



**Academia**

# Desarrollo Web Frontend

# Hyper text markup language HTML 5



# CONTENIDO

## ▪ HTML

- ✓ Características de html5
- ✓ Elementos, etiquetas y atributos.
- ✓ Estructura básica de un documento: html, head, body, meta, title, link
- ✓ Estructura del cuerpo: header, nav, section, aside, footer
- ✓ Encabezados: H1...H6
- ✓ Enlaces o hipervínculos
- ✓ Imágenes
- ✓ Listas y listas anidadas
- ✓ Tablas
- ✓ Formularios: form, input, select, radio, button, submit
- ✓ Elemento DIV

# Competencias

- ✓ Conocer los diferentes elementos involucrados en la estructura básica de un documento HTML.
- ✓ Desarrollar formularios web mediante el uso de los controles disponibles en HTML



# Teoría de la programación web

# Web Programming theory

## Orígenes de la web

Tim Berners Lee inventó la web (World Wide Web) en 1989, desencadenando una revolución de la información.

Berners Lee trabajaba para la CERN<sup>1</sup>, un centro de investigación científica de Ginebra, formado por científicos e investigadores.

El propósito de Berners Lee era posibilitar el uso compartido de miles de documentos científicos.

Su trabajo culminó con la creación de un lenguaje de hipertexto que hacía posible enlazar y distribuir documentos relacionados a través de un equipo informático o de una red

# Web Programming theory

## Orígenes de la web

Para poder enviar y recibir documentos a través de la web desarrolló un protocolo simple, conocido como HTTP (HyperText Transfer Protocol) y el lenguaje HTML (HyperText Markup Language) para crear documentos que se pudieran transferir con características de hipertexto.

La web inicial, tal como la creó Tim Berners Lee era como una biblioteca en línea, con documentos conectados mediante enlaces, donde una comunidad científica podía leer y acceder libremente a la información con propósitos de investigación y con afán educativo.

Los sitios web de ese entonces eran colecciones de documentos escritos en lenguaje HTML, y en consecuencia, las páginas web estaban basadas en texto, eran simples y además, estáticas.



# Web Programming theory

## Sitios web estáticos

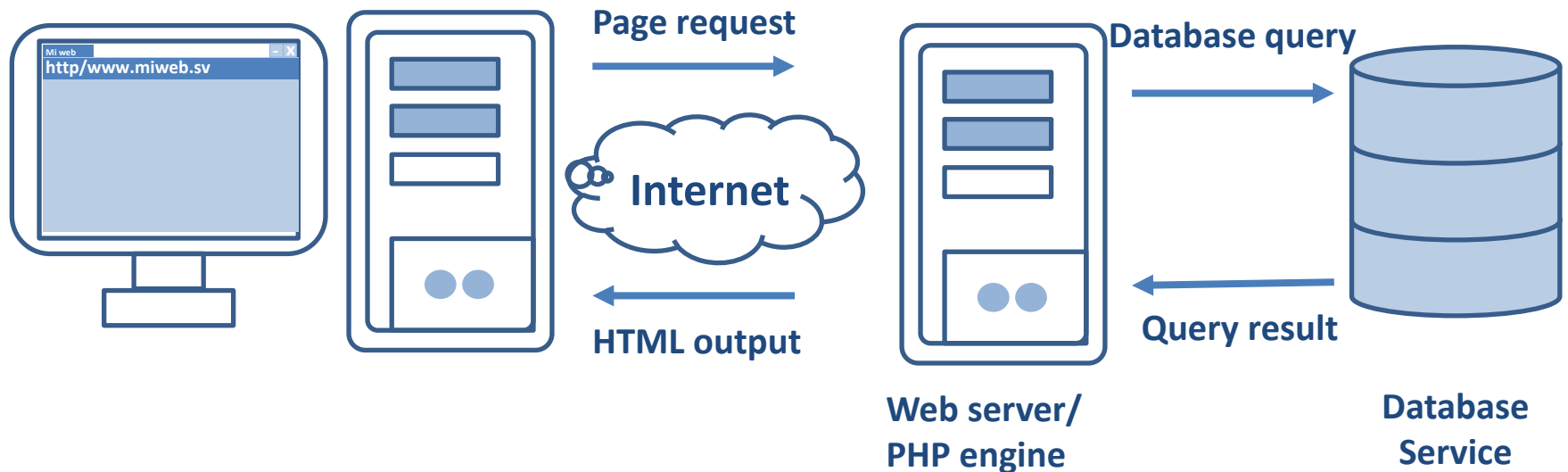
Que una página sea estática, quiere decir que cada vez que se carga esa página en el navegador, siempre tendrá exactamente el mismo contenido, la misma apariencia, independiente de quién ingrese y cuándo lo haga.

Las páginas web estáticas se podían utilizar para mostrar contenido como texto, imágenes y enlaces, pero no existe una forma que permita recordar nombres de usuarios y sus preferencias, realizar pagos en línea, crear información personalizada a partir de una base de datos, incrustar vídeos o subir archivos a un servidor y ponerlo a disposición en ese mismo momento, realizar registros de usuarios, etc.

# Web Programming theory

## Sitios web dinámicos

En un sitio web dinámico el contenido de las páginas no existe previamente en el código del documento web, este contenido se genera cada vez que el usuario visita o carga la página en su navegador, ya sea normalmente o iniciando sesión.



# Web Programming theory

## Sitios web dinámicos

Esto significa que el documento web puede contener entre sus líneas instrucciones en un lenguaje de programación o scripts que mediante estructuras de control, funciones o métodos de una clase puede generar el contenido a partir de cálculos o mediante lógica de programación.

Adicionalmente, en una página web dinámica del lado del servidor, puede incluir sentencias en un lenguaje de script o de programación para recuperar información almacenada en una base de datos y a partir de esta generar el contenido.

