



Modelling humanities data with TEI-XML

SCHOLARLY EDITING AND MANUSCRIPT CATALOGUING IN THE DIGITAL AGE

Dr Katarzyna Anna Kapitan
20 November 2024

Annotations Recap

There were 44 leaders sailing from Greece to Troy. Their names are: Agamemnon Menelaus Arcesilaus Prothoenor Ascalaphus Ialmenusus Epistrophus Schedius Ajax Telamonius Teucus Amphimachus Diores Thalpius Polyxenus Nestor Thoas Nierus Ajax Oileus Antiphus Phidippus Idomeneus Meriones Ulixes Eumeles Protesilaus Podacres Podalirius Machaon Achilles Patroclo Tlepolemus Eurypylus Antiphus Amphimachus Polypoetes Leonteus Diomedes Euryalus Sthenelus Philoctetes Guneus Prothous Agapenor Menestheus

Their names in alphabetic order are:

- Achilles
- Agamemnon
- Agapenor
- Ajax Oileus
- Ajax Telamonius
- Amphimachus
- Amphimachus
- Antiphus
- Antiphus

```
20 | <teiHeader>
21 |   <listPerson>
22 |     <person xml:id="Aga_001">
23 |       <persName>Agamemnon</persName>
24 |       <idno type="wikidata">Q128176</idno>
25 |       <note>King of Mycenae; son of Atreus in Greek mythology.</note>
26 |     </person>
27 |   </listPerson>
28 |   </sourceDesc>
29 | </fileDesc>
30 | </teiHeader>
31 | <text>
32 |   <body>
33 |     <div type="chapter" n="14">
34 |       <p>
35 |         <s>Deinde ornati cum classe Graeci Athenas convenerunt:</s>
36 |         <s><persName ref="#Aga_001">Agamemnon</persName> ex Mycenis cum navibus numero
37 |           type="ship" value="100"> C</s> <s>Menelaus ex Sparta cum navibus numero
```

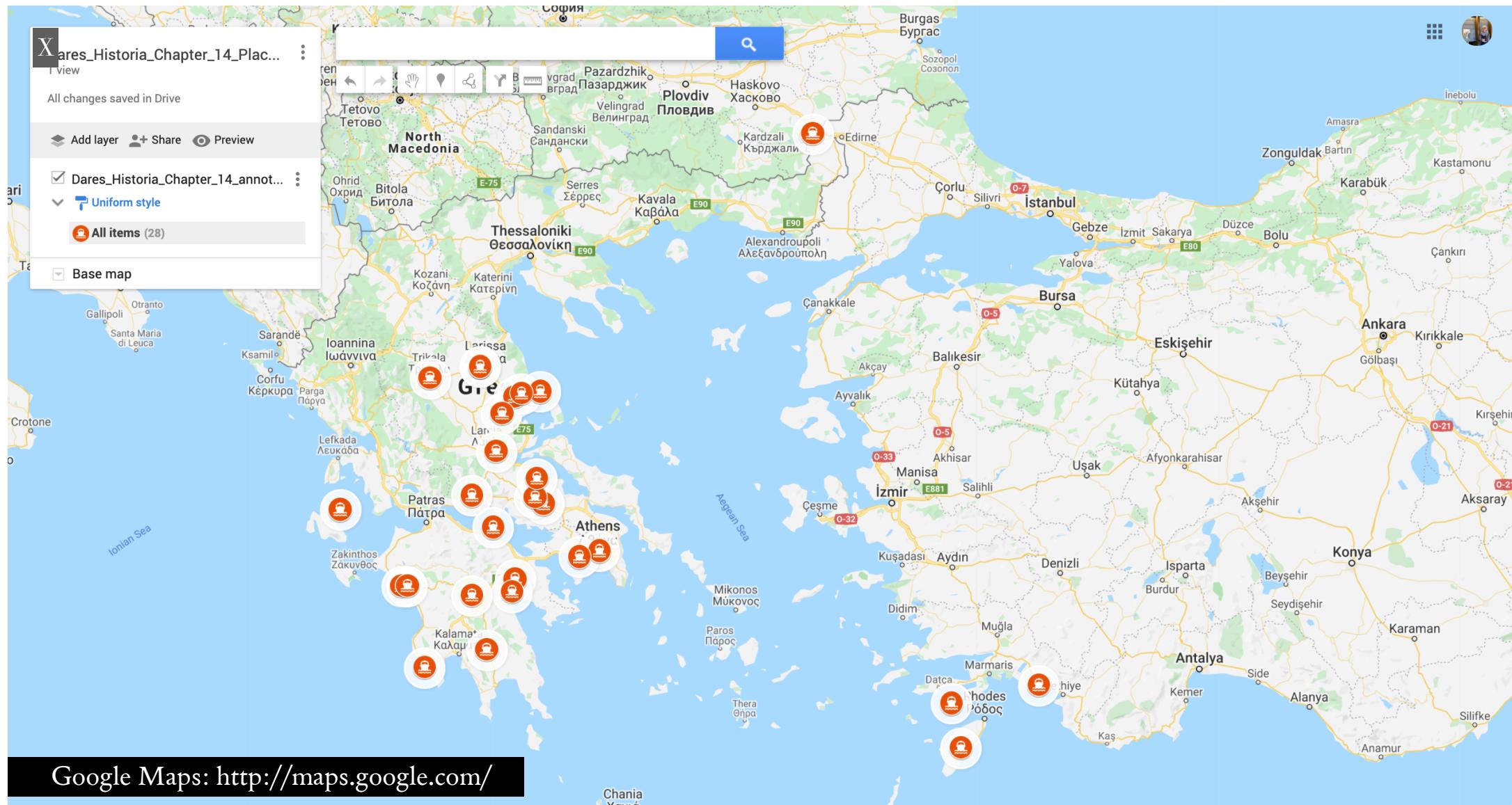
blesum. uultu bonū. aduncū. oculis rotundis. speciosum; Palamedem.
 gracile. longū. sapientem. animo magnū. blandū. Podaliriu. crassū. valente.
 supbū. tristem. Machaonē. fortē. magnū. certū. prudentē. paciente. misericordem.
 Merionē. rofū. medioerū statura. corpore rotundo. uiriosū. pru
 nacem. crudelē. in patientem. Beresidam. formosā. non alta statura.
 candidā. capillo molli et flauo. specilus uulstis. oculis venustis.
 corpore equali. blandam. affabiliē. uerecundā. animo pia. Deinde
 ornati cū classib. athenas conuener^z. Agamīnon ex micenis. cum
 nauib. numero centū. Menelaus ex sparta; nauib. numero sexaginta.
 Archelaus ex ptenor. ex boetia nauib; numero L. Ascalaphus et aliū
 nus ex oreomeno nauib; numero xxx. Epistrophus et secdius ex
 phodienno. nauib; numero xl. Ajax telamonius. ex salamina. ad
 duō secum reverū. frēm bubationē. Amphimachus. dorū. thesium.
 poluxenū. nauib; numero xl. Nestor. ex pilo. nauib; numero octoginta.
 Thoas ex etholia. nauib; numero xl. Uenerius ex im. nauib; numero
 quinquaginta trib. Ajax oileus ex locris. nauib; numero xxxvii.
 Antiphus philipp^z. thoas ex caledonx. nauib; numero xxx. Idomeneus
 et meriones. ex creta. nauib; numero octoginta. Ulxes ex itacha.
 nauib; numero xl. Emelovs ex pirgis. nauib; numero x. Proctlius.
 et potareus. ex pilaca. nauib; numero xl. Podalirius et macheon.
 ex colaphisi uercreci. nauib; numero xxxiib. Achilles cū patroclo.
 cum aliis. cum aliis. cum aliis. cum aliis. ex thaphia. Tlepolem^z

