THE COMPLETE XSLT COURSE

School of progrmming
Eugene Ibiso

GOALS OF THIS COURSE

01

INSIGHT & UNDERSTANDING OF XSLT

after going through this you will have a deep insight and a better understanding of what xslt is all about

02

SOLID TRANSFORMATION WRITING SKILLS

Improvement in your writting and debugging skill

03

PROBLEM SOLVING:

BASIC EVERYDAY XML PROCESSING

ADVANCED, NON TRADITIONAL SKILL

WELCOME MESSAGE

Presentations are communication tools that can be used as demonstrations, lectures, speeches, reports, and more.

BRIEF WORKTHROGH : MODULE 1

XSLT PROCCESSING MODEL

PURPOSE OF THIS MODULE

Describe the xslt processing model:

Trees

- Inputs
- Outputs
- Temporary

XSLT Stylesheet Modules
External context
The XSLT Processor
What kind of language is XSLT
Basic XSLT Concepts

- Template Rules
- Built-in templates
- Sequence constructors
- Variables and expressions
- Parameters
- Context

BRIEF WORKTHROGH : MODULE 2

STYLESHEET STRUCTURE AND MODULARIZATION

PURPOSE OF THIS MODULE

Describe the structure of an XSLT Stylesheet:

XSLT global level elements:

- Declarations

XSLT non-global level elements:

- Instructions

Conditional compilation

- The use-when attribute
- Writing transformation for both XSLT 1.0 and XSLT 2.0

Modularization

- Including stylesheet modules
- Importing stylesheets modules
- Import precedence

BRIEF WORKTHROGH : MODULE 3

CREATING NODES AND SEQUENCES. REPETITION, APPLYING TEMPLATES, MODES

PURPOSE OF THIS MODULE

Describe the XSLT copying/creating/referencing instructions:

- Copying nodes and sequences
- Creating nodes and sequences
- Attribute-value templates
- Creating atomic values and references to nodes

Discuss repetitive processing on a sequence of items:

- The <xsl :for-each> instruction

Describe how to apply templates on a sequence of nodes:

- The <xsl :for-templates> instruction

Show how to apply additional templates matching a node:

- The <xsl :apply-templates> instruction
- The <xsl :next-match> instruction

BRIEF WORKTHROGH : MODULE 4

CALLABLE UNITS

PURPOSE OF THIS MODULE

Describe the XSLT callable units

- Named templates
- Functions

Define parameters

- How to define
- Typing
- How to pass actual argument values
- Defaulting and required value
- Tunneling

Explain the invocation of a callable unit

Define functions overloading

BRIEF WORKTHROGH : MODULE 5

STRING PROCESSING

PURPOSE OF THIS MODULE

Describe the string processing common to XSLT 1.0/XSLT 2.0
XPATH 1.0 string processing functions

- Explain the string-processing functions in xpath 2.0

Discus the string/sequence relationship
Explain some xpath 2.0 functions on sequences and their use for strings.
Describe the xpath 2.0 functions that use RegEx
Explain the XSLT 2.0 RegEx capabilities

BRIEF WORKTHROGH : MODULE 6

KEYS, SORTING, GROUPING, NUMBERING, AND MULTIPLE DOCUMENTS PROCESSING

PURPOSE OF THIS MODULE

Describe the multiple document processing of XSLT
Discus the ways to perform sort.
Explain indexing of nodes using “keys”
Discuss the ways to arrange a sequence of items into groups
Describe the ways to allocate numbers to a sequence of nodes ad to format these numbers

BRIEF WORKTHROGH : MODULE 7

XSLT DESIGN PATTERNS

PURPOSE OF THIS MODULE

Describe the “Overriding the identity rule” design pattern.
Explain “using recursion” in a declarative language
Discus the “Multi-pass processing” design pattern.
Describe the “Conditional instructions in XSLT” and when/how to avoid them.
Describe “using lookup tables”.
Explain the “Fill-in-the blanks” design pattern

BRIEF WORKTHROGH : MODULE 8

HIGHER ORDER FUNCTIONS AND FUNCTIONAL PROGRAMMING IN XSLT

PURPOSE OF THIS MODULE

Explain “Imperative ”, “Declarative” and “Functional” programming.
Describe “Higher-Order Functions” and their simulation in XSLT 1.0 and XSLT 2.0
Discus the “Function on lists” and some basic FP design patterns.
Explain “Dynamic generation of functions”
Describe the “FXSL Library” for functional programming in XSLT 2.0 and XSLT 1.0

BRIEF WORKTHROGH : MODULE 9

SOLVING NON-TRADITIONAL PROBLEMS

PURPOSE OF THIS MODULE

Dispel the myth of “impossible with XSLT problem”

Show examples of creating XML from plain text:

- Create and work with natural language directions:
  - Spelling checking
  - Anagrams

Explain how to solve graph problems:

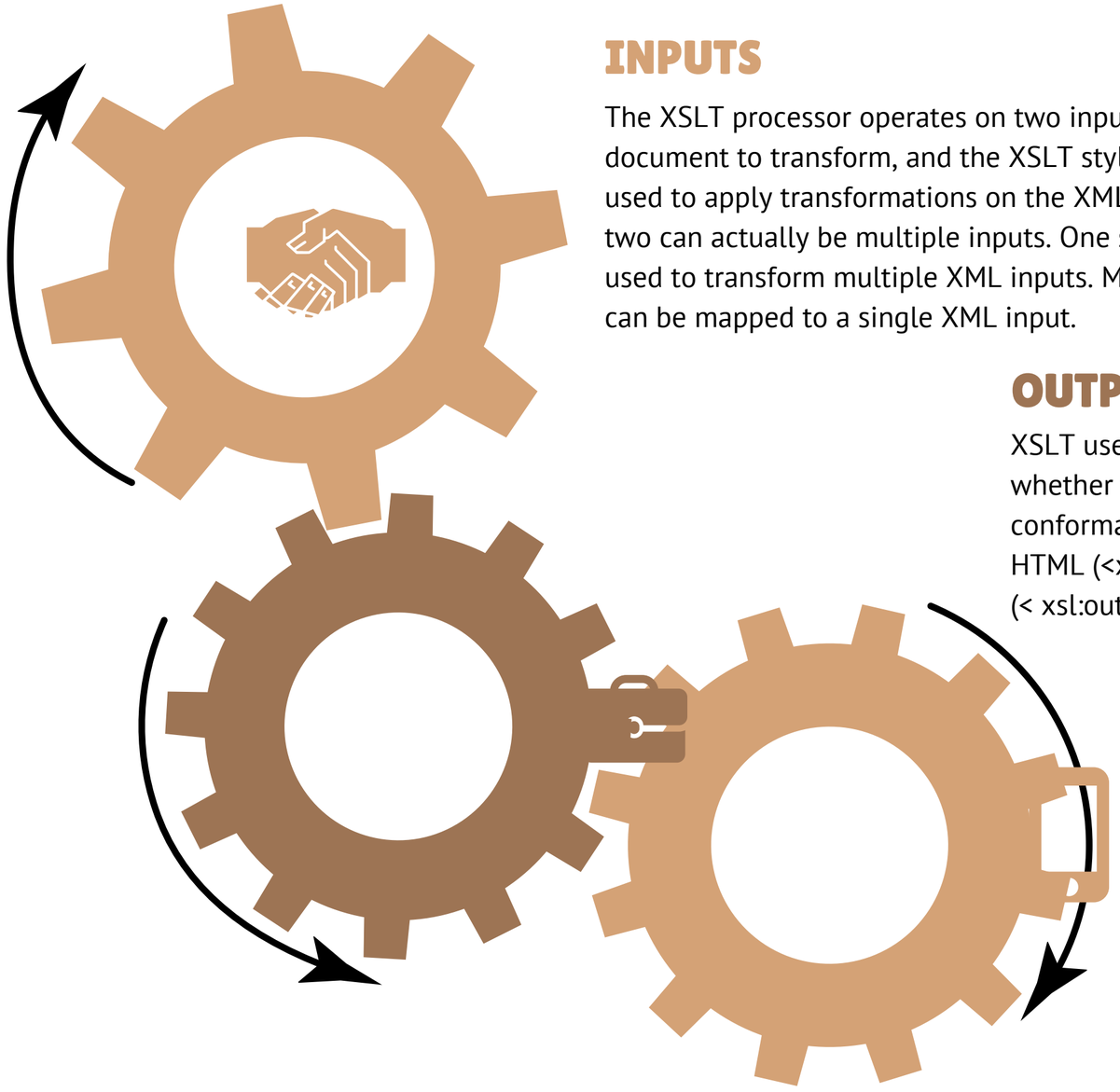
- Topological sorting

Show an example of solving a Number Theory problem:

- Working with prime numbers

XSLT PROCESSING MODELS

TREES



INPUTS

The XSLT processor operates on two inputs: the XML document to transform, and the XSLT stylesheet that is used to apply transformations on the XML. Each of these two can actually be multiple inputs. One stylesheet can be used to transform multiple XML inputs. Multiple stylesheets can be mapped to a single XML input.

OUTPUT

XSLT uses the <xsl:output> element to determine whether the output produced by the transformation is conformant XML (<xsl:output method="xml"/>), valid HTML (<xsl:output method="html"/>), or unverified text (< xsl:output method="text"/>)

TEMPORARY

XSLT 2.0 eliminates result tree fragments and replaces them with a more powerful feature: temporary trees. Once you create a temporary tree in an xsl:variable, xsl:param, or xsl:with-param element, you can do anything with it that you can do with a source tree.