



TEMA 2: HTML5

Introducción

- Qué es HTML y su evolución.
- Qué es CSS y su evolución.
- Partes de una página Web.
- Añadir contenido a una página Web.
- Dotar de estilo nuestra Web usando CSS.



Introducción HTML

LAS SIGLAS DE HTML

HYPertext

“...texto que enlaza a otro texto, a otros contenidos...”

Esto es la base del funcionamiento de la web donde hay contenidos y páginas interconectadas

MARKUP

Traducción de marcas o Etiqueta.

```
<p>HOLA</p>
```

LANGUAGE

Tiene normas, estructuras y convenciones para definir contenido y estructura de una Web pero....**NO ES UN LENGUAJE DE PROGRAMACIÓN**

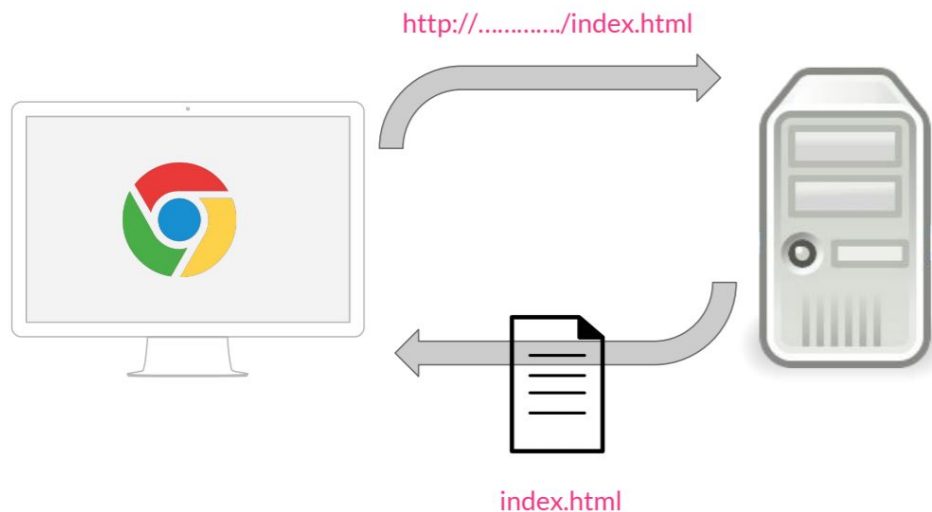
2.UNA DEFINICIÓN

...Es un **estándar** que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una **estructura** básica y un **código** (denominado código HTML) para la definición de **contenido de una página web**, como texto, imágenes, videos, juegos, entre otros...



3. FUNCIONAMIENTO

¿QUÉ SUCEDE CUANDO VISITO UNA WEB?



4. Referencias

Para aprender:

W3C SCHOOL

<https://www.w3schools.com/>





Introducción CSS

1.LAS SIGLAS

CASCADING

Porque los estilos,
salvo que
indiquemos lo
contrario, se
propagarán en
cascada

STYLE (Estilo)

Porque podemos
establecer
los estilos visuales y
el
diseño de nuestras
páginas
web

SHEETS (Hojas)

Porque son documentos
“aparte” del HTML

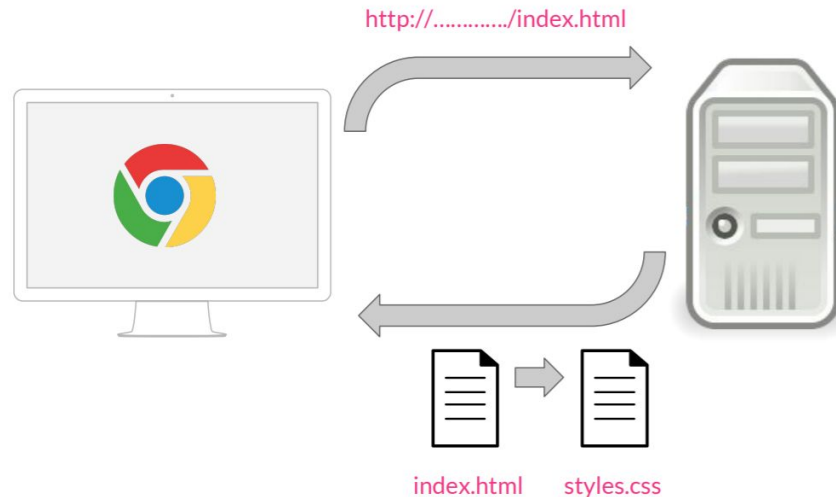


2. UNA DEFINICIÓN

CSS ...es un lenguaje de **diseño gráfico** para definir y crear la presentación de un documento estructurado escrito en un lenguaje de marcado. Es muy usado para establecer el **diseño visual de los documentos web**, e interfaces de usuario escritas en HTML...

