

# XPath - XSL

Angelo Mario Del Grosso

CNR-ILC

<http://ilc.cnr.it/>

[angelo.delgrosso@ilc.cnr.it](mailto:angelo.delgrosso@ilc.cnr.it)

**Selezione, Elaborazione e Presentazione di documenti XML-TEI mediante  
i linguaggi XPath e XSL**

Istituto di Linguistica Computazionale “A. Zampolli”,  
14th May 2025

# Outline

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

## 1 Lesson path

## 2 Introduction

## 3 XPath

## 4 XSL Transformations

## 5 XSL in action

## 6 References

# Progress status

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

1 Lesson path

2 Introduction

3 XPath

4 XSL Transformations

5 XSL in action

6 References

# Lesson path

XPath - XSL

A.M. Del  
Grosso

Lesson path

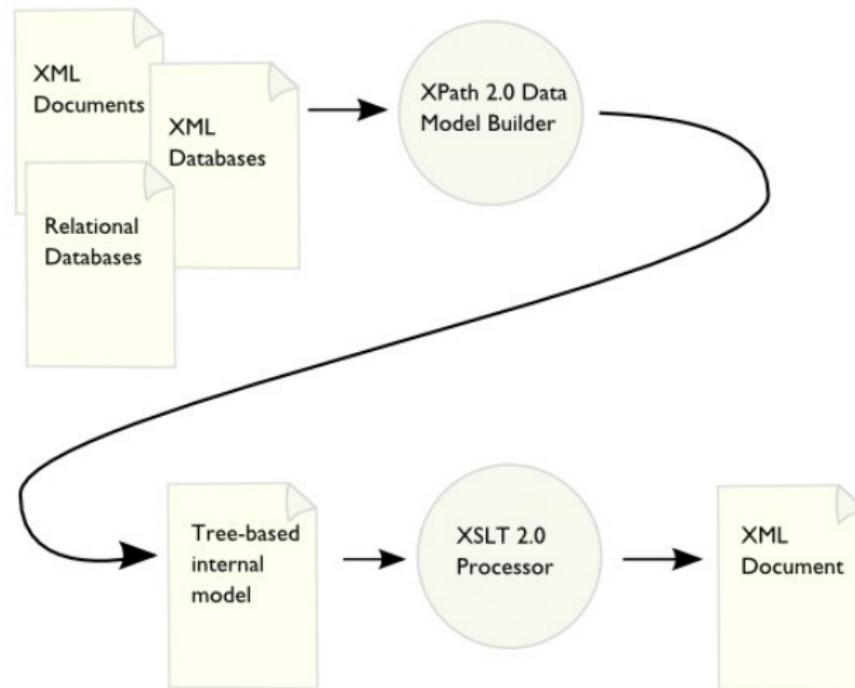
Introduction

XPath

XSL Trans-  
formations

XSL in action

References



# Progress status

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

1 Lesson path

2 Introduction

3 XPath

4 XSL Transformations

5 XSL in action

6 References

# XSL XQuery

## A family of languages

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

**XSL is a family of W3C recommendations for defining XML document transformation and presentation.**

## XSL

- **XSL Transformations (XSLT):**  
*a language for transforming XML*
- **The XML Path Language (XPath):**  
*an expression language to refer to parts of an XML document;*
- **XSL Formatting Objects (XSL-FO):**  
*an XML vocabulary for specifying formatting semantics.*

## XQuery

XQuery is a query language for XML to extract data.

# XSL Transformations

## Versioning

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

W3C > Standards > All Standards and Drafts

Skip

### XSLT COVER PAGE

2021-03-30	Recommendation	<b>XSL Transformations (XSLT) Version 2.0 (Second Edition)</b> This specification defines the syntax and semantics of XSLT 2.0, a language for transforming XML documents into other XML documents. XSLT 2.0 is a revised version of the XSLT 1.0 Recommendation [XSLT 1.0] published on 16 November 1999. XSLT 2.0 is designed to be used in conjunction with XPath 2.0, which is defined in [XPath 2.0]. XSLT shares the same data model as XPath 2.0, which is defined in [Data Model], and it uses the library of functions and operators defined in [Functions and Operators]. XSLT 2.0 also includes optional facilities to serialize the results of a transformation, by means of an interface to the serialization component described in [XSLT and XQuery Serialization]. <i>This document contains hyperlinks to specific sections or definitions within other documents in this family of specifications. These links are indicated visually by a superscript identifying the target specification: for example XP for XPath, DM for the XDM data model, FO for Functions and Operators.</i>
2017-06-08	Recommendation	<b>XSL Transformations (XSLT) Version 3.0</b> This specification defines the syntax and semantics of XSLT 3.0, a language for transforming XML documents into other XML documents.
1999-11-16	Recommendation	<b>XSL Transformations (XSLT) Version 1.0</b> This specification defines the syntax and semantics of XSLT, which is a language for transforming XML documents into other XML documents. XSLT is designed for use as part of XSL, which is a stylesheet language for XML. In addition to XSLT, XSL includes an XML vocabulary for specifying formatting. XSL specifies the styling of an XML document by using XSLT to describe how the document is transformed into another XML document that uses the formatting vocabulary. XSLT is also designed to be used independently of XSL. However, XSLT is not intended as a completely general-purpose XML transformation language. Rather it is designed primarily for the kinds of transformations that are needed when XSLT is used as part of XSL.

### RELATED RETIRED SPECIFICATIONS

2001-08-24	Retired	<b>XSL Transformations (XSLT) Version 1.1</b>
------------	---------	---

# XSL Transformations

## Versioning

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

W3C > Standards > All Standards and Drafts

> Skip <

### XPATH COVER PAGE

2017-03-21	Recommendation	<b>XML Path Language (XPath) 3.1</b> XPath is an expression language that allows the processing of values conforming to the data model defined in the XQuery and Xpath Data Model.
2014-04-08	Recommendation	<b>XML Path Language (XPath) 3.0</b> XPath 3.0 (renamed from XPath 2.1 to align with the family of "3.0" specifications) is an expression language that allows the processing of values conforming to the data model defined in [XQuery and XPath Data Model (XDM) 3.0]. Some of the important new features since XPath 2.0 are: Literal function items, inline functions, dynamic function invocations, and function item coercion Clarification of rules associated with sequence type matching let expressions EQNames (QNames with a namespace URI instead of a namespace prefix) Support for union types in casting and function arguments
2010-12-14	Recommendation	<b>XML Path Language (XPath) 2.0 (Second Edition)</b> XPath is a way to refer to parts of an XML document. XPath 2.0 is based on the XQuery 1.0 and XPath 2.0 Data Model (XDM), and also introduces Schema awareness and data typing.
1999-11-16	Recommendation	<b>XML Path Language (XPath) Version 1.0</b> XPath is a language for addressing parts of an XML document, designed to be used by both XSLT and XPointer.

# XSL Transformations

## elements

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

80 elements ca

xsl:accept	xsl:copy	xsl:include	xsl:non-matching-substring	xsl:sort
xsl:accumulator	xsl:copy-of	xsl:iterate	xsl:number	xsl:source-document
xsl:accumulator-rule	xsl:decimal-format	xsl:key	xsl:on-completion	xsl:stream
xsl:analyze-string	xsl:document	xsl:map	xsl:on-empty	xsl:strip-space
xsl:apply-imports	xsl:element	xsl:map-entry	xsl:on-non-empty	xsl:stylesheet
xsl:apply-templates	xsl:evaluate	xsl:matching-substring	xsl:otherwise	xsl:template
xsl:assert	xsl:expose	xsl:merge	xsl:output	xsl:text
xsl:attribute	xsl:fallback	xsl:merge-action	xsl:output-character	xsl:transform
xsl:attribute-set	xsl:for-each	xsl:merge-key	xsl:override	xsl:try
xsl:break	xsl:for-each-group	xsl:merge-source	xsl:package	xsl:use-package
xsl:call-template	xsl:fork	xsl:message	xsl:param	xsl:value-of
xsl:catch	xsl:function	xsl:mode	xsl:perform-sort	xsl:variable
xsl:character-map	xsl:global-context-item	xsl:namespace	xsl:preserve-space	xsl:when
xsl:choose	xsl;if	xsl:namespace-alias	xsl:processing-instruction	xsl:where-populated
xsl:comment	xsl:import	xsl:next-iteration	xsl:result-document	xsl:with-param
xsl:context-item	xsl:import-schema	xsl:next-match	xsl:sequence	

# XSL Transformations

## Documentation

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

## xsl:template

Defines a processing rule for source elements or other nodes of a particular type.

