

The Extensible Markup Language (XML) defines a way of marking up text to describe the structure of data. XML allows you to create your own markup language: you define the tags that give meaning to your data. The World-Wide Web Consortium (W3C) creates and maintains the definition of XML, making it a standard for creating markup languages. Industries and organizations use XML to write rules defining their own markup languages.

In this three-day course students will learn how to create well-formed XML documents. In addition, they will learn about the most important supplementary technologies that support XML, including XML Schema for validation as well as XSLT for transformation.

Course Objectives:

- Explain what XML is, and how it is used in application and document development.
- Write well-formed documents that conform to XML's basic rules of syntax.
- Validate XML documents with XML Schemas.
- Use XML Namespaces to distinguish between XML tags.
- Transform an XML document into an HTML document using XSLT.
- Use XPath to navigate a document tree.
- Explain how programs can use DOM and SAX to parse XML documents.

Audience: Application developers, web developers and administrators, and XML authors.

Prerequisites: HTML. Familiarity with web and data processing concepts. Programming experience is helpful, but not necessary.

Number of Days: 3 days

<p>1 Course Introduction Course Objectives Overview Suggested References</p>	<p>2 Getting Started with XML Data and Document Structure XML Well-Formed XML Valid vs. Well-Formed XML XML Schema Presentation Style XSL-FO and XSLT Using XML</p>	<p>3 Writing Well-Formed XML XML Fundamentals Tag Attributes Naming Rules</p>	<p>4 Namespaces Why Namespaces? Namespace Prefixes and Declaration Multiple Namespace Declarations Declaring Namespaces in the Root Element Default Namespaces</p>	<p>5 Validating XML with XML Schemas Schema Overview Associating XML with a Schema Simple and Built-in Types</p>
---	---	--	---	---

	Complex Types		
	Element Declarations		
	Attribute Declarations		
	Choices		
	Named Type and Anonymous Types		
6	Intro to XSLT	10	XML in Applications
	Stylesheet, Source, and Result		Reasons and Places for Using XML
	XSLT Processors		DOM Parsers
	Processor Implementations		SAX Parsers
	XPath Basics		Web Services
	xsl:stylesheet	11	Appendix A - Using XML Schema
	xsl:template		with Namespaces
	xsl:value-of		Qualified and Unqualified XML
	xsl:apply-templates		Associating Qualified XML with a Schema
	xsl:output		Associating a Schema with a Namespace
7	XPath Nodetypes		Controlling Element and Attribute Qualification
	XPath Expressions		Merging Schema with the Same Namespace
	XPath Context		Merging Schema with Different Namespaces
	XPath Location Steps	12	Appendix B - Validating XML with DTDs
	Element and Root Nodes		XML DTDs
	Text and Attribute Nodes		DOCTYPE
	Comment and Processing Instruction Nodes		Element Conditions and Quantifiers
	Namespace Nodes		Attributes
	Wildcards		Attribute Types
	Whitespace		REQUIRED, IMPLIED, and FIXED
	Default Template Rules		Parsed General Entities
8	XPath Axes and Predicates		Parsed Parameterized Entities
	Location Paths and Location Steps		DTDs and Namespaces
	Peer Axis Types		
	More Peer Axis Types		
	Descendant Axis Types		
	Ancestor Axis Types		
	Node Tests		
	Predicates		
	Functions		
9	XSLT Flow Control		
	xsl:if		
	xsl:choose		
	xsl:for-each		
	xsl:sort		
	Named Templates		
	Mode		