Introducción a XML

XML

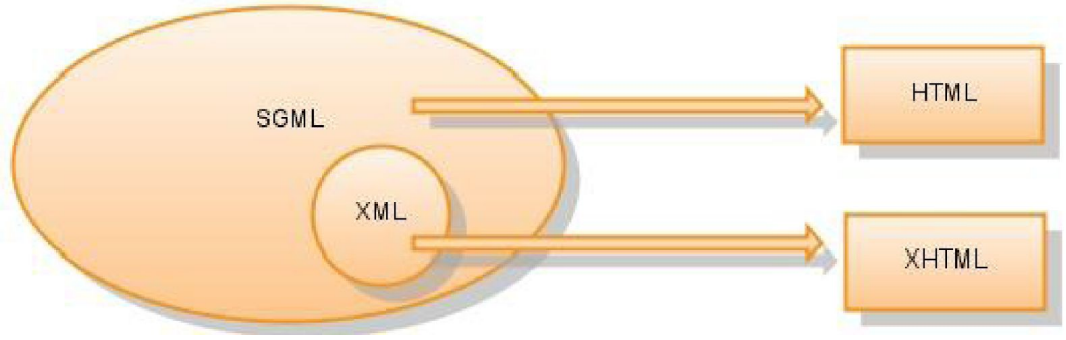
XML es el acrónimo de *eXtensible Markup Language*, o lo que es lo mismo: Lenguaje Extensible de Marcado. Sirve para codificar documentos a través de marcas textuales (etiquetas) de manera que se añade cierta información a la propia información del

documento. Esta nueva información se puede referir a su estructura, presentación, significado,…

Historia

IBM creó el lenguaje GML, que posteriormente se estandarizó en SGML (sta*ndard Generalized Markup Language) (*Estándar de Lenguaje de Marcado Generalizado). El SGML no es un lenguaje, sino un metalenguaje, que sirve para construir lenguajes de marcas concretos para almacenar tipos de documentos reales. El SGML sirvió como base para crear multitud de lenguajes de marcas, entre ellas HTML.
SGML se convirtió en un lenguaje muy potente y complejo y por lo tanto muy difícil de aprender y utilizar.
Por eso se creó XML, que no es más que un subconjunto de SGML, creado para simplificar la creación de gramáticas de lenguajes de marcado.

Igual que HTML se creó en base a SGML, el XHTML se creó en base a XML.



El xHTML tiene la misma funcionalidad que HTML pero cumpliendo las estrictas especificaciones del XML.

Codificación de documentos

Unicode define tres formas de codificación bajo el nombre **UTF** o Formato de Transformación Unicode (*Unicode Transformation Format*):

- UTF-8 — codificación orientada a byte con símbolos de longitud variable.
- UTF-16 — codificación de 16 bits de longitud variable
- UTF-32 — codificación de 32 bits de longitud fija.

Usos de XML

El XML se puede usar para infinidad de trabajos y aporta muchas ventajas en amplios escenarios. Veamos algunas ventajas del XML en algunos campos prácticos.

- Comunicación de datos. Si la información se transfiere en XML, cualquier aplicación podría escribir un documento de texto plano con los datos que estaba manejando en formato XML y otra aplicación recibir esta información y trabajar con ella.
- Migración de datos. Si tenemos que mover los datos de una base de datos a otra sería muy sencillo si las dos trabajasen en formato XML.
- Aplicaciones web. Hasta ahora cada navegador interpreta la información a su manera y los programadores web tienen que hacer unas cosas u otras en función del navegador del usuario. Con XML tenemos una sola aplicación que maneja los datos y para cada navegador o soporte podremos tener una hoja de estilo o similar para aplicarle el estilo adecuado. Si mañana nuestra aplicación debe correr en WAP solo tenemos que crear una nueva hoja de estilo o similar.

EL LENGUAJE XML

El lenguaje XML se creó para estructurar, almacenar y transportar información. Un documento XML no es más que información textual acotada por etiquetas.
Un documento XML no tiene etiquetas preestablecidas. XML permite al autor del documento definir su estructura y crear sus propias etiquetas.

