

```
type="text/javascript"
var rightColumnUri = "/de
var currentContext = "op
</script>
<noscript>
    <a style="display:no
</noscript>
<div id="right_column">
    <iframe src="" id="i
        [Your user agent
        you may visit <a
    </iframe>
```

7

ESTRUCTURA DE UN DOCUMENTO HTML

7.1. IDENTIFICACIÓN SGML

Antes de comenzar a utilizar cualquier lenguaje de marcas es conveniente familiarizarnos con una serie de normas básicas que debemos tener en cuenta como:

• Etiquetas

Las etiquetas, también conocidas como marcas, definen una serie de elementos que forman el léxico del lenguaje HTML. Se encuentran entre los signos de *menor que* (<) y *mayor que* (>). Podemos diferenciar entre dos tipos de etiquetas: **cerradas** y **abiertas**.

– Etiquetas cerradas

Constan de una para la apertura (indica el comienzo de la etiqueta) y otra para el cierre (indica que hemos terminado de trabajar con ella y lleva el símbolo «/» antes del nombre).

Por ejemplo:

```
<p>
```

– Etiquetas abiertas

Cuentan con una única palabra reservada para indicar el inicio y el fin a la vez.

Por ejemplo:

```
<hr>
```

En HTML5 no es necesario escribir las etiquetas de cierre gracias a los navegadores actuales.

• Atributos

Los atributos, al igual que las etiquetas, se pueden definir tanto en mayúsculas como en minúsculas, a pesar de que los valores que les asignen sí guardarán diferenciación. Por ello, es recomendable utilizar siempre minúsculas y, de esta forma, evitar confusiones.

Las etiquetas pueden contener atributos si necesitan realizar alguna configuración sobre alguna característica determinada. Estos atributos se definen a continuación de la palabra reservada en la etiqueta de apertura separada por un espacio en blanco y antes del signo de cierre. Asignaremos el valor correspondiente al atributo a través del signo "=".

PARA + INFO

Cada comando cuenta con una serie de atributos con sus correspondientes valores; por ejemplo:

```
<p ALIGN = "left">
```

Definición de un párrafo y alineación del texto a la izquierda.

Es recomendable poner el valor entre dobles comillas para que sea más legible.

- **Comentarios**

Los comentarios son líneas que definen, de cara al usuario, lo que vamos realizando en cada momento, aunque no son interpretadas por el navegador.

De esta forma, si alguien necesita trabajar con el código, es capaz de interpretarlo con un simple vistazo, gracias a los comentarios que aparecen en él.

Los comentarios van escritos entre los símbolos: `<!--` y `-->`.

Por ejemplo:
`<!-- Esto es un comentario -->`

- **Estructura básica**

Como ya hemos indicado, las etiquetas que utilizemos en HTML siempre van a ir entre los símbolos `"<"` y `">"`. Y, cada vez que tengamos que cerrar una etiqueta, pondremos el nombre correspondiente comenzando con el símbolo `"/"`.

Todas las etiquetas afectan al código que se encuentre delimitado entre la apertura y el cierre de la etiqueta.

<code><html></code>	<i>Inicio del documento</i>
<code><head></code>	<i>Comienzo de la cabecera</i>
<code><title></code>	<i>Inicio del título del documento</i>
Título	
<code></title></code>	<i>Fin del título del documento</i>
...	
<code></head></code>	<i>Fin de cabecera</i>
<code><body></code>	<i>Inicio del cuerpo</i>
...	
<code></body></code>	<i>Fin de cuerpo</i>
<code></html></code>	<i>Fin del documento</i>

HTML5, mediante unas etiquetas nuevas, añade una serie de características y elementos cuya función es facilitar la tarea a los autores de la aplicación web.

HTML5 se basa, principalmente, en una estructuración avanzada que se encarga de definir los contenidos agrupándolos en distintas etiquetas. Estas tienen un nombre asignado según la tarea que se va a realizar:

- `<header>`: define un conjunto de ayudas introductorias en una página.

Podemos encontrar elementos como:

- El menú de navegación principal con los enlaces a las secciones de la web.
- La marca o logotipo de la web.
- Una pequeña descripción (en ocasiones).
- El buscador de la página (en ocasiones).
- `<nav>`: define los enlaces de navegación.
- `<article>`: crea algún artículo que se haya publicado.
- `<section>`: parte correspondiente a algún artículo.
- `<aside>`: define contenido lateral de la página.
- `<footer>`: define el pie de página.
- `<dialog>`: define distintos diálogos o comentarios.