3.FUNCIONAMIENTO

¿QUÉ SUCEDE CUANDO VISITO UNA WEB?



4. Referencias

Para aprender:

W3C SCHOOL

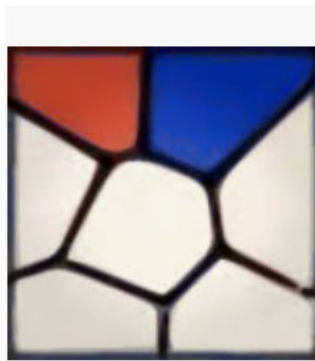
<https://www.w3schools.com/>



A series of five dark blue horizontal bars of equal length, stacked vertically on the left side of the slide.

ENTORNO DE TRABAJO PARA EL CURSO





CHEATSHEETS!

<https://web.stanford.edu/group/csp/cs21/htmlcheatsheet.pdf>

<https://websitesetup.org/css3-cheat-sheet/>





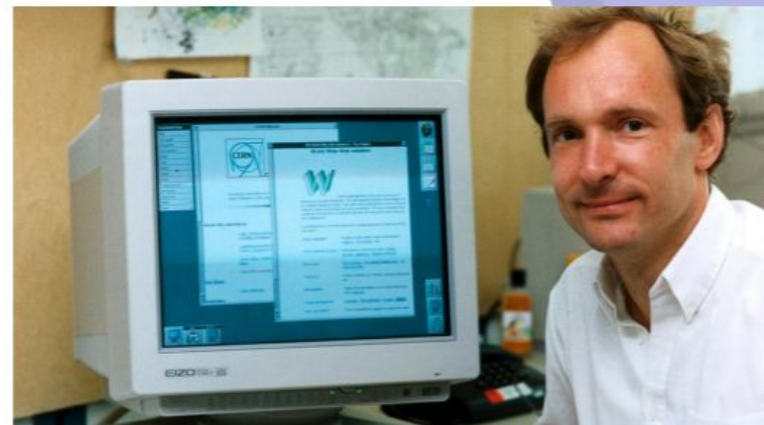
ENTORNO DE TRABAJO PARA EL CURSO



EVOLUCIÓN DE HTML

HTML 1.0 (1991)

- ▶ Creado por Tim Berners Lee en el CERN para que la comunidad científica compartiera información
- ▶ Derivado de SGML
- ▶ 20 etiquetas / 13 aún perduran
- ▶ Primer navegador Nexus



EVOLUCIÓN DE HTML

HTML 2.0 (1994)

- ▶ Primera especificación oficial
- ▶ Incluía todo HTML 1.0
- ▶ Aparecen nuevos navegadores Lynx, Cello, Mosaic
- ▶ 19 nuevas etiquetas (imágenes, formularios etc..)
- ▶ que asocia un documento con un DTD para validarlo

EVOLUCIÓN DE HTML

HTML 3.2 (1997)

- ▶ Fue un recomendación
- ▶ 1996 Internet Explorer
- ▶ Se añaden tablas, mapas etc...
- ▶ Desaparecen etiquetas como marquee y blink (MS)
- ▶ Primera versión desarrollada enteramente por W3C
- ▶ Aparece CSS y los navegadores empiezan a adoptarlo



EVOLUCIÓN DE HTML

HTML 4.01 (1999)

- ▶ Fue un recomendación
- ▶ Ya incluye hojas de estilos
- ▶ Se quitan etiquetas de estilos de versiones anteriores
- ▶ Se mejora la presentación de fuentes, fondos y colores



EVOLUCIÓN DE HTML

XHTML 1.0 (2000)


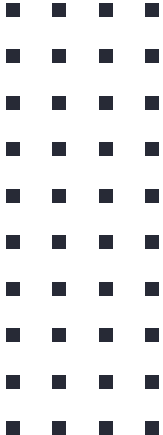

- ▶ Nueva versión de HTML con la rigidez de XML
- ▶ Pocas etiquetas se dejan atrás y básicamente es un conjunto de reglas sobre cómo escribir adecuadamente
- ▶ Validadores



EVOLUCIÓN DE HTML

HTML 5.0 (2014)

- ▶ Fue un recomendación y las sucesivas versiones también (aún no es estándar)
- ▶ Comenzó a desarrollarse por las pocas posibilidades de aplicaciones complejas de XHTML
- ▶ Etiquetas Semánticas
- ▶ APIs
- ▶ Simplifica DOCTYPE, Link, Script
- ▶ Mejora formularios



Etiquetas, atributos y comentarios

Etiquetas

Anteriormente hemos definido HTML como un **lenguaje de marcas o etiquetas** pero, ¿qué es eso de una etiqueta? y ¿qué tipos de etiquetas tengo?

Básicamente podemos distinguir entre **dos tipos de etiquetas**:

Etiquetas con contenido

Etiquetas sin contenido

Etiquetas con contenido.

