



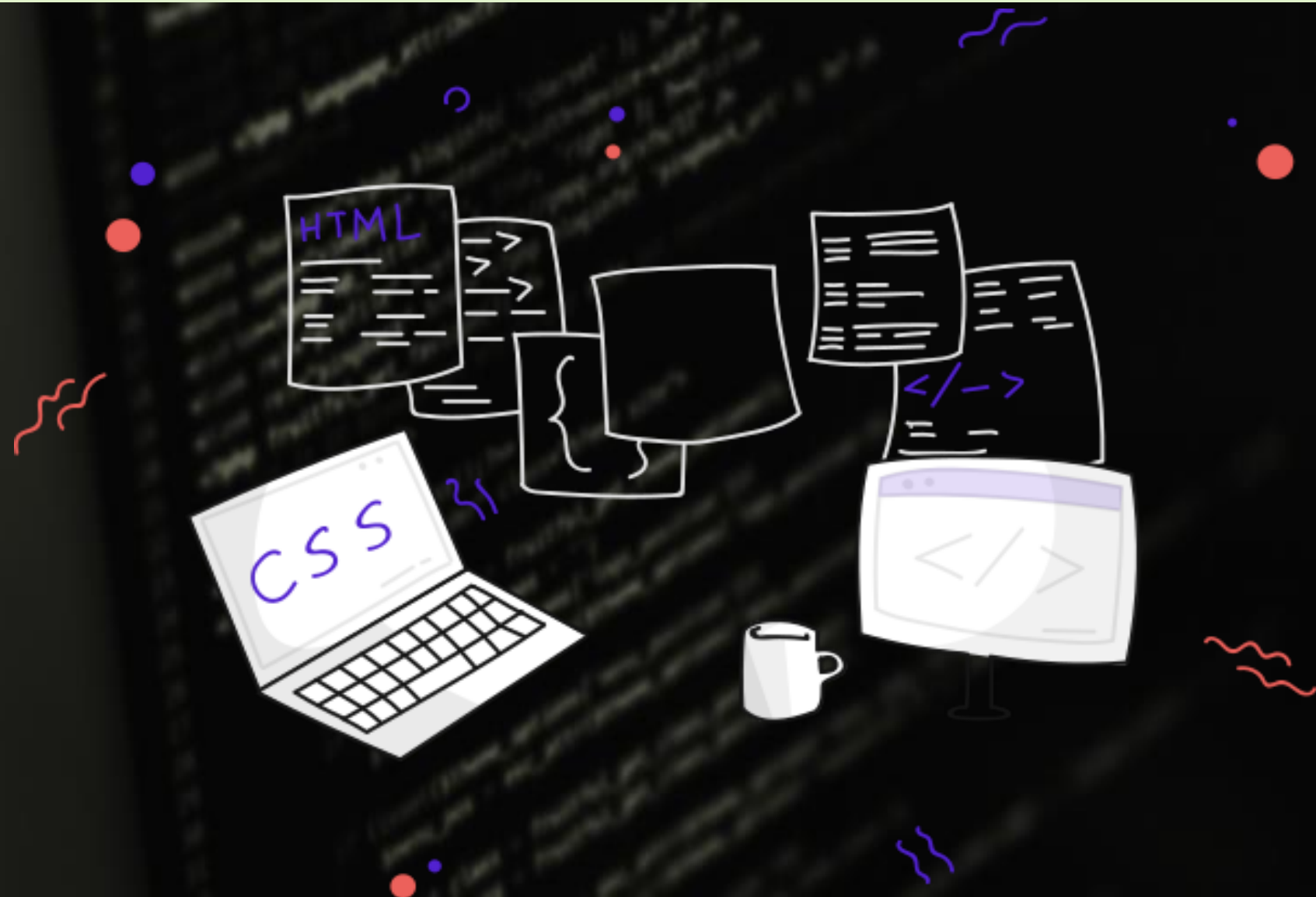
EFA
MORATALAZ

*1º CFGS Desarrollo de
Aplicaciones Web*

Lenguajes de Marcas y Sistemas de Gestión de la información

María Fátima Sánchez Fuentes

UT2 – LENGUAJES PARA LA VISUALIZACIÓN DE INFORMACIÓN HTML





EFA
MORATALAZ

*1º CFGS Desarrollo de
Aplicaciones Web*

Lenguajes de Marcas y Sistemas de Gestión de la información

INDICE

UT2 – LENGUAJES PARA LA VISUALIZACIÓN DE INFORMACIÓN HTML

1. HTML.

1.1 Estructura.

1.2 Agrupación de contenido.

1.3 Secciones.

1.4 Semántica de texto.

1.5 Contenido incrustado.

1.6 Información tabulada.

1.7 Formularios.

1.8 Atributos.

1.9 Comentarios. Caracteres. Detalles de estructura.

1.10 DOCTYPE.

1.11 Documentos HTML correctos.

1.12 Referencias absolutas y relativas.

HTML

1

HTML nace en el año 1991 de la mano del científico Tim Berners-Lee (conocido como el padre de la web). En un principio era denominado “*HTML Tags*”.

La W3C (World Wide Web Consortium) ha ido definiendo **nuevas versiones** de HTML con el paso de los años.

- HTML 2.0.
- HTML 3.2.
- HTML 4.01.
- HTML 5.
- HTML 5.1.
- HTML 5.2.



