

CONSTRUCCIÓN DE PÁGINAS WEB

HTML



CSS



JAVASCRIPT



mardeasa

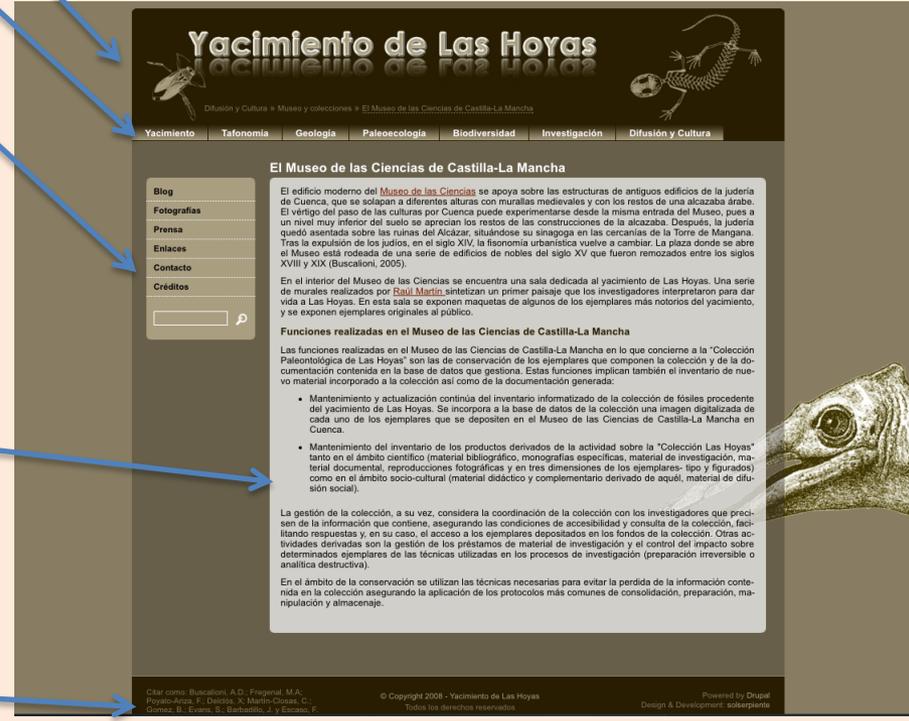
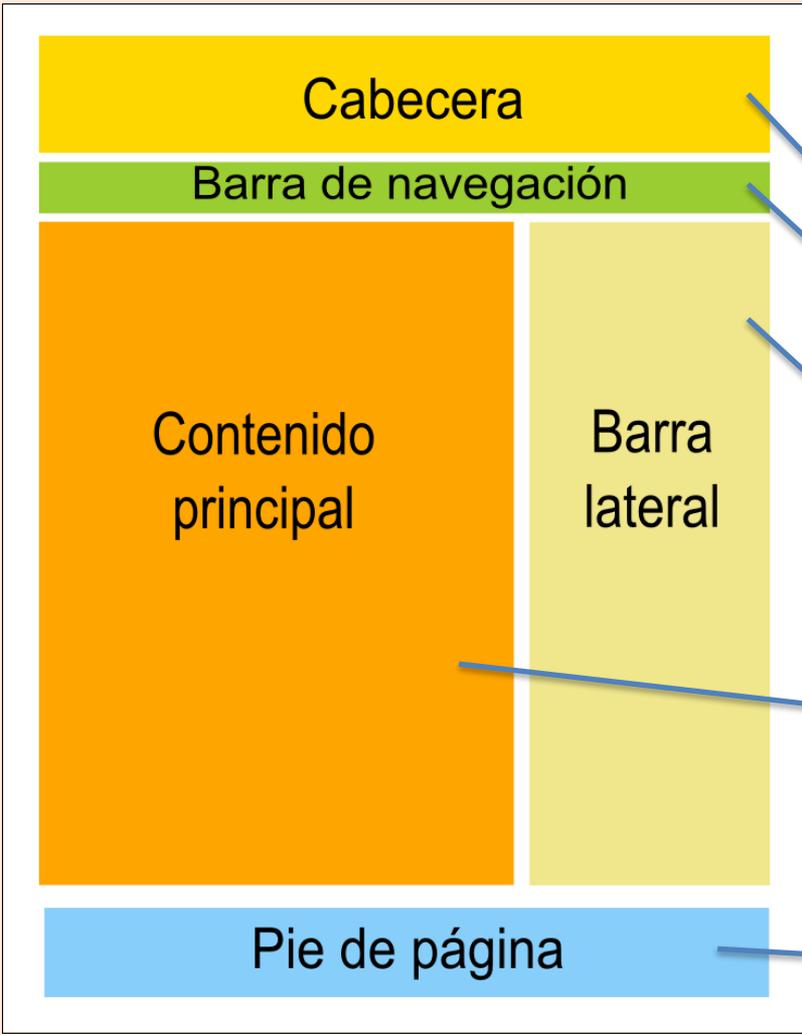