Son etiquetas que tiene tres partes (por este orden):

Apertura de la etiqueta

Contenido de la etiqueta

Cierre de la etiqueta

Algo así:

`<etiqueta>`

Contenido de la etiqueta

`</etiqueta>`

HTML tiene un número limitado de etiquetas y no es necesario conocer todas. Con una lista más o menos pequeña podemos construir la gran mayoría de páginas web. En caso contrario, siempre podemos visitar las referencias.



Etiquetas con contenido.

Un ejemplo de etiqueta de HTML con contenido sería un párrafo:

<p>

HOLA MUNDO

</p>

<p> Sería la apertura

HOLA MUNDO sería el contenido, en este caso sólo texto

</p> Sería el cierre de la etiqueta

Etiquetas sin contenido

Son etiquetas que sólo tienen parte de apertura y carecen de contenido. Pueden estar cerradas o no, aunque yo recomiendo que se cierren. De esta manera:

`<etiqueta >`

`<etiqueta />`

Un ejemplo de etiqueta HTML sin contenido sería una imagen:

``

Atributos

Las etiquetas pueden tener atributos de los que nos interesa saber lo siguiente:

- Proporcionan información adicional sobre la etiqueta.
- Las etiquetas puede tener o no tener atributos e incluso tener más de uno.
- Siempre se añaden en la etiqueta de apertura.
- Hay atributos generales (para todas) o específicos (para algunas).
- Se representan `nombre_atributo="valor_atributo"`.

Un ejemplo:

```

```

Tipos de etiquetas

Hay muchos tipos de etiquetas pero a lo largo de este curso algunas de las que vamos a ver son:

- Etiquetas de cabecera
- Etiquetas de texto
- Etiquetas de imágenes, tablas, listas y enlaces
- Etiquetas multimedia
- Etiquetas semánticas
- Y alguna más ...

Comentarios

Los comentarios son textos que ponemos dentro de los archivos HTML que no son mostrados por los navegadores y que sirven para dar información adicional o explicaciones sobre el código HTML desarrollado.

Los comentarios van encerrados en esta estructura `<!-- -->` y un ejemplo sería:

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

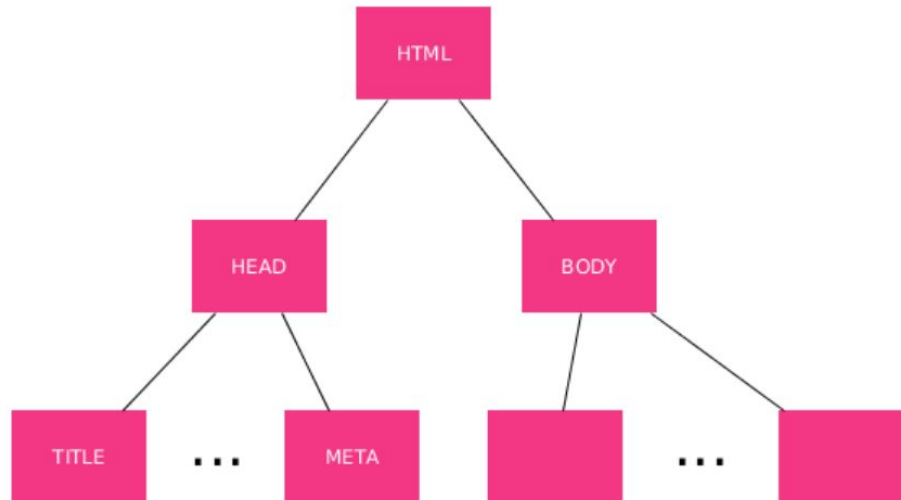


ESTRUCTURA DE UNA WEB



1. EL ARBOL DOM

La estructura de una página Web tiene forma de árbol genealógico con un único nodo raíz:



Esto significa lo siguiente:

Hay una **etiqueta padre** de todo, la etiqueta **html** y entre la apertura y el cierre de esta etiqueta meteremos el resto de nuestra página.

La etiqueta html tiene **dos hijos**, la etiqueta **head** que es la cabecera, que ya veremos qué elementos contiene, y la etiqueta **body** que es la que en realidad contiene los elementos de mi web.

A su vez esas dos etiquetas anteriores puede tener sus propios hijos y así sucesivamente.

<!DOCTYPE html>

Declaración del tipo de documento.

Define que este documento es un documento HTML5

```
<!DOCTYPE html>
<html lang="es">
<head>
  <!--Esto no se ve -->
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Mi primera página</title>
</head>
<body>
  <!-- Esto tampoco se ve -->
  <h1>HOLA MUNDO</h1>
</body>
</html>
```



LA CABECERA DE MI PÁGINA



LA CABECERA (HEAD)

- ▶ No representa contenido alguno
- ▶ Contiene metadata que son elementos que describen la información de la página
- ▶ Contiene enlaces a hojas de estilos y scripts usados por nuestra página.