Estructura básica de fichero HTML

```
<!doctype html>
<html lang="es">
  <head>
    <title>Título de documento</title>
    ... otras etiquetas
  </head>
  <body>
    <p class="parrafo"> Este es un parrafo </p>
    ... otras etiquetas
  </body>
</html>
```

Estructura



- El **elemento raíz <html>** es la raíz del documento HTML bien formado y actúa de contenedor para TODOS los elementos que conforman el documento.

Como se puede apreciar el elemento <html> **encierra al resto de elementos**, a excepción del <!DOCTYPE>, que no es considerado un elemento.

La etiqueta **<!DOCTYPE HTML>** especifica y declara la **versión** de HTML.

EJEMPLO:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Página web muy básica</title>
  </head>
  <body>
    <h1>¡Bienvenid@!</h1>
  </body>
</html>
```

- El **elemento <head>** es el contenedor de aquellos elementos que contienen **metadatos** del documento, como por ejemplo, el título, hojas de estilo, el autor...
 - Etiqueta **<title>** → *Especifica el título que se muestra en la barra del explorador.*
 - Etiqueta **<meta>** → *Sirve para especificar el código de caracteres usado.*
 - Etiqueta **<style>** → *Se usa para la definición de estilos.*
 - Etiqueta **<link>** → *Se utiliza para vincular hojas de estilo externas.*

EJEMPLO:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Lenguajes de marcas curso 22/23</title>
    <meta charset="utf-8">
    <meta name="author" content="MFS">
    <link rel="stylesheet" type="text/css" href="styles.css">
    <style type="text/css">
      h1 {
        color: gray;
      }
    </style>
  </head>
  <body>

</body>
</html>
```

- **<title>**

La directiva **<title>** sirve para especificar el título del documento y es visualizada en la barra de título de la ventana correspondiente al programa navegador.

```
<title>Lenguajes de marcas curso 22/23</title>
```

- **<meta>**

Las **<meta>** son identificadores ocultos, **no se muestran**.

También nos ayudan a mejorar nuestra posición en el listado de los resultados de los mayores buscadores.

Las **<meta>** se deben de colocar en la **cabecera** de la página web, entre **<head>** y **</head>**.

```
<meta charset="utf-8">  
<meta name="author" content="MFS">
```


- `<link>`

El método más habitual de añadir una **hoja de estilo externa** a una pagina web es utilizar la directiva `<link>` de HTML.

La directiva se escribe de esta forma:

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

La directiva `<link>` es una directiva vacía, pues solo tiene solamente una etiqueta de inicio. Para terminar la etiqueta tenemos que añadir una barra de cierre (`/>`).

- **`rel="stylesheet"`**: indica el **tipo de vínculo** (en este caso a una hoja de estilo).
- **`type="text/css"`**: permite al navegador saber de qué **tipo de datos** se trata (El contenido es CSS).
- **`href`**: **ubicación del archivo externo** de CSS en el sitio. El valor de esta propiedad es un URL y variará en función de dónde se encuentra su archivo de CSS.

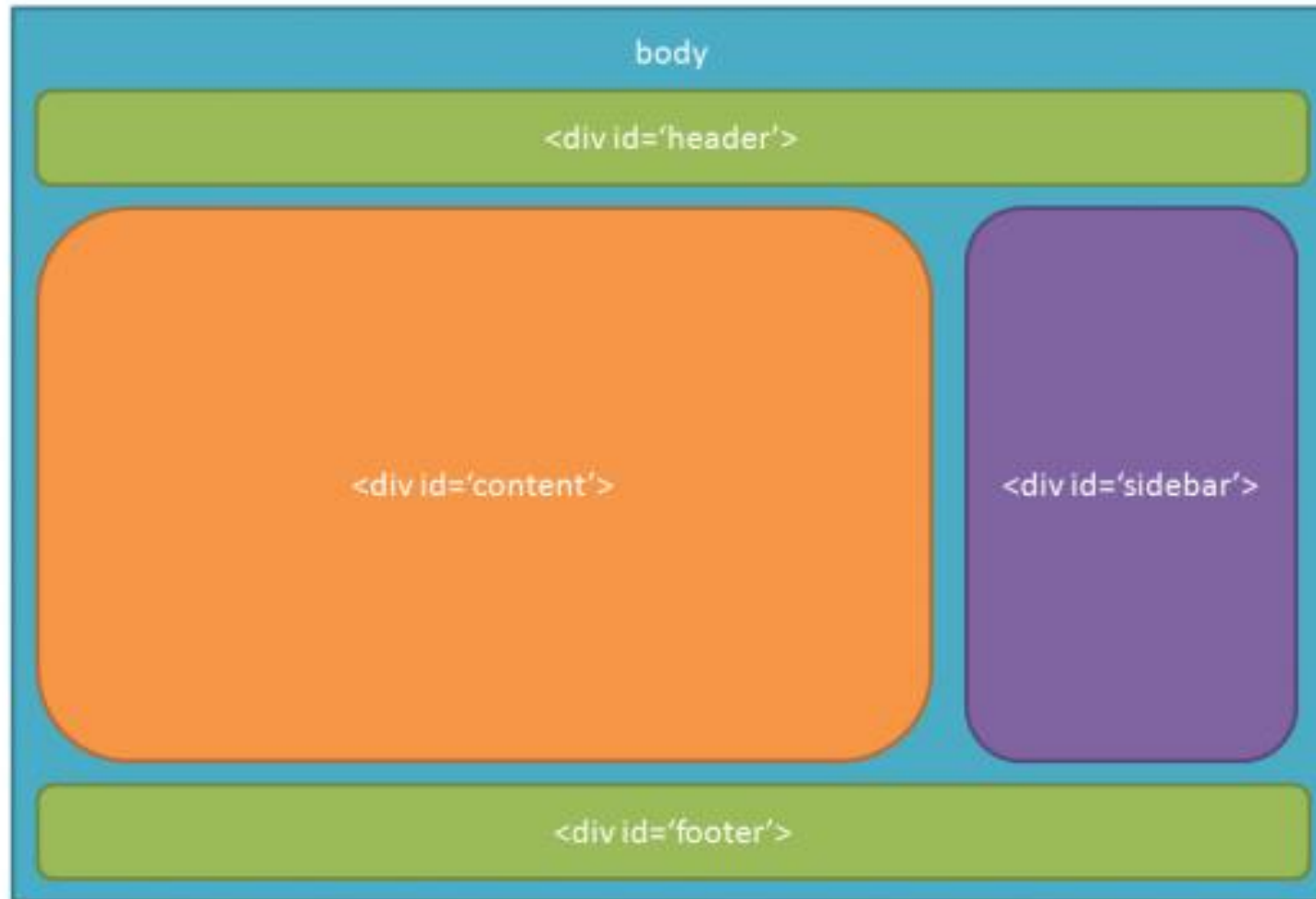
- El elemento **<body>** es el contenedor de los aspectos representables de un documento, es decir, es la sección donde se especifican los elementos, sus atributos...