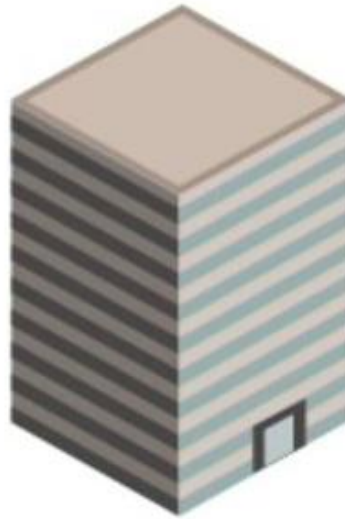
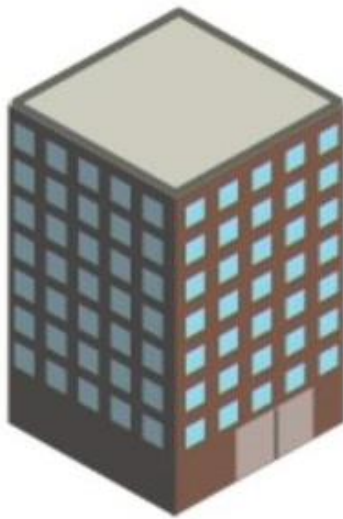
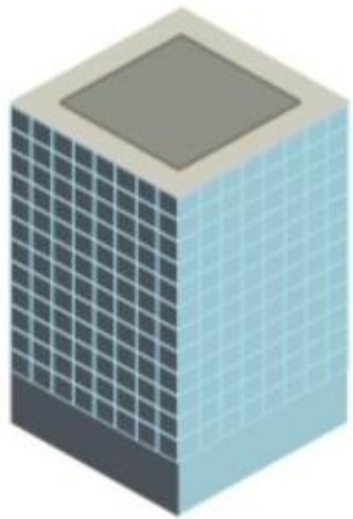


# Lenguaje de etiquetas de hipertexto

# OBJETIVOS

- Utilizar el lenguaje de etiquetas html5 para la estructuración del contenido de una página web básica acorde a un requerimiento entregado.

# HTML



CSS



HTML

# HTML

## Teoría

El lenguaje HTML está basado en elementos.

Un elemento HTML usualmente está compuesto por una etiqueta de apertura,

una etiqueta de cierre y el contenido escrito entre ellas.

El contenido de un elemento HTML puede ser: texto, imágenes u otros elementos HTML.



## Teoría

HTML5 (HyperText Markup Language, versión 5) es la quinta revisión del lenguaje HTML. Esta nueva versión, y en conjunto con CSS3, define los nuevos estándares de desarrollo web, rediseñando el código para resolver problemas y actualizarlo.





# Características de HTML5

## Características de HTML5

1. Nuevas etiquetas semánticas para estructurar los documentos HTML, destinadas a reemplazar la necesidad de tener una etiqueta `<div>` que identifique cada bloque de la página.
2. Los nuevos elementos multimedia como `<audio>` y `<video>`.
3. La integración de gráficos vectoriales escalables (SVG) en sustitución de los genéricos `<object>`, y un nuevo elemento `<canvas>` que nos permite dibujar en él.
4. El cambio, redefinición o estandarización de algunos elementos, como `<a>`, `<cite>` o `<menu>`.
5. MathML para fórmulas matemáticas.
6. Almacenamiento local en el lado del cliente.
7. Y otros muchos nuevos APIs que veremos a lo largo de los siguientes capítulos.

# Elementos, etiquetas y atributos

HTML es un lenguaje de marcas compuesto por un conjunto de elementos que son la base de su estructura. Estos elementos conforman una caja de herramientas que los desarrolladores pueden utilizar para dar forma a sus documentos. Por ejemplo, puedes encontrar elementos que insertan párrafos, videos u otros documentos, o elementos que marcan citas, texto importante o los títulos de libros, películas, trabajos y otras obras.

```
1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <title> HTML Page </title>
5     <meta http-equiv="content-language" content="en">
6     <meta name="description" content="">
7     <meta name="keywords" content="">
8     <meta name="ROBOTS" content="INDEX, ALL">
9     <link rel="shortcut icon" href="/favicon.ico">
10    <link rel="stylesheet" href="http://localhost/css.css" type="text/c
    <script type="text/javascript" src="http://localhost/javascript.js">
```

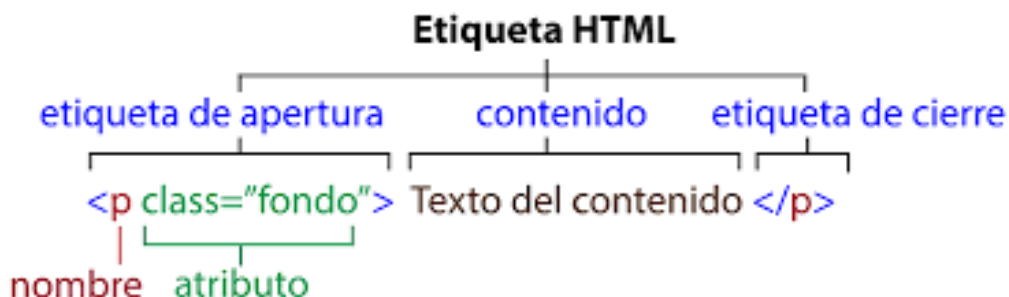
# Elementos, etiquetas y atributos

Un elemento HTML está normalmente compuesto por dos etiquetas: la etiqueta de apertura y la etiqueta de cierre.

Una etiqueta de apertura se compone del nombre del elemento encerrado por los símbolos menor-que "<" y mayor-que ">".

La etiqueta de cierre se construye de igual manera que la de apertura pero, en este caso, el nombre del elemento es precedido por una barra diagonal ("/").

Los atributos son valores que se pueden asignar a las etiquetas, como colores, fondos, etc.

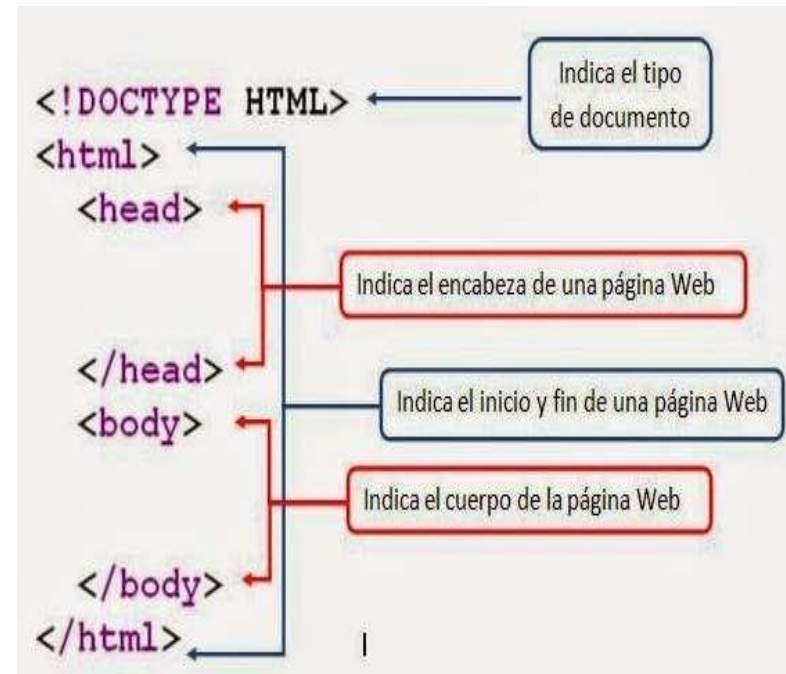


# Estructura básica de un documento: html, head, body, meta, title, link

## UNA PÁGINA HTML SE COMPONE DE MINIMO 3 ELEMENTOS

El elemento **HTML** marca el inicio y el fin del Documento. Su interior, se divide en dos partes:

1. La cabecera, marcada por la **etiqueta head**, incluye información sobre la página que es relevante para el navegador web que la interpreta.
2. El cuerpo, marcado por la **etiqueta body** incluye el contenido que será publicado.