CREACIÓN DE PÁGINAS WEB CON EL LENGUAJE DE MARCAS

Estructura de una página web



Un problema en HTML es la colocación de los diferentes elementos. En este esquema podemos ver una estructura simple de una página web.



Estructura del cuerpo

```
<header><h1></h1></header>
```

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

```
<section>  
  <h2></h2>  
  <article>  
    <h3></h3>  
  </article>  
  <article></article>  
</section>
```

```
<aside>  
</aside>
```

```
<footer></footer>
```

Representación visual de un diseño utilizando elementos HTML5

HTML5 ofrece nuevos elementos semánticos para definir la estructura de la web, con ello ayudaremos a los **motores de búsqueda** a comprender mejor la estructura de nuestra web haciéndola más visible de cara a los buscadores, más accesible y más sostenible.

Estructura del cuerpo. <header>

<header>

Uno de los nuevos elementos incorporados en HTML5 es <header>. No debe ser confundido con <head> usado antes para construir la cabecera del documento. La intención de <header> es proveer información introductoria (títulos, subtítulos, logos), y es usado solo para el cuerpo o secciones específicas dentro del cuerpo:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <title>Este texto es el título del documento</title>
</head>
<body>
  <header>
    <h1>Este es el título principal del sitio web</h1>
  </header>
</body>
</html>
```

Estructura del cuerpo. <nav>

<nav>

La siguiente sección es la Barra de Navegación. Esta barra es generada con el elemento <nav>:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <title>Este texto es el título del documento</title>
</head>
<body>
  <header>
    <h1>Este es el título principal del sitio web</h1>
  </header>
  <nav>
    <ul>
      <li>principal</li>
      <li>fotos</li>
      <li>videos</li>
      <li>contacto</li>
    </ul>
  </nav>
</body>
</html>
```

El elemento <nav> se encuentra dentro de las etiquetas <body> pero es ubicado después de la etiqueta de cierre de la cabecera (</header>), porque <nav> no es parte de la cabecera sino una nueva sección.

Las etiquetas son usadas para definir cada ítem de la lista.

Estructura del cuerpo. <main>

<main>

Identifica el contenido principal de la página y sólo puede haber uno por cada página.

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <title>Este texto es el título del documento</title>
</head>
<body>
  <header>
    <h1>Este es el título principal del sitio web</h1>
  </header>
  <nav>
    <ul>
      <li>principal</li>
      <li>fotos</li>
      <li>videos</li>
      <li>contacto</li>
    </ul>
  </nav>
  <main>
  </main>
</body>
</html>
```

El elemento <main> se encuentra dentro de las etiquetas <body> y hace referencia al contenido principal de una página web. No es absolutamente necesario ponerlo, pero si lo utilizamos, únicamente debe haber uno por página. Este elemento puede contener a <section> y <article>.

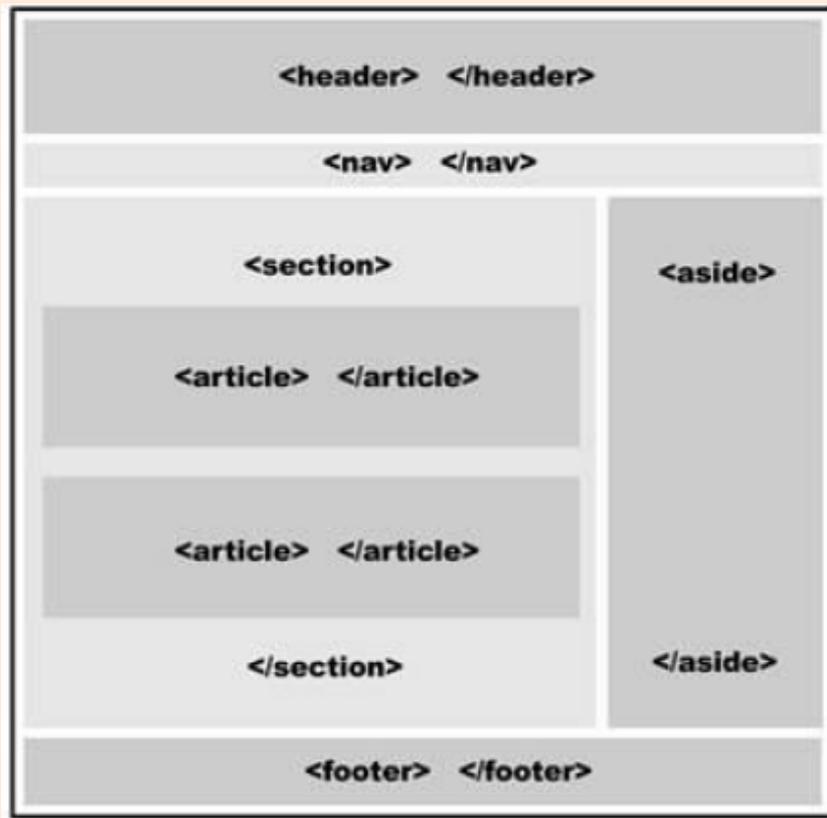
Estructura del cuerpo. <section>

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <title>Este texto es el título del documento</title>
</head>
<body>
  <header>
    <h1>Este es el título principal del sitio
web</h1>
  </header>
  <nav>
    <ul>
      <li>principal</li>
      <li>fotos</li>
      <li>videos</li>
      <li>contacto</li>
    </ul>
  </nav>
  <section>
</section>
</body>
</html>
```