Etiquetas cabecera

<title>

<style>

<meta>

<link>

<script>

<base>

Etiqueta <meta>

Charset= juego de caracteres

Si quiero que mi página web se vea bien en todos los navegadores independientemente del idioma por defecto, necesito:

<meta charset="UTF-8">

Juego de caracteres universal.

Etiqueta <meta>

Puedo describir que es lo que hay en mi página Web, para los buscadores como google.

<meta name="description" content=".....">

Etiqueta <meta>

Puedo incluir las palabras clave (keyword) para categorizar tu Web, importante porque es lo que leen los buscadores.

<meta name="keywords" content="k1,k2...">



Etiqueta <meta>

Puedo incluir el autor de la página Web.

<meta name="author" content=".....">



Etiqueta <meta>

Para que la página no se te descuadre y se escale al tamaño del dispositivo.

<meta name="viewport" content="width=device -width, initial-scale=1.0">

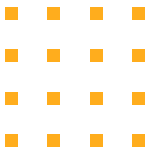



Etiqueta <meta>

Si quisiéramos que nuestra página se refrescara cada 30 segundos (páginas de deportes, loterías, apuestas...)

<meta http-equiv="refresh" content="30">





Añadiendo Contenido

Etiquetas Básicas.



ELEMENTOS DE TEXTO



1.ENCABEZADOS

Son etiquetas que nos van a permitir añadir “títulos” o encabezados a distintas secciones de nuestra página.

Estas etiquetas tienen el siguiente formato:

`<h1>....</h1>`

`<h2>....</h2>`

`<h3>....</h3>`

`<h4>....</h4>`

`<h5>....</h5>`

`<h6>....</h6>`

2. ETIQUETAS DE FORMATO

Hay multitud, todas le dan cierto estilo al texto que contienen y destacaremos las siguientes:

`...` para poner un texto en negrita.

`<i>...</i>` para poner un texto en cursiva.

`...` para mostrar un texto tachado.

`...` para poner un texto con énfasis (similar a cursiva).

`^{...}` para poner un texto como superíndice de otro texto.

`_{...}` para poner un texto como subíndice de otro texto.

`<u>...</u>` para poner un texto subrayado (nuevo en html5).

2. ETIQUETAS DE FORMATO

`<q>...</q>` para mostrar una pequeña cita.

`<cite>...</cite>` para mostrar el título de una referencia bibliográfica.

`<time>...</time>` para mostrar horas.

`<address>...</address>` para mostrar direcciones.

`<blockquote>...</blockquote>` para poner citas largas.



3. OTROS ELEMENTOS

`
` Salto de línea

`<hr/>` Separación de Tema

`<p>...</p>` Párrafo





Imágenes



Imágenes

Imágenes simples

La inclusión de imágenes simples se viene haciendo desde las primeras versiones de HTML con la etiqueta sin contenido ``

Los atributos más comunes que le podemos poner a esta etiqueta son:

src: que indica la ruta en la que se encuentra el archivo de la imagen.

alt: un texto alternativo para describir la imagen en caso de que no se cargue o para dispositivos especiales para usuarios con discapacidad visual (por ejemplo)

width: para especificar la anchura de la imagen . Si no se escogerá la anchura propia de la misma.

height: para especificar la altura de la imagen . Si no se escogerá la altura propia de la misma.

Rutas

Relativas: Toman como base el directorio en el que se encuentra nuestro fichero. Son las recomendadas.

```

```

Absolutas: Toman como base el directorio raíz de mi equipo (/ en Linux y c:\ en Windows). Cuidado, sólo funcionarán en tu mismo equipo.

```

```

Url: La dirección de Internet de un recurso (fichero, imagen etc..). Puede desaparecer y entonces dejará de mostrarse en nuestra web.

```

```





Enlaces



Enlaces

Los enlaces, que se representan mediante el uso de la etiqueta `<a>` es una de las construcciones más importantes en HTML. Esta etiqueta puede tener varios atributos, de los cuales los más importantes son:

- `href`: que es la dirección de Internet de destino (ya sea otra página web, un imagen, un fichero o lo que sea).
- `target`: que indica dónde voy a abrir ese enlace. Si no pongo nada se abrirá en la misma pantalla y si le doy el valor `target="_blank"` se abrirá en una nueva ventana de mi navegador.

Enlaces dentro de la misma página

Puedo enlazar enlaces dentro de la misma página con construcciones como la siguiente:

...

```
<a href="#contacto">Contacto</a>
```

...

```
<section id="contacto">
```

.....

```
</section>
```

....





