

ISSD

I2

Informática 2

Módulo didáctico - 2015

—*Asc*—
Analista de
Sistemas

HTML5
¡ALLÁ VAMOS!



Unidad 1



Clase 2



El primer paso: HTML5



| Comprender cuáles son los requerimientos o conocimientos previos con los que debes contar para comenzar con el desarrollo del lenguaje HTML5.
| Entender de forma detallada la sintaxis del lenguaje HTML5 y sus etiquetas semánticas.
| Comprender las tags relacionadas con párrafos, formatos en general, como así también sus atributos y propiedades.
| Entender y aprender a trabajar con imágenes y formatos en el ámbito del lenguaje HTML5.

¿Y porque debemos comenzar por HTML y no otro lenguaje?

La programación web dinámica, simple y sencilla, es una de las tendencias actuales del desarrollo web, dejando de lado las animaciones flash, que si bien tienen un atractivo visual interesante, no son buenas para el posicionamiento web, ya que no cuentan con texto que soporte el código web. Por eso es importante que comiences aprendiendo la estructura básica de un sitio para luego pasar a lenguajes de desarrollo web más complejos.





HTML5... ¡Allá vamos!

Como vimos en la clase anterior, comenzaremos con el estudio de HTML5. La idea es que le dediques tiempo a la materia para practicar, es decir, mientras más horas tengas programando más experiencia vas a adquirir como desarrollador web. No hay otra forma de aprender más que de los errores que seguro vas a cometer al comenzar a trabajar con el código, eso es algo normal que se irá depurando a medida que vayas realizando ejercicio tras ejercicio.

La intención es que codifiques en forma limpia y que no utilices ningún editor web, ya que si dependés de este no aprenderás las etiquetas, atributos y propiedades del lenguaje HTML.



Desempeño de Exploración

En este punto te proponemos debatir sobre los conceptos expuestos en clases y también que busques ejemplos y relaciones con otras materias. La intención es que comiencen a perfilarse todas las metas de comprensión y tópicos de la materia.



Desempeño de Investigación

También realizaremos investigaciones sobre los temas expuestos en clase en forma grupal para debatir en los encuentros subsiguientes. Armaremos galerías de imágenes, trabajaremos con herramientas auxiliares como editores fotográficos, paleta de colores, fuentes, etc.

¿Qué necesitás para empezar a programar?

No necesitas muchos requerimientos para comenzar, entre ellos te nombraré algunos:

| Editor de texto para escribir el código fuente: puede ser Notepad ++ (que lo puedes descargar de cualquier sitio de descarga) o Bloc de notas (aplicación que trae incorporada windows). Existen otros editores, podés utilizar otro con el que te sientas cómodo.

| Galería de imágenes: esta te ayudara para salir al paso en caso de que no tengas conexión a internet, arma una con formatos .jpg, .png, .gif (dichos formatos son los que soporta HTML).

| Un editor de imágenes: si necesitás retocar alguna imagen podés utilizar cualquier

aplicación que uses en Multimedia 1 o Diseño Web (Photoshop, Fireworks, etc. o alguna con la que te identifiques o te sea fácil de usar).

| Un servidor (Host): podés conseguir uno gratuito subscribiéndote a él, o podés adquirir uno pago. La idea es que tengas tu propio espacio para subir tus trabajos y presentarlos en caso de que lo necesites. Este ítem no es obligatorio en la materia

| ¡Pilas, onda y ganas para dedicarle a la materia!

Comenzando con HTML5

¿Qué son las etiquetas?

Es algo sencillo, son elementos dentro del código de hipertexto que funcionan como una especie de marca, que le otorga a la sección encerrada dentro de la etiqueta una característica específica, como color de fuente, tamaño, tipo de elemento y hasta la sección que ocupa dentro del archivo HTML, o en este caso HTML5. Las etiquetas se inician de la siguiente manera:

`<nombre_etiqueta>`

y se cierran con:

`</nombre_etiqueta>`



Las etiquetas son elementos dentro del código de hipertexto que funcionan como una especie de marca, que le otorgan a la sección encerrada dentro de ella una característica específica.

¿Qué son etiquetas semánticas?

Semántica: Perteneciente o relativo a la significación de las palabras. Esa es la definición de la Real Academia de la Lengua Española y así de sencillo es entender este tipo de etiquetas del lenguaje HTML: Son aquellas que dan un significado a las partes del documento.

Por decirlo de otra forma, estamos ante etiquetas que indican qué es el contenido que contienen, en lugar de como se debe formatear al mostrar el documento HTML en un cliente web.

En HTML5 se han incorporado muchas etiquetas que sirven para indicar qué son y cuál es el significado de los elementos que contienen dentro. Esas son las etiquetas semánticas, que ganan mucha importancia en el marco del HTML y de la composición de un documento web por ayudar a motores de búsqueda como Google a indexar más correctamente los contenidos de un sitio. Dentro del etiquetado semántico también tenemos varias funciones, pero para los que nos dedicamos a maquetar páginas web, hay un conjunto de ellas que nos interesan especialmente. Son las que sirven para definir el esquema principal del documento, como

HEADER, ARTICLE, FOOTER, etc. Todas esas etiquetas semánticas nos indican qué es el contenido que engloban y cuál es su relación con el conjunto de elementos del documento HTML.

Consejo: Utiliza nombres en tus archivos que tengan algunas normas básicas para ahorrarte disgustos y líos. Nuestro consejo es que no utilices acentos ni espacios ni otros caracteres raros. También te ayudará escribir siempre las letras en minúsculas. Esto no quiere decir que debes hacer nombres de archivos cortos, es mejor hacerlos descriptivos para que te aclaren lo que hay dentro. Algún carácter como el guion "-" o el guion bajo "_" te puede ayudar a separar las palabras. Por ejemplo quienes_somos.html.

En este sencillo ejemplo que muestra la estructura de un documento HTML5, observarás que la mayoría de las directivas aparecen en parejas y rodean un bloque de texto u otras directivas.

Las directivas están colocadas entre los símbolos menor (<) y mayor (>). La directiva de inicio de cada par indica al navegador donde comienza la instrucción, y la directiva de final le indica donde termina. Las directivas de cierre siempre incluyen una barra (/) después del símbolo menor (<).

¡Manos a la obra!

Vamos a comenzar la clase aplicando los siguientes pasos:

1 | Abrir el Programa *editor de Texto* (en este caso podemos utilizar *Notepad++* o cualquier editor que tengas a mano).


2 | Escribiremos los códigos o "etiquetas". Las instrucciones, códigos o "tags" de **html** van entre dos signos < y >. Es indiferente escribir las instrucciones en mayúsculas o minúsculas. Vas a encontrar mucha bibliografía donde el código esta todo en Mayúscula, esto permite diferenciar las etiquetas del contenido de la página.

Para crear una página web sencilla tenemos que escribir en nuestro editor (Bloc de notas) las siguientes líneas de código:


```
<!DOCTYPE HTML>
<html>
<head>
<title>Mi primera página</title>
</head>
<body>
<p>Mi primer párrafo</p>
</body>
</html>
```

3 | Guardá el documento asignándole un nombre con la extensión (.html) ejemplo index.html

4 | Para visualizar el resultado en el navegador hacé doble click en el archivo html que acabás de guardar recién, o bien, abrí el archivo desde el menú de opciones de tu navegador. Podés ver los resultados en los 3 navegadores por excelencia que son Chrome, IE, y Mozilla.



¿Qué hace falta para que una página web se vea bien?



Hace falta incluir por lo menos las directivas que componen la estructura básica de un documento HTML5

Estas directivas fundamentales son:

| La primera línea de código es la declaración `<!DOCTYPE HTML>` de un documento HTML. Esta etiqueta indica qué versión de HTML se utiliza en el documento, para que los navegadores puedan aplicar la gramática y sintaxis correspondiente al él. La versión de HTML5 simplifica mucho esta directiva. Las versiones anteriores HTML usaban declaraciones más complicadas para varios formatos HTML. Por ejemplo en la versión de Html 4.01 strict, la directiva es:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN" "http://www.w3.org/TR/html4/strict.dtd">
```

Y en la versión Html 4.01 transitional es:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">
```

| La directiva `<html>` aparece una vez al principio de la página web y de nuevo con una barra añadida (`</html>`) al final y le indica al navegador que todo lo que se encuentra entre ellas es código html y se debe interpretar

como tal. Si pensamos en una página web como si fuera un árbol, la directiva `<html>` sería la raíz de todo. Desde la raíz salen dos ramas que representan a las dos partes principales de cualquier página web, el encabezado (`<head>`) y el cuerpo (`<body>`).

| Con la directiva `<head>` y `</head>` se delimita la cabecera del documento que suele contener información sobre el mismo como título, idioma, palabras clave, autor, etc. La información que se especifique en la cabecera no es mostrada por el navegador al usuario con la única excepción de la etiqueta `<title>` y `</title>`

| La directiva <title> y </title> sirve para especificar el título del documento y es visualizada en la barra de título de la ventana correspondiente al programa navegador.

| La directiva <body> delimita el cuerpo de un documento html y engloba todo el contenido que se ve en una página web: titulares, texto, imágenes, listas, tablas, formularios.

Estructura de una web en HTML5

Anteriormente, a la hora de diseñar una web, había ciertos elementos o partes de la página que eran claramente identificados y, a grandes rasgos, eran aceptados por la mayoría de la comunidad, ya sean usuarios o diseñadores. Nos referimos a la cabecera (header), al pie de página (footer), a la barra lateral (aside).

Sin embargo, en ediciones previas estos elementos eran diseñados en la web como **simples divisiones mediante el uso de la etiqueta <div>**, teniendo que identificar cada una de ellas como lo que realmente eran. Así teníamos una lista de etiquetas divs, cada una con su correspondiente identificador, al

que teníamos que darle nosotros mismos: <div id="header"> o <div id="footer">. Sin embargo, también podríamos llamarlos <div id="cabecera"> o <div id="pie-de-pagina">, con lo que cada diseñador tenía su propia forma de identificar las partes de sus webs.

Efectivamente, parece lógico que estos elementos, que siempre se repiten en todos los diseños web que trabajamos, tuvieran sus propias etiquetas semánticas e identificativas. Así pues, pasamos de un <div id="cabecera"> a un <header>, mucho más limpio, fácil para los diseñadores y muy útil para los buscadores como Google, que podrán identificar de forma más rápida la estructura de una web.

Las etiquetas de HTML5 que normalmente siempre están presentes en todos los diseños web que realizamos son:

<header></header>:

Es el elemento que abre la web. Normalmente lleva ubicado el logotipo o el texto representativo. En los casos que la web lleve publicidad, también se suele ubicar el banner de cabecera dentro de él. En muchos casos contiene al menú de navegación <nav> aunque no necesariamente.