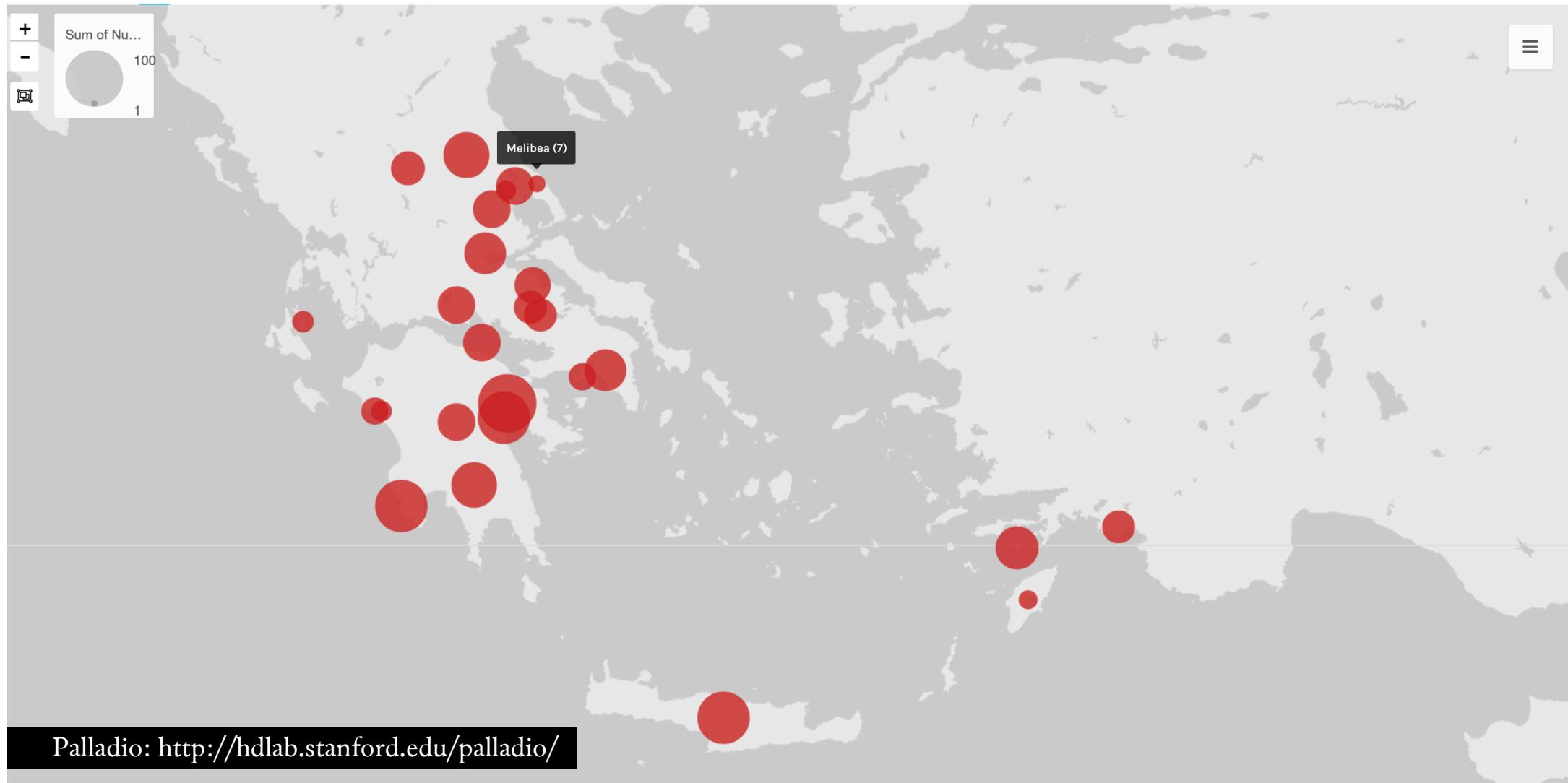
Switzerland, St Gall, Stiftsbibliothek, Cod. Sang. 197, p. 102 (detail).

There were 44 leaders sailing from Greece to Troy. Their names are: Agamemnon Amphimachus Diores Thalpius Polyxenus Nestor Thoas Nierus Ajax Oileus Antip Tlepolemus Eurypylus Antiphus Amphimachus Polypoetes Leonteus Diomedes Ei

Their names in alphabetic order are:

- Achilles
- Agamemnon
- Agapenor
- Ajax Oileus
- Ajax Telamonius
- Amphimachus
- Amphimachus
- Antiphus
- Antiphus
- Arcesilaus
- Ascalaphus
- Diomedes
- Diores
- Epistrophus
- Eumelus
- Euryalus
- Eurypylus
- Guneus
- Ialmenusus
- Idomeneus
- Leonteus
- Machaon
- Menelaus
- Menestheus
- Meriones
- Nestor
- Nierus
- Patroclo
- Phidippus
- Philoctetes
- Podacres



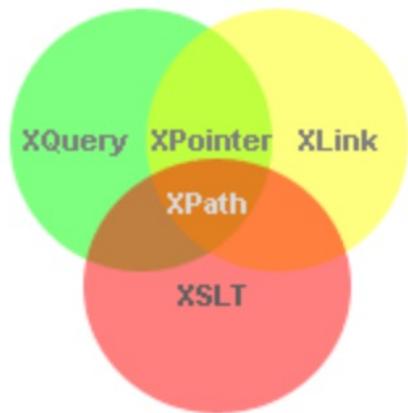


XPath

What is XPath?

XPath is a major element in the XSLT standard.

XPath can be used to navigate through elements and attributes in an XML document.



- XPath stands for XML Path Language
- XPath uses "path like" syntax to identify and navigate nodes in an XML document
- XPath contains over 200 built-in functions
- XPath is a major element in the XSLT standard
- XPath is a W3C recommendation

https://www.w3schools.com/xml/xpath_intro.asp

XPath

XPath – Important for, among other things:
Data export, transformations (XSLT), finding things in your document...

Below on section from the XSLT file used to extract relevant info from Dares, chapter 14.

```
able name="totalShipNumber" select="sum(text//num/@value)"/>
able name="totalPlaces" select="count(text//placeName)"/>
ame from <xsl:value-of select="$totalPlaces"/> cities in Greece a
ht <xsl:value-of select="$totalShipNumber"/> ships.</p>
city brought the following number of ships: <ul>
xsl:for-each select="text//s/placeName">
```

[next](#) [prev](#) [first](#)

Basic filepath-like path expressions

A bare-bones path expression is similar to filesystem addressing: if the path starts with a forward slash ("/"), then it represents a path from the root; if it does not start with a solidus then it represents a path from "here"

```
/TEI/teiHeader/fileDesc/titleStmt/title
```

