

transform.xq for dummies

Eric van der Vlist

vdv@dyomedea.com
[@evlist](#)



<http://vdv.re/xmlondon2015>

topic title

Data-Driven Programming in XQuery

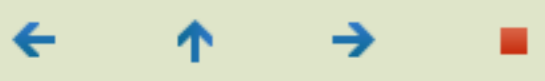
Wish me good luck

- New computer
- New OS version (Ubuntu 15.04)
- New oXygen version (17)
- New presentation format (George Bina's DITA CSS)
- New topics (XQuery, higher-order functions)

Copyright © Dyomedea, 2015

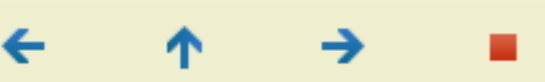
Text Grid Author

topic title



Data-Driven Programming in XQuery

Introduction



Copyright © Dyomedea, 2015

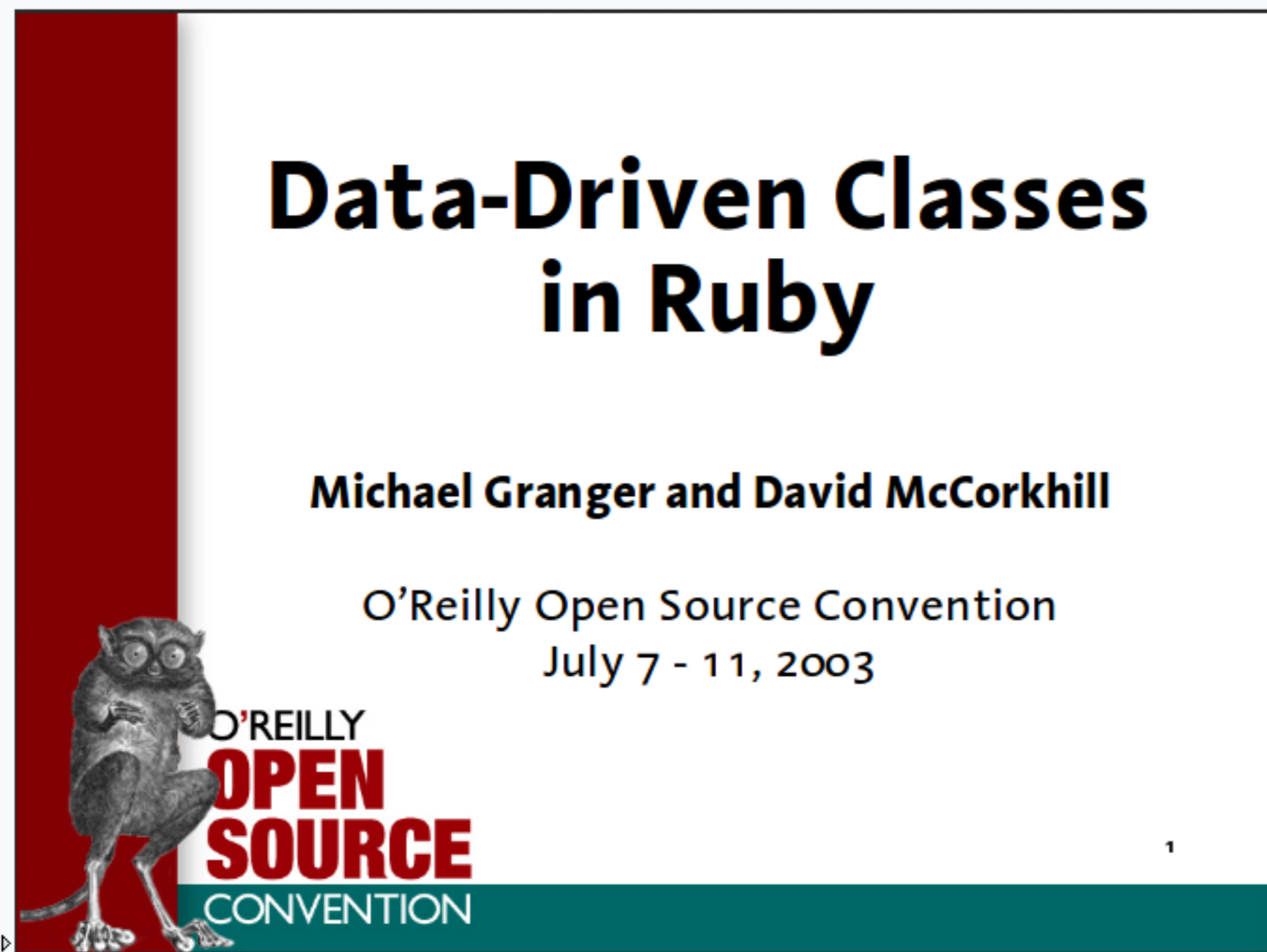
Text Grid Author

topic topic

Navigation arrows: left, up, right, and a red square stop button.

Introduction

A looong time ago



[granger_mccorkhill.pdf](#)

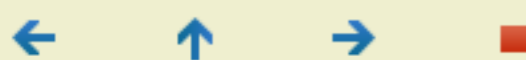
Navigation arrows: left, up, right, and a red square stop button.