*Available in XSLT 1.0 and later versions. Available in all Saxon editions.*

- **Category:** declaration
- **Content:** (`xsl:context-item?`, `xsl:param*`, \*sequence-constructor\*)
- **Permitted parent elements:** `xsl:package`; `xsl:stylesheet`; `xsl:transform`; `xsl:override`

## Attributes

`match?`

`pattern`

Pattern to identify the type of node to be processed. The most common form of pattern is simply an element name. However, more complex patterns may also be used: the syntax of patterns is given in more detail in XSLT Pattern Syntax. The following examples show some of the possibilities:

Pattern

Meaning



# XPath functions

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

Number of functions: 177 of 177

[abs](#)  
[accumulator-after](#)  
[accumulator-before](#)  
[adjust-date-to-timezone](#)  
[adjustdateTime-to-timezone](#)  
[adjust-time-to-timezone](#)  
[analyze-string](#)  
[apply](#)  
[available-environment-variables](#)  
[available-system-properties](#)  
[avg](#)  
[base-uri](#)  
[boolean](#)  
[ceiling](#)  
[codepoint-equal](#)  
[codepoints-to-string](#)  
[collation-key](#)  
[collection](#)  
[compare](#)  
[concat](#)  
[contains](#)  
[contains-token](#)  
[copy-of](#)  
[count](#)  
[current](#)  
[current-date](#)  
[current-datetime](#)  
[current-group](#)  
[current-grouping-key](#)

[fold-left](#)  
[fold-right](#)  
[for-each](#)  
[for-each-pair](#)  
[format-date](#)  
[format-datetime](#)  
[format-integer](#)  
[format-number](#)  
[format-time](#)  
[function-arity](#)  
[function-available](#)  
[function-lookup](#)  
[function-name](#)  
[generate-id](#)  
[has-children](#)  
[head](#)  
[hours-from-datetime](#)  
[hours-from-duration](#)  
[hours-from-time](#)  
[id](#)  
[idref](#)  
[implicit-timezone](#)  
[in-scope-prefixes](#)  
[index-of](#)  
[innermost](#)  
[insert-before](#)  
[iri-to-uri](#)  
[json-doc](#)  
[json-to-xml](#)

[parse-xml](#)  
[parse-xml-fragment](#)  
[path](#)  
[position](#)  
[prefix-from-QName](#)  
[put](#)  
[QName](#)  
[random-number-generator](#)  
[regex-group](#)  
[remove](#)  
[replace](#)  
[resolve-QName](#)  
[resolve-uri](#)  
[reverse](#)  
[root](#)  
[round](#)  
[round-half-to-even](#)  
[seconds-from-datetime](#)  
[seconds-from-duration](#)  
[seconds-from-time](#)  
[serialize](#)  
[snapshot](#)  
[sort](#)  
[starts-with](#)  
[static-base-uri](#)  
[stream-available](#)  
[string](#)  
[string-join](#)  
[string-length](#)

# XPath

## Documentation

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

### fn:substring

`substring($sourceString as xs:string?, $start as xs:double) → xs:string`

Returns a substring of a given string starting at the given starting position and continuing to the end of the string.

#### Arguments

\$sourceString	xs:string?	The input string
\$start	xs:double	The position of the first character of the input string to be included in the result

#### Result

xs:string

`substring($sourceString as xs:string?, $start as xs:double, $length as xs:double) → xs:string`

Returns a substring of a given string starting at the given starting position and continuing to the end of the string, or \$length characters if shorter.

#### Arguments

\$sourceString	xs:string?	The input string
\$start	xs:double	The position of the first character of the input string to be included in the result
\$length	xs:double	The number of characters to be included in the result

#### Result

xs:string

### Namespace

<http://www.w3.org/2005/xpath-functions>

### Links to W3C specifications

[XPath 2.0 Functions and Operators](#)

[XPath 3.0 Functions and Operators](#)

[XPath 3.1 Functions and Operators](#)

# W3C XML standards

## XML specs

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

### XML family specifications

[https:](https://www.w3.org/TR/?tag=xml&status=REC&version=latest)

[//www.w3.org/TR/?tag=xml&status=REC&version=latest](https://www.w3.org/TR/?tag=xml&status=REC&version=latest)

# Tree data model (XDM)

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References



- The W3C specifications for XSLT, XQuery, and XPath **model an XML document as a tree**. This data model is known as **XDM**, and the nodes of an XDM tree are known as **XDM nodes**.
- XDM defines the information contained in the input to an XSLT processor as well as it defines all **permissible values of expressions** in the XSLT
- The **node-sets of XPath 1.0** are replaced in XPath 2.0 by **sequences of nodes**.

# Tree data model (XDM)

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References



- The W3C specifications for XSLT, XQuery, and XPath **model an XML document as a tree**. This data model is known as **XDM**, and the nodes of an XDM tree are known as **XDM nodes**.
- XDM defines the information contained in the input to an XSLT processor as well as it defines all **permissible values of expressions** in the XSLT
- The **node-sets of XPath 1.0** are replaced in XPath 2.0 by **sequences of nodes**.

# Tree data model (XDM)

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References



- The W3C specifications for XSLT, XQuery, and XPath **model an XML document as a tree**. This data model is known as **XDM**, and the nodes of an XDM tree are known as **XDM nodes**.
- XDM defines the information contained in the input to an XSLT processor as well as it defines all **permissible values of expressions** in the XSLT
- The **node-sets of XPath 1.0** are replaced in XPath 2.0 by **sequences of nodes**.

# Tree data model (XDM)

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References



- The W3C specifications for XSLT, XQuery, and XPath **model an XML document as a tree**. This data model is known as **XDM**, and the nodes of an XDM tree are known as **XDM nodes**.
- XDM defines the information contained in the input to an XSLT processor as well as it defines all **permissible values of expressions** in the XSLT
- The **node-sets of XPath 1.0** are replaced in XPath 2.0 by **sequences of nodes**.