<section>

La columna Contenido Principal contiene la información más relevante del documento y puede estar dividida en varios bloques o columnas. Debido a que el propósito de estas columnas es más general, el elemento que especifica estas secciones se llama <section>.

Estructura del cuerpo. <article>



<article>

Las etiquetas `<article>` se encuentran ubicadas dentro del elemento `<section>`. Las etiquetas `<article>` en nuestro ejemplo pertenecen a esta sección, son sus hijos, del mismo modo que cada elemento dentro de las etiquetas `<body>` es hijo del cuerpo. Y al igual que cada elemento hijo del cuerpo, las etiquetas `<article>` son ubicadas una sobre otra.

Estructura del cuerpo. <aside>

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <title>Este texto es el título del documento</title>
</head>
<body>
  <header>
    <h1>Este es el título principal del sitio
web</h1>
  </header>
  <nav>
    <ul>
      <li>principal</li>
      <li>fotos</li>
      <li>videos</li>
      <li>contacto</li>
    </ul>
  </nav>
  <section>
  </section>
  <aside>
    <blockquote>Mensaje número
1</blockquote>
    <blockquote>Mensaje número
2</blockquote>
  </aside>
</body>
</html>
```

<aside>

En un típico diseño la columna llamada Barra Lateral se ubica al lado de la columna Contenido 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. El elemento <aside> podría estar ubicado al lado derecho o izquierdo de nuestra página, la etiqueta no tiene una posición predefinida sólo describe la información que contiene, no el lugar dentro de la estructura.

Estructura del cuerpo. <footer>

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8" />
  <title>Este texto es el título del documento</title>
</head>
<body>
  <header>
    <h1>Este es el título principal del sitio
web</h1>
  </header>
  <nav>
    <ul>
      <li>principal</li>
      <li>fotos</li>
      <li>videos</li>
      <li>contacto</li>
    </ul>
  </nav>
  <section>
  </section>
  <aside>
    <blockquote>Mensaje número 1</blockquote>
    <blockquote>Mensaje número 2</blockquote>
  </aside>
  <footer>
    Derechos Reservados &copy; 2016-2017
  </footer>
</body>
</html>
```

<footer>

Para finalizar la construcción de la plantilla o estructura elemental de nuestro documento, sólo necesitamos un elemento más para cerrar nuestro diseño y otorgarle un final al cuerpo del documento. El elemento específico para este propósito se denomina <footer>.

<footer> puede ser usado múltiples veces dentro del cuerpo para representar también el final de diferentes secciones (del mismo modo que la etiqueta <header>).

Role en HTML5

Role es un atributo XHTML que nos permite pasar información a los navegadores acerca del propósito de ese elemento, permitiendo escanear rápidamente el contenido de nuestra web, haciéndola más accesible.

Resumiendo: nos proporcionará ventajas de accesibilidad, adaptación de dispositivos, procesos del lado servidor y de comprensión de descripción de datos.

