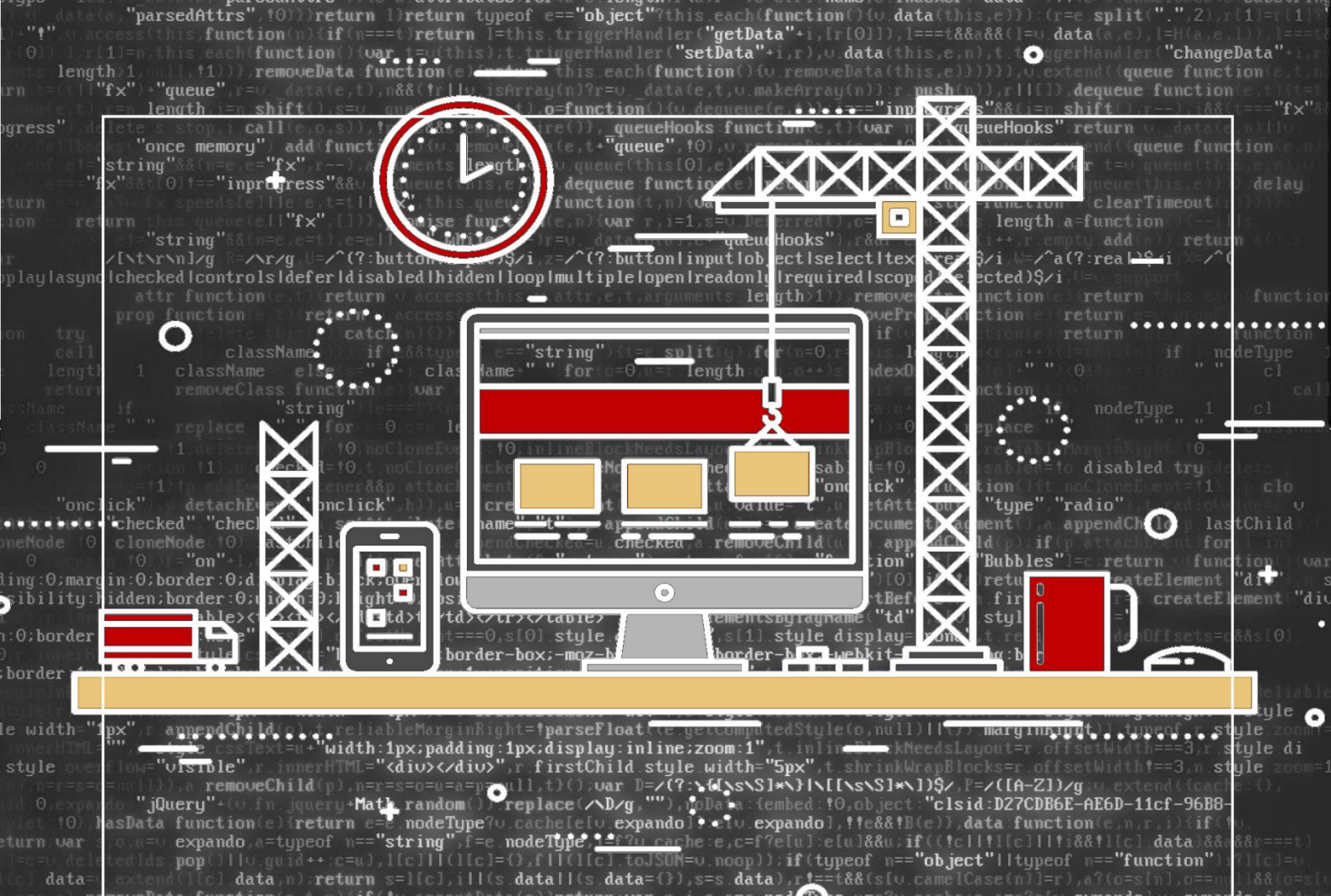


IFCD0110 CONFECCIÓN Y PUBLICACIÓN DE PÁGINAS WEB

Módulo formativo: MF0950_2: Construcción de páginas web

Unidad formativa: UF1302 Creación de páginas web con lenguajes de marcas

Unidad 2. Documentos HTML. Evolución de HTML a HTML5. Estructura de una página web. Etiquetas, comentarios, cabecera.

FORMACIÓN PROFESIONAL PARA EL EMPLEO.