The screenshot shows the Oxygen XML Editor interface. The main content area displays a slide with a light blue background. At the top right of the slide, the title "Data-Driven Programming in XQuery" is written in red. Below the title, the word "Introduction" is written in a large, bold, red font. Underneath "Introduction", the word "Eureka" is written in a smaller, bold, red font. Below "Eureka", there is a bulleted list of three items:

- Data-driven techniques minimize the impact of new or changing data structures and APIs
- Expressing functionality with data allows for run-time changes
- Allows code to be managed in parceled chunks

Below the list, there is a blue underlined link: [granger_mccorkhill.pdf](#). At the bottom right of the slide, the text "Copyright © Dyomedea, 2015" is visible. The editor's toolbar and menu are visible at the top, and the status bar at the bottom shows the file path, cursor position (U+0000), and a notification for 6 new messages.

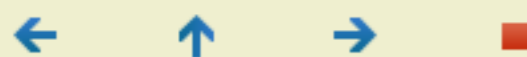
topic topic title

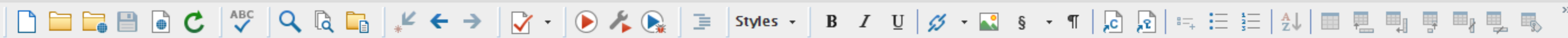


Introduction

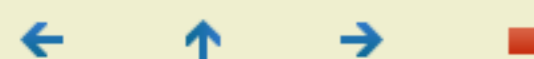
A source of inspiration

- [XML Driven Classes in Python](#) ([OSCON 2004](#))
- [TreeBind](#) ([XTech 2006](#) and [XML 2006](#))





topic topic title



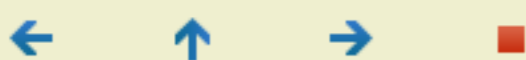
Data-Driven Programming in XQuery

Introduction

What are “data-driven” classes?

- All classes are data-driven in some sense.
- Data-driven classes are classes defined by dynamic information.
 - Classes which modify or create their behavior based on state (data).
 - Classes not (completely) defined by source or object code

 [granger_mccorkhill.pdf](#)



Copyright © Dyomedea, 2015

Text Grid Author

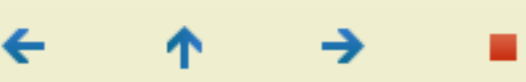
topic topic title



Data-Driven Programming in XQuery

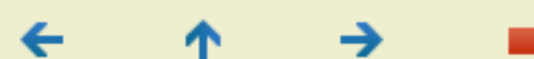
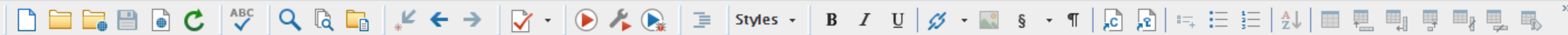
Introduction

What is Data-Driven programming?



Copyright © Dyomedea, 2015

Text Grid Author



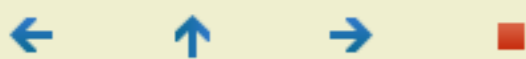
Introduction

What is Data-Driven programming?

Wikipedia

“In computer programming, data-driven programming is a programming paradigm in which the program statements describe the data to be matched and the processing required rather than defining a sequence of steps to be taken.[1] Standard examples of data-driven languages are the text-processing languages sed and AWK,[1] where the data is a sequence of lines in an input stream – these are thus also known as line-oriented languages – and pattern matching is primarily done via regular expressions or line numbers.”

--  [Wikipedia](#)



The screenshot shows the Oxygen XML Editor interface. The main content area displays a slide with the following text:

topic topic topic title

← ↑ → ■ Data-Driven Programming in XQuery

Introduction

What is Data-Driven programming?