El *elemento header* representa “un grupo de ayudas de navegación”. Es decir que cualquiera que sea el contenido que estuviera acostumbrado a incluir dentro de `<div id=“header”>` ahora lo tiene que incluir en el nuevo elemento header.

Lo más novedoso de este elemento es que podemos incorporar un header en cada sección del contenido, más concretamente en cualquier bloque de contenido que pueda necesitar su propio encabezado. Un elemento header se situará frecuentemente al principio de una página o sección y también se puede utilizar para incluir contenido introductorio o ayudas de navegación que son específicas de cualquier sección de una página, o bien que se aplican a toda la página o ambas.

Veamos las siguientes etiquetas: (1)

El elemento nav representa un grupo de vínculos de navegación. Aunque el uso más común será el de englobar una lista no ordenada de vínculos. También se podría situar alrededor de un párrafo de texto que contenga los principales vínculos de navegación para una página o sección de la misma.

No se recomienda que el elemento nav se utilice para una lista breve de vínculos en un pie de página. En cualquier caso, el elemento nav se debería reservar para la navegación.

El elemento section se describe como una representación de una sección genérica de un documento o aplicación. Una sección, en este contexto, es una agrupación temática de contenido, normalmente con un encabezado.

(2)

El contenido del elemento section debería ser temático, por lo que sería incorrecto emplearlo de forma genérica para agrupar piezas no relacionadas con el contenido.

Ejemplos de uso para elementos section:

| Secciones individuales de una interfaz agrupada en pestañas.

| Segmentos de una página “sobre mi”, “contactar”.

| Diferentes partes de una extensa página “términos de servicio”.

| Varias secciones de un sitio web de noticias: artículos, novedades, noticias internacionales, deportes, etc.

`<nav></nav>`:

Representa a todos los enlaces de navegación que llevan a las distintas secciones de nuestra web. Es una de las partes más importante, pues debe ser fundamental para que el usuario sea capaz de navegar por las categorías y páginas.

`<section></section>`: Es la etiqueta que viene a sustituir por defecto al `<div>`, y lo que indica es que estamos en una sección distinta, como si de un capítulo de un libro se tratase.

`<article></article>`: La etiqueta `article` está diseñada para contener una unidad propia de información y contenido, que a su vez puede tener su cabecera (header), cuerpo (section) y pie de artículo (footer). Un `section` puede contener varios artículos independientes.



Es posible anidar elementos `section` dentro de elementos `section` ya existentes. Por ejemplo, en una web de noticias online, el `section` de noticias del mundo se podría dividir aún más en un `section` para cada región del mundo.

El elemento `article` representa una composición autónoma en un documento, página, aplicación o sitio y, en principio se distribuye o reutiliza de forma independiente, por ejemplo en sindicación.

El elemento `article` es similar al elemento `section` pero la diferencia principal es la siguiente: mientras un `section` puede contener cualquier contenido que se pueda agrupar temáticamente, un `article` debe ser una sola pieza de contenido que pueda valerse por sí mismo, por ejemplo si un contenido se puede volver a publicar en otro sitio sin ser modificado o enviado como una actualización (RSS, twitter, facebook), tiene cualidades de un artículo.

Ejemplos de usos del elemento `article`:

- | Mensajes de foro.
- | Artículos de revista o periódicos.
- | Entradas de blog.
- | Comentarios enviados por usuarios.

Los elementos `article` también se pueden anidar. Otra opción disponible es anidar a una `section` dentro de un `article` y viceversa. (1)

`<aside></aside>`: Viene a sustituir al sidebar o barra lateral, aunque su uso es para llenarlo de contenido que no esté relacionado directamente con esa página, por ejemplo un listado de enlaces, anuncios, iconos sociales, etc.

```

2 <aside>
3   <h2>Switzerland</h2>
4   <p>Switzerland, a land-locked country in the middle of geographic
5   Europe, has not joined the geopolitical European Union, though it is
6   a signatory to a number of European treaties.</p>
7   <a href="#">Lorem Ipsum dolor sit amet</a>
8   <p>Switzerland, a land-locked country in the middle of geographic
9   Europe, has not joined the geopolitical European Union, though it is
10  a signatory to a number of European treaties.</p>
11 </aside>

```

Este elemento `aside` representa una parte de la página que está relacionada con el contenido. Se podría utilizar para englobar una parte del contenido que es tangencial a:

- | Un contenido específico independiente (como un artículo o sección).
- | Toda una página o documento, como se acostumbra a hacer cuando se añade una "barra lateral" a una página o sitio web. El elemento `aside` nunca se debería emplear

para englobar secciones de la página que forman parte del contenido principal.

Ejemplos de usos del elemento `aside` (1)

- | Barra lateral.
- | Lista secundaria de vínculos.
- | Bloque de publicidad.

El **elemento footer** representa un pie de página para la sección de contenido. Una página puede constar de varios footer. Típicamente contiene información acerca de el autor de la sección, datos de derechos de autor o enlaces a documentos relacionados. Ejemplos de usos del elemento footer (1):

- | Información de copyright.
- | Listas de vínculos relacionados.
- | Información del autor.

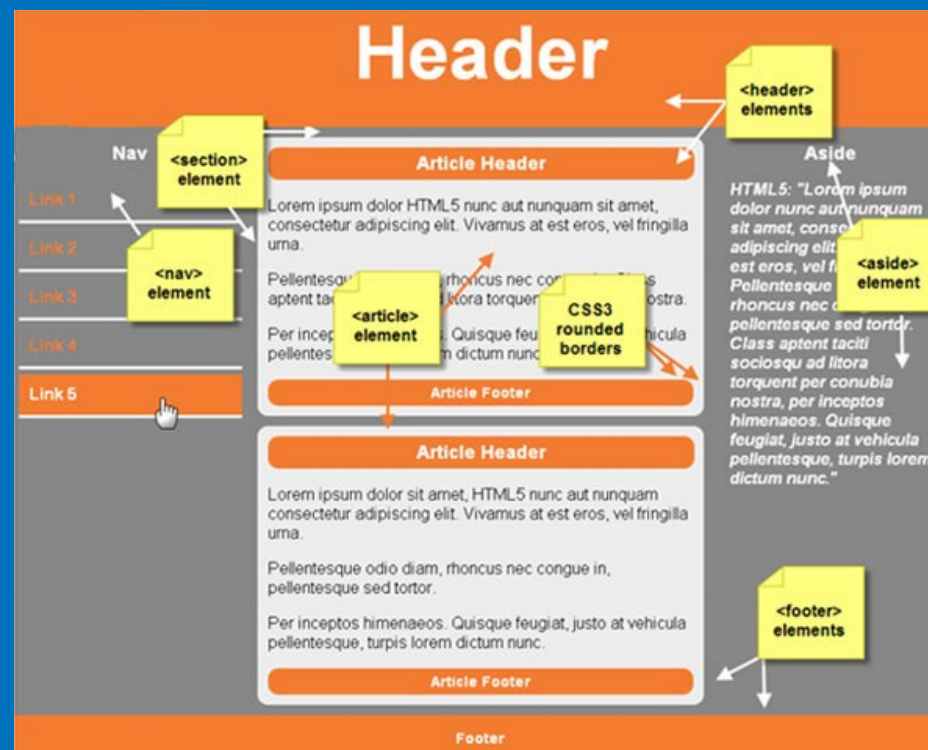
A modo de resumen, podemos ver lo que venimos trabajando en la Imagen (2)

<footer></footer>: Representa el pie de página, presente en casi todas las webs. Nos sirve para ubicar los datos de contacto, algo de información extra, iconos sociales, actualizaciones de twitter, etc. Permite inclusive un segundo menú de navegación.

```

2 <footer>
3   <nav>
4     <p>
5       <a href="/credits.html">Credits</a>
6       <a href="/tos.html">Terms of Service</a>
7       <a href="/index.html">Blog Index</a>
8     </p>
9   </nav>
10  <p>Copyright © 2014 LoremIpsum.com</p>
11 </footer>
12

```



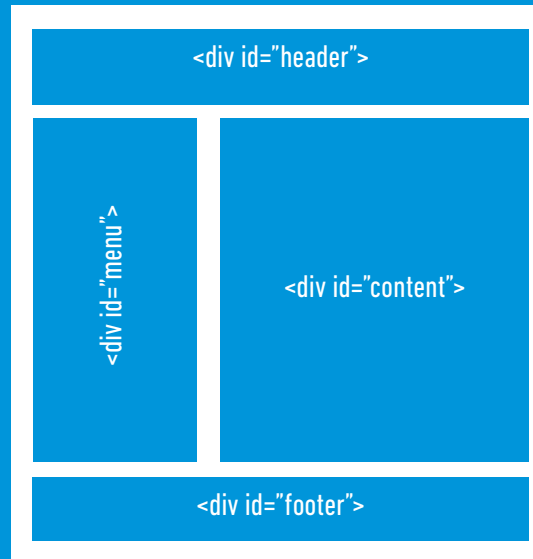
En HTML 4.01 hubiésemos usado algo como lo que se puede observar en la imagen (1).

Mientras que en HTML 5 podemos hacer más explícito el contenido de cada sección en lugar de usar identificadores para los DIVs tal como se observa en la imagen (2).

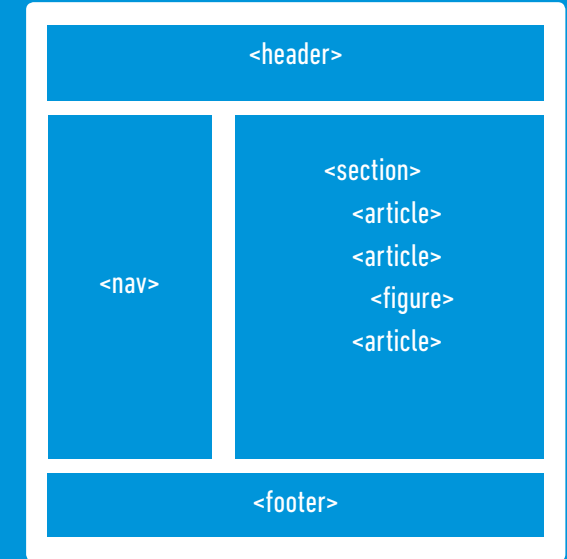
Claramente podemos adivinar que HEADER sirve para definir el encabezado del sitio, NAV contiene los enlaces para navegar y FOOTER la información del pie de página.