# Estructura básica de un documento: html, head, body, meta, title, link

## CABECERA DEL DOCUMENTO (HEAD)

La cabecera del documento es la sección comprendida entre `<head>` y `</head>`. En ella se debe encontrar, obligatoriamente, el título (entre las etiquetas `<title>` y `</title>`). La siguiente tabla muestra un resumen de elementos que pueden ir dentro de la etiqueta head.

Etiqueta en cabecera	Función	¿Es obligatoria?
<code>&lt;title&gt;</code>	Da un título al documento HTML	Sí
<code>&lt;base&gt;</code>	Define ruta de acceso	No
<code>&lt;link&gt;</code>	Define archivos vinculados	No
<code>&lt;meta&gt;</code>	Define metadatos como descripción y palabras clave	No
<code>&lt;script&gt;</code>	Delimita scripts incluidos	No
<code>&lt;style&gt;</code>	Delimita definición de estilos	No

# Estructura básica de un documento: html, head, body, meta, title, link

**Etiquetas <title>** Sirve para describir el contenido de la página por ejemplo:

```
<title>Manuales y tutoriales sobre programación </title>
```

**Etiquetas <link>** Sirven para indicar que el documento html está relacionado con otro archivo o recurso externo. Por ejemplo `<link rel="stylesheet" type="text/css" href="estilos.css">` sirve para indicar que el documento HTML está vinculado al archivo estilos.css

**Etiqueta <style>** Sirve para incluir estilos CSS que permiten dotar de colores, bordes, imágenes de fondo, etc. a los elementos de la página web.

**Etiquetas <meta>** Sirven para incluir información que no se muestra como parte de la página web pero sirve para informar de características de la página web, como su descripción breve y sus palabras clave.

# Estructura básica de un documento: html, head, body, meta, title, link

## EL CUERPO (BODY) DEL DOCUMENTO HTML

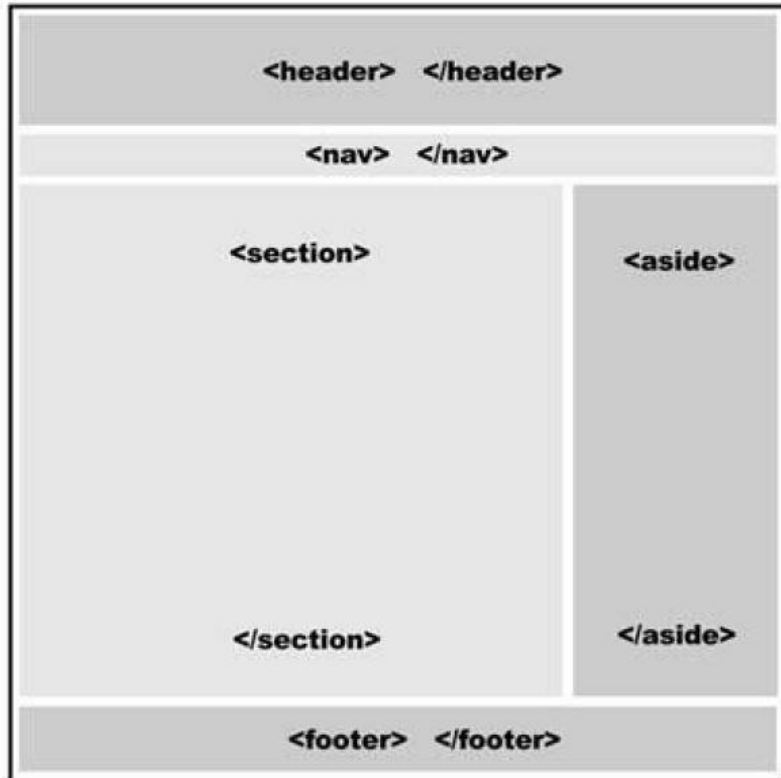
### Estructura básica



El **cuerpo (body)** del documento html es normalmente lo más importante. Es aquí donde debemos colocar el contenido de nuestra página: texto, fotos, etc.

El cuerpo está delimitado por las etiquetas **<body>** (apertura) y **</body>** (cierre).

# Estructura del cuerpo: header, nav, section, aside, footer





# Estructura del cuerpo: header, nav, section, aside, footer

## Cabecera <header>

Uno de los nuevos elementos incorporados en HTML5 es **<header>**. El elemento **<header>** no debe ser confundido con **<head>** usado antes para construir la cabecera del documento.

Del mismo modo que **<head>** la intención de **<header>** es proveer información introductoria (títulos, subtítulos, logos), pero difiere con respecto a **<head>** en su alcance. Mientras que el elemento **<head>** tiene el propósito de proveer información acerca de todo el documento, **<header>** es usado solo para el cuerpo o secciones específicas dentro del cuerpo.

A diagram consisting of a light gray rectangular box with a thin black border. Inside the box, the text "<header> </header>" is centered in a black, monospaced font.

**<header> </header>**

# Estructura del cuerpo: header, nav, section, aside, footer

## Cabecera <header>

```
<!DOCTYPE html>
<html>
<body>
  <header>
    <h1>Most important heading here</h1>
    <h3>Less important heading here</h3>
    <p>Some additional information here.</p>
  </header>
  <p>Lorem Ipsum dolor set amet....</p>

</body>
</html>
```

# Estructura del cuerpo: header, nav, section, aside, footer

## Barra de Navegación <nav>

Esta barra es generada en HTML5 con el elemento **<nav>**. El elemento **<nav>** se encuentra dentro de las etiquetas **<body>** pero es ubicado después de la etiqueta de cierre de la cabecera (**</header>**), no dentro de las etiquetas **<header>**. Esto es porque **<nav>** no es parte de la cabecera sino una nueva sección.



```
<nav> </nav>
```