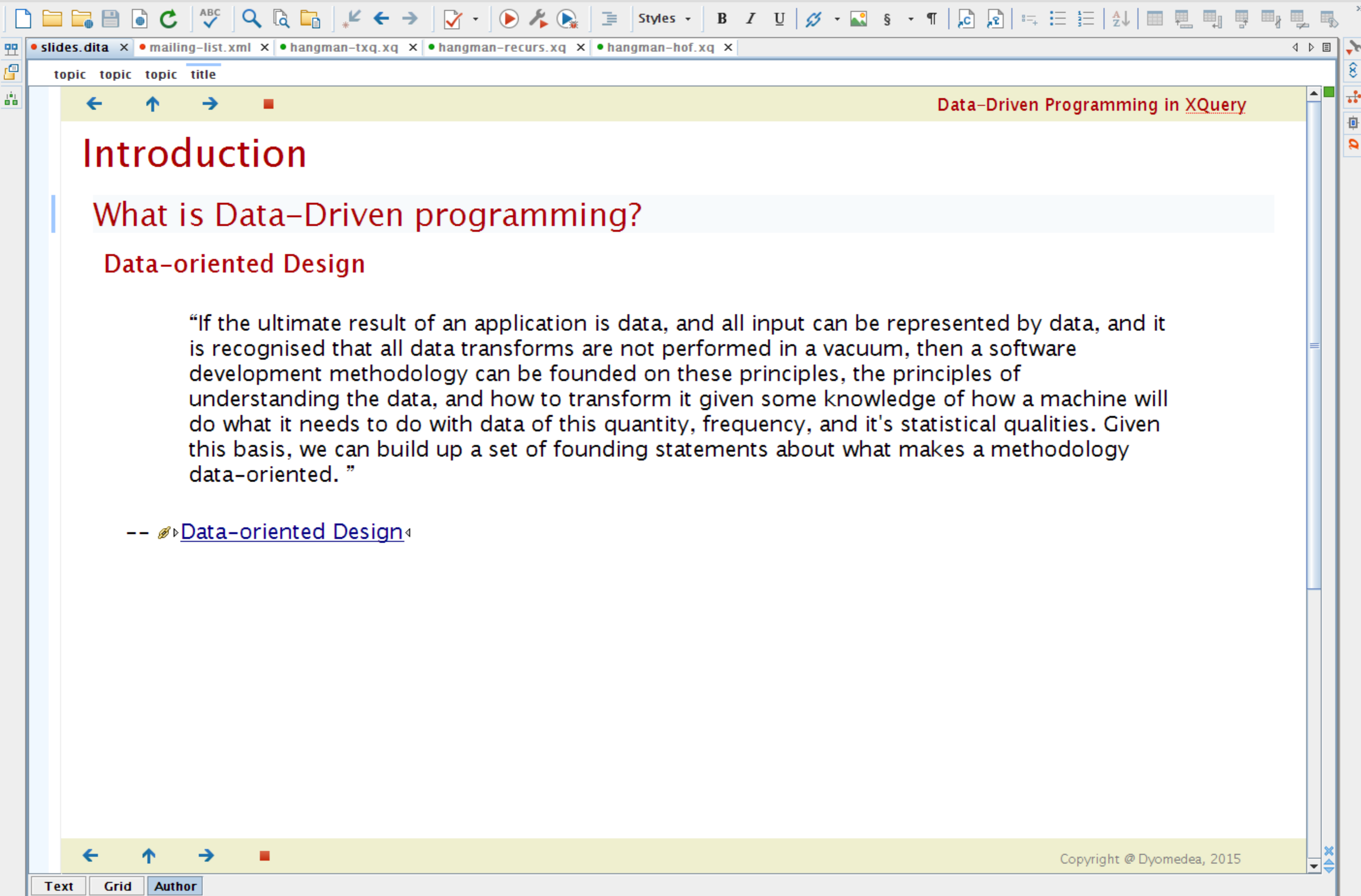
c2.com

“Data Driven Programs are programs which process data files whose contents cause the program to do something different. The extreme case is an interpreter and the interpretable program files. ”

-- [c2.com](#)

← ↑ → ■ Copyright © Dyomedea, 2015

Text Grid Author

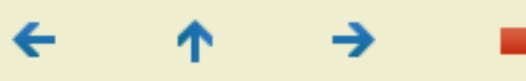


The screenshot shows the Oxygen XML Editor interface. The main content area displays a slide with the following structure:

- Navigation icons: back, forward, search, and a red square.
- Page title: Data-Driven Programming in XQuery
- Section header: Introduction
- Section header: What is Data-Driven programming?
- Section header: Data-oriented Design
- Text block: "If the ultimate result of an application is data, and all input can be represented by data, and it is recognised that all data transforms are not performed in a vacuum, then a software development methodology can be founded on these principles, the principles of understanding the data, and how to transform it given some knowledge of how a machine will do what it needs to do with data of this quantity, frequency, and it's statistical qualities. Given this basis, we can build up a set of founding statements about what makes a methodology data-oriented."
- Text block: -- [Data-oriented Design](#)
- Page footer: Copyright © Dyomedea, 2015

At the bottom of the editor, there are tabs for 'Text', 'Grid', and 'Author', with 'Author' currently selected. The status bar at the very bottom shows the file path, line number (U+0000), and a notification for 6 new messages.

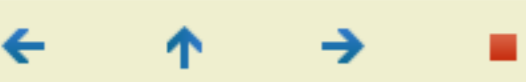
topic topic



Data-Driven Programming in XQuery

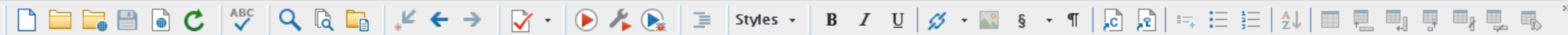
Introduction

Examples



Copyright © Dyomedea, 2015

Text Grid Author



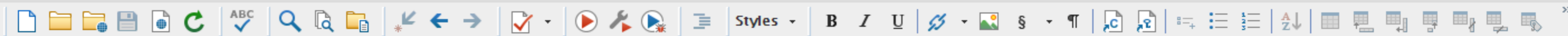
Introduction

Examples

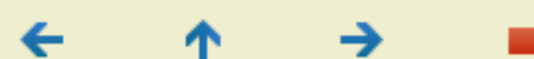
Hello world

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("Hello, World");  
    }  
}
```

- This hello world is 0% data-driven.
- The JVM that runs this program is 100% data-driven.



topic topic topic title



Data-Driven Programming in XQuery

Introduction

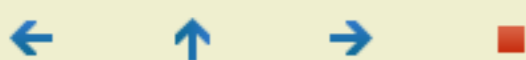
Examples

A backup utility

```
<backups>
  <task name="apache-norm">
    <postgres name="test" user="apache" dump="dbtestdump.sql"/>
    <mysql name="wikixmlfr" user="wikixmlfr" password="XXXX"
      dump="/home/apache/wikixmlfrdump.sql"/>
    <user name="apache" host="gwnormandy.dyomedea.com">
      <exclude>xmlfr/</exclude>
      <exclude>vdv/albums/</exclude>
    </user>
  </task>

