

			Clase 7
--	--	--	-------------------

Tópico

HTML

Metas de comprensión

- ✓ Los alumnos comprenderán cuales son los requerimientos o conocimientos previos con los que debe contar para comenzar a programar con HTML
- ✓ Los alumnos comprenderán en forma detallada la sintaxis del lenguaje HTML5 y sus etiquetas semánticas
- ✓ Los alumnos comprenderán las tags relacionadas con párrafos, formatos en general, como así también sus atributos y propiedades
- ✓ Los alumnos comprenderán y aprenderán a trabajar con imágenes y formatos en el ámbito del lenguaje HTML5

Desempeño de Exploración

Debate sobre los conceptos expuestos en clase. Ejemplificación y relación con otras materias. Comienzan a perfilarse todas las metas de comprensión y tópicos de la materia.

Desde la intuición y la propia experiencia de usuario de internet, conversamos y debatimos en clase los nuevos conceptos adquiridos por otras materias y la experiencia que trae el alumno con respecto a la programación web. También se hace hincapié en las expectativas laborales y profesionales que poseen los alumnos y lo que los impulsa o motiva a cursar esta materia.

Desempeño de investigación

Se realizan investigaciones sobre los temas expuestos en clase en forma grupal para debatir en clases subsiguientes. Se arman galerías de imágenes, se trabaja con herramientas auxiliares como editores fotográficos, paleta de colores, fuentes etc.

Introducción

¡Cada clase según su contenido puede tener un numero variable de páginas, a leer, no te asustes!! La materia esta lo más actualizado posible al año 2022, de manera tal que tengas todos los temas en forma completa, de allí su extensión.

Algunas recomendaciones que te pueden ayudar a la hora de comprender el material de estudio:

- ✓ Lee varias veces la clase si fuera necesario.
- ✓ Subraya, destaca o resume los conceptos que creas principales o de importancia en cada tema.
- ✓ Puedes grabar la lectura de la clase (no necesaria puedes ser tú, puede ser un familiar, amigo etc.) para poder escucharla luego en el colectivo, recreo, relax o fin de semana etc.
- ✓ Muchas veces los números o medidas (velocidades, tamaños, resoluciones, latencia, potencia, capacidad, etc.) no son tan importantes ya que la informática evoluciona día a día y esos son bastantes cambiantes en la guerra de las empresas y fabricantes para sobresalir en el mercado, por eso no es necesario que los memorices a todos a menos que el tutor te lo indique.
- ✓ Puedes realizar grafica con cuadros,
- ✓ Puedes ampliar tu conocimiento con investigaciones adiciones en la web o viendo videos en youtube que refuercen los conceptos
- ✓ Puedes consultarle a tu tutor por cualquiera de las vías indicadas en el campus por el temario si tienes alguna duda, consulta o inquietud.
- ✓ Trata de organizar tu tiempo para la lectura y la comprensión del material
- ✓ Este texto va a acompañar a todas las clases para recordarte como leer, estudiar y comprender el material de la materia
- ✓ Cada clase en el campus podrá estar acompañada de material adicional como profundizar los temas para la mejor interpretación de los mismos con videos, encuestas, foros, actividades individuales y/o grupales.
- ✓ Busca algún compañero de estudio para poder compartir conocimiento, apuntes y metodología de estudio
- ✓ No esperes hasta último momento para realizar tus consultas, leer o ponerte al tanto con la materia

¿Te pusiste a pensar porque debemos aprender a programar en HTML? Acá van algunas de las respuestas a esta inquietud.



Hoy en día es muy importante y relevante el conocimiento de

ciertos lenguajes de programación y de las herramientas de edición de páginas web, así como aprender a trabajar y construir dicha página web para su desarrollo, su creación, su diseño y su mejora continua.

El mercado laboral de los programadores, los informáticos, los ingenieros y tantos profesionales relacionados con las Pcs, la creación de contenido o la programación y gestión de apps, páginas webs y plataformas digitales, deben estar en continua evolución y continuo aprendizaje. Quien se duerme se pierde, como dice el dicho popular.

Y es que internet y las tecnologías cambian cada año, evolucionan y se mejoran sin pausa ni descanso. HTML y CSS son unos de los lenguajes más estables y continuos, que, si bien se han mejorado y se han vuelto más efectivos y eficientes, presentando nuevas versiones y nuevas generaciones. En general todo el mercado de programación, tecnología e internet está evolucionando continuamente, lo que no tiene por qué ser un problema, sino una oportunidad para seguir aprendiendo es por eso que comenzamos una nueva etapa en nuestra materia.

Acerca del empleo

Este anuncio proviene de un tablero de empleos. [Más información](#)

Nos encontramos en la búsqueda de un Desarrollador WEB para trabajar en una reconocida empresa de Tecnología dedicada al Control de Tiempos y Accesos, reportando directamente a su Gerencia de Investigación y Desarrollo.
RESPONSABILIDADES: Participación en análisis, diseño y programación para el desarrollo de nuevos productos. Mantenimiento productos liberados. Testing - Documentación. Manejo de versiones e instaladores.
REQUISITOS: Experiencia laboral en programación WEB. Muy buen manejo de Javascript, HTML, VSS. Conocimientos de Vue.js. Valorado React. Ser estudiante avanzado o graduado de carreras afines a Sistemas. Excelente actitud y potencial para aprender.
BENEFICIOS: Trabajo remoto en relación de dependencia. Medicina Prepaga. Crecimiento Profesional. Capacitación permanente.

¡Manos a la obra!

Introducción

¡Cada clase según su contenido puede tener un numero variable de páginas, a leer, no te asustes!! La materia esta lo más actualizado posible al año 2022, de manera tal que tengas todos los temas en forma completa, de allí su extensión.

Algunas recomendaciones que te pueden ayudar a la hora de comprender el material de estudio:

- ✓ Lee varias veces la clase si fuera necesario.
- ✓ Subraya, destaca o resume los conceptos que creas principales o de importancia en cada tema.
- ✓ Puedes grabar la lectura de la clase (no necesaria puedes ser tú, puede ser un familiar, amigo etc.) para poder escucharla luego en el colectivo, recreo, relax o fin de semana etc.
- ✓ Esta parte de la materia se aprender con horas en máquina, es decir codificando los ejercicios.
- ✓ Puedes ampliar tu conocimiento con investigaciones adiciones en la web o viendo videos en youtube que refuercen los conceptos
- ✓ Puedes consultarle a tu tutor por cualquiera de las vías indicadas en el campus por el temario si tienes alguna duda, consulta o inquietud.
- ✓ Trata de organizar tu tiempo para la lectura y la comprensión del material
- ✓ Este texto va a acompañar a todas las clases para recordarte como leer, estudiar y comprender el material de la materia
- ✓ Cada clase en el campus podrá estar acompañada de material adicional como profundizar los temas para la mejor interpretación de los mismos con videos, encuestas, foros, actividades individuales y/o grupales.
- ✓ Busca algún compañero de estudio para poder compartir conocimiento, apuntes y metodología de estudio
- ✓ No esperes hasta último momento para realizar tus consultas, leer o ponerte al tanto con la materia

HTML

El **lenguaje de marcado de hipertexto (HTML)** es un lenguaje informático que forma parte de la mayoría de las páginas web y aplicaciones en línea. Un **hipertexto** es un texto que se utiliza para enlazar con otros textos, mientras que un **lenguaje de marcado** es una serie de marcas que indican a los servidores web la estructura y el estilo de un documento.

