

# Technical Training Solutions



## INTRODUCTION TO XML, XSLT & XPATH (3 Day Course)

Technical  
Course  
Outline

### Course Summary

The goal of this course is to provide students with an understanding of what Extensible Markup Language (XML) is, why it is used and how to integrate it with existing web and pc based applications for richer, better performing applications.

Students will learn the rules for creating “well-formed” XML documents that describe data. From there, students will learn how to create a set of validation rules for XML documents through the use of a “DTD” (Document Type Definition) as well as W3C Schemas. Students will then learn to take XML documents and perform powerful operations on them using the Extensible Stylesheet Language Transformations (XSLT) and XPath languages.

### Intended Audience:

This course is intended for **application developers** who wish to leverage the power that XML, XSLT, and XPath offer and incorporate them into their applications.

### Prerequisites:

To ensure an effective learning environment **for all attendees**, it is imperative that **each student** meet minimum course prerequisites.

- Basic word processing and Windows skills (i.e. switching between open windows, file copying, renaming, & moving)
- Successful completion of, or 90% proficiency with, the topics from the following courses  
(Please refer to each course outline to make sure you meet these requirements):
  - “Introduction to HTML, XHTML, & CSS”
  - “Introduction to JavaScript”

### Course Contents:

#### Overview Of XML

- What is XML & Why Use It?
- XML Parsers & W3C Standards
- Terminology
  - Markup, Tags, Attributes, Elements, Text
  - PCData vs. CData
  - Root Element / Document Element

#### Introduction To Writing XML Documents

- The XML Declaration
- XML Prologue & Processing Instructions
- Rules For Writing “Well-Formed” XML
  - Case-Sensitivity
  - Comments & Whitespace
  - Tag/Attribute Naming Standards & Best Practices
  - Tag Closure & Self-Termination
  - Attribute Values
  - Element Nesting
  - Illegal Characters: Escape Codes & CData Sections
- Attribute vs. Element Based XML

#### XML Validation

- Why Validate?
- Document Type Definitions vs. Schemas
- Document Type Definitions (DTD's)
  - SGML **<!ELEMENT>** and **<!ATTLIST>** Declarations
    - Specifying An Element's Content Model
      - Specifying Sequences and/or Choices
    - Cardinality
    - Specifying Attribute Value Choices or Default Values
    - Validating Against A DTD
- The W3C XML Schema Standard
  - Creating Schema Definitions
  - Element & Attribute Declarations
  - Data Typing
  - Simple vs. Complex Types
    - Performing Schema Validations
      - Linking XML To An .xsd
      - The XMLSchemaCache

#### XML Namespaces

- The Need To Group & Organize XML Data
- Using Prefixes To Identify An Element's Group
  - The Problem With Prefixes Alone
- Using W3C Namespace Naming Standards
- Associating Elements & Attributes With A Namespace
- Setting Default Namespaces

#### Enhancing Data Presentation Via Style Sheets

- CSS (Cascading Style Sheets)
  - Use & Limitations
- XSL & XSLT
  - Thinking In Terms Of “Nodes”
    - Node Types
    - The “Source Tree” & The “Result Tree” Of Nodes
    - Determining The Context Node
    - The “Document Node” / “Root Node”
  - The W3C XSL Namespace
  - Defining & Applying XSL Template Rules
  - Returning Source Tree Data
  - XSL Looping, XSL Decision Making, & XSL Sorting
  - Built-In XSL Functions
- XPath
  - XPath Expressions & Location Paths (The 13 Different XPath Axes)
    - XPath Predicates
    - Abbreviated Syntax
  - Built-In XPath Functions

#### The W3C DOM (Document Object Model)

- What Is The XML DOM?
- The XML Document Tree
- DOM Objects, Properties & Methods
- Client-Side DOM Programming Via JavaScript
  - Loading & Parsing XML Data With MSXML