CIFP VIRGEN DE GRACIA

CIFP VIRGEN DE GRACIA

Castilla-La Mancha

GOBIERNO DE ESPAÑA

MINISTERIO DE TRABAJO Y ECONOMÍA SOCIAL

SERVICIO PÚBLICO DE EMPLEO ESTATAL SEPE

1. Evolución de HTML

- **HTML1.0 (1991)**:
  - Creado por Tim Berners Lee en CERN para que la comunidad científica compartiera información.
  - Derivado de SGML.
  - 20 etiquetas / 13 aún perduran.
  - Primer navegador Nexus
- **HTML2.0 (1994):**
  - Primera especificación oficial.
  - Incluye todo lo de la versión 1.0.
  - Aparecen nuevos navegadores Lynx, Cello, Mosaic.
  - 19 nuevas etiquetas (imágenes, formularios, …)
  - <!DOCTYPE> que asocia un documento con un DTD para validarlo. (DTD (Definición de Tipo de Documento. Es un documento que define la estructura que va a tener nuestro HTML: elementos, atributos, notaciones, … que pueden aparecer, el orden y el número de veces. El procesador utiliza la DTD para verificar si un documento es válido o no)
- **HTML3.2 (1997):**
  - En 1996 aparece Internet Explorer.
  - Se añaden tablas, mapas, etc.
  - Desaparecen algunas etiquetas como markee y blink.
  - Es la primera versión enteramente desarrollada por W3C.
  - Aparece CSS y los navegadores empiezan a adoptarlo.
- **HTML4.01 (1999):**
  - Fue una recomendación.
  - Ya incluye hojas de estilos.
  - Se quitan las etiquetas de estilos de versiones anteriores.
  - Se mejora la presentación de fuentes, fondos y colores.
- **XHTML1.0 (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.
- **HTML5.0 (2014):**
  - Fue una recomendación y las sucesivas versiones también (aún no es estándar).

- Comenzó a desarrollarse por las pocas posibilidades de aplicaciones complejas XHTML.
- Etiquetas semánticas.
- APIs
- Simplifica DOCTYPE, Link, Script.
- Mejora formularios.

2. Documentos HTML.

HTML es el lenguaje de marcado estándar para la creación de páginas web.

HTML describe la estructura de una página web.

HTML consta de una serie de elementos.

Los elementos HTML le dicen al navegador cómo mostrar el contenido.

Los elementos HTML etiquetan partes de contenido como “esto es un encabezado”, “esto es un párrafo”, “esto es un enlace”, etc.

Ejemplo:

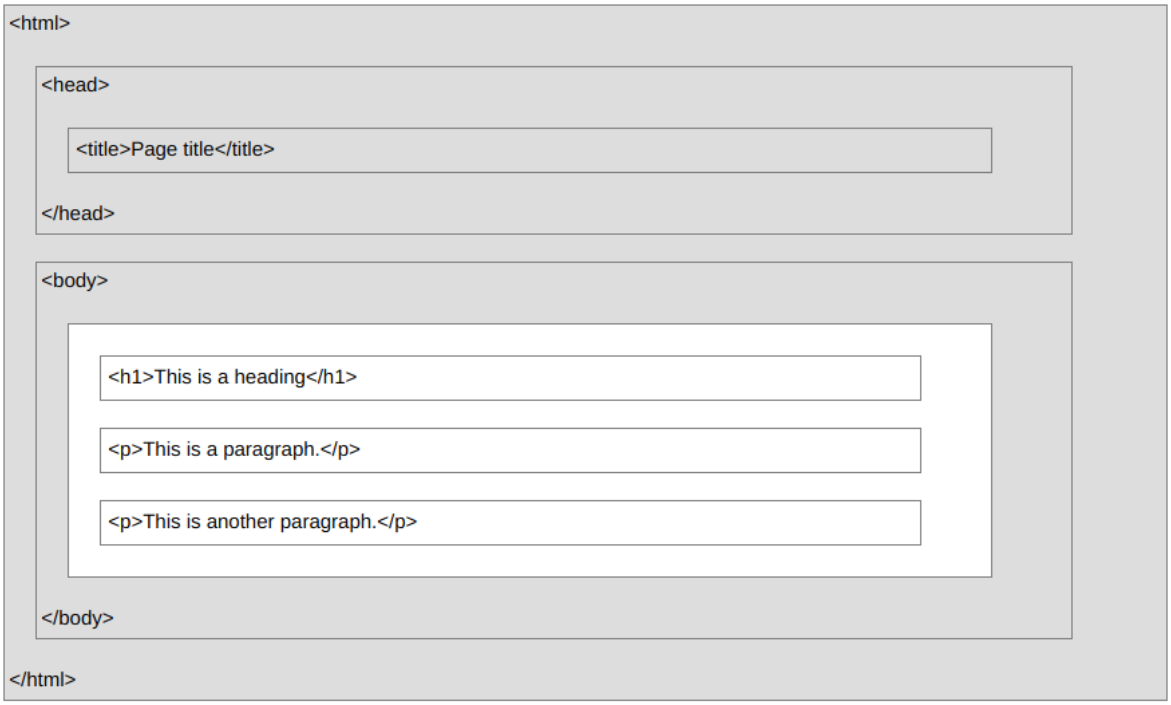
```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

Elemento: Un elemento en HTML es definido mediante una etiqueta de inicio, algo de contenido y una etiqueta de fin.

3. Estructura de una página web.



5. Etiquetas y elementos.

Las etiquetas, como hemos dicho, sirven para delimitar elementos de la página.

Los elementos, son cada uno de los componentes de una página: por ejemplo, un párrafo es un elemento de la página, una tabla también. Incluso hay elementos que contienen otros elementos (las tablas constan de filas y las filas de celdas, por ejemplo).

Las etiquetas son textos encerrados entre los signos de mayor y menor (**< >**). Cuando un navegador se encuentra un texto así encerrado, entenderá que dentro de los símbolos menor y mayor lo que se indica es el inicio de un elemento. El inicio del elemento se marca con el nombre del elemento entre los símbolos < y >.