Los documentos HTML son archivos que terminan con una extensión .html o .htm. El navegador web lee el archivo HTML y muestra su contenido para que los usuarios de Internet puedan verlo.

Todas las páginas HTML cuentan con una serie de elementos HTML, formados por un conjunto de etiquetas y atributos. Los elementos HTML son los bloques de construcción de una página web. Una *etiqueta* indica al navegador dónde empieza y termina un elemento, mientras que un *atributo* describe las características de un elemento.

¿Cómo se relacionan HTML, CSS y JavaScript?

HTML se utiliza para añadir elementos de texto y crear la estructura del contenido. Sin embargo, no es suficiente para construir un sitio web profesional y totalmente responsivo. Por eso, HTML necesita la ayuda de las Hojas de Estilo en Cascada (CSS) y de JavaScript para crear la mayor parte del contenido del sitio web.

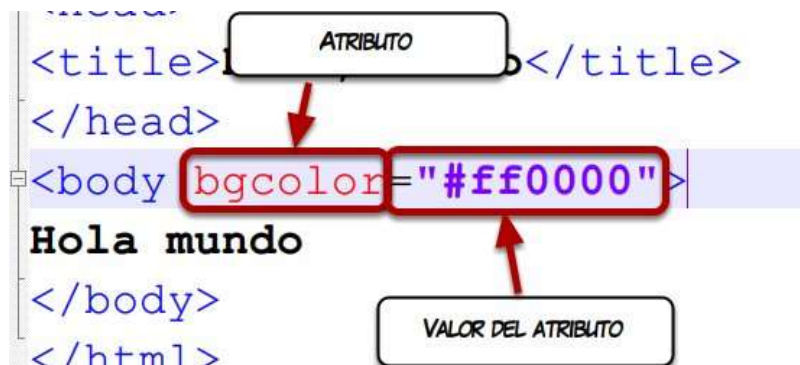
El CSS es responsable de estilos como el fondo, los colores, el diseño, el espaciado y las animaciones. Por otro lado, *JavaScript* añade funcionalidades dinámicas como sliders, pop-ups y galerías de fotos. Estos tres lenguajes son los fundamentos del desarrollo front-end.



Diseño responsivo permite a los usuarios disfrutar de todas las funcionalidades del sitio web desde cualquier tipo de dispositivo.

¿Qué son las etiquetas?

Las etiquetas HTML son indicadores del lenguaje HTML, que le dan formato y estructura al contenido de las páginas web, como lo dijimos anteriormente. Es un estándar adoptado por todos los desarrolladores que, mediante etiquetas de texto, les permite a los navegadores interpretar los elementos de una página web, no importa en dónde, cuándo o en qué idioma fue creada, para mostrar su contenido a los usuarios. Para ser más claros, te mostraremos cuáles son las partes básicas de las etiquetas de HTML:



- ✓ **Elemento:** es el nombre de la etiqueta, generalmente con una apertura y cierre. Por ejemplo:
`<title>Partes básicas de etiquetas HTML</title>`, que indica que esa frase es un título.



- ✓ **Atributo:** es lo que diferencia a un elemento y que le da ciertas características, como tamaño, color, tipo de fuente. Se escribe antes del signo de igual (=), junto al elemento de la etiqueta, únicamente en la apertura, y antes de la variable. Por ejemplo:



`. Este texto aparecerá en tipografía Arial`.

- ✓ **Contenido:** es el texto que modifica las etiquetas, lo que aparece entre las etiquetas y atributos.

¿Qué es HTML5?

Los estándares HTML se han ido actualizando, cambiando y mejorando, **HTML5**, que es en realidad la actualización que se produjo sobre el estándar HTML desde la versión HTML4.

Utiliza **las mismas reglas básicas que HTML4**, aunque incluye en sus parámetros algunas etiquetas y atributos nuevos, los cuales han sido incluidos para brindar la posibilidad de disponer de una mejor semántica y elementos dinámicos que se activan a través de JavaScript cuando desarrollamos una página web. Al mismo tiempo, en la versión 5 del estándar se incluyeron **diferentes tipos de entrada nuevos para los formularios**, gracias a lo cual ha sido posible incluir en dichos formularios mayor cantidad de información de los usuarios, tales como correo electrónico, URL, teléfono y demás.

Al mismo tiempo, **en el estándar HTML5 fueron eliminados algunos elementos de estilo que generaban distintos inconvenientes de accesibilidad a las páginas web**, y que en realidad no se utilizaban demasiado.

¿Qué son etiquetas semánticas?

HTML5 se han incorporado muchas etiquetas que sirven para indicar qué son y cuál es el significado de los elementos que contienen dentro. 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. Para ello vamos a comenzar codificando:

1. Abrir el Programa *editor de Texto* (podemos utilizar Notepad++ o cualquier editor que tengas a mano SublimeText, VisualCode, etc)
2. Escribiremos los códigos o “etiquetas”, las instrucciones, códigos o "tags" de **html** van entre dos signos < y >. La instrucción es indiferente si se escriben en mayúsculas o minúsculas. Vas a encontrar mucha bibliografía donde el código este todo en Mayúscula, esto permite diferencias las etiquetas del contenido de la página.
3. Para crear una página web sencilla tenemos que escribir en nuestro editor las siguientes líneas de código:

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

```
<p>Mi primer párrafo</p>
</body>
</html>
```

4. Guardaremos el documento asignándole un nombre con la extensión (.html) ejemplo index.html



Utiliza como nombres en tus archivos algunas normas básicas para no tener problemas a la hora de grabar el archivo. No utilices acentos ni espacios ni otros caracteres. Algún carácter como el guion "-" o el guion bajo "_" te puede ayudar a separar las palabras. Por ejemplo: quienes_somos.html.

5. Para *visualizar* el resultado en el navegador haz doble click en el archivo html que acabas de guardar recién, lo puedes buscar desde el explorador de archivos de Windows en la carpeta por defecto donde hayas configurado el guardado, o bien en un directorio donde grabes tus trabajos.

