Hands-On

Mastering XML

Course Description

This hands on XML programming class presents a thorough introduction to creating, validating, transforming, transporting and formatting XML data.

The course covers structuring data with XML validating that data with document type definitions (DTDs) and schemas creating and viewing XML documents transforming XML documents with the XML Stylesheet Language (XSL, XSLT and XPATH) use of XML in the deployment of Web Services referencing XML data via the document object model (DOM) and parsing libraries and parsing XML via the Simple API for XML (SAX). The course will also focus on creating, validating and transforming data in a Java or .NET application development environment.

Extensive examples and demos are provided, as well as comprehensive hands on lab exercises that reinforce the concepts being taught and introduce the practical application of XML to business problems.

Students Will Learn

- **XML Syntax**
- **DTDs and Schemas**
- **Namespaces and Scoping**
- **XPATH and XSLT**
- **XSL Format Objects**
- **SOAP and Web Services**
- **DOM and SAX Parsing**

Prerequisites

Experience with a programming or scripting language, such as Java, JavaScript, VB or C is helpful but not necessary.

Course Outline

The Need for XML

© Copyright 2024 - BTS, Inc. 'All Rights Reserved'

- Structured Data and Formatting
- The Need for Structured Data
- The Need for Richer Formatting Control
- Advantages of XML
- XML as an International Standard
- SGML, XML, and HTML
- World Wide Web Consortium (W3C) Specifications and Grammars
- XML Applications and Tools
- Creating and Viewing XML Documents
- Transforming XML Documents

XML Document Syntax

- The Concept of Well-Formedness
- Well-Formed Structure
- Elements and Tags
- The XML Declaration
- Editing and Viewing XML Documents
- The Document Type Declaration
- Start and End Tags
- Empty Tags
- Element Nesting
- Element Attributes
- Comments
- Special Characters and Built-In Entities
- CDATA Sections
- Embedded XML
- External XML
- XML Parsers
- Displaying Data in Tables Using DHTML

Validating XML Documents with DTDs

- The Concept of Data Validation
- Writing Document Type Definition (DTD) Files
- Internal and External DTDs
- Validating Parsers
- The !DOCUMENT Tag
- Specifying Valid Elements and Subelements
- Specifying Valid Attributes
- Specifying Valid Entities
- Specifying Parameter Entities
- The CDATA Type
- The ID, IDREF, and IDREFS Types
- The ENTITY and ENTITIES Types
- The NMTOKEN and NMTOKENS Types
- The NOTATION Type
- Enumeration's
- Conditional Sections
- Creating Internal DTDs
- Linking External DTDs to XML Documents
- Validation Tools

XML Namespaces

- The Need for Namespaces

© Copyright 2024 - BTS, Inc. 'All Rights Reserved'

- Specifying a Namespace
- URLs, URIs, and URNs
- Qualifying Names
- Namespace Scoping
- The HTML Namespace
- Additional Significant Namespaces

Validating XML Documents with Schemas

- Schema Design Goals (Limitations of DTDs)
- Mixing DTDs and Schemas
- Resolving Name Conflicts with Namespaces
- Schema Composition
- Linking Schemas to XML Documents
- Annotation Declarations
- Element Declarations
- Attribute Declarations
- W3C Schema Data Types
- Specifying Simple Types
- Regular Expressions
- Working with User-Defined Data Types
- Union and List Types
- Specifying Complex Types
- Deriving Complex Types Using Inheritance
- Reusable Groups
- Substitution Groups
- Identity Elements
- Validating Uniqueness
- Validating Required Fields
- Combining and Redefining Schemas

Transforming XML Documents with XSLT and XPath

- The Limitations of CSS
- XSL Stylesheet Advantages
- Transformation vs. Formatting
- XSLT and XSL-FO
- XSLT Templates
- XPath Data Model
- Declaring XSL Stylesheets
- Built-In Templates
- Using Templates as Subroutines - xsl:apply-templates
- XPath Expression Syntax
- XPath Functions and Predicates
- Inserting Elements - xsl:element
- Inserting Attributes - xsl:attribute
- Extracting Node Values - xsl:value-of
- Looping - xsl:for-each
- Sorting - The order-by Attribute
- Simple Conditionals - xsl:if
- Multiple Conditionals - xsl:choose, xsl:when, and xsl:otherwise
- Copying Nodes - xsl:copy
- Modular Design Using Named Templates
- Rendering HTML Using XSLT and CSS
- XML-to-XML Transformations

Introduction to Simple Object Access Protocol (SOAP)

© Copyright 2024 - BTS, Inc. 'All Rights Reserved'

- Purpose of SOAP
- SOAP's Use of XML and Schemas
- Elements of a SOAP Message
- Sending and Receiving SOAP Messages (SOAP Clients and Receivers)
- SOAP Namespaces and Schemas
- Handling SOAP Faults
- Current SOAP Implementations

Introduction to Web Services

- Architecture and Advantages of Web Services
- SOAP Messages with Attachments
- Purpose of Web Services Description Language (WSDL)
- WSDL Elements
- Creating and Examining WSDL Files
- Overview of Universal Description, Discovery, and Integration (UDDI)
- UDDI Registries (Public and Private)
- Core UDDI Elements
- UDDI Classification Schemes and Programming Interfaces
- Deploying and Consuming Web Services
- ebXML Specifications
- ebXML Registry and Repository

Formatting XML Documents with XSL-FO

- XML Documents with XSL-FO
- Purpose of XSL Formatting Objects (XSL-FO)
- XSL-FO Documents and XSL-FO Processors
- XSL-FO Namespace
- Page Format Specifiers
- Page Content Specifiers
- Presentation Conversion

Introduction to the XML Document Object Model (XMLDOM)

- Purpose of a Document Object Model (DOM)
- Core, HTML, and XML DOM
- Data Object Tree
- XMLDOM Parsers
- The Top-Level Document Object
- Primary Nodes and Node Collections (NodeList and NamedNodeMap)

Programming the XML Document Object Model with Java and .NET

- XMLDOM Data Access Methods
- Core DOM Objects
- Microsoft XMLDOM
- Java API for XML Processing (JAXP)
- Sources of Java XML Parsers
- Installing Parsers
- The Parser Object
- The Document Object
- Loading an XML File
- Traversing the Document Tree
- Using the Performance Monitor to Record and Analyze Performance Data
- Validating and Transforming an XML Document

© Copyright 2024 - BTS, Inc. 'All Rights Reserved'

- Creating an XML Document with DOM

Using the SAX Parser in Java and .NET Applications

- Purpose of the Simple API for XML (SAX)
- Event-Driven Programming Model
- Types of SAX Parsers
- Obtaining a SAX Parser at Runtime
- Importing Interfaces
- The ContentHandler Interface
- The DocumentHandler Interface
- Declaring Handlers
- Loading and Processing an XML File
- Validating XML Data with the SAX Parser

Equipment Requirements

(This apply's to our hands-on courses only)

BTS always provides equipment to have a very successful Hands-On course. BTS also encourages all attendees to bring their own equipment to the course. This will provide attendees the opportunity to incorporate their own gear into the labs and gain valuable training using their specific equipment.

Course Length

5 Days

© Copyright 2024 - BTS, Inc. 'All Rights Reserved'