

XML

XML - Extensible Markup Language

- Por Tim Bray (Netscape), Jean Paoli (Microsoft) y C.M. Sperberg-McQueen (Universidad de Illinois, Chicago) en 1998
- **Lenguaje de marcado de propósito general** -> se puede usar con cualquier propósito porque **las etiquetas/elementos no están definidas**. Ej: Lenguajes como XHTML, MathML, SVG, RSS, RDF, entre otros están basados en XML
- Formato simple basado en texto utilizado para estructurar información: documentos, datos, configuración, libros, transacciones, *facturas* y mucho más. **Busca compartir información a través de diferentes sistemas**, un ejemplo es el internet

Ejemplo: MathML

We will now prove the Pythagorean theorem:

Statement: In a right angled triangle, the square of the hypotenuse is equal to the sum of the squares of the other two sides.

i.e, If a and b are the legs and c is the hypotenuse then $a^2 + b^2 = c^2$.

```
<article id="wikiArticle">
  <p>
    We will now prove the Pythagorean theorem:
  </p>
  <p>
    <strong>Statement</strong>: In a right angled triangle, the square of the hypotenuse is equal
to the sum of the squares of the other two sides.
  </p>
  <p>
    i.e, If a and b are the legs and c is the hypotenuse then
    <math>
      <mrow>
        <msup>
          <mi>&nbsp; a</mi><mn>2</mn>
        </msup>
        <mo> + </mo>
        <msup>
          <mi> b </mi><mn>2</mn>
        </msup>
        <mo> = </mo>
        <msup>
          <mi> c </mi><mn>2</mn>
        </msup>
      </mrow>
    </math>.
  </p>
</article>
```

Objetivos originales con

comentarios de Tom Bray

1. Debe ser usable en Internet y por una variedad amplia de aplicaciones
2. ~~Debe ser compatible con SGML~~
3. Debe ser fácil de crear documentos XML así como programas que procesen documentos XML -> *"easy" means a CS bachelor's degree ought to be able to write basic XML processing machinery in less than a week. We did meet our goal, qualitatively*
4. El numero de condiciones opcionales debe ser idealmente cero -> *los documentos XML deben ser procesados por cualquier procesador*
5. Los documentos XML deben ser legibles por humanos y razonablemente claros -> *more pleasant than binary format ;)*
6. El diseño XML debe ser rápido -> *OPEN, nonproprietary*
7. El diseño XML debe ser formal y conciso -> *usable para programadores*
8. **Omitir** elementos del marcado es de **mínima** importancia -> *no se deben omitir*

**“Whenever there is a conflict between conciseness
and clarity, clarity wins.”**

–Tim Bray

Ventajas de XML

- Redundancy: XML is *very verbose*, pretende ser lo suficientemente explícito para evitar errores al anidar elementos

<outside>

 <inside>Soy</inside> súper verboso

</outside>

- Self-describing: precisamente, auto descriptivo. Se puede imaginar el formato con solo leer el documento XML
- Network effect: puede ser leído y procesado por cualquier herramienta/parser-analizador de XML
- Reliability: Un documento XML no será procesado si contiene errores. Sí hay errores muestra el lugar y el error para que pueda ser corregido

Verbose:

¿y HTML?

- Se podría decir que *HTML is a presentation language* 🙋🙋, whereas *XML is a data-description language* 🖥️📱
- Las reglas de sintaxis son más estrictas que HTML y existen diferencias como:
 - Todos los elementos deben ir cerrados o marcados vacíos
 - Los valores de los atributos en los elementos, siempre se encierran en “” comillas
 - Se pueden usar caracteres Unicode con referencias de caracteres hexadecimales (ej: & para &) en vez de numéricas y se pueden usar entidades personalizadas provenientes de un DTD. Los únicos caracteres reservados o entidades son: <, >, &, " and ' para <, >, &, " y '.

- Para ser válido, un documento XML debe ajustarse con **reglas semánticas específicas**, usualmente se establecen en un esquema XML o un DTD. Un documento XML que contiene una etiqueta/elemento no definido, es inválido

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE body [
  <!ENTITY warning "Warning: Something bad happened... please
refresh and try again.">
]>
<body>
  &warning;
</body>
```


DTD - Document Type Definition

- Compara vocabulario y validez de la estructura de los documentos XML contra las reglas gramaticales del lenguaje
- Se especifica dentro del mismo documento o aparte

```
<?xml version="1.0"
encoding="UTF-8"?>
<!DOCTYPE note SYSTEM "Tweet.dtd">
<note>
  <to>Yehuda Katz</to>
  <from>Tom Dale</from>
  <heading>Apologize</heading>
  <body>I'd like to apologize for
earlier accusing you of breaking the
build on OS X machines. You broke
the build on all platforms! Such
dedication to cross platform
equality.</body>
</note>
```

Tweet.dtd

```
<!DOCTYPE tweet
[
  <!ELEMENT note (to,from,heading,body)>
  <!ELEMENT to (#PCDATA)>
  <!ELEMENT from (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body (#PCDATA)>
]>
```

https://www.tutorialspoint.com/xml/xml_dtds.htm

<!DOCTYPE html>

- *Doctype* asegura que el navegador haga el mejor esfuerzo por seguir las especificaciones relevantes, en vez de usar un modo diferente de render que sea incompatible con algunas especificaciones
- Netscape Navigator VS. Microsoft Internet Explorer -> W3C: standard compliant sites VS. old legacy sites -> quirks mode, almost standards mode y full standards mode
 - quirks mode: emula comportamiento de IE5 para los sitios pre-web standards, almost standards mode: algunos quirks y full standards mode: comportamiento descrito por las especificaciones de HTML y CSS

Activating Browser Modes with Doctype. <https://hsivonen.fi/doctype/>

Presentación

- Si no se especifica un estilo específico, se mostrará como raw XML en el navegador.
- Se puede especificar el estilo de presentación:

- **CSS**

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

- **XLS** - Extensible Stylesheet Language Transformations (XSLT) que puede transformar XML a otros lenguajes como HTML

```
<?xml-stylesheet type="text/xsl" href="transform.xsl"?>
```

https://developer.mozilla.org/en-US/docs/Web/XML/XML_introduction

XSD - XML Schema Definition

- Describe la validez de la estructura y contenido de un documento XML. Define elementos, atributos y tipo de datos
- Similar a un esquema de base de datos que describe la información de la base de datos

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs = "http://www.w3.org/2001/XMLSchema">
  <xs:element name = "contact">
    <xs:complexType>
      <xs:sequence>
        <xs:element name = "name" type = "xs:string" />
        <xs:element name = "department" type = "xs:string" />
        <xs:element name = "phone" type = "xs:int" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
<body>
<contact>
  <name>Grettel Barcelo</name>
  <department>EIC Computación</department>
  <phone>80446 1814</phone>
</contact>
</body>
```

<https://create.kahoot.it/k/490308b0-8b2c-41ce-b72e-56658410a3e4>

Laboratorio 4

- Descarga myFirstXML.xml y myFirstXML.css
- Ábre myFirstXML.xml en tu navegador y arregla los errores
 - Puedes utilizar <https://www.xmlvalidation.com/> u otro validador
- Debes obtener un resultado parecido a myFirstXML.jpg
- Sube el archivo corregido a Bb

<https://www.sitepoint.com/really-good-introduction-xml/>