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 tags están colocadas entre los símbolos menor (<) y mayor (>). La etiqueta de inicio de cada par indica al navegador donde comienza la instrucción, y la etiqueta de final le indica donde termina. La etiqueta de cierre siempre incluye una barra (/) después del símbolo menor (<). Para que una página web se vea bien es necesario incluir por lo menos las directivas que compone estructura básica de un documento Html5:

- ✓ La primera línea de código es la declaración **<!DOCTYPE HTML>** de un documento Html. Esta etiqueta indica que versión de HTML se utiliza en el documento, para que los navegadores puedan aplicar la gramática y sintaxis correspondiente al documento. La versión de Html5 simplifica mucho esta directiva. Las versiones anteriores Html usaba declaraciones más complicadas para varios formatos Html.
- ✓ **<html>** aparece una vez al principio de la página web y de nuevo con una barra añadida (**</html>**) al final, indica al navegador que todo lo que se encuentra entre ellas es código html y se debe interpretar como tal.
- ✓ Con **<head>** y **</head>** se delimita la cabecera del documento que suele contener información sobre el documento 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>**
- ✓ **<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.
- ✓ **<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

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 anteriores ediciones, 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, 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.

- ✓ 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. Un elemento header se puede situar frecuentemente al principio de una página o sección. En él 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.
- ✓ 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
- ✓ El **elemento section** 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. 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 mí” “contactar”
 - Diferentes partes de una extensa página “términos de servicio”
 - Varias secciones de un sitio web de noticias: artículos, novedades, noticias internacionales, deportes...

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

- ✓ **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 mientras un *section* puede contener cualquier contenido que se pueda agrupar temáticamente, un *article* debe ser una sola pieza de contenido que pueda valerle 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. 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.

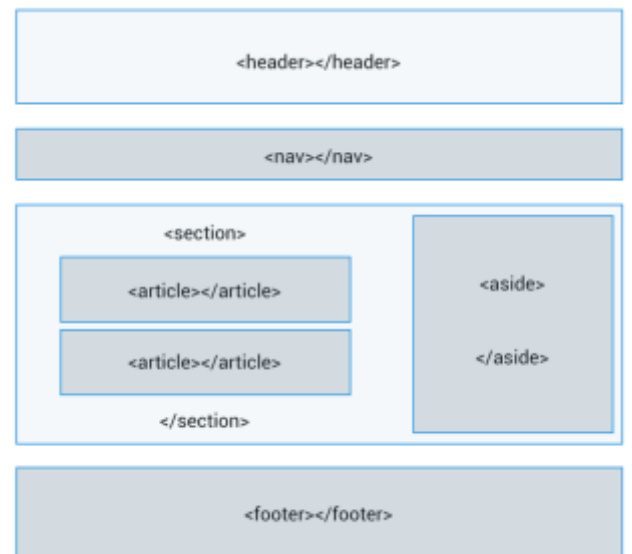
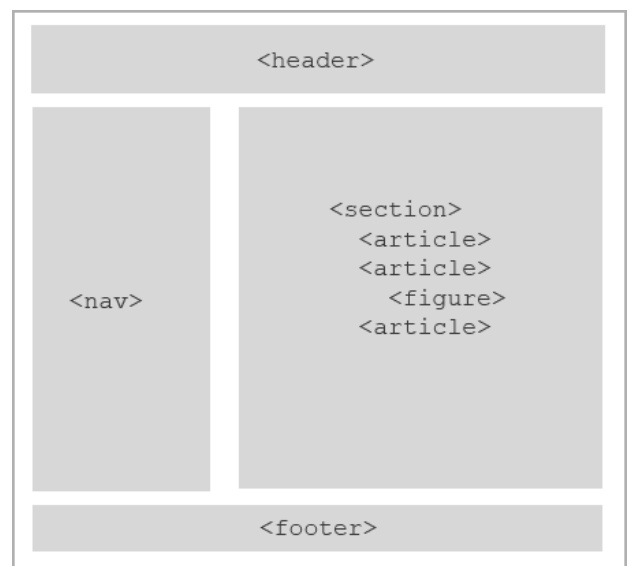
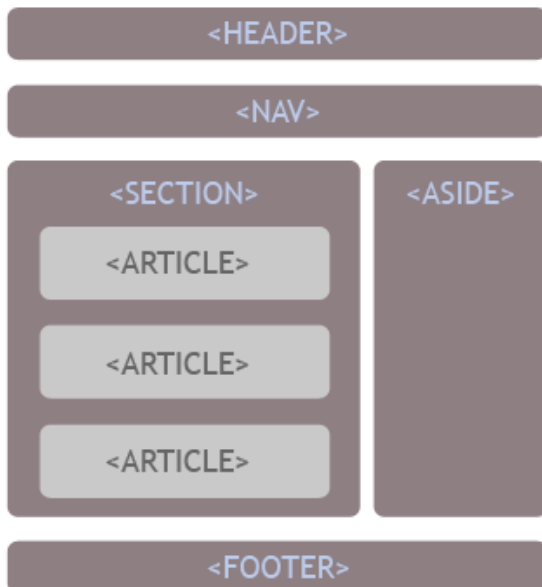
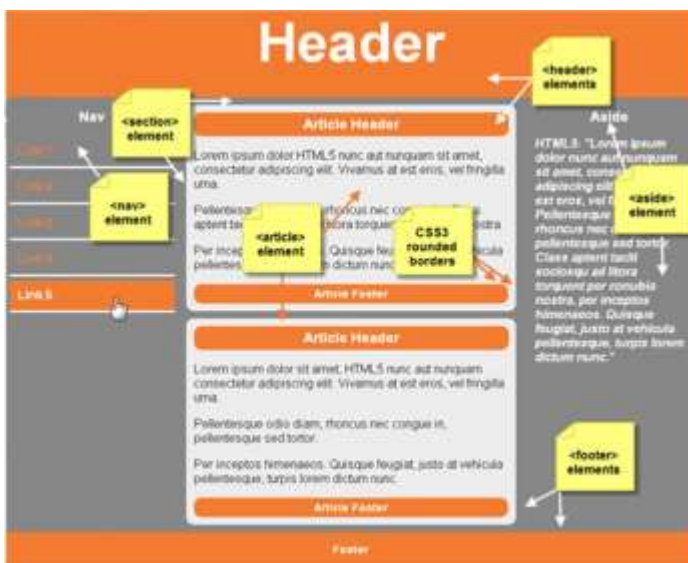
- ✓ Este **elemento aside** representa una parte de la página que está relacionada con el contenido. Se podría utilizar para englobar una parte del contenido que es tangencial a:

El elemento *aside* nunca se debería emplear para englobar secciones de la página que forman parte del contenido principal. Usos posibles:

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

- ✓ El **elemento footer** representa un pie de página para la sección de contenido. Usos más frecuentes:

- Información de copyright
- listas de vínculos relacionados
- información del autor

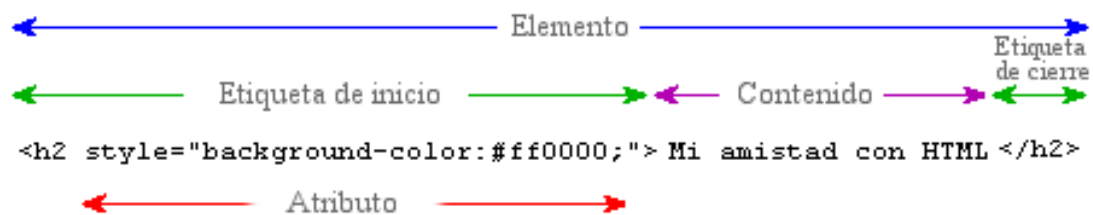


Claramente podemos adivinar que *HEADER* sirve para definir el encabezado del sitio, *NAV* contiene los enlaces para navegar y *FOOTER* la información del pie de página. Luego aparecen una serie de etiquetas para especificar dentro del área de contenido el tipo de

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. Todo texto que se encuentre entre los caracteres < y > se considerará una etiqueta, si la etiqueta no fuera una de las validas del lenguaje **HTML5** no será tenida 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 de obtendrá el efecto que deseábamos.

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.



Etiquetas Anidadas

Hasta ahora hemos visto que las etiquetas **html5** suelen ir en parejas. También es posible incluir unas etiquetas dentro de otras. A esto se llama anidamiento. Ejemplo:

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

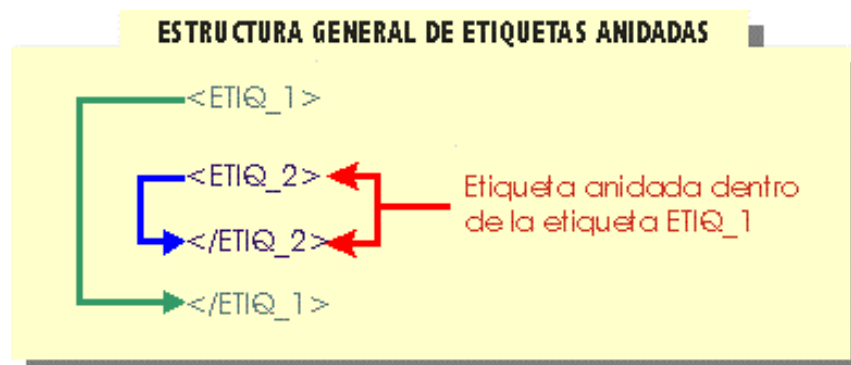
El **anidamiento** puede ser poniendo las etiquetas en cualquier orden, pero siempre se ha de guardar simetría.

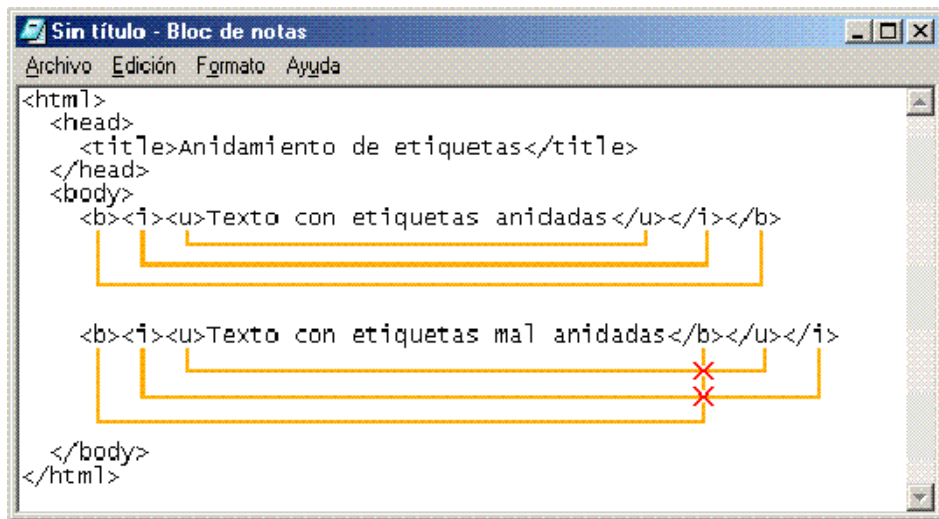
Esto es incorrecto:

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

Y esto es la forma correcta:

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





Propiedades del Body

<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">

Pondrás 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 una página web sin texto? Nada, ¿Verdad? Podemos gestionar párrafos con la etiqueta P, de manera 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, ¿cómo lo hacemos? Tenemos las siguientes opciones:

- ✓ Podemos utilizar el tag **** (bold en inglés)
- ✓ Etiqueta **strong**: lo que incluyamos entre sus etiquetas de apertura y cierre se mostrará, como forma habitual, en negrita.

- ✓ 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*).

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

```
<!DOCTYPE HTML>
<html>
<head>
<title>Ejercicios prácticos HTML5</title>
</head>

<body>
<section>
  <p>Es simplemente el texto de relleno de las imprentas y archivos de texto. Ha sido el
  texto de relleno estándar de las industrias desde el año 1500, cuando un impresor (N. del
  T. persona que se dedica a la imprenta) desconocido usó una galería de textos y los mezcló
  de tal manera que logró hacer un libro de textos</p>
</section>
</body>
</html>
```

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. 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. Ejemplos:

```
<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

```

<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>

```

Fuente

La etiqueta **** es usada para modificar el tamaño el color y la fuente del texto. Utiliza los atributos "size", "color" y "face" para personalizar tu texto. En resumen, para establecer el tamaño de la fuente, usamos el atributo *size*. Se aceptan valores entre 1 (el más pequeño) y 7 (el más grande). El valor estándar de un texto es 3.

Texto de tamaño 1 (size="1")
 Texto de tamaño 2 (size="2")
 Texto de tamaño 3 (size="3")
 Texto de tamaño 4 (size="4")
 Texto de tamaño 5 (size="5")
 Texto de tamaño 6 (size="6")
 Texto de tamaño 7 (size="7")

```
<p><font size="5">Este es el tamaño de fuente 5</font></p>
```

Para establecer el color del texto, usaremos simplemente *color*

```

<font color="#990000">Este texto es #990000</font> <br >
<font color="red">Este texto es rojo</font>

```

Elige un estilo de letra usando el atributo, *face* de la etiqueta *font*. Podrás escoger cualquier fuente que tengas instalado, pero escoge la fuente con cuidado porque el usuario que visite tu página no podrá ver la fuente si no tiene instalada tu misma fuente. Verá en cambio la fuente por defecto, que es Times New Roman. La solución será escoger varias fuentes que sean parecidas.