A continuación resumimos los mas importantes:

```
<header role="banner"></header>  
<nav role="navigation"></nav>  
<main role="main"></main>  
<section role="main">  
<article role="article"></article>  
<aside role="complementary"></aside>  
<footer role="conteninfo"> </footer>
```

<div>

La etiqueta <div> se emplea para definir un bloque de contenido o sección de la página, para poder aplicarle diferentes estilos e incluso para realizar operaciones sobre ese bloque específico.

Visualmente la etiqueta no provoca ningún cambio, pero en la estructura interna del documento hemos aplicado una división.

```
<div>
  <h1>Índice</h1>
  Página principal<br />
  Material multimedia<br />
  Autores<br />
</div>
```



<div> id y class

Adicionalmente, la etiqueta div se suele acompañar de uno o dos parámetros destinados a identificar ese bloque:

- **id**: permite establecer un identificador único para el bloque. Así podremos referirnos al bloque de forma inequívoca.
- **class**: es similar a id, pero con la ventaja de que se puede repetir; así que podemos tener varios <div> diferentes con la misma clase.

Usando id y class podemos diferenciar unos div de otros y así aplicarles estilos CSS diferentes o hacer que actúen de forma distinta.

Como se puede apreciar, las etiquetas id y class se pueden mezclar. Se mantiene siempre esa norma: los id son únicos dentro de una página, mientras que los class se suelen repetir para que varios elementos tomen un mismo estilo.

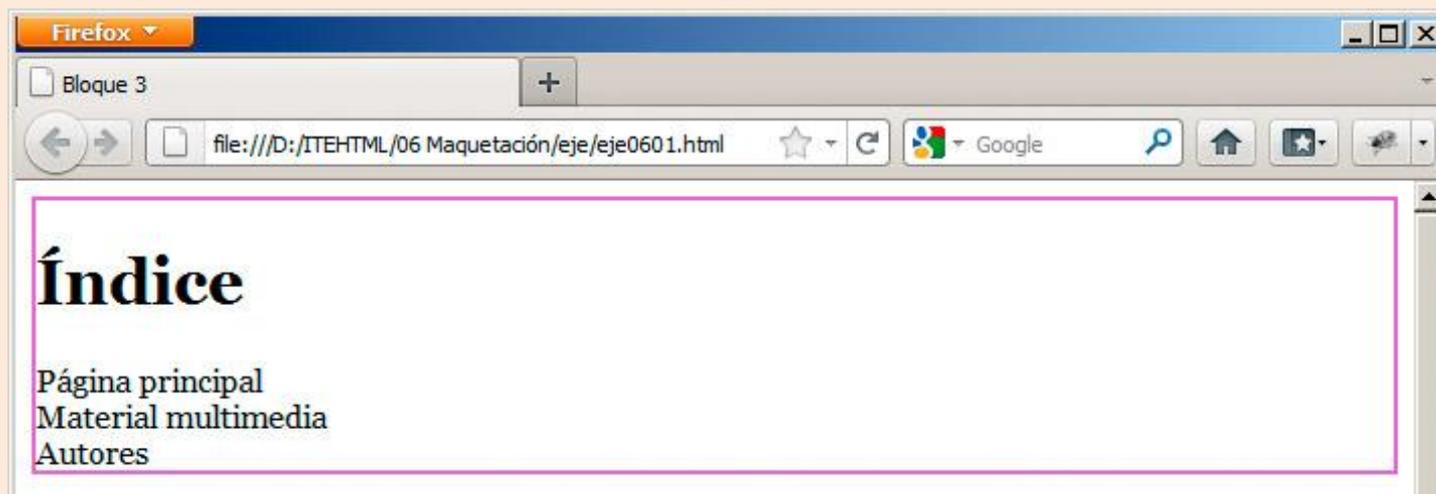
```
<div id="indice">
  <br>índice
</div>
<div class="destacado">
  Este texto se destaca mediante un estilo.
</div>
<div id="articulo15" class="articulos_estandar">
  Texto de un artículo
</div>
```

<div> style

Probemos ahora a aplicar una modificación a la apariencia de ese bloque, añadiendo un estilo de borde a la etiqueta <div>; quedaría así:

```
<div style="border: 2px solid rgb(204, 102, 204);">  
<h1>Índice</h1>  
Página principal<br />  
Material multimedia<br />  
Autores<br />  
</div>
```

Estos estilos, vienen definidos por un nuevo lenguaje de programación, denominado CSS, que veremos más adelante.