# XML Trees

## Hierarchical Ordered Nodes

XPath - XSL

A.M. Del  
Grosso

Lesson path

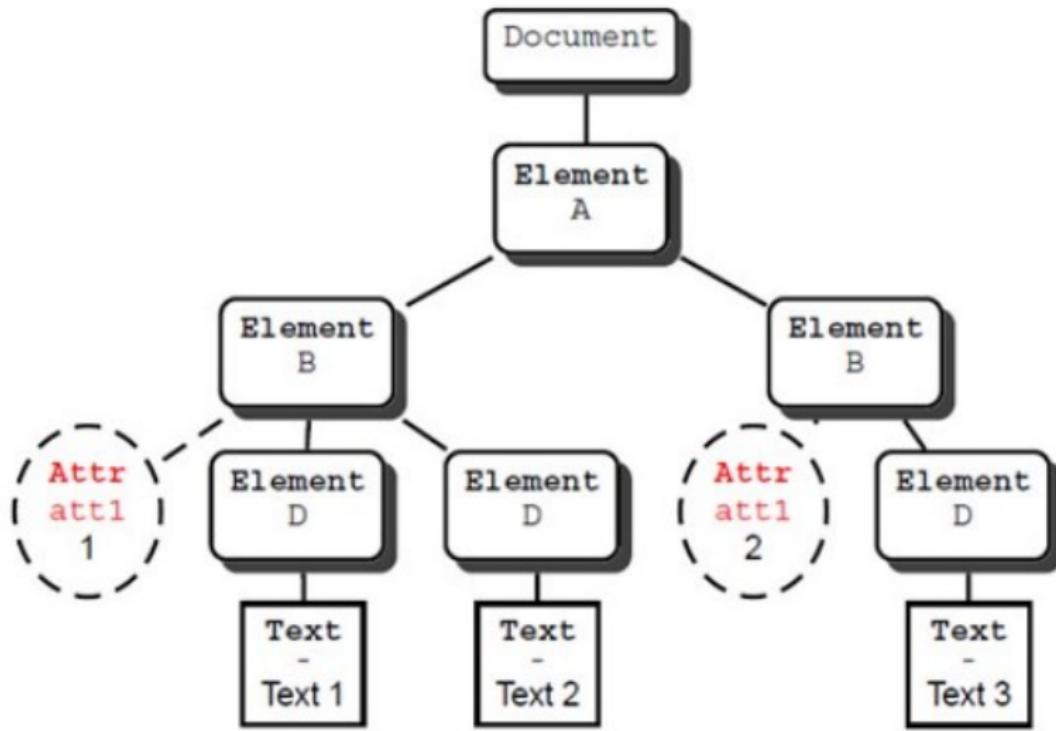
Introduction

XPath

XSL Trans-  
formations

XSL in action

References



# XPath Data Model

## XDM

XPath - XSL

A.M. Del  
Grosso

Lesson path

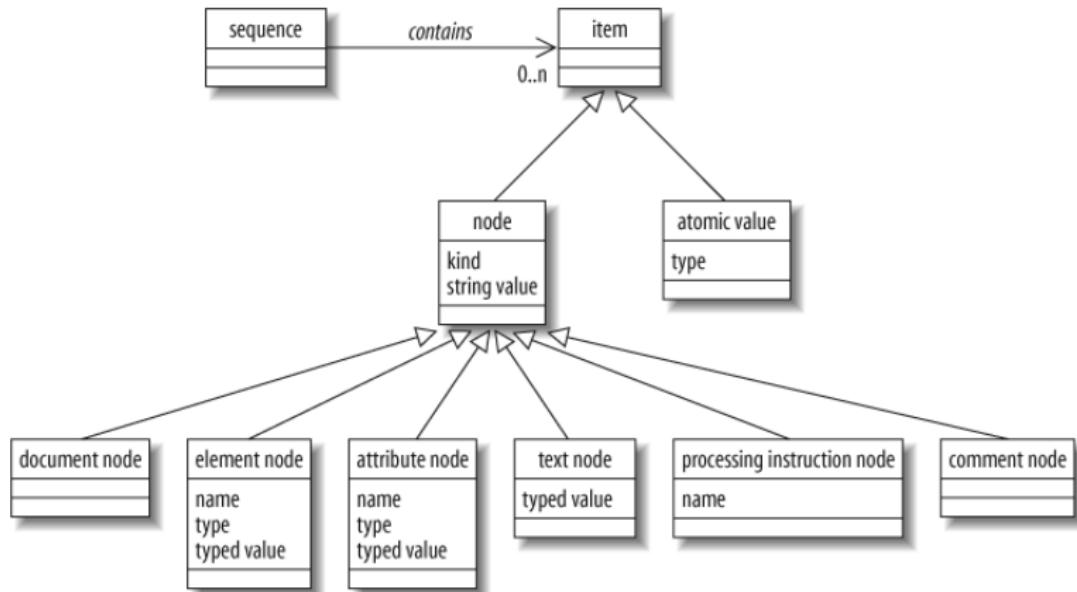
Introduction

XPath

XSL Trans-  
formations

XSL in action

References



# Fondamenti Extensible Stylesheet Language

XPath - XSL

A.M. Del  
Grosso

Lesson path

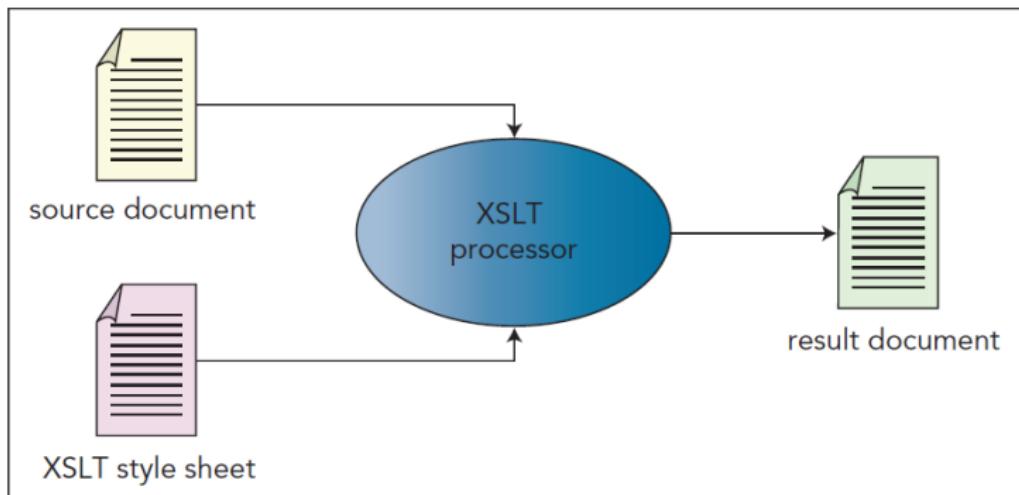
Introduction

XPath

XSL Transformations

XSL in action

References



# XML Document

## Exemple from The Inscriptions of Roman Tripolitania

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

```
<?xml version="1.0" encoding="UTF-8"?>
<TEI xml:lang="en">
  <teiHeader> ...
  </teiHeader>
  <text>
    <body>
      <div type="bibliography"> ...
      </div>
      <div subtype="text-constituted-from" type="history"> ...
      </div>
      <div type="edition" xml:lang="la"><head xml:lang="en">Text</head><ab> ...
      </ab></div>
      <div type="apparatus"> ...
      </div>
      <div type="translation"><head>Translation</head> ...
      </div>
      <div type="commentary"> ...
      </div>
      <div type="figure"> ...
      </div>
    </body>
  </text>
</TEI>
```

# eXtensible Style Sheet

## Example

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="1.0">
3  --- xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4  --- <xsl:output method="text" />
5  --- <xsl:template match="/">
6  -----<xsl:apply-templates select="current()/descendant::text" />
7  -----</xsl:template>
8  -----<xsl:template match="div[@type='edition']|div[@type='translation']">
9  -----<xsl:value-of select="normalize-space(.)" />
10 -----</xsl:template>
11 -----<xsl:template match="div" />
12 </xsl:stylesheet>
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL

wsl + □ ☰ ...

```
angelodel80@LAPTOP-V8V3MLGO:/mnt/c/Users/angel/risorse/universita/corsoCodifica/tools$ java -jar SaxonHE10-3J/
saxon-he-10.3.jar -s:source/IRT030.xml -xsl:source/built-in.xsl
```

Text Imperator Caesar Marci Antonini Pii fili diui Pii nep diui Hadriani pronep diui Traiani Parthici  
abnep diui Neruae adnep Luci Aeli Aureli Commod August

Translation Emperor Caesar case unknown son of Marcus Antonius Pius, grandson of deified Pius, great  
grandson of deified Hadrian, great great grandson of deified Trajan victor in Parthia, great great great grand  
son of deified Nerva, Lucius Aelius Aurelius Commodus Augustus



### How Does it work

The XSLT processor **uses the instructions** in the Style Sheet to process the input XML document by **traversing the document's hierarchy**.

### How Does it work

XSLT instructions indicate what **portion of the tree should be traversed**, how it should be inspected, and what **output fragment should be generated** at each point.

# Ordered Hierarchy of Content Objects

## OHCO

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

### ■ OHCO

The nodes of the tree are ordered. The child nodes of a parent node, which are siblings of one another, **occur in a particular order.**

*This is why XML can be described as representing an ordered hierarchy of content objects.*

# Progress status

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

1 Lesson path

2 Introduction

3 XPath

4 XSL Transformations

5 XSL in action

6 References

# Selecting and Processing XML Document Trees

## Basic Concepts

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

## XPath

Path **expressions** will return **node sequences** whose nodes are in document order

# Selecting and Processing XML Document Trees

## Basic Concepts

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

### Sequence

The group of nodes that an XPath expression returns is a sequence, which is a technical term for an **ordered collection of items** that permits duplicates

# Selecting and Processing XML Document Trees

## XPath expression language

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

## XPath

- Select nodes sequence from XML tree
- Process data via functions

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

## XPath

XPath expressions are extremely accurate (selection of elements, attributes, texts etc.)