```

<p><font face="Bookman Old Style, Book Antiqua, Garamond">Este párrafo ha
formateado su fuente con la etiqueta font!</font></p>

```

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.

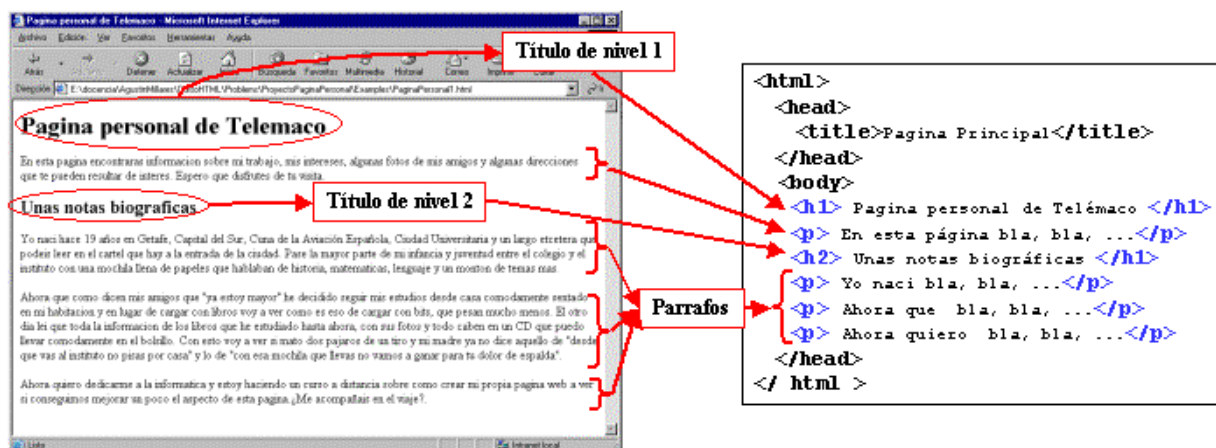
Las etiquetas que definen las cabeceras serán **<H1>**, **<H2>**, **<H3>**, **<H4>**, **<H5>**, **<H6>**. El texto indicado entre las etiquetas de inicio y de fin será el afectado por las cabeceras:

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

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 queramos del documento para resaltar cualquier texto. El modo en que se presentan las cabeceras puede variar de un navegador a otro, y solo 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



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 dondequiera que la ubiquemos.

```
<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>
```

Los colores y HTML

En la composición de webs juegan un papel muy importante los colores. Se indican en valores RGB, es decir, que para conseguir un color cualquiera mezclaremos cantidades de Rojo, Verde y Azul. Los valores RGB se indican en numeración hexadecimal, en base 16. (Los dígitos pueden crecer hasta 16. Como no hay tantos dígitos numéricos se utilizan las letras de la A a la F.

Cada código HTML comprende el símbolo «#» y 6 letras o números. Estos números se expresan en el sistema de numeración hexadecimal. Por ejemplo «FF» en hexadecimal representa el número 255 en Decimal.

```
<font color="#FF0000">Rojo</font>
```

Al Atributo color le damos un valor RGB en formato hexadecimal. El caracter # se coloca al principio de la cadena. Otros colores de ejemplo:

Naranja **#FF8000**

Verde turquesa **#339966**

Azul oscuro **#000080**

Comentarios en HTML

Son 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.

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 de comentario, que terminará cuando insertemos la cadena de fin: `-->`, dos guiones y el símbolo *mayor que*. 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.

Líneas horizontales (Separadores)

Otra de las formas de 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 por lo que ya hemos explicado antes es siempre 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 pixels (valor absoluto) o mediante un tanto por ciento, p.ej. 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. Ejemplo:

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

que nos dan:



- ✓ **<HR ALIGN= " *left / center / right*"></HR>**, que en el caso de que el ancho no sea todo el de la pantalla alinea la línea horizontal a la izquierda, la centrara en pantalla (valor por defecto) o la alinea a la derecha de esta. Ejemplo:

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

que nos da:



- ✓ **<HR COLOR="color"></HR>**, que nos va a permitir definir el color de la línea horizontal, y en donde el color puede venir definido por su valor hexadecimal o por su nombre web estándar. El valor por defecto es el que por defecto tengamos configurado en nuestro PC. Ejemplos

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

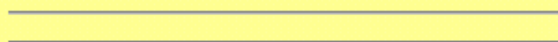
que nos dan:



- ✓ **<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. Ejemplo:

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

que nos dan:



- ✓ **<HR SIZE="número"></HR>**, que nos permite definir la altura o espesor de la línea horizontal. Su valor por defecto suele equivaler a un tamaño **SIZE="2"**. Ejemplo:

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

que nos dan:



- ✓ **<HR TITLE="texto"></HR>**, que 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 (ver tema de Hojas de Estilo en Cascada aplicadas al formateo de texto) en capítulos posteriores.

Atributo	Función
<i>noshade</i>	No muestra sombra, evitando el efecto en tres dimensiones.
<i>width= num / %</i>	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 píxeles
<i>align= "posición"</i>	Alinea la línea a la izquierda (<i>left</i>), a la derecha (<i>right</i>) o la centra (<i>center</i>)
<i>size = numero"</i>	Indica el grosor de la línea en píxeles

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. Versiones anteriores a la 9 del navegador internet explorer no soportan la etiqueta. La sintaxis de la etiqueta es la siguiente.