```
Texto normal <strong>texto negrita</strong>
```

La siguiente imagen refleja las etiquetas utilizadas de modo incorrecto:

```
Texto normal <strong><em>texto negrita y cursiva
</strong></em>
```

```
Texto normal <strong><em>texto negrita y cursiva</em>
</strong>
```

6. Estructura de un documento HTML.

Ver ejemplos 0 y 1. Utilizaremos herramientas de desarrollo y el validador.

Para crear una página web usando **HTML 5** la estructura básica es la siguiente:

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

    </body>
</html>
```

Donde el significado de los elementos es el siguiente:

- **línea DOCTYPE**. La etiqueta **DOCTYPE** indica el tipo de HTML que estamos utilizando (concretamente HTML 5). Se trata de una línea que la actualidad siguen ignorando casi todos los navegadores, pero que es muy recomendable (y muy respetuoso con la norma) escribir. En ella se indica la versión HTML que sigue el documento.
- **etiqueta html.** Todo documento HTML está encerrado dentro de la apertura y el cierre de esa etiqueta. Marca el principio y fin del mismo (marca el **elemento raíz** de un documento HTML). Se le debe añadir el atributo **lang** para indicar el idioma en el que está escrito el documento.
- **cabecera, etiqueta head**. Encierra las etiquetas de cabecera, las cuales proporcionan información que el navegador debe tener en cuenta para el contenido

del documento aparezca de forma correcta y también contiene elementos que proporcionan información a los buscadores (como Google). Los elementos habituales que contiene son:

- **etiquetas meta**. Que sirven para indicar aspectos de funcionamiento de la página web como, cabeceras del protocolo http, indicaciones sobre cada cuánto caduca la página, palabras clave para buscar la página en los navegadores, codificación de caracteres de la página,… En este caso sólo se usa la etiqueta **meta** obligatoria que indica que estamos codificando el texto usando el formato **utf-8** de **Unicode**.
- **título, etiqueta title**. El título de la página, que los navegadores suelen colocar en la barra de título de la ventana en la que se muestra la página.
- **declaración de archivos externos y scripts**. En el ejemplo no hay declaraciones de este tipo.

- **cuerpo**. Encerrado en la etiqueta **body**, encierra el contenido en sí de la página web. Lo que el navegador muestra por pantalla es lo que se coloca en este apartado.

7. Espacios en blanco.

FUNCIONAMIENTO DE LOS ESPACIOS EN BLANCO

El texto dentro de las páginas web no respeta los espacios en blanco ni tabuladores que coloquemos en el código, a la hora de mostrar el contenido por pantalla. Solo se considera el primer espacio en blanco, el resto se eliminan.