Para diferenciar secciones en el cuerpo se hace uso de la etiqueta **<div>** (*diapositiva 15*).

EJEMPLO:

```
<!DOCTYPE html>
<html>
  <head>
    <title>Lenguajes de marcas curso 22/23</title>
    <meta charset="utf-8">
  </head>
  <body>
    <p>Esto es un elemento</p>
  </body>
</html>
```

Para diferenciar secciones en el cuerpo se hace uso de la etiqueta **<div>** (*diapositiva 15*).



De modo que....

Raíz

```
<!DOCTYPE html>
```

```
<html>
```

Cabecera

```
<head>
```

```
<title>Lenguajes de marcas curso 22/23</title>
```

```
<meta charset="utf-8">
```

```
</head>
```

```
<body>
```

```
<p>Esto es un elemento</p>
```

```
</body>
```

```
</html>
```

Cuerpo

Agrupación de contenido

1.2

A continuación se exponen los **elementos** de **AGRUPACIÓN DE CONTENIDO** más relevantes:

- El elemento **<p>** representa un párrafo (bloque de texto compuesto por oraciones).

EJEMPLO:

HTML	CSS	OUTPUT
<pre>1 <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.</p> 2 3 </pre>		<p> Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged.</p>

- El elemento **<div>** permite crear **DIVISIONES** en el documentos, es decir, bloques. Es muy útil para el posicionamiento.

EJEMPLO:

HTML	CSS	OUTPUT
<pre>1 <div title="Lorem ipsum"> 2 3 <p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. </p> 4 5 <p>It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout.</p> 6 7 </div></pre>		<p>Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book.</p> <p>It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout.</p>

- El elemento **** representa una lista **no ordenada** de elementos. Los elementos que contiene esta lista se representan por el elemento ****.

EJEMPLO:

		OUTPUT
HTML	CSS	Lista de la compra
<pre>1 <p>Lista de la compra</p> 2 3 4 5 Manzanas 6 7 Sal 8 9 Helados 10 11 </pre>		

- El elemento **** representa una lista **ordenada** de elementos. Los elementos que contiene esta lista se representan por el elemento ****.

EJEMPLO:

	HTML	CSS	OUTPUT
1 2 3 4 5 6 7 8 9 10 11	<pre><p>Pasos para hacer manzana de azúcar:</p> Calentar en microondas azúcar Sacar el azúcar fundido y meter las manzanas Meter las manzanas en el congelador </pre>		<p>Pasos para hacer manzana de azúcar:</p> <ol style="list-style-type: none">1. Calentar en microondas azúcar2. Sacar el azúcar fundido y meter las manzanas3. Meter las manzanas en el congelador

Secciones

1.3

A continuación se exponen los **elementos** de **SECCIONES** más relevantes:

- **<h1>**, **<h2>**, **<h3>**, **<h4>**, **<h5>** y **<h6>** establecen encabezados. Son títulos que definen secciones implícitas en el documento.

EJEMPLO:

```
<h1>Encabezado nivel 1</h1>
```

```
<h2>Encabezado nivel 2</h2>
```

```
<h3>Encabezado nivel 3</h3>
```

```
<h4>Encabezado nivel 4</h4>
```

```
<h5>Encabezado nivel 5</h5>
```

```
<h6>Encabezado nivel 6</h6>
```

Semántica de texto

1.4

A continuación se exponen los **elementos** de **SEMÁNTICA DE TEXTO** más relevantes:

- **<a>** permite crear **ENLACES** a otras páginas de internet, archivos...

EJEMPLO:

HTML	CSS	OUTPUT
<pre>1 <p>Entra a la página oficial de la EFA</p> 2 3 EFA 5</pre>		<p>Entra a la página oficial de la EFA</p> <p>EFA</p>

Cuando una página web es muy larga o tiene muchos apartados, es útil crear enlaces que nos permitan movernos directamente al lugar que nos interesa. Estos reciben el nombre de **MARCADORES**.