```
<!DOCTYPE HTML>
<html>
<head>
<title>Ejercicios prácticos HTML5</title>
</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 (N. del T. persona que se dedica a la imprenta) 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>
```

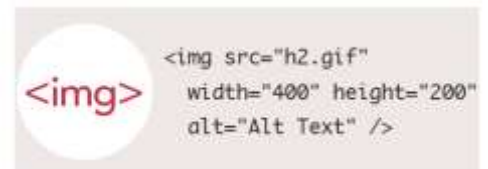
Imágenes

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 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.

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



- ✓ **:** 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 imprescindible 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:

- ✓ **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.
- ✓ **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.

Image width



Image hspace and vspace



use top and bottom

- **Righ:** 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.
- ✓ **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 puedes agregarle bordes a una imagen con la propiedad **BORDER**.



Hipervínculos

Un *enlace* puede ser fácilmente detectado en una página. Basta con deslizar el puntero del mouse sobre las imágenes o el texto y ver cómo cambia de su forma original transformándose por regla general en una mano con un dedo señalador.

Los **vínculos** o **links** o enlaces son simples referencias a otros documentos, pero que en realidad además de la referencia en sí permiten disecionarse al lugar donde apuntan. Esto no solo puede hacerse con texto sino también con imágenes. Todos los **links** tienen dos partes.

- ✓ Una es la *referencia* al lugar a donde nos llevan
- ✓ La otra es el *direccionamiento* a esa referencia que se activa mediante el clic del mouse

Es decir que los links nos permiten relacionar páginas entre sí. Esta característica de los archivos **HTML** nos permite organizar la información en diferentes archivos, y así poder crear una Web, en donde la información se encuentra distribuida en diferentes archivos enlazados entre sí. Un texto que hace referencia a un *link* suele presentarse subrayado y en colores (azul), todo esto dependiendo de la configuración que tenga el explorador del usuario. Existen muchas situaciones en las cuales la utilización de link es muy importante; por ejemplo, saltar a un texto dentro del mismo documento, saltar a otro documento o hacer referencia a información en un lugar determinado dentro de otro documento. En función del destino los enlaces clásicamente son agrupados del siguiente modo:

- ✓ **Enlaces internos:** los que se dirigen a otras partes dentro de la misma página.
- ✓ **Enlaces locales:** los que se dirigen a otras páginas del mismo sitio web.
- ✓ **Enlaces remotos:** los dirigidos hacia páginas de otros sitios web.
- ✓ **Enlaces con direcciones de correo:** para crear un mensaje de correo dirigido a una dirección.
- ✓ **Enlaces con archivos:** para que los usuarios puedan hacer download de ficheros.

La etiqueta `<a>`, sirve para definir y crear un hiperenlace, el cual permite realizar enlaces del documento web con otros documentos o recursos disponibles.

Los enlaces no visitados se visualizan normalmente con el color azul, los enlaces visitados con el color púrpura y por último los enlaces activos mediante el color rojo.

Por defecto al pulsar sobre un enlace, el contenido al cual se está accediendo se ha de mostrar en la misma ventana del navegador, sustituyendo el documento web actual por el que se quiere acceder. La sintaxis para esta etiqueta con todos sus atributos es la siguiente:

`Texto enlace`

Entre los atributos a destacar que tiene la etiqueta se encuentran los siguientes:

- ✓ **href**: este atributo permite establecer la URL en la que se encuentra el documento el cual se quiere enlazar, con el documento web. Se puede especificar una URL absoluta, relativa o un ancla "enlace" en el mismo documento.

Las posibles formas de sintaxis para definir esta etiqueta con el atributo href son las siguientes:

`Texto del enlace`

`Texto del enlace`

`Texto del enlace`

- ✓ **target**: el atributo permite especificar el destino por defecto para todos los enlaces y formularios del documento web.

La sintaxis para definir la etiqueta con el atributo target es la siguiente:

`Texto del enlace`

Entre los posibles valores que se le pueden aplicar al atributo target, encontramos los siguientes:

target	Descripción
<code>_blank</code>	Especifica que el destino sea una nueva ventana del navegador, o una nueva pestaña.
<code>_self</code>	Por defecto. Abre el destino en el mismo lugar donde se haya realizado el clic.
<code>_parent</code>	Abre el destino en un marco asociado.
<code>_top</code>	Abre el destino ocupando todo el cuerpo de la ventana.
<code>framename</code>	Abre el destino en el marco del cual se especifica el nombre.

Enlaces internos

Los enlaces *HTML* que se hacen con otras partes de la misma página. Son los enlaces que apuntan a un lugar diferente dentro de la misma página. Este tipo de enlaces son esencialmente utilizados en páginas donde el acceso a los contenidos puede verse dificultado debido al gran tamaño de la misma. Mediante estos enlaces podemos ofrecer al visitante la posibilidad de acceder rápidamente al principio o final de la página o bien a diferentes párrafos o secciones.

Supongamos que queremos crear un enlace que apunte al final de la página. Lo primero será colocar nuestro enlace origen. Lo pondremos aquí mismo y lo escribiremos del siguiente modo:

```
<a href="#abajo">Ir abajo</a>
```

Enlace con final de este documento, para que probéis su funcionamiento:

[Ir abajo](#)

Como puedes ver, el contenido del enlace es el texto "*Ir abajo*" y el destino, *abajo*, es un punto de la misma página que todavía no hemos definido. Ojo al símbolo #; es él quien especifica al navegador que el enlace apunta a una sección en particular.

En segundo lugar, hay que generar un enlace en el destino. Este enlace llevara por nombre *abajo* para poder distinguirlo de los otros posibles enlaces realizados dentro de la misma página. En este caso, la etiqueta que escribiremos será ésta:

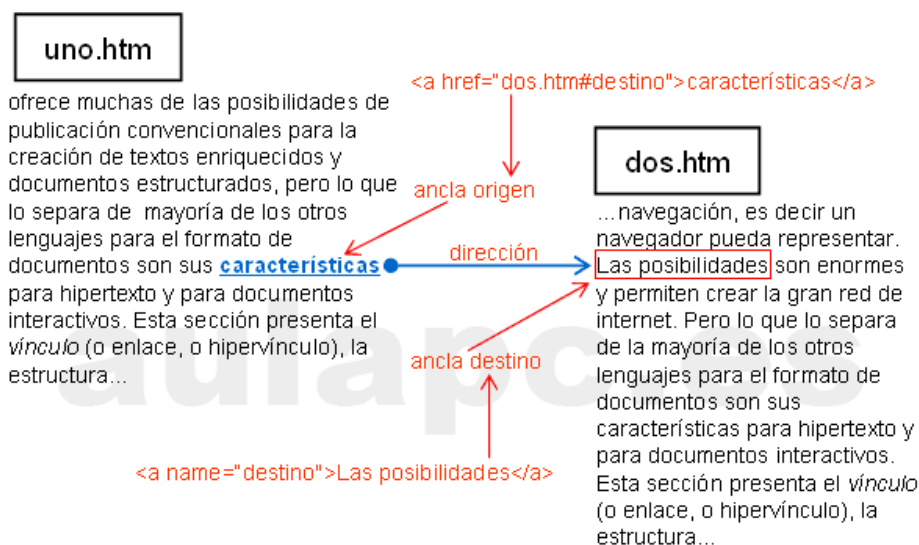
```
<a name="abajo"></a>
```

A decir verdad, estos enlaces, aunque útiles, no son los más extendidos de cuantos hay.

La tendencia general es la de crear páginas (archivos) independientes con tamaños más reducidos enlazados entre ellos por enlaces locales. De esta forma evitamos el exceso de tiempo de carga de un archivo y la introducción de exceso de información que pueda desviar la atención del usuario.

Una aplicación corriente de estos enlaces consiste en poner un pequeño índice al principio de nuestro documento donde introducimos enlaces origen a las diferentes secciones. Paralelamente, al final de cada sección introducimos un enlace que apunta al índice de manera que podamos guiar al navegante en la búsqueda de la información útil para él.

A su vez dentro de una página podemos movernos con lo que se llaman **Marcadores**, que son vínculos que referencian la misma página, tema que veremos más adelante.



