



2. Utilización de lenguajes de marcas en entornos web.

| | |
|-------|----------------------|
| Autor | Xerach Casanova |
| Clase | Lenguajes de Marcas |
| Fecha | @Dec 7, 2020 3:08 PM |

1. Introducción a la utilización de lenguajes de marcas en entornos web

1.1. HTML: evolución y versiones

1.2. HTML y XHTML

Diferencias sintácticas y estructurales con HTML

1.3. Estructura de un documento HTML

Prólogo

Ejemplar

1.4. Identificación de etiquetas y atributos HTML

1.5. Clasificación de los atributos comunes según su funcionalidad

Atributos básicos o globales.

Atributos para internacionalización.

Atributos de eventos y atributos para elementos que pueden obtener foco

2. Elementos HTML

2.1. Elementos de estructura básica de un documento

2.2. Elementos de la sección cabecera

Elementos contenedores

Elementos no contenedores

2.3. Encabezados y párrafos

2.3.1. Saltos de línea y espacios en blanco

2.3.2. Comentarios

2.4. Semántica a nivel texto

2.5. Elementos de listas

2.6. Elementos de tablas

2.7. Elementos de formularios

2.7.1. Declaración de formulario

2.7.2. Campos de formulario - input

2.7.3. Campos de formulario - Área de texto

2.7.4. Campos de formularios - Lista desplegable

2.7.5. Campos de formularios - checkbox

2.7.6. Botones de formulario

2.7.7.- Otros campos de formulario

2.7.8. Campos de formulario con formato de fecha

2.7.9. Campos de formulario - Rangos

2.7.10. Organización de formularios

2.8. Multimedia

2.9. Secciones y etiquetas semánticas

2.10. Elemento iframe

3. Hojas de estilo o CSS

3.1. Cómo incluir CSS en un documento HTML o XHTML

3.1.1. Elemento span

3.2. Sintaxis de las reglas de estilo

3.3. Cascada y herencia de estilos

3.4. Selectores

3.4.1. Selectores de clase

3.4.2. Selectores de ID

3.4.3. Selectores descendientes

3.4.4. Selector hijo

3.5. Propiedades principales

3.5.1. Propiedades de color y fondo

| |
|----------------------------------|
| 3.5.2. Propiedades de fuente |
| 3.5.2. Propiedades de texto |
| 3.5.3. Propiedades de lista |
| 3.5.5. Propiedad display |
| 3.6. Avanzado: Modelo de cajas |
| 3.6.1. Propiedades de caja |
| 3.6.2. Unidades de tamaño |
| 3.6.3. Ejemplos del modelo cajas |
| 3.6.4. Posicionamiento |
| Mapa Conceptual |

1. Introducción a la utilización de lenguajes de marcas en entornos web

1.1. HTML: evolución y versiones

HTML es un estándar reconocido por todos los navegadores, los cuales visualizan una página HTML de forma muy similar, independientemente del SO.

El origen fue un sistema de hipertexto para compartir documentos electrónicos en 1990. Las dos primeras propuestas de estándar HTML llegaron a ser oficiales (HTML y HTML+).

- HTML 2.0 fue la primera versión oficial de HTML. IETF (Internet Engineering Task Force) publicó el estándar en 09/1995.
- HTML 3.2 el 14/01/1997 W3C se incorporan los applets de java y texto alrededor de imágenes.
- HTML 4.0 se publica el 24/04/1998 se incorporan las hojas de estilo CSS y posibilidad de incluir pequeños programas.
- HTML 4.01 el 24/12/1999 se actualiza la versión anterior y W3C no actualiza hasta marzo de 2007 debido a la presión de WHATWG (Web Hypertext Application Technology Working Group).
- HTML 5 es el estándar actual y la versión más avanzada, la cual ha ido evolucionando (5.1, 5.2 y 5.3).

El cambio más significativo de HTML 5 es el paso a CSS de todo lo relativo a la presentación del documento. HTML solo se encarga de la estructura, información y semántica.

1.2. HTML y XHTML

XHTML (Extensible HyperText Markup Language) es similar a HTML, siendo una adaptación a XML.

HTML 4.01 es incluido en XHTML 1.0 debido a que solo añade unas pocas modificaciones.

XHTML soluciona los problemas del desorden y permisividad de HTML añadiendo normas en la forma de escribir etiquetas y atributos. Permite:

- Sencillez a la hora de editar y mantener el código.
- Al ser regular, es más fácil escribir código que lo procese.
- Permite utilizar herramientas creadas para trabajar XML genéricos, editores XSLT, etc...

Diferencias sintácticas y estructurales con HTML

El HTML se considera XML bien formado cuando:

- Toda la página está contenida en el elemento raíz <html>
- Los nombres de etiquetas y atributos están siempre en minúsculas.
- El valor de los atributos entre comillas (incluso numéricos).
- En atributos que el nombre coincide su valor, no se da el valor por entendido (no se pueden comprimir).
- Las etiquetas se cierran siempre. En etiquetas vacías se utiliza la etiqueta que se abre y se cierra de una sola vez (ejemplo
).

Los diseñadores web suelen trabajar con HTML y los desarrolladores con XHTML, pero los tres primeros requisitos son buena práctica y se deben cumplir.

En errores de sintaxis, HTML muestra la mayor parte del contenido posible, pero XHTML puede dar errores de sintaxis que no muestren documento.

1.3. Estructura de un documento HTML

Igual que en XML, tiene prólogo y ejemplar.

Prólogo

Le indica al navegador el documento que se va a iniciar y la versión de HTML. Para la versión 4 hay tres tipos:

- HTML 4.0 Strict. Es la DTD utilizada por defecto. No permite uso de elementos declarados "deprecated" en otras versiones o recomendaciones HTML.

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

- HTML 4.0 Transitional. Permite el uso de los elementos deprecated.

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

- HTML 4.0 Frameset. Una variante para documentos que usan frames, en los cuales se reemplaza la etiqueta body por frameset.

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

- HTML5 solo utiliza una declaración

```
<!DOCTYPE html>
```

Ejemplar

Un documento HTML está dentro de las etiquetas <html></html>, que a su vez se divide en:

- Cabecera: <head></head> contiene info sobre el título de la página, autor, palabras clave, etc... Es obligatorio definir el título del documento con <title></title> (el título aparece en la barra del título de la parte superior, el resto de información no se muestra).
- Cuerpo: contiene la info que se presenta en pantalla y se delimita entre las etiquetas <body></body> o <frameset></frameset>

1.4. Identificación de etiquetas y atributos HTML

Una de las diferencias entre HTML y XML, es que HTML tiene etiquetas limitadas por las definidas en el propio lenguaje.