# Estructura del cuerpo: header, nav, section, aside, footer

## Barra de Navegación <nav>

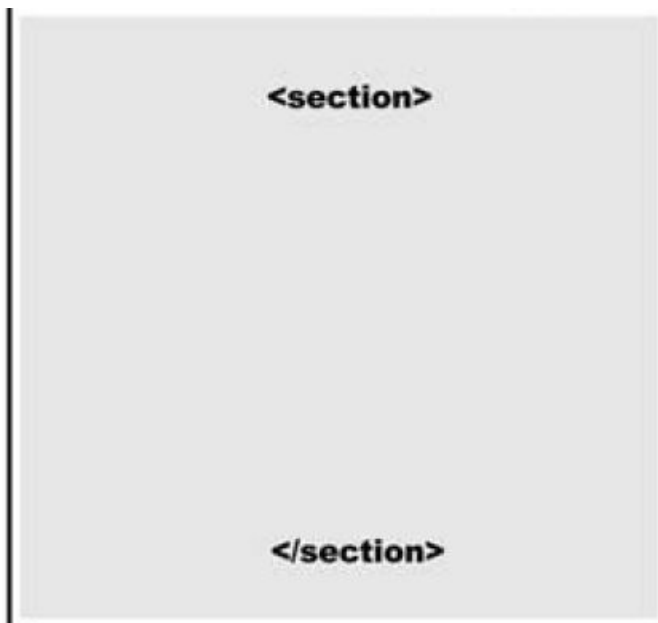
```
<!DOCTYPE html>
<html>
<body>

<nav>
<a href="/html/">HTML</a> |
<a href="/css/">CSS</a> |
<a href="/js/">JavaScript</a> |
<a href="/jquery/">jQuery</a>
</nav>

</body>
</html>
```

# Estructura del cuerpo: header, nav, section, aside, footer

**<section>**



Sección de Información Principal

**<section>**

Esta sección es la columna de Información Principal, contiene la información más relevante del documento y puede ser encontrada en diferentes formas (por ejemplo, dividida en varios bloques o columnas).

Debido a que el propósito de estas columnas es más general, el elemento en HTML5 que especifica estas secciones se llama simplemente **<section>**.

# Estructura del cuerpo: header, nav, section, aside, footer

## **<section>**

```
<!DOCTYPE html>
<html>
<body>

<section>
  <h1>WWF</h1>
  <p>The World Wide Fund for Nature (WWF) is an international organization working on issues
regarding the conservation, research and restoration of the environment, formerly named the
World Wildlife Fund. WWF was founded in 1961.</p>
</section>

<section>
  <h1>WWF's Panda symbol</h1>
  <p>The Panda has become the symbol of WWF. The well-known panda logo of WWF originated from
a panda named Chi Chi that was transferred from the Beijing Zoo to the London Zoo in the same
year of the establishment of WWF.</p>
</section>

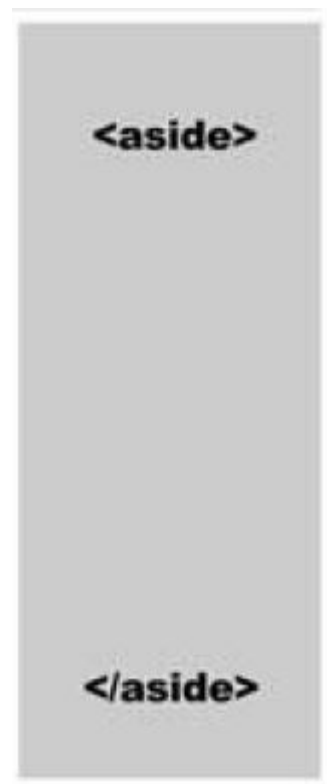
</body>
</html>
```

# Estructura del cuerpo: header, nav, section, aside, footer

## Barra Lateral <aside>

La columna llamada Barra Lateral se ubica al lado de la columna Información Principal.

Esta es una columna o sección que normalmente contiene datos relacionados con la información principal pero que no son relevantes o igual de importantes.



# Estructura del cuerpo: header, nav, section, aside, footer

## Barra Lateral <aside>

```
<!DOCTYPE html>
<html>
<body>

<p>My family and I visited The Epcot center this summer.</p>

<aside>
  <h4>Epcot Center</h4>
  <p>The Epcot Center is a theme park in Disney World, Florida.</p>
</aside>

<p><strong>Note:</strong> The aside tag is not supported in Internet
Explorer 8 and earlier versions.</p>

</body>
</html>
```



# Estructura del cuerpo: header, nav, section, aside, footer

## Pie de página <footer>

Para finalizar la construcción de la plantilla o estructura elemental una pagina o un documento HTML5, solo necesitamos un elemento más. Ya contamos con la cabecera del cuerpo, secciones con ayuda para la navegación, información importante y hasta una barra lateral con datos adicionales, por lo tanto lo único que nos queda por hacer es cerrar nuestro diseño para otorgarle un final al cuerpo del documento. HTML5 provee un elemento específico para este propósito llamado <footer>.

A diagram showing a footer element. It consists of a light gray rectangular box with a thin black border. Inside the box, the text "<footer> </footer>" is centered in a black, monospace-style font.

`<footer> </footer>`

# Estructura del cuerpo: header, nav, section, aside, footer

## Pie de página <footer>

```
<!DOCTYPE html>
<html>
<body>

<footer>
  <p>Posted by: Hege Refsnes</p>
  <p>Contact information: <a href="mailto:someone@example.com">someone@example.com</a>.</p>
</footer>

<p><strong>Note:</strong> The footer tag is not supported in Internet Explorer 8 and earlier
versions.</p>

</body>
</html>
```

# Encabezados: H1...H6

## Encabezados

Los encabezados, se utilizan para establecer títulos dentro del contenido. Están estructurados en una jerarquía de 6 niveles, dónde el nivel 1 representa titulares de mayor importancia y el nivel 6 se utiliza para titulares menos relevantes.

`<h1>Hola Mundo!</h1>`

`<h2>Hola Mundo!</h2>`

`<h3>Hola Mundo!</h3>`

`<h4>Hola Mundo!</h4>`

`<h5>Hola Mundo!</h5>`

`<h6>Hola Mundo!</h6>`

**Hola Mundo**

**Hola Mundo**

**Hola Mundo**

**Hola Mundo**

**Hola Mundo**

**Hola Mundo**

# Etiquetas relacionadas con texto

Etiqueta HTML	Definición
<code>&lt;p&gt;&lt;/p&gt;</code>	Define un párrafo.
<code>&lt;br&gt;</code>	Define un salto de línea.
<code>&lt;b&gt;&lt;/b&gt;</code>	Define texto en negrita.
<code>&lt;i&gt;&lt;/i&gt;</code>	Permite texto en cursiva.
<code>&lt;u&gt;&lt;/u&gt;</code>	Permite aplicar un subrayado a un texto.
<code>&lt;del&gt;&lt;/del&gt;</code>	Tacha el texto dentro de la etiqueta.
<code>&lt;hr&gt;&lt;/hr&gt;</code>	Define una línea horizontal, para especificar un cambio en el contenido.
<code>&lt;details&gt;&lt;/details&gt;</code>	Define texto adicional que el usuario puede ver o ocultar.
<code>&lt;summary&gt;&lt;/summary&gt;</code>	Define el título utilizado para la etiqueta details.
<code>&lt;sub&gt;&lt;/sub&gt;</code>	Permite definir un subíndice.
<code>&lt;sup&gt;&lt;/sup&gt;</code>	Permite definir un superíndice.