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="2.0">
3      xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
4      <xsl:output method="text"/>
5
6      <xsl:template match="/">
7          ...
8              <xsl:value-of
9                  select="/TEI/text/body/div/@type" />
10             ...
11         </xsl:template>
12     </xsl:stylesheet>
13
14
15
```

out-text.txt U ×

1 bibliography history edition apparatus translation commentary figure

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

## XPath Basics

Path expressions are used to navigate from a current location  
(called the context node) to other nodes in the tree.

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

## XPath Basics

Steps in a path expression are indicated with slash characters  
and the context node changes with each step.

## XPath Basics

- "div" (*div child*)
- "div/head" (*child of div*)
- "div/\*/persName" (*child of child*)
- "div//persName" (*descendant*)

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

## XPath

The XPath expression can be absolute or relative to the **context node**

## XPath

The XPath expression encompasses three components: (**Axes, Test, Predicate**)

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

## XPath

XPath expressions navigate the XML tree by using the so called (*expression axes*).

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

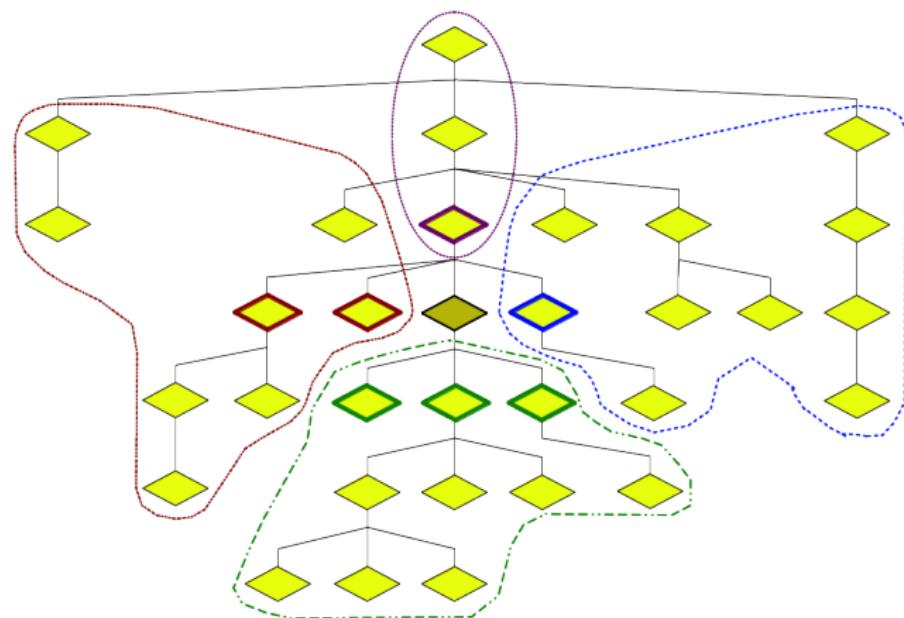
Introduction

XPath

XSL Trans-  
formations

XSL in action

References



# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

Axes	Depiction	Nodes
child	Dark green edges	The three nodes immediately below the current location
descendant	Dashed green line	The three child nodes mentioned above, plus the seven nodes below them, all the way down (their children and their children's children)
parent	Magenta edges	The node immediately above the current location
ancestor	Magenta dashed line	The parent plus its parent, and its parent's parent
preceding-sibling	Dark red edges	The two nodes to the left of the current location that have the same parent
preceding	Dark red dashed line	The preceding-sibling nodes plus the six other nodes that are entirely to the left of the current location
following-sibling	Blue edges	The node to the right of the current location that has the same parent
following	Blue dashed line	The following-sibling node plus the nine other nodes that entirely to the right of the current location

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

Symbol	Meaning	Expanded version
.	current context node	<code>self::*</code> (for elements)
..	parent element	<code>parent::*</code>
//	descendant axis	<code>descendant::</code> . At the beginning of a path expression, it means that the path starts at the document node.
@	attribute axis	<code>attribute::</code>

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="2.0"
3  .... xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
4  .... <xsl:output method="text" encoding="UTF-8" />
5  ....
6  .... <xsl:template match="/">
7  ....
8  .... |.... <xsl:value-of select="TEI/descendant::langUsage/language" />
9  ....
10 ....
11 </xsl:template>
12 </xsl:stylesheet>
```

out-text.txt U X

```
1 Arabic English French German Ancient Greek Transliterated Greek Modern Greek Hebrew Italian Latin Punic
```

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

## XPath Predicate

Predicates are **conditional expression** and are used to **filter the results** of the path expression.

## XPath Predicate

Predicate expression are written in **square brackets after the step** in the path expression to which they apply

**Any expression in square brackets that filters a step in a path expression is a predicate**

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="2.0">
3      xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
4      <xsl:output method="text" encoding="UTF-8" />
5
6      <xsl:template match="/">
7          <xsl:value-of select="//div[1]" />
8
9      </xsl:template>
10
11
12  </xsl:stylesheet>
```

out-text.txt U X

```
1
2      Bibliography
3          Not previously published.
4
```

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

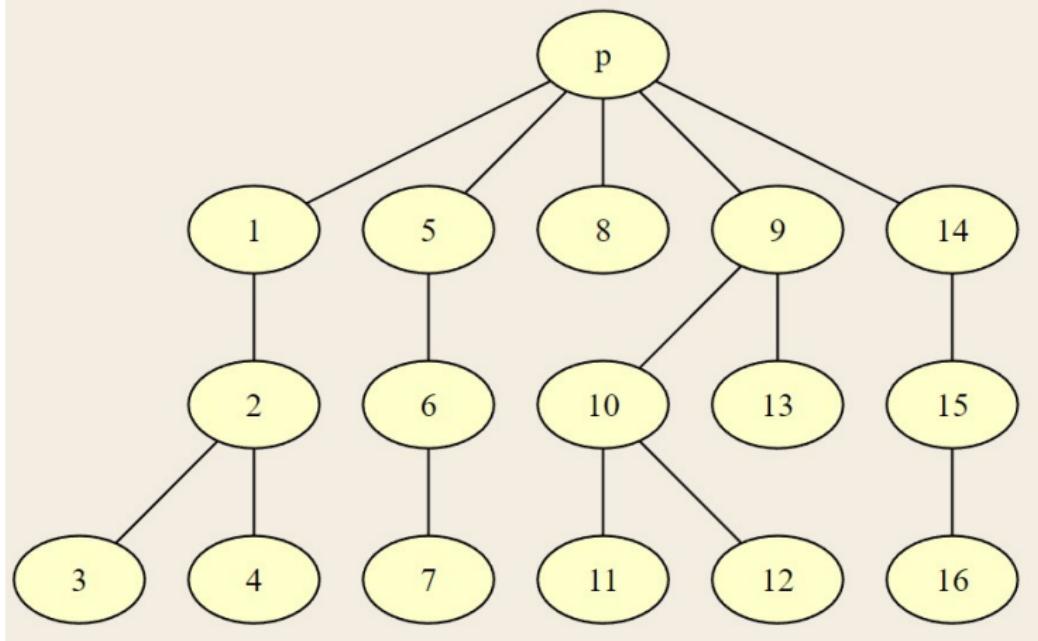
Introduction

XPath

XSL Trans-  
formations

XSL in action

References



# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

Description	Value	General
Equal to	eq	=
Not equal to	ne	!=
Greater than	gt	> (&gt;)
Greater than or equal to (not less than)	ge	>= (&gt;=)
Less than	lt	< (&lt;)
Less than or equal to (not greater than)	le	<= (&lt;=)

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

## Selecting and Processing XML Document Trees

### Examples of Predicates

- `div[@type]`
- `div[@type="edition"]`

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

## XPath Basics

### "jolly" element selection

- \*
- \*[@type]
- \*[@type="book"]

# Selecting and Processing XML Document Trees

## XPath: functions

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

### Esempio predicati

- `//div[@type="edition"]`
- `//div[@type!="translation"]`
- `//div[@n > 2]`
- `//div[1]`
- `//div[last()]`
- `//div[position() = last() - 1]`
- `//div[position() mod 2 = 0]`

# Selecting and Processing XML Document Trees

## XPath location path

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

## XPath Basics

Use of OR to select more than one named elements and ID  
function to select the element with the given ID

- "title | author"
- id("irt1952")

# Selecting and Processing XML Document Trees

## XPath location path

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="2.0">
3  ....xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
4  ....<xsl:output method="text" />
5  ....
6  ....<xsl:template match="/">
7  ....
8  ....<xsl:value-of
9  .... select="descendant::langUsage/language[@ident='it']/preceding-sibling::*[1]/text()"
10 .... />
11 ....
12 ....</xsl:template>
13
14 </xsl:stylesheet>
15
```