Luego aparecen una serie de etiquetas para especificar dentro del área de contenido el tipo de información que agregamos. SECTION obviamente marca el inicio y fin de una sección, ARTICLE sirve para indicar cada artículo (muy útil en un blog, por ejemplo) mientras que FIGURE es usado para señalar que un elemento se trata de una ilustración que acompaña un texto. Estos elementos pueden ser tanto una imagen, como un video o un audio. (3)

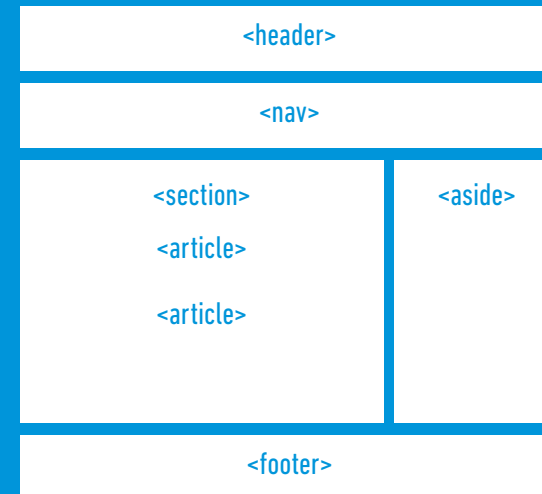
1



2

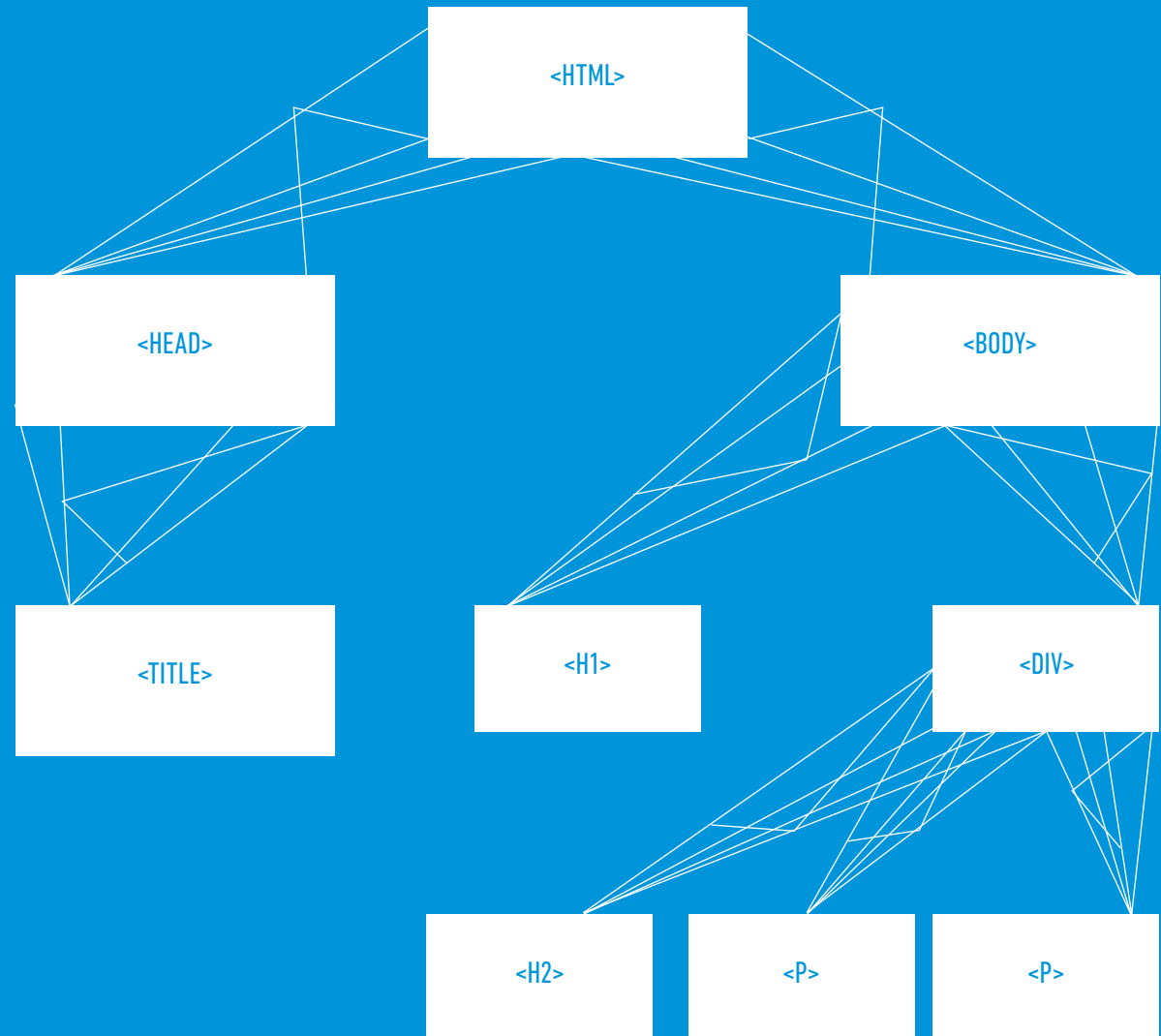


3



Concepto DOM

El *Modelo en Objetos para la Representación de Documentos* o **DOM** (Document Object Model) es un estándar de objetos para representar documentos *HTML* y *XML*. A través de él, los programas pueden acceder y modificar el contenido, estructura y estilo. (1)



Presta atención al gráfico (1). La etiqueta `<html lang="es">` que podemos leer en la imagen indica el tipo de lenguaje con el que se trabaja, es= español.

Etiquetas y atributos

A continuación te invitamos a abordar las etiquetas más utilizadas y sus atributos. Las **etiquetas** pueden presentar modificadores que llamaremos atributos, los que nos permitirán definir diferentes posibilidades de la instrucción **HTML**. Estos **atributos** se definirán en la etiqueta de inicio y consistirán normalmente en el nombre del atributo y el valor que toma, separados por un signo de igual.

El orden en que se incluyan los atributos es indiferente y no afecta al resultado. Si se incluye varias veces el mismo atributo con distintos valores, el resultado obtenido será imprevisible dependiendo de cómo lo interprete el navegador. Cuando el valor que toma el atributo tiene más de una palabra, deberá expresarse entre comillas, en otro caso no será necesario. En lo que sigue te proponemos profundizar este tema con ejemplos. Tené

en cuenta que es posible añadir atributos a determinados elementos.(2)

PERO... ¿Qué es un atributo?

Como recordarás, los elementos sirven para estructurar el contenido de todo el documento HTML5 e indican al navegador cómo debería presentarse el sitio web (por ejemplo, `
` informa al navegador que debe introducir un salto de línea). En algunos elementos se puede añadir más información y esa información extra se denomina atributo.

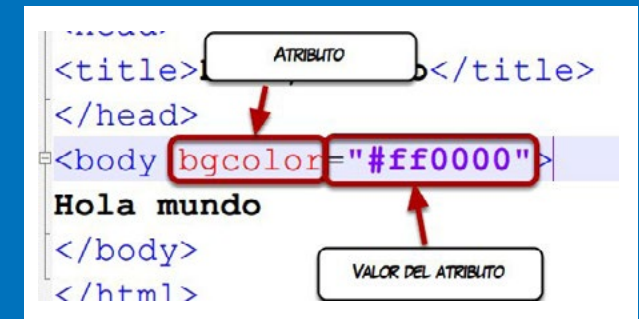
Ejemplo:

`<body bgcolor="#ff0000">`

Los atributos se incluyen siempre dentro de la etiqueta de inicio y llevan inmediatamente detrás el signo igual ("=") con los valores de dichos atributos entre comillas (""). El punto y coma que sigue al atributo se usa para separar los diferentes comandos de estilo. Es importante que utilices la misma ortografía que usamos en los ejemplos; de otro modo, los navegadores no serán capaces de

Documento html5

```
<!DOCTYPE html>
<html lang="es">
<head>
  <title>Documento HTML5</title>
  <meta-charset="UTF-8">
</head>
<body>
  <p>Espacio para el contenido</p>
</body>
</html>
```



interpretar el código. Además, no olvides cerrar siempre las comillas detrás del atributo ya que ese es un error común en los programadores que recién comienzan con este código.

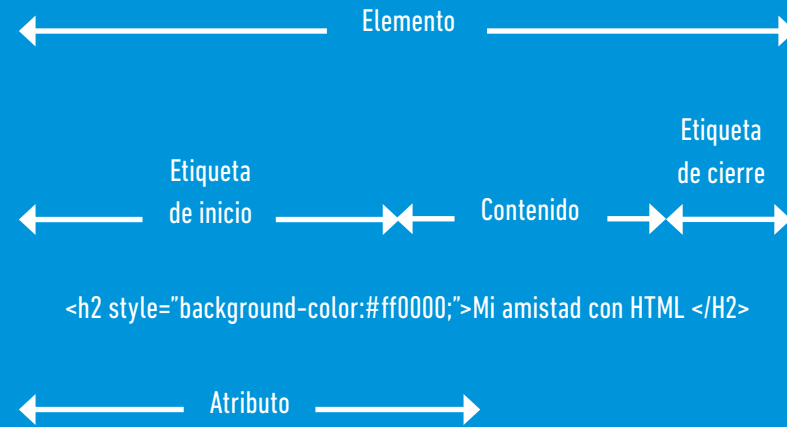
¿De qué partes se compone exactamente un elemento?

Por lo general, todo elemento se compone de una etiqueta de inicio, sin atributos o con uno o más atributos, el contenido correspondiente y una etiqueta de cierre. Así de sencillo como podemos ver en la imagen (1).

Etiquetas Correctas

Todo texto que se encuentre entre los caracteres < y > se considerará una etiqueta, si la etiqueta no fuera una de las válidas del lenguaje HTML5 no será tomada en cuenta, sin causar ningún tipo de error, dejándose el texto o las etiquetas a las que afectaba como si no existiera la etiqueta extraña.

Cuando se comete un error sintáctico al expresar una etiqueta o un atributo, no se producirá ningún error, simplemente no se obtendrá el efecto que deseábamos.



Consejo: También es interesante prestar atención a cómo se vería el documento en los distintos tamaños de la ventana del navegador, teniendo en cuenta que las personas tienen monitores con diferentes resoluciones.

¿Cuál es mejor y cual es peor? ¿El formateo físico o el planteamiento lógico?

Como todo en esta vida, depende del caso concreto.



Etiquetas lógicas y físicas

| Los **estilos físicos** formatean cada porción de texto atendiendo tan sólo a cómo debe aparecer finalmente en pantalla, sin preocuparse en absoluto en qué peso o papel desempeña cada parte del documento en el esquema total del mismo.

| ****: muestra el texto en un tipo de letra más grueso.

| **<i>**: muestra el texto inclinado (en cursiva).

| **<u>**: subraya el texto.

| **<tt>**: utiliza un tipo de letra no proporcional.

| **<big>**: muestra el texto con un tipo de letra más grande.

| **<small>**: utiliza un tipo de letra menor.

| **<s>**: utiliza un tipo de letra tachado.

| **<sub>**: escribe como subíndice.

| **<sup>**: escribe como superíndice.

| Los **estilos lógicos**, se caracterizan porque en ellos hacemos un estudio lógico de cada parte del documento, observando la importancia que tiene en el global del mismo distinguiendo entre cabeceras, citas, subíndices, comienzos de capítulo, etc., por lo que formatearemos cada parte del documento de acuerdo a su importancia en el conjunto global.

| ****: muestra el texto en un tipo de letra más fuerte, normalmente en negrita.

| ****: muestra el texto enfatizado, normalmente en cursiva.

| **<code>**: suele utilizarse para mostrar algún tipo de código, suele usar un tipo de fuente no proporcional.

| **<blockquote>**: bloque de texto indentado.

