Introduction to XML

Assoc. Prof. Dr. Kanda Runapongsa Saikaew
Dept. of Computer Engineering
Faculty of Engineering
Khon Kaen University

Topics

- What is XML?
- Why XML?
- Where does XML come from?
- Where is XML being used today?
- What is going on standards front?

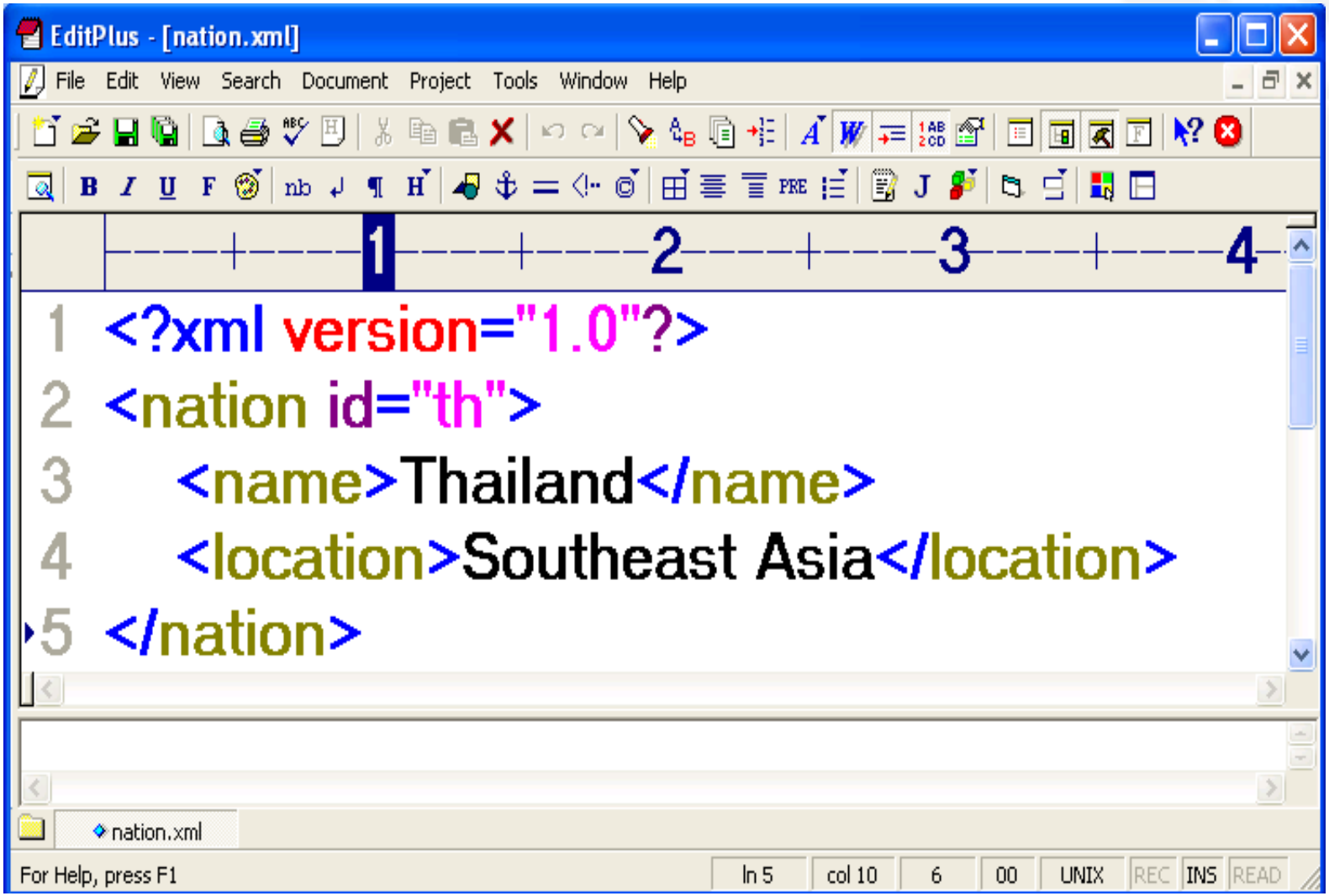
What is XML? (1/2)

```
<?xml version="1.0"?>
<nation id="th">
    <name>Thailand</name>
    <location>Southeast Asia
    </location>
</nation>
```