```

<!DOCTYPE HTML>
<html>
<head>
<title>Ejemplos HTML5</title>
</head>
<body>
<p>Ejercicios de programación HTML5</p>
<p>
<a href="#Parrafo1">Enlace Al Primer Párrafo</a>
<a href="#Parrafo2">Enlace Al Segundo Párrafo</a>
<a href="#Parrafo3">Enlace Al Tercero Párrafo</a>
</p>
<p><a name="Parrafo1">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 (N. del T. persona que se dedica a la imprenta) desconocido usó
una galería de textos y los mezcló de tal manera que logró hacer un libro de textos espécimen.
No sólo sobrevivió 500 años, sino que también ingresó como texto de relleno en documentos
electrónicos, quedando esencialmente igual al original. Fue popularizado en los 60s con la
creación de las hojas "Letraset", las cuales contenían pasajes de Lorem Ipsum, y más
recientemente con software de autoedición, como por ejemplo Aldus PageMaker, el cual incluye
versiones de Lorem Ipsum./a></p>

<p><a name="Parrafo2">Es un hecho establecido hace demasiado tiempo que un lector se
distrayera con el contenido del texto de un sitio mientras que mira su diseño. El punto de usar
Lorem Ipsum es que tiene una distribución más o menos normal de las letras, al contrario de
usar textos como por ejemplo "Contenido aquí, contenido aquí". Estos textos hacen parecerlo un
español que se puede leer. Muchos paquetes de autoedición y editores de páginas web usan el
Lorem Ipsum como su texto por defecto, y al hacer una búsqueda de "Lorem Ipsum" va a dar
por resultado muchos sitios web que usan este texto si se encuentran en estado de desarrollo.
Muchas versiones han evolucionado a través de los años, algunas veces por accidente, otras
veces a propósito (por ejemplo insertándole humor y cosas por el estilo).</a></p>

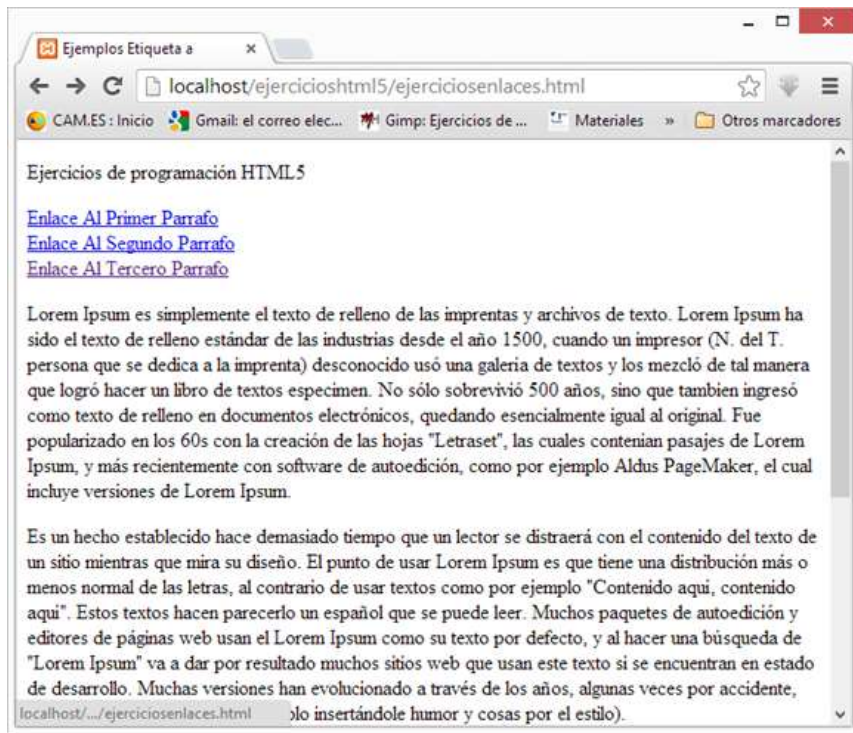
<p><a name="Parrafo3">Al contrario del pensamiento popular, el texto de Lorem Ipsum no es
simplemente texto aleatorio. Tiene sus raíces en una pieza clásica de la literatura del Latín, que
data del año 45 antes de Cristo, haciendo que este adquiera más de 2000 años de antigüedad.
Richard McClintock, un profesor de Latín de la Universidad de Hampden-Sydney en Virginia,
encontró una de las palabras más oscuras de la lengua del latín, "consectetur", en un pasaje de
Lorem Ipsum, y al seguir leyendo distintos textos del latín, descubrió la fuente indudable.
Lorem Ipsum viene de las secciones 1.10.32 y 1.10.33 de "de Finibus Bonorum et Malorum"
(Los Extremos del Bien y El Mal) por Cícero, escrito en el año 45 antes de Cristo. Este libro es
un tratado de teoría de éticas, muy popular durante el Renacimiento. La primera línea del Lorem
Ipsum, "Lorem ipsum dolor sit amet..", viene de una línea en la sección 1.10.32</a></p>

<p><a href="http://www.google.com">Visite Google.com</a></p>
</body>
</html>

```

Para realizar los enlaces como valor del atributo href, se ha de especificar el id asignado a cada párrafo precedido del símbolo "#".

El resultado que se ha de obtener ha de ser similar al que se muestra en la siguiente imagen:



Enlaces locales

*Enlaces **HTML** con otras páginas del mismo sitio web.* Como hemos dicho, un sitio web está constituido de páginas interconectadas. Para crear este tipo de enlaces, deberemos codificar lo siguiente:

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

Por regla general, para una mejor organización, los sitios suelen estar ordenados por directorios. Estos directorios suelen contener diferentes secciones de la página, imágenes, sonidos...Es por ello que en muchos casos no nos valdrá con especificar el nombre del archivo, sino que tendremos que especificar además el directorio en el que nuestro archivo.html está alojado.

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo del uso de marcadores </title>
  </head>
  <body>
```

```
<a name="arriba"></a>
```

En esta página puedes ir al `primer` apartado, al `segundo` o al `tercero`.

```
<a name="primero"><h1>Primer apartado</h1></a>
```

Aquí tienes el primer apartado. Al pulsar sobre el enlace, el navegador habrá saltado a esta parte de la página. Quizás si todo entra en la pantalla no logres ver el efecto pero prueba a poner más

texto aquí o hacer zoom y conseguirás verlo.

```
<a name="segundo"><h1>Segundo apartado</h1></a>
```

Aquí tienes el segundo apartado. Al pulsar sobre el enlace, el navegador habrá saltado a esta parte de la página. Quizás si todo entra en la pantalla no logres ver el efecto pero prueba a poner más texto aquí o hacer zoom y conseguirás verlo.

```
<a name="tercero"><h1>Tercer apartado</h1></a>
```

Aquí tienes el tercer apartado. Al pulsar sobre el enlace, el navegador habrá saltado a esta parte de la página. Quizás si todo entra en la pantalla no logres ver el efecto pero prueba a poner más texto aquí o hacer zoom y conseguirás verlo.