# Enlaces o hipervínculos

Esta etiqueta se utiliza para crear los llamados enlaces, vínculos o hipervínculos. La idea es establecer una referencia a una dirección o URL donde está alojado ese otro documento de destino, que puede ser una página web, un archivo PDF, una imagen o cualquier otro tipo de documento.

```
<a href="http://sitio.com">Enlace</a>
```

# Enlaces o hipervínculos

Esta etiqueta debe tener, como mínimo, el atributo **href**, ya que es el atributo con el que se especifica la dirección URL al documento que se quiere enlazar. Aún así, la etiqueta **<a>** también tiene varios atributos opcionales, veamos un resumen de todos ellos:

Atributo	Valor	Descripción
<b>href</b>	<u>URL</u>	Enlace al documento que se quiere cargar. <b>Atributo obligatorio.</b>
<b>download</b>	<u>nombre.ext</u>	Descarga el enlace (href) en lugar de abrirlo. Si se indica valor, se renombra.
<b>target</b>	<b>_blank</b>	Abre el enlace en una nueva pestaña.
	<b>_self</b>	Abre el enlace en la pestaña o <b>&lt;iframe&gt;</b> actual.
	<b>_parent</b>	Abre el enlace en el documento padre.
	<b>_top</b>	Abre el enlace en el documento raíz (padre global).
	<u>nombre</u>	Abre el enlace en el <b>&lt;iframe&gt;</b> con el nombre especificado.
<b>rel</b>	<b>alternate</b>	Indica que el enlace es una versión alternativa (idioma o formato diferente).
	<b>author</b>	Indica que el enlace es la web del autor de la página actual.

# Enlaces o hipervínculos

Veamos a continuación algunos ejemplos de enlaces utilizando algunos atributos indicados en la tabla superior para comprender así su funcionamiento:

```
<ul>
  <li><a rel="author" href="http://www.emezeta.com/">Emezeta</a></li>
  <li><a href="http://lenguajecss.com/" target="_blank">LenguajeCSS</a></li>
  <li><a href="http://pagina.com/documento.pdf" download="A-38.pdf">PDF</a></li>
  <li><a href="http://pagina.com/documento-en.pdf" hreflang="en">PDF</a></li>
</ul>
```

HTML

# Rutas absolutas



Anatomía de una ruta absoluta

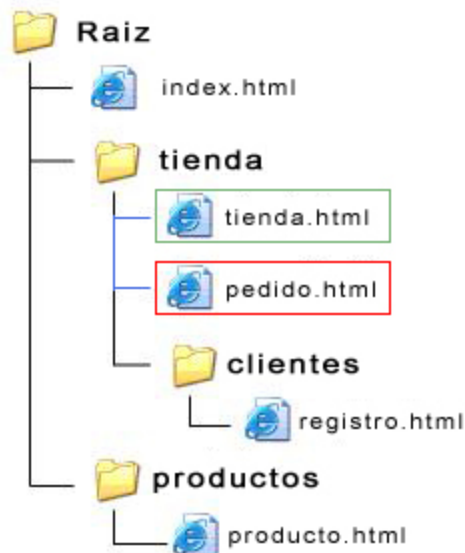


# Rutas relativas

## Enlace hacia un archivo que está en el mismo directorio

Supongamos que queremos añadir un link en el archivo **"pedido.html"** que enlace con el archivo **"tienda.html"** que están dentro del directorio **"tienda"**. Al estar los dos recursos en el mismo sitio, lo único que tendríamos que hacer sería añadir en el enlace la ruta **"tienda.html"**. A continuación os dejamos el código HTML que habría que utilizar:

```
<a href="tienda.html">Tienda</a>
```



# Rutas relativas

Crear un enlace hacia un archivo que está en una subcarpeta del mismo nivel

En este caso supongamos que queremos añadir un enlace en **"pedido.html"** que apunte hacia el archivo **"registro.html"** que está dentro del subdirectorio **"clientes"**. En este caso, la ruta relativa hacia ese archivo sería la siguiente: **"clientes/registro.html"**. Si lo pasamos a HTML nos quedaría algo similar a lo siguiente.

```
<a href="clientes/registro.html">Registro</a>
```

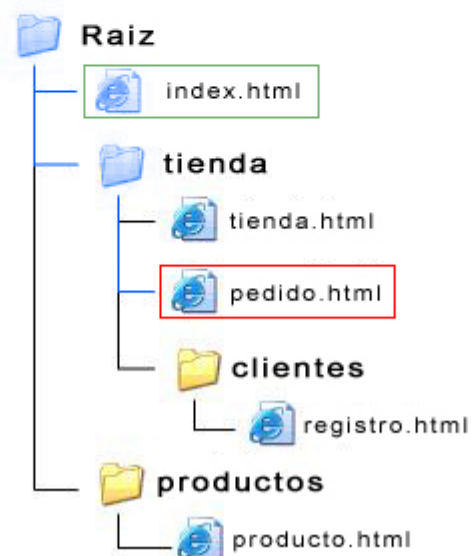


# Rutas relativas

## Enlazar hacia un archivo que se encuentra en un nivel superior

Supongamos que ahora queremos poner un enlace en nuestra página **"pedido.html"** que vaya al **"index.html"** de nuestro portal y que se encuentra en un nivel superior al nuestro. En este caso, la ruta relativa a utilizar sería **"../index.html"**. En formato HTML quedaría de la siguiente forma.

```
<a href="../index.html">Index</a>
```

