An Introduction to XML Databases: Creating a TEI-Based Website with the eXist-db XML Database

Joseph Wicentowski, Ph.D.
U.S. Department of State

July 2011

Goals

By the end of this workshop you will know:

1. about a flexible set of technologies (*XPath*, *XQuery*, and *native XML databases*) for answering questions about and publishing your TEI documents
2. about *eXist-db*: a free, open-source native XML database
3. how to install and use eXist-db and oXygen to query and create a website out of your TEI works

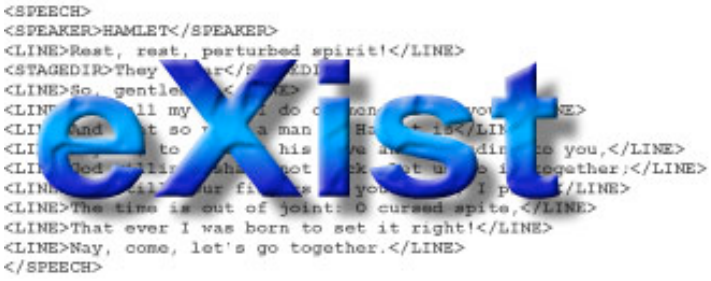
Completing the TEI Toolset

By now you've decided on:

- TEI: Your data format
- oXygen: Your XML editing Swiss army knife
  - Edit / author documents
  - Traverse documents with XPath tools
  - Transform documents with TEI XSLT
- So what's missing?
  - An *easy* way to analyze and ask questions *across any or all* of your TEI documents
  - A search engine and database for querying your content; think of your TEI content as a database
  - A web server for publishing your TEI documents

There are many tools that might help you in each of these respects, but eXist-db fills all these gaps in a very elegant way.

What is eXist-db?



eXist-db Logo

- a native XML database
- a free, open source product
- a community-driven project
- TEI-friendly and popular among TEI users (and those with lots of XML)
- Integrates very nicely with oXygen

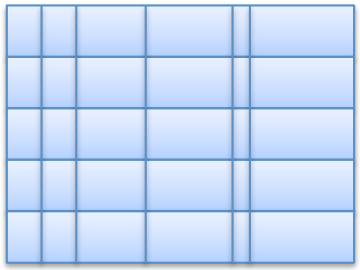
Brief Case Study

history.state.gov

- Homepage of Office of the Historian (U.S. Department of State)
- Launched January 2009, built 100% on eXist-db
- TEI-based digital edition of *Foreign Relations of the United States*, the official documentary record of U.S. foreign relations
- 140+ volumes (and growing), containing 50,000+ primary source archival documents
- 5-10 MB TEI file for each volume (total: 2 GB XML + 10 GB page images)
- Rapid full text search, research tools
- Toolset:
  - oXygen for XML and XQuery authoring
  - eXist-db for website development and production server
  - An eXist-db-powered web-based *content management system* for editing metadata as well as and editing and annotating TEI

Why a Native XML Database?

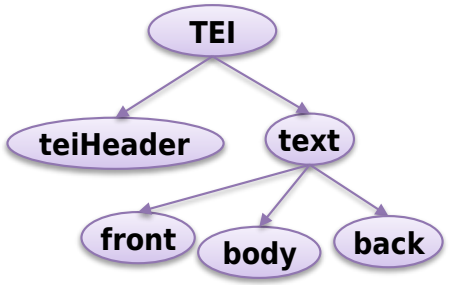
Tables



Example data
Simple lists
Excel spreadsheets
Relational databases

Query Language
SQL

Trees



Example data
HTML
XML
Taxonomies

Query Language
XPath/XQuery

Tables vs. Trees (Credit: Dan McCreary)