Volver [arriba](#arriba).

```
</body>
```

```
</html>
```



Enlaces Relativos y Absolutos

En los enlaces, se puede especificar la ruta absoluta a los mismos, o una ruta relativa teniendo en cuenta el documento actual. En el siguiente ejemplo se muestra el uso de dichas rutas mostrando enlaces, a sitios web que se encuentran fuera del lugar donde está almacenado el documento y otros enlaces que muestran rutas relativas desde la posición actual del documento a otros documentos ubicados en el mismo sitio de alojamiento.

```
<html>
<head>
<title>Ejemplos Etiqueta a</title>
</head>
<body>
<p>Enlaces a buscadores externos</p>
<p><a href="http://www.google.com">Buscador Google</a></p>
<p><a href="http://www.bing.es">Buscador Bing</a></p>
<p>Enlaces a documentos con rutas relativas</p>
<p><a href="./imagenes/imagen1.jpg">Imagen 1</a></p>
<p><a href="../documentos/pdfs/texto1.pdf">Texto 1</a></p>
<p><a href="/paginas/segunda.html">Siguiete Página</a></p>
</body>
```


</html>

- ✓ **Enlaces absolutos:** Para crear un enlace absoluto, basta con especificar la dirección en el atributo **REF**. Ejemplo:

```
<A HREF="http://www.issd.edu.ar/">ISSD</A>
```

- ✓ **Enlaces relativos:** Para crear un enlace relativo, basta con especificar la dirección relativa en el atributo REF. Ejemplo:

```
<A HREF="introHTML.html">Página inicial</A>
```

Enlaces a correo electrónico, ftp

El siguiente ejemplo muestra el uso de realiza un enlace a una dirección de correo electrónico, aunque el enlace sea a una dirección de correo electrónico, para que el mensaje sea enviado se requiere que el usuario realice esta acción, ya que este tipo de enlaces abren el programa de correo electrónico para enviar el mensaje. Los desarrolladores web suelen emplear formularios para el envío de mensajes de correo, con programación en lenguajes como php, asp, javascript, etc.

```
<!DOCTYPE HTML>
<html>
<head>
<title>Ejemplos Etiqueta a</title>
</head>
<body>
<p>Ejercicios de programación HTML5</p>
<p>El siguiente ejemplo muestra el uso de enlaces a correo electrónico</p>
<p>Para enviar un mensaje a
<a href="mailto:formacion@alumno.com.ar">formacion@alumno.com.ar</a></p>
<p>Para acceder a una dirección de ftp</p>
<p>El siguiente <a href="ftp://ftp.alumno.com.ar/archivo.zip">enlace</a> , accede a un archivo
en un servidor ftp.</p>
</body>
</html>
```



Conclusión

HTML es el principal lenguaje de marcado de Internet. Cada página HTML tiene una serie de elementos que crean la estructura del contenido de una página web o aplicación.

Es un lenguaje apto para principiantes con gran apoyo y se utiliza principalmente para páginas web estáticas. HTML funciona mejor junto con CSS para el estilo y JavaScript para la funcionalidad. Si estas interesado en el desarrollo web como parte profesional de tu vida, esta parte de la materia es de suma importancia, para que adquieras destreza y experiencia en el lenguaje trata de desarrollar la mayor cantidad de ejercicios posibles esto te ayudara con la comprensión de la sintaxis y la rapidez para codificar.

¡Buena suerte, nos vemos en la próxima clase!



Ejercitación

Codifica los siguientes ejemplos, estos te permitirán adquirir habilidad en maquetado. Trata de el ejercicio quede lo más parecido a la muestra. Puedes colocar cualquier texto o imagen, lo importante que no pierda la estructura de maquetación.

Ejercicio 1 (Texto- Imagen)

Images in HTML



Ejercicio 2 (Texto-Imagen-Texto- 3 subtextos_)



Ejercicio 3 (3 Textos a la derecha – imagen a la izquierda)



Ejercicio 4 (imagen – titulo- párrafo- imagen-párrafo)



Ejercicio 5 (Imagen-Titulo-titulo-párrafo- texto con link)



NUESTRAS OFERTAS DEL MES

En la Agencia de viajes Los Alpes estamos celebrando nuestros primeros 20 años, y por ello le hemos preparado una serie de **OFERTAS Y VIAJES ESPECIALES** que no podrá resistir. Entre ellas destacan:

[ANDORRA](#) [ORLANDO](#) [LOS ANGELES](#) [PARÍS](#) [MOSCÚ](#)

Ejercicio 6 (Titulo- Párrafo - imagen)

Escalar una imagen

El elemento de imagen contiene tres atributos: fuente (src) alt text (alt) y estilo (style)

El ancho y alto de la imagen se especifica en el atributo de estilo de la imagen



Ejercicio 7 (Texto, imagen- 2 textos- imagen)

03-DIC / 2013
FESTEJA EL ESPIRITU DE DAR



#undiaparadar
UN MOVIMIENTO SOLIDARIO



Ejercicio 8 (Titulo- Subtitulo- Titulo – texto con link - imágenes)



Autoevaluación

En base a los conocimientos adquiridos, con sus propias palabras responde las siguientes preguntas:

1. ¿De qué se encarga la W3C?
2. ¿Cuáles son las etiquetas físicas y cual las lógicas? Diferencia
3. ¿Que son las zonas semánticas de Html5?
4. ¿Cuáles son las tags que dejaron de utilizarse en versión de html5?
5. Describe el formato de imágenes WEBP
6. Haz una lista de los caracteres y símbolos especiales en HTML
7. ¿Es mejor usar imágenes en formato GIF o JPG para ser insertada en una página web?
¿Porque?
8. ¿Quién determina las versiones de HTML?
9. ¿Cuál es la función de un maquetaador web?
10. ¿Para qué sirve maquetar un sitio?
11. Haz una lista de algunas aplicaciones gráficas para maquetar páginas web.
12. ¿Que es el DOM de Html?