out-text.txt U ×

1 Hebrew

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

## XPath functions

- Functions operate on the information returned by a path expression or another function
- Functions can be nested
- functions can be used in predicates to filter expressions

# Selecting and Processing XML Document Trees

## XPath functions

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

```
2 <xsl:stylesheet version="2.0">
3   xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
4   <xsl:output method="text" encoding="UTF-8" />
5   <xsl:template match="/">
6     <xsl:value-of select="concat(
7       name(TEI/descendant::langUsage/language[1]), ',',
8       count(TEI/descendant::langUsage/language), ',',
9       //w[contains(., 'filii')]/@lemma, ',',
10      upper-case(
11        substring(TEI/descendant::langUsage/language[@ident eq 'it'], 0, 4)
12      ), ','
13    )"
14   "/>
15 </xsl:template>
16 </xsl:stylesheet>
```

out-text.txt U X

1 language 12 filius ITA

# Selecting and Processing XML Document Trees

## XPath functions

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

### fn:count

Counts the number of items in a sequence.

`count($arg as item()*) → xs:integer`

#### Arguments

\$arg      item()\*

The sequence whose items are  
to be counted

#### Result

xs:integer

### Namespace

<http://www.w3.org/2005/xpath-functions>

### Links to W3C specifications

[XPath 2.0 Functions and Operators](#)

[XPath 3.0 Functions and Operators](#)

[XPath 3.1 Functions and Operators](#)

### Saxon availability

Available in XPath 2.0, XSLT 2.0, XQuery 1.0, and later versions. Available in all Saxon

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

## fn:contains

Returns true if the second string is a substring of the first.

`contains($arg1 as xs:string?, $arg2 as xs:string?) → xs:boolean`

Arguments		
\$arg1	xs:string?	The containing string
\$arg2	xs:string?	The contained string
Result	xs:boolean	

`contains($arg1 as xs:string?, $arg2 as xs:string?, $collation as xs:string) → xs:boolean`

Arguments		
\$arg1	xs:string?	The containing string
\$arg2	xs:string?	The contained string
\$collation	xs:string	The collation to be used for comparing the strings
Result	xs:boolean	

## Namespace

<http://www.w3.org/2005/xpath-functions>

## Links to W3C specifications

[XPath 2.0 Functions and Operators](#)

[XPath 3.0 Functions and Operators](#)

[XPath 3.1 Functions and Operators](#)

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

## fn:name

`name() → xs:string`

Returns the name of the context node, as a string in the lexical form of a QName.

*There are no arguments*

<i>Result</i>	xs:string
---------------	-----------

`name($arg as node())? → xs:string`

Returns the name of the supplied node, as a string in the lexical form of a QName.

*Arguments*

\$arg	node()?
-------	---------

The node whose name is  
required

<i>Result</i>	xs:string
---------------	-----------

## Namespace

<http://www.w3.org/2005/xpath-functions>

## Links to W3C specifications

[XPath 2.0 Functions and Operators](#)

[XPath 3.0 Functions and Operators](#)

[XPath 3.1 Functions and Operators](#)

## Saxon availability

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

## fn:substring

`substring($sourceString as xs:string?, $start as xs:double) → xs:string`

Returns a substring of a given string starting at the given starting position and continuing to the end of the string.

### Arguments

<code>\$sourceString</code>	<code>xs:string?</code>	The input string
<code>\$start</code>	<code>xs:double</code>	The position of the first character of the input string to be included in the result

### Result

`xs:string`

`substring($sourceString as xs:string?, $start as xs:double, $length as xs:double) → xs:string`

Returns a substring of a given string starting at the given starting position and continuing to the end of the string, or `$length` characters if shorter.

### Arguments

<code>\$sourceString</code>	<code>xs:string?</code>	The input string
<code>\$start</code>	<code>xs:double</code>	The position of the first character of the input string to be included in the result
<code>\$length</code>	<code>xs:double</code>	The number of characters to be included in the result

### Result

`xs:string`

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

## fn:upper-case

Converts a string to upper case.

`upper-case($arg as xs:string?) → xs:string`

### Arguments

\$arg      xs:string?

The string to be converted to  
upper-case

### Result

xs:string

## Namespace

<http://www.w3.org/2005/xpath-functions>

## Links to W3C specifications

[XPath 2.0 Functions and Operators](#)

[XPath 3.0 Functions and Operators](#)

[XPath 3.1 Functions and Operators](#)

## Saxon availability

Available in XPath 2.0, XSLT 2.0, XQuery 1.0, and later versions. Available in all Saxon

# Selecting and Processing XML Document Trees

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

## fn:concat

Concatenates the string-values of the arguments into a single string. There must be at least two arguments.

`concat($arg1 as xs:anyAtomicType?, $arg2 as xs:anyAtomicType?, $etc... as xs:anyAtomicType?) → xs:string`

### Arguments

\$arg1	xs:anyAtomicType?	The first string
\$arg2	xs:anyAtomicType?	The second string
\$etc...	xs:anyAtomicType?	The third and subsequent strings (as many as required)

### Result

xs:string

## Namespace

<http://www.w3.org/2005/xpath-functions>

## Links to W3C specifications

[XPath 2.0 Functions and Operators](#)

[XPath 3.0 Functions and Operators](#)

[XPath 3.1 Functions and Operators](#)

# Progress status

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

1 Lesson path

2 Introduction

3 XPath

4 XSL Transformations

5 XSL in action

6 References

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

## Basics

A transformation expressed in XSLT **describes rules for transforming input data into output data**. The inputs and outputs will all be instances of the XDM data model.

# eXtensible Style Sheet

## XSL-T

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

### Basics (cont.)

The input often is an XML document referred to as the **source tree**, and the output is a document referred to as the **result tree**.

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

## Basics (cont.)

The transformation is achieved by **a set of template rules**. A template rule associates a pattern, which typically **matches nodes** in the source document, with a **sequence constructor**.

# eXtensible Style Sheet

## XSL-T

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

### Basics (cont.)

The structure of the result trees can be completely different from the structure of the source trees.

# XSLT main elements

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

## URL to Documentation

- <https://xsltdev.com/xslt/xsl-stylesheet/>
- <https://xsltdev.com/xslt/xsl-template/>
- <https://xsltdev.com/xslt/xsl-value-of/>
- <https://xsltdev.com/xslt/xsl-apply-templates/>
- <https://xsltdev.com/xslt/xsl-for-each/>
- <https://xsltdev.com/xslt/xsl-if/>
- <https://xsltdev.com/xslt/xsl-choose/>

# XSLT main elements

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

## URL to Documentation

- <https://xsltdev.com/xslt/xsl-sort/>
- <https://xsltdev.com/xslt/xsl-variable/>
- <https://xsltdev.com/xslt/xsl-element/>
- <https://xsltdev.com/xslt/xsl-attribute/>
- <https://xsltdev.com/xslt/xsl-key/>
- <https://xsltdev.com/xslt/xsl-preserve-space/>
- <https://xsltdev.com/xslt/xsl-strip-space/>
- <https://xsltdev.com/xslt/xsl-analyze-string/>

# XSLT main elements

## xsl:stylesheet

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

## xsl:stylesheet

The `xsl:stylesheet` element is always the top-level element of an XSLT stylesheet. The name `xsl:transform` may be used as a synonym.