- Relational Database: a collection of tables (rows and columns) for storing data and relationships - well-suited to tabular data
- Native XML Database: uses XML documents as the fundamental unit of storage and XML for the internal data model - well-suited to complex, nested, 'semi-structured' documents like TEI

eXist-db's flavor of native XML database

- Easy to download, install, and get started (Mac, PC, Linux)
- Just drag and drop XML into the database (via WebDAV, etc.)
- Supports *XQuery*, the W3C XML Query Language, for querying XML
- eXist-db automatically indexes the entire XML structure, so structural (path) queries are *much* faster than searching files on the filesystem
- In addition, eXist-db's customizable indexing system let you create fulltext search engines out of any TEI elements & attributes you want, with Google-style query syntax
- Query your documents quickly in the *XQuery Sandbox*
- Save your queries into eXist-db, making them into *web pages*
- Entire web applications can be written in XQuery (+ XSLT, XHTML, CSS and Javascript)
- Supports XPath, XSLT, XQuery Update, and Full Text Search (leverages Lucene); flexible URL rewriting

Getting eXist-db

- Download from exist-db.org
- (Windows users note: Requires Java *JDK* be installed first)
- Installing eXist actually puts all of the exist-db.org resources on your computer
  - Searchable Documentation
  - Searchable Function Library
  - XQuery Sandbox (a real gem for quick queries)
  - Demos (get ideas, see examples of XQuery in action)
- Before long, you'll have your TEI files stored in the eXist database, and you'll be writing queries in the Sandbox and in oXygen

XPath and XQuery in ~10 Minutes

Understanding XPath and XQuery is easy if you understand some basics about XML — and you already do, since you use TEI!

- *Elements* and their *namespaces*
- *Attributes*
- *Text*

These are all types of XML *nodes*. And from any node in an XML document you can get to any other node, by traversing XPath *axes*.

XPath

XPath is a language for addressing parts of an XML document (although it's *not* a full programming language.) It's common to both XSLT and XQuery.

An XPath expression contains one or more "location steps", separated by slashes. Each location step has the following unabbreviated form: `axis-name::node-test[predicate]`

The most common XPath axes have abbreviated forms: *child* (whose shorthand is `/`), *parent* (`../`), *descendant-or-self::node()* (`//`), *self* (`.`), and attribute (`@`) are the most common:

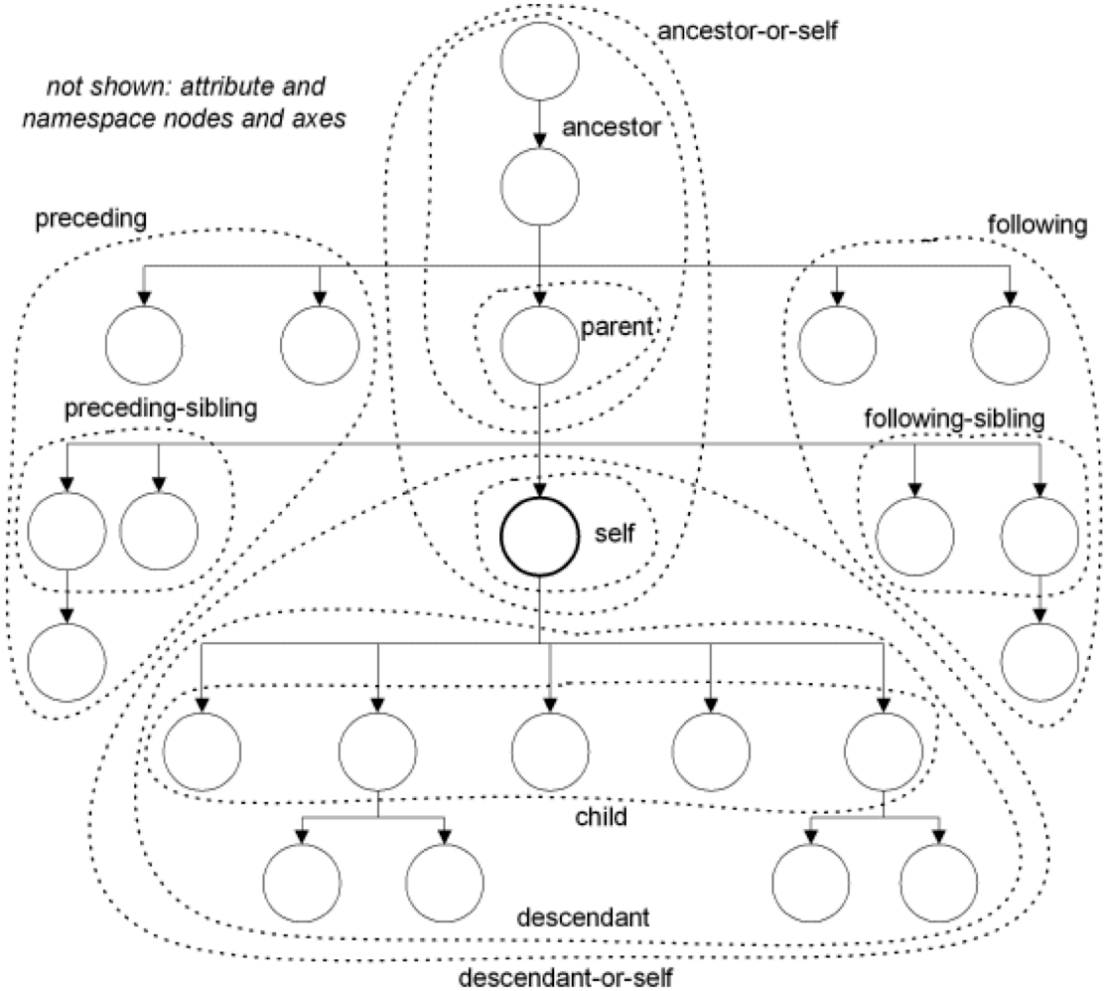
- `div/head` returns all of a div's child head elements