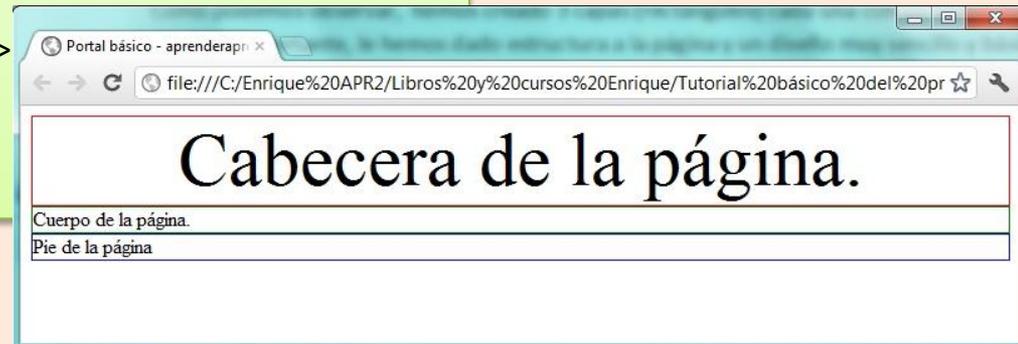


<div>

Por tanto, la principal utilidad de las capas, es la de poder separar por partes la página HTML (estructurar). Tener bien estructurado un documento HTML (página web) es importante por distintos motivos, por ejemplo:

1. Nos permitirá establecer un estilo diferente (diseño) para cada contenedor.
2. Nos permitirá identificar cuándo un usuario hace click sobre la página, en qué contenedor ha hecho click.
3. Nos permitirá crear efectos especiales para una capa concreta.

```
<body>
  <div style="text-align: center; font-size: 60px; border: 1px solid
red;">
    Cabecera de la página.
  </div>
  <div style="border: 1px solid green;">
    Cuerpo de la página.
  </div>
  <div style="border: 1px solid blue;">
    Pie de la página
  </div>
</body>
```



Aplicando estilos a <div>

HTML

```
<body>
<h1>Simulación de frames 1</h1>

<div id="indice">
<ul>
  <li><a href="#apartado-1">Apartado 1</a></li>
  <li><a href="#apartado-2">Apartado 2</a></li>
</ul>
</div>

<div id="contenido">
<div id="apartado-1">
<h2>Apartado 1</h2>
<p>Este es el apartado 1</p>
</div>

<div id="apartado-2">
<h2>Apartado 2</h2>
<p>Este es el apartado 2</p>
</div>
</div>
</body>
```

CSS

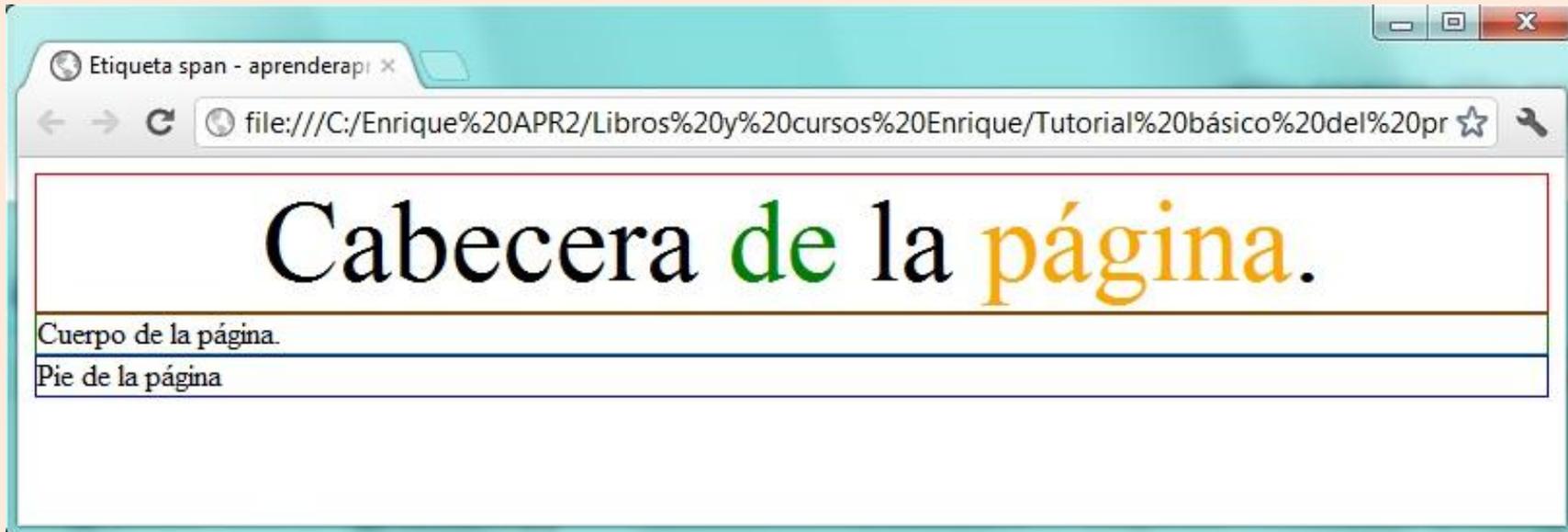
```
<style type="text/css">
div#indice {
  background-color: lightblue;
  height: 70%;
  left: 5%;
  position: absolute;
  top: 20%;
  width: 15%;
}
div#contenido {
  background-color: orange;
  height: 70%;
  left: 25%;
  overflow: hidden;
  position: absolute;
  top: 20%;
  width: 70%;
}
div div {
  height: 100%;
  width: 100%;
}
```