Las distintas aplicaciones que utilizan estos archivos pueden acceder a la información o gestionarla como les interese: enviarla, transformarla, visualizarla, …

Se dice que es extensible porque se pueden incorporar nuevos contenidos a los documentos (nuevas etiquetas o atributos) sin tener que cambiar las aplicaciones que los utilizan.

Los elementos: etiquetas, atributos y contenido

Las partes que componen un documento XML, son los elementos, y se forman con etiquetas: textos reconocidos por el analizador y delimitados por los caracteres < >.
Los elementos pueden tener contenido (otros elementos, texto o ambos), y pueden ser elementos vacíos. Opcionalmente también pueden incorporar pares atributo=”valor”.

Elemento con contenido:

```
<nombreEtiqueta>
        Contenido
</nombreEtiqueta>
```

Elemento sin contenido:

```
<nombreEtiqueta …… />
```

Elemento con atributos y contenido:

```
<nombreEtiqueta [atributo1=”valor1”] [atributo2=”valor2”] …… >
        Contenido
</nombreEtiqueta>
```

Elemento con atributos y sin contenido:

<nombreEtiqueta [atributo1=”valor1”] [atributo2=”valor2”] ……/>

Ejemplos:

<titulo>XML</titulo>
<cantidad unidades=”kg”>70</cantidad>
<asignatura numHoras=”10”>Programación</asignatura>

Aunque cada documento XML tiene sus propios elementos, atributos y estructura adaptada a su finalidad, existe un atributo especial que se suele utilizar en todos ellos cuando se quiere indicar el idioma en el que está escrito el contenido. Este atributo se llama **xml:lang** y su valor es el código de dos letras que identifica el idioma (es, fr, …)

<descripcion xml:lang=”es”>…….</descripcion>

En un documento XML los elementos deben cumplir unas reglas:

- Se pueden anidar y se cierran en orden inverso a su apertura.
- Los elementos sin contenido se deben abrir y cerrar en el mismo momento: <etiqueta/>
- Un atributo no puede aparecer más de una vez en un elemento.
- Los valores de los atributos irán siempre entre comillas dobles o sencillas.
- Todos los elementos del documento se anidan en forma de árbol, existiendo un único elemento raíz del que cuelgan todos los demás.
- Los documentos XML pueden contener comentarios. <!—comentario -->

Los nombre de los elementos y atributos deben cumplir:

- Comenzar siempre por una letra.
- No pueden comenzar por XML ni variantes de mayúsculas/minúsculas.
- No puede contener espacios.
- Distinguen entre mayúsculas/minúsculas.

Las entidades

Las entidades son estructuras XML con un nombre asociado. Para referenciar una entidad se hace con la siguiente sintaxis:

&nombreDeEntidad;

Al referenciar una entidad por su nombre se añade automáticamente su contenido en el lugar de su referencia.

Entidades predefinidas:

Carácter	Referencia
&	&amp;
<	&lt;
>	&gt;
"	&quot;
'	&apos;
	&#codChar;

A veces el uso de entidades puede ser muy incómodo y la lectura del documento muy complicada. Por eso en ocasiones se puede utilizar una sección <![CDATA[texto no procesable....]]> para que no sea procesado como parte de XML. Esto permite utilizar los caracteres < y > por ejemplo y no serán considerados como separadores de etiquetas.

Ejemplo:

```
<codigoFuente>&lt;H1&gt;El operador de concatenación:
&amp;&lt;/H1&gt;</codigoFuente>
```

Se puede sustituir por:

```
<códigoFonte>
<![CDATA[
        <H1>El operador de concatenación:&</H1>
]]>
</códigoFonte>
```

Espacios de nombres

Los espacios de nombres son una recomendación W3C (*World Wide Web Consortium*) para que los nombres comunes de elementos y etiquetas no colisionen, bien porque provienen de distintos documentos XML a un posible documento destino, bien porque nos interesa separar de una manera estructurada el contenido de un documento. (Son colecciones de nombres. Permiten cualificar los nombres.)

Es decir, si a cada uno de los elementos o atributos se les asigna un espacio de nombres (donde si deben ser únicos), en el caso de combinarlos no colisionarían. Así por ejemplo, podremos tener una misma etiqueta llamada <direccion> para clientes y proveedores y no se confundirían puesto que cada una estará precedida por su espacio de nombres.