HTML tiene una gran cantidad de etiquetas, pero no son suficientes para crear páginas complejas, por ello se añade información adicional a las etiquetas mediante atributos (dando lugar a elementos).

Estos atributos tienen un conjunto de valores que se les puede asignar y si alguno no es válido, el navegador los ignora.

1.5. Clasificación de los atributos comunes según su funcionalidad

Atributos básicos o globales.

Se usan en casi todas las etiquetas.

- **name="texto"**. Permite asignar el nombre texto a un objeto HTML
- **title="texto"**. Asigna un título a un elemento HTML mejorando su accesibilidad. Este título es mostrado en navegadores cuando se pasa el ratón por encima, útil en elementos: a, link, img, object, abbr y acronym

- **id="texto"**. Permite identificar el elemento HTML sobre el que se aplica de forma individual. Útil trabajando con CSS y Javascript. No puede empezar por números y solo puede contener letras, números, guiones medios o bajos.
- **style="texto"**. Aplica al elemento el estilo llamado texto directamente.
- **class="texto"**. Aplica al elemento el estilo "texto" definido en las CSS. No puede empezar por números y solo puede contener letras, números, guiones medios o bajos.

Atributos para internacionalización.

Lo utilizan páginas en varios idiomas para indicar de forma explícita el idioma de sus contenidos.

- **dir**. Indica la dirección del texto, solo puede tomar dos valores: **ltr** de izda. a dcha o **rtl** de derecha a izquierda.
- **lang="código"**. Especifica el idioma del elemento mediante un código predefinido en el documento RFC1766 (en, en-US, ja, es, fr, fr-CA...)
- **xml:lang="código"**. Especifica el idioma del elemento mediante un código definido según RFC 1766.

en XML xml:lang tiene más prioridad que lang y es obligatorio si se incluye ese atributo.

Atributos de eventos y atributos para elementos que pueden obtener foco

Utilizados en webs dinámicas con JavaScript.

2. Elementos HTML

Un elemento HTML está formado por

- Etiqueta de apertura.
- Cero o más atributos.
- Opcionalmente un texto encerrado por la etiqueta (no todas las etiquetas encierran texto).
- Etiqueta de cierre.

Pueden ser de dos tipos

- **Elementos en línea**: ocupan el espacio necesario para mostrar sus contenidos
- **Elementos de bloque**. Siempre empiezan en una nueva línea y ocupan todo el espacio disponible hasta el final de la línea aunque el contenido no llegue hasta allí. Puede ser texto, elementos en línea u otros elementos de bloque.

```
<!DOCTYPE html>

<html>
  <head>
    <title>Ejemplo de la diferencia entre los elementos en línea
      y los elementos de bloque
    </title>
  </head>
  <body>
    <h1>Los encabezados son elementos de bloque.</h1>
    <p>Y los párrafos también.</p>
    <a href="#">Los enlaces son elementos de línea</a>
    <p>Incluso si esta definido dentro de un párrafo,
      <strong>un texto resaltado</strong>
      sigue siendo un elemento en línea.</p>
  </body>
</html>
```

Los encabezados son elementos de bloque.

Y los párrafos también.

[Los enlaces son elementos de línea](#)

Incluso si esta definido dentro de un párrafo, **un texto resaltado** sigue siendo un elemento en línea.

2.1. Elementos de estructura básica de un documento

`<html></html>` define el inicio y el final del documento HTML. Contiene

`<head></head>` define inicio y final de cabecera del documento

`<body></body>` define inicio y final del cuerpo del documento.

```
<!DOCTYPE html>

<html>
  <head>
    <title>Ejemplo de la estructura de un documento HTML</title>
  </head>
  <body>
    Aquí&iacute; es donde se coloca la informaci&ocute;n que se quiere visualizar en el navegador.
  </body>
</html>
```

2.2. Elementos de la sección cabecera

Elementos contenedores

| Elemento | Descripción |
|------------------------|--|
| title | Título del documento. Elemento obligatorio. |
| script | Script incrustado. Su contenido ha de ir situado entre las marcas de comentarios ya que no ha de ser interpretado. |
| style | Estilo aplicado al documento utilizando CSS. Su contenido ha de ir situado entre las marcas de comentarios ya que no ha de ser interpretado. |

Elementos no contenedores

| Elemento | Descripción |
|-------------------------|---|
| base | URI base del documento |
| isindex | Prompt de entrada de datos. (Eliminado de los estándares web aunque todavía algún navegador lo soporta) |
| link | Enlaces a documentos externos de librerías |
| meta | Metadatos sobre la página, como la codificación de caracteres, descripción o autores. |

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>El título es obligatorio</title>
  </head>
  <body>
    ...
  </body>
</html>
```

UTF-8 permite que los acentos se vean bien.

2.3. Encabezados y párrafos

Para agrupar párrafos se usa el elemento `<p></p>`

Para encabezados se utilizan 6 elementos

`<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`

A menor número mayor importancia de encabezado. Se utilizan para organizar jerárquicamente el contenido.

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

```
<head>
  <meta charset="UTF-8">
  <title>Párrafos y encabezadaos</title>
</head>
<body>
  <h1>Equipos</h1>
  <p>Primer párrafo</p>
  <p>Segundo párrafo</p>
  <h2>Recién ascendidos</h2>
  <p>...</p>
  <h1>Jugadores</h1>
  <p>...</p>
</body>
</html>
```

Equipos

Primer párrafo

Segundo párrafo

Recién ascendidos

...

Jugadores

...

2.3.1. Saltos de línea y espacios en blanco

Los documentos HTML ignoran los saltos de línea y para varios espacios consecutivos solo muestran uno.

 introduce saltos de línea y añade espacios en blanco.

[illegible]

Esto se ve en una línea sin espacios de más

Aquí hay espacios

Esto
introduce un salto de línea

2.3.2. Comentarios

se pueden introducir comentarios de una o varias líneas, no son procesados por los navegadores.

```
<!-- comentario
de varias
líneas -->
```

2.4. Semántica a nivel texto

- **<a>** para definir hipervínculos.
 - Su atributo más importante es **href="URL"**, a veces se utiliza **href="#"** para referirse a la misma página del vínculo.
 - **target** permite elegir donde se abrirá el vínculo.
 - **target="_blank"** en ventana nueva.
 - **target="_self"** para que se abra en la misma (valor por defecto).
- **** para representar texto importante.
- **** para indicar énfasis.
- **
** salto de línea.
- **<small>** para comentarios accesorios (letra pequeña).

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Semántica a nivel de texto</title>
  </head>
  <body>
    Texto marcado como <strong>importante</strong>.
    <br>
    Texto con <em>énfasis</em>
    <br>
    Texto marcado <small> con el elemento small </small>
    <br>
    Pulsa <a href="https://www.w3.org/">aquí</a> para ir a la página del W3C.
  </body>
</html>
```

Texto marcado como **importante**.
 Texto con *énfasis*
 Texto marcado con el elemento small
 Pulsa [aquí](https://www.w3.org/) para ir a la página del W3C.