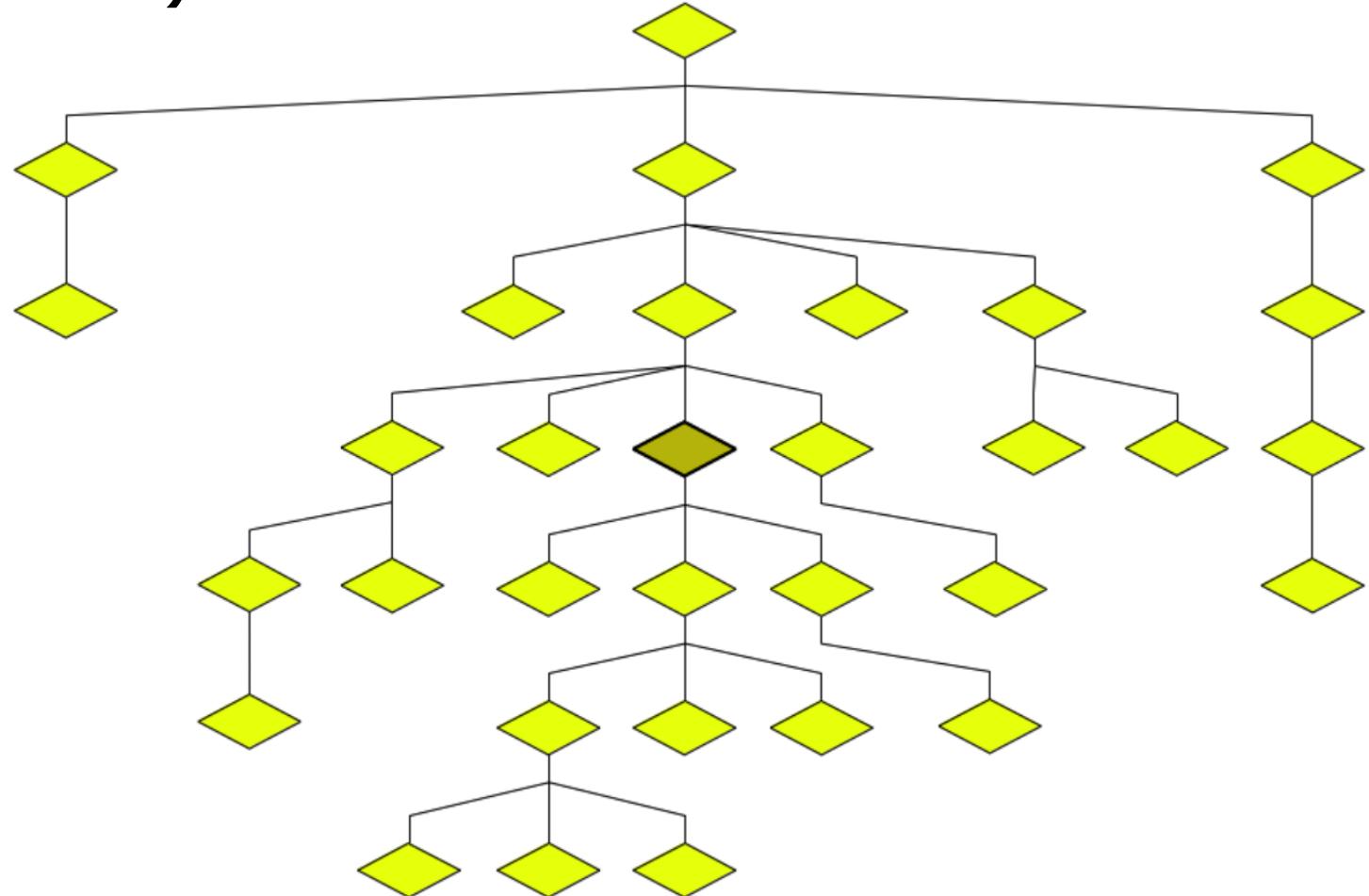
```
list/item/label
```



XPath and Schematron for TEI
Customization, slide 4 of 24

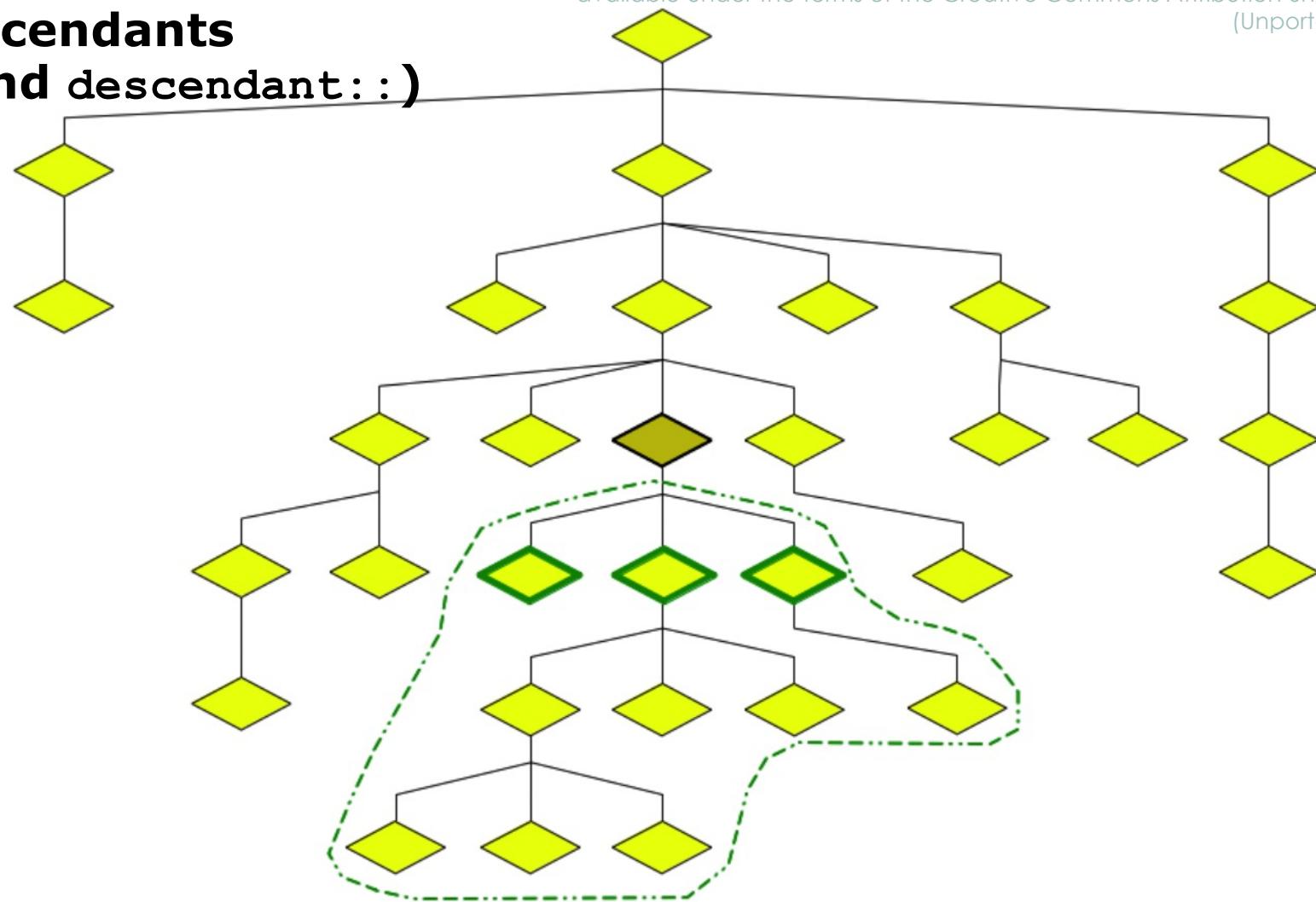
© 2010 Syd Bauman and the Women Writers Project This TEI-encoded XML file is available under the terms of the Creative Commons Attribution-ShareAlike 3.0 (Unported) license.

tree (with self::)



descendants **(child:: and descendant::)**

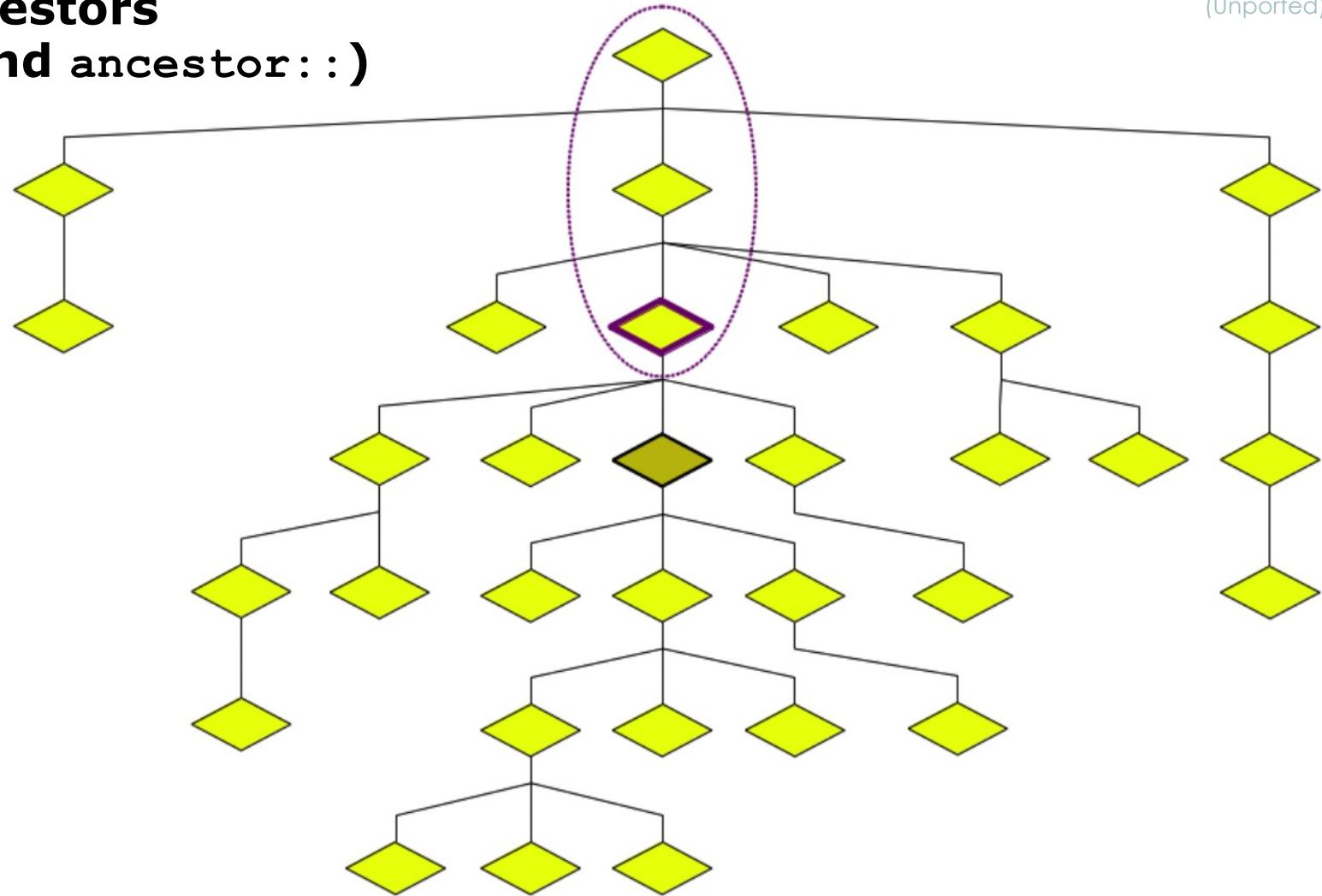
© 2010 Syd Bauman and the Women Writers Project This TEI-encoded XML file is available under the terms of the Creative Commons Attribution-ShareAlike 3.0 (Unported) license.