A la hora de dar formato a nuestro texto web podemos hacerlo partiendo de dos planteamientos diferentes, cada uno de los cuales presenta sus ventajas y sus inconvenientes, por lo que muchas veces tendremos que recurrir a una mezcla de ambos tipos de estilos si nuestro diseño así lo exige.



```
<html>
<head>
<title>Anidamiento de etiquetas</title>
</head>
<body>
<b><i><u>Texto con etiquetas anidadas</u></i></b>
<br>
<b><i><u>Texto con etiquetas mal anidadas</b></u></i></b>
</body>
</html>
```

Veamos esto con más detenimiento. Generalmente el **formateo físico** está indicado en aquellos casos en los que andamos apurados de espacio para presentar un texto o unos datos concretos, en esas situaciones en que deseamos una presentación fija, que se muestre en pantalla siempre igual sea cual sea el navegador y la resolución del usuario. Esto es así ya que este tipo de formateo nos permite un mayor control absoluto sobre los resultados.

Por otro lado, el **planteamiento lógico** es adecuado en aquellos casos en que deseamos obtener documentos con unos esquemas constructivos perfectamente definidos, que puedan sufrir variaciones dependiendo de la resolución y versión del navegador conservando su estilo en conjunto,

manteniéndose cada parte del documento con un estilo adecuado a su papel en el total del mismo.

Etiquetas Anidadas

Hasta ahora hemos visto que las etiquetas HTML5 suelen ir en parejas y también vimos que es posible incluir unas etiquetas dentro de otras. Esto es lo que precisamente se llama anidamiento. Por ejemplo:

```
<b>negrita</b>
<i>cursiva</i>
<b><i>cursiva y negrita</i></b>
<b><i>cursiva, negrita <small>y
pequeña</small></i></b>
```

El **anidamiento** puede darse poniendo las etiquetas en cualquier orden, pero siempre debe guardar simetría.

Esto es incorrecto:

```
<b><i>.....</b></i>
```

Y esto es la forma correcta:

```
<b><i>.....</i></b>
```

Entonces *¿cómo es la estructura general de las etiquetas anidadas?*

Ahora avancemos un poco más, te sugerimos que vos también vayas haciendo lo que te vamos mostrando así te es más fácil recordarlo.



Propiedades del Body

La directiva **<BODY>** además de indicar el inicio y el final del cuerpo de nuestra página, permite indicar el fondo de dicho documento. Si no incluimos ningún parámetro, el fondo de nuestro documento será de color blanco. Podemos cambiar el color de fondo con el parámetro **bgcolor** de la siguiente forma:

<BODY bgcolor=" código de color ">

El código de color empieza con el símbolo # y continua con seis cifras hexadecimales. Cada par de estas cifras indican el dominio de cada

color principal en la mezcla final. Las dos primeras cifras indican el dominio del color rojo, las dos siguientes del color verde, y las dos últimas del color azul. El usar un número hexadecimal de dos cifras para cada color nos permite indicar dominios desde el 0 (00) al 256 (FF). Por ejemplo, para que el fondo fuese de color verde puro, la orden sería:

<BODY bgcolor="#00FF00">

Aparte del color, podemos indicar un fondo gráfico para el documento. Para ello utilizaremos el parámetro **background**. El fondo vendrá indicado por un archivo de

imagen (gif o jpg) que se reflejará en la página en mosaico. Por ejemplo:

<BODY background="fondo.gif">

Pondrá como fondo de nuestro documento la imagen fondo.gif repitiéndola tantas veces como sea necesario para que cubra toda la página.

Lógicamente, cuanto mayor sea la imagen, más tiempo tardará el navegador en visualizar el fondo de nuestra página.

Párrafos

¿Qué es una página web sin texto? Nada, ¿Verdad?

Para gestionar párrafos podemos hacerlo con la etiqueta P y en la medida que lo incluyamos entre la etiqueta de apertura <p> y la de cierre </p> será tratado por el navegador como un párrafo.

Ahora bien, supongamos que queremos resaltar una parte del texto por ser más relevante ¿cómo lo hacemos? Las dos vías más utilizadas son las siguientes:

| Etiqueta *strong*: lo que incluyamos entre sus etiquetas de apertura y cierre se mostrará, como forma habitual, en negrita. Un sustituto, aunque no muy utilizado, es el elemento *b*. La etiqueta strong, como su nombre indica, le da más *fuerza* a la parte resaltada, destacando ese texto entre los demás.

| Etiqueta *em*: Al igual que la etiqueta strong, sirve para resaltar una parte del texto, pero en esta ocasión no le daremos más fuerza o peso, sino que mayor *énfasis* (del inglés emphasis).

No es estrictamente necesaria la etiqueta de cierre </P>, ya que los navegadores asumen que comienza un nuevo párrafo en cuanto se encuentran con una etiqueta <P>, salvo en el caso de que párrafos sucesivos usen una alineación diferente, en cuyo caso sí que es necesaria cada etiqueta de cierre. De todas formas, y como regla general en todo el código HTML5, es siempre conveniente poner las etiquetas de cierre, ya que así evitamos posibles errores y obtenemos un código limpio y claro (1).

La etiqueta <P> admite una serie de atributos que la complementan, entre los cuales se destacan:

<P ALIGN="left / center / right / justify">...</P>, que nos permite alinear el texto del párrafo a la izquierda (por defecto), centrado, a la derecha o justificado, en cuyo caso las líneas de texto ocuparan siempre la misma anchura total, encargándose el navegador de crear espaciados entre las palabras de cada línea para conseguir este efecto.

El siguiente ejemplo muestra el uso de la etiqueta <p>:

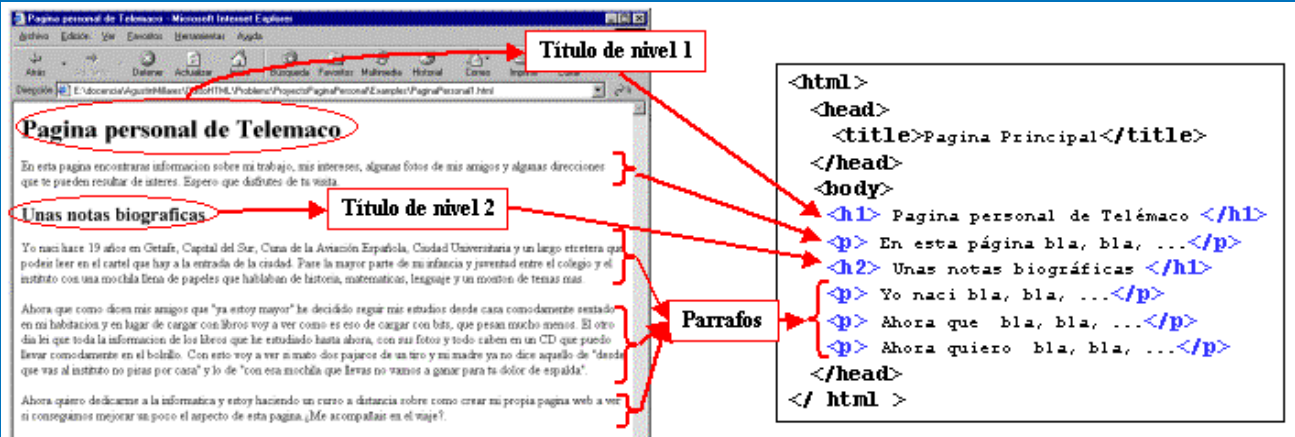
```
<!DOCTYPE HTML>
<html>
<head>
<title>Ejercicios prácticos HTML5</title>
<meta charset="utf-8">
<meta name="author" content="Alumno">
</head>

<body>
<section>
<p>Lorem Ipsum es simplemente el texto de relleno de las imprentas y archivos de texto. Lorem Ipsum ha sido el texto de relleno estándar de las industrias desde el año 1500, cuando un impresor desconocido usó una galería de textos y los mezcló de tal manera que logró hacer un libro de textos espécimen.</p>
</section>
</body>
</html>
```

```
<html>
<head>
<title>Hola, mundo</title>
</head>
<body bgcolor="#ff0000">
<p>Hola mundo.</p>
<p>Las cosas van bien.</p>
<p>A veces llueve.</p>
</body>
</html>
```

Consejo

No debemos utilizar las etiquetas de encabezado para formatear el texto, es decir, si queremos colocar un tipo de letra más grande y en negrita debemos utilizar las etiquetas que existen para ello (que veremos enseguida). Los encabezados son para colocar titulares en páginas web y es el navegador el responsable de formatear el texto de manera que parezca un titular. Cada navegador, pues, puede formatear el texto a su gusto con tal de que parezca un titular.



Por ejemplo:

`<P ALIGN="left">Párrafo alineado a la izquierda</P>`

`<P ALIGN="center">Párrafo centrado</P>`

`<P ALIGN="right">Párrafo alineado a la derecha</P>`

Esto daría como resultado:

Párrafo alineado a la izquierda

Párrafo centrado

Párrafo alineado a la derecha

Cabeceras

En un documento HTML es posible definir seis tipos distintos de cabeceras que serán normalmente el título del documento y los distintos sub- apartados que lo forman.

Las etiquetas que definen las cabeceras serán `<H1>`, `<H2>`, `<H3>`, `<H4>`, `<H5>`, `<H6>`. El texto indicado entre las etiquetas de inicio y de fin será el afectado por las cabeceras:

Para la instrucción:

`<H1>Este texto aparecerá resaltado</H1>`

El resultado será:

Este texto aparecerá resaltado

Lo normal es utilizar las cabeceras de forma consecutiva y descendente, es decir, comenzaremos el documento con la cabecera `<H1>` que definirá el título, luego para los apartados principales utilizaremos la cabecera `<H2>`, para sub- apartados `<H3>`, etc. utilizándolas así de forma consecutiva.

Sin embargo podremos utilizar cualquier cabecera en el punto que deseamos del documento para resaltar cualquier texto. El modo en que se presentan las cabeceras puede variar de un navegador a otro, y sólo se puede asegurar que se mostrará de distinta forma, en orden de importancia. Lo normal es que la `<H1>` sea negrita y muy grande, `<H2>` negrita y grande, `<H3>` itálica y grande (1).