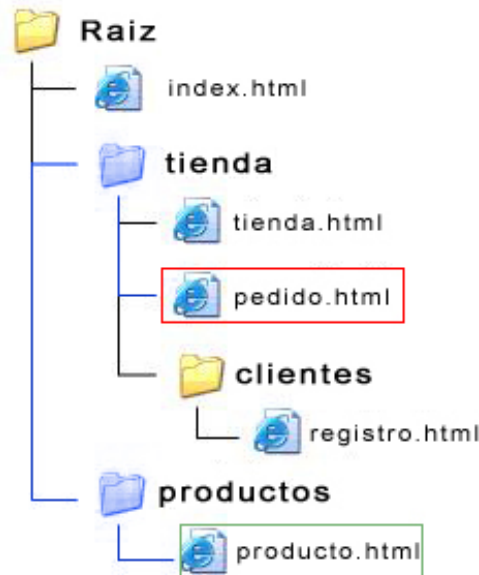


# Rutas relativas

## Enlazar hacia un archivo que se encuentra en una subcarpeta distinta

Supongamos que en este caso queremos enlazar en el archivo **"pedido.html"** el recurso **"producto.html"** que se encuentra en otra subcarpeta distinta a donde está el archivo **"pedido.html"**. En este caso, la ruta relativa a utilizar sería: **"../productos/producto.html"**. En formato HTML quedaría de la siguiente forma.

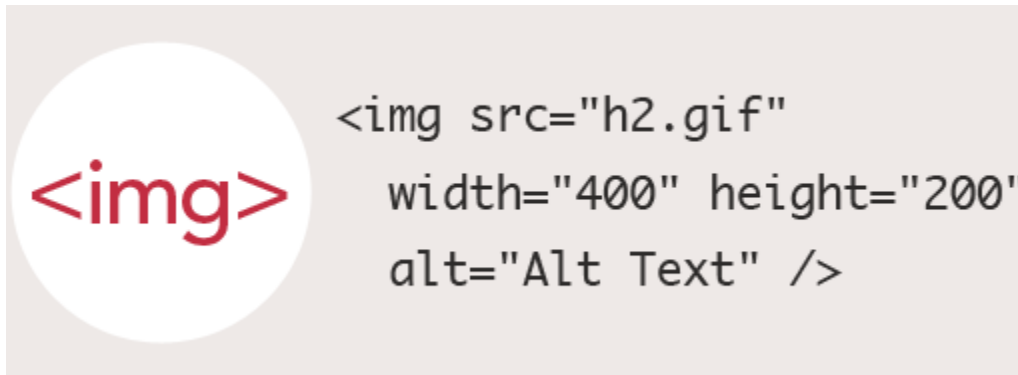
```
<a href="../productos/producto.html">Producto</a>
```



# Imágenes

Las imágenes utilizadas en una página web pueden ser de dos tipos: de contenido o de decoración.

En el primer caso, si la imagen pertenece al contenido y tema tratado en esa página, debería incluirse mediante una etiqueta **HTML <img>**, pero si por el contrario pertenece a la decoración de la página, deberíamos incluirla como un fondo mediante por ejemplo la propiedad **CSS background-image**.



# Imágenes

## Etiquetas para imágenes

Para incluir imágenes en el contenido de una página utilizaremos la etiqueta **<img>**, que es una etiqueta muy sencilla, que dispone de varios atributos para modificar como se verá la imagen (los atributos `src` y `alt` son siempre obligatorios):

Atributo	Descripción
<code>src</code>	Indica el nombre o la URL de la imagen a mostrar. <b>Atributo obligatorio.</b>
<code>alt</code>	Establece un texto alternativo que describa la imagen a mostrar. <b>Atributo obligatorio.</b>
<code>width</code>	Indica el ancho de la imagen. No se debe indicar unidad. Se aconseja hacerlo desde CSS.
<code>height</code>	Indica el alto de la imagen. No se debe indicar unidad. Se aconseja hacerlo desde CSS.

# Imágenes

Un ejemplo básico para colocar una imagen sería el siguiente:

```

```

\* Los atributos width y height redimensionan la imagen al tamaño de ancho y alto indicado, pero la imagen realmente tiene su propio tamaño

# Imágenes

## Formatos de imágenes

En el ámbito informático existen múltiples formatos de imágenes pero no todos son aptos para utilizar en web. Los formatos más utilizados son:

Formato	Características	¿Recomendado?
<a href="#">PNG</a>	Soporta transparencia. Compresión sin pérdidas. Imágenes «lisas».	Sí
<a href="#">JPG</a>	Compresión con pérdidas. Ideal para imágenes con texturas.	Sí
<a href="#">SVG</a>	Formato vectorial. Ideal para imágenes escalables.	Sí
<a href="#">GIF</a>	Formato para imágenes pequeñas y animadas.	Sí, pero evitar.
<a href="#">WEBP</a>	Alternativa libre de Google al JPEG. Soporta transparencias.	<a href="#">Con precaución</a>
<a href="#">JPEG2000</a>	Evolución del JPEG.	No, <a href="#">poco soporte</a>
<a href="#">JPEG-XR</a>	Alternativa libre de Microsoft al JPEG.	No, <a href="#">poco soporte</a>
<a href="#">APNG</a>	Alternativa libre a GIF. Compatible con PNG. Soporta animaciones.	<a href="#">Con precaución</a>
<a href="#">AVIF</a> 	Formato de imagen basado en AV1. No confundir con videos AVI.	No, <a href="#">poco soporte</a>



# Listas y listas anidadas

En HTML se pueden tener listas no ordenadas, listas ordenadas o listas de descripción.

## Unordered List

- The first item
- The second item
- The third item
- The fourth item

## Ordered List

1. The first item
2. The second item
3. The third item
4. The fourth item

## Description List

**The first item**  
Description of item  
**The second item**  
Description of item

# Listas y listas anidadas

## Lista no ordenada (Unordered List)

Una lista no ordenada comienza con la etiqueta `<ul>`. Cada elemento de la lista comienza con la etiqueta `<li>`.