2.5. Elementos de listas

Hay tres tipos: Ordenadas, desordenadas y de definición.

| Elemento | Descripción |
|---------------------------|---|
| <u>ul</u> | Delimita los elementos que forman una lista desordenada |
| <u>ol</u> | Delimita los elementos que forman una lista ordenada. Tiene varios atributos. |
| <u>dl</u> | Delimita los elementos que forman una lista de definición |
| <u>li</u> | Cada uno de los elementos de una lista ordenada o desordenada. |
| <u>dt</u> | Cada uno de los términos que se definen de una lista de definición. |
| <u>dd</u> | Cada una de las definiciones de una lista de definición. |

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Listas</title>
```

```

</head>
<body>
  <h1>Ejemplo de lista desordenada: Módulos de 1º de ASIR</h1>
  <ul>
    <li>Fundamentos de Hardware</li>
    <li>Gestión de Bases de Datos</li>
  </ul>
  <h1>Ejemplos de listas ordenadas: Módulos de 1º de ASIR</h1>
  Comenzando en 1
  <ol>
    <li>Fundamentos de Hardware</li>
    <li>Gestión de Bases de Datos</li>
  </ol>
  Comenzando en 4
  <ol start = "4">
    <li>Fundamentos de Hardware</li>
    <li>Gestión de Bases de Datos</li>
  </ol>
  <h1>Ejemplo de lista de definición: Módulos de 1º de ASIR</h1>
  <dl>
    <dt>Fundamentos de Hardware</dt>
    <dd>Componentes físicos de un ordenador</dd>
    <dt>Gestión de Bases de Datos</dt>
    <dd>Diseño y uso de bases de datos relacionales</dd>
  </dl>
</body>
</html>

```

Ejemplo de lista desordenada: Módulos de 1º de ASIR

- Fundamentos de Hardware
- Gestión de Bases de Datos

Ejemplos de listas ordenadas: Módulos de 1º de ASIR

Comenzando en 1

1. Fundamentos de Hardware
2. Gestión de Bases de Datos

Comenzando en 4

4. Fundamentos de Hardware
5. Gestión de Bases de Datos

Ejemplo de lista de definición: Módulos de 1º de ASIR

Fundamentos de Hardware
 Componentes físicos de un ordenador
 Gestión de Bases de Datos
 Diseño y uso de bases de datos relacionales

2.6. Elementos de tablas

Los elementos de las tablas son los siguientes (sin necesidad de usarlos todos)

| Elemento | Descripción |
|--------------------------|--|
| table | Delimita el contenido de una tabla. |
| tr | Delimita cada una de las líneas de la tabla. |
| td | Delimita el contenido de cada celda de la tabla. |
| colgroup | Permite agrupar columnas. |
| tbody | Permite agrupar líneas de la tabla. |
| thead | Define la línea cabecera de la tabla. |
| th | Delimita cada una de las celdas de la cabecera |
| tfoot | Define la fila pie de la tabla. |
| caption | Para añadir una leyenda a la tabla |

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Tablas</title>
  </head>
  <body>
    <table>
      <caption>Tabla de socios</caption>
      <tr>
        <th>Nombre</th>
        <th>Apellido</th>
        <th>Edad</th>
      </tr>
      <tr>
        <td>Juan</td>
        <td>Puertas</td>
        <td>54</td>
      </tr>
      <tr>
        <td>Eva</td>
        <td>Montes</td>
        <td>44</td>
      </tr>
    </table>
  </body>
</html>

```

| Tabla de socios | | |
|-----------------|----------|------|
| Nombre | Apellido | Edad |
| Juan | Puertas | 54 |
| Eva | Montes | 44 |

2.7. Elementos de formularios

Recogen información que el usuario introduce en el navegador. Se deben validar los datos para detectar errores en local en caso de haberlos y así evitar sobrecargar el servidor con tareas innecesarias.

2.7.1. Declaración de formulario

<form></form> se utiliza para abrir y cerrar formulario y permite especificar los siguientes atributos:

- **name** - Nombre de formulario
- **action** - Acción que se ejecuta cuando se envía el formulario.
- **enctype** - formato en el que se envían los valores del formulario.
- **method** - método de transmisión de datos (get o post).
 - **GET**
 - Permite pasar valores en ASCII hasta 100 caracteres.

- Las variables que se transmiten se ven en la URL y van concatenadas con (&)

- **POST**

- Permite pasar valores de variables y archivos.
- Carece de restricciones de longitud.
- Las variables no son visibles en URL.

2.7.2. Campos de formulario - input

el elemento **<input>**, sin etiqueta de cierre, se utiliza para varios tipos de control según se declare el atributo type. Además otros atributos son:

- **name** - nombre del campo.
- **size** - caracteres visibles en el campo (por defecto 20).
- **maxlength** - máximo de caracteres permitidos a introducir.
- **value** - valor por defecto del campo.
- **placeholder** - valor sugerido, se representa en gris y desaparece al hacer foco en él.
- **readonly** - no puede ser modificado.
- **autofocus** - se sitúa el foco en él en cada recarga de página.
- **required** - campo obligatorio.

```
<input type="text" name="usuario" size="30" maxlength="20" placeholder="Escriba aquí el nombre de usuario" required>
```

Usuario

2.7.3. Campos de formulario - Área de texto

Son campos de texto que permiten un mayor número de caracteres y se especifican con la etiqueta **<textarea>**. Sus atributos principales son:

- **name**: nombre del campo.
- **rows**: número de filas (reemplazable por la propiedad CSS height)-
- **cols**: número de columnas (reemplazable por CSS width).

También puede utilizar los siguientes atributos input: **placeholder, readonly, autofocus, maxlength, required.**

```
Describe a continuación sus intereses: <br> <textarea name="area"></textarea>
```

Describe a continuación sus intereses:

2.7.4. Campos de formularios - Lista desplegable

Permite al usuario seleccionar un valor entre diferentes opciones. Se usan las siguientes etiquetas.

<select></select> para definir la lista. Sus atributos son:

- **name** - Nombre del elemento.
- **size** - número de elementos de la lista que se mostrarán.
- **multiple** - permite seleccionar varias opciones de la lista.