La **estructura** de los marcadores es muy similar a cuando creamos un hipervínculo a otra página.

Se suele usar para crear una **especie de índice** y queremos que se pueda acceder a los distintos capítulos desde ese índice.

PASOS:

1. Establecer el nombre del marcador.

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

2. Esta etiqueta puede contener texto/imagen/blanco en su interior.

```
<a href="#nombre_marcador">Ir a pagina de login</a>
```

EJEMPLO:

```
<!DOCTYPE html>
<html>
<head>
<title> MFS </title>
</head>

<body>
<a href="#nombre_marcador1">Titulo 1</a> |
<a href="#nombre_marcador2">Titulo 2</a> |
<a name="nombre_marcador_arriba"></a>

<a name="nombre_marcador1"></a>
<h4>PARTE NOMBRE_MARCADOR1</h4>
<p>BIENVENID@</p>

<a name="nombre_marcador2"></a>
<h4>PARTE NOMBRE_MARCADOR2</h4>
<p>Sed ut perspiciatis unde omnis iste natus error

<a href="#nombre_marcador_arriba">Ir arriba</a>

</body>

</html>
```

[Titulo 1](#) | [Titulo 2](#) |

PARTE NOMBRE_MARCADOR1

BIENVENID@

PARTE NOMBRE_MARCADOR2

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium do

[Ir arriba](#)

- **
** inserta un salto de línea.

EJEMPLO:

HTML	CSS	OUTPUT
<pre>1 <p>Te amo sin saber cómo, ni cuándo, ni de dónde,
 2 3 te amo directamente sin problemas ni orgullo:
 4 5 así te amo porque no sé amar de otra manera,
 6 7 sino así de este modo en que no soy ni eres,
 8 9 tan cerca que tu mano sobre mi pecho es mía,
 10 11 tan cerca que se cierran tus ojos con mi sueño.</p></pre>		<p>Te amo sin saber cómo, ni cuándo, ni de dónde, te amo directamente sin problemas ni orgullo: así te amo porque no sé amar de otra manera, sino así de este modo en que no soy ni eres, tan cerca que tu mano sobre mi pecho es mía, tan cerca que se cierran tus ojos con mi sueño.</p>

- **** pone el texto en negrita.

EJEMPLO:

HTML		CSS	OUTPUT
1	<code><p>Es super importante prestar atención en clase para sacar buenas notas y poder graduarse.</p></code>		Es super importante prestar atención en clase para sacar buenas notas y poder graduarse.
2			

- `<i>` poner el texto en cursiva.

EJEMPLO:

HTML		CSS	OUTPUT
1	<code><p> Si tienes dudas, pregunta a la profesora</code>		Si tienes dudas, pregunta a la profesora <i>MFS</i> , en clase o por correo electrónico.
2	<code><i>MFS</i>, en clase o por correo electrónico.</p></code>		

Contenido incrustado

1.5

A continuación se exponen los **elementos** de **CONTENIDO INCRUSTADO** más relevantes:

- **** representa una imagen. Es necesario poner la ubicación del recurso externo en el atributo **src** (lo veremos más adelante).

EJEMPLO:

HTML	CSS	OUTPUT
<pre>1 </pre>		

Información tabulada



A continuación se exponen los **elementos** de **INFORMACIÓN TABULADA** más relevantes:

- **<table>** representa una tabla.

Las tablas actúan de contenedor para otros elementos, entre los que destacan:

- **<th>**: representa la cabecera de la tabla.
- **<tr>**: representa una fila.
- **<td>**: representa una columna.

EJEMPLO:

HTML	CSS	OUTPUT
<pre>1 <table class="default"> 2 <tr> 3 <td>Alejandro Nieto</td> 4 <td>9</td> 5 </tr> 6 <tr> 7 <td>Teresa Jiménez</td> 8 <td>7.05</td> 9 </tr> 10 <tr> 11 <td>Amalia Sánchez</td> 12 <td>9.5</td> 13 </tr> 14 </table></pre>		<div>Alejandro Nieto 9 Teresa Jiménez 7.05 Amalia Sánchez 9.5</div>

Hemos visto una tabla simple... podemos hacer tablas más complejas con los siguientes elementos:

- `<tbody>`: representa el cuerpo de la tabla.
- `<thead>`: representa la sección de encabezado de la tabla.
- `<tfoot>`: representa la sección de pie de la tabla.

HTML

CSS

```

1 <table class="default">
2   <thead>
3     <tr>
4       <th>Alumno</th>
5       <th>Nota</th>
6     </tr>
7   </thead>
8   <tbody>
9     <tr>
10      <td>Alejandro Nieto</td>
11      <td>9</td>
12    </tr>
13    <tr>
14      <td>Teresa Jiménez</td>
15      <td>9</td>
16    </tr>
17  </tbody>
18  <tfoot>
19    <tr>
20      <th>Nota media</th>
21      <th>9</th>
22    </tr>
23  </tfoot>
24 </table>

```

HTML

CSS

```

10 <td>Alejandro Nieto</td>
11 <td>9</td>
12 </tr>
13 <tr>
14   <td>Teresa Jiménez</td>
15   <td>9</td>
16 </tr>
17 </tbody>
18 <tfoot>
19   <tr>
20     <th>Nota media</th>
21     <th>9</th>
22   </tr>
23 </tfoot>
24 </table>

```

OUTPUT

Alumno	Nota
Alejandro Nieto	9
Teresa Jiménez	9
Nota media	9

Formularios



A continuación se exponen los **elementos** de **FORMULARIOS** más relevantes:

- **<form>** inserta un componente diseñado para contener controles para interactuar con el usuario y gestionar los datos **introducidos** por el mismo para **enviarlos al servidor**.