*Available in XSLT 1.0 and later versions. Available in all Saxon editions.*

- **Content:** (*declarations*)

### Attributes

`id?`

*id*

Used to reference stylesheet modules embedded in a document.

`version`

*decimal*

Standard attribute that may appear on any XSLT element. Indicates the version of XSLT required by the stylesheet.

<https://xsltdev.com/xslt/xsl-stylesheet/>



# XSLT main elements

## xsl:stylesheet

```
<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="2.0">
    ...
        xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
    ...
</xsl:stylesheet>
```

<https://xsltdev.com/xslt/xsl-stylesheet/>

# XSLT main elements

## xsl:template

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

## xsl:template

Defines a processing rule for source elements or other nodes of a particular type.

*Available in XSLT 1.0 and later versions. Available in all Saxon editions.*

- **Category:** declaration
- **Content:** ( `xsl:context-item?` , `xsl:param*` , \*sequence-constructor\* )
- **Permitted parent elements:** `xsl:package` ; `xsl:stylesheet` ; `xsl:transform` ;  
`xsl:override`

### Attributes

`match?`

`pattern`

Pattern to identify the type of node to be processed. The most common form of pattern is simply an element name. However, more complex patterns may also be used: the syntax of patterns is given in more detail in XSLT Pattern Syntax. The following examples show some of the possibilities:

<https://xsltdev.com/xslt/xsl-template/>

# XSLT main elements

## xsl:template

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

```
<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="2.0">
    <!-- xmlns:xsl="http://www.w3.org/1999/XSL/Transform" -->

        <!--<xsl:template match="/">

        </xsl:template>

        <!--<xsl:template match="div">
            ...
            </xsl:template>

        <!--<xsl:template match="head">
            ...
            </xsl:template>

        <!--<xsl:template match="lb">
            ...
            </xsl:template>

    </xsl:stylesheet>
```

<https://xsltdev.com/xslt/xsl-template/>

# XSLT main elements

## xsl:value-of

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

## xsl:value-of

Evaluates an expression as a string, and outputs its value to the current result tree.

*Available in XSLT 1.0 and later versions. Available in all Saxon editions.*

- **Category:** instruction
- **Content:** sequence-constructor
- **Permitted parent elements:** any XSLT element whose content model is sequence-constructor; any literal result element

## Attributes

`select?`

`expression`

Identifies the expression. If this is not specified, the value to be output is obtained by evaluating the sequence constructor contained within the `xsl:value-of` element. The full syntax of expressions is outlined in XPath Expression Syntax. Here are some examples of expressions that can be used in the `select` attribute:

<https://xsltdev.com/xslt/xsl-value-of/>

# XSLT main elements

## xsl:value-of

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

```
<xsl:stylesheet version="2.0">
  <!-- xmlns:xsl="http://www.w3.org/1999/XSL/Transform" -->

  <xsl:template match="/">
    <xsl:value-of select="concat('&#10;', normalize-space(descendant::titleStmt/title=>string()))" />
  </xsl:template>

  <xsl:template match="div">
    <xsl:value-of select="@type" />
  </xsl:template>

  <xsl:template match="head">
    <xsl:value-of select="." />
  </xsl:template>

  <xsl:template match="lb">
    <xsl:value-of select="concat('&#10;', current()/@n)" />
  </xsl:template>

</xsl:stylesheet>

txt.txt u ×
<?xml version="1.0" encoding="UTF-8"?>
Dedication to Commodus
```

<https://xsltdev.com/xslt/xsl-value-of/>

# XSLT main elements

## xsl:apply-templates

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

## xsl:apply-templates

Causes navigation from the current element, usually but not necessarily to process its children.  
Each selected node is processed using the best-match `xsl:template` defined for that node.

*Available in XSLT 1.0 and later versions. Available in all Saxon editions.*

- **Category:** instruction
- **Content:** (`xsl:sort` | `xsl:with-param`)\*
- **Permitted parent elements:** any XSLT element whose content model is sequence-constructor; any literal result element

### Attributes

`select?`

`expression`

Sequence of nodes to be processed. If this attribute is omitted, then all the immediate children of the current node are processed.

`mode?`

<https://xsltdev.com/xslt/xsl-apply-templates/>



# XSLT main elements

## xsl:apply-templates

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

```
<?xml version="1.0" encoding="UTF-8"?>

<xsl:stylesheet version="2.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <xsl:template match="/">
    <xsl:value-of select="concat(&#10;, normalize-space(descendant::titleStmt/
      title=>string()))" />
    <xsl:apply-templates select="//text//div" />
  </xsl:template>

  <xsl:template match="div">
    <xsl:text>&#10;</xsl:text>
    <xsl:value-of select="@type" />
    <xsl:text>&#10;</xsl:text>
  </xsl:template>

  <xsl:template match="head">
    <xsl:value-of select=".." />
  </xsl:template>

  <xsl:template match="lb">
    <xsl:value-of select="concat(&#10;, current()/
      @n)" />
  </xsl:template>

</xsl:stylesheet>
```

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  Dedication to Commodus
3  bibliography
4
5  history
6
7  edition
8
9  apparatus
10
11 translation
12
13 commentary
14
15 figure
16
```

<https://xsltdev.com/xslt/xsl-apply-templates/>

# XSLT main elements

## xsl:apply-templates

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0">
  <!-- xmlns:xsl="http://www.w3.org/1999/XSL/Transform" -->
  <xsl:template match="/">
    <xsl:value-of select="concat('&#10;', normalize-space(descendant::titleStmt/title[string()]))"/>
    <xsl:apply-templates select="//text//div" />
  </xsl:template>

  <xsl:template match="div">
    <xsl:text>&#10;</xsl:text>
    <xsl:value-of select="name(.)=>concat(':::',@type)" />
    <xsl:text>&#10;</xsl:text>
    <xsl:apply-templates select="head" />
  </xsl:template>

  <xsl:template match="head">
    <xsl:value-of select="name(.)=>concat(':::',.,'.')"/>
  </xsl:template>

  <xsl:template match="lb">
    <xsl:value-of select="concat('&#10;', current()/@n)" />
  </xsl:template>
</xsl:stylesheet>
```

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  Dedication to Commodus
3  div : bibliography
4  head : Bibliography
5  div : history
6  head : Text constituted from
7  div : edition
8  head : Text
9  div : apparatus
10 head : Apparatus
11 div : translation
12 head : Translation
13 div : commentary
14 head : Commentary
15 div : figure
16 head : Photographs
```

<https://xsltdev.com/xslt/xsl-apply-templates/>

# XSLT main elements

## xsl:apply-templates

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

```
<?xml version="1.0" encoding="UTF-8"?>
<xsl:stylesheet version="2.0">
  <!-- xmlns:xsl="http://www.w3.org/1999/XSL/Transform">

  <!--xsl:strip-space elements="" /-->

  <xsl:template match="/">
    <xsl:value-of select="concat('&#10;', 'title : ', normalize-space(descendant::titleStmt/title>string()))"
      " />
    <xsl:apply-templates select="//text//div" />
  </xsl:template>

  <xsl:template match="div">
    <xsl:text>&#10;</xsl:text>
    <xsl:value-of select="name(.)=>concat('. : ', @type)" />
    <xsl:text>&#10;</xsl:text>
    <xsl:apply-templates select="head" />
  </xsl:template>

  <xsl:template match="head">
    <xsl:value-of select="name(.)=>concat('. : ', ., '&#10;')"
      " />
    <xsl:apply-templates select="following-sibling::*" />
  </xsl:template>

  <xsl:template match="lb">
    <xsl:value-of select="concat('&#10;', 'riga ', current()
      ()&#n, ' : ')"/>
  </xsl:template>

