

# Tema 1: Introducció a HTML

---

**Web estàtica amb MkDocs**

*Enric*

*None*

## Table of contents

---

1. Tema 1 .....	3
1.1 UT 2. Estructura de los documentos: HTML, XHTML, XML .....	3

# 1. Tema 1

---

## 1.1 UT 2. Estructura de los documentos: HTML, XHTML, XML

---

### Recordatorio

Esta unidad trata sobre **los lenguajes de marcas** utilizados para estructurar documentos web: **HTML, XHTML y XML**. Entender sus diferencias es esencial para desarrollar aplicaciones web correctamente.

### 1. Introducción

**HTML (HyperText Markup Language)** es el lenguaje de marcas utilizado para crear la mayor parte de las páginas web.

#### Tipos de lenguajes

No se debe confundir **lenguaje de marcas** con **lenguaje de programación**.

Un **lenguaje de marcas** o **lenguaje de marcado** define un conjunto de reglas para codificar documentos en un formato que es legible tanto por personas como por máquinas, mientras que un **lenguaje de programación** proporciona comandos y sintaxis que permiten escribir programas entendidos por la computadora.

HTML es un estándar reconocido en todos los navegadores; por lo tanto, todos ellos visualizan una página HTML de forma muy similar, independientemente del sistema operativo sobre el que se ejecuten.

### Diferencia entre HTML y XML

Aunque ambos son **lenguajes de marcas**, HTML está diseñado para **mostrar información** en navegadores web, mientras que **XML** se utiliza principalmente para **almacener e intercambiar datos** de forma estructurada.

### Ejemplo de documento HTML

#### Éjemplo de documento HTML con numeración y botón de copia

```

1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8" />
5      <title>Título del documento</title>
6    </head>
7    <body>
8      <p>LMSXI</p>
9    </body>
10   </html>
```

#### Estructura general d'un document HTML

```

graph TD
A["Inici"] --> B[DOCTYPE html]
B --> C[html]
C --> D["head"]
D --> E["Titol del document"]
C --> F["body"]
F --> G["header"]
F --> H["<h1> Encabezado </h1>"]
F --> I["<p> Contingut de la pàgina </p>"]
F --> J["<footer> Peu de pàgina </footer>"]
```

## 2. Versiones

El origen de **HTML** fue un sistema de hipertexto para compartir documentos electrónicos en **1990**.

La primera propuesta oficial para convertir HTML en un estándar se realizó en **1993**.

Ninguna de las dos primeras propuestas de estándar que se hicieron (**HTML** y **HTML+**) consiguió convertirse en estándar oficial.

A lo largo de los años se publicaron diferentes versiones del estándar.

A continuación se muestran las más importantes:

# HTML



### HTML 1.0

Fue la versión básica de HTML, con soporte para elementos sencillos como **texto e imágenes**.

Era una versión muy limitada, sin características avanzadas relacionadas con la forma en que se representa el contenido en el navegador.

No ofrecía soporte para elementos como **tablas** o **fuentes tipográficas**.

Además, el **W3C** todavía no existía antes de HTML 2.0, por lo tanto no hay detalles oficiales sobre esta primera versión.

---

## HTML 2.0

Fue la **primera versión oficial de HTML**.

El estándar fue publicado por el **IETF (Internet Engineering Task Force)** en **noviembre de 1995**.

---

## HTML 3.2

Publicada el **14 de enero de 1997** por el **W3C**.

Entre sus principales novedades incluía:

- Los *applets* de **Java**.
  - Posibilidad de mostrar **texto alrededor de las imágenes**.
- 

## HTML 4.0

Publicada el **24 de abril de 1998**.

Entre las novedades que presentaba se encontraban:

- Las **hojas de estilo CSS**, utilizadas hoy en día para aplicar estilos a las páginas web.
  - La posibilidad de incluir **pequeños programas** o *scripts* en las páginas web.
- 

## HTML 4.01

Publicada el **24 de diciembre de 1999**.

En ese momento el **W3C** detuvo la actividad de estandarización de HTML hasta **marzo de 2007**, momento en que se retoma gracias al impulso del grupo **WHATWG (Web Hypertext Application Technology Working Group)** y a la publicación de los borradores de **HTML5**.

---

## HTML 5

**HTML5** es la versión más avanzada y considerada el estándar actual.

Ha evolucionado a través de diferentes especificaciones (HTML 5.1, HTML 5.2 y HTML 5.3).

Su lanzamiento inicial fue el **28 de octubre de 2014**, pero se trata de una **versión viva**, en la que se sigue trabajando.

Las especificaciones más recientes se pueden encontrar en la web del **WHATWG**.