La definición de un espacio de nombres se realiza a través del atributo **xmlns** (*XML NameSpacing*) cuyo valor será una URL única (*no se comprueba su existencia*).

Se puede referenciar un espacio de nombres para una etiqueta y su contenido definiéndolo directamente como atributo en la propia etiqueta:
De esta manera, cada etiqueta irá precedida de su espacio de nombres. Pero esto hace que la legibilidad de los documentos se vea seriamente comprometida, puesto que las etiquetas pueden tener la forma

<**http://www.xunta.es/cursos:**direccion>.

Para ello crearemos un prefijo más corto utilizando el atributo xmlns de la forma xmlns:prefixo=”valor URL”.
Ejemplo:

```
<xg:cursos xmlns:xg="http://www.xunta.es/cursos">
        <xg:curso nombre =”C#”>
                <xg:alumno ...
                ...
                </xg:alumno ...
        ...
        </xg:curso>
...
</xg:cursos>
```

Los distintos espacios de nombres utilizados en un documento se definen en el elemento raíz con tantos atributos xmlns: como sean necesarios.
Si un atributo xmlns no define un prefijo sino que simplemente tiene un valor: xmlns=”http://www.....”, definen el espacio de nombres por defecto.

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<document xmlns:jorge="http.//www.jorgesanchez.net/document"
                xmlns:html="http://www.w3c.org/html">
<jorge:title>
        Documento de prueba
</jorge:title>
<jorge:content>
    <html:html>
        <html:head>
```

En el ejemplo se usa el prefijo *jorge* para indicar etiquetas del espacio de nombres *www.jorgesanchez.net/document* y *html* para el espacio de nombres de HTML.

Ejemplo:

```
<?xml version="1.0" encoding="UTF-8"?>
<document xmlns ="http.//www.jorgesanchez.net/document"
                xmlns:html="http://www.w3c.org/html">
<title>
    Documento de prueba
</title>
<content>
```

<html:html> <html:head>
Las etiquetas sin prefijo se entiende que pertenecen al espacio de nombres *www.jorgesanchez.net/document*, para las del otro espacio se usa el prefijo.

El documento XML

Prólogo	Declaración XML Tipo de documento Instrucciones de procesamiento
Cuerpo	

Prólogo: Declaración XML

<?xml version="1.0" encoding="UTF-8" ?>

Los atributos son:

- version: la versión del XML utilizada (habitualmente 1.0, ya disponible la 1.1).
- encoding: se refiere al sistema de codificación utilizado.

Prólogo: tipo de documento

1.

```
<!DOCTYPE HTML PUBLIC "-/ /W3C/ /DTD HTML 3.2 Final/ /EN">
```

2.

```
<!DOCTYPE elementoRaiz SYSTEM "definicion.dtd">
<!DOCTYPE clientes SYSTEM  "http://www.servidor.com/defs/clientes.dtd">
```

3.

```
<!DOCTYPE elementoRaíz [
        definición del DTD
]>
```

DOCTYPE especifica el nombre del elemento raíz del documento y, a continuación, el tipo de documento mediante alguno de los siguientes elementos:

1. PUBLIC es el identificador público que hace referencia al DTD. El documento es un estándar.
2. SYSTEM es un URI (Identificador Universal de Recursos) localizando la ruta del DTD.

3. el DTD integrado en el propio documento XML.

Prólogo: instrucciones de procesamiento

Su sintaxis es <*?fuente instrucciones?*>, donde fuente es el nombre de la aplicación que interpretará las instrucciones. Su nombre no puede comenzar por ninguna combinación de mayúsculas/minúsculas de "xml" puesto que están reservadas para los estándares XML.

Ejemplos de instrucciones de procesamiento muy utilizadas son:

- <?xml-stylesheet ... ?> para aplicar una hoja de estilo al documento.
- <?php ... ?>para añadir código PHP

Por ejemplo, si queremos usar una hoja de estilo CSS para presentar el contenido del documento añadiremos la siguiente línea:

```
<?xml-stylesheet type="text/css" href="estilos.css"?>
```

Cuerpo

Consta de las etiquetas XML necesarias para estructurar el documento junto con su contenido. Recordar que existirá, por lo menos, un elemento raíz, para que el documento sea considerado como bien formado.

Validación de documentos XML

La validación de un documento se realiza mediante el uso de otro documento que nos permita especificar la estructura de un tipo de documento XML concreto para una aplicación específica. En dicho documento definimos su estructura y sintaxis, es decir, sus elementos, atributos, entidades y sus posibles combinaciones y anidamientos.
Los documentos XML bien formados que cumplen las restricciones indicadas se dice que son documentos validados o válidos.
Estudiaremos dos mecanismos para realizar dicha validación: DTD y XML Schemas.

Especificaciones relacionadas con XML

Existen multitud de especificaciones para trabajar con documentos XML, muchas de ellas están aún en una fase inicial de su desarrollo.

- Definición del tipo de documento: DTD, XML Schema.
- Enlazar entre sí partes o documentos → XML: XLink, XPointer.
- Buscar/filtrar información en el documento: XPath.
- Creación de formularios/interfaces de usuario: XForms.

• Presentación y/o transformación de documentos: CSS, XSL, XSLT.
• Lenguaje de consulta de bases de datos XML: XQuery.

Ámbitos de aplicación

Las ventajas que añade el formato XML para almacenar y transportar información hace que se utilice en multitud de ámbitos, como por ejemplo:

• Documentos ofimáticos: formato open document (ODF – OASIS Open Document Format for Office Applications), y documentos Microsoft Office (OOXML – Office Open XML).

• Protocolos estándar: WSDL (Web Services Description Language), SOAP (Simple Object Access Protocol).

• Subscripción a fuentes de información (feeds): RSS e ATOM.

• Nuevos archivos de configuración de aplicaciones UNIX/LINUX.