Además de estas etiquetas, HTML5 también cuenta con componentes como `<div>` y ``, utilizados para poder agrupar los diferentes elementos "hijos" haciendo uso de atributos como: *class*, *id* o *tittle*. De esta manera, se pretende usar una misma semántica con un estilo común.

7.2. CABECERA

La cabecera del programa se encuentra siempre entre:

```
<head>
...
</head>
```

Lo que se haya, a su vez, dentro de un elemento superior como es `<html>`.

Dentro de la cabecera es donde vamos a definir los elementos generales, como el título de la página:

- `<title>`: es el título que va a aparecer en el navegador web, en la barra superior.
- `<meta>`: encargada de indicar el contenido de nuestras palabras junto con los términos clave. Esta directiva suele llevar dos atributos (*name* y *content*) que hacen referencia al nombre de la página y a sus principales contenidos.

`<meta name = "description" content = "Página principal ILERNA con alumnos y profesores">`

`<meta name = "keywords" content = "Nombre y apellidos de alumnos matriculados">`

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Reg CSS</title>
  </head>
  <body>
    <div class="afr1">
      <div class="afr2"></div>
      <div class="afr3">
        <div class="afr4"></div>
      </div>
    </div>
    <h1>Registration</h1>

    <form action="" method="post">
      <p>Many fields</p>
      <p>And we have some question:</p>

      <!-- Add a box here -->
      <label for="subscribe-field">Would you like to receive
      <input type="submit" value="Send">
    </form>
  </body>
</html>
```

PARA + INFO

Otro uso diferente de la etiqueta `<meta>` es el “refresco automático”: transcurrido un tiempo estimado, la misma página pasa a actualizarse. Es bastante recomendable para aquellas en las que el contenido se modifica con mucha frecuencia.

```
<meta http-equiv = "refresh" content = "10"; url = "http://www.google.es">
```

Con esta instrucción estamos indicando que, pasados 10 segundos, se va a acceder a la página web de Google.

- `<base>`: podemos indicar una URL base de algún documento, sonido, gráfico, etc., que hagan referencia a una determinada página web. Cuando creamos una página web, es conveniente que, inicialmente, realicemos una planificación de su diseño para, después, ordenar la información y los recursos que se van a ofrecer. Para llevar a cabo esta tarea, es recomendable que hagamos uso de una estructura de directorios.
- `<style>`: gracias a esta etiqueta podemos definir los estilos de nuestro HTML:

```
<html>
<head>
<style>
h1 {color:red;}
p {color:blue;}
</style>
</head>
<body>

<h1>Titulo</h1>
<p>Párrafo</p>

</body>
</html>
```

Etiquetas que definen metadatos son `<link>` y `<script>`.

7.3. CUERPO DEL DOCUMENTO

El cuerpo del programa se encuentra siempre entre:

```
<body>
...
</body>
```

Siempre está situado detrás de la cabecera `<head>`. Va a contener todo el cuerpo correspondiente a una determinada página web junto con los elementos propios de la página: gráficos, textos, imágenes, etcétera.

En él, tenemos la posibilidad de definir una serie de acciones necesarias para **eventos** más concretos, entre las que podemos destacar:

- **Onload**: ejecuta un evento inmediatamente después de cargar la página.
- **Onunload**: ejecuta un evento cuando un usuario descarga un documento.
- **Online**: ejecuta un evento cuando el navegador se abre y tiene salida a internet.
- **Offline**: ejecuta un evento cuando el navegador se abre y trabaja sin conexión.
- **Onafterprint**: ejecuta un evento cuando la página ha comenzado a imprimir o si el diálogo con la impresora ha sido cerrado.
- **Onbeforeprint**: ejecuta un evento cuando la página está a punto de ser impresa.

Además, existe un conjunto de atributos (opcionales) de `<body>` que permiten realizar diferentes configuraciones sobre la apariencia de un documento. Hoy en día están obsoletos, ya que HTML5 no los soporta.

Algunos de ellos son: *bgcolor*, *text*, *link* y *vlink*.

