

CLASIFICACIÓN DE LOS LENGUAJES DE MARCAS

Marcado de presentación

Indica el formato del texto. Este tipo de marcado es útil para maquetar la presentación de un documento para su lectura, pero resulta insuficiente para el procesamiento automático de la información. El marcado de presentación resulta más fácil de elaborar, sobre todo para cantidades pequeñas de información. Sin embargo, resulta complicado de mantener o modificar, por lo que su uso se ha ido reduciendo en proyectos grandes en favor de otros tipos de marcado más estructurados. Ejemplo: Word.

Marcado de procedimientos

Está enfocado hacia la presentación del texto, aunque también es visible para el usuario que edita el texto. El programa que representa el documento debe interpretar el código en el mismo orden en que aparece. Ejemplos: LaTeX, HTML.

Marcado descriptivo

Utiliza etiquetas para describir los fragmentos de texto, pero sin especificar cómo deben ser representados, o en qué orden. Ejemplos: XML.

EVOLUCIÓN DEL HTML

Se trata de un lenguaje de marcas utilizado para la creación de páginas web. Su origen se remonta a 1980, cuando el físico Tim Berners-Lee propuso un nuevo sistema de "hipertexto" para compartir documentos.

El primer documento formal con la descripción de HTML se publicó en 1991 bajo el nombre HTML Tags. La primera propuesta oficial para convertir HTML en un estándar se realizó en 1993. Aunque se consiguieron avances significativos ninguna de las dos propuestas de estándar, llamadas HTML y HTML+, consiguieron convertirse en estándar oficial.

En 1995, el organismo IETF organiza un grupo de trabajo de HTML y consigue publicar el estándar HTML 2.0, el primer estándar oficial de HTML. En 1997 se publica la versión 3.2 por el W3C. Esta revisión incorpora los últimos avances de las páginas web desarrolladas hasta 1996, como applets de Java y texto que fluye alrededor de las imágenes.

HTML 4.0 se publica en 1998 y supone un gran salto desde las versiones anteriores. Entre sus novedades más destacadas se encuentran las hojas de estilos CSS, la posibilidad de incluir pequeños programas o scripts en las páginas web, mejora de la accesibilidad de las páginas diseñadas, tablas complejas y mejoras en los formularios. La última especificación oficial de HTML se publica en 1999 y se denomina HTML 4.01. Se trata de una revisión y actualización de la versión HTML 4.0, por lo que no incluye novedades significativas.

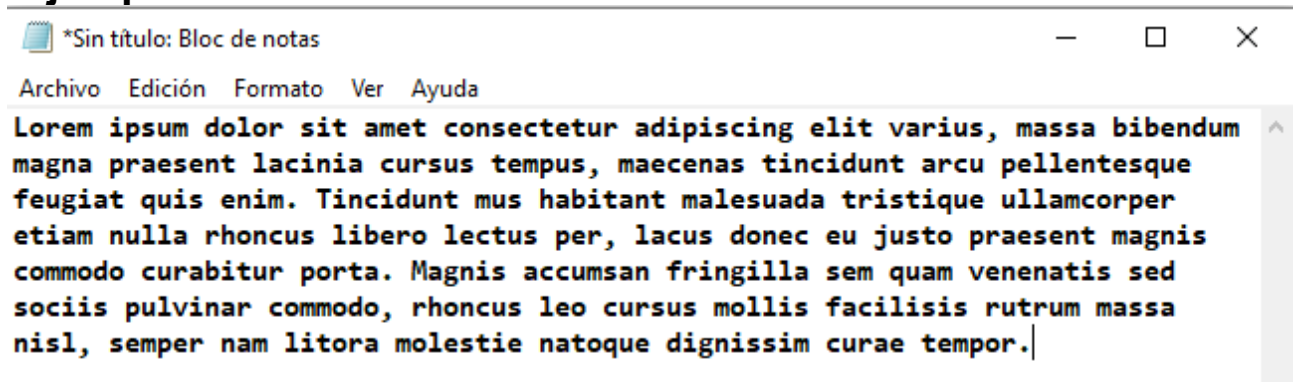
Hasta 2008 no se publica el primer borrador oficial de HTML 5. Paralelamente se desarrolla una versión avanzada de HTML y basada en XML llamada XHTML.

COMPARACIÓN HTML Y XHTML

HTML y XHTML son las dos variantes que existen en el lenguaje HTML 5. Las diferencias entre esas dos variantes son básicamente sintácticas: la sintaxis HTML está inspirada en la norma SGML (aunque no la cumple estrictamente), mientras que la sintaxis XHTML está basada en la recomendación XML (aunque tampoco la cumple estrictamente). En general, la sintaxis XHTML es más "estricta", en el sentido de imponer restricciones en la forma de escribir etiquetas, atributos o valores, mientras que la sintaxis HTML es más "flexible". Estas son algunas diferencias:

- En XHTML no puede haber texto no incluido en alguna etiqueta, pero en HTML puede haberlo.
- En XHTML, los nombres de elementos y atributos deben escribirse en minúsculas. En HTML no es necesario.
- En XHTML, las referencias a entidades en hexadecimal deben escribirse en minúsculas. En HTML se pueden escribir en minúsculas o mayúsculas.
- Para evitar problemas con los caracteres "<" y "&", en XHTML el código javascript (y de estilo) se suele incluir en secciones CDATA. En HTML no es necesario.
- Los documentos XHTML deben tener un único elemento raíz de todo el documento, el elemento <html>. Los documentos HTML pueden no tenerlo.
- En XHTML, los elementos no vacíos (los que tienen etiqueta de apertura) deben tener etiqueta de cierre. En HTML no es necesario.
- En XHTML, la etiqueta de apertura de los elementos vacíos debe acabar en " />" o debe haber una etiqueta de cierre. En HTML no puede haber etiqueta de cierre.
- En XHTML, los atributos deben tener siempre valor (en el caso de atributos booleanos, el valor debe ser el propio nombre del atributo o una cadena vacía). En HTML, los atributos pueden no tener valor asignado.
- En XHTML los valores de los atributos deben escribirse en minúsculas. En HTML se pueden escribir en minúsculas o mayúsculas.
- En XHTML, los valores de los atributos deben escribirse entre comillas (dobles o simples). En HTML no es necesario si no hay espacios en blanco.

Ejemplo de un fichero de texto sin formato



```
*Sin título: Bloc de notas

Archivo Edición Formato Ver Ayuda

Lorem ipsum dolor sit amet consectetur adipiscing elit varius, massa bibendum
magna praesent lacinia cursus tempus, maecenas tincidunt arcu pellentesque
feugiat quis enim. Tincidunt mus habitant malesuada tristique ullamcorper
etiam nulla rhoncus libero lectus per, lacus donec eu justo praesent magnis
commodo curabitur porta. Magnis accumsan fringilla sem quam venenatis sed
sociis pulvinar commodo, rhoncus leo cursus mollis facilisis rutrum massa
nisl, semper nam litora molestie natoque dignissim curae tempor.
```

Pon un ejemplo de un fichero XML

```
<?xml version="1.0"?>
<Catalog>
  <Book id="bk101">
    <Author>Garghentini, Davide</Author>
    <Title>XML Developer's Guide</Title>
    <Genre>Computer</Genre>
    <Price>44.95</Price>
    <PublishDate>2000-10-01</PublishDate>
    <Description>An in-depth look at creating applications
    with XML.</Description>
  </Book>
  <Book id="bk102">
    <Author>Garcia, Debra</Author>
    <Title>Midnight Rain</Title>
    <Genre>Fantasy</Genre>
    <Price>5.95</Price>
    <PublishDate>2000-12-16</PublishDate>
    <Description>A former architect battles corporate zombies,
    an evil sorceress, and her own childhood to become queen
    of the world.</Description>
  </Book>
</Catalog>
```

Estructura básica de un documento HTML

```
<HTML>
  <HEAD>
    <TITLE>Título de la página</TITLE>
    ...
  </HEAD>

  <BODY>
    Aquí iría el contenido de la página
  </BODY>
</HTML>
```

Estructura básica de un documento XHTML

```
<?xml version="1.0" encoding="ISO-8859-15"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
"http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html version="xhtml 1.1"
  xml:lang="es"
  xmlns="http://www.w3.org/1999/xhtml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/MarkUp/SCHEMA/xhtml11.xsd">
  <head>
    <title>Documento XHTML básico</title>
  </head>
  <body>
  </body>
</html>
```

Comparación de XML con SGML

SGML (**S**tandard **G**eneralized **M**arkup **L**anguage o "Estándar de Lenguaje de Mercado Generalizado"):

- SGML es muy potente pero también muy complejo.
- No ha sufrido modificaciones desde su creación en 1986.
- Soporte habil en indexacion, busqueda y recuperación.
- En el cliente tiene problemas en el procesamiento.

HTML (**H**yper**T**ext **M**arkup **L**anguage "Lenguaje de Mercado de Hipertexto"):

- Procesamiento facil del lado del cliente.
- Problemas con la presentacion semantica.
- Etiquetas fijas que determina el aspecto de los documentos.
- Simplicidad, constuido solamente para publicación.
- Facilidad de compatibilidad con otros lenguajes.

XML (**eX**tensible **M**arkup **L**anguage o "lenguaje de marcas extensible"):

- Etiquetado especifico, vocabulario propio para el etiquetado.
- No determina el aspecto del documento.
- Separación entre el diseño lógico y el diseño de presentación.
- Permite redactar metainformación (Información de la información)

Como se puede apreciar, las diferencias están en que SGML es el lenguaje estándar de los marcados de texto y desde su creación no ha sido modificado lo que hace a su vez sea el lenguaje mas robusto. HTML es simple por que fue diseñado para publicación, pero es mas accesible para el lado del cliente en la presentación, aun a pesar de su simplicidad puede ser una herramienta muy potente si se combina con otros lenguajes. XML en cambio es la evolucion del SGML, que además permite añadir metainformación a los textos.

Enumera varios programas que se pueden utilizar como visor de ficheros XML.

- Usar un editor de texto.
- Usar un navegador web.
- Usar Excel.
- Usar un visor de XML como XMLExplorer.