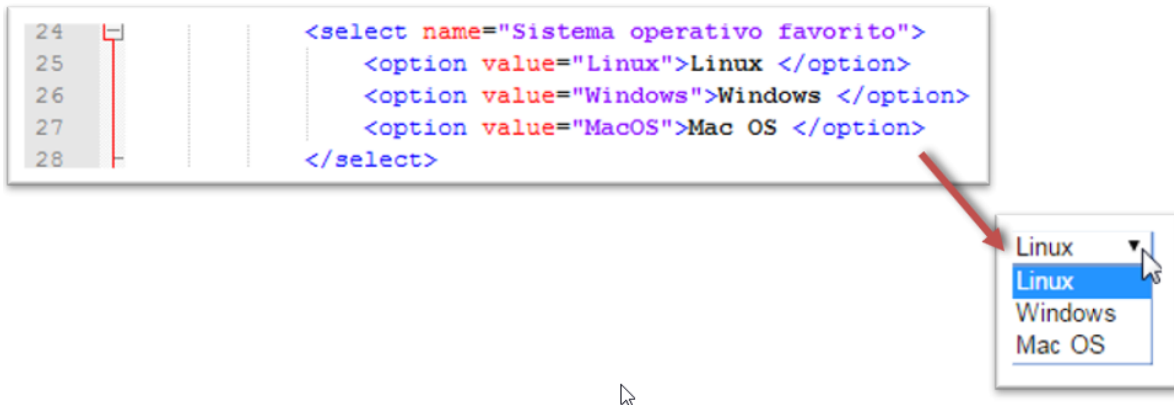
<option></option> para definir cada elemento de la lista. Sus atributos son:

- **selected** - seleccionado por defecto.

- **valor** - valor que se transmite cuando se envía el formulario.

Por último seleccione el sistema operativo de su ordenador:

```
<select name="Sistema operativo favorito">
  <option value="Linux">Linux </option>
  <option value="Windows">Windows </option>
  <option value="MacOS">Mac OS </option>
</select>
```



2.7.5. Campos de formularios - checkbox

Para que el usuario indique una o varias opciones: **<input type="checkbox">**

- **name** - obligatorio.
- **checked** - marcado por defecto.
- **value** - valor a enviar en formulario.

Estoy conforme con la política de privacidad de la página

```
<input type="checkbox" name="conforme" checked>
```

Estoy conforme con la política de privacidad de la página ☒

2.7.6. Botones de formulario

Botón de envío

<input type="submit">

Envía el formulario a un destinatario (servidor web, email...), dependiendo del valor del atributo action de la etiqueta **<form>**. Su atributo es:

- **value** - indica el texto del botón.

Botón de anulación

<input type="reset">

Borra el contenido de los campos cumplimentados del formulario. También utiliza el atributo value.

2.7.7.- Otros campos de formulario

| Campo | Etiqueta | Notas |
|--|--|--|
| Oculto | <code><input type="hidden"></code> | |
| De contraseña | <code><input type="password"></code> | |
| Para envío de ficheros | <code><input type="file"></code> | Requiere formulario definido con post y enctype="multipart/form-data" |
| Para recoger correo electrónico | <code><input type="email"></code> | Valida formato automáticamente |
| Para recoger una URL | <code><input type="url"></code> | Valida formato automáticamente |
| Para recoger números enteros en un rango | <code><input type="number"></code> | Atributos: max – valor máximo min – valor mínimo step – incremento del contador |

2.7.8. Campos de formulario con formato de fecha

Se pueden utilizar diferentes variaciones dependiendo de lo que queramos recoger:

`<input type="datetime-local">` Día, mes, año y hora

`<input type="date">` Día, mes, año

`<input type="month">` Mes y año

`<input type="time">` Hora. Sus atributos son:

- **min** - menor hora seleccionable
- **max** - mayor hora seleccionable

2.7.9. Campos de formulario - Rangos

Permite seleccionar un valor dentro de un rango.

`<input type="range">`

Atributos:

- **min** - valor mínimo del rango.
- **max** - valor máximo del rango.
- **step** - valor del incremento del contador.
- **value** - valor inicial.

2.7.10. Organización de formularios

Etiqueta <label>

Permite asociar una etiqueta con su nombre a cada campo, además constituye una ayuda de usabilidad para invidentes.

```
<label for="conforme">Acepto el acuerdo de licencia</label>
<input type="checkbox" name="licencia" id="conforme" value="ok">
```

En este ejemplo label se asocia al input mediante el atributo for y el valor del identificador (id) de la etiqueta input.

Acepto el acuerdo de licencia ☐

2.8. Multimedia

| Elemento | Descripción |
|------------------------|--|
| img | Permite insertar una imagen en una página web. |
| audio | Para insertar audio |
| video | Para insertar vídeo |
| object | Para incrustar contenido multimedia (audio, vídeo, ficheros PDF, applets...) |

Elemento

No tiene etiqueta de cierre y sus atributos son:

- **src** - ruta de la imagen, local o URL, en general se utiliza jpg, png y gif.
- **alt** - Texto alternativo. Es oblogatorio.
- **height** - Altura, si no es especifica se escoge el original y si se especifica, se escala. HTML5 lo soporta.
- **width** - Anchura de la imagen.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset='UTF-8'>
    <title>Imágenes</title>
  </head>
  <body>
    <p> Imagen en internet </p>
    <img src='https://i.blogs.es/aa1b9a/luna-100mpx/450_1000.jpg' alt='la luna'>
    <p>Imagen no encontrada</p>
    <img src='http://rutaincorrecta.jpg' alt='la luna'>
    <p>Imagen local</p>
    <img src='index.jpeg' alt='logo'>
  </body>
</html>
```

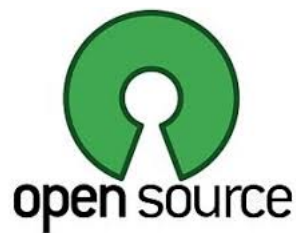
Imagen en internet



Imagen no encontrada

la luna

Imagen local



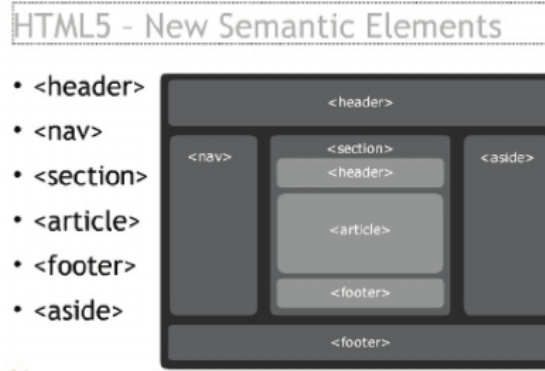
En la segunda imagen al no encontrarla muestra texto alternativo.

2.9. Secciones y etiquetas semánticas

Se usa el elemento `<div>` para agrupar otros elementos o secciones, tanto para organizar contenido como para posicionarlo con hojas de estilo.

En HTML5 aparecen varias etiquetas para estructurar contenido de página. Solo se debe usar `<div>` cuando no haya una etiqueta más apropiada.

- **`<header>`** Contenido introductorio. Suele contener elementos encabezado: `h1`, `h2`...
- **`<aside>`** contenido parcialmente relacionado con el principal. No tiene por qué ser en el lateral.
- **`<footer>`** Contenido sobre la sección correspondiente, como el autor. No tiene por qué ser la parte inferior.
- **`<section>`** Una sección genérica dentro del documento.
- **`<article>`** Representa un elemento que se puede distribuir de manera independiente o reutilizable.
- **`<nav>`** Contiene vínculos, internos o externos, se suele usar para barra de navegación.



2.10. Elemento iframe

Permite integrar una web dentro de otra, sustituye a frameset de HTML4, ya obsoleta. Atributos:

height - altura, por defecto en píxeles, se puede fijar con CSS y se usa barra de desplazamiento si no cabe todo el contenido.

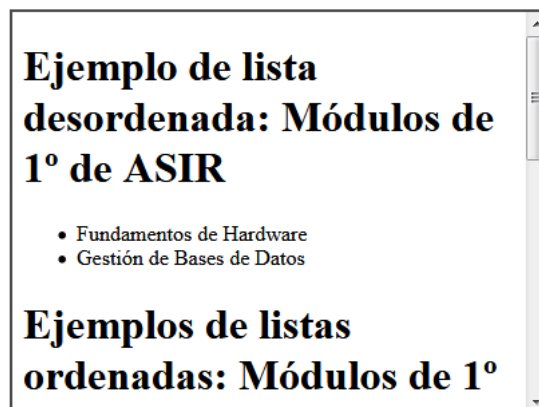
width - como el anterior.

src - ruta del contenido inicial del iframe

name - nombre, para referirse al iframe desde un vínculo.

[Ejemplo de listas](#) [Ejemplo de tablas](#)

Aquí está el iframe



El contenido del iframe cambia al seguir los vínculos, para eso se establece en el atributo target del vínculo el nombre del iframe.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>iframe</title>
  </head>
  <body>
    <a href = "listas.html" target = "contenido">Ejemplo de listas</a>
    <a href = "tablas.html" target = "contenido">Ejemplo de tablas</a>
    <p>Aquí está el iframe</p>
    <iframe height = "300" width = "400" name = "contenido" src = "listas.html">
  </body>
</html>
```

3. Hojas de estilo o CSS

CSS (Cascading Style Sheets) permite controlar el estilo y formato de múltiples web al mismo tiempo. Hasta su aparición, las páginas web definían su aspecto en cada elemento de las etiquetas HTML.

CSS separa los contenidos de la página de su aspecto. Cualquier cambio en el estilo CSS afecta a todas las páginas vinculadas a ella en las que aparezca ese elemento.

Están compuestas por una o más reglas de estilo aplicadas a HTML o XML. Una vez creados los contenidos de estos se utiliza CSS para definir el formato de sus elementos.

CSS aparece poco después del lenguaje de etiquetas SGML. En 1995 W3C añade el desarrollo y estandarización CSS a su grupo de trabajo.

CSS1 (1996).

CSS2 (1998)

CSS3 (Junio de 1999). Se divide en varios documentos separados llamados módulos. Un módulo primero debe pasar por la fase de candidato antes de ser aprobado.

CSS4. Última versión. Algunos módulos están especificación nivel 3 mientras otros (como selectores) están en nivel 4.

3.1. Cómo incluir CSS en un documento HTML o XHTML

Se hace de 3 formas distintas.

- **Declaración en línea:** desaconsejada.

```
<p style="color: red;"> ... </p>
```

- **Declaración interna:** se declaran en el encabezado de la página mediante <style>

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset='UTF-8'>
    <title>CSS interna</title>
    <style>
      p {color:green}
    </style>
  </head>
  <body>
    <p>Hola</p>
  </body>
</html>
```

- **Declaración externa:** se declara la hoja de estilo en otro documento (con extensión .css) y se llama a ella desde las páginas con la etiqueta link

```
<link rel="stylesheet" type="text/css" href="rutaArchivo.css">
```

El código del .css no tendrá etiqueta de declaración de estilo.

También se pueden llamar a hojas de estilo externas con la etiqueta @import

```
<style type="text/css">
  @import url("formato1.css");
</style>
```

3.1.1. Elemento span

Para dar estilo a un texto no marcado.

```
<p> Parte de este párrafo <span style="color:red">está en rojo</span>
```

3.2. Sintaxis de las reglas de estilo

Cada uno de los estilos de un CSS se denomina regla, formada por: selector, llave de apertura, declaración y llave de cierre.

```
p { color: blue;}
```

3.3. Cascada y herencia de estilos

Ante estilos contradictorios, el navegador aplica la siguiente precedencia:

1. Declaración en línea.
2. Declaración interna.
3. Declaración externa.
4. Propiedades por defecto del navegador.

Las hojas de estilo tienen herencia en sus propiedades. Si tenemos varios elementos html anidados, estos heredan estilos de los externos, siempre y cuando no tengan los suyos propios definidos.

3.4. Selectores

Los selectores permiten identificar a qué elementos de nuestro código HTML vamos a aplicar el estilo definido:

Selector universal (*)

Se aplica a todos los elementos de la página.

```
* {  
  margin: 10px;  
  padding: 5px;  
};
```

Selectores de etiqueta

Se aplica el estilo a la etiqueta.

```
p { text-align: center;}
```

Selector múltiple

Se aplica el estilo a todas las etiquetas.

```
p, h1, h2 { text-align: center;}
```

3.4.1. Selectores de clase

Se utilizan para asignar estilos a etiquetas de una clase determinada, sin que afecten a todas las etiquetas del mismo tipo.

En HTML:

```
<p class="parrafoCentrado"> ... </p>
```

en CSS:

```
p.parrafoCentrado {text-align: center; }
```

Utilizando la misma definición omitiendo el identificador de etiqueta se aplicaría el estilo a todas las etiquetas que pertenezcan a la clase "parrafoCentrado":

```
.parrafoCentrado { text-align: center; }
```

Ejemplo:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset='UTF-8'>
    <title>Ejemplo del uso de clases en hojas de estilo</title>
    <style type="text/css">
      .clase_azul{color:blue}
      p.clase_roja{color:#ff0000; font-style:italic; font-weight:bolder; font-family:courier;}
    </style>
  </head>
  <body>
    <h3 class="clase_azul">Ejemplo del uso de clases en hojas de estilo</h3>
    <p>Cualquier elemento sobre el que apliquemos la clase clase_azul tendrá el texto azul.</p>
    <p class="clase_azul"> Incluso el párrafo.</p>
    <p class="clase_roja">Sobre el párrafo podemos aplicar la clase clase_roja y el texto será rojo, en negrita cursiva y 1
    <h3 class="clase_roja"> Pero este texto no aparecerá formateado ya que regla de la clase clase_roja solo actúa sobre pá
  </body>
</html>
```

Ejemplo del uso de clases en hojas de estilo

Cualquier elemento sobre el que apliquemos la clase clase_azul tendrá el texto azul.