Saltos de Línea

El salto de línea está definido con la etiqueta `
`. Es utilizado cuando queremos terminar una línea sin necesidad de terminar con el párrafo. La etiqueta `
` obliga a saltar de línea donde quiera que la ubiquemos. (1)

Comentarios en HTML

Como ya mencionamos anteriormente, en el ámbito de HTML se llama *comentario* a las notas que el autor o autores ponen en el código para facilitar su entendimiento.

Estos comentarios no son mostrados por el navegador y por tanto sólo serán visibles al leer el código HTML de la página web.

En general es recomendable ir insertando comentarios al crear una página para marcar determinadas partes y así encontrarlas más fácilmente.

Algunos usos que suelen darse a los comentarios son:

| Notas para recordar detalles del código

la próxima vez que vayamos a cambiarlo, como por ejemplo para indicar por qué hemos usado una etiqueta y no otra, por qué

hemos usado una lista numerada y no una no numerada, etc. Este tipo de comentarios son muy usados cuando las páginas son complejas.

| **Apuntar qué queda por hacer** en una determinada sección, indicar cómo es conveniente cambiarla o bien para señalar el comienzo o fin de una determinada sección de la página. También sirven para identificar fácilmente partes importantes del código o aquellas partes que cambian más a menudo.

| Usos particulares de cada Desarrollador.

De hecho los comentarios pueden usarse para cualquier cosa y cada programador de páginas web tiene su propio modo de usarlos.

Para crear un comentario no se usa una etiqueta, aunque es una estructura parecida. En primer lugar ponemos una cadena que indica el comienzo del comentario: `<!--`, esto es, el símbolo **menor que**, seguido del símbolo **fin de exclamación** y de **dos guiones**, todo ello sin espacios entre ellos. Todo el texto que le siga será parte del comentario que terminará cuando insertemos la cadena de fin: `-->`, dos guiones y el símbolo **mayor que**. (2)

```
<HTML>
<HEAD>
<TITLE>Ejemplo 2</TITLE>
</HEAD>
<BODY>
  <H1>Mi primera página</H1>
```

```
<P>Hola mundo, esta es mi página, que
tiene párrafos y unos cuantos
saltos de línea también.</P>
```

```
1 salto de línea<br>
2 saltos de línea<br>
3 saltos de línea<br>
</BODY>
</HTML>
```

La estructura de un comentario es por tanto:

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

Suele ser recomendable dejar un espacio entre ambas cadenas y el texto anterior y posterior, tal y como acabamos de mostrar.

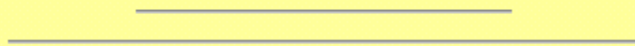
Veamos algunos ejemplos prácticos de comentarios:

```
<!--Aquí comienza el cuerpo de la página-->
<!--Cambiar este texto para que se entienda mejor-->
<!--Debería añadir más enlaces en esta página-->
```


1

```
<HR WIDTH="200"></HR>
<HR WIDTH="50%"></HR>
```

que nos dan:



3

```
<HR WIDTH="300" COLOR="red"></HR>
<HR WIDTH="300" ALIGN="LEFT" COLOR="#003366"></HR>
```

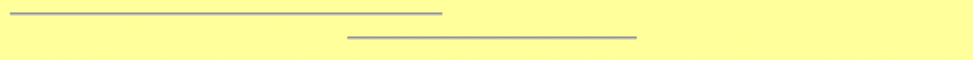
que nos dan:



2

```
<HR ALIGN="left" WIDTH="300"></HR>
<HR ALIGN="center" WIDTH="30%"></HR>
<HR ALIGN="right" WIDTH="150"></HR>
```

que nos da:



4

```
<HR WIDTH="300" SIZE="1" COLOR="red"></HR>
<HR WIDTH="300" SIZE="6" COLOR="red"></HR>
```

que nos dan:



Líneas horizontales (Separadores)

Otra de las formas de la que disponemos en **HTML** para crear separaciones entre textos (o bloques del documento en general) es la utilización de líneas horizontales, lo que se consigue con el uso de la pareja de etiquetas **<HR>...</HR>** (del inglés Horizontal Rules) que origina en el documento una línea horizontal embutida en el fondo del mismo. La etiqueta **HR** no precisa etiqueta de cierre, pero como ya hemos explicado previamente, es conveniente ponerla siempre.

La etiqueta **HR** admite los siguientes atributos:

| **<HR WIDTH="anchura"></HR>**, donde el valor de la anchura puede venir expresado mediante un número entero, que representa una longitud en píxeles (valor absoluto) o mediante un tanto por ciento, por ejemplo 40%, que representa la porción que va a ocupar la línea del ancho total de la pantalla del navegador (tamaño relativo). El valor por defecto es el 100%, es decir, si no especificamos ancho para la línea horizontal, esta ocupará todo el ancho de la pantalla. Un ejemplo lo podemos observar en la imagen (1).

| **<HR ALIGN=" left / center /right"></HR>**, que en el caso de que el ancho no sea el de toda la pantalla, alineará la línea horizontal a

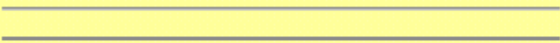
la izquierda, la centrará en pantalla (valor por defecto) o la alineará a la derecha de esta (2).

| **<HR COLOR="color"></HR>**, que nos va a permitir definir el color de la línea horizontal, y también definir el color por su valor hexadecimal o por su nombre web estándar. El valor por defecto es el que tengamos configurado en nuestra PC. Como ejemplo, mirá la imagen (3).

| **<HR SIZE="número"></HR>**. Nos permite definir la altura o espesor de la línea horizontal. Su valor por defecto suele equivaler a un tamaño SIZE="2". Observá el ejemplo (4).

```
<HR WIDTH="300"></HR>
<HR WIDTH="300" NOSHADE></HR>
```

que nos dan:



Atributo	Función
noshade	No muestra sombra, evitando el efecto en tres dimensiones.
width= num / %	Indica el ancho de la línea en tanto por ciento en función del ancho de la ventana del visor. También se puede especificar un número que indicaría el ancho de la línea en pixeles.
align= "posición"	Alinea la línea a la izquierda (left), a la derecha (right) o la centra (center).
size = numero"	Indica el grosor de la línea en pixeles

| **<HR NOSHADE></HR>**. Por defecto la línea se nos va a presentar con un efecto de sombra tridimensional. Mediante este atributo podemos evitar este efecto, presentándose entonces la línea con forma de dibujo plano, sin sombras ni efectos tridimensionales. Generalmente, si asignamos un color determinado a la línea horizontal se elimina este efecto automáticamente (1).

| **<HR TITLE="texto"></HR>**. Asigna un texto identificativo a la línea, que aparece en forma de "tooltips" al situar el cursor sobre ella.

| **<HR STYLE="estilo" / CLASS="clase" / ID="identificador"></HR>**, para asignar estilos propios a la línea horizontal. Este tema lo abordaremos cuando lleguemos a "Hojas de Estilo en Cascada aplicadas al formateo de texto" en capítulos posteriores.

Para un resumen, mirá el cuadro (2).



Tomá nota

Los atributos tienen sus valores indicados entre comillas ("), pero si no los indicamos entre comillas también funcionará en la mayoría de los casos. Sin embargo, es aconsejable que las pongamos siempre para acostumbrarnos a utilizarlas, para dar homogeneidad a nuestros códigos y para evitar errores futuros en sistemas más quisquillosos.

Sec. Escape	Símbolo
<	Signo < (menor que)
>	Signo > (mayor que)
&	Signo & (ampersand)
"	Se mostrará el signo de comillas "

¿Cómo se deben incluir los caracteres acentuados y algunos caracteres especiales? Simplemente utilizando una serie de secuencias de escape que, al visualizarse el documento, se sustituyen por el carácter deseado.

Caracteres especiales

Los caracteres acentuados y algunos caracteres especiales que usa el lenguaje HTML para definir sus etiquetas no se pueden incluir en un documento de manera normal. Para esto se deben utilizar una serie de secuencias de escape que al mostrar el documento se sustituyen por el carácter deseado.

Estas secuencias de escape comienzan todas con el símbolo **ampersand (&)**, seguido de un texto (siempre en minúsculas) que define el carácter deseado y termina con el símbolo

punto y coma (;). El error más común cuando se usan estas secuencias de escape es no utilizar el punto y coma final, en cuyo caso se mostrará el literal que define la secuencia, en lugar del carácter deseado. No es necesario dejar espacios en blanco ni antes ni después de los caracteres especiales para que queden perfectamente encuadrados en la palabra.

Si incluimos en el texto de un documento HTML el símbolo **menor que (<)** o **mayor que (>)** se interpretará siempre como la definición de una etiqueta y por tanto no se mostrarán al interpretar el documento.

Para expresar estos símbolos y otros del lenguaje HTML usaremos las secuencias de escape del cuadro (1).

Caracteres acentuados

Existen una serie de etiquetas que nos permiten mostrar los caracteres acentuados y caracteres latinos (ñ). Estos caracteres, si se incluyen en un documento HTML sin utilizar las secuencias de escape, se mostrarán correctamente. Vos con tu ordenador, y este con su navegador, podrán leer estos caracteres correctamente siempre y cuando

tengan el mismo código (español) que el autor del documento.

Los primeros 127 caracteres del código ASCII son comunes para todos los países e incluyen todas las letras del alfabeto. A partir del 128 son específicos para cada lenguaje y entre estos se incluyen los caracteres acentuados y la letra ñ. Por tanto si alguien desde otro país que tenga un código incompatible intenta leer su documento, probablemente encuentre caracteres extraños que no sepa interpretar y por tanto no será capaz de leer los caracteres acentuados.

Existen diversas secuencias que definen los distintos tipos de acentos: agudo, grave o circunflejo. Para el acento agudo usaremos el literal **acute**, tanto para las mayúsculas como para las minúsculas. Por tanto incluiremos el símbolo de ampersand (&), la vocal que deseamos acentuar, la palabra *acute* y el símbolo punto y coma (;). Representándose los acentos tal como lo vemos en la representación (1).

Para la letra ñ usaremos la secuencia tilde, del modo que indica el cuadro (2) y para el acento grave usaremos grave, logrando como resultado lo que se ve en el cuadro (3).

<i>Sec. Escape</i>	<i>Letra</i>	<i>Sec. Escape</i>	<i>Letra</i>
´	á	Á	Á
é	é	É	É
í	í	Í	Í
ó	ó	Ó	Ó
ú	ú	Ú	Ú

<i>Sec. Escape</i>	<i>Letra</i>	<i>Sec. Escape</i>	<i>Letra</i>
ñ	ñ	Ñ	Ñ

<i>Sec. Escape</i>	<i>Letra</i>	<i>Sec. Escape</i>	<i>Letra</i>
à	à	À	À
è	è	È	È
ì	ì	Ì	Ì
ò	ò	Ò	Ò
ù	ù	Ù	Ù

Otros Símbolos

Pero eso no es todo, existen aún más símbolos que son importantes de aprender. A saber: para expresar una carácter por su valor en el código ASCII, usaremos el símbolo #, seguido de su equivalente numérico.