Los elementos de la lista por defecto se marcarán con *pequeños círculos negros*.

```
<ul>
  <li>Inicio</li>
  <li>Nosotros</li>
  <li>Contáctenos</li>
</ul>
```

- Inicio
- Nosotros
- Contáctenos

# Listas y listas anidadas

## Listas no ordenadas (Unordered List)

Un atributo de estilo puede ser añadido a una lista no ordenada, para definir el estilo del marcador:

Style	Descripción
list-style-type:disc	Los elementos de la lista se marcará con círculos negros
list-style-type:circle	Los elementos de la lista serán marcados con círculos
list-style-type:square	los elementos de la lista estará marcado con cuadrados
list-style-type:none	Los elementos de la lista no se marcarán

# Listas y listas anidadas

## Listas ordenadas (Ordered List)

Una lista ordenada comienza con la etiqueta `<ol>`. Cada elemento de la lista comienza con la etiqueta `<li>`. Los elementos de la lista serán **marcados con números**.

```
<ol>  
  <li>Elemento 1</li>  
  <li>Elemento 2</li>  
  <li>Elemento 3</li>  
</ol>
```

1. Elemento 1
2. Elemento 2
3. Elemento 3

# Listas y listas anidadas

## Listas ordenadas (Atributo Type)

Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

# Listas y listas anidadas

## Listas de descripción (Description List)

Una lista de descripción comienza con la etiqueta `<dl>`. Cada termino (name) de la lista utiliza la etiqueta `<dt>` y cada descripción del termino hace uso de la etiqueta `<dd>`.

```
<dl>
  <dt>Coffee</dt>
  <dd>- black hot drink</dd>
  <dt>Milk</dt>
  <dd>- white cold drink</dd>
</dl>
```

Coffee  
- black hot drink

Milk  
- white cold drink

# Listas y listas anidadas

## Listas anidadas

Una lista puede contener una o más listas (anidadas)

```
<ul>
  <li>Café</li>
  <li>Té
    <ul>
      <li>Té negro</li>
      <li>Té Verde</li>
    </ul>
  </li>
  <li>Leche</li>
</ul>
```

- Café
- Té
  - Té negro
  - Té Verde
- Leche

# Tablas

Una tabla no es otra cosa más que un medio de organizar datos en filas y columnas. Este concepto ha estado presente en nuestra sociedad por un largo período de tiempo y ha sido adoptado por HTML en sus etapas iniciales, como una forma de transmitir información que, de otro modo, no sería comprendida tan fácilmente.

Como muchas otras estructuras de HTML, las tablas son construidas utilizando elementos. En particular, una tabla básica puede ser declarada usando tres elementos:

- table (el contenedor principal)
- tr (representando a las filas contenedoras de las celdas)
- td (representando a las celdas).

```
<table>
  <tr>
    <td>Celda 1</td>
    <td>Celda 2</td>
    <td>Celda 3</td>
  </tr>
  <tr>
    <td>Celda 4</td>
    <td>Celda 5</td>
    <td>Celda 6</td>
  </tr>
</table>
```



# Tablas

## Etiquetas relacionadas a Table

Tag	Description
<u>&lt;table&gt;</u>	Defines a table
<u>&lt;th&gt;</u>	Defines a header cell in a table
<u>&lt;tr&gt;</u>	Defines a row in a table
<u>&lt;td&gt;</u>	Defines a cell in a table
<u>&lt;thead&gt;</u>	Groups the header content in a table
<u>&lt;tbody&gt;</u>	Groups the body content in a table
<u>&lt;tfoot&gt;</u>	Groups the footer content in a table

# Tablas

## Rowspan y Colspan

No	ID	Nombre completo	
		Apellido	Nombre

# Tablas

## Rowspan y Colspan

2022		
FIESTA		

# Tablas

## Rowspan y Colspan

APRIL		

# Formularios:

## Form, input, select, radio, button, submit

### Formulario <form>

Un formulario es un conjunto de controles (botones, cajas de texto, casillas de verificación, botones radio, etc) que permiten al usuario introducir datos y enviarlos al servidor web para su procesamiento.

Todos los formularios comienzan con un elemento **<form>** y cierra con **</form>**

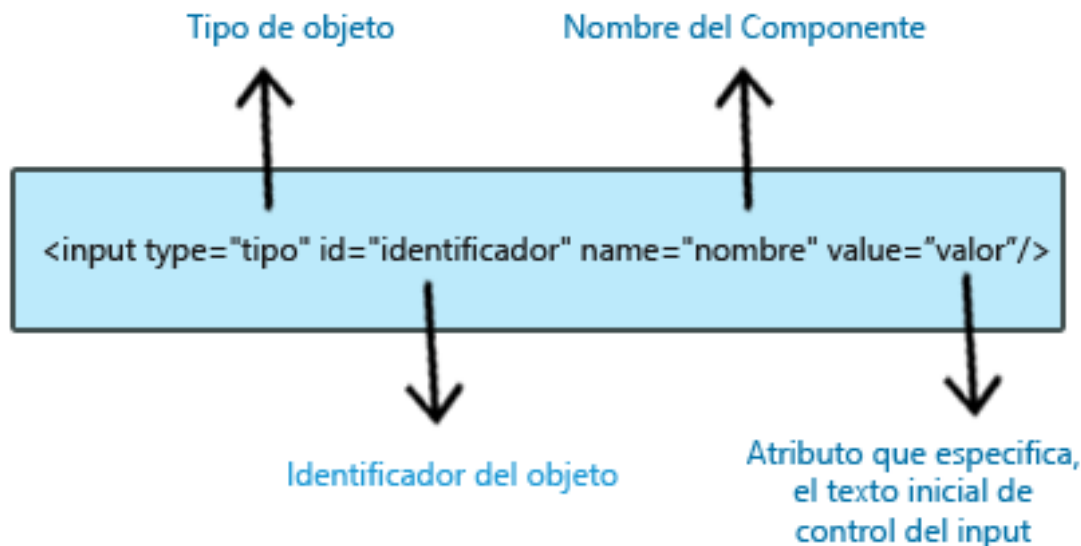
```
<form action="#" method="post">  
  
</form>
```

# Formularios:

## Form, input, select, radio, button, submit

### Elemento `<input>`

El elemento HTML `<input>` se usa para crear controles interactivos para formularios basados en la web con el fin de recibir datos del usuario. El tipo de control mostrado depende del atributo **"type"**.



# Formularios:

## Form, input, select, radio, button, submit

### Elemento <input>

**TYPE:** determina el tipo de entrada que se va a utilizar y sus valores pueden ser uno de los siguientes:

**1.Text**, Cuando este valor es establecido, un control de entrada de texto es mostrado. Este tipo de control solo puede recibir información de tipo texto.

```
<input type="text" id="nombre" name="nombre"/>
```

Nombre

# Formularios:

## Form, input, select, radio, button, submit

### Elemento <input>

**2.Password**, muestra una caja donde se tecleará texto cuyo eco será sustituido por asteriscos.

```
<input type="password" id="pass" name="pass"/>
```

Contraseña



# Formularios:

## Form, input, select, radio, button, submit

Elemento <input>

**3.Checkbox**, un casilla que sólo puede tener dos estados, {on, off}

```
<input type="checkbox" id="transporte" name="transporte"/>
```

Puestos de trabajo buscados

- ☐ Dirección
- ☐ Técnico
- ☐ Empleado

# Formularios:

## Form, input, select, radio, button, submit

Elemento <input>

### 3.Checkbox

```
<input type="checkbox" id="direccion" name="direccion" />Dirección  
<input type="checkbox" id="transporte" name="transporte" />Transporte  
<input type="checkbox" id="empleado" name="empleado" />Empleado
```