Entre los **principales controles** destacan los siguientes:

- **<button>**
- **<fieldset>**
- **<input>**
- **<keygen>**
- **<object>**
- **<output>**
- **<select>**
- **<textarea>**
- ****

- **<button>** un botón que, cuando es presionado, puede llevar a cabo las siguientes acciones dependiendo de su atributo *type* concreto.
 - *submit* → cuando es presionado envía el formulario al agente procesador.
 - *reset* → cuando es presionado restaura todos los campos del formulario a sus valores iniciales.
 - *button* → no lleva a cabo ninguna acción. *Se suele usar para el control de eventos.*

<button type="submit" value="Submit">Submit</button>

<button type="reset" value="Reset">Reset</button>

- `<button>` `type=button`
- `<button>` `type=reset`
- `<button>` `type=submit`

- **<input>** representa un campo de datos tipado que permite al usuario editar su valor. El atributo *type* define qué tipo de datos va dentro de la casilla de input, entre ellos destacan:

- `<input> type=hidden`

- `<input> type=text`

- `<input> type=search`

- `<input> type=tel`

- `<input> type=url`

- `<input> type=email`

- `<input> type=password`

- `<input> type=date`

- `<input> type=month`

- `<input> type=week`

- `<input> type=time`

- `<input> type=datetime-local`

- `<input> type=number`

- `<input> type=range`

- `<input> type=color`

- `<input> type=checkbox`

- `<input> type=radio`

- `<input> type=file`

- `<input> type=submit`

- `<input> type=image`

- `<input> type=reset`

- `<input> type=button`

Sus atributos permiten ajustar características del *input*, este es un ejemplo del tipo *text* y del tipo *button*.

EJEMPLOS:

HTML	CSS	OUTPUT
<pre>1 <input type="text" id="name" name="name" 2 required 3 minlength="4" maxlength="8" size="10"></pre>		<input type="text"/>

HTML	CSS	OUTPUT
<pre>1 <input type="button" value="Púlsame"></pre>		<input type="button" value="Púlsame"/>

En esta página puedes buscar todos los atributos de cada tipo de *input* en concreto:

<https://developer.mozilla.org/es/docs/Web/HTML/Element/input/checkbox>

- **<textarea>** representa un campo para la entrada de texto multilínea (texto amplio). Sus atributos permiten ajustar características del *textarea*.

EJEMPLOS:

HTML	CSS	OUTPUT
<pre>1 <textarea name="mensaje" placeholder="Comenta qué te parece el post."></textarea> 2</pre>		
HTML	CSS	OUTPUT
<pre>1 <textarea name="mensaje" rows="10" cols="10" placeholder="Comenta qué te parece el post."></textarea> 2</pre>		

<textarea> != <input type="text">

- Textarea → Varias líneas.
- Text → Una línea.

required

Cuando un determinado campo de formulario se debe rellenar **obligatoriamente**, se utiliza el atributo **required**.

Si el campo en la entrada de datos está vacío, el navegador lo notifica dentro del campo del formulario.

Se puede poner en el elemento `<input>` en los siguientes atributos:

- email.
- password.
- number.
- checkbox.
- search.
- radio.
- date pickers.
- tel.
- text.
- url.
- file.

```
<form action="005-ejemplo-atributo-required-html5.php" method="POST">  
    Fecha de nacimiento: <input type="date" name="fechanac" required>  
    <input type="submit" value="Enviar">  
</form>
```



Si queremos especificar el modo de **transmisión de datos**, es decir, la forma en que se envía la información del formulario al servidor, usamos el atributo **@method**.

Existen dos tipos de transmisión de datos al servidor:

- **“get”**: `<form method="get">`. Es el valor por defecto del atributo.
Indica al navegador que adjunte a la **URL** los valores que el usuario ha introducido en el formulario.
- **“post”**: `<form method="post">`
Su función es idéntica al *get*, solamente que la información se envía en el cuerpo de la solicitud **HTTP**, por lo que la información no es visible para todos.

```
<form action="005-ejemplo-atributo-required-html5.php" method="POST">  
  Fecha de nacimiento: <input type="date" name="fechanac" required>  
  <input type="submit" value="Enviar">  
</form>
```

Atributos



En diapositivas anteriores hemos hablado de algunas **propiedades** asociadas a algunos **elementos**, como por ejemplo, el *input*, ¿lo recuerdas?

Estas propiedades asociadas son conocidas con el nombre de **atributos**, y estos atributos pueden tener **valores**.

Consideraciones:

- Un elemento puede tener cero, uno o varios atributos.
- El orden en el que se especifiquen estos atributos no es relevante.
- El nombre del atributo va seguido por el signo “=” y el valor del atributo.
- El valor del atributo se pone con comillas (simples o dobles).
- Si el valor de un atributo lleva comillas, lo especificamos de la siguiente forma:

Atributo		Valor	
HTML		CSS	
1	<input	type="text"	id="name" name="name"
	required		
2		minlength="4"	maxlength="8" size="10">

href="http://www.ya.com/cine'filo"

href='http://www.ya.com/cine"filo'

LISTADO DE ATRIBUTOS DE HTML5

<https://developer.mozilla.org/es/docs/Web/HTML/Attributes>

Comentarios.
Caracteres.
Detalles de
estructura.