### PRINCIPALES CAMBIOS INTRODUCIDOS:

- Introducción de **nuevos elementos y atributos** que reflejan los usos actuales de los sitios web modernos.
- Inclusión de **elementos semánticos**, como `<header>` o `<footer>`, que permiten describir el significado del contenido.
- Incorporación de nuevas **funcionalidades (HTML5 APIs)**, como almacenamiento sin conexión (*offline*), funcionalidad *drag & drop*, uso de la **geolocalización**, etc.
- **Separación entre contenido y presentación**: los estilos se definen mediante **CSS**, y HTML se encarga de la estructura y la semántica.

### Obsolescencia

Esto provoca que muchos elementos y atributos de versiones anteriores queden obsoletos, como `<font>`, `<center>` o el atributo `align`.

### 3. Sintaxis

Un documento HTML está formado por etiquetas y atributos.

#### 3.1 Etiquetas y atributos

##### ETIQUETAS

Las etiquetas permiten estructurar el contenido en un documento HTML.

Una etiqueta (tag) es el identificador de un elemento (element).

Suponiendo el siguiente código HTML:

```
1  <div class="center">Texto</div>
```

El elemento sería `<div class="center">Texto</div>` y la etiqueta `<div>`.

Al igual que en XML, las etiquetas pueden ser:

- De apertura: `<etiqueta>`
- De cierre: `</etiqueta>`

También existen elementos vacíos:

```
1  <input />
```

Una de las diferencias con XML es que la cantidad de etiquetas de HTML está limitada a aquellas que están definidas por el lenguaje.

##### ATRIBUTOS

Aunque HTML define una gran cantidad de etiquetas, éstas no son suficientes para crear páginas complejas, ya que la definición completa de ciertos elementos, como las imágenes y los enlaces, requiere información adicional.

Como no es posible crear una etiqueta por cada elemento diferente, se añade la información adicional a las etiquetas mediante los atributos.

Para cada uno de los atributos hay definido un conjunto de valores que se le puede asignar.

Si el valor de un atributo no es válido, el navegador lo ignora.

Cada una de las etiquetas HTML define los atributos que puede utilizar, aunque algunos de ellos son comunes a muchas etiquetas.

#### Ejemplo práctico de etiqueta y atributo

```
1  
```

En este ejemplo: - `<img>` es la **etiqueta**. - `src` y `alt` son **atributos** que proporcionan información adicional al navegador.

#### 3.2 Tipos de atributos

A continuación, se presenta una clasificación de los atributos comunes según su funcionalidad.

##### ATRIBUTOS BÁSICOS O GLOBALES

Se pueden usar en casi todas las etiquetas HTML.

###### **name**

Permite asignar un nombre a un objeto HTML.

```
1  <input name="password" />
```

**title**

Asigna un título a un elemento HTML, mejorando así la accesibilidad. Dicho título es mostrado por los navegadores cuando el usuario pasa el ratón por encima del elemento. Es especialmente útil con los elementos `a`, `link`, `img`, `object`, `abbr` y `acronym`.

```
1 <a title="Título de etiqueta"></a>
```

**id**

Permite identificar al elemento HTML sobre el que se aplica de forma única mediante un identificador único. Solo es útil cuando se trabaja con CSS y con JavaScript.

```
1 <div id="principal"></div>
```

Los identificadores:

- Pueden empezar por números.
- Solo puede contener letras, números, guiones medios (-) y/o guiones bajos (\_).

**style**

Permite aplicar a un elemento HTML un estilo CSS directamente.

```
1 <span style="color: red;"></span>
```

**class**

Permite aplicar a un elemento HTML los estilos de una clase definida en una hoja de estilos CSS.

```
1 <div class="rojo"></div>
```

```
1 .rojo {
2   color: red;
3 }
```

Los nombres de las clases:

- Pueden empezar por números.
- Solo puede contener letras, números, guiones medios (-) y/o guiones bajos (\_).

**ATRIBUTOS PARA INTERNACIONALIZACIÓN**

Estos atributos se utilizan en las páginas que muestran sus contenidos en varios idiomas, o aquellas que quieren indicar de forma explícita el idioma de sus contenidos.

**dir**

Indica la dirección de escritura del texto. Solo puede tomar dos valores:

- `ltr` (left to right): de izquierda a derecha. Es el valor por defecto.
- `rtl` (right to left): de derecha a izquierda.

**lang**

Especifica el idioma del elemento mediante un código predefinido. Los posibles valores de este atributo se definen en el RFC 1766, el cual hace referencia a la norma ISO 639.

```
1 <html lang="es"></html>
```

Algunos ejemplos de valores posibles son:

- en → Inglés
- es → Español
- ja → Japonés
- fr → Francés

#### **xml:lang**

Al igual que el atributo `lang`, especifica el idioma del elemento mediante un código definido según el RFC 1766.

```
1 <html xml:lang="it"></html>
```

En las páginas XHTML, el atributo `xml:lang` tiene más prioridad que `lang` y es obligatorio incluirlo siempre que se incluya el atributo `lang`.