La etiqueta span al igual que casi todas las etiquetas en HTML tiene su etiqueta de apertura y su respectiva etiqueta de cierre .

Sirve normalmente para crear subdivisiones dentro de una etiqueta en la misma línea. Esto nos permite aplicar estilos diferentes a cada subdivisión sin necesidad de crear nuevas etiquetas.

Imaginemos que en el ejemplo anteriormente mostrado, dentro de la cabecera cuya fuente es de color negro, quisiéramos añadir texto de distintos colores pero que ese texto perteneciera a la misma línea de la cabecera. Para ello, utilizaríamos la etiqueta span.

```
<body>
  <div style="text-align: center; font-size: 60px; border: 1px solid red;">
    Cabecera
    <span style="color: green;">de</span> la <span style="color: orange;">página</span>.
  </div>
  <div style="border: 1px solid green;">
    Cuerpo de la página.
  </div>
  <div style="border: 1px solid blue;">
    Pie de la página
  </div>
</body>
```

Como podemos observar, la etiqueta span permite crear subdivisiones que nos servirán para poder identificar y aplicar estilos a porciones de texto dentro de una misma línea.



CREACIÓN DE PÁGINAS WEB CON EL LENGUAJE DE MARCAS

Marco flotante

Técnica del Marco Flotante

Un marco flotante es una zona de una ventana de navegador que puede mostrar un documento HTML independiente de lo que se muestra en el resto de la ventana. Los marcos permiten dividir la ventana de un navegador en varias regiones, cada una las cuales puede mostrar un documento HTML diferente. Se suele utilizar para vincular y mostrar diferentes páginas web a través de la navegación, en una misma posición, sin tener que duplicar estructuras.

Para utilizarlo, en primer lugar lo definimos, pudiéndolo ubicar dentro de una `<div>`:

```
<div id="contenido">  
  <iframe name="flotante" width="900" height="720" src="pagina000.html" frameborder="0">  
  </iframe>  
</div>
```

Y posteriormente, haremos una referencia en el vínculo, mediante el atributo `target` como destino "flotante" o el nombre que hayamos utilizado al definirlo.

```
<a href="pagina000.html" target="flotante">
```

Técnica del Marco Flotante

Los márgenes de las páginas vinculadas mediante marco flotantes deben estar a cero, ya que en caso contrario, nos pueden generar barras de desplazamiento horizontales y/o verticales.

Además, las páginas, deben estar alineadas a la izquierda.

Se desaconseja el uso de marcos para el diseño de páginas Web.

Estas son algunas de las desventajas que presenta el uso de marcos:

- Lograr una alineación gráfica precisa de los elementos en distintos marcos puede resultar difícil.
- Comprobar las opciones de navegación puede llevar mucho tiempo.
- Los URL de las páginas con marcos no aparecen en el navegador, por lo que puede resultar complicado para un visitante marcar una página concreta.

Insertar objetos <object>

La etiqueta `<object>` permite insertar cualquier tipo de documentos (no solamente de texto) dentro de un documento html. Para poder mostrar esos archivos, el navegador suele necesitar plug-ins.

