



Reglas y normas HTML5

-
- C.F.G.S. DAW
 - 6 horas semanales
 - Mes aprox. de impartición: Oct - Nov
 - iPasen - cjaedia071@g.educaand.es

Carmelo José Jaén Díaz

Índice



Objetivo

Glosario

Interacción persona - ordenador

Objetivos

Características. Usable.

Características. Visual.

Características. Educativo y actualizado.

OBJETIVO



- Analizar y seleccionar los colores y las tipografías adecuados para la visualización en la pantalla.
- Utilizar marcos y tablas para presentar la información de manera ordenada.
- Identificar nuevos elementos y etiquetas en HTML5.
- Reconocer las posibilidades de modificar etiquetas HTML.
- Valorar la utilidad de las hojas de estilo para conseguir un diseño uniforme en todo el sitio web.

GLOSARIO



Formularios. Documentos interactivos utilizados para recoger información en un sitio web. Esta información es enviada al servidor, donde es procesada. Cada formulario contiene uno o varios tipos de controles que permiten recolectar la información de varias formas diferentes.

Fuentes seguras. Fuentes tipográficas que los usuarios tenían instaladas por defecto en su dispositivo. En la actualidad, gracias a que la mayoría de los navegadores soportan la directiva @font-face, es posible utilizar casi cualquier tipografía a través de Google Fonts.

Guías de estilo. Documentos con directrices que permiten la normalización de estilos. En estas guías se recogen los criterios y normas que debe seguir un proyecto; de esta forma se ofrece una apariencia más uniforme y atractiva para el usuario.

HTML. Lenguaje de marcado de hipertexto utilizado en las páginas web. Este tipo de lenguaje presenta una forma estructurada y agradable, con hipervínculos que conducen a otros documentos y con inserciones multimedia (sonido, imágenes, vídeos...).

GLOSARIO



HTML5. Última versión del lenguaje para la programación de páginas web HTML. Los sitios implementados con este lenguaje solo pueden visualizarse correctamente en los navegadores más actuales.

Legibilidad. Cualidad deseable en una familia tipográfica. Se trata de la facilidad de la lectura de una letra. Esta cualidad puede venir determinada por varios parámetros como el interletrado, el interpalabrado o el interlineado.

Marcos. Son las ventanas independientes incorporadas dentro de la página general. Gracias a ellos, cada página quedará dividida en varias subpáginas, permitiendo realizar un diseño más organizado y limpio a la vista. Con HTML5 ha quedado obsoleto.

Tipografía. Se trata del tipo de letra que se escoge para un determinado diseño. Según la RAE, significa "modo o estilo en que está impreso un texto" o "clase de tipos de imprenta".

INTRODUCCIÓN



Seguir las reglas y normas de HTML5 es fundamental para garantizar que el código sea limpio, legible y fiable. Estas normas no solo facilitan la colaboración y el mantenimiento de proyectos, sino que también aseguran que las páginas web funcionen correctamente en todos los navegadores y dispositivos.



REGLAS Y NORMAS HTML5



Con el aumento de la complejidad y la interactividad en los sitios web, es más importante que nunca adherirse a las mejores prácticas de HTML5 para ofrecer una experiencia de usuario óptima y eficiente. Veamos las reglas y normas que nos ayudan a seguir buenas prácticas en la implementación de código HTML:

- **Estructura básica:** Todo documento HTML5 debe tener una estructura básica que incluya las etiquetas `<!DOCTYPE html>`, `<html>`, `<head>`, y `<body>`. La etiqueta `<html>` es el elemento raíz y debe contener todas las demás etiquetas.
- **Etiquetas en mayúsculas o minúsculas:** Las etiquetas HTML no distinguen entre mayúsculas y minúsculas. Por lo tanto, `<HTML>` y `<html>` son equivalentes. Se recomienda seguir siempre la misma sintaxis, preferiblemente letras en minúsculas.