  <!-- ... -->
</backups>
```

[backup.xml](#)



Copyright © Dyomedea, 2015

The screenshot shows the Oxygen XML Editor interface. The main content area displays a slide with a yellow header bar containing navigation icons and the title "Data-Driven Programming in XQuery". Below the header, the word "Introduction" is written in a large red font. Underneath, the word "Examples" is written in a smaller red font. A sub-section titled "A mailing list manager" is followed by a bulleted list of four items. The bottom of the slide features a yellow footer bar with navigation icons and the text "Copyright © Dyomedea, 2015". The editor's toolbar and menu bar are visible at the top, and the status bar at the bottom shows the file path, cursor position (U+0000), and a notification for 6 new messages.

topic topic topic title

← ↑ → ■ Data-Driven Programming in XQuery

Introduction

Examples

A mailing list manager

- Used by the XML Guild
- Powered by TreeBind
- XML elements are mapped to classes and methods that wrap IMAP and SMTP actions.
- See [mailing-list.xml](#).

← ↑ → ■ Copyright © Dyomedea, 2015

Text Grid Author



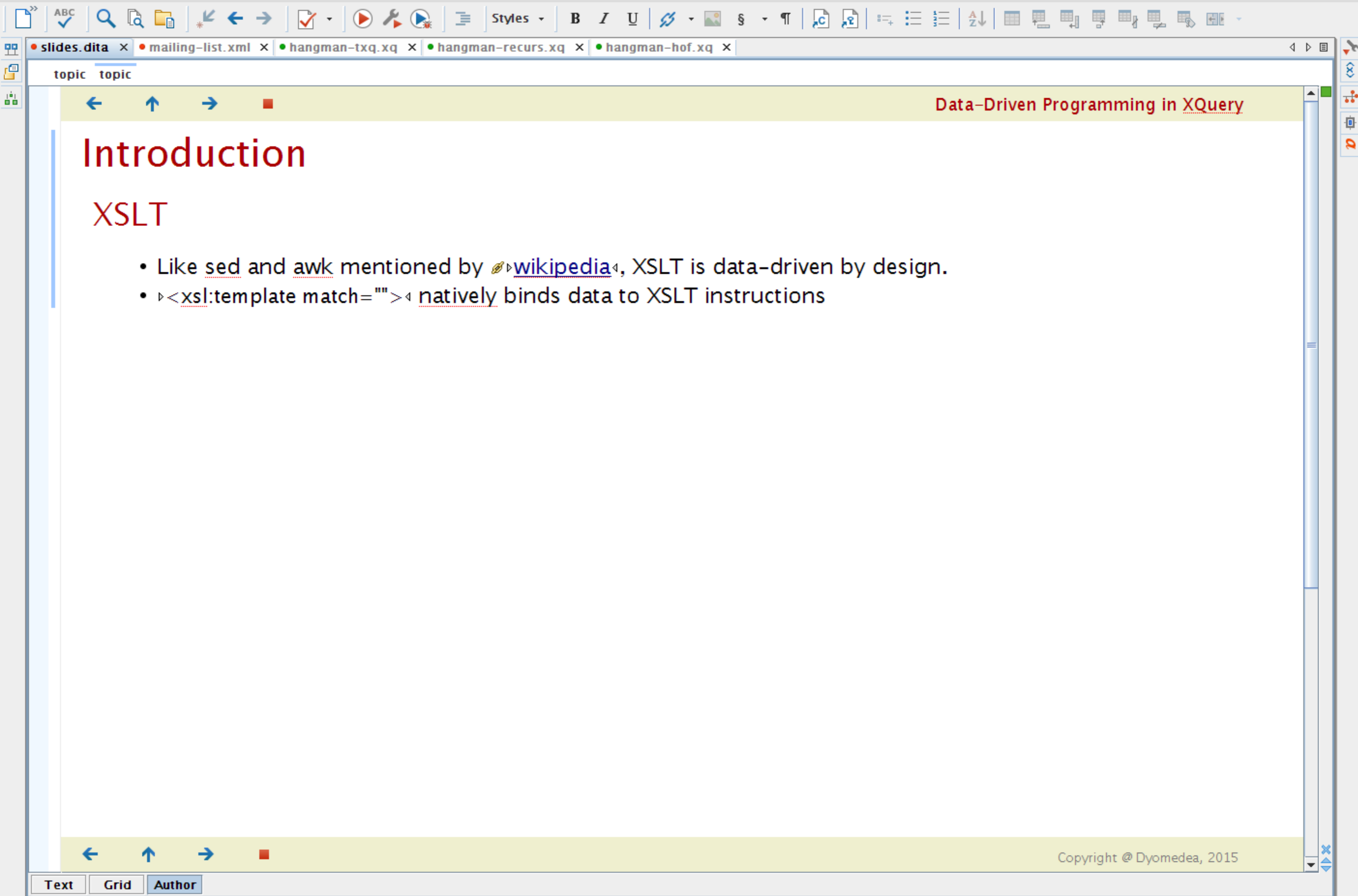
slides.dita x mailing-list.xml x hangman-txq.xq x hangman-recurs.xq x hangman-hof.xq x

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <listManager>
3   <server>localhost</server>
4   <storeType>imap</storeType>
5   <user>...</user>
6   <password>...</password>
7   <port>143</port>
8   <folderManager>
9     <folder>INBOX</folder>
10    <messageHandler>
11      <ifEither>
12        <ifIsRecipient>info@xmlguild.org</ifIsRecipient>
13        <ifIsRecipient>info@thexmlguild.org</ifIsRecipient>
14        <ifIsRecipient>info@xmlguild.info</ifIsRecipient>
15        <ifIsRecipient>info@xml-guild.org</ifIsRecipient>
16        <ifIsRecipient>info@xml-guild.com</ifIsRecipient>
17      </ifEither>
18    <sendToList>
19      <subjectPrefix>[the XML Guild]</subjectPrefix>
20    <footer><![CDATA[
21  --
22  The XML Guild
23  where you find established XML experts . . .
24  http://xmlguild.org/
25  info@xmlguild.org
26  ]]></footer>
27    <recipient>vdv@dyomedeia.com</recipient>
28    <recipient>...</recipient>
29    <recipient><!-- other recipients removed --></recipient>
30    <envelopeFrom>info-bounce@xmlguild.org</envelopeFrom>
31    <header name="Precedence">List</header>
32    <header name="List-Id">&lt;info.xmlguild.org&gt;</header>
33    <header name="List-Post">&lt;mailto:info@xmlguild.org&gt;</header>
34    <server>localhost</server>
35    <user>...</user>
36    <archive>archive</archive>
37  </sendToList>
38  <moveTo>done</moveTo>
39 </messageHandler>
40 <messageHandler>
41   <moveTo>unparsed</moveTo>
42 </messageHandler>