What is XML? (2/2)

- XML stands for Extensible Markup Language
- It becomes the standard for data interchange on the Internet
- XML is a text-based markup language
  - Encode the meaning of data by using tags which are acted as markup
  - Tags are surrounded by < and >
  - Example: <Nationality>Thai</Nationality>
- It is also a meta-markup language

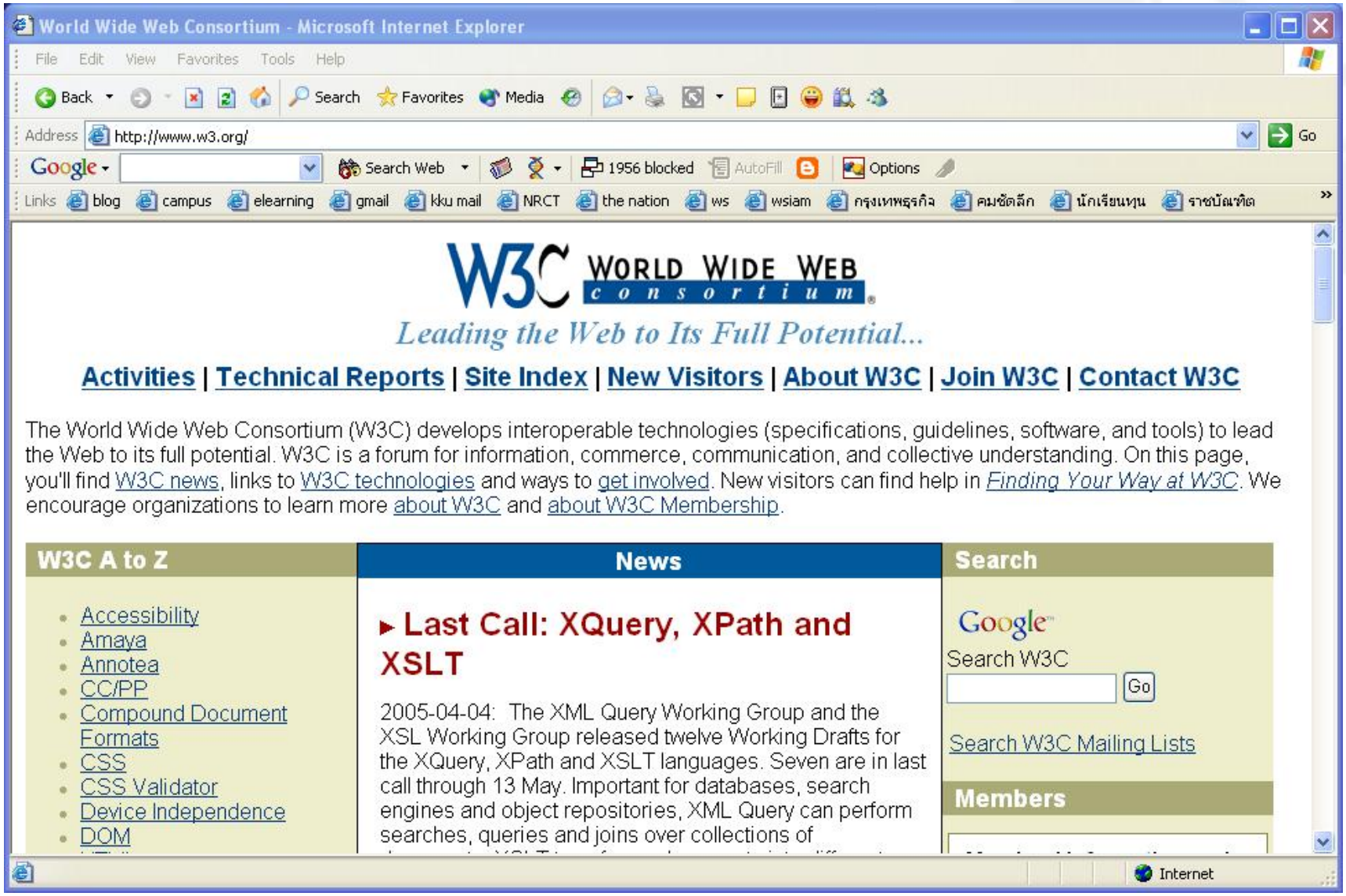
An XML Document in Text Editor



Markup Language

- Used to markup data
  - Methodology for encoding data with some information
- Examples
  - Yellow highlighter on a string of text as emphasizer
    - Example: Many people view Thai as friendly people
  - Comma between pieces of data as separator
    - Example: People need food, clothes, medicine, and house

XML: Markup Language by W3C



XML, HTML, and SGML

- XML is a markup language defined by the World Wide Web Consortium (W3C, www.w3c.org)
- Markup languages describe the way the content of the document should be interpreted
- The markup language that most people know is HTML
- Both HTML and XML are defined based on SGML (Standard Generalized Markup Language)