Ejemplo: 2.EspaciosEnBlanco.html

La razón es permitir una mayor legibilidad del código, permitiendo en él que coloquemos el código con sangrías y espacios que mejoren su lectura, para que el mantenimiento del mismo sea más cómodo.

ETIQUETA PRE

Si deseamos que los espacios que colocamos en el código sí sean tenidos en cuenta, podemos utilizar la etiqueta **pre**.

Realmente, con carácter general, no se aconseja el uso de pre porque ayuda a adquirir malos hábitos y dificulta el aprendizaje de las formas avanzadas de maquetar páginas web.´

Sí hay contextos en el que es obligado su uso, por ejemplo para mostrar textos referidos a ejemplos de código de lenguajes de programación. Ejemplo:

Ejemplo: 3.CodigoJava.html

USO DE

Otra forma de obligar a tener en cuenta espacios en blanco es el código ** **. Dicho código es una entidad. Si, por ejemplo, escribimos cuatro veces seguidas * *

estaremos dejando cuatro espacios en blanco que el navegador sí tendrá en cuenta. Ejemplo:

```
texto         otro texto
```

8. Codificación de caracteres en HTML.

USO DE LA ETIQUETA META CHARSET

Los documentos de texto se pueden codificar de diferentes formas. Ejemplos de esas formas son la codificación ASCII clásica, o la tabla ISO 8859-1 que es una tabla ASCII extendida pensada para las lenguas de Europa Occidental[1].

Hay que tener en cuenta que las páginas web se pueden ver, una vez publicadas en Internet, en cualquier parte del mundo y el mismo código puede interpretarse de forma diferente si no avisamos de la forma en la que hemos codificado el documento.

Hasta hace unos años, para escribir en español se aconsejaba usar la codificación ISO-8859-1 (todavía hay millones de páginas que usan dicha codificación), pero ahora lo recomendable es usar **Unicode**, concretamente en su forma **UTF-8**. Para indicar que nuestra página utiliza Unicode UTF-8 basta con indicar en la cabecera, la etiqueta:

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

En versiones anteriores a HTML5, podemos encontrar:

```
<meta http-equiv="content-type"
      content="text/html; charset=UTF-8">
```

INDICACIÓN EN EL SOFTWARE DE ESCRITURA DE LA PÁGINA

No vale sólo con hacer esa indicación. Debemos estar seguros de que realmente estamos codificando en Unicode. Por ejemplo si usamos el Bloc de notas de Windows para escribir la página web y guardamos sin cuidado, no estaremos codificando el texto en Unicode. Hay que estar seguros que al guardar estamos realmente guardando el texto usando Unicode UTF-8:

Si no aseguramos que estamos codificando de esa forma, al abrir la página no se mostrarán bien los caracteres fuera del ASCII, cuando la persona que ve nuestra página usa otro sistema diferente del nuestro (por ejemplo si abre la página en un sistema Linux y nosotros habíamos usado la codificación nativa de Windows).

Hoy en día todos los editores profesionales guardan de forma nativa en UTF8, pero no está de más asegurarnos.

9. Códigos de idioma.

Como se ha indicado anteriormente, el atributo **lang** permite indicar dos letras que significan el idioma en el que está escrito el texto del elemento. Lo normal es indicarle en el elemento html (por ejemplo <html lang=”es”>), pero hay veces que una página contiene elementos cuyo contenido está escrito en un idioma diferente al indicado como general en el apartado **<html>**.

Como el atributo lang, es común a todos los elementos, se puede indicar el mismo cuando sea necesario. Ejemplo:

```
<p>
        Como dirían en Estados Unidos:
        <span lan=”en”>Welcome!</span>
</p>
```

¿Por qué tomarse tantas molestias? Para que, por ejemplo, los buscadores de Internet distingan el texto en el idioma apropiado y adapten mejor nuestras búsquedas. Otras herramientas (como correctores ortográficos, traductores en línea, etc.) funcionan mucho mejor marcando adecuadamente el idioma.

Podemos ser más específicos en el idioma e indicar no solo el idioma sino el dialecto concreto del mismo. Para ello se acompaña de un guión al código del idioma seguido de dos letras mayúsculas. La norma ISO que contiene los códigos disponibles es la norma 639-2.

Por ejemplo si queremos marcar un elemento con el idioma español, pero indicando que es español de México, haríamos lo siguiente:

```
<p>
        Como dirían en México:
        <span lan=”es-MX”>¡Híjole!</span>
</p>
```

10. Contenido y contenedores.

En HTML los elementos que muestran algo en la página, se colocan dentro de la sección **body**.