```

Text Grid Author



topic topic

← ↑ → ■ Data-Driven Programming in XQuery

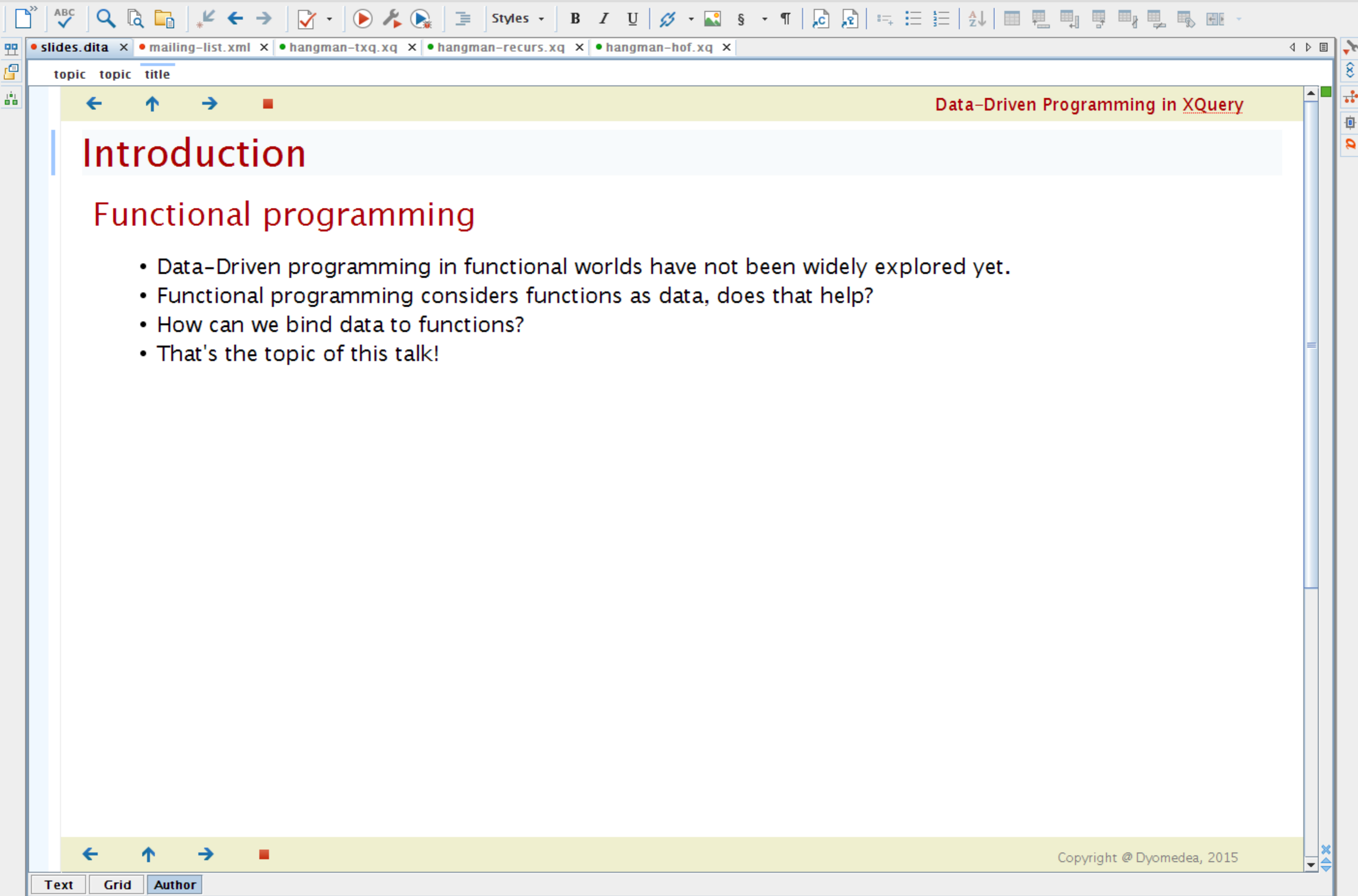
Introduction

XSLT

- Like [sed](#) and [awk](#) mentioned by [wikipedia](#), XSLT is data-driven by design.
- `<xsl:template match="">` natively binds data to XSLT instructions

← ↑ → ■ Copyright © Dyomedea, 2015

Text Grid Author



topic topic title

← ↑ → ■ Data-Driven Programming in XQuery

Introduction

Functional programming

- Data-Driven programming in functional worlds have not been widely explored yet.
- Functional programming considers functions as data, does that help?
- How can we bind data to functions?
- That's the topic of this talk!