`<!--` (apertura) y `-->` (cierre) para **COMENTARIOS**. Estos no se muestran en la página web, son indicaciones/notas del programador.



HTML		CSS	OUTPUT
1	<code><h1>Profesora de Lenguaje de marcas: MFS</h1></code>		Profesora de Lenguaje de marcas: MFS
2			
3	<code><!-- esto es un comentario que no se ve --></code>		
4			
5			
6			
7			

SÍMBOLOS/CARACTERES ESPECIALES

Á	Á	À	À	Ü	Ü	Û	Û
É	É	È	È	ä	ä	â	â
Í	Í	Ì	Ì	ë	ë	ê	ê
Ó	Ó	Ò	Ò	ï	ï	î	î
Ú	Ú	Ù	Ù	ö	ö	ô	ô
á	á	à	à	ü	ü	û	û
é	é	è	è	Ã	Ã	å	å
í	í	ì	ì	Ñ	Ñ	Å	Å
ó	ó	ò	ò	Õ	Õ	Ç	Ç
ú	ú	ù	ù	ã	ã	ç	ç
Ä	Ä	Â	Â	ñ	ñ	Ý	Ý
Ë	Ë	Ê	Ê	õ	õ	ý	ý
&luml;	Ï	Î	Î	Ø	Ø	ÿ	ÿ
Ö	Ö	Ô	Ô	ø	ø	Þ	þ
				Ð	Ð	þ	þ
				ð	ð	Æ	Æ
				ß	ß	æ	æ

CODIFICACIÓN DE CARACTERES

Un sistema de codificación de caracteres define la forma en que se codifica un carácter del lenguaje humano en un sistema de representación.

Los **sistemas más utilizados** son:

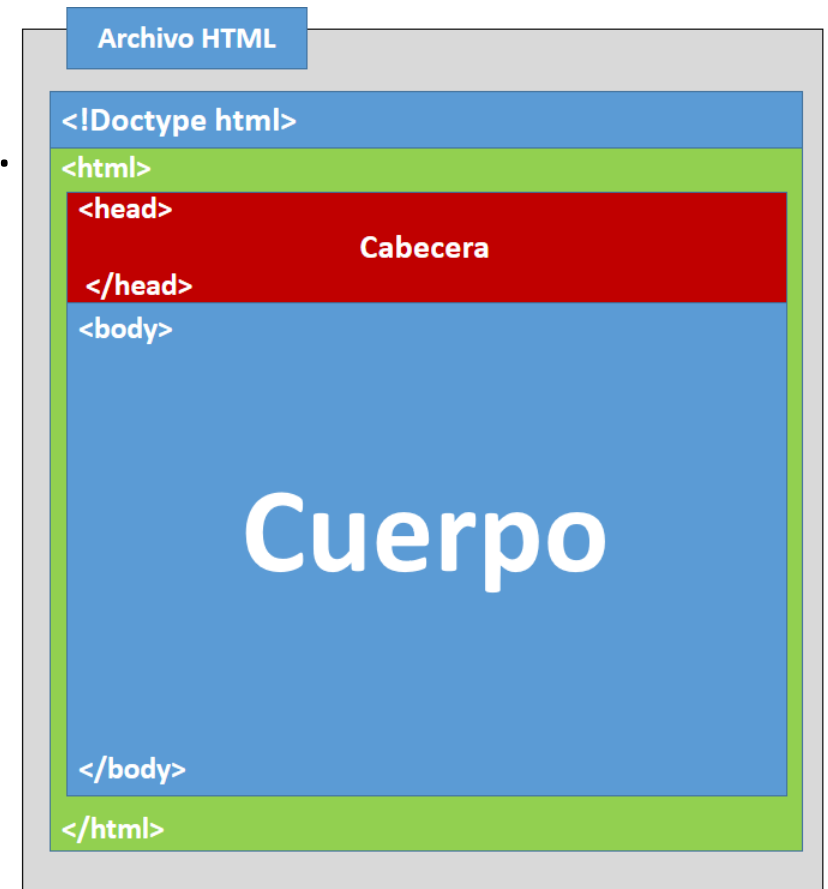
- UTF-8.
- ISO-8859-1.

El sistema usado se especifica dentro del elemento `<head>`.

```
<head>  
  <meta charset="UTF-8">  
</head>
```

La **estructura básica** de una página web se compone de las siguientes etiquetas:

- Una **declaración** de la versión de HTML.
 - Aparecerá en la primera línea del documento.
- Una **cabecera** con información acerca del documento.
 - `<head>` y `</head>`
- Un **cuerpo** con contenido real del documento.
 - `<body>` y `</body>`



DOCTYPE





¿Recuerdas que era una DTD en XML?

Era un documento que definía las reglas específicas que debe cumplir un documento XML...

