

## HTML5 Conceptualmente:

El paso más importante del nuevo estándar es la inclusión del DOM (Modelo de Objetos del Documento) como componente elemental y básico de HTML, pudiendo presentar mayores funcionalidades.

**DOM** es un conjunto de entidades y funciones que nos permiten trabajar con las distintas partes de un documento mediante un lenguaje simple pero potente como JavaScript.

HTML5 no es un lenguaje estático dado que fue diseñado para ser ampliado a medida que surjan las necesidades, esto hará que no se quede retrasado respecto a la evolución de la Web.

Es de suponer entonces que HTML5 durará en el tiempo y que proporcionará un entorno estable para el diseño de páginas Web.

### Los elementos que desaparecen en HTML5:

- Por ser mejor resueltos mediante CSS: basefont, big, center, font, strike, tt, u
- Por perjudicar la usabilidad y la accesibilidad: frame, frameset, noframes.
- Por resultar confusos o reemplazados por otros: acronym, applet, isindex, dir.

HTML es básicamente un mecanismo para representar contenido estático en la interfaz del usuario; la necesidad de gestionar la interactividad con el usuario se resuelve mediante el uso de código de programa (por ejemplo, JavaScript) que se ejecuta dentro del navegador del cliente sin necesidad de utilizar una transmisión de datos hasta el servidor.

La etiqueta <SCRIPT> permite pasar código al agente de usuario dentro del código HTML; opcionalmente se puede utilizar un archivo externo.

**Un Agente de usuario:** Es cualquier aplicación informática que funciona como cliente en un protocolo de red; de modo genérico, es una aplicación que accede a la World Wide Web. Por lo tanto, todos los navegadores web son agentes de usuario y también lo son los robots de los buscadores y otros programas similares.

## Estructura básica de los documentos HTML:

Para que un documento se pueda considerar HTML5 debe cumplir ciertos requisitos mínimos, en caso contrario, y dado que los navegadores son compatibles hacia atrás, será interpretado como un documento HTML pero no como HTML5.

### Existen dos elementos básicos que no pueden faltar para que el documento sea HTML5.

- Un comentario inicial <!DOCTYPE HTML> que indica que el documento está escrito en HTML.
- Una etiqueta <TITLE></TITLE>

La etiqueta <TITLE> ya existía pero su codificación era más compleja, su función es la de dar el título a la ventana, pero no debe faltar para que el documento sea considerado como HTML5.

### Categorías y modelos de contenido en HTML5:

HTML4 distingue elementos del lenguaje diferentes “a nivel de bloque” y en “línea”.

Este enfoque ya no se hace en HTML5 ya que ésta es una característica propia de la presentación del elemento en la página, tarea que vamos a delegar a CSS.

**En HTML5 se utilizan las categorías de:**

- Elementos de Frase: A los que antes se denominaban en línea (inline)
- Elementos de Flujo: Parecidos a los que antes se denominaban elementos de bloque.  
→ Muchos elementos como <A>, <INS>, <DEL> pueden pertenecer a ambas categorías.
- Elementos Incrustados ó embebidos: Aquellos con capacidad para importar contenido desde otro recurso:  
<AUDIO>,<VIDEO>,<EMBED>,<IMG>, etc

**Etiquetas y atributos:****Las Etiquetas:**

Los documentos HTML están compuestos por una estructura de elementos denominados etiquetas y por texto.

Las etiquetas en general tienen una definición de apertura y otra de cierre, pero en algunos casos se puede omitir la etiqueta de cierre.

Las etiquetas se pueden anidar. Ten en cuenta de no dejar ninguna etiqueta abierta pendiente de cerrar.

**Los Atributos:**

Son los parámetros que nos permiten configurar el modo en que actúan las etiquetas.

Los atributos, cuando son necesarios, siempre se incluyen en la etiqueta de apertura, nunca en la de cierre.

Los atributos están compuestos por dos partes: nombre y valor, separados por un signo “=”.

Si el valor se omite, por ser una cadena vacía, puede omitirse también el signo “=”.

```
<meta charset=UTF-8>
<input name=id1 disabled>
<!-- lo que es igual a --->
<input name=id1 disabled="">
```

**El Texto y los caracteres especiales.**

En su origen HTML era un lenguaje orientado a la gestión del texto y es así como están formados la gran mayoría de los documentos.

Pero HTML se encontró con un primer problema, los caracteres especiales: ¢ , ´ , ñ ...

Provocando que caracteres propios de otros lenguajes sean caracteres especiales para HTML y el problema proviene de que el estándar ASCII de 7 bits es muy limitado y no puede resolver este problema.

Por tanto, la solución es, que para definir el juego de caracteres a interpretar usamos la etiqueta:

<meta charset="..."> , por ejemplo UNICODE(UTF-8)

**Para superar problemas puntuales de algún carácter especial se puede utilizar la siguiente codificación:**

&#NRO: siendo NRO un valor entero que indica el código del carácter. Ej: &#241 es la letra ñ

**Otro modo de codificación es usar el formato hexadecimal:**

&#COD: siendo CDO un valor hexadecimal. Ej: %#xF1 es la letra ñ

Los acentos (agudos, graves y circunflejos), la diéresis y otros signos de puntuación se resuelven mediante el uso de valores simbólicos que indican que a un determinado carácter se le añadirá un signo de puntuación. Acuérdate que acaban en ;

```
&aacute; :indica la á (acento agudo)
&ntilde; :indica la ñ
&ograve; :indica la ò (acento grave)
```

## Estructura General:

### Secciones principales:

<HTML>, <HEAD>, <BODY> que delimitan las principales secciones del documento HTML.