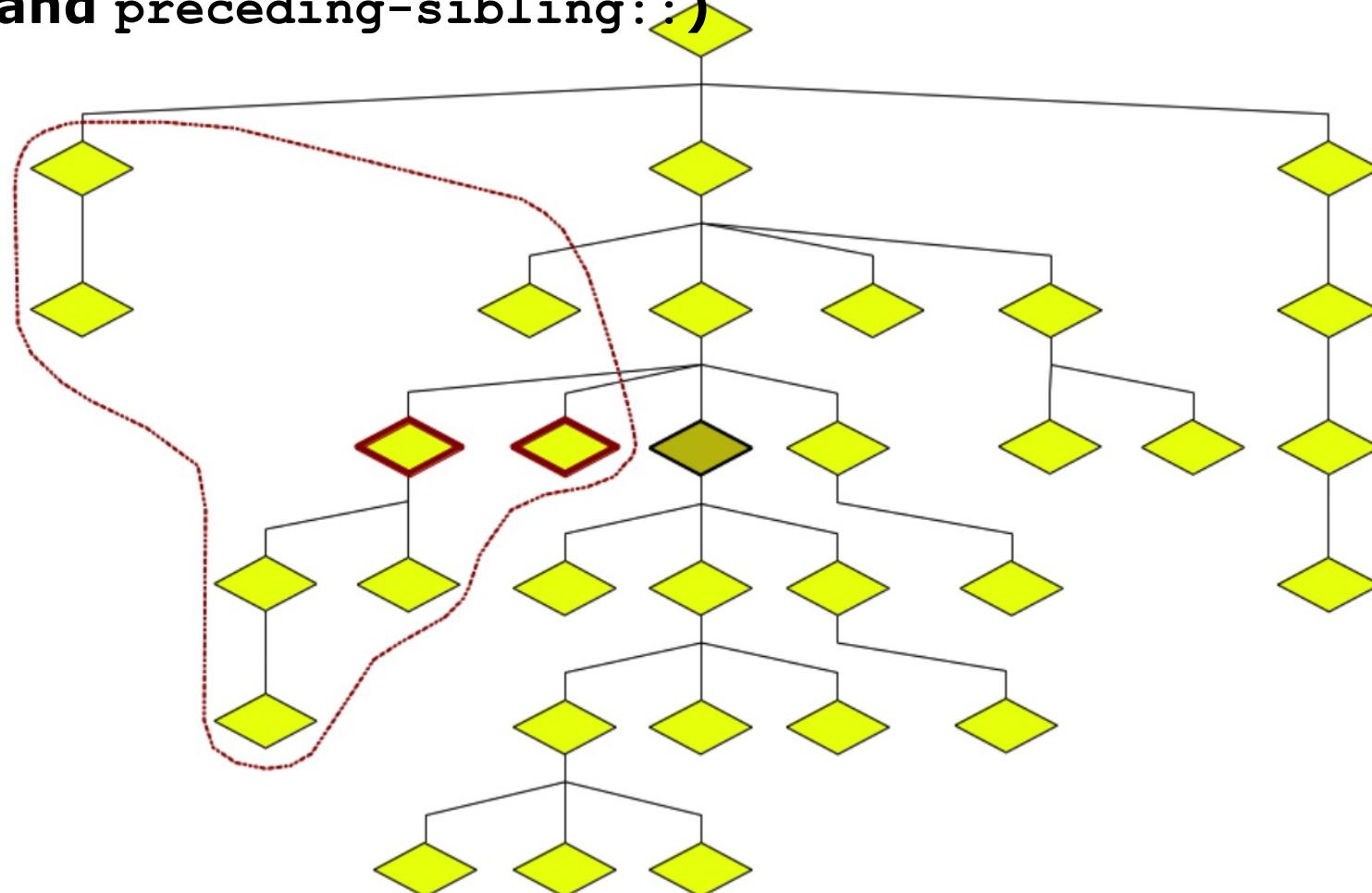
ancestors

(parent:: and ancestor::)

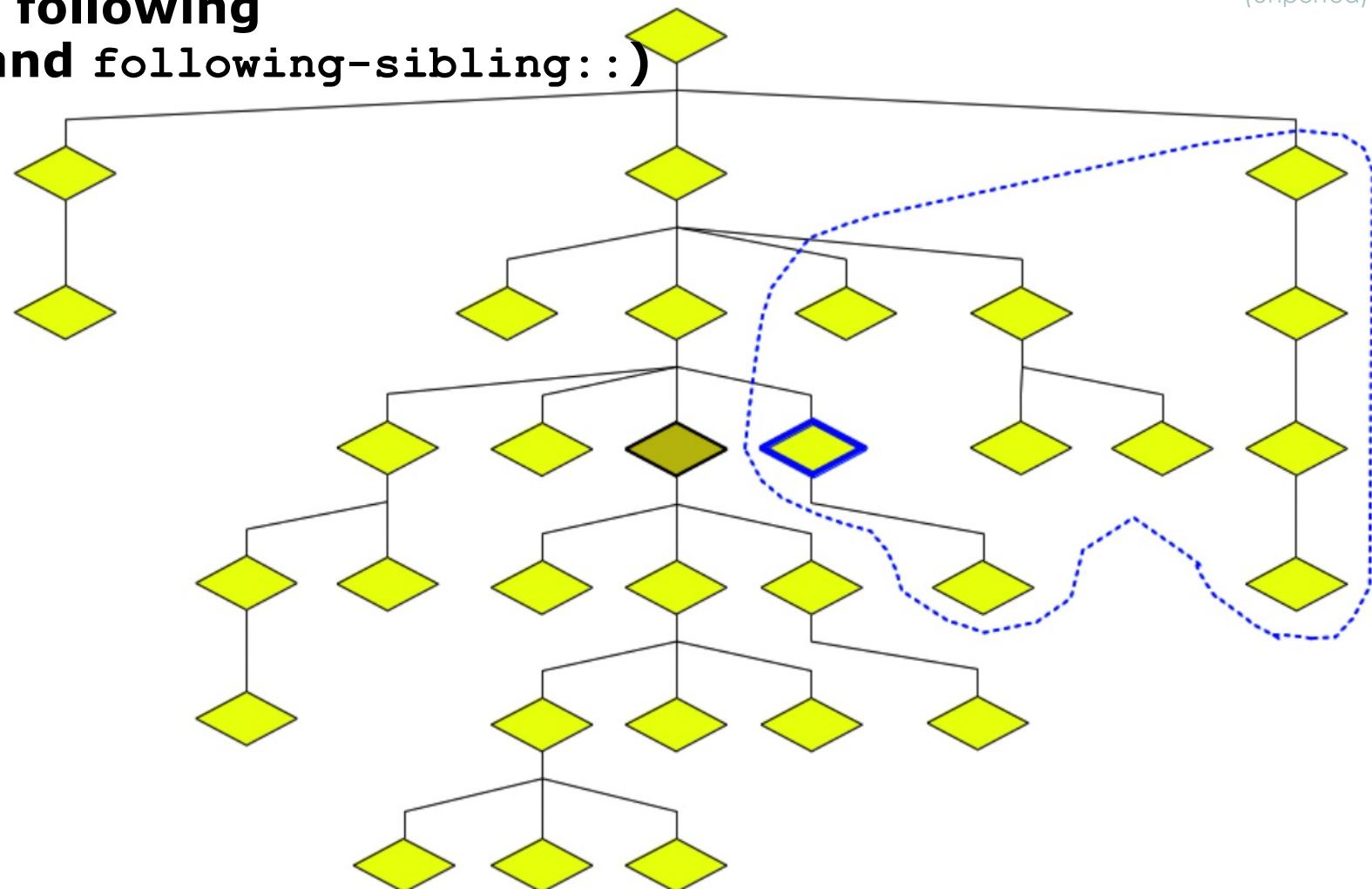
© 2010 Syd Bauman and the Women Writers Project This TEI-encoded XML file is available under the terms of the Creative Commons Attribution-ShareAlike 3.0 (Unported) license.