Incluso el párrafo.

Sobre el párrafo podemos aplicar la clase clase_roja y el texto será rojo, en negrita cursiva y la familia del texto courier.

Pero este texto no aparecerá formateado ya que regla de la clase clase_roja solo actúa sobre párrafos.

3.4.2. Selectores de ID

Permite seleccionar un elemento de la página por medio de su atributo ID. Se asocia a elementos de estilo que se van a aplicar de manera excepcional una única vez. El valor del atributo id no se debe repetir en dos elementos diferentes de la página.

3.4.3. Selectores descendientes

Asocia elementos que están dentro de otros elementos. Por ejemplo:

```
p h1 {color: red; };
```

Se aplicará el estilo a todas las etiquetas h1 que estén dentro de bloques p. Se debe tener en cuenta que

- No tiene por qué ser descendiente directo.
- El nivel de anidación puede tener varios niveles:

```
p a b i {text-decoration: underline; }
```

En este ejemplo se aplica a elementos en cursiva que están dentro de etiquetas de negrita, anidados dentro de enlaces que se encuentren en párrafos.

Hay que tener en cuenta que en el ejemplo anterior no sería lo mismo si estuvieran los selectores separados por coma.

También podemos combinar el selector universal con selectores descendientes:

```
p * b {color: #0000FF;}
```

Se aplica a todas las etiquetas que estén anidadas en cualquier otra etiqueta que a su vez esté dentro de una etiqueta tipo <p>, pero no a las que están dentro de <p> directamente.

3.4.4. Selector hijo

Es parecido al anterior, pero solo afecta al primer nivel de anidamiento.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset='UTF-8'>
    <title>Selector hijo</title>
    <style>
      section > p {color:red}
    </style>
  </head>
  <body>
    <p>Párrafo inicial</p>
    <section>
      <p>Párrafo hijo de sección</p>
      <article>
        <p>Párrafo nieto de sección</p>
      </article>
    </section>
  </body>
</html>
```

Párrafo inicial

Párrafo hijo de sección

Párrafo nieto de sección

3.5. Propiedades principales

3.5.1. Propiedades de color y fondo

No todos los nombres de colores son admitidos en el estándar, así que se aconseja utilizar RGB.

| Elemento | Descripción |
|---------------------------------------|---|
| color | Indica el color del texto. Lo admiten casi todas las etiquetas de HTML . El valor de este atributo es un color, con su nombre o su valor 👉 RGB . |
| background-color | Indica el color de fondo del elemento. El valor de este atributo es un color, con su nombre o su valor RGB. |
| background-image | Permite colocar una imagen de fondo del elemento. El valor que toma es el nombre de la imagen con su camino relativo o absoluto |
| background-repeat | Indica si ha de repetirse la imagen de fondo y, en ese caso, si debe ser horizontal o verticalmente. Los valores que puede tomar son: <code>repeat-x</code> , <code>repeat-y</code> o <code>no-repeat</code> . |
| background-attachment | Especifica si la imagen ha de permanecer fija o realizar un scroll. Los valores que pueden tomar son: <code>scroll</code> o <code>fixed</code> . |
| background-position | Es una medida, porcentaje o el posicionamiento vertical u horizontal con los valores establecidos que sirve para posicionar una imagen. Los valores que puede tomar son: porcentaje, tamaño, o <code>[top, center, bottom]</code> <code>[left, center, right]</code> |
| background | Establece en un solo paso cualquiera de las propiedades de background anteriores. Los valores que puede tomar son: <code>background-color</code> , <code>background-image</code> , <code>background-repeat</code> , <code>background-attachment</code> , <code>background-position</code> . |

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo de atributos CSS de color y fondo</title>
    <style type="text/css">
      body { background-color: black; color:yellow; }
      p { color: #ffffff;}
    </style>
  </head>
  <body>
    <h3>Ejemplo del uso de atributos de color y fondo</h3>
    <p>El texto de cualquier elemento, salvo el del párrafo que es blanco, es amarillo y el fondo negro.</p>
  </body>
</html>
```

Ejemplo del uso de atributos de color y fondo

El texto de cualquier elemento, salvo el del párrafo que es blanco, es amarillo y el fondo negro.

3.5.2. Propiedades de fuente

| Elemento | Descripción |
|------------------------------|--|
| font-size | Indica el tamaño de la fuente. Puede ser un tamaño absoluto, relativo o en porcentaje. Toma valores de unidades de CSS . |
| font-family | Establece la familia a la que pertenece la fuente. Si el nombre de una fuente tiene espacios se utilizan comillas para que se entienda bien. El valor es el nombre de la familia fuente. |
| font-weight | Define el grosor de los caracteres. Los valores que puede tomar son: <code>normal</code> , <code>bold</code> , <code>bolder</code> , <code>lighter</code> , 100, 200, 300, 400, 500, 600, 700, 800 o 900. |
| font-style | Determina si la fuente es normal o cursiva. El estilo <code>oblique</code> es similar al cursiva. Los valores posibles son: <code>normal</code> , <code>italic</code> , <code>oblique</code> . |
| font-variant | Determina si la fuente es normal o mayúsculas pequeñas. Los valores que puede tomar son: <code>normal</code> , <code>small-caps</code> . |
| line-height | El alto de una línea y por tanto, el espaciado entre líneas. Es una de esas características que no se podían modificar utilizando HTML. |
| font | Permite establecer todas las propiedades anteriores en el orden que se indica a continuación: <code>font-style</code> , <code>font-variant</code> , <code>font-weight</code> , <code>font-size</code> [<code>line-height</code>], <code>font-family</code> . Los valores han de estar separados por espacios. No es obligatorio el uso de todos los valores. |