← ↑ → ■ Copyright © Dyomedea, 2015

Text Grid Author

The screenshot shows the Oxygen XML Editor interface. The main content area displays a slide with a yellow header bar containing navigation icons (back, forward, search) and the title "Data-Driven Programming in XQuery". Below the header, the slide content is divided into two sections: "Introduction" and "Conclusion". The "Conclusion" section contains a bulleted list of points. At the bottom of the slide, there is a footer with navigation icons and the text "Copyright © Dyomedea, 2015". The editor's status bar at the very bottom shows the file path, cursor position (U+0000), and a notification for 6 new messages.

topic topic title

← ↑ → ■ Data-Driven Programming in XQuery

Introduction

Conclusion

- It's a paradigm
- Programming languages features and libraries can facilitate data-driven programming
- Data binding libraries are helpful to do data-driven programming in object oriented worlds
- XSLT is data-driven by design
- What about functional programming?

← ↑ → ■ Copyright © Dyomedea, 2015

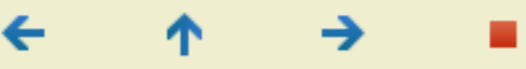
Text Grid Author

topic



Data-Driven Programming in XQuery

Functional XQuery



Copyright © Dyomedea, 2015

Text Grid Author

Functional XQuery

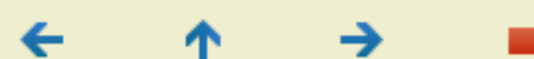
Hangman

- Rules borrowed from [Wikipedia](#).
- State persisted in XML ([hangman.xml](#)).

```
<hangman status="in-progress" misses="Z">
  <word>
    <letter guessed="true">H</letter>
    <letter guessed="true">A</letter>
    <letter guessed="true">N</letter>
    <letter guessed="false">G</letter>
    <letter guessed="true">M</letter>
    <letter guessed="true">A</letter>
    <letter guessed="true">N</letter>
  </word>
  <display>
    <head shown="true"/>
    <body shown="true"/>
    <right_arm shown="true"/>
    <left_arm shown="true"/>
    <right_leg shown="false"/>
    <left_leg shown="false"/>
  </display>
</hangman>
```



topic topic title



Data-Driven Programming in XQuery

Functional XQuery

In Python

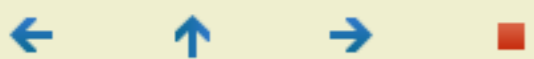
[Reference implementation](#) in Python using the XML-Driven classes presented at OSCON 2004.

```
# Class Letter: individual letters to be guessed
class Letter(XmlObject.XmlObjectElement):
```

```
    def addGuess(self, guess):
        if self._value() == guess:
            self.guessed._set('true')
            return True
        else:
            return False
```

```
    def isGuessed(self):
        return self.guessed._value() == 'true'
```

```
XmlObject.XmlObjectElement_letter = Letter
```



Copyright © Dyomedea, 2015

The screenshot shows the Oxygen XML Editor interface. The top menu bar includes File, Edit, Find, Project, Options, Tools, DITA, Document, Window, and Help. Below the menu is a toolbar with various icons for editing and navigation. The main content area displays a slide with a yellow header bar containing navigation arrows and the text "Data-Driven Programming in XQuery". The slide title is "Functional XQuery" in large red font, followed by "In XSLT" in smaller red font. A link to "hangman.xsl" is shown with a pencil icon. Below the link is a code block containing XSLT template code. At the bottom of the slide, there are navigation arrows and the text "Copyright © Dyomedea, 2015". The bottom status bar shows the file path, line number (U+0000), and a notification for 6 new messages.

topic topic title

← ↑ → ■ Data-Driven Programming in XQuery

Functional XQuery

In XSLT

[hangman.xsl](#)

```
<xsl:template match="letter[.= $guess]/@guessed">  
  <xsl:attribute name="guessed">true</xsl:attribute>  
</xsl:template>
```

← ↑ → ■ Copyright © Dyomedea, 2015

Text Grid Author

topic topic title

Navigation icons: back, up, down, stop

Data-Driven Programming in XQuery

Functional XQuery

transform.xq



Navigation icons: back, up, down, stop

Copyright © Dyomedea, 2015

Text Grid Author

slides.dita × mailing-list.xml × hangman-txq.xq × hangman-recurs.xq × hangman-hof.xq × hangman.py ×