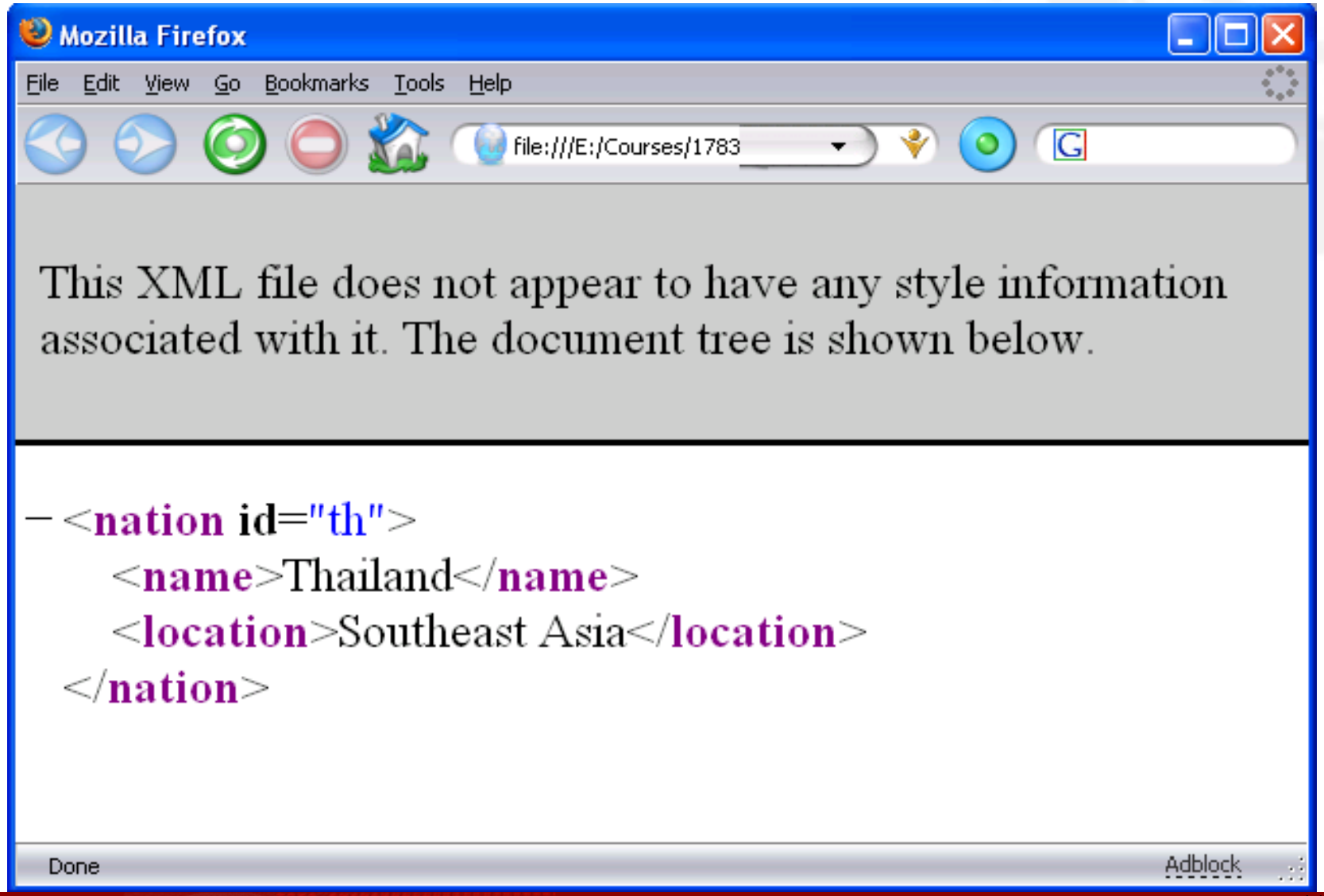
SGML

- SGML is used for documents in many fields, such as Aerospace, Semiconductor, and Publishing
- Several barriers prevented SGML over the Web
  - Complex and unstable software
  - Obstacles to interchange of SGML data
  - No widely supported style sheets