Además hay que tener en cuenta que hay elementos pensados para contener grandes cantidades de información (como **div**) y elementos que contienen poco texto (como **strong**). Es correcto que aparezca un elemento strong dentro de un elemento p, pero no al revés:

```
<strong><p>Hola</p></strong> -->¡¡Incorrecto!!
<p><strong>Hola</p></strong> -->Correcto
```

La mayoría de elementos en HTML son contenedores de otros textos y elementos. Hay que conocer que tipos de elementos tenemos. Los tipos están relacionados con la propiedad CSS **display**, que se encarga de decir cómo tiene que mostrarse un elemento. Por defecto los elementos funcionan de esta forma:

- **Elementos de tipo** *inline*. Son elementos que se muestran seguidos en la misma línea. Los elementos de este tipo son los que marcan texto simple. Todas las etiquetas de marcado de texto tienen este *display*: **strong, em, span, mark, abbr,** etc.

  Ejemplo, el código:

```
<strong>Hola</strong> <em>Hola</em>
```

Hola *Hola*

- **Elementos de tipo** *block*. Son contenedores más grandes. Los elementos que marcan párrafos enteros (como **p**, **h1**, **h2** o **blockquote**) y los contenedores de secciones (como **div, section, article** o **figure**) tienen este **display**. Su característica es que, por defecto, tienen forma rectangular y abarcan toda la anchura de su contenedor. Si indicamos para ellos un color de fondo, este se muestra en forma de rectángulo (en los inline, el color de fondo abarca exactamente al texto, como si lo hubiéramos pintado con un rotulador). Ejemplo:

```
<h1>Hola</h1> <p>Hola</p>
```

Hola

Hola

- **Elementos de tipo** *inline-block.* Son rectangulares como los **block**, pero no abarcan todo su contenedor. Se adaptan a su contenido. Ejemplos son los elementos **button** o **image**:

```
<strong>Hola</strong> <button>Hola</button>
```

El resultado muestra un botón (elemento rectangular) seguido del texto de tipo **strong**. Si el botón tuviera un display de tipo **block**, saltaría a la línea siguiente.

En todo caso, el **display** de un elemento se puede modificar desde lenguaje CSS, pero no conviene hacer combinaciones anti naturales ya que se pierde el sentido semántico del código.

11. Atributos comunes.

Todos los elementos tienen una serie de atributos comunes a todos ellos. Eso significa que podemos utilizarles independientemente de cuál sea el elemento.

El atributo común más conocido es **id**, que permite identificar a un elemento con un identificador único, pero hay muchos más. Los principales son:

atributo	significado
lang	Indica el lenguaje en el que está escrito el texto del elemento
style	Permite indicar código CSS directamente. Ese código solo afectará a ese elemento.
class	Permite indicar una clase CSS para el elemento
id	Permite identificar al elemento. Un identificador es un nombre único que se le da al elemento y que no se puede repetir en otro elemento dentro de la misma página web. El identificador sólo puede contener letras (del alfabeto inglés) y números (no se pueden usar espacios en blanco). Además debe empezar por una letra. El atributo id es uno de los fundamentales.
title	Permite poner un título al elemento. Es una especie de descripción corta que es muy útil en aquellos elementos que requieren de una explicación semántica como un abreviatura, una sigla, una imagen, etc. Normalmente, los navegadores reaccionan a este atributo mostrando un cartelito con el texto indicado en title, cuando el cursor del ratón se aproxima a un elemento que use este atributo

Hay muchos más atributos comunes:

atributo	significado
dir	Indica la dirección de lectura del texto contenido en el elemento (permitiría aplicar correctamente la forma de mostrar texto en árabe por ejemplo). Puede ser uno de estos dos valores: ltr (*left to right*, de izquierda a derecha) o rtl (*right to left*, de derecha a izquierda).
contenteditable	Si vale true, el usuario puede modificar el contenido del elemento. Es de HTML 5
draggable	Atributo incluido en HTML 5. Especifica (con valor true) que el párrafo tiene posibilidad de ser arrastrado con el ratón o los dedos (en dispositivos táctiles). El control del arrastre de todos modos debe de ser controlado por JavaScript.
hidden	No soportado por Internet Explorer. Cuando aparece este atributo (sin valor) hace que el elemento que lo contiene se le marque como no relevante y quede oculto.
spellcheck	Si vale true el párrafo queda marcado como revisable por herramientas de corrección de texto. Es de HTML 5
onclick ondblclic onkeyup onkeydown onkeypress onmouseover onmouseout onmousedown onmouseup onmousemove	Permiten asignar código Javascript a alguna de esas acciones (un clic de ratón, un doble clic, pulsar tecla, dejar de pulsar tecla, mantener pulsada la tecla, entrar con el ratón en el elemento, mover el ratón por encima, sacar el ratón del elemento, pulsar tecla de ratón, liberar tecla de ratón,...)