topic topic title

← ↑ → ■ Data-Driven Programming in XQuery

Functional XQuery

XQuery with transform.xq

▶ [hangman-txq.xq](#) ◀ using ▶ [transform.xq](#) ◀.

```

(: Guesses :)
tfm:rule('letter/@guessed', function($mode, $node) {
  attribute {xs:QName(name($node))} {
    if ($node/.. = $guess)
    then 'true'
    else $node
  }
}),

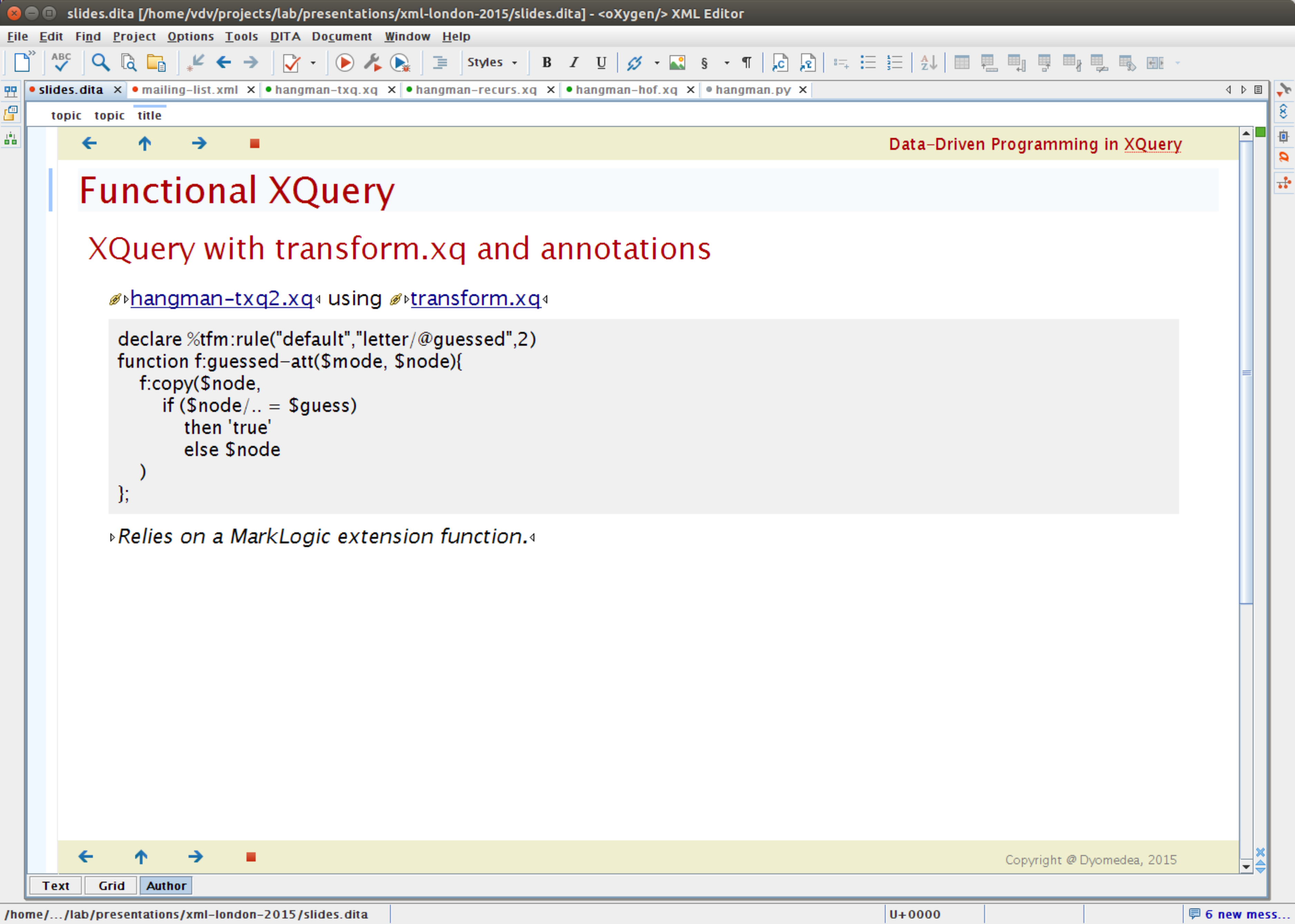
```

▶ *Doesn't work on Saxon for whatever reason.* ◀

← ↑ → ■

Copyright © Dyomedea, 2015

Text Grid Author



The screenshot shows the Oxygen XML Editor interface. The top menu bar includes File, Edit, Find, Project, Options, Tools, DITA, Document, Window, and Help. Below the menu is a toolbar with various icons for editing and navigation. The main window displays a slide with a yellow header bar containing navigation arrows and the text "Data-Driven Programming in XQuery". The slide content includes a title "Functional XQuery" and a subtitle "XQuery with transform.xq and annotations". A code block shows XQuery code for a function named f:guessed-att. Below the code, a note states "Relies on a MarkLogic extension function." The footer of the slide contains navigation arrows and the text "Copyright © Dyomedea, 2015". The bottom status bar shows the file path, page number "U+0000", and a notification for "6 new mess...".

topic topic title

← ↑ → ■ Data-Driven Programming in XQuery

Functional XQuery

XQuery with transform.xq and annotations

▸ [hangman-txq2.xq](#) using ▸ [transform.xq](#)

```
declare %tfm:rule("default", "letter/@guessed", 2)
function f:guessed-att($mode, $node){
  f:copy($node,
    if ($node/.. = $guess)
      then 'true'
      else $node
  )
};
```

▸ Relies on a MarkLogic extension function.

← ↑ → ■ Copyright © Dyomedea, 2015

Text Grid Author

The screenshot shows the Oxygen XML Editor interface. The top menu bar includes File, Edit, Find, Project, Options, Tools, DITA, Document, Window, and Help. Below the menu is a toolbar with various icons for editing and navigation. The main window displays a slide with a yellow header bar containing navigation arrows and the text "Data-Driven Programming in XQuery". The slide content features a large red heading "Functional XQuery" and a sub-heading "What transform.xq brings us". Below this is a bulleted list of three items. At the bottom of the slide, there is a footer with navigation arrows and the text "Copyright © Dyomedea, 2015". The editor's status bar at the very bottom shows the file path, cursor position (U+0000), and a notification for 6 new messages.

topic topic title

← ↑ → ■ Data-Driven Programming in XQuery

Functional XQuery

What transform.xq brings us

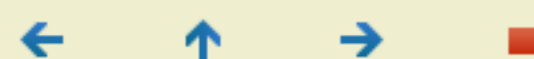
- A parser for a subset of XSLT match patterns (converted to XQuery functions).
- A way to bind XQuery functions (playing the role of XSLT templates) to match patterns.
- *▶ What could we do without this outstanding library?*