Predicates, expressions encased in square brackets, restrict the results to those that with match conditions:

- `//div[@type eq 'cartoon']` returns a sequence of the div elements whose type attribute equals ‘cartoon’
- `//persName[. eq 'Cummings']` returns a sequence of the persName elements whose value is ‘Cummings’

XPath Axes

Including these most common axes there are 13 total XPath axes:



XPath Axes (Credit: George Hernandez)

Consider printing this image for reference.

Items and Sequences

A key concept in XPath are *items* and *sequences*. Examples of items:

- `'a'` (a string)
- `1` (an integer)
- `<p>Hi!</p>` (an element containing text)
- `<TEI/>` (a root element of a TEI document)
- `doc('/db/punch/data/1914-07-01.xml')` (an entire document stored in the eXist-db database)

Sequences (comma-separated, parentheses-encased lists; or XQuery/XPath expressions)

- `('a', 'b', 'c')`
- `(1, 2, 3)`
- `(<p/>, <persName/>, <list><item/></list>)`
- `('a', 1, <p/>)`
- `collection('/db/punch')//tei:l[contains(., 'love')]`

How XPath *Expressions* operate on *Items*

- Arithmetic expressions & functions
  - 1 + 2
  - *avg(* (10,100,1000) *)*
- Text & String functions
  - *concat(*'a', 'b'*)*
  - *substring-before(*'Text Encoding Initiative', 'Init'*)*
- Other functions
  - *count*(1, 'a', <p/>)
  - <hi rend="italic"/>/@rend/*string()*
  - *current-date()*
- Filter your sequence with *predicates*
  - ('Lou', 'James', 'Sebastian')*[starts-with(., 'J')]*
  - (1, 2, 3)*[. < 3]*

XQuery

- XQuery builds on XPath, and is an easy-to-learn, flexible, and powerful language for querying XML and transforming it. By storing your TEI in eXist-db, you can query across your entire TEI corpus. You can also benefit from eXist-db's XQuery Update support, which allows you to alter XML in the database.
- XQuery supports many expressions:
  - Literals (string literals like `'a'` and numeric literals like `1`)
  - Variables (`$foo`), to which you bind values
  - Functions, either built-in like `substring-before('hello', 'l')` or your own
  - Comments `(: this is a comment! :)`
  - Comparisons: =, <, >, eq
  - Conditionals: if then else
  - FLWOR Expressions: the core of XQuery

XQuery FLWOR Expressions

Unique to XQuery, *FLWOR* (pronounced 'flower') Expressions give you more control over your queries than XPath alone.