REGLAS Y NORMAS HTML5



- **Caracteres especiales:** Algunos caracteres especiales, como `<`, `>`, `&`, `"`, y `'`, tienen significados especiales en HTML. Para incluir estos caracteres en tu código, puedes usar entidades HTML (`<` representa el símbolo `<`), su código decimal (`<` representa el símbolo `<`) o hexadecimal. Ver cómo representar caracteres especiales.
- **Código con tabulados y sangrías:** El código HTML indentado con tabulados facilita la lectura y su comprensión. Además, puede ayudar a evitar errores, ya que facilita la identificación de los elementos HTML. Por ello, cuando escribimos código HTML utilizamos la tabulación para entender mejor la estructura de etiquetas. Sin embargo, hay que tener en cuenta también que la **velocidad de carga** es uno de los factores más influyentes en la experiencia que reciben los usuarios. Por ello, para **reducir el tamaño de los ficheros HTML** hay que evitar los tabulados, los espacios en blanco y los comentarios innecesarios. Una vez implementado nuestro código, se puede minificar mediante herramientas específicas.

REGLAS Y NORMAS HTML5



- Los saltos de línea y los espacios no tienen ningún efecto en el código HTML: Los navegadores web ignoran los saltos de línea y los tabuladores en el código HTML contenido entre las etiquetas `<body></body>`. Para escribir espacios en blanco podemos usar ` ` y para escribir saltos de línea la etiqueta `
`.
- Uso de comillas: Aunque HTML5 permite el uso de comillas simples (') alrededor de los valores de los atributos, es una buena práctica utilizar comillas dobles (") para que el código sea más consistente.
`Enlace`
- Pruebas en diferentes navegadores: Prueba tu sitio web en diferentes navegadores y dispositivos para garantizar que se vea y funcione correctamente en todas partes.

REGLAS Y NORMAS HTML5



- **Validación del código:** Utiliza herramientas de validación HTML para garantizar que tu código cumpla con las especificaciones y estándares HTML. Un código HTML válido es más compatible y menos propenso a errores. Te recomendamos utilizar el Validador de W3C para comprobar la calidad y estructura de tu código HTML. Este proceso se abordará más detalladamente en la unidad de test de verificación y herramientas, donde explicaremos cómo utilizar validadores y otras herramientas útiles para asegurar la calidad de tu proyecto.
- **Implementar los factores HTML clave para el SEO:** La estructura y las prácticas de HTML5 pueden influir en la optimización de motores de búsqueda (SEO). Es importante asegurarse de que el código esté optimizado para SEO para mejorar la visibilidad y el ranking de los motores de búsqueda.

REGLAS Y NORMAS XHTML



XHTML (Extensible Hypertext Markup Language) es una versión de HTML que sigue reglas y normas específicas para asegurar que los documentos web sean bien formados y cumplan con los estándares web.

Aunque HTML5 ha reemplazado en gran medida a XHTML en el desarrollo web moderno, comprender las reglas y normas de XHTML sigue siendo importante por varias razones. XHTML, o Extensible Hypertext Markup Language, es una versión más estricta de HTML basada en XML.

Además, el conocimiento de XHTML proporciona una base sólida para aprender y dominar HTML5 y otras tecnologías web modernas. Muchos principios de XHTML son aplicables a HTML5, facilitando la transición a nuevos estándares. La comprensión de XHTML también ayuda a mejorar la validez y el rendimiento del código, identificar errores y optimizar el desarrollo web.

REGLAS Y NORMAS XHTML



HTML5 es más permisivo en términos de sintaxis que XHTML. Es una buena práctica seguir las convenciones de HTML5 en lugar de mezclarlas con las de XHTML.

Veamos algunas reglas y normas clave de XHTML para que sepas diferenciarlas con respecto a HTML5:

- **Document Type Declaration (DTD):** Debes incluir una declaración de tipo de documento (DOCTYPE) al comienzo de tu documento XHTML. Esto especifica la versión de XHTML que estás utilizando y establece el estándar para el documento.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

REGLAS Y NORMAS XHTML



- Etiquetas en minúsculas: XHTML requiere que todas las etiquetas y atributos se escriban en minúsculas. Aunque HTML no es tan estricto, XHTML lo exige.