Además, para el acento circunflejo utilizaremos el literal *circ* y para la diéresis utilizaremos el literal *uml*.

Y también, para expresar los símbolos de apertura de interrogación, apertura de exclamación y éstos acentos usaremos (1)

Algunos otros símbolos especiales los podés ver en el cuadro (2).

Sec. Escape	Letra	Sec. Escape	Letra
¿	¿	¡	¡
ü	ü	Ü	Ü
î	î	Î	Î

Sec. Escape	Símbolo	Sec. Escape	Símbolo
ç	ç	Ç	Ç
®	® Símbolo de registrado	©	© Símbolo de Copyright.
&#nnn;	Donde nnn es un número decimal, el carácter nnn del código ISO-8859-1 (ASCII).		

Y ahora veamos algunas etiquetas...

Etiqueta `<samp>`

La etiqueta `<samp>`, permite definir texto en el documento resaltado semánticamente, indicando que el texto marcado es la salida que proporciona un programa de ordenador.

La sintaxis de la etiqueta es la siguiente:

`<p>Texto <samp>marcado</samp> como salida de datos de un pro-grama</p>`

El ejemplo de la derecha muestra el uso de la etiqueta (1).

Etiqueta `<kbd>`

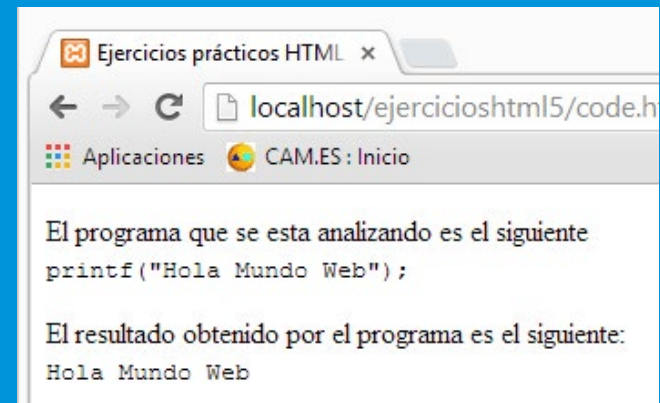
La etiqueta `<kbd>`, permite especificar un formato de texto para distinguir lo que el usuario tendría que teclear en la solicitud de un programa.

La sintaxis de la etiqueta es la del punto (2).

`<p>Texto <kbd>marcado</kbd> con etiqueta keyboard</p>`

```
<!DOCTYPE HTML>
<html>
<head>
<title>Ejercicios prácticos HTML5</title>
<meta charset="utf-8">
</head>
<body>
<section>
<p>El programa que se está analizando es el siguiente
<code>printf("Hola Mundo Web");</code></p>
<p>El resultado obtenido por el programa es el siguiente:
<samp>Hola Mundo Web</samp></p>
</section>
</body>
</html>
```

El resultado en el navegador es este:



```
<!DOCTYPE HTML>
<html>
<head>
<title>Ejercicios prácticos HTML5</title>
<meta charset="utf-8">
</head>
<body>
<p>El programa que se está analizando es el siguiente
<code>printf("Hola Mundo Web");</code>
<samp>Teclee su dirección web: </samp></p>
<p>Ha de teclear una dirección valida de in-ternet:
<kbd>www.aclases.com.ar</kbd> a continuación pulsar Intro</p>
</body>
</html>
```

Etiqueta Mark

La etiqueta `<mark>`, permite realizar el marcado de un texto dentro del documento web, el efecto visual en el navegador del texto seleccionado, es como si se hubiera pasado un bolígrafo resaltador de texto por encima del texto que se haya seleccionado.

Hay que tener en cuenta que las versiones anteriores a la 9 del navegador Internet Explorer no soportan la etiqueta.

La sintaxis de la etiqueta es la del cuadro (1)

Etiqueta Hgroup

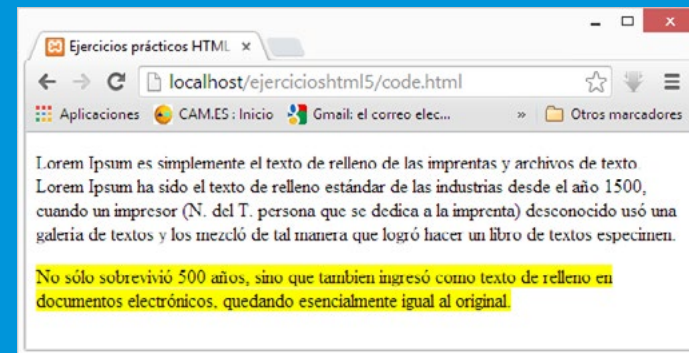
La función del elemento "hgroup" es agrupar un conjunto de encabezados en un sólo bloque para que puedan ser manejados como una unidad. Cuando mencionamos la palabra encabezado nos estamos refiriendo nada más y nada menos que a las etiquetas que van de `h1` a `h6`, y que nos sirven para darle un formato a los títulos de nuestros artículos, entradas o nombres, lo que hace resaltarlos del resto del texto. Si tuviéramos un título y un subtítulo que quisiéramos

`<p>El <mark>texto siguiente esta
remarcado</mark>, dentro del párrafo</p>`

El siguiente ejemplo muestra el uso de la etiqueta.

```
<!DOCTYPE HTML>
<html>
<head>
<title>Ejercicios prácticos HTML5</title>
<meta charset="utf-8">
</head>
<body>
<p>Lorem Ipsum es simplemente el texto  
de relleno de las imprentas y archivos  
de texto. Lorem Ipsum ha sido el texto  
de relleno estándar de las industrias  
desde el año 1500, cuando un impresor  
desconocido usó una galería de textos y  
los mezcló de tal manera que logró hacer  
un libro de textos espécimen. </p>
<p><mark>No sólo sobrevivió 500 años,  
sino que también ingresó como texto  
de relleno en documentos electrónicos,  
quedando esencialmente igual al  
original.</mark></p>
</body>
</html>
```

El resultado en el navegador del ejercicio anterior ha de ser similar al que podés ver en la siguiente imagen:



relacionar y agrupar, el tag adecuado para ello sería "hgroup". Una aplicación común en un sitio web es agregar un "eslogan" al título de nuestra página o quizás la fecha al título de una entrada, para ello comúnmente se maneja un nivel de encabezado para el título principal y otro nivel de encabezado inferior para el lema o slogan.

Hgroup será el encargado de agrupar los dos títulos para manejarlos como unidad en el árbol del documento, esto sin comprometer el valor semántico de cada etiqueta. Debemos aclarar que dentro de un elemento "hgroup" solo podemos manejar etiquetas de tipo encabezado, por lo que no es permitido introducir párrafos o algún otro tipo de etiqueta.

Su aplicación en el código HTML quedaría de la siguiente manera:

```
<hgroup>
<h1>La Diseñadora</h1>
<h2>Portal del Diseñador Web y
Webmaster</h2>
</hgroup>
```

Etiqueta aside (1)

El elemento *aside* es utilizado para envolver todo aquel contenido que sea relativo a la sección principal de un sitio y es a menudo representado con las famosas barras laterales en los sitios web.

Aunque si bien es cierto que la mayor parte del tiempo encontraremos a esta etiqueta envolviendo contenido que se encuentra a un lado del sitio, debemos aclarar que la etiqueta en sí no está relacionada con ninguna posición, tanto puede estar a un lado como en el centro o incluso en el tope superior o inferior de la página.

Su función, como ya la mencionamos, consiste en envolver una sección de contenido que tiene alguna clase de relación con el texto más grande donde aparece, motivo por el cual no puede valerse por sí misma como una pieza portable de contenido. Es por ello que se puede hacer uso de ella para mostrar mensajes de publicidad, alguna cita o una aclaración relacionada con la entrada o el texto donde se despliega.

Su aplicación en el código HTML quedaría de la siguiente manera:

```
<aside style="background: #DEDEDE;
color:steelblue; float:right; width:220px;
padding: 20px; border: 1px solid #CCCCCC">
<span style="font-weight:bold">¿Necesitas
una web?</span><br/> ¡Nosotros te la
diseñamos!
</aside>
<hgroup>
<h1>Diseño de formularios con CSS y
HTML</h1>
<h2>Artículo creado el 9/05/2014</h2>
</hgroup>
<p>En este artículo, veremos la variedad
de caminos con los cuales podemos hacer
formularios atractivos para nuestros sitios
y servicios web usando sencillamente CSS.
Los formularios son parte importantes,
pero poco elegantes, de muchos sitios. CSS
hace más fácil la creación de formularios
atractivos rápidamente </p>
```

0=0	8=8
1=1	9=9
2=2	A=10
3=3	B=11
4=4	C=12
5=5	D=13
6=6	E=14
7=7	F=15

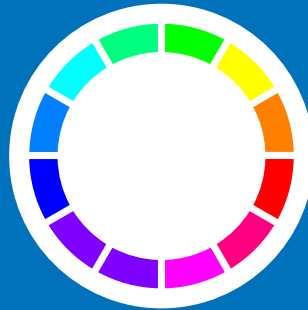


Tabla de color

¿Cómo se cambiaría
la fuente para
escribir en rojo?

`Rojo`



Los colores y HTML

En la composición de webs juegan un papel muy importante los colores, los cuales se indican en valores RGB. Pero ¿qué quiere decir esto? que para conseguir un color cualquiera mezclaremos cantidades de Rojo, Verde y Azul.

Los valores RGB se indican en numeración hexadecimal, en base 16, es decir, los dígitos pueden crecer hasta 16 y como no hay tantos

dígitos numéricos se utilizan las letras de la A a la F. (1)

Para conseguir un color, mezclaremos valores de esta manera:

RRGGBB

Donde cada valor puede crecer desde 00 hasta FF.

Entonces, al Atributo color le damos un valor RGB en formato hexadecimal. El caracter # se coloca al principio de la cadena.

Otros colores podrían ser:

Naranja	#FF8000
Verde turquesa	#339966
Azul oscuro	#000080

Utilizaremos siempre
estos valores:

00
33
66
99
CC
FF

¡Importante! Los colores son
compatibles en todos los sistemas.



Como a las páginas web las tienen que ver todos los usuarios, y los sistemas que ellos utilizan para entrar son distintos, hay que utilizar colores compatibles con la paleta de todos ellos.