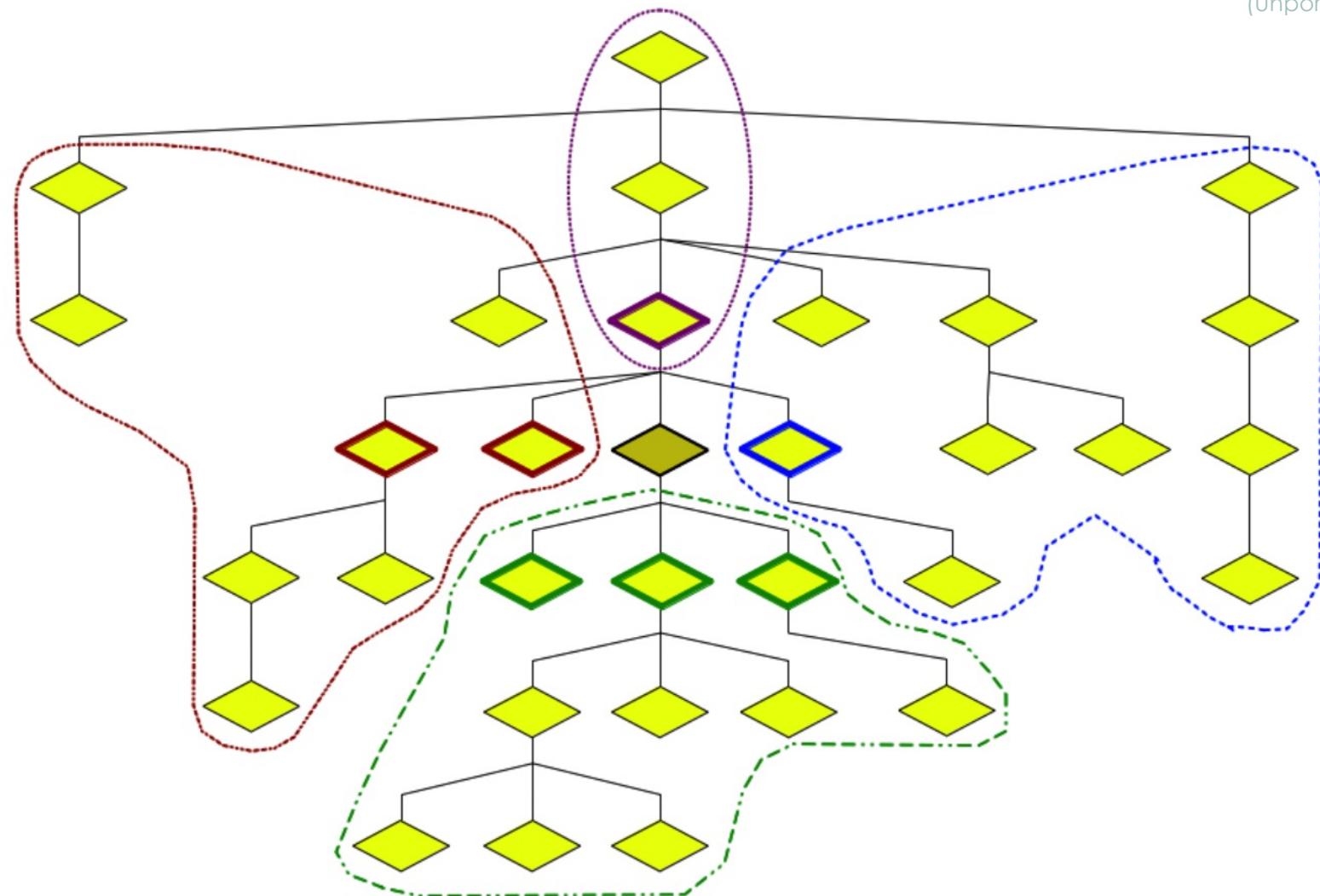
preceding (preceding:: and preceding-sibling::)



following **(following:: and following-sibling::)**



© 2010 Syd Bauman and the Women Writers Project This TEI-encoded XML file is available under the terms of the Creative Commons Attribution-ShareAlike 3.0 (Unported) license.



XPath Syntax

Expression	Description
<i>nodename</i>	Selects all nodes with the name " <i>nodename</i> "
/	Selects from the root node
//	Selects nodes in the document from the current node that match the selection no matter where they are
.	Selects the current node
..	Selects the parent of the current node
@	Selects attributes

https://www.w3schools.com/xml/xpath_syntax.asp

XPath Syntax

Wildcard	Description
*	Matches any element node
@*	Matches any attribute node
node()	Matches any node of any kind

Path Expression	Result
bookstore	Selects all nodes with the name "bookstore"
/bookstore	Selects the root element bookstore Note: If the path starts with a slash (/) it always represents an absolute path to an element!
bookstore/book	Selects all book elements that are children of bookstore
//book	Selects all book elements no matter where they are in the document
bookstore//book	Selects all book elements that are descendant of the bookstore element, no matter where they are under the bookstore element
//@lang	Selects all attributes that are named lang

https://www.w3schools.com/xml/xpath_syntax.asp

AxisName	Result
ancestor	Selects all ancestors (parent, grandparent, etc.) of the current node
ancestor-or-self	Selects all ancestors (parent, grandparent, etc.) of the current node and the current node itself
attribute	Selects all attributes of the current node
child	Selects all children of the current node
descendant	Selects all descendants (children, grandchildren, etc.) of the current node
descendant-or-self	Selects all descendants (children, grandchildren, etc.) of the current node and the current node itself
following	Selects everything in the document after the closing tag of the current node

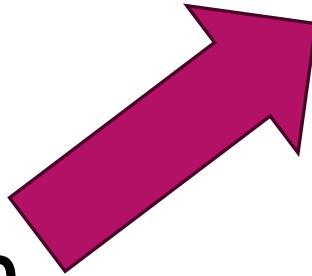
https://www.w3schools.com/xml/xpath_axes.asp

Testing XPath in Oxygen

test_Xpath.xml

On GitHub: /Classes/Week8/Exercises/test_XPath.xml

XPath 2.0



The screenshot shows the Oxygen XML Editor interface. On the left is the XML editor pane displaying the following XML code:

```
<?xml version="1.0" encoding="UTF-8"?>
< xmlDoc>
  <header><p>Para in header</p></header>
  <text ana="myText">
    <div n="1">
      <p n="1.1">Para 1 in section 1</p>
      <p n="1.2">Para 2 in section 1</p>
    </div>
    <div n="2">
      <p n="2.1">Para 1 in section 2</p>
      <p n="2.2">Para 2 in section 2</p>
    </div>
    <div n="3">
      <p n="3.1">Para 1 in section 3</p>
      <p n="3.2">Para 2 in section 3</p>
    </div>
  </text>
</ xmlDoc>
```

The XML code is color-coded for syntax highlighting. A context menu is open over the first 'div' element, listing options like 'Open Perspective', 'Show View', and 'Configure Toolbars...'. The 'Show View' option is currently selected. The right side of the interface contains various toolbars and a sidebar with links to other tools and resources.