#### Dentro de la etiqueta <HEAD>

Esta etiqueta delimita la sección de encabezamiento y encierra un bloque de información sobre el documento y suele contener el siguiente grupo de etiquetas:

**<TITLE>:** Obligatoria para que el documento sea HTML5, parece una etiqueta inofensiva pero es muy importante para que la página web esté bien posicionada en los buscadores. Es importante que el título contenga palabras clave.

**<BASE>:** Permite especificar un directorio que actuará como base para las referencias relativas que usemos en el documento HTML.

Esta dirección, reemplaza a la base que se toma por defecto que es la dirección del propio documento.

Por ejemplo si hacemos referencia a la página desde donde cogemos una imagen.

La directiva <BASE> es muy útil para modificar los direccionamientos cuando se prueba una página en un servidor de prueba o en un entorno local, con sólo cambiar el atributo href de esta etiqueta estamos modificando todos los direccionamientos relativos sin tocar nada más en el documento.

Atributos específicos de <BASE>:

target: Si se especifica debe contener un nombre de destino válido, que se usará como entorno de navegación de hiperenlaces y formularios del documento.

href: Dirección URL válida que se usa como dirección base.

### **Trabajo en Localhost**

Cuando hacemos nuestras prácticas usamos páginas almacenadas en el disco duro del equipo local y no en un servidor web remoto

#### **¿Cómo nos referimos a un sitio local?**

Supongamos que los archivos HTML que queremos abrir en el ordenador están en la carpeta siguiente:

<C:/PruebasHTML/Codigo/>

Y supongamos que las imágenes están en la siguiente carpeta:

<C:/PruebasHTML/imag/>

Si utilizamos la siguiente directiva <BASE>: <file://localhost/C:/Pruebas/Codigo/>

El protocolo en este caso no es http sino file y con localhost se indica que la referencia es al equipo local, luego de esto se incluye la ruta de acceso que define la base.

**<LINK>:** Permite establecer enlaces con otros recursos.

**href:** Dirección url

**rel:** Tipo de enlace. → Cuando son hojas de estilo su valor es stylesheet

Si se omite alguno de estos atributos la etiqueta no tiene efecto.

**<META>:** Es un contenedor de metainformación relacionada con el documento, solo puede haber una etiqueta <META> en el documento.

Para incluir información que queremos que sea utilizada por los robots de los buscadores.

La etiqueta <META> se usa para metadatos que no se pueden expresar mediante las otras etiquetas del grupo: TITLE, BASE, LINK ó STYLE.

Debe tener siempre uno de estos tres atributos:

**name:** Identifica el tipo de información

**charset:** Determina el conjunto de caracteres del documento.

**Content:** Indica los metadatos.

**Http-equiv:** Indica alguna directiva pragma:

content-language (asigna el lenguaje predeterminado)

content-type (alternativa al uso del atributo charset)

default-style (define la hoja de estilo por defecto)

refresh (fuerza la recarga de un documento)

set-cookie (asigna cookie)

content (es el atributo que complementa a los anteriores conteniendo el valor)

**<STYLE>**: Permite incluir información sobre el estilo dentro del propio documento.

Atributos:

**type:** Indica el estilo y debe ser una tipo MIME válido, en caso de que no se informe el atributo type se presupone "text/css"

**media:** Indica a qué medio se aplica el estilo. Si no se indica se presupone "all", es decir que se aplica a todos los medios del documento.

**title:** Define una hoja de estilo alternativa.

### Ejemplo:

```
<!DOCTYPE HTML>
<html>
  <head>
    <meta CHARSET="UTF-8">
    <base href="http://www.misitio.es/">
    <title>Mi primer HTML</title>
    <link REL="STYLESHEET" HREF="default.css">
    <script SRC="codigo1.js"></script>
    <meta name="aplicacion-name" content="ejemplo 01">
  </head>
  <body>
    ...
  </body>
</html>
```

### Bloques de Texto:

**<P>**: Para definir párrafos formateados.

**<H1>...<H6>**: Para definir encabezamientos.

**<BLOCKQUOTE>**: Para definir texto de un segundo nivel.

**<PRE>**: Para definir texto preformateado.

**<UL>, <OL>, <DL>**: Para producir listas sin numerar, ordenadas y definiciones.

**<DIV>**: Una división de documentos.

**Marcas de caracteres:**

El texto puede contener elementos inline (embebidos dentro del texto) para realizar marcas de caracteres.

**<B>**: Negrita

**<I>**: Cursiva

**<BIG>**: Texto Grande

**<SMALL>**: Texto pequeño

**<EM>**: Texto resaltado

**<STRONG>**: Texto en negrita

**<TT>**: Texto tipo teletipo (todas las letras del mismo ancho).

**Caracteres especiales:**

Para ello usamos & y no olvides el ;

**&lt;**: Representa <

**&gt;**: Representa >

**&amp;**: Representa &

**&nbsp;**: Representa un espacio que no debe producir rotura, para mantener juntas las palabras.

**&#xxx;**: Introducimos el código ASCII del símbolo

**Elementos de organización:**

**<TABLE>**: Para ordenar los datos en filas y columnas <TR>, <TD>, <TH>

**<FORM>**: Para formularios.

**Enlaces a otras páginas:**

**<A>**: Define la referencia que puede ser interna del sitio, dentro de la propia página ó externa.

**Imágenes:**

**<IMG>**: Incluye un enlace a un archivo de imagen con información adicional para indicar cómo mostrar la interfaz

**Scripts:**

**<SCRIPT>**: permite incrustar código Javascript ó hacer referencia al archivo Javascript