En **HTML** una DTD describe:

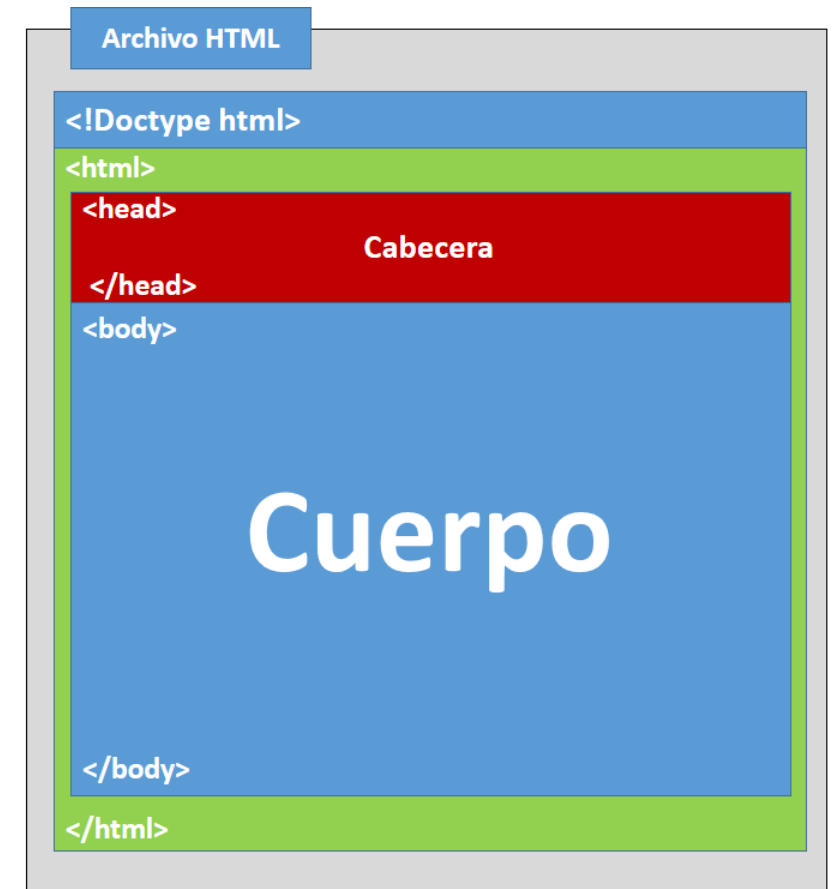
- **Elementos:** indican qué etiquetas son permitidas y el contenido de dichas etiquetas.
- **Estructura:** indican el orden en que van las etiquetas en el documento.
- **Anidamiento:** indica qué etiquetas van dentro de otras.

El *Doctype* o “Definición de tipo de documento” es una instrucción que va al comienzo del documento HTML que indica qué **VERSIÓN** de HTML estamos usando.

HTML 5 nos ofrece un **DOCTYPE** que no hace referencia a ninguna Definición de tipo de documento.

```
<!DOCTYPE html>
```

Las versiones anteriores (centrándonos en **HTML 4**) pueden usar DTD's especificadas de otras formas.



HTML 4 contempla 3 DTD diferentes que se podrán utilizar como declaraciones:

- **DTD estricta:** incluye todos los elementos y atributos que no han sido declarados *deprecated* en otras versiones.

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

- **DTD transicional:** incluye todos los elementos de la versión estricta y también los elementos y atributos *deprecated* de otras versiones.

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

- **DTD para documentos con marcos:** incluye todos los elementos de la versión transicional y también los elementos y atributos relativos a la creación de marcos (frames).

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

Documentos HTML correctos



Un documento HTML **correcto** es aquel que se ha escrito siguiendo los estándares vigentes para la versión que se utiliza.

Los programadores hacen uso de un validador para escribir código HTML correcto:

https://validator.w3.org/#validate_by_upload

Cuando se logra un documento HTML correcto y validado puede ser premiado con la posibilidad de incluir un sello de calidad correspondiente a la versión utilizada.



Referencias absolutas y relativas

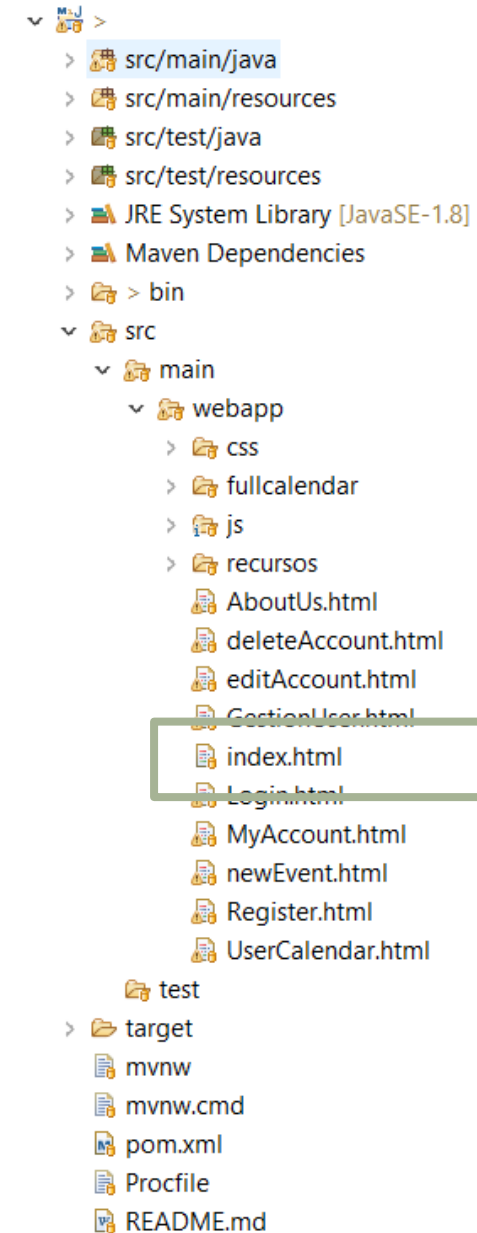
1.12

La **estructura de una página web** es la siguiente:

- Hay un documento raíz o principal denominado ***index.html***

A partir de este se crea un árbol jerárquico organizando los documentos y recursos web.