</xsl:stylesheet>
```

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  title : Dedication to Commodus
3  div : bibliography
4  head : Bibliography
5  Not previously published.
6  div : history
7  head : Text constituted from
8  Transcription (Reynolds, Ward-Perkins)
9  div : edition
10 head : Text
11 riga 1 : ImperatorCaesarMarciAntoniniPiifili
12 riga 2 : diuipinepediuHadrianiprompediu
13 riga 3 : TraianiParthiciabnepdiuNeruaeadnep
14 riga 4 : LuciAeliAureliCommmodAugust
15 div : apparatus
16 head : Apparatus
17 Line 4 is represented by the upper parts only of
two letters with rounded bowls.
18 div : translation
19 head : Translation
20 Emperor Caesar son of Marcus Antonius Pius,
grandson of deified Pius, great grandson of
deified Hadrian, great great grandson of deified
Trajan Victor in Parthia, great great great
grandson of deified Nerva, Lucius Aelius Aurelius
Commodus Augustus
21 div : commentary
22 head : Commentary
23 No comment
24 div : figure
25 head : Photographs
26 head : Ward-Perkins Archive, BSR (BSR 48.XXVII.13)
```

<https://xsltdev.com/xslt/xsl-apply-templates/>

# XSLT main elements

## xsl:for-each

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

## xsl:for-each

Causes iteration over the nodes selected by a node-set expression.

Available in XSLT 1.0 and later versions. Available in all Saxon editions.

- **Category:** instruction
- **Content:** ( `xsl:sort` \* , sequence-constructor )
- **Permitted parent elements:** any XSLT element whose content model is sequence-constructor; any literal result element

### Attributes

`select`

`expression`

Defines the nodes over which the statement will iterate. The XSLT statements subordinate to the `xsl:for-each` element are applied to each source node selected by the node-set expression in turn. The full syntax of expressions is outlined in XPath Expression Syntax.

<https://xsltdev.com/xslt/xsl-for-each/>



# XSLT main elements

## xsl:for-each xsl:sort

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xsl:stylesheet version="2.0">
3  ...  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
4  ...  <xsl:output method="text" encoding="UTF-8" />
5
6  ...  <xsl:template match="/">
7  ...    <xsl:for-each select="distinct-values(//w/@lemma)">
8  ...      <xsl:sort select=". " data-type="text" lang="la"/>
9  ...      <xsl:text>&#32;</xsl:text>
10     <xsl:value-of select=". " />
11     <xsl:text>&#32;</xsl:text>
12   </xsl:for-each>
13 </xsl:template>
14
15 </xsl:stylesheet>

out-text.txt U X
1 diuus filius imperator
```

<https://xsltdev.com/xslt/xsl-for-each/>

# XSLT main elements

## xsl:if xsl:key

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

### xsl:if

Used for conditional processing. It takes a mandatory test attribute, whose value is a boolean expression. The contents of the `xsl:if` element are expanded only if the expression is true.

*Available in XSLT 1.0 and later versions. Available in all Saxon editions.*

- **Category:** instruction
- **Content:** sequence-constructor
- **Permitted parent elements:** any XSLT element whose content model is sequence-constructor; any literal result element

### Attributes

`test`

*expression*

The boolean expression to be tested. The full syntax of boolean expressions is outlined in XPath Expression Syntax.

<https://xsltdev.com/xslt/xsl-if/>



# XSLT main elements

xsl:key xsl:if

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

```
1   <?xml version="1.0" encoding="UTF-8"?>
2   <xsl:stylesheet version="2.0">
3   |   ... xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
4   |   ... <xsl:output method="text" encoding="UTF-8" />
5
6   |   ... <xsl:key name="lang" match="language" use="@ident"></xsl:key>
7
8   |   ... <xsl:template match="/">
9   |   |   ... <xsl:if test="key('lang','la')">
10  |   |   |   ... <xsl:value-of select="key('lang','la')"/>
11  |   |   ... </xsl:if>
12  |   ... </xsl:template>
13
14  </xsl:stylesheet>
15
```

out-text.txt U ×

1 Latin

<https://xsltdev.com/xslt/xsl-key/>

# XSLT main elements

## xsl:variable

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

## xsl:variable

Used to declare a variable and give it a value. If it appears at the top level (immediately within `xsl:stylesheet`) it declares a global variable, otherwise it declares a local variable that is visible only within the stylesheet element containing the `xsl:variable` declaration. The value of a variable can be referenced within an expression using the syntax `$name`.

*Available in XSLT 1.0 and later versions. Available in all Saxon editions.*

- **Category:** declaration
- **Category:** instruction
- **Content:** sequence-constructor
- **Permitted parent elements:** `xsl:package` ; `xsl:stylesheet` ; `xsl:transform` ;  
`xsl:override` ; `xsl:function` ; any XSLT element whose content model is sequence-constructor; any literal result element

## Attributes

`name`

`eqname`

Defines the name of the variable.

# XSLT main elements

## xsl:variable

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
3   <xsl:output method="text" encoding="UTF-8" />
4   <xsl:key name="lang" match="language" use="@ident"></xsl:key>
5
6   <xsl:template match="/">
7     <xsl:if test="key('lang', 'la')">
8       <xsl:variable name="latLang" select="key('lang', 'la')"/>
9       <xsl:value-of select="">
10      <xsl:with-param value="concat('Selected Language: ', $latLang/text(), ' (' , $latLang/@ident, ')')"/>
11    </xsl:if>
12  </xsl:template>
13
14 </xsl:stylesheet>
```

out-text.txt U ×

1 Selected Language: Latin (la)

<https://xsltdev.com/xslt/xsl-variable/>

# Progress status

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

1 Lesson path

2 Introduction

3 XPath

4 XSL Transformations

5 XSL in action

6 References

# eXtensible Style Sheet

## Working with XSL-T

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

```
<teiHeader>
  <fileDesc>
    <titleStmt>
      <title>
        <rs type="textType">Christian</rs>
        <rs type="textType">funerary</rs> inscription for Helladios</title>
        <editor>J. M. Reynolds</editor>
        <editor>J. B. Ward-Perkins</editor>
      </titleStmt>
      <publicationStmt>
        <authority>Centre for Computing in the Humanities, King's College London</authority>
        <idno type="filename">IRT256a</idno>
        <availability>
          <p>Creative Commons licence Attribution UK 2.0 (<ref>http://creativecommons.org/licenses/by/uk/2.0/</ref>)
          <p>All reuse or distribution of this work must contain somewhere a link back to the original source</p>
        </availability>
      </publicationStmt>
    <sourceDesc> ...
```

*The Inscriptions of Roman Tripolitania -*  
<https://inslib.kcl.ac.uk/irt2009/>

# eXtensible Style Sheet

## Working with XSL-T

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- $Id$ -->
3 <xsl:stylesheet>
4   ... xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
5   ... xmlns:tei="http://www.tei-c.org/ns/1.0"
6   ... xmlns:xs="http://www.w3.org/2001/XMLSchema"
7   ... exclude-result-prefixes="t" version="2.0">
8 ...
9 ...
10 ...
11 ...
12 ...
13 ...
14 ...
15 ...
16 ...
17 ...
18 ...
19 ...
20 ...
21 ...
22 ...
23 ...
24 ...
25 ...
26 ...
27 ...
28 ...
29 ...
30 ...
31 ...
32 ...
33 ...
34 ...
35 ...
36 ...
37 ...
38 ...
39 ...
40 ...
41 ...
42 ...
43 ...
44 ...
45 ...
46 ...
47 ...
48 ...
49 ...
50 ...
51 ...
52 ...
53 ...
54 ...
55 ...
56 ...
57 ...
58 ...
59 ...
60 ...
61 ...
62 ...
63 ...
64 ...
65 ...
66 ...
67 ...
68 ...
69 ...
70 ...
71 ...
72 ...
73 ...
74 ...
75 ...
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
angelodel80@LAPTOP-V8V3MLG0:/mnt/c/Users/angel/risorse/universita/corsoCodifica/tools$ java -jar SaxonHE10-
3J/saxon-he-10.3.jar -s:source/IRT256a.xml -xsl:source/Stylesheets-9.4/start-edition.xsl -o:source/out/out-
-text-epidoc.html
```