☐ Dirección ☐ Transporte ☐ Empleado

# Formularios:

## Form, input, select, radio, button, submit

### Elemento <input>

**4.Radio**, Los botones radio trabajan de forma muy similar a las casillas de verificación, permitiendo a los usuarios elegir entre dos valores posibles: marcado y no marcado.

```
<input type="radio" id="direccion" name="direccion" />Dirección  
<input type="radio" id="transporte" name="transporte" />Transporte  
<input type="radio" id="empleado" name="empleado" />Empleado
```

☐ Dirección ☐ Transporte ☐ Empleado

# Formularios:

## Form, input, select, radio, button, submit

### Elemento <input>

**5.Submit**, un botón que produce que el formulario actual sea enviado al URL especificado en el atributo **ACTION** del formulario.

```
<input type="submit" id="submit" value="Enviar" />
```

# Formularios:

## Form, input, select, radio, button, submit

### Elemento <input>

**6.Button:** este tipo de control crea un botón regular (que luce igual a los botones *submit*) que, a diferencia del **submit**, no tiene ninguna acción asociada predeterminada. Si no se le asigna una acción manualmente, nada sucederá cuando sea presionado..

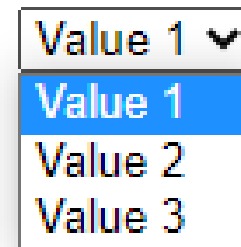
```
<input type="button" id="submit" value="Enviar" />
```

# Formularios:

## Form, input, select, radio, button, submit

**7.Select:** El elemento select (**<select>**) de HTML representa un control que muestra un menú de opciones. Las opciones contenidas en el menú son representadas por elementos **<option>**

```
<select name="select">  
  <option value="value1">Value 1</option>  
  <option value="value2">Value 2</option>  
  <option value="value3">Value 3</option>  
</select>
```



# Elemento DIV

El elemento **div** es un contenedor genérico sin un significado semántico en particular.

Se utiliza comúnmente en el desarrollo de documentos con propósitos estilísticos, en conjunto con los atributos **style** y **class**.

```
<div>
  <h2>Ejemplo de div</h2>
  <p>
    Esto es un párrafo dentro de un div.
  </p>
</div>
```

## Ejemplo de div

Esto es un párrafo dentro de un div.

# Elemento DIV

```
<div id="header">
```

```
<div id="nav">
```

```
<div id="main">
```

```
<div id="aside">
```

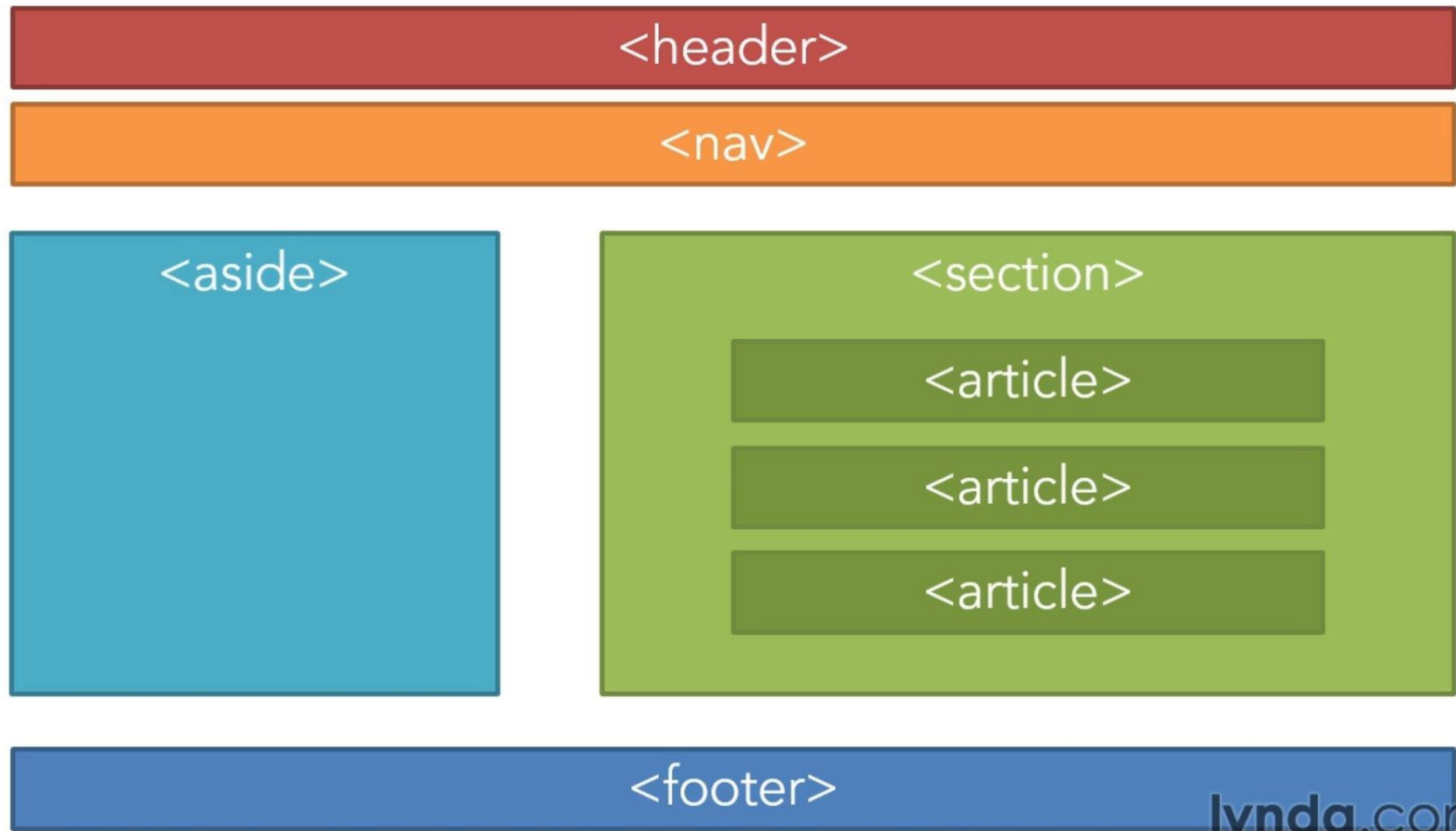
```
<div id="article">
```

```
<div id="article">
```

```
<div id="footer">
```



# HTML5



# ¡Feliz Noche a todos!



kodigo