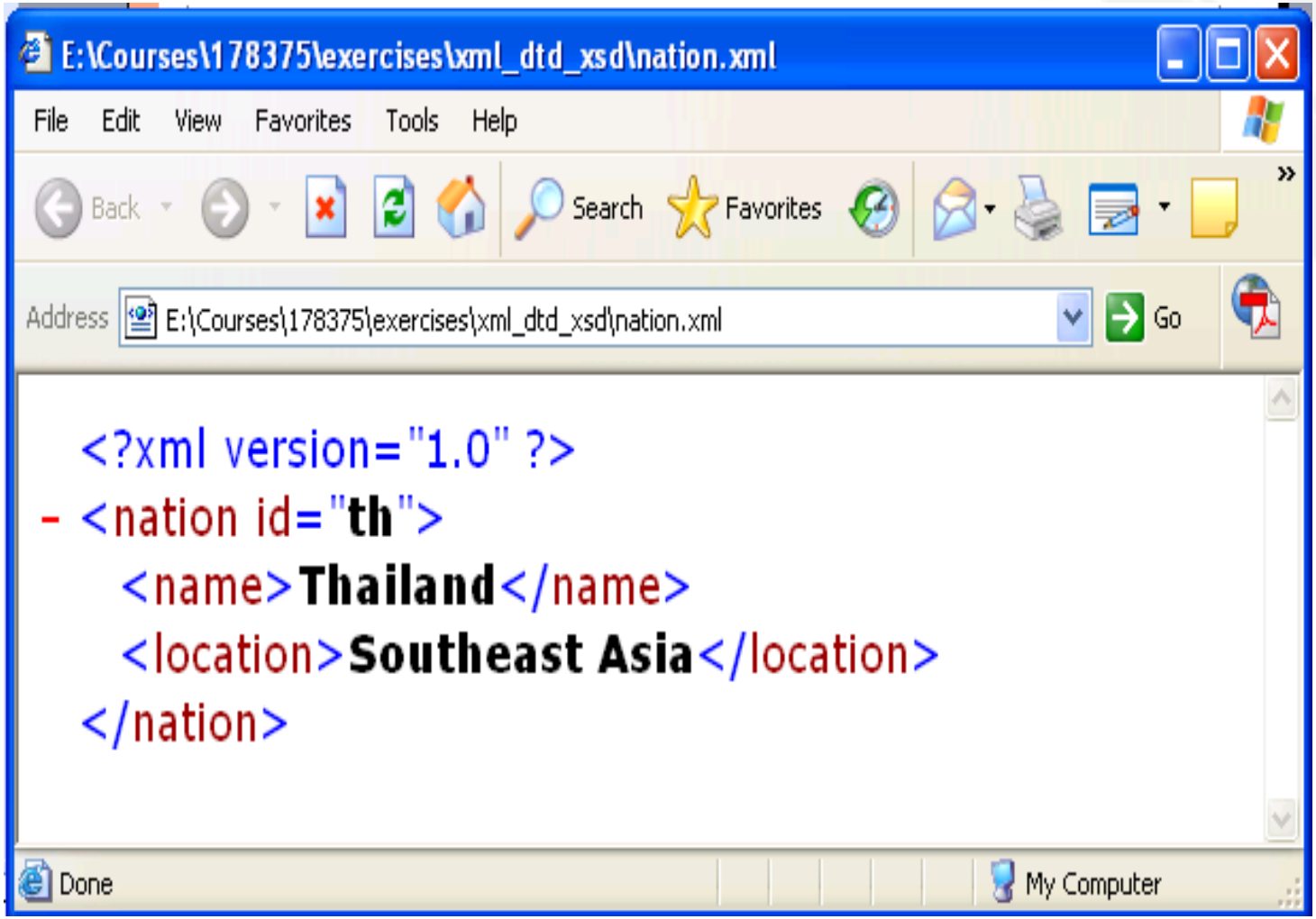
HTML

- The most popular markup language
  - In 1998, Google search 28 million pages
  - In 2005, Google search 8 billion pages
  - In 2008, Google search 1 trillion pages
  - In 2016, Google search 130 trillion pages
- Designed for presentation for data
  - Examples: <html>, <head>, <body>, <title>
- HTML documents are processed by HTML processing application (Browser)

View an XML Document with Firefox Browser



View an XML Document with Internet Explorer Browser



Strengths of HTML

- Easy to implement and author
  - Small number of tags
  - Simple relationship between tags
  - Syntax-checking is very forgiving
  - Limited number of formats possible
  - Viewers can be small and simple
- HTML trades power for ease of use

Weaknesses of HTML (1/2)

- Fixed set of tags
  - Not user extensible
    - Dependency to “markup language” definition process
  - Dependency to vendors
    - Vendor proprietary tags
    - Implementation not in sync
      - Netscape browser vs. Internet Explorer browser
- Predefined semantics for each tag
- Predefined data structure

Weaknesses of HTML (2/2)

- No formal validation
- Does not support semantic search
- Based on solely on appearance (rendering) NOT on content
- Formatting too simple
  - Limited control
- Cannot process complex documents
- Have no document structure to enable automation

What We Cannot Do with HTML

- We cannot create our own tags that are meaningful for each application
- We cannot have the way to specify a set of data that everyone agrees upon
- We cannot change shared data easily with minimal effort

The Purpose of XML