La forma de conseguir esto es limitando nuestros colores a los que se pueden conseguir, tal como lo muestra el cuadro (1)

Ejemplos de ellos son:

#3366FF #FF9900 #666666

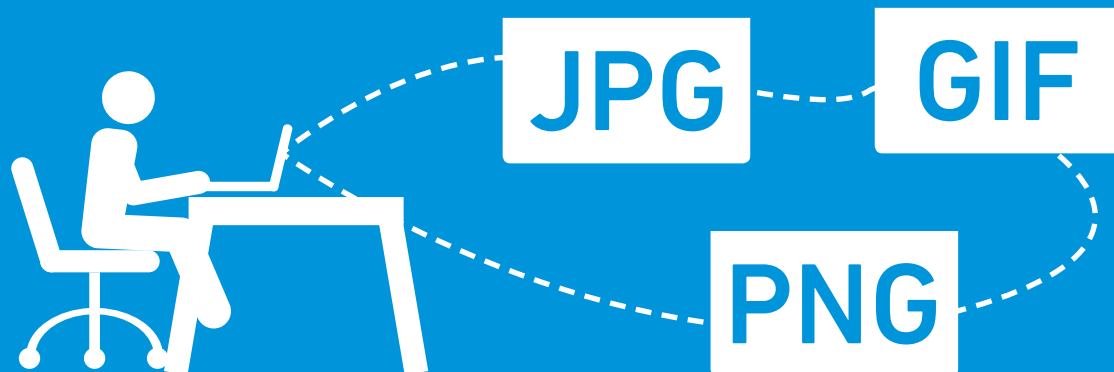
Las Imágenes y HTML

Un documento **HTML** puede incorporar imágenes o gráficos, no sólo como fondo del documento (con el atributo **BGCOLOR** de la etiqueta **<BODY>**) sino dentro del propio cuerpo del documento. Existen distintos tipos de imágenes:

Jpg: (con extensión JPG) son imágenes en color real (paleta de colores RGB de 24 bits) comprimidas mediante un algoritmo

matemático. Se emplean cuando la imagen precisa una gran cantidad de colores (fotografías principalmente), prevaleciendo la calidad de la imagen y su colorido sobre el tamaño del archivo.

Gif: En el formato GIF, las imágenes sólo se representan mediante una paleta de 256 colores, por lo que no permiten la presentación de imágenes muy complejas. Son las más empleadas por su pequeño tamaño. Puede escogerse uno de los 256



¿Cuáles son los distintos tipos de imágenes que existen? Principalmente, JPG, GIF, GIF Animado y PNG. La elección de cuál usar dependerá de la necesidad que tengas.

colores para hacerlo transparente, con lo que el navegador no representa dicho color en el documento, viéndose su fondo.

| Gif animado: Consiste en una serie de imágenes en formato GIF (y por tanto en 256 colores) que se alternan en la pantalla del ordenador, obteniéndose una película. El *GIF animado* puede repetirse un número determinado de veces o indefinidamente, de forma que la película se repita ininterrumpidamente, volviendo al principio cuando se ve la última imagen.

| PNG: (*Portable Network Graphics*) es un formato gráfico basado en un algoritmo de compresión sin pérdida para bitmaps. Este

formato fue desarrollado en buena parte para solventar las deficiencias del formato GIF y permite almacenar imágenes con una mayor profundidad de color.

¿Cómo insertar imágenes?

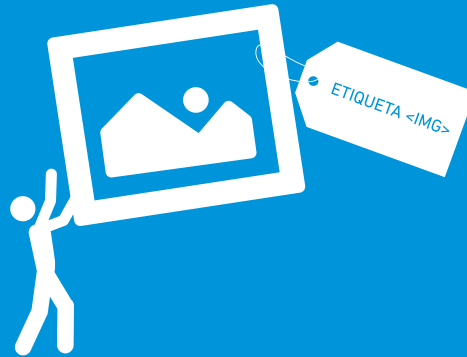
La manera de insertar imágenes en un documento es mediante la etiqueta . Las propiedades de esta etiqueta indican opciones relacionadas con la imagen, como ubicación, texto alternativo, etc. Veremos a continuación todas ellas:

| : Para insertar una imagen en un documento HTML se emplea la etiqueta , que carece de etiqueta de cierre. Por el

contrario presenta varios atributos, uno de los cuales, el atributo SRC, es imprescindible para su funcionamiento.

| SRC: El atributo SRC es indispensable dentro de la etiqueta , ya que especifica la URL (absoluta o relativa) de la imagen que se desea insertar en el documento HTML. Así, para insertar una imagen llamada foto.jpg que se encuentre en el directorio imágenes, respecto a la ubicación del documento, escribiremos:

Si por el contrario la imagen se halla en el directorio imágenes del sitio www.dominio.ar, deberemos indicar:



Las imágenes se insertan en un documento mediante la etiqueta ``. Las propiedades de esta etiqueta indican opciones relacionadas con la imagen, como la ubicación por ejemplo.

``

| Width y Height: se especifican, en píxeles, la anchura y altura de la imagen que se ha de insertar. Si se especifican mayores que la propia imagen, ésta se deformará hasta ocupar el tamaño indicado. Si por el contrario se especifican menores, la imagen se reducirá. No es habitual indicar un tamaño mayor (al aumentar se muestran defectos). La reducción no significa que el tamaño de la imagen (y la memoria que ocupa) disminuya, sólo se muestra en el tamaño deseado. Para ahorrar espacio de disco y tiempo de descarga, si necesitamos reducir una imagen, en lugar de emplear los atributos `WIDTH` y `HEIGHT` es mejor editar la imagen con un

programa de tratamiento de imágenes y dejarlas en el tamaño que vaya a ser utilizado en el sitio web. A continuación puedes ver la imagen JPEG anterior, estableciendo mediante estos atributos un tamaño determinado.

| Align: Este atributo indica como se ha de distribuir el texto que acompaña a la imagen. Los valores más habituales son:

| Left: La imagen se sitúa a la izquierda de la página y el texto se distribuye a su derecha.

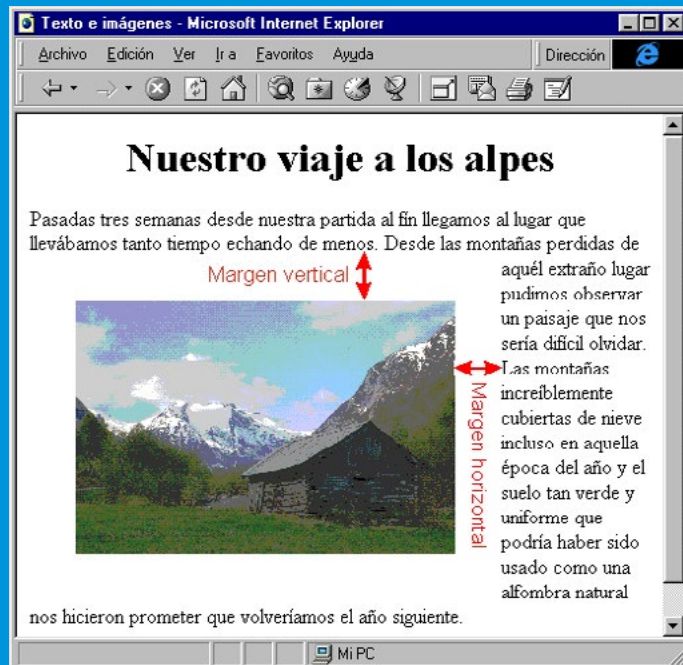
| Right: La imagen se coloca a la derecha de la página y el texto se distribuye a su izquierda.

| Bottom: La imagen se alinea, donde ha sido insertada, coincidiendo su línea inferior con la parte inferior de la línea de texto.

| Top: La parte superior de la imagen se alinea con la parte superior de la línea de texto.

| Middle: El centro de la imagen se alinea con la línea de texto.

Mientras que los valores **left, right y center** posicionan la imagen respecto a los bordes del navegador, los valores **bottom, top y middle** dejan la imagen en el lugar en el que se ha insertado la etiqueta ``.



| Hspace y Vspace: Los atributos *HSPACE* (de *Horizontal Space*) y *VSPACE* (de *Vertical Space*) indican, en píxeles, el espacio libre (horizontal y verticalmente) alrededor de la imagen insertada. De esta forma, cuando la imagen se alinea a la izquierda o la derecha, las líneas de texto no terminarán o empezarán junto a la imagen (1).

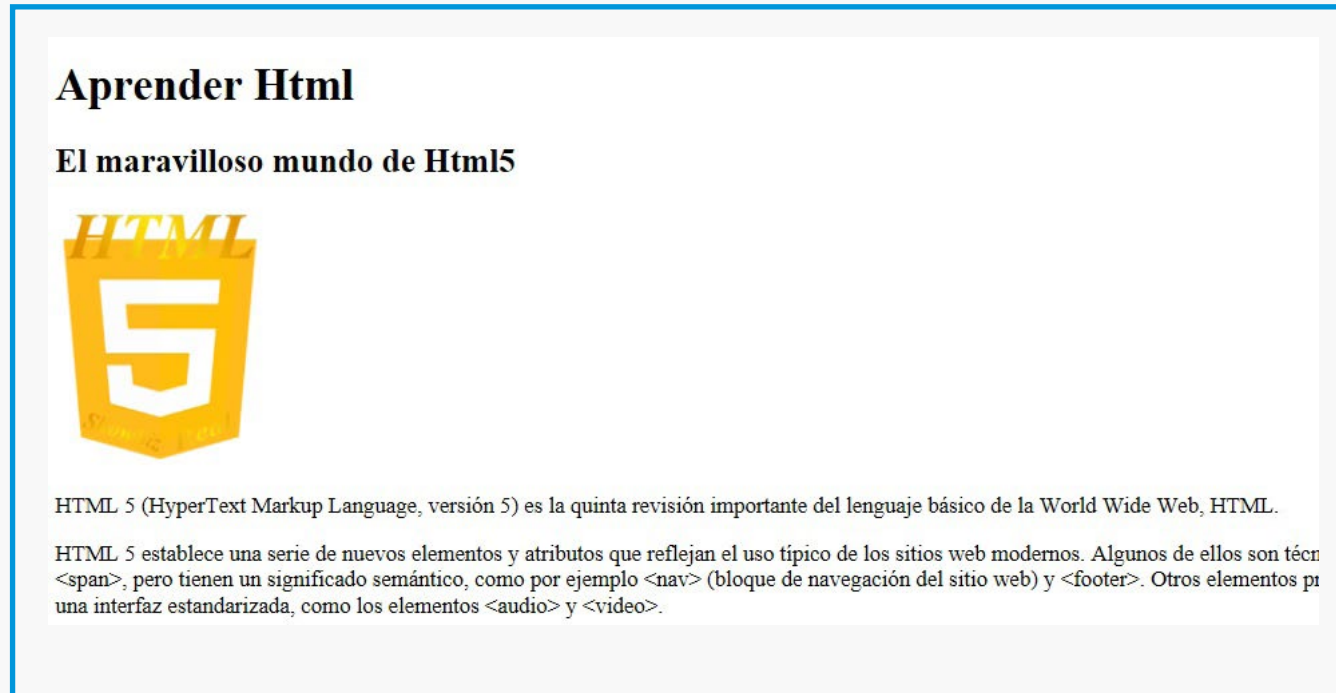
| ALT: El atributo *ALT* contiene un texto alternativo que se empleará en el caso de que el navegador no pueda representar el gráfico.