• Lenguajes para la web: xHTML – la versión de HTML creada en XML, SMIL (Synchronized Multimedia Integration Language) – el nuevo lenguaje (estándar abierto W3C) para presentaciones multimedia, WML (Wireless Markup Language) – lenguaje para páginas de dispositivos móviles con tecnología WAP.

• AJAX (Asynchronous JavaScript + XML).

Software para edición de XML

Usaremos el programa **XMLPad** que bajamos de la url: http://www.wmhelp.com/xmlpad3.htm

1/2/2018

Añadir PLUGIN MANAGER en Notepadd++

https://notepad-plus-plus.org/community/topic/14496/no-plugin-manager/2

if you want to add it automatically:

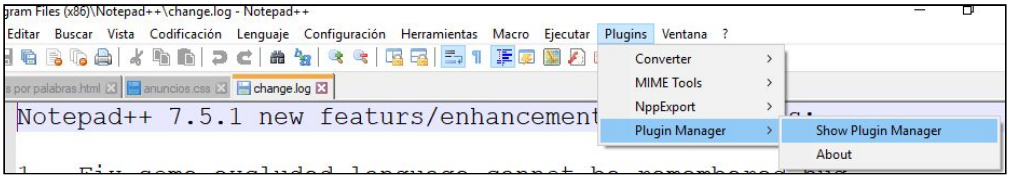
download and install version 7.3.3:

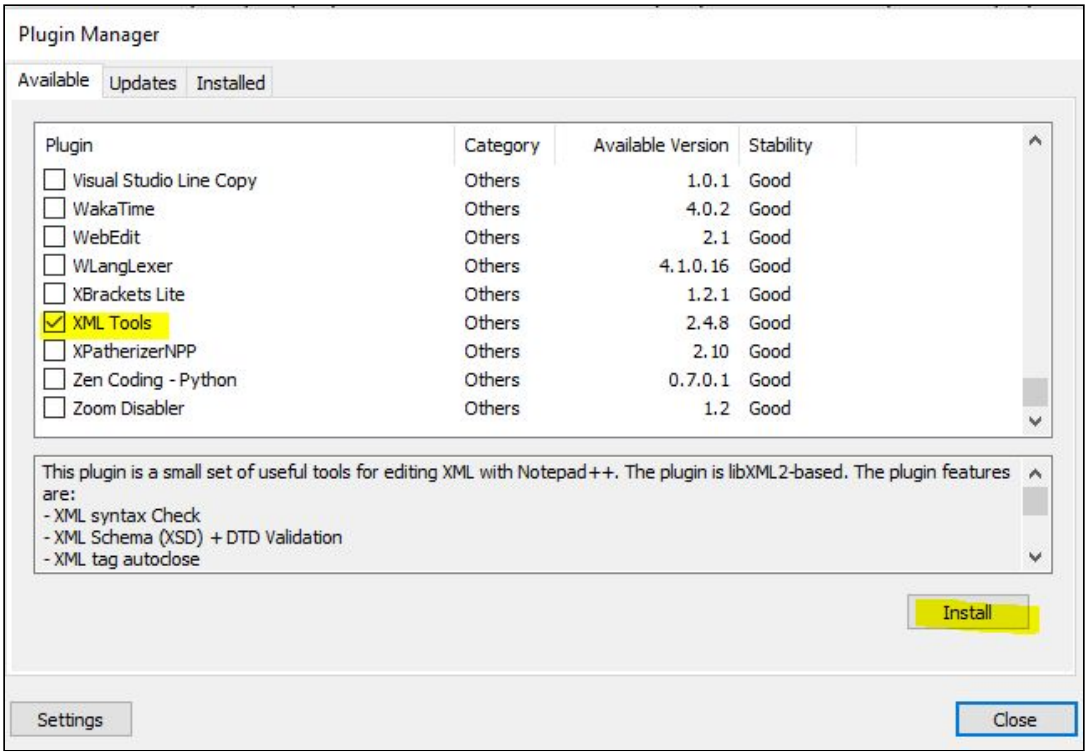
https://notepad-plus-plus.org/download/v7.3.3.html

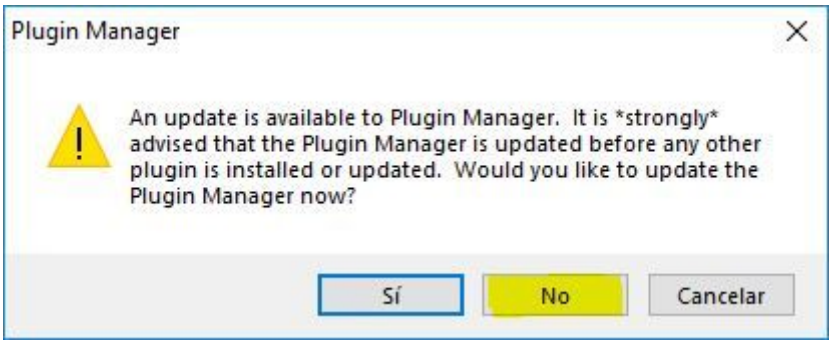
and then download and upgrade it with the latest version:

https://notepad-plus-plus.org/download/v7.5.1.html

Añadir plugin xml tools.


gram Files (x86)\Notepad++\change.log - Notepad++
Editar Buscar Vista Codificación Lenguaje Configuración Herramientas Macro Ejecutar Plugins Ventana ?
Converter
MIME Tools
NppExport
Plugin Manager
Show Plugin Manager
About
change.log
Notepad++ 7.5.1 new featurs/enhancement


Plugin Manager
Available
Updates
Installed
Plugin
Category
Available Version
Stability
Visual Studio Line Copy Others 1.0.1 Good
WakaTime Others 4.0.2 Good
WebEdit Others 2.1 Good
WLangLexer Others 4.1.0.16 Good
XBrackets Lite Others 1.2.1 Good
XML Tools Others 2.4.8 Good
XPatherizerNPP Others 2.10 Good
Zen Coding - Python Others 0.7.0.1 Good
Zoom Disabler Others 1.2 Good
This plugin is a small set of useful tools for editing XML with Notepad++. The plugin is libXML2-based. The plugin features are:
- XML syntax Check
- XML Schema (XSD) + DTD Validation
- XML tag autoclose
Install
Settings
Close


Plugin Manager
An update is available to Plugin Manager. It is *strongly* advised that the Plugin Manager is updated before any other plugin is installed or updated. Would you like to update the Plugin Manager now?
Sí
No
Cancelar