- Easy for information to be reused, interchanged, and automated
- Deliver information on the Web
- Let users design their own markup language
- Could drive arbitrarily complex distributed processes

Key Features of XML

- Extensibility
- Media and Presentation independence
  - Separation of contents from presentation
- Structure
- Validation

Extensibility (1/2)

- XML is Meta-markup language
- You define your own markup languages (tags) for your own problem domain
- Infinite number of tags can be defined
  - Need for domain-specific standards
  - XSLT

Extensibility (2/2)

- Tags can be more than formatting
  - Semantics data representation
  - Business rules
    - ebXML
  - Data relationship
    - EJB 2.0 Container Managed Persistence
  - Formatting
    - XSL
  - Anything you want

Media (Presentation) Independence (1/2)

- Clear separation between contents and presentation
- Contents of data
  - What the data is
  - Is represented by XML document
- Presentation of data
  - What the data looks like
  - Can be specified by stylesheet

Media (Presentation) Independence (2/2)

- Stylesheet
  - Instruction of how to present XML data
  - CSS
    - Tailored for HTML browser
  - XSL
    - XML based
    - General purpose
    - Work with XSLT

Separation of Contents from Presentation

- Searching and retrieving data is easy and efficient
  - Tags give search'able information
- Many applications use the same data in different ways
  - Employee data can be used by
    - Payroll application and Facilities application
- Enables portability of data
  - Portable over time and space

XSLT Transformation

- Example (XML -> HTML)

XML:

<email>joe@nbc.com</email>

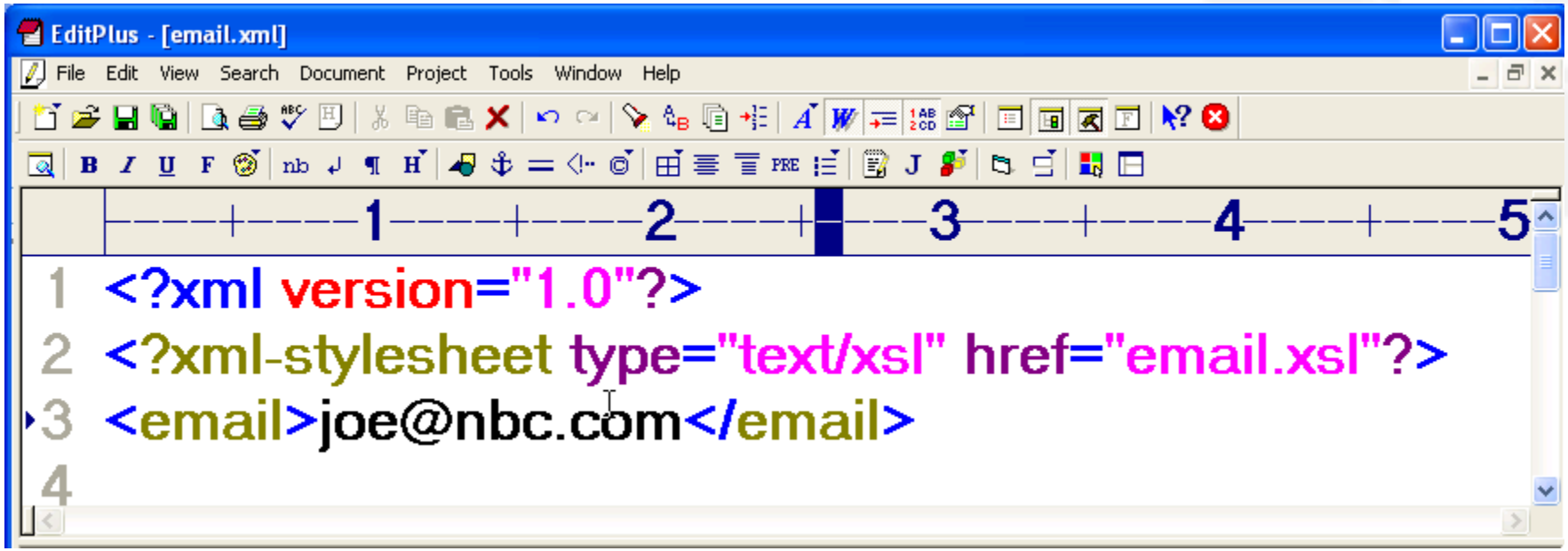
XSLT stylesheet can say:

- Start a new line
- Convert “email” XML tag to “To:” HTML tag
- Display “To:” in bold, followed by a space
- Display your email address