El texto alternativo también se mostrará cuando situemos el puntero del mouse sobre el gráfico.

También podés agregarle bordes a una imagen con la propiedad *BORDER* como te mostramos en la figura (2).

Veamos un ejemplo:

La siguiente imagen es el resultado del código de la derecha (1).



Por otra parte, si queremos establecer una imagen como hipervínculo deberemos escribir el siguiente código:

<!DOCTYPE HTML>

<html>

<head>

<title>Mi primera pagina web</title>

</head>

<body>

<h1>Aprender Html</h1>

<h2>El maravilloso mundo de Html5</h2>

<p>HTML 5 (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. </p>

<p>HTML 5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Algunos de ellos son técnicamente similares a las etiquetas <div> y ;, pero tienen un significado semántico, como por ejemplo <nav> (bloque de navegación del sitio web) y <footer>. Otros elementos proporcionan nuevas funcionalidades a través de una interfaz estandarizada, como los elementos <audio> y <video>.</p>

</body>

</html>

Atributo	Función
alt= "Texto"	Mostrará el texto indicado en el caso de que el navegador utilizado para ver la página no sea capaz de visualizar la imagen.
align= TOP / MIDDLE /BOTTOM / RIGHT / LEFT	Indica cómo se alineará el texto que siga a la imagen. <i>TOP</i> alinea el texto con la parte superior de la imagen, <i>MIDDLE</i> con la parte central, <i>BOTTOM</i> con la parte inferior <i>LEFT</i> dejará la imagen a la izquierda del texto y <i>RIGHT</i> , a la derecha del texto.
border= número	Indica el tamaño del "borde" de la imagen. A toda imagen se le asigna un borde que será visible sólo cuando la imagen forme parte de un enlace.
height= número	Indica el alto de la imagen en puntos o en porcentaje. Se usa para variar el tamaño de la imagen original
width= número	Indica el ancho de la imagen en puntos o en porcentaje. Se usa para variar el tamaño de la imagen original.
hspace= número	Indica el número de espacios horizontales, en puntos (píxeles), que separarán la imagen del texto que la siga y la anteceda.
vspace= número	Indica el número de puntos (píxeles) verticales que separarán la imagen del texto que le siga y la anteceda.

Desempeños

Ahora te proponemos que **codifiques los siguientes ejemplos**. Son varios, lo que te permitirá adquirir habilidad en maquetado **¡Suerte y mucha inspiración!**



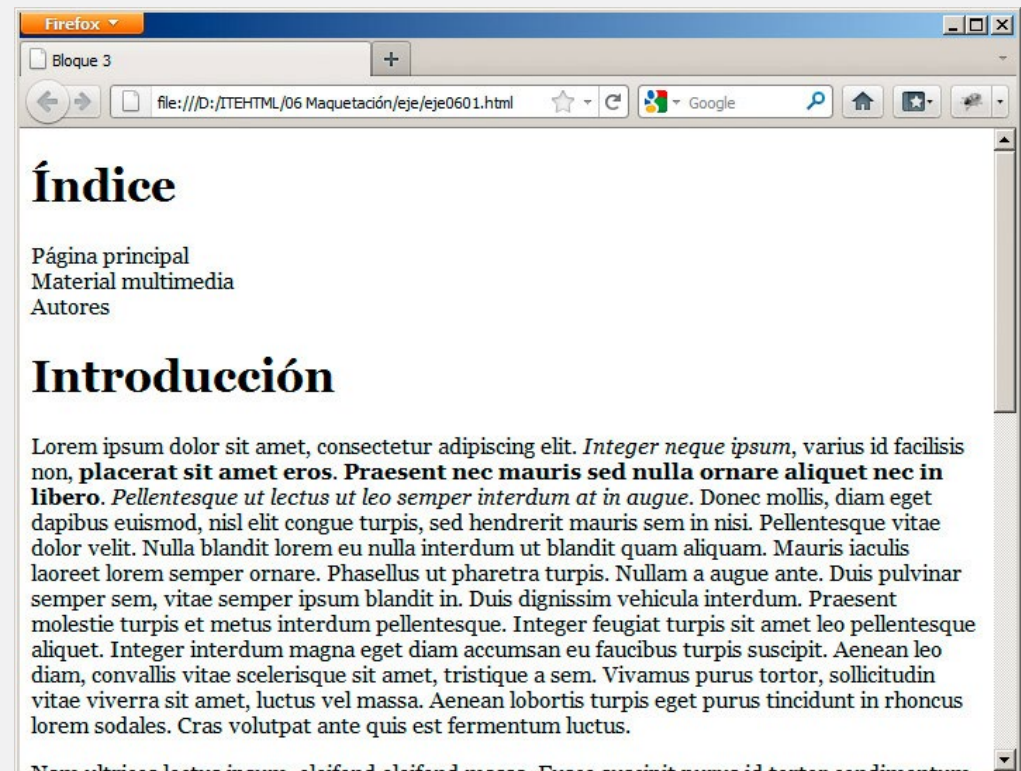
Nota

No necesariamente los ejercicios deben quedar tan cual aparecen en la clase, podés utilizar tu imaginación y creatividad para darle un toque extra. La idea es que puedas utilizarlos como una guía. ¡Se vos mismo y dale tu toque personal!

Podés agregar cualquier imagen o retocarla en un editor de tu elección, también agregar fondos en las páginas. Todo queda a tu orden, siempre y cuando se parezca a lo que te proponemos.

Desempeño 1

Desempeño 2



Desempeño 3

Desempeño 4



En los ejercicios que siguen deberás codificar el texto y también imágenes .



→ Desempeño 5

→ Desempeño 6

Simple

Excluye lo que no es importante



Rafael Nadal - Feliciano López

Sigue en directo la eliminatoria de cuartos de final del Masters de Madrid a partir de las 20.00 horas

• Federer arrolla a Del Potro • **ESPECIAL**

 Desempeño 7

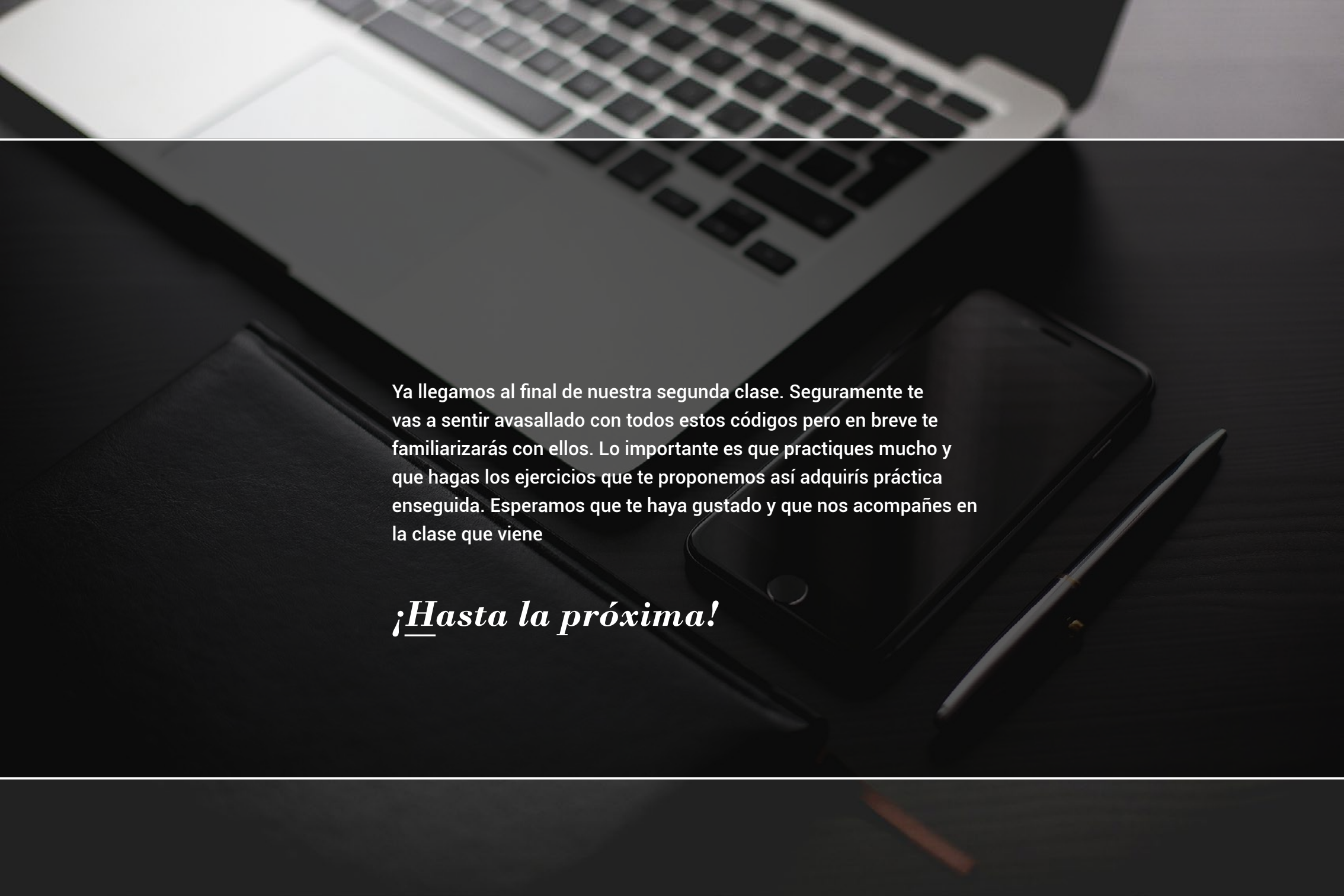
 Desempeño 8





Autoevaluación

- 1 | Investiga qué etiquetas quedaron en desuso en la versión HTML5.
- 2 | ¿Cómo puedo distanciar dos imágenes desde el lado superior y desde el lado izquierdo?
- 3 | ¿Es mejor usar imágenes en formato GIF o JPG para ser insertada en una página web? ¿Por qué?
- 4 | ¿Quién determina las versiones de HTML?
- 5 | ¿Cuál es la función de un maquettador web?
- 6 | ¿Cómo se estructuraba Html antes de la versión 5?



Ya llegamos al final de nuestra segunda clase. Seguramente te vas a sentir avasallado con todos estos códigos pero en breve te familiarizarás con ellos. Lo importante es que practiques mucho y que hagas los ejercicios que te proponemos así adquirirás práctica enseguida. Esperamos que te haya gustado y que nos acompañes en la clase que viene

¡Hasta la próxima!