Insertar una página web

Se puede insertar una página web dentro de otra página web mediante la etiqueta `<object>`, como muestra el siguiente ejemplo.

```
<object
  type="text/html"
  data="ejemplo_object.html"
  style="width: 400px; height:200px;">
  ERROR (no puede mostrarse el objeto)
</object>
```

Insertar objetos <object>

Insertar un archivo flash SWF

Se puede insertar un archivo flash SWF en cualquier página web, como muestra el siguiente ejemplo.

En este caso tendríamos que utilizar la etiqueta <param> para introducir los parámetros necesarios.

```
<object
  type="application/x-shockwave-flash"
  data="ejemplo_flash.swf"
  style="width: 200px; height: 50px;">
  <param
    name="movie"
    value="ejemplo_flash.swf" />
  ERROR (no puede mostrarse el objeto)
</object>
```

Insertar objetos <object>

Insertar un vídeo de YouTube

Los vídeos de YouTube pueden obtenerse en formato Flash, por lo que se pueden insertar en una página web mediante la etiqueta <object>, como muestra el siguiente ejemplo. Para insertar otro vídeo habría que sustituir el código del vídeo en el atributo data (en el ejemplo, el código del vídeo es vZV-t3KzTpw).

```
<object
  type="application/x-shockwave-flash"
  data="https://www.youtube.com/v/vZV-t3KzTpw"
  style="width: 425px; height:350px;" >
  <param
    name="movie"
    value="https://www.youtube.com/v/vZV-t3KzTpw" />
  ERROR (no puede mostrarse el objeto)
</object>
```

Insertar objetos <object>

Insertar un archivo PDF

Se puede insertar un archivo PDF en cualquier página web, como muestra el siguiente ejemplo. Para insertar otro archivo PDF habría que indicar la URI del archivo mediante el atributo data (en el ejemplo, es simplemente ejemplo.pdf).

```
<object
  type="application/pdf"
  data="ejemplo.pdf"
  style="width: 400px; height: 550px;">
  ERROR (no puede mostrarse el objeto)
</object>
```

VIDEO EN HTML5

<video>

La etiqueta <video> tiene un peso muy fuerte en HTML5 por que viene a facilitarnos las cosas en cuanto a la necesidad de agregar contenido audio visual en nuestra pagina Web. Agregar video en HTML5 no tiene mucha complejidad, es muy fácil y rápido de escribir.

```
<video src="video.mp4" width="640" height="360" controls autoplay preload>
</video>
```

- **SRC:** Nos enlaza el archivo de video que queremos reproducir.
- **WIDTH:** Nos define el ancho del video en pixeles.
- **HEIGHT:** Nos define la altura del video en pixeles.
- **CONTROLS:** Nos permite implementar los controles del reproductor por defecto del navegador como, botón play-pause, seek y volumen.
- **AUTOPLAY:** Nos permite reproducir el archivo de video desde que se carga la pagina.
- **PRELOAD:** Nos carga un poco el archivo de video antes de iniciar la reproducción en el buffer para que no se trabaje por reproducir mas de lo que carga.

VIDEO EN HTML5

Implementación de formatos de video

Tenemos muchos formatos de video que debemos implementar ya que cada motor de renderizado de los navegadores tiene soporte para un codec de video diferente.

Veamos la tabla de compatibilidad

- Gecko: OGG
- WebKit: OGG, H.264, WebM
- Safari e Internet Explorer 9 (WebKit – Trident): H.264
- Presto: OGG, H.264

Hay que hacer que todos los navegadores acepten en la etiqueta video todos los formatos de video.

Sólo hay que poner los diferentes SOURCE para cada formato de video.

VIDEO EN HTML5

```
<video width="640" height="360" controls autoplay preload>  
  
    <source src="mivideo.mp4" type='video/mp4; codecs="avc1,mp4a"' />  
    <source src="mivideo.ogv" type='video/ogg; codecs="theora,vorbis"' />  
    <source src="mivideo.webm" type='video/webm; codecs="vp8,vorbis"' />  
  
</video>
```

La diferencia es poca, ahora solo quitamos el atributo SRC y agregamos los SOURCE de los diferentes archivos de video.