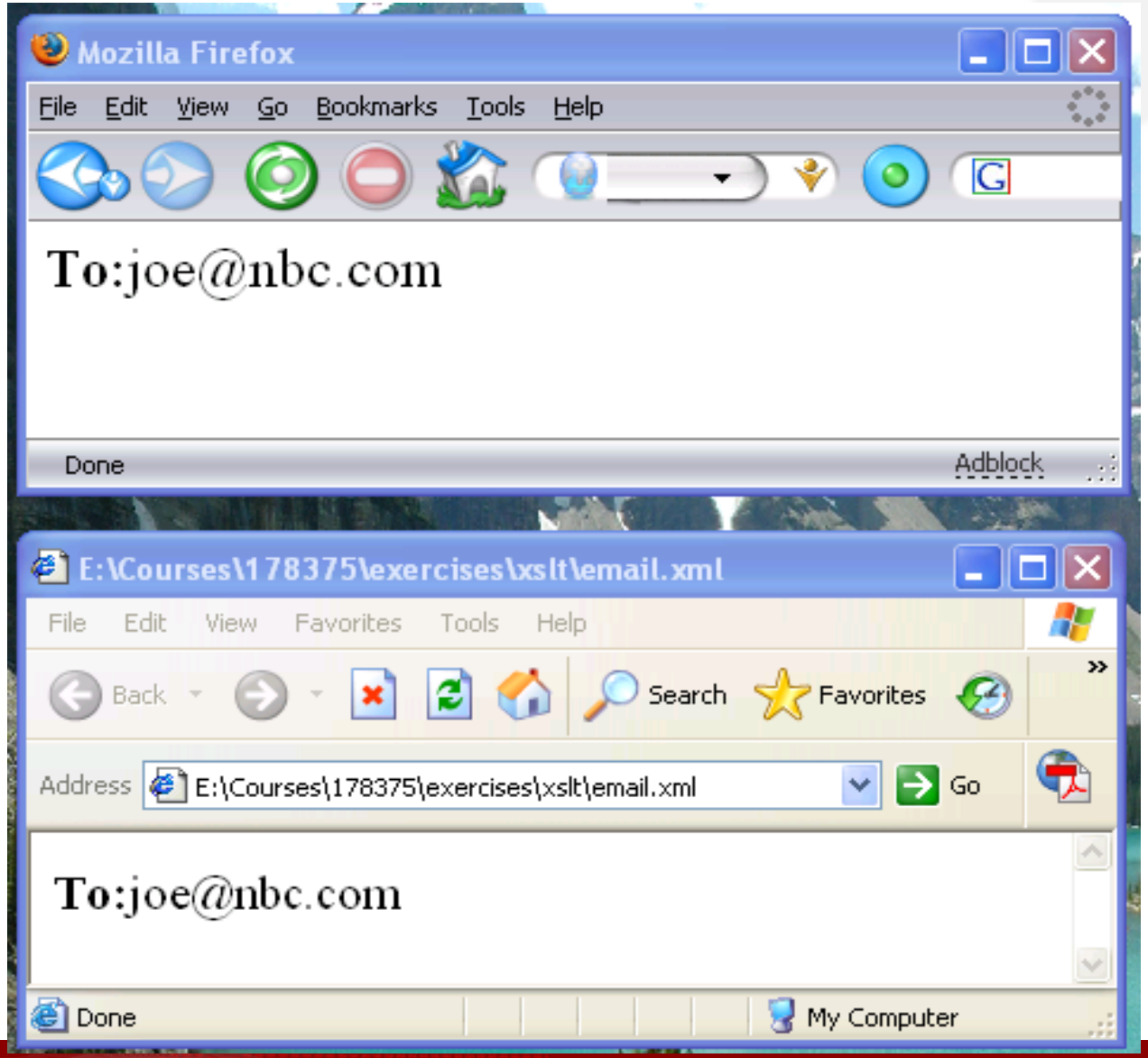
Which produces

To:joe@nbc.com

Input XML File



Output HTML File in Browsers



Input XSL File

EditPlus - [email.xsl]

File Edit View Search Document Project Tools Window Help

```
1  <?xml version="1.0"?>
2  <xsl:stylesheet version="1.0"
3  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
4  <xsl:output method="html"/>
5  <xsl:template match="/">
6  <html>
7      <body>
8          <b>To:</b>
9          <xsl:value-of select="email"/>
10     </body>
11 </html>
12 </xsl:template>
13 </xsl:stylesheet>
```

Structure: HTML vs. XML

- HTML (Automatic Presentation of Data)

   John Doe 1234 // Display in bold

- XML (Automatic Interpretation of Data)

```
<employee>
    <name>John Doe</name>
    <employeeID>1234</employeeID>
</employee>
```

XML Structure

- Relationship
  - Employee is made of Name and EmployeeID
- Hierarchical (Tree-form)
  - Faster to access
  - Easier to rearrange
  - Can be any number of depth
- Enables to build large and complex data
- Portability of relationship and hierarchical structure

Desirable Features of XML

- Semantics of data
- Plain Text
- Easily Processed
- Inline usability
- Internationalized
- License-free

Semantics of Data

- Meaning of data
- XML tags "indirectly" specifies the semantical meaning
  - Does <name> means "firstname lastname" or "lastname firstname"?
- Potential for divergence
  - Industry collaboration to agree upon the semantical meanings of tags
  - Need for transformation (XSLT)

Plain Text

- Can use any text-editing tool
- Easier for humans to read and edit
  - Configuration information
  - Information description
  - Short notices
- Any operating system supports reading and writing text

Easily Processed

- Set of Well-formed rules
- Validity checking
- Ready-to-use tools
  - Parsers and validators
  - Transformers
  - Browsers
  - IDE

Inline Usability

- Can integrate data from multiple resources
  - Can be displayed or processed as a single document
- Modularization without using Linking
- Example
  - A book made of independently written chapters
  - Same Copyright text in many books

Internationalized

- XML is Unicode-based
  - You can mix languages
- Both markup and content
- XML tools must support both UTF-8 and UTF-16 encodings
- Critical for world-wide adoption of XML as universal data representation

Where Does XML Get Used?