`<p>Este es un párrafo de ejemplo en XHTML.</p>`

- Elementos anidados y cerrados: Debes asegurarte de que todos los elementos se abran y cierren correctamente y que estén anidados de manera adecuada. No se permiten elementos mal anidados en XHTML.

```
<ul>
  <li>Elemento 1</li>
  <li>Elemento 2</li>
</ul>
```

REGLAS Y NORMAS XHTML



- **Atributos con comillas:** Las comillas pueden ser dobles o simples (pero siempre formando parejas con las mismas).

```

```

- **Elementos vacíos:** Los elementos que no tienen contenido (etiquetas de autocierre o vacías) deben cerrarse con una barra diagonal antes del corchete de cierre (por ejemplo, ``, `
`) mientras que en HTML5 no es necesario. Ver más información sobre el cierre de etiquetas en HTML5 y XHTML y el uso de la barra inclinada.

```
<br />
```

REGLAS Y NORMAS XHTML



- Uso de entidades: Debes utilizar entidades HTML para representar caracteres especiales, como `<` para `<` y `>` para `>`.

`<p>`Este es un ejemplo de ``texto enfatizado`` en XHTML.`</p>`

- Elementos vacíos: Los elementos que no tienen contenido (etiquetas de autocierre o vacías) deben cerrarse con una barra diagonal antes del corchete de cierre (por ejemplo, ``, `
`) mientras que en HTML5 no es necesario. Ver más información sobre el cierre de etiquetas en HTML5 y XHTML y el uso de la barra inclinada.

`
`

REGLAS Y NORMAS XHTML



- Uso de entidades: Debes utilizar entidades HTML para representar caracteres especiales, como `<` para `<` y `>` para `>`.
`<p>Este es un ejemplo de texto
enfaticado en XHTML.</p>`
- Atributos: No se pueden minimizar los atributos de los elementos. Por ejemplo, en HTML es válido `<OPTION
VALUE=»Mi valor» SELECTED>`, en XHTML debe escribirse: `<option value=»Mi
valor» selected=»selected» />`

REGLAS Y NORMAS XHTML



- **Uso de etiquetas de cierre:** Todas las etiquetas que requieren cierre, como `<p>`, ``, `<div>`, deben cerrarse correctamente. En HTML5, algunas etiquetas que requieren un cierre en XHTML, como `<p>` o ``, pueden no cerrarse explícitamente, y el navegador las interpretará correctamente.
 - XHTML: `<p>Este es un párrafo.</p>`
 - HTML5: `<p>Este es un párrafo.`

Veamos un ejemplo de código HTML y su equivalente en XHTML:

HTML y XHTML

Diferencias



documento.html

```
<!DOCTYPE html>
<html>
<head>
  <title>Ejemplo HTML</title>
</head>
<body>
  <h1>Título del artículo</h1>
  <p>Este es un párrafo de ejemplo.</p>
  <ul>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
  </ul>
  <br>
</body>
</html>
```

