores </ figcaption>  
</figure>
```

## Resumen

- Hay varios formatos de imagen adaptados a la web:

JPEG para las fotos;

PNG: para todos los otros ejemplos;

GIF: similar a PNG, con un número más limitado de colores, pero que pueden ser animados.

- Inserta una imagen con la etiqueta <img />. Se debe incluir al menos dos atributos: src (nombre de la imagen) y alt (breve descripción de la imagen).

- Si una imagen ilustra el texto (y no sólo es decorativa), es aconsejable colocarla dentro de una etiqueta <figure>.

- La etiqueta <figcaption> permite escribir el título de la imagen.