- Simple and complex data representation
- Integration of heterogeneous applications
- Portable data representation
- Displaying and publishing

Data Representation

- XML encodes the data for a program to process
- Readable by humans
- Be able to be processed by computers
- Complex relationship can be represented
- Internationalized
- Many 3rd-party tools
  - Editing, Syntax checking

Data Representation Examples

- Configuration files
  - EJB deployment descriptor
- “make” files (Apache ANT project)
- File format for electronic office documents
  - OASIS OpenDocument format (ODF)
  - Microsoft Office Open XML (OOXML)

Integration of Heterogeneous Applications

- Typically used with Messaging system
- XML message is minimum contract for communication
  - Loosely-coupled communication
- Enables easy EAI (Enterprise Application Integration)
  - Payroll, Finance, Products
- E-commerce
  - Supplier, distributor, manufacturer, retail

Portable Data Representation

- Non-proprietary
  - Application independent
  - Object-model independent
  - Language independent
  - Platform independent
  - Communication protocol independent
  - Communication media independent
- Used for means of “information exchange”

Portable Data Representation Examples

- Purchase order, Invoice
- Business transactional semantics
- Patient record
- Mathematical formula
- Musical notation
- Manufacturing process

Displaying and Publishing

- Common data for different presentations
- Separation of contents from presentation
- Examples
  - Web information presented to different client types
  - Information rendered to different medium

Developer Activities on XML (1/2)

- Creating XML document
  - Mostly by text-editor or WISWIG tools
  - Programmatically
- Sending and Receiving XML document
  - Over any kind of transports
    - HTTP, SMTP, FTP, …
  - Through programming APIs
    - Socket APIs

Developer Activities on XML (2/2)

- Parsing XML document
  - Convert XML document into programming objects
- Manipulating programming objects
  - Application specific way
  - Examples
    - Display
    - Save them in database
    - Create new XML document

XML Standards

- XML Specification
  - XML, Namespaces
- Validation
  - W3C XML Schema
- Parser
  - DOM, SAX , StAX
- Style and Query
  - XSL, XSLT, XPath
- Security
  - XML Digital Signature, XML Encryption

XML Applications

- Web Services
  - XML data is exchanged between service provider & service requester
  - RSS, ATOM
- AJAX
  - Asynchronous JavaScript and XML
  - AJAX allows Web developers to create interactive Web pages without having to wait for pages to load

XML Applications

- Web Services
  - XML data is exchanged between service provider & service requester
  - RSS, ATOM
- AJAX
  - Asynchronous JavaScript and XML
  - AJAX allows Web developers to create interactive Web pages without having to wait for pages to load

XML in Modern Software

- Android
  - Declare UI elements in XML
- WPF
  - WPF employs XAML, a derivative of XML, to define and link various UI elements
- Firefox Extension
  - Use XUL (XML User Interface Language, pronounced zool) to define GUIs

References (1/2)

- XML standards portal http://www.w3.org/xml
- XML resources
  - http://www.xml.com
  - http://www.oasis-open.org
  - http://www.xml.org
- XML Tutorials
  - http://www-106.ibm.com/developerworks/views/xml/tutorials.jsp
  - http://www.zvon.org
- Sang Shin XML Course Page http://www.javapassion.com/xml/

References (2/2)

- Wikipedia, “OpenDocument”, http://en.wikipedia.org/wiki/OpenDocument
- Wikipedia, “Office Open XML”, http://en.wikipedia.org/wiki/Office_Open_XML
- Devx.com”, StaX: DOM Ease with SAX Efficiency”, http://www.devx.com/Java/Article/30298