Xpath: //p

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xmlDoc>
3  [ 3.1] <header><p>Para in header</p></header>
4  [ 4.1] <text ana="myText">
5  [ 5.1]   <div n="1">
6  [ 6.1]     <p n="1.1">Para 1 in section 1</p>
7  [ 7.1]     <p n="1.2">Para 2 in section 1</p>
8  [ 8.1]   </div>
9  [ 9.1]   <div n="2">
10 [ 10.1]     <p n="2.1">Para 1 in section 2</p>
11 [ 11.1]     <p n="2.2">Para 2 in section 2</p>
12 [ 12.1]   </div>
13 [ 13.1]   <div n="3">
14 [ 14.1]     <p n="3.1">Para 1 in section 3</p>
15 [ 15.1]     <p n="3.2">Para 2 in section 3</p>
16 [ 16.1]   </div>
17 [ 17.1] </text>
18 [ 18.1] </xmlDoc>
```

//div/p

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 ▷ < xmlDoc>
3   < header><p>Para in header</p></ header>
4 ▷ < text ana="myText">
5   ▷ < div n="1">
6     < p n="1.1">Para 1 in section 1</p>
7     < p n="1.2">Para 2 in section 1</p>
8   </ div>
9   ▷ < div n="2">
10    < p n="2.1">Para 1 in section 2</p>
11    < p n="2.2">Para 2 in section 2</p>
12  </ div>
13  ▷ < div n="3">
14    < p n="3.1">Para 1 in section 3</p>
15    < p n="3.2">Para 2 in section 3</p>
16  </ div>
17 </ text>
18 </ xmlDoc>
19
```

//p//text()

The screenshot shows an XML document named "test_XPath.xml" in an editor. The XML structure is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlDoc>
    <header><p>Para in header</p></header>
    <text ana="myText">
        <div n="1">
            <p n="1.1">Para 1 in section 1</p>
            <p n="1.2">Para 2 in section 1</p>
        </div>
        <div n="2">
            <p n="2.1">Para 1 in section 2</p>
            <p n="2.2">Para 2 in section 2</p>
        </div>
        <div n="3">
            <p n="3.1">Para 1 in section 3</p>
            <p n="3.2">Para 2 in section 3</p>
        </div>
    </text>
</xmlDoc>
```

The nodes selected by the XPath query //p//text() are highlighted in blue. These include all

elements within the element, specifically:

- Para in header
- Para 1 in section 1
- Para 2 in section 1
- Para 1 in section 2
- Para 2 in section 2
- Para 1 in section 3
- Para 2 in section 3

//parent::header

```
1  <?xml version="1.0" encoding="UTF-8"?>
2  <xmlDoc>
3  <header><p>Para in header</p></header>
4  <text ana="myText">
5    <div n="1">
6      <p n="1.1">Para 1 in section 1</p>
7      <p n="1.2">Para 2 in section 1</p>
8    </div>
9    <div n="2">
10      <p n="2.1">Para 1 in section 2</p>
11      <p n="2.2">Para 2 in section 2</p>
12    </div>
13    <div n="3">
14      <p n="3.1">Para 1 in section 3</p>
15      <p n="3.2">Para 2 in section 3</p>
16    </div>
17  </text>
18  </xmlDoc>
19
```

//text/*

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 ▼ <xmlDoc>
3 <header><p>Para in header</p></header>
4 ▼ <text ana="myText">
5 ▼   <div n="1">
6     <p n="1.1">Para 1 in section 1</p>
7     <p n="1.2">Para 2 in section 1</p>
8   </div>
9 ▼   <div n="2">
10    <p n="2.1">Para 1 in section 2</p>
11    <p n="2.2">Para 2 in section 2</p>
12   </div>
13 ▼   <div n="3">
14     <p n="3.1">Para 1 in section 3</p>
15     <p n="3.2">Para 2 in section 3</p>
16   </div>
17 </text>
18 </xmlDoc>
```

Translate

1. //title
2. //book/title
3. //chapter/footnote
4. //chapter//footnote
5. What's the difference between 3 and 4?

Exercise 1

Using Dares_Historia_Chapter_14_annotationsSpoiler.xml
(Week8/Exercises), find all:

1. geographical coordinates encoded in the authority list,
how many did you find?
2. names of people mentioned in the text, how many did
you find?
3. names of places mentioned in the text, how many did
you find?