accesskey	Permite especificar una tecla para hacer que el control del usuario pase al elemento que la contiene. Su uso más habitual es con las etiquetas de enlace (a). Lo malo es que no funciona de forma muy intuitiva, en el ejemplo:
	`<p>`
	`Puedes ir a`
	`<a href=”http://www.elpais.es” accesskey=”p”>`
	`El Pais`
	`</a> o`
	`<a href=”http://www.elmundo.es” accesskey=”m”>`
	`El Mundo`
	`</a>`
	`</p>`
	Cada elemento a permite, haciendo clic, ir al periódico *El Pais* y *El Mundo* respectivamente. Las teclas indicadas son *p* para El País y *m* para El Mundo. En realidad la mayoría de navegadores usarán como teclas *Alt+p* y *Alt+m*, algunos ignorarán este atributo y otros pueden pedir además usar la tecla Intro.
	No obstante este atributo es interesante con uso de software especial que facilite la navegación para personas con necesidades especiales (como por ejemplo los invidentes).

tabindex	Atributo al que se le indica un número. Los elementos con tabindex más bajo obtienen primero el foco. Al pulsar el tabulador se resaltará ese elemento para poder utilizarle con el teclado. Es decir establece los saltos que ocurrirán al pulsar la tecla tabulador

12. Inserción de archivos CSS y JavaScript.

CSS

Si deseamos aplicar en nuestro documento código CSS procedente de un archivo aparte, debemos utilizar el elemento link cuya sintaxis es:

```
<link href=”ruta” rel=”stylesheet”>
```

Los archivos CSS contienen instrucciones que permiten modificar la apariencia de nuestro documento.

La etiqueta **link** debe ir en el apartado **head** de la página, normalmente después del título y las etiquetas **meta**.

INSERCIÓN DE ARCHIVOS EXTERNOS JAVASCRIPT

En este caso la etiqueta que lo permite es **script**. Esta etiqueta, a diferencia de **link**, sí tiene cierre.

Sirve para incrustar código JavaScript en el documento. Ejemplo:

```
<script>
  alert(“Hola”);
</script>
```

Si el código JavaScript procede de otro archivos, el código de ese archivos se incrusta de esta forma:

```
<script src="ficheroJavaScript.js"></script>
```

13. Comentarios.

Los comentarios en HTML son unos trocitos de código que nos permiten introducir en nuestra página web **contenidos que no se van a ver en el navegador**. Si quieres verlos, vas a tener que meterte en el código fuente de la web (“botón derecho->ver código fuente" o algo similar, depende del navegador).

Pues para *comentar* el código, obviamente, y poner "Esto lo puse aquí por esto, y esto otro porque tal...". Puede parecer una tontería, pero, cuando meses después tengas que volver sobre el código de esa web **te aseguro que lo vas a agradecer** ya que te habrás olvidado de las razones de poner determinado código. Esto es especialmente importante en lenguajes de programación *reales* (al fin y al cabo, HTML no lo es). Todos los lenguajes de programación tienen una forma de introducir comentarios.

Los comentarios en HMTL comienzan con el símbolo de "menor que", una admiración y dos guiones, y terminan con dos guiones y el símbolo de "mayor que". **O más sencillo**: es una etiqueta HTML **que empieza con una admiración y dos guiones y termina con dos guiones:**

```
<!-- Esto es un comentario -->
<p>Esto es un párrafo HTML</p>
<!-- Esto es otro comentario -->
```

```
<!-- Aquí empieza la cabecera de la web -->
<header>Aquí metemos el contenido de la cabecera de</header>
<!-- Fin de la cabecera -->
```

14. Bibliografía.

- https://www.w3schools.com/
- https://desarrolloweb.com/
- http://jorgesanchez.net
- https://www.campusmvp.es
- https://oscarmaestre.github.io/lenguajes_marcas/index.html