documento.xhtml

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title>Ejemplo XHTML</title>
</head>
<body>
  <h1>Título del artículo</h1>
  <p>Este es un párrafo de ejemplo.</p>
  <ul>
    <li>Elemento 1</li>
    <li>Elemento 2</li>
  </ul>
  <br />
</body>
</html>
```

HTML y XHTML

Diferencias



- **Declaración DOCTYPE:** En XHTML, debes especificar una declaración DOCTYPE que apunte a la versión de XHTML que estás utilizando. En el ejemplo, estamos utilizando la versión «XHTML 1.0 Strict».
- **Namespace XML:** En XHTML, debes agregar un atributo **xmlns** al elemento **<html>** para especificar el espacio de nombres XML. Esto no es necesario en HTML.
- **Etiquetas en minúsculas:** XHTML requiere que todas las etiquetas y atributos se escriban en minúsculas, como se muestra en el ejemplo.
- **Elementos vacíos:** Debes cerrar los elementos vacíos con una barra diagonal antes del corchete de cierre en XHTML, como **
**.

Como podemos ver XHTML es más estricto en cuanto a la estructura y la sintaxis que HTML, por lo que debes seguir estas normas y reglas para asegurarte de que tu documento sea bien formado y cumpla con los estándares.

HTML y XHTML

Diferencias



Los espacios de nombres (namespaces) son fundamentales en XHTML, ya que se basa en XML, un lenguaje que requiere documentos bien formados y estructurados. Un espacio de nombres se declara utilizando el atributo **xmlns** en la etiqueta **<html>**, y su función principal es evitar conflictos cuando se utilizan múltiples vocabularios en el mismo documento. Por ejemplo, en XHTML, la declaración **xmlns="http://www.w3.org/1999/xhtml"** indica que el documento sigue las reglas del vocabulario XHTML. En HTML5, los espacios de nombres no son obligatorios ni se utilizan de la misma manera, ya que HTML5 es más flexible y menos estricto en cuanto a la estructura y sintaxis.

¿SE USA XHTML EN LA ACTUALIDAD?



XHTML no se utiliza ampliamente en la actualidad para el desarrollo web. XHTML fue una especificación del World Wide Web Consortium (W3C) que combinaba la sintaxis de XML con la de HTML. Se introdujo como una evolución de HTML para que los documentos web fueran más estrictos y estuvieran bien formados según las reglas de XML.

Sin embargo, a medida que la web evolucionó, HTML5 se convirtió en la especificación dominante para el desarrollo web. HTML5 ofrece muchas características nuevas y mejoradas, y ha sido ampliamente adoptado por los desarrolladores web y los navegadores.

Las principales razones por las que XHTML ha dejado de utilizarse ampliamente son:

Más estricto: Aunque XHTML se diseñó para ser más estricto que HTML, en la práctica, los navegadores no eran tan rigurosos en la validación de XHTML. Esto llevó a que los desarrolladores continuaran utilizando

¿SE USA XHTML EN LA ACTUALIDAD?



Las principales razones por las que XHTML ha dejado de utilizarse ampliamente son:

- **Más estricto:** Aunque XHTML se diseñó para ser más estricto que HTML, en la práctica, los navegadores no eran tan rigurosos en la validación de XHTML. Esto llevó a que los desarrolladores continuaran utilizando HTML más tradicional y menos estricto.
- **Mayor complejidad:** La sintaxis más estricta de XHTML a menudo resultaba en documentos más complejos y difíciles de mantener, lo que no era práctico en muchos casos.
- **HTML5 es más flexible:** HTML5 ofrece muchas de las mismas ventajas que se buscaban con XHTML, como soporte para XML, pero sin la complejidad y las restricciones de XHTML.

VENTAJAS DE LOS ESPACIOS DE NOMBRES



Es importante mencionar algunas de las ventajas que ofrecen los espacios de nombres en lenguajes como XML y XHTML:

- **Evitan conflictos de nombres:** Los espacios de nombres permiten usar múltiples vocabularios en un mismo documento sin generar confusión entre elementos o atributos con nombres similares pero de diferentes orígenes.
- **Claridad y organización:** Facilitan la diferenciación de los elementos al indicar claramente a qué vocabulario pertenece cada etiqueta o atributo, lo que mejora la legibilidad y el mantenimiento del código.
- **Modularidad:** Hacen posible la integración de distintos lenguajes en un solo documento, como incluir gráficos SVG o ecuaciones MathML junto con XHTML, garantizando que cada conjunto de etiquetas se interprete correctamente.

VENTAJAS DE LOS ESPACIOS DE NOMBRES



- **Compatibilidad y estandarización:** Al seguir las reglas de XML, los documentos que usan espacios de nombres son más compatibles con diversas herramientas y tecnologías, asegurando que los datos sean bien formados y validados según los estándares.
- **Interoperabilidad:** Los espacios de nombres permiten que aplicaciones y sistemas procesen datos de diferentes fuentes de manera precisa, facilitando la interacción entre lenguajes de marcas.

HTML en sí no utiliza espacios de nombres de la misma manera que lo hacen XML o XHTML. Sin embargo, HTML5 es compatible con los espacios de nombres cuando se combinan diferentes vocabularios basados en XML, como SVG (gráficos vectoriales escalables) o MathML (lenguaje de marcado matemático), que pueden integrarse en documentos HTML. En estos casos, los espacios de nombres se utilizan para diferenciar los elementos de HTML de los de otros lenguajes.

VENTAJAS DE LOS ESPACIOS DE NOMBRES



Por ejemplo, cuando se incrusta un gráfico SVG en un documento HTML5, se utiliza el espacio de nombres `xmlns="http://www.w3.org/2000/svg"` para que el navegador sepa cómo interpretar los elementos SVG:

```
<svg xmlns="http://www.w3.org/2000/svg" width="100" height="100">
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4"
fill="yellow" />
</svg>
```

Aunque HTML5 no requiere espacios de nombres para su propio vocabulario, los soporta cuando se usan otros lenguajes basados en XML dentro de documentos HTML.