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo de atributos CSS de fuente</title>
    <style type="text/css">
      body { background-color: black; color:yellow; font-family: courier }
      p { color: #ffffff; font:italic 900 12px Verdana; }
    </style>
  </head>
  <body>
    <h3>Ejemplo del uso de atributos de fuente</h3>
    <p>El texto de cualquier elemento es de la familia Courier y amarillo, salvo el del párrafo que es Verdana, blanco y de tamaño 12 px.</p>
  </body>
</html>
```

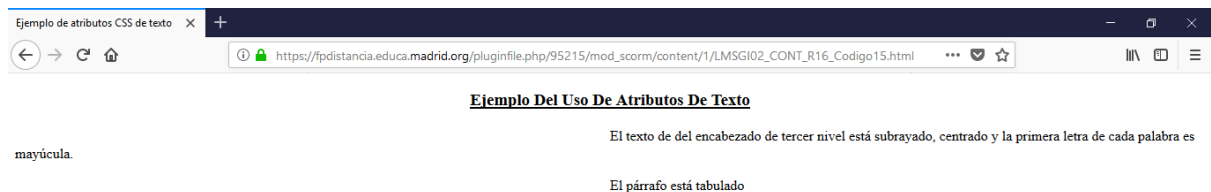
Ejemplo del uso de atributos de fuente

El texto de cualquier elemento es de la familia Courier y amarillo, salvo el del párrafo que es Verdana, blanco y de tamaño 12 px.

3.5.2. Propiedades de texto

| Elemento | Descripción |
|---------------------------------|--|
| text-decoration | Establece si el texto está subrayado, sobrerayado o tachado. los valores que puede tomar son: none, underline, overline, line-through o blink |
| text-align | Indica la alineación del texto. Aunque las hojas de estilo permiten el justificado de texto no funciona en todos los sistemas. Los valores que puede tomar son: left, right, center o justify |
| text-indent | Determina la tabulación del texto. Los valores que toma son una longitud, en unidades CSS, o un porcentaje de la establecida. |
| text-transform | Nos permite transformar el texto, haciendo que tenga la primera letra en mayúsculas de todas las palabras, todo en mayúsculas o minúsculas. Los valores que puede tomar son: capitalize, uppercase, lowercase o none |
| word-spacing | Determina el espaciado entre las palabras. Los valores que puede tomar es un tamaño. |
| letter-spacing | Determina el espaciado entre letras. Los valores que puede tomar es un tamaño. |
| vertical-align | Establece la alineación vertical del texto. Sus valores posibles son: baseline, sub, super, top, text-top, middle, bottom, text-bottom O UN porcentaje. |
| line-height | Altura de la línea. Puede establecerse mediante un tamaño o un porcentaje |

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo de atributos CSS de texto</title>
    <style type="text/css">
      h3 { text-decoration:underline; text-align: center; text-transform: capitalize }
      p { text-indent: 50%; }
    </style>
  </head>
  <body>
    <h3>Ejemplo del uso de atributos de texto</h3>
    <p>El texto de del encabezado de tercer nivel está subrayado, centrado y la primera letra de cada palabra es mayúcula.
    <p>El párrafo está tabulado</p>
  </body>
</html>
```



3.5.3. Propiedades de lista

| Elemento | Descripción |
|-------------------------------------|---|
| list-style-type | Indica cual es el símbolo que se utiliza como marcador en las listas. Valores que puede tomar son: disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, none. |
| list-style-image | Permite utilizar el uso de una imagen como marcador en una lista. El valor que toma es la ruta del fichero imagen |
| list-style-position | Determinan la posición del marcador en una lista. Puede tomar los valores: outside o inside. |
| list-style | Permite establecer de una única vez todas las características de una lista. Hay que seguir el orden siguiente: list-style-type, list-style-position y list-style-image. |

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset='UTF-8'>
    <title>Estilo para listas</title>
    <style>
      #flecha{ list-style-image: url("flecha.png") }
      .circ{ list-style-type: circle }
      #armenio{ list-style-type: armenian }
    </style>
  </head>
  <body>
    <p>Lista con imagen</p>
```

```

<ul id="flecha">
  <li>Patatas</li>
  <li>Peras</li>
</ul>
<p>Lista con círculo</p>
<ul class="circ">
  <li>Patatas</li>
  <li>Peras</li>
</ul>
<p>Alfabeto armenio</p>
<ol id="armenio" reversed>
  <li>Peras</li>
  <li>Manzanas</li>
</ol>
</body>
</html>

```

Lista con imagen



Lista con círculo

- Patatas
- Peras

Alfabeto armenio

Ք. Peras
Ա. Manzanas

3.5.5. Propiedad display

Se usa para:

- Hacer que un elemento sea de bloque o de línea.
- Ocultarlo o hacerlo visible (a través de javascript).

En el siguiente ejemplo vemos varios vínculos (elementos en línea). Los que tienen la clase "especial" están definidos como elementos de bloque:

```

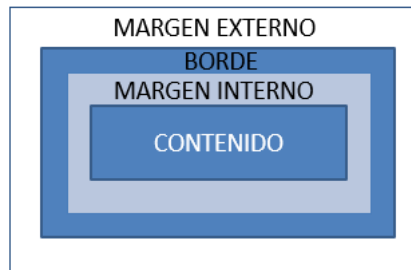
<!DOCTYPE html>
<html>
  <head>
    <title>Ejemplo de atributos CSS de fuente</title>
    <style>
      .especial { display: block}
    </style>
  </head>
  <body>
    <a href="#">Primer vínculo (normal)</a>
    <a href="#">Segundo vínculo (normal)</a>
    <a href="#" class="especial">Tercer vínculo (especial)</a>
    <a href="#" class="especial">Cuarto vínculo (especial)</a>
  </body>
</html>

```

[Primer vínculo \(normal\)](#) [Segundo vínculo \(normal\)](#)
[Tercer vínculo \(especial\)](#)
[Cuarto vínculo \(especial\)](#)

3.6. Avanzado: Modelo de cajas

La W3C define lo que llaman "modelo de caja", que es una zona rectangular que rodea a cada uno de los elementos de la web:



- **Contenido:** contenido de la etiqueta.
- **Margen interior.** Distancia desde el contenido hasta el borde del elemento (**padding**).
- **Borde.** El borde del elemento (**border**).
- **Margen exterior.** Distancia desde el borde del elemento a los elementos adyacentes (**margin**).

3.6.1. Propiedades de caja

Contenido

No se tiene en cuenta el borde y los márgenes.

| FORMATO | PROPIEDAD | VALORES |
|---------|------------------------|-------------------------------------|
| anchura | width | auto Longitud en px o equivalente |
| altura | height | auto Longitud en px o equivalente |

Margen interno

- Utilizando padding solamente se aplica a los cuatro lados.
- Con dos valores, el primero aplica al superior y al inferior y el segundo a los laterales.
- Con tres valores se aplica el primero al superior, el segundo a los laterales y el tercero al inferior.
- Con 4 valores se aplica en orden: superior, derecho, inferior, izquierdo.

| FORMATO | PROPIEDAD | VALORES |
|---------|------------------------|-------------------------------------|
| anchura | width | auto Longitud en px o equivalente |
| altura | height | auto Longitud en px o equivalente |

Borde

- Si utilizamos border-color, border-width o border-style se aplicará a los cuatro lados.
- Con dos valores se aplica el primero al superior y al inferior y el segundo a los laterales.
- Con tres valores se aplica el primero al superior, el segundo a los laterales y el tercero al inferior.
- Con cuatro valores se aplica en orden: superior, derecha, inferior, izquierda.

| FORMATO | PROPIEDAD | VALORES |
|------------------|--|--|
| Color del borde | border-color border-top-color border-bottom-color border-right-color border-left-color | Color en alguna de las notaciones permitidas transparent |
| Grueso del borde | border-width Border-top-width Border-bottom-width Border-right-width Border-left-width | Valor de longitud thin médium thick |
| Estilo del borde | border-style border-top-style border-bottom-style border-right-style border-left-style | Solid dashed dotted double ridge Groove inset outset hidden none |

Margen externo

Mismo funcionamiento que el padding para declarar valores.

| FORMATO | PROPIEDAD | VALORES |
|--------------------------|--|--|
| Ancho del margen externo | margin margin-top margin-bottom margin-right margin-left | auto valor de longitud valor de porcentaje |

3.6.2. Unidades de tamaño

CSS permite utilizar diferentes tipos de unidades, que pueden ser:

- **Absolutas:** cualquier longitud expresada en estas unidades muestra el mismo tamaño.

| Unidades absolutas | |
|--------------------|-------------|
| cm | Centímetros |
| mm | Milímetros |
| in | Pulgadas |
| px * | Píxeles |
| pt | Puntos |
| pc | Picas |

- **Relativas:** dependen del tamaño de otro elemento.

| UNIDADES RELATIVAS | |
|--------------------|---|
| em | Relativa al tamaño del tipo de letra por defecto |
| porcentajes (%) | Relativos a las dimensiones del elemento contenedor |
| ex | Relativa al valor de x-height de la fuente actual |
| ch | Relativa al ancho del cero "0" |
| rem | Relativa al tamaño de letra del elemento raíz |

Para imágenes y vídeos se suele utilizar píxeles. Es la única unidad absoluta usada habitualmente.

Ejemplo de porcentajes 1:

La primera sección tiene fondo verde y su anchura es la mitad del elemento contenedor <body> la cual ocupa toda la anchura disponible, la segunda sección ocupa el 30%

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Porcentajes</title>
    <style>
      #s1{
        background-color:green;
        width:50%;
      }
      #s2{
        background-color:blue;
        width:30%;
      }
    </style>
  </head>
  <body>
    <section id = "s1">Contenido sección 1</section>
    <section id = "s2">Contenido sección 2</section>
  </body>
</html>
```



Ejemplo de porcentajes 2:

Con el alto ocurre lo mismo, pero depende del contenido de la página, a no ser que se fije un valor para el elemento como en este ejemplo. <body> tiene una altura fijada en píxeles y el tamaño de las secciones tienen porcentajes relativos al <body>

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Porcentajes</title>
    <style>
      body{
        height:800px;
      }
      #s1{
        background-color:green;
        width:50%;
        height:40%;
      }
      #s2{
        background-color:blue;
        width:30%;
        height:20%;
      }
    </style>
  </head>
  <body>
    <section id = "s1">Contenido sección 1</section>
    <section id = "s2">Contenido sección 2</section>
  </body>
</html>
```



Ejemplo en unidades em

Para las fuentes es habitual utilizar la unidad em, que hace referencia al tamaño de la fuente actual.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Unidades em</title>
    <style>
      .grande {font-size: 2em;}
      .muyGrande {font-size: 4em}
    </style>
  </head>
  <body>
    <p>Normal</p>
    <p class = "grande">Grande</p>
    <p class = "muyGrande">Muy grande</p>
  </body>
</html>
```

Normal

Grande

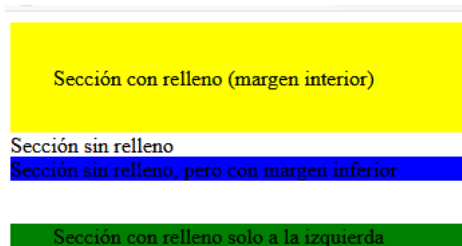
Muy grande

3.6.3. Ejemplos del modelo cajas

Ejemplo 1:

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Relleno y margen</title>
    <style>
      #relleno{
        background-color:yellow;
        padding: 2em; /*se aplica a izquierda, derecha arriba y abajo*/
      }
      #rellenoIzq{
        background-color:green;
        padding-left: 2em; /*se aplica a izquierda, derecha arriba y abajo*/
      }
      #rellenoMargen{
        background-color:blue;
        margin-bottom: 2em; /*se aplica a izquierda, derecha arriba y abajo*/
      }
    </style>
  </head>
  <body>
    <section id = "relleno">Sección con relleno (margen interior)</section>
    <section>Sección sin relleno</section>
    <section id = "rellenoMargen">Sección sin relleno, pero con margen inferior</section>
```

```
<section id = "rellenoIzq">Sección con relleno solo a la izquierda</section>
</body>
</html>
```



Ejemplo 2:

```
<html>
  <head>
    <meta charset="UTF-8">
    <title>Relleno y margen</title>
    <style>
      #borde1{
        background-color:yellow;
        border-color: green;
        border-style: solid;
      }
      #borde2{
        background-color:pink;
        border-width: 16px;
        border-left-color: green;
        border-top-color: green;
        border-left-style: dashed;
        border-top-style: dashed;
      }
    </style>
  </head>
  <body>
    <section id = "borde1">Sección con borde1</section>
    <br>
    <section id = "borde2">Sección con borde2</section>
  </body>
</html>
```



3.6.4. Posicionamiento

Con CSS es posible modificar el posicionamiento por defecto de los elementos, con las propiedades position y float.

Con la propiedad float, los elementos flotan hacia la izquierda o la derecha, todos los elementos flotados se van situando uno junto a otro y si no hay espacio pasan a una nueva línea.

En este ejemplo se utiliza la propiedad float para maqueta una página sencilla sobre los elementos <nav> y <section>.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>Etiquetas semánticas y float</title>
    <style>
      body{
        background-color:pink;
      }
    
```

```

header{ background-color:blue;}
nav {

    background-color:red;
    width:10%;
}
section {
    background-color:green;
    width:90%;
}
footer {
    background-color:yellow;
}
nav, section {
    height:500px;
    float: left;
}
}
</style>
</head>
<body>
<header>Encabezado</header>
<nav>Vinculos</nav>
<section>Contenido principal del página</section>
<footer>Página creada por...</footer>
</body>
</html>

```



Mapa Conceptual