Problema 1:

Realiza un documento HTML, que visualizado en el navegador tenga el



Augusto



Listas

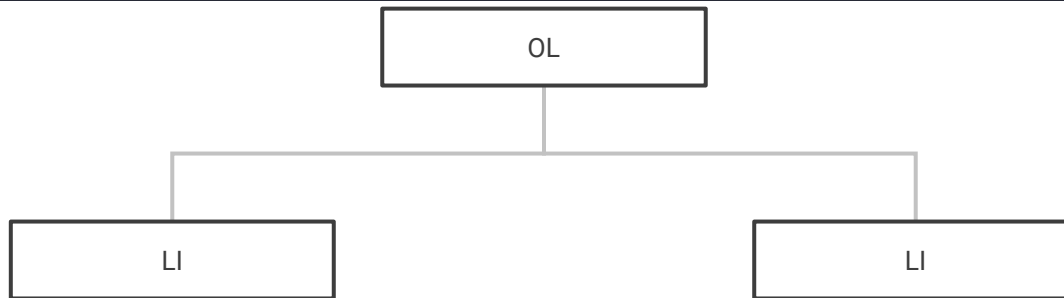


Listas

Las listas son una de las construcciones más usadas a la hora de elaborar textos, no sólo en HTML. Pueden ser de 3 tipos:

- **Listas Numeradas:** que son aquellas que expresan un orden entre los diferentes elementos de la lista. Este orden podrá ser numérico, alfabético etc..
- **Listas no numeradas:** que simplemente muestra los elementos de la lista uno tras otro.
- **Listas de definición:** que muestran diversos términos junto con su definición.

Lista Numeradas



<!-- Listas numeradas -->

<h3>Lista Numerada - Ordered List (ol)</h3>

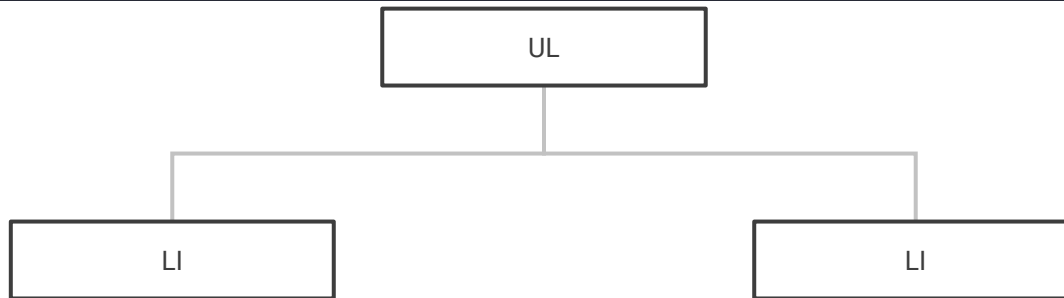
Primer elemento

Segundo elemento

Tercer Elemento

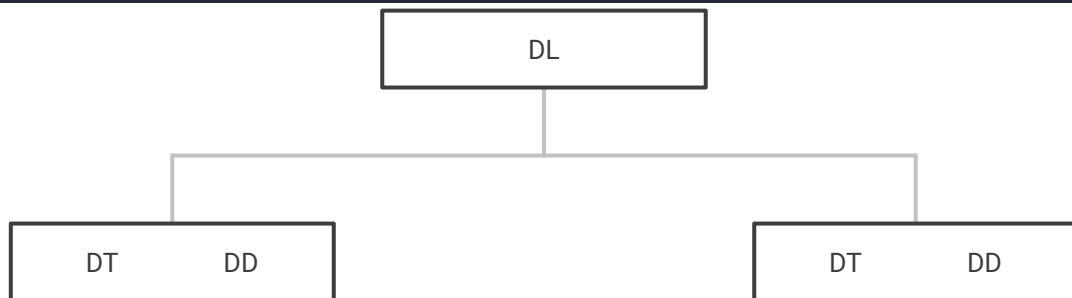
Cuarto elemento

Lista No Numeradas



```
<!-- Listas no numeradas-->
<h3>Lista NO Numerada - Unordered List (ul)</h3>
<ul>
  <li>Primer elemento</li>
  <li>Segundo elemento</li>
  <li>Tercer Elemento</li>
  <li>Cuarto elemento</li>
</ul>
```

Listas de definición



!-- Listas de definición-->

```
<h3> Lista de descripción - Description List (dl)</h3>
```

```
<dl>
```

```
<dt>Primer término</dt>
```

```
<dd>Definición del primer término</dd>
```

```
<dt>Segundo término</dt>
```

```
<dd>Definición del segundo término</dd>
```

```
<dt>Tercer término</dt>
```

```
<dd>Definición del tercer término</dd>
```

```
<dt>Cuarto término</dt>
```

```
<dd>Definición del cuarto término</dd>
```

```
</dl>
```

Lenguajes



Problema

Realiza un documento sobre el siguiente aspecto:

- i. Lenguajes de marcas
 - a. XML
 - b. HTML
 - c. CSS
- ii. Lenguajes de programación
 - a. C++
 - b. Java
 - c. Python

Tipos de redes

- Topología

Bus

Los equipos están conectados a un bus central

Estrella

Se conectan todos los nodos a un nodo central

Árbol

Vista como una colección de redes en estrella ordenadas en una jerarquía

- Extensión

- 1. PAN
- 2. LAN
- 3. WAN



Tablas



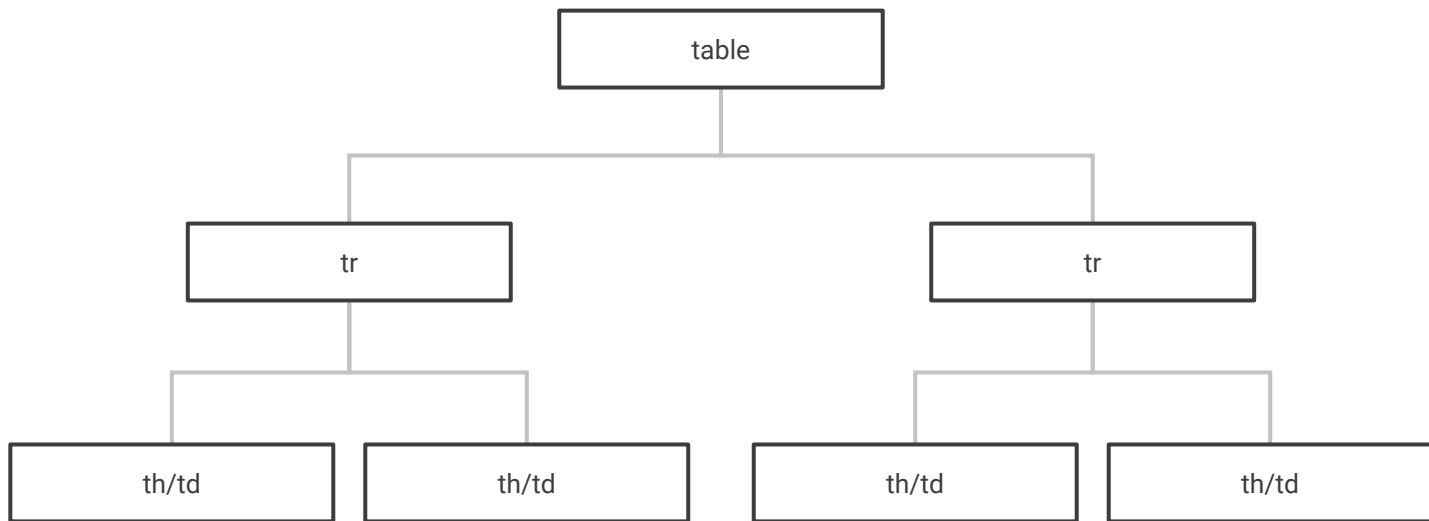
Tablas Simples

La estructura del árbol DOM más simple para una tabla sería similar a la siguiente:

Una etiqueta **<table>** que contiene toda la tabla.

Como hijos directos, tantas etiquetas **<tr>** como filas queramos que tenga nuestra tabla.

Dentro de cada filas tantas etiquetas **<th>** o **<td>** como celdas queramos que tengan nuestras filas. La única diferencia entre estas dos es que el contenido en la primera se presenta centrado y el texto en negrita.



NOTA: Si no coinciden el número de celdas en todas las filas veremos que suceden cosas “extrañas”.

NOTA: La anchura de las celdas de una misma columna será la anchura de la más ancha de la columna.

NOTA: La altura de las celdas de una misma fila será la altura de la más alta de la fila.





Problema 3

Realiza la siguiente tabla:

Contactos

| Nombre | Apellidos | Dirección | Teléfono | Email |
|--------|-----------|-----------|-----------|------------------|
| Pedro | Pérez | Ruíz | 666223455 | pepe@gmail.com |
| Ana | López | Fuentes | 66778855 | alopez@gmail.com |
| María | Fernández | Hoz | 768223455 | mafe@gmail.com |
| Sara | Gallardo | Río | 686283955 | saga@gmail.com |

Tabla Completa

Existe además una forma más precisa y completa de construir tablas. Una forma que puede contener (aunque no es obligatorio) otras etiquetas dentro de la etiqueta raíz `<table>`. Estas nuevas etiquetas pueden ser:

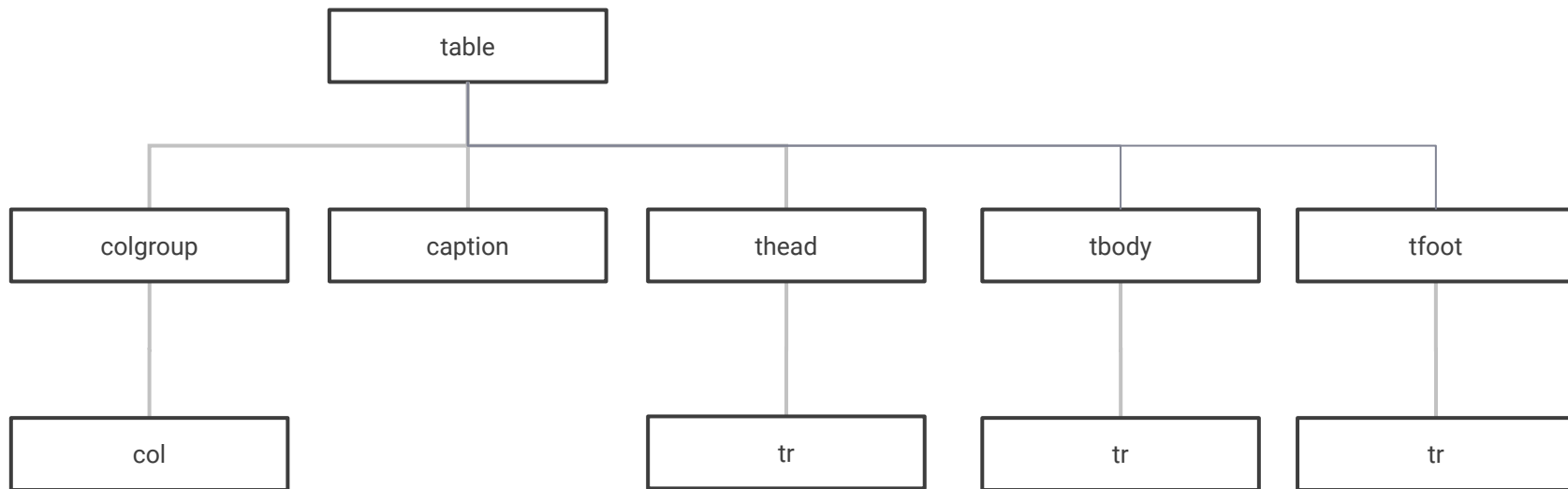
`<colgroup>` que nos permite agrupar las columnas para darles estilos. Cada uno de esos grupos lo definiremos usando una etiqueta con un atributo `span` para definir el número de columnas de cada grupo.

`<caption>` para añadir un título o leyenda a la tabla en la parte superior.

`<thead>` que contendrá la fila (usualmente) o filas que sean la cabecera de una tabla.

`<tbody>` que es donde pondremos las filas que son el contenido propiamente dicho de la tabla, el cuerpo.

`<tfoot>` que es donde pondremos las filas que son el pie de nuestra tabla.





Problema 4A

Realiza la siguiente tabla:

Tabla Completa

Alumnos LMSGI

| Nombre | Apellidos | Dirección | Teléfono | Email |
|--------|-----------|-----------|-----------|---------------|
| Ana | Pérez | Río | 666444666 | ana@email.es |
| Luis | López | González | 333225522 | luis@email.es |
| María | Fernández | Mar | 66337733 | mar@email.es |
| Sara | Gallardo | Fuentes | 44444444 | sgal@email.es |

Pie de la tabla donde pongo
más texto para que se vea
como crecen



Tablas Complejas

Con todo lo explicado anteriormente podemos ver que las tablas que conseguimos son relativamente sencillas. En la vida real, nos encontraremos con estructuras tabulares más complejas. Éstas también se pueden construir en HTML utilizando los siguientes atributos en las etiquetas y , es decir, en las celdas.

rowspan: me va a permitir que una celda ocupe más de una fila.

colspan: me va a permitir que una celda ocupe más de una columna.





Problema 4b:

Realiza los cambios necesarios para la siguiente tabla:

Tabla Completa

Alumnos LMSGI

| Nombre | Apellidos | Dirección | Teléfono | Email |
|--------|-----------|-----------|-----------|---------------|
| Ana | Pérez | Río | 666444666 | ana@email.es |
| Luis | López | González | 333225522 | luis@email.es |
| María | Fernández | Mar | 66337733 | mar@email.es |
| Sara | Gallardo | Fuentes | 44444444 | sgal@email.es |

Pie de la tabla donde pongo más texto para que se vea como crecen



Formulario



¿Qué es un formulario?

¿Donde encontramos formularios? Formularios de registro, usuario y contraseña, solicitudes etc...

- Los formularios son elementos, que como los enlaces, permiten una interacción del usuario con la página web.
- Su tarea principal es recoger información. El usuario, por tanto, debe introducir esa información en los campos del formulario.
- Una vez se ha recogido esa información el formulario la enviará al servidor, por correo o donde decida el programador para ser tratada, mostrada y/o almacenada.



Estructura del formulario

Elemento padre, la etiqueta **<form>** que va a contener en su interior todos los elementos que conformen nuestros formularios.