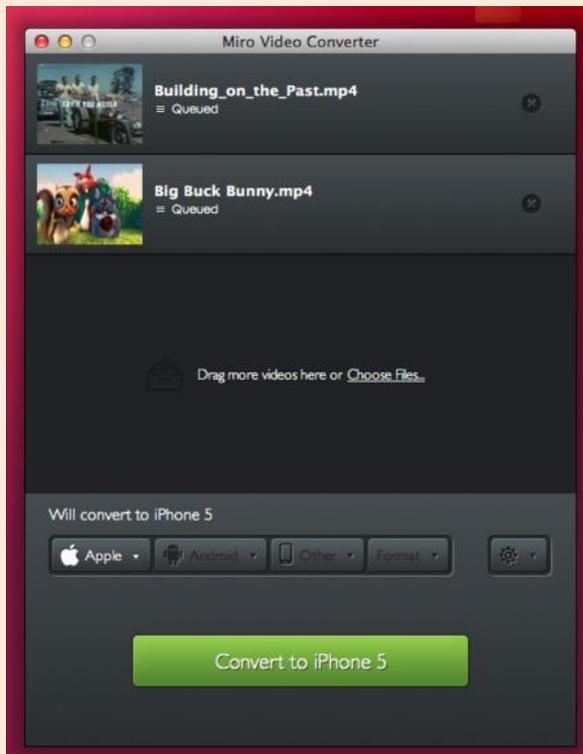
SRC: Nos enlaza el archivo de video que queremos reproducir.

Type: Le decimos que tipo de video o codec va reproducir

Esta etiqueta es bastante fácil de usar y podemos jugar mucho con los estilos de CSS para darle estilo, y si practicamos mucho podemos empezar a crear un reproductor más personalizado.

VIDEO EN HTML5

Y algunas herramientas para convertir videos a otros formatos:



[Miro Video Converter](#)
[Zencoder](#)
[Encoding](#)

AUDIO EN HTML5

<audio>

Una de las principales cosas que llama la atención cuando hablamos de HTML5, son las famosas etiquetas <audio> y <video>. Estas etiquetas tienen su gran fama por que nos permiten consumir multimedia sin necesidad de instalar plugins adicionales al navegador en nuestro ordenador.

```
<audio width="300" height="32" src="micancion.mp3" controls autoplay preload>
</audio>
```

- **SRC:** Nos enlaza el archivo de audio que queremos reproducir.
- **CONTROLS:** Nos permite implementar los controles del reproductor por defecto del navegador como, botón play-pause, seek y volumen.
- **AUTOPLAY:** Nos permite reproducir el archivo de audio desde que se carga la pagina.
- **PRELOAD:** Nos carga un poco el archivo de audio antes de iniciar la reproducción en el buffer para que no se trabe por reproducir mas de lo que carga.

AUDIO EN HTML5

Implementación de formatos de audio

Tenemos muchos navegadores con diferente compatibilidad, pero lo que más nos interesa son los Motores de Renderizado, estos se encargan de renderizar el código de nuestra pagina web e implementar el contenido multimedia.

Veamos qué motores corresponden a cada navegador:

- Google Chrome: WebKit
- Safari: WebKit
- Mozilla FireFox: Gecko
- Internet Explorer: Trident
- Opera: Presto

	WebKit	Gecko	Trident (IE9)	Presto
Mp3	*		*	
Ogg	*(Safari no)	*		*
Wav	*(Safari si)	*	*	*

AUDIO EN HTML5

Ahora que se tenemos conocimiento de que motores o navegadores aceptan un formato, pasamos a hacer compatible AUDIO con todos ellos.

Es algo diferente de implementar ahora, por que hay que decirle los diferentes archivos que existen para que el navegador tome alguno de los que puede reproducir.

```
<audio controls autoplay preload>  
  <source src="cancion.ogg" type="audio/ogg" />  
  <source src="cancion.mp3" type="audio/mpeg" />  
  <source src="cancion.wav" type="audio/wav" />  
  Tu navegador no soporta esta característica  
</audio>
```

AUDIO EN HTML5

Y algunas herramientas para convertir audios a otros formatos:

http://www.freemake.com/es/free_audio_converter/
<https://www.online-convert.com/es>
<http://online-audio-converter.com/es/>



Colocar un video de Youtube

YouTube ES

Buscar

13:03 / 14:07

LOS MEJORES DESCUIDOS en TELEVISION Metidas De Pata Mas Vergonzosas # 44 Momentos Graciosos TV

Hanny Videos Funny

Suscribirse 144 K

Descargar 720

Descargar 1080

726.356 visualizaciones

Añadir a

Compartir

Más

2.080

521

Compartir

Insertar

Enviar por correo

```
cc="https://www.youtube.com/embed/99swGaNw35w?rel=0&amp;showinfo=0" frameborder="0" gesture="media" allowfullscreen></iframe-
```

MOSTRAR MÁS



Colocar un mapa de GOOGLEMAPS