```
1 <!DOCTYPE HTML><html xmlns:i18n="http://apache
org/cocoon/i18n/2.1">
2 ...
3 ...
4 ...
5 ...
6 ...
7 ...
8 ...
9 ...
10 ...
11 ...
12 ...
```

*The Inscriptions of Roman Tripolitania -*  
<https://inslib.kcl.ac.uk/irt2009/>

# eXtensible Style Sheet

## Working with XSL-T

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

### Bibliography

Not previously published.

### Text constituted from

Transcription (Reynolds, Ward-Perkins)

### Text

[Εύμοιριτο] Ἐλλάδιος ἔζησε ἔτι ν'  
[πλέον ἔλατ(τ)ον μῆν[ες γ'] ἡμέρας  
[δεκαπέντε ὁ Χριστὸς μετὰ τ(ο)ῦ πνεύ-  
[ματός σ(ο)υ ἐτ]ελεύ[τισεν μ]ηνὶ Παυνὶ<sup>5</sup>  
[έβδο]μι κατὰ τοὺς Ἀφρ<ο>υς

### Apparatus

#### Apparatus

The supplied letters are no longer legible.

#### Translation

[May he be well off]. Helladios lived approximately 50 years, three months and fifteen days. Christ be with your spirit. He died on the seventh of the month of Payn among the Africans.

# eXtensible Style Sheet

## Working with XSLT

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

```
<parameter>
  <name>edition-type</name>
  <value>interpretive</value>
  <value on="yes">diplomatic</value>
</parameter>


<parameter>
  <name>line-inc</name>
  <value>1</value>
</parameter>
```

*The Inscriptions of Roman Tripolitania -*  
<https://inslib.kcl.ac.uk/irt2009/>

# eXtensible Style Sheet

## Working with XSL-T

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

### Text

- 1 [.....]ΕΛΛΑΔΙΟΣΕΖΗΣΕΕΤΙΝ
- 2 [.....]Ν[...]ΗΜΕΡΑΣ
- 3 [.....]ΠΙΝΕΥ
- 4 [.....]ΕΛΕΥ[.....]ΝΙΑΥΝΙ
- 5 [...]ΜΙΚΑΤΑΤΟΥΣΑΦΡΥΣ

### Apparatus

#### Apparatus

The supplied letters are no longer legible.

#### Translation

[.....]. Helladios lived approximately 50 years, three months and fifteen days. Christ be with your spirit.  
He died on the seventh of the month of Payn among the Africans.

*The Inscriptions of Roman Tripolitania -*  
<https://inslib.kcl.ac.uk/irt2009/>

# eXtensible Style Sheet

## Working with XSL-T

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

```
<parameter>
  <name>edn-structure</name>
  <value>default</value>
  <value>ddbdp</value>
  <value>hgv</value>
  <value>inslib</value>
  <value>iосре</value>
  <value>edh</value>
  <value>edh-db</value>
  <value>rib</value>
  <value>sammelbuch</value>
  <value on="yes">sample</value>
  <value>eagle</value>
  <value>igcyr</value>
</parameter>
```

# eXtensible Style Sheet

## Working with XSL-T

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

Current repository: Unknown

Text type: Christianfunerary

Editor(s): J. M. Reynolds, J. B. Ward-Perkins

Changes history: 2010-08-18 GB Converted from TEI P4 (EpiDoc DTD v. 6) to P5 (EpiDoc RNG schema v. 8); 2008-09-09 ZA converted 2009-05-19 RV Added Figures; 2009-08-24 RV Added Figures

Publication details: Centre for Computing in the Humanities, King's College London;

Creative Commons licence Attribution UK 2.0 ()

All reuse or distribution of this work must contain somewhere a link back to the URL

Interpretive

Text

- 1 [Εύμοιριτο] Έλλαδιος ἔζησε ἐπι ν'
- 2 [πλέον ἔλατ(τ)ον μῆν[ες γ'] ἡμέρας
- 3 [δεκαπέντε ὁ Χριστός μετά τ(ο)ῦ πνεύ-
- 4 [ματός σ(ο)υ ἐτ]ελεύτισεν μι]νι Παυνί
- 5 [έβδο]μι κατὰ τοὺς Αφρο>υς

Diplomatic

Text

- 1 [.....]ΕΛΛΑΔΙΟΣΕΖΗΣΕΕΤΙΝ
- 2 [.....]Ν[...]ΗΜΕΡΑΣ
- 3 [.....]ΠΝΕΥ
- 4 [.....]ΕΛΕΥ[.....]ΝΙ.ΑΥΝΙ
- 5 [...]ΜΙΚΑΤΑΤΟΥΣΑΦΡΥΣ

# eXtensible Style Sheet

## Working with XSL-T

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Trans-  
formations

XSL in action

References

```
<p><b>Text type:</b>
<xsl:choose>
  <xsl:when test="//t:textClass//t:keywords//t:term[@type='textType']">
    <xsl:apply-templates select="//t:textClass//t:keywords//t:term[@type='textType']"/>
  </xsl:when>

  <!--xsl:when test="//t:teiHeader//t:rs[@type='textType']"-->
  <xsl:apply-templates select="//t:teiHeader//t:rs[@type='textType']"/>
</xsl:when-->
  <xsl:when test="//t:teiHeader//t:rs[@type='textType']">
    <xsl:for-each select="//t:teiHeader//t:rs[@type='textType']">
      <xsl:apply-templates select="current()"/><xsl:text></xsl:text>
    </xsl:for-each>
  </xsl:when>
</xsl:choose>
```

*The Inscriptions of Roman Tripolitania -*  
<https://inslib.kcl.ac.uk/irt2009/>

# eXtensible Style Sheet

## Working with XSLT

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

The screenshot shows a software interface for XSLT transformation. At the top, there are three tabs: "new-document 1", "transform.xq", and "htm-tpl-struct-sample.xsl". The "transform.xq" tab is currently selected. Below the tabs is a code editor containing XQuery code:

```
1 xquery version "3.1";
2
3 let $xml := doc("IRT256a.xml")
4 let $xsl := doc("Stylesheets-9.4/start-edition.xsl")
5
6 return transform:transform($xml,$xsl,())
7
```

Below the code editor is a navigation bar with a back arrow, a path indicator "/db/apps/asp-aiucd-2023/transform.xq", and several checkboxes: "XHTML Output" (unchecked), "Indent" (checked), "Live Preview" (checked), "Highlight Index Matches" (checked), and a plus icon.

The main pane displays the transformed XML output:

```
1 <html xmlns:i18n="http://apache.org/cocoon/i18n/2.1">
  <head>
    <title>IRT256a. Christian funerary inscription for Helladios</title>
    <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
    <link rel="stylesheet" type="text/css" media="screen, projection" href="global.css" />
  </head>
  <body>
    <h1>IRT256a. Christian funerary inscription for Helladios</h1>
    <div id="mainContent">
```

*The Inscriptions of Roman Tripolitania -*  
<https://inslib.kcl.ac.uk/irt2009/>

# Progress status

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

1 Lesson path

2 Introduction

3 XPath

4 XSL Transformations

5 XSL in action

6 References

# Bibliography

deepen into XPath and XSLT

XPath - XSL

A.M. Del  
Grosso

Lesson path

Introduction

XPath

XSL Transformations

XSL in action

References

## Some References

- XQuery and XPath Data Model 3.1  
<https://www.w3.org/TR/xpath-datamodel-31/>
- XSLT Recommendations  
<https://www.w3.org/TR/xslt/>
- XPath Recommendations  
<https://www.w3.org/TR/xpath/>
- Kay, M. (2011). XSLT 2.0 and XPath 2.0 Programmer's Reference. Wiley.
- Williams, I. (2009). Beginning XSLT and XPath: Transforming XML Documents and Data. Wiley.
- Walmsley, P. (2015) XQuery: Search Across a Variety of XML Data. O'Reilly.
- Saxonica documentation:  
<https://www.saxonica.com/documentation11/documentation.xml>