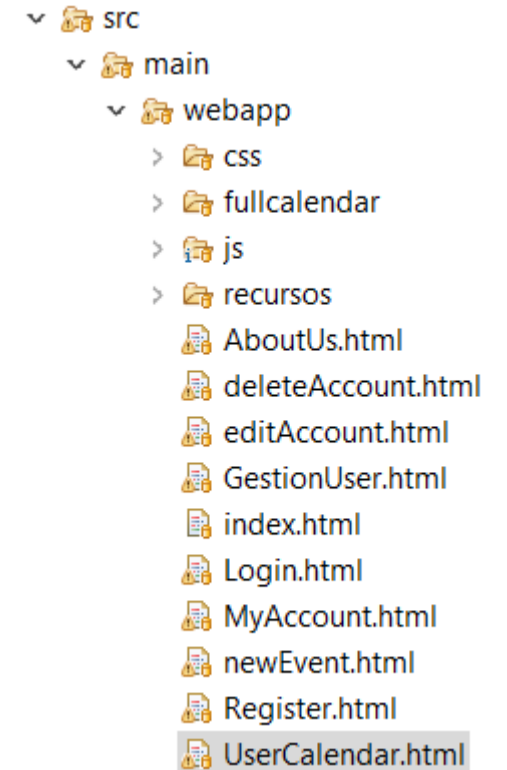
El documento *index.html* contiene las referencias/direcciones a archivos con los que se comunica (otros documentos *HTML*).



De forma que encontramos rutas en el propio código del documento HTML....

Estas pueden ser:

```
<link href='fullcalendar/main.css' rel='stylesheet' />  
<link href='css/calendario.css' rel='stylesheet' />  
<link href='css/UserCalendar.css' rel='stylesheet' />
```



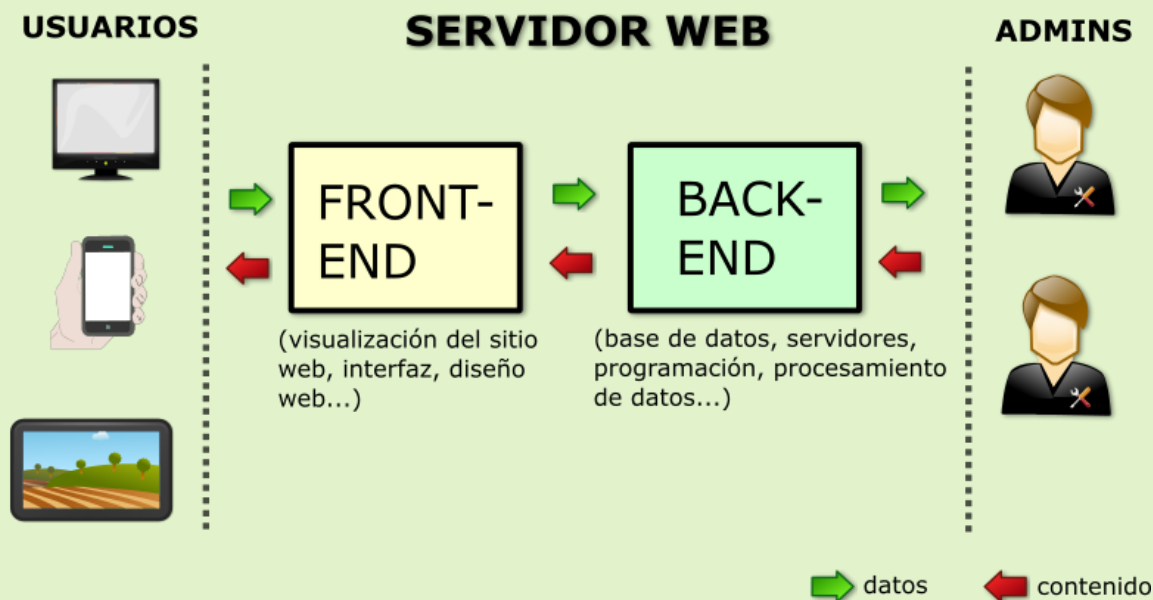
ABSOLUTAS	RELATIVAS
<ul style="list-style-type: none">Se usan para referenciar sitios EXTERNOS (urls externas). <p><i>http://www.google.es</i> <i>ftp://127.36.21.131</i> <i>file:///C:/Documentos</i></p>	<ul style="list-style-type: none">Se utilizan para navegar por los documentos de la misma página web. <p>index.html index.html/img</p>



¿Pero... cómo llegan los datos de la parte FRONT a la parte END?

Front-end: es el lado **cliente**. Consiste en la parte de la aplicación que **interactúa** con el **usuario**.

Back-end: es el lado **servidor**. Consiste en el conjunto de aplicaciones que se encuentran en el servidor, donde se encuentra la lógica del programa que maneja los datos procedentes del *front*. **No** es directamente **accesible** por el **usuario**.



Ver HTML CHEAT SHEET
(en la asignatura)

