

XSLT Programming

CDT754

This course builds the skills needed by developers who will be using XSLT to build production applications for document publishing or data interchange. It covers beginning, intermediate and some advanced XSLT topics. The course is taught using XSLT 2.0 and it addresses the differences between XSLT 2.0 and XSLT 1.0.

Audience

- Anyone interested in using XSLT to build production applications for document publishing or data interchange
- This course is a prerequisite for CDT758, Formatting XML Documents with XSL-FO.

Prerequisites

- CDT753, Introduction to XML

Course Length

- Three Days

Teaching Methods

- Lecture
- Hands-On Exercises

Learning Objectives

- Create a basic HTML page from a simple XML document
- Use templates to create an HTML page from a document-oriented XML source
- Create a stylesheet that uses conditional XSLT to process an XML document with variable content
- Use functions to manipulate strings, numbers, dates and times
- Create a stylesheet that uses variables and parameters
- Create multiple output documents from a single XSLT transformation
- Use XSLT sorting and grouping capabilities to organize the output from a transformation
- Use XML IDs to generate links in generated HTML documents
- Create utility templates using named templates and functions
- Build XSLT applications using XSLT stylesheets that reside in multiple files
- Understand how to write stylesheets that are compatible with both XSLT 1.0 and XSLT 2.0
- Use XSLT to implement dynamic client-side transformation
- Understand how XSLT is used to implement dynamic server-side transformations

Course Outline

TG5

Review of XML <ul style="list-style-type: none">▪ Markup Languages▪ XML Rules▪ Creating Markup Languages▪ Presenting XML with CSS Creating HTML from XML <ul style="list-style-type: none">▪ XSL: The Extensible Stylesheet Language▪ Using XSLT Processors▪ Simplified Stylesheets▪ The XSLT Namespace▪ Generating HTML Pages Templates <ul style="list-style-type: none">▪ XSLT Stylesheet Structure▪ The Node Tree▪ XSLT Processing Model▪ Templates as Mapping Rules▪ Using Templates with Modes Conditional Processing <ul style="list-style-type: none">▪ Processing Optional Elements▪ Using the Ancestry of Source XML▪ Using the Location of Result XML▪ Conditional Elements in XSLT▪ Conditional Expressions in XPATH▪ Testing Elements and Attributes▪ Filtering XML	Manipulating Atomic Values <ul style="list-style-type: none">▪ Atomic Values▪ Manipulating Strings▪ Manipulating Numbers▪ Manipulating Dates, Times and Durations▪ Manipulating Qualified Names▪ Manipulating URIs Variables and Parameters <ul style="list-style-type: none">▪ Defining Variables▪ Using Parameters Paths and Sequences <ul style="list-style-type: none">▪ Node Trees Revisited▪ Matching Nodes▪ Selecting Nodes▪ Sequences Result Trees <ul style="list-style-type: none">▪ Generating Nodes▪ Creating Result Documents▪ Controlling Output Sorting and Grouping <ul style="list-style-type: none">▪ Sorting in Different Orders▪ Sorting Non-alphabetically▪ Multiple Sorts▪ Flexible Sorting▪ Grouping by Position▪ Grouping in Sequence▪ Multilevel Grouping	IDs, Keys and Numbering <ul style="list-style-type: none">▪ Searching▪ Generating IDs▪ Numbering Named Templates, Stylesheet Functions and Recursion <ul style="list-style-type: none">▪ Named Templates▪ Stylesheet Functions▪ Recursion Building XSLT Applications <ul style="list-style-type: none">▪ Splitting Up Stylesheets▪ Reusing Stylesheets▪ Accessing Data▪ Retrieving Referenced Information Backwards Compatibility and Extensions <ul style="list-style-type: none">▪ Backwards Compatibility▪ Same-Version Compatibility▪ Extensions to XSLT and XPATH Dynamic XSLT <ul style="list-style-type: none">▪ Server-Side Transformations▪ Client-Side Transformations▪ Client-Side or Server-Side?
---	--	---