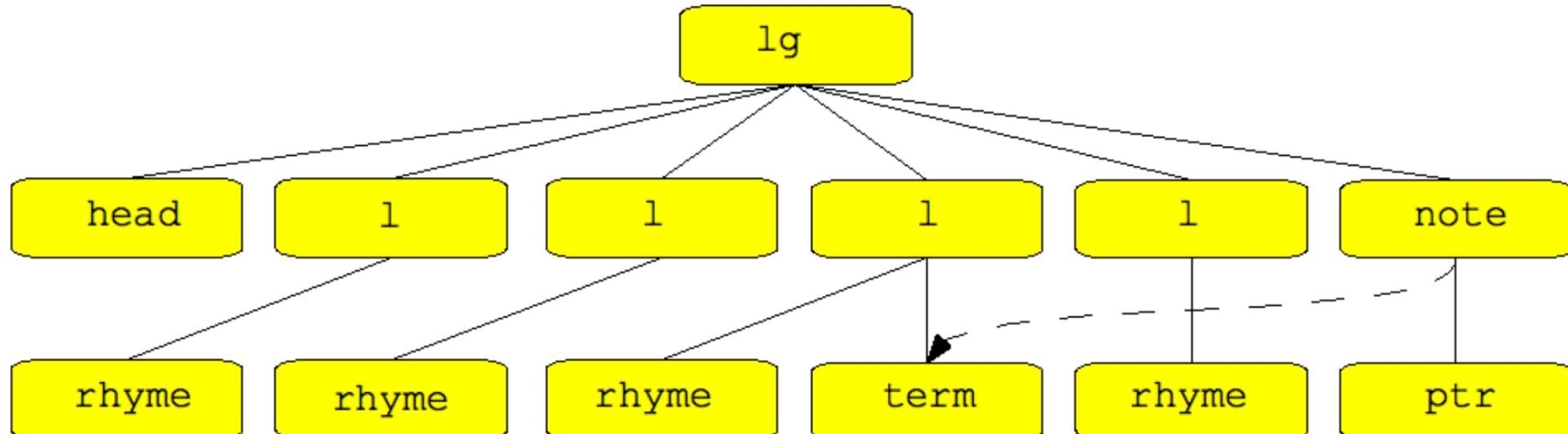
Sample document instance

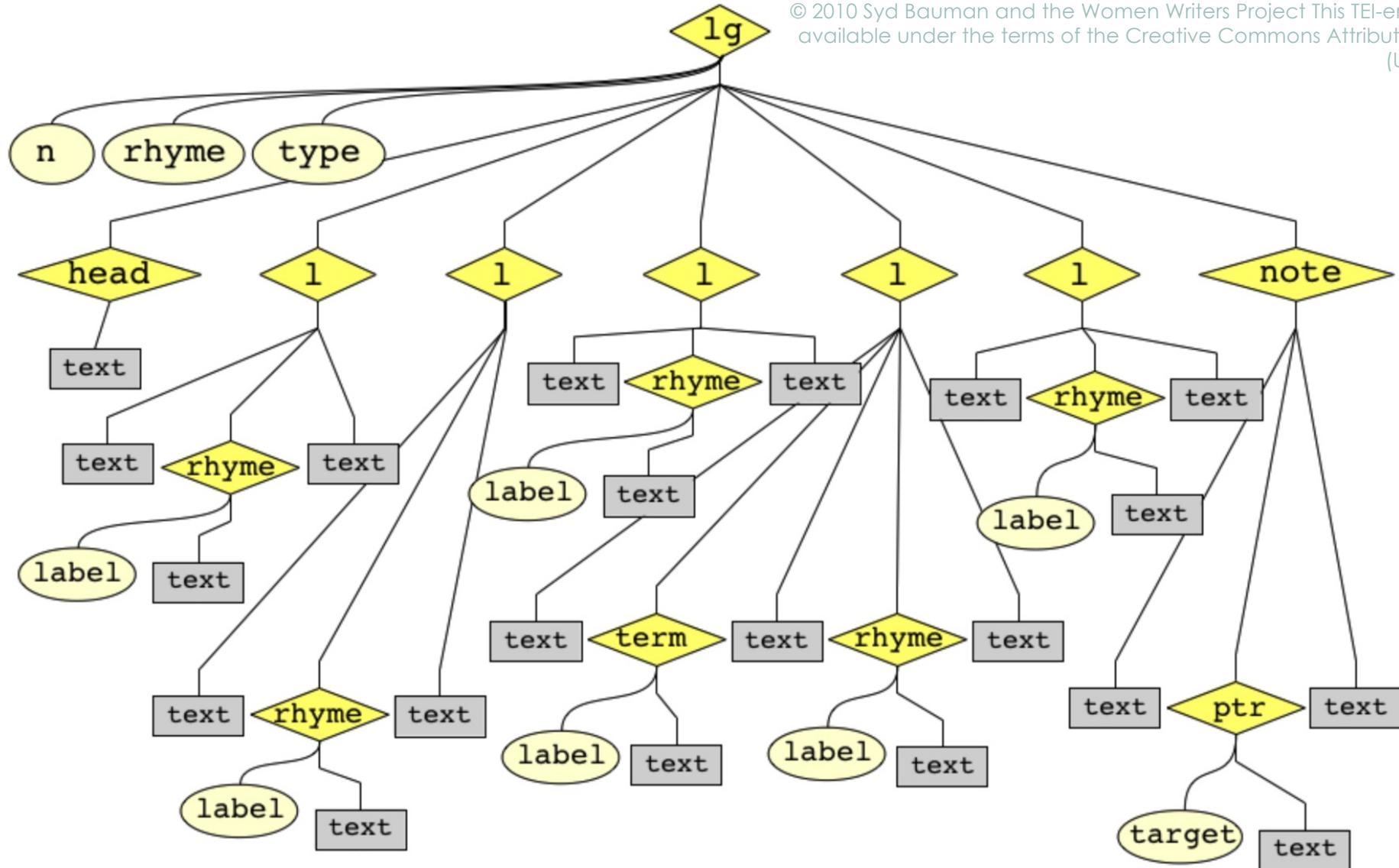
```
<?xml version="1.0" encoding="UTF-8"?>
<lg type="limerick" rhyme="aabba" n="3">
  <head>Warp Speed, Ms Bright!</head>
  <l>There was a young lady named <rhyme label="a">Bright</rhyme>,</l>
  <l>Who travelled much faster than <rhyme label="a">light</rhyme>,</l>
  <l>She departed one <rhyme label="b">day</rhyme>,</l>
  <l>In a <term xml:id="t17">relative</term> way <rhyme label="b">way</rhyme>,</l>
  <l>And returned on the previous <rhyme label="a">night</rhyme>.</l>
  <note target="#t17">See
    <ptr target="http://en.wikipedia.org/wiki/Theory_of_relativity"/>.</note>
</lg>
```



Simplified XML tree

© 2010 Syd Bauman and the Women Writers Project This TEI-encoded XML file is available under the terms of the Creative Commons Attribution-ShareAlike 3.0 (Unported) license.





Don't want them all?

- Many of the above XPaths return multiple nodes — what if you only want a particular one?
- If you only want Act 3, Scene 1:

```
/TEI/text/body/div[3]/div[1]
```

- Works well presuming you know what you want by element count.
- But in many cases, that is at least inconvenient, if not outright unknown.
- No matter how many <div>s there are, we know this scene has the identifier "sha-ham301" Thus:

```
//div[ @xml:id = 'sha-ham301' ]
```

selects the same node.



Predicates

An XPath *predicate* filters the nodes retrieved by a given step

Predicates are expressed after the node test in square brackets

The following are based on http://www.wwp.neu.edu/outreach/seminars/cust_2011-08/demos/xslt_intro/places.xml

XPath	selects
//listPlace/place[1]	the first <place> of each <listPlace> (of which there only happens to be one)
//*[@cRef]	all elements that have a cRef= attribute
//title[@level='m']	all monographic titles
/TEI/text//name[not(@key)]	<name> elements that are missing their key= attributes
//lg[@type='song']/l[1]	list first line of each song (16 nodes)
(//lg[@type='song']/l)[1]	returns first line of all songs (1 node)



//p[@n="2.1"]

The screenshot shows a TEI-XML editor interface with the following details:

- Toolbar:** Includes icons for search, settings, checkmark, play, refresh, and various file operations.
- Search Bar:** Displays the XPath query `//p[@n="2.1"]`.
- Document Tab:** Shows the file `test_XPath.xml*`.
- Document Structure:** The XML code is displayed with line numbers and color-coded elements:
 - Line 1: `<?xml version="1.0" encoding="UTF-8"?>`
 - Line 2: `< xmlDoc >`
 - Line 3: `<header><p>Para in header</p></header>`
 - Line 4: `<text ana="myText">`
 - Line 5: `<div n="1">`
 - Line 6: `<p n="1.1">Para 1 in section 1</p>`
 - Line 7: `<p n="1.2">Para 2 in section 1</p>`
 - Line 8: `</div>`
 - Line 9: `<div n="2">`
 - Line 10: `<p n="2.1">Para 1 in section 2</p>` (highlighted in blue)
 - Line 11: `<p n="2.2">Para 2 in section 2</p>`
 - Line 12: `</div>`
 - Line 13: `<div n="3">`
 - Line 14: `<p n="3.1">Para 1 in section 3</p>`
 - Line 15: `<p n="3.2">Para 2 in section 3</p>`
 - Line 16: `</div>`
 - Line 17: `</text>`
 - Line 18: `</ xmlDoc >`
 - Line 19:

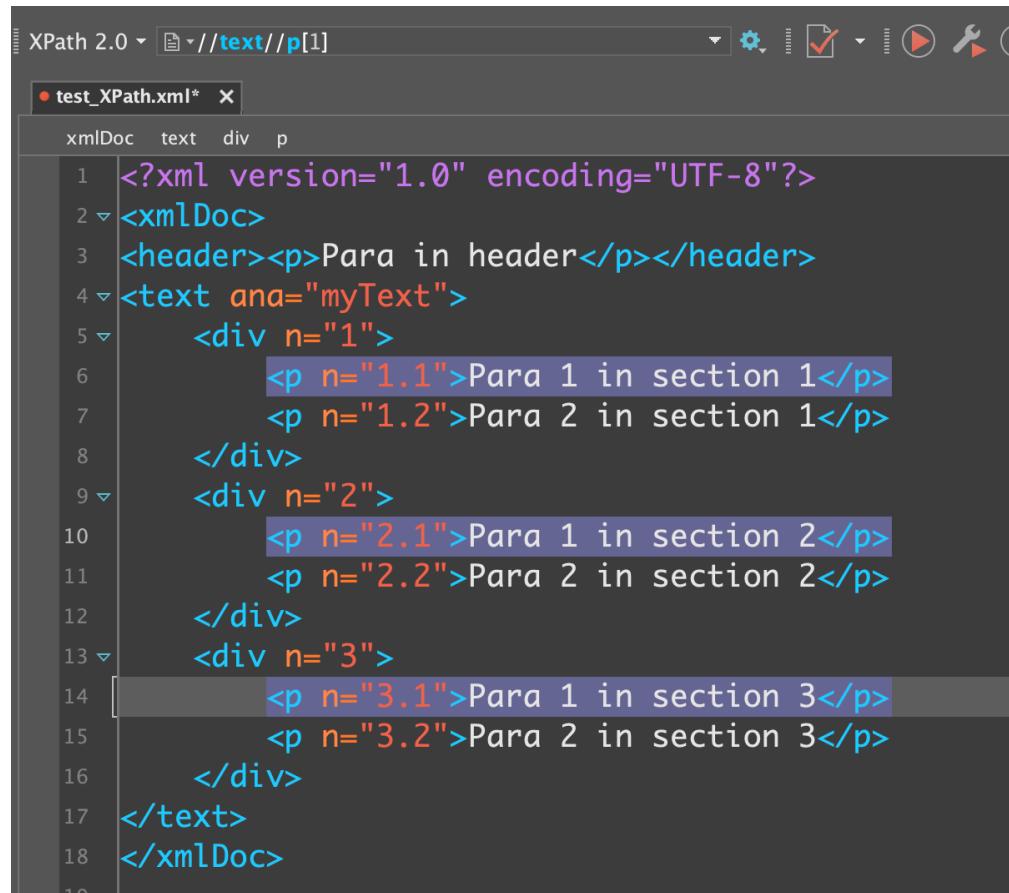
//*[@n]