Esta etiqueta puede tener varios **atributos** de entre los que destacamos:

- **method** que indica cómo se va a pasar la información al destino. Puede ser por GET (se ve la información en la barra del navegador) y por POST (no se ve y es la opción por defecto).
- **action** que indica el destino de nuestros datos: URL o dirección de Internet, correo electrónico...
- Existen más opciones. Os invito a visitar la documentación de referencia.

```
<form method="POST" action="formulario.html">
```

Labels e Input

ETIQUETAS:

Normalmente para la recogida de información los formularios usando etiquetas: **<input>**.

Para saber qué campo estamos rellenando ponemos una etiqueta para dar nombre: **<label>**

ATRIBUTOS:

Asociamos ambas con los atributos: **for** (label) e **id** (input)

El campo input además debe llevar el atributo **name** para identificar el campo dentro de un formulario. Este atributo nos va a resultar necesario para poder enviar la información

```
<label for="nombre">Nombre:</label>  
<input type="text" id="nombre" name="nombre"/>
```

Tipos de input

<input type="text"...> Muestra un campo de entrada de texto de una sola línea.

<input type="radio"...> Muestra un botón de opción (para seleccionar una de las muchas opciones)

<input type="checkbox"....> Muestra una casilla de verificación (para seleccionar cero o más de muchas opciones)



Listas desplegables

Al hacer click sobre ellos nos muestran varias opciones de las que podremos escoger una o varias.

Usaremos las etiquetas **<select>** como etiqueta padre y una etiqueta **<option>** para cada una de las opciones que tengamos en la lista desplegable.

```
<label for="provincia">provincia</label>
<select id ="provincia" name="provincia">
  <option>Cantabria</option>
  <option>Asturias</option>
  <option>León</option>
</select>
```

Áreas de Texto

Son campos de formulario para meter texto largos. Tienen dos atributos interesantes.

rows: para indicar el número de filas que tiene el campo.

cols: para indicar el número de columnas que tiene el campo.

```
<textarea cols="10" rows="10">
```

```
</textarea>
```



Botones

La etiqueta **<button>**

El atributo más importante que tienen es **type** que puede tomar tres valores:

- **button**: se comporta como un botón que puede estar presionado o no.
- **reset**: al hacer click borra todos los datos introducidos previamente en el formulario y pone los valores por defecto.
- **submit**: al hacer click envía los datos introducidos al destino que hemos puesto en el action del formulario.

```
<button type="button">Haz clic</button>
```

```
<button type="reset">Haz clic para resetear</button>
```

```
<button type="submit">Haz clic para enviar</button>
```

Agrupando campos

Para agrupar distintos campos utilizamos la etiqueta **<fieldset>** junto a la etiqueta **<legend>** para dar nombre al grupo:

```
<fieldset>  
  <legend>Datos Personales</legend>  
  ....  
</fieldset>
```


Atributos interesantes

Required: campo obligatorio

Readonly: Campo de solo lectura

Disable: deshabilitar un campo

Placeholder: quiero meter un campo introductorio



Multimedia

- La etiqueta raíz será `<audio>` o `<video>` según lo que queramos mostrar en nuestra navegador.
- Dentro de esta etiqueta raíz tendremos tantas etiquetas `<source>` como formatos distintos de ese mismo elemento ofrezcamos. Si ofertamos varias fuentes el navegador decidirá cuál reproduce.

Etiqueta `<source>` tiene dos atributos fundamentales:

- `src` para indicar dónde está el fichero multimedia (una ruta).
- `type` para indicar el tipo de fichero, por ejemplo: `type="audio/mp3"` o `type="video/webm"`

Las etiquetas <audio> y <video> pueden tener los siguientes atributos.

- controls para mostrar los controles gráficos de reproducción (play, stop, pause, volumen).
- loop si queremos que se reproduzca en bucle de manera infinita.
- muted si queremos que se reproduzca en silencio.

Adicionalmente la etiqueta <video> puede tener más atributos:

width y height para indicar la anchura y altura del vídeo.

Etiquetas semanticas

Definen el tipo de contenido que tienen ciertas secciones de una página web:

- Header: Cabecera de la página web.
- Footer: Pie de la página web.
- Main: Sección principal, el grueso de información.
- Section: Secciones en que se divide el main.
- Article: Zona de la web que tiene sentido por sí misma.
- Aside: Información adicional.
- Summary: resumen
- Nav: panel de navegación.

Se comportan con la etiqueta <div> pero aportan significado al contenido

Más etiquetas interesantes

`<iframe>`: para meter una página web dentro de otra página web

`<canvas>`: se utiliza para dibujar, usa JavaScript

`<div>`: maquetar

`<script>`: enlazar con ficheros escritos en JavaScript

`<object>`: define un contenedor para un pdf o un recurso externo.