← ↑ → ■ Copyright © Dyomedea, 2015

Text Grid Author



topic topic title



Data-Driven Programming in XQuery

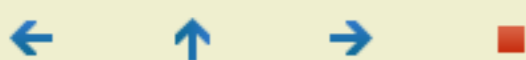
Functional XQuery

Classical XQuery

▸ [hangman-recurs.xq](#)

```
...
  else if ($node instance of attribute(guessed)) then
    f:copy($node,
      if ($node/.. = $guess)
        then 'true'
        else $node
    )
  )
...
```

▸ *Could be much worse!*

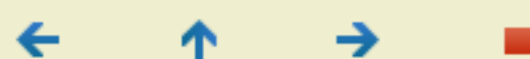


Copyright © Dyomedea, 2015

Text Grid Author



topic topic title



Data-Driven Programming in XQuery

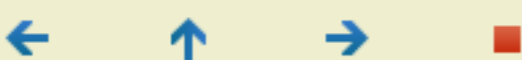
Functional XQuery

With higher-order functions

▸ [hangman-hof.xq](#)

```
...  
(: @guessed attributes :)  
f:rule(  
  function($node as node()) as xs:boolean {  
    $node instance of attribute(guessed)  
  }, function($node as node(), $transform as function(*)) as node() {  
    f:copy($node, if ($node/.. = $guess)  
      then 'true'  
      else $node)  
  },  
  ...  
)
```

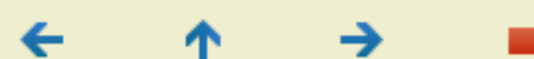
▸ *Similar to transform.xq without match patterns.*



Copyright © Dyomedea, 2015



topic topic title



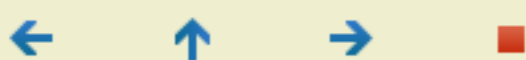
Data-Driven Programming in XQuery

Functional XQuery

Missing maps in XQuery 3.0

Dirty hack used by [hangman-hof.xq](#) to simulate maps:

```
(: borrowed from transform.xq :)  
declare function f:rule(  
  $predicate as (function(node()) as xs:boolean),  
  $action as (function(node(), function(*)) as node(?)  
  as function(*) {  
  
  function($k as xs:string) {  
    switch($k)  
      case 'predicate' return $predicate  
      case 'action' return $action  
      default return ()  
  }  
};
```



Copyright © Dyomedea, 2015

Text Grid Author

Functional XQuery

The magic of higher-order functions

The `f:transform()` function in [hangman-hof.xq](#):

```
(: Some high order functions magic :)  
declare function f:transform($rules as function(*)*) as function(*) {  
  function($node as node(), $transform as function(*)) as node()? {  
    if (head($rules)('predicate')($node))  
      then head($rules)('action')($node, $transform)  
    else if (exists(tail($rules)))  
      then f:transform(tail($rules))($node, $transform)  
    else ()  
  }  
};
```

The screenshot shows the Oxygen XML Editor interface. The main content area displays a slide with a light blue background. At the top of the slide, there is a navigation bar with left and right arrows, an up arrow, and a red square. The title of the slide is "Data-Driven Programming in XQuery" in red text. Below the title, the main heading "Functional XQuery" is displayed in a large, bold, red font. Underneath, the word "Conclusion" is written in a smaller red font. A bulleted list follows, containing four items. At the bottom of the slide, there is another navigation bar with left and right arrows, an up arrow, and a red square. The footer of the slide reads "Copyright © Dyomedea, 2015". The editor's status bar at the bottom shows the path "/home/.../lab/presentations/xml-london-2015/slides.dita", the cursor position "U+0000", and a notification for "6 new mess...".

topic topic title

← ↑ → ■ Data-Driven Programming in XQuery

Functional XQuery

Conclusion

- Higher-order functions do facilitate data-driven programming in XQuery.
- Maps are missed in XQuery 3.0 but can be simulated.
- transform.xq gives you all you need but you can also easily write your own mechanism.
- Good old if/then/else and recursion isn't that awful after all!

← ↑ → ■ Copyright © Dyomedea, 2015

Text Grid Author

The screenshot shows the Oxygen XML Editor interface. The main content area displays a presentation slide with a light blue background. At the top right of the slide, the text "Data-Driven Programming in XQuery" is visible. The main title of the slide is "Functional XQuery" in a large, bold, red font. Below the title is the section "Acknowledgments" in a smaller red font. Under "Acknowledgments", the text "Thanks to :" is followed by a bulleted list of names and acknowledgments. The slide has a light green header and footer bar. The footer bar contains navigation icons on the left and the text "Copyright © Dyomedea, 2015" on the right. The editor's toolbar and menu are visible at the top, and the status bar is at the bottom.

topic topic title

← ↑ → ■ Data-Driven Programming in XQuery

Functional XQuery

Acknowledgments

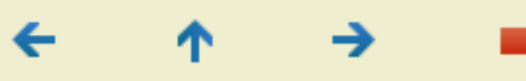
Thanks to :

- Tony Graham, Patrick Durusau and my three anonymous reviewers for their reviews, comments, corrections, questions and suggestions.
- John Snelson for his very inspiring transform.xq.
- George Bina for pushing me into dita.
- XML London for this gathering.

← ↑ → ■ Copyright © Dyomedea, 2015

Text Grid Author

topic topic title



Data-Driven Programming in XQuery

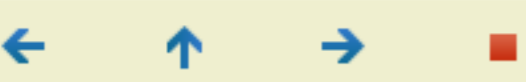
Functional XQuery

Questions

vdv@dyomedea.com

<http://vdv.re/xmlLondon2015>

@evlist



Copyright © Dyomedea, 2015

Text Grid Author