#### ATRIBUTOS PARA OBTENER EL FOCO

Permite definir el foco en elementos como formularios.

#### **autofocus**

Pone el foco de forma automática en el elemento que lo tenga definido.

```
1 <input autofocus />
```

#### ATRIBUTOS DE EVENTOS

Se entiende por evento aquello que ocurre cuando un usuario interactúa con la página web. Por ejemplo, hacer clic, hacer doble clic, pasar el ratón por encima de un elemento, etc.

Estos atributos solo se utilizan en las páginas web dinámicas, para las cuales se utiliza el lenguaje de programación JavaScript.

Como no es objetivo de la materia, no se va a contemplar.

### 3.3 XHTML



El lenguaje **XHTML** (**eXtensible HyperText Markup Language**) es muy similar al lenguaje HTML. De hecho, no es más que una adaptación de HTML al lenguaje XML.

El estándar XHTML 1.0 solo añade pequeñas mejoras y modificaciones menores al estándar HTML 4.01, por lo que este último está prácticamente incluido en el primero. Pasar del HTML 4.01 Strict a XHTML no requiere casi ningún cambio.

El lenguaje HTML tiene una sintaxis muy permisiva, por lo que es posible escribir sus etiquetas y atributos de muchas formas diferentes. Las etiquetas, por ejemplo, pueden escribirse en mayúsculas, minúsculas e incluso combinando mayúsculas y minúsculas. El valor de los atributos de las etiquetas se puede indicar con o sin comillas. Además, el orden en el que se abrían y cerraban las etiquetas no era importante.

La flexibilidad de HTML da lugar a páginas con un código desordenado, difícil de mantener y muy poco profesional.

XHTML soluciona estos problemas añadiendo ciertas normas en la forma de escribir las etiquetas y atributos. Las normas son las mismas que para un documento XML común bien formado. De esta forma, se consigue:

- Sencillez a la hora de editar y mantener el código.
  - Al ser más regular, es más fácil escribir código que lo procese.
  - Como es XML, se pueden utilizar fácilmente herramientas creadas para procesar documentos XML genéricos (editores, XSLT, etc.).
- 

### 3.4 XHTML vs HTML

Existen unas diferencias entre HTML y XHTML a nivel sintáctico y estructural. En general, los diseñadores web suelen trabajar con HTML. El XHTML es más apreciado por los desarrolladores, que valoran la regularidad adicional. De cualquier manera, los tres primeros puntos de la anterior lista se consideran una buena práctica y se suelen cumplir siempre.

Para que el código HTML se pueda considerar XML bien formado, y por tanto, XHTML, tiene que cumplir:

- Toda la página debe estar contenida en un elemento raíz `<html>`.
- Los nombres de las etiquetas y atributos siempre se escriben en minúsculas.
- El valor de los atributos, incluso los numéricos, siempre se encierra entre comillas.
- En los atributos en los que el nombre coincide con su valor, no puede darse el valor por entendido, es decir, no se pueden comprimir. Este tipo de atributos no son muy habituales. Por ejemplo:

```
1 <div value="value"></div>
```

no es posible ponerlo como:

```
1 <div value></div>
```

- Todas las etiquetas deben cerrarse siempre. XHTML permite que, en lugar de abrir y cerrar de forma consecutiva la etiqueta (`<br><br/>`), se pueda utilizar la sintaxis `<br />` para indicar que es una etiqueta vacía que se abre y se cierra en ese mismo punto.

Por otro lado, hay que tener en cuenta que los navegadores no procesan HTML y XHTML exactamente igual:

- Si un documento HTML contiene errores, el navegador intentará mostrar el mayor contenido posible.
- Si un documento XHTML contiene algún error, el navegador no mostrará el documento.

#### Error común

No olvides **cerrar correctamente las etiquetas** en HTML y **seguir la sintaxis XML** en XHTML.

Los navegadores son tolerantes con HTML, pero en XHTML **un solo error impide que se muestre la página**.

#### MIME TYPE

Un navegador web sabe si el documento que se está sirviendo es un HTML o un XHTML por el MIME (Multipurpose Internet Mail Extensions). MIME es una cabecera del protocolo HTTP que indica al navegador qué tipo de documento se está enviando.

El MIME para un documento HTML es el siguiente:

```
1   text/html
```

El MIME para un documento XHTML es alguno de los dos siguientes:

```
1   application/xhtml+xml
2   application/xml
```