The screenshot shows an XML editor interface with the following details:

- Toolbar:** Includes icons for search, filter, run, and save.
- Query Bar:** Displays "XPath 2.0" and the query `//*[@n]`.
- Results Pane:** Titled "Results" and lists "Description – 9 items".
- Items:** A list of nine items, each with a tooltip:
 - n="1" (Para 1 in section 1)
 - n="2" (Para 2 in section 1)
n="2" (Para 1 in section 2)
 - n="3" (Para 2 in section 2)
 - n="3" (Para 1 in section 3)
 - n="3" (Para 2 in section 3)
- Code View:** Shows the XML document structure with the following code:

```
<?xml version="1.0" encoding="UTF-8"?>
<xmlDoc>
  <header><p>Para in header</p></header>
  <text ana="myText">
    <div n="1">
      <p n="1.1">Para 1 in section 1</p>
      <p n="1.2">Para 2 in section 1</p>
    </div>
    <div n="2">
      <p n="2.1">Para 1 in section 2</p>
      <p n="2.2">Para 2 in section 2</p>
    </div>
    <div n="3">
      <p n="3.1">Para 1 in section 3</p>
      <p n="3.2">Para 2 in section 3</p>
    </div>
  </text>
</xmlDoc>
```

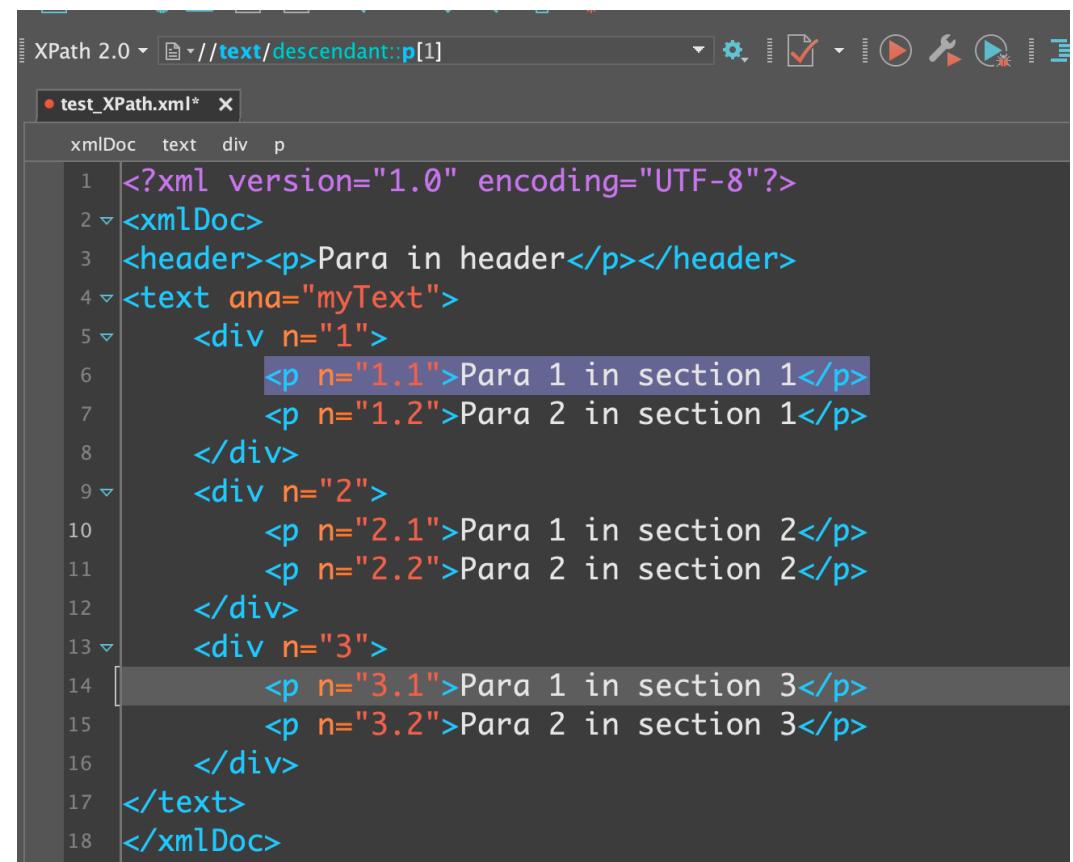
//text//p[1]



The screenshot shows an XML document with three sections (n=1, 2, 3) each containing two paragraphs (n=1.1, 1.2 and n=2.1, 2.2). The XPath expression //text//p[1] selects the first paragraph from each section, resulting in three paragraphs: "Para 1 in section 1", "Para 2 in section 1", and "Para 1 in section 2".

```
<?xml version="1.0" encoding="UTF-8"?>
< xmlDoc>
<header><p>Para in header</p></header>
<text ana="myText">
    <div n="1">
        <p n="1.1">Para 1 in section 1</p>
        <p n="1.2">Para 2 in section 1</p>
    </div>
    <div n="2">
        <p n="2.1">Para 1 in section 2</p>
        <p n="2.2">Para 2 in section 2</p>
    </div>
    <div n="3">
        <p n="3.1">Para 1 in section 3</p>
        <p n="3.2">Para 2 in section 3</p>
    </div>
</text>
</ xmlDoc>
```

//text/descendant::p[1]



The screenshot shows the same XML document as above. The XPath expression //text/descendant::p[1] selects all paragraphs in the document, resulting in six paragraphs: "Para in header", "Para 1 in section 1", "Para 2 in section 1", "Para 1 in section 2", "Para 2 in section 2", and "Para 1 in section 3".

```
<?xml version="1.0" encoding="UTF-8"?>
< xmlDoc>
<header><p>Para in header</p></header>
<text ana="myText">
    <div n="1">
        <p n="1.1">Para 1 in section 1</p>
        <p n="1.2">Para 2 in section 1</p>
    </div>
    <div n="2">
        <p n="2.1">Para 1 in section 2</p>
        <p n="2.2">Para 2 in section 2</p>
    </div>
    <div n="3">
        <p n="3.1">Para 1 in section 3</p>
        <p n="3.2">Para 2 in section 3</p>
    </div>
</text>
</ xmlDoc>
```

Translate

1. //book[@category='fiction']
2. //*[@type]
3. //chapter[5]/s[1]

Translate

1. Give me all items of a list.
2. Give me the first item of a list
3. Give me all elements that have attribute 'ana'
4. Give me all children elements of the first division element, which itself is a child of text
5. Give me all title elements which have an attribute 'type' with the attribute value 'uniform'.

Exercise 2

- Using Dares_Historia_Chapter_14_annotationsSpoiler.xml (Week8/Ex), find out:
1. which place name is the first one mentioned in the chapter
 2. which place name is the second one mentioned in the chapter
 3. which place name is the first one mentioned in the third sentence
 4. which person name is the fourth mentioned in the chapter.
 5. find out how many ships brought Achilles

Exercise 3.1

- ▶ **In groups:** prepare **five** XPath questions (in prose) for other groups to answer for one of the following files.
 - ▶ **Group 1:** Munich_Bayerische_Staatsbibliothek_ClM_305_transcrSpoiler.xml (Week6/Ex1)
 - ▶ **Group 2:** Dares_editionSpoiler.xml (Week7/Ex1)
 - ▶ **Group 3:** Ex1_Paris_BnF_Latin_5691_Description_NKY.xml (Week5/Ex1)
 - ▶ **Group 4:** Ex2_FirstFolioSpoiler_Bod.xml (Week3/Exercises)

Exercise 3.2

- ▶ Post your questions in:
<https://tinyurl.com/XPathQuiz>