'FLWOR' stands for:

- for: iterate through a sequence, assigning each item to a *variable* ($ + a name of your choosing starting, e.g. $people)
- let: name a sequence, assigning the whole sequence a variable
- where: filter a sequence (optional)
- order by: order a sequence (optional)
- return: return the resulting sequence (required)

FLWOR expressions are great for ordering your results, and for queries that are more complex than XPath allows

Example FLWOR Expressions

- ```
  for $item in ('c', 'b', 'a')
  order by $item
  return $item
  ```
  - Returns ('a', 'b', 'c')
- ```
  let $people := ('Lou', 'Sebastian', 'James')
  for $person in $people
  let $greeting := concat('Hello, ', $person)
  return $greeting
  ```
  - Returns ('Hello, Lou', 'Hello, Sebastian', 'Hello, James')
- ```
  for $role in collection('/db/punch/data')//tei:role
  order by $role
  return $role
  ```
  - Returns all role elements in the Punch collection in (implicitly) alphabetical order


  ```

How to Alternate between XML and XQuery in your queries

Soon you will be writing more complex queries that nest XQuery expressions inside of XML. For example, you may write a table of contents that displays chapter headings, and a list of section headings inside this.

How to alternate between XML and XQuery? Curly braces {} !

```
<ul>
    {
    for $head in $div/tei:head
    return
    <li>
        {
        $head/text()
        }
    </li>
    }
</ul>
```

Using curly braces

That's the core of XQuery in ~10 minutes!

TEI, eXist-db, oXygen, and XQuery

A typical set of steps for querying and developing TEI webpages with eXist-db

- Step 1: Get your TEI into eXist-db
- Step 2: Browse/edit your TEI with oXygen through the Database Explorer
- Step 3: Write simple XQueries in the XQuery Sandbox
- Step 4: Move to oXygen for turning your XQueries into web pages

Step 1: Getting your TEI into eXist-db

- There are several ways!
- In Windows XP, set up a WebDAV connection to eXist: go to My Network > Add Network Places > http://localhost:8080/exist/webdav/db. Provide your eXist-db username and password. Then just drag your files from the desktop into the eXist-db WebDAV window.
- For WebDAV on other platforms, see eXist-db's WebDAV documentation.
- Or use eXist's Java-based admin client.
- Or use oXygen 12.2+'s Database Explorer > Import Files (or Import Folders).

Step 2a: Browse/edit your TEI with oXygen through the Database Explorer

- Open oXygen's *Data Source Explorer* via Window > Show View > Data Source Explorer
- The Data Source Explorer window will open. Click on the yellow gear icon above "Connections."
- Under Data Sources, click on New.
  - Name the new data source as "eXist Data Source"
  - Select eXist from the "Type" dropdown menu
  - Add 5 key files from your eXist-db installation directory: (1) exist.jar from the main directory, and from lib/core, (2) ws-commons-1.0.2.jar, (3) xmldb.jar, (4) xmlrpc-client-3.1.2.jar, (5) xmlrpc-common-3.1.2.jar.
- Under Connections, click on New.
  - Select your eXist-db data source
  - Name the connection "eXist-db on localhost 8080"
  - Change `<host/>` to "localhost" (delete the brackets)
  - Enter "admin" for username, and your eXist-db admin password. Click OK.

Step 2b: Tell oXygen to use eXist-db to validate XQuery

By telling oXygen to use eXist-db to validate XQuery, you can get feedback from eXist-db about any errors in the XQueries that you're writing in oXygen:

- Under Window > Preferences > XQuery > XQuery Validate With, select "eXist-db on localhost 8080"
- Click OK.

Now, with these steps done, oXygen is fully configured to both browse eXist-db's database and use it to provide feedback on your XQuery work.

If the "Data Source Explorer" windows is not open in oXygen, open it via Window > Show View > Data Source Explorer, and "pin" it so it stays open.

Step 3: Write simple XQueries in eXist-db's XQuery Sandbox

oXygen's XPath/XQuery functions let us query a single document at a time, but the eXist-db XQuery Sandbox lets us query our entire collection of TEI files:

- Go to http://localhost:8080/exist/sandbox and enter these queries:
- ```
  declare namespace tei = "http://www.tei-c.org/ns/1.0";
  count( collection('/db/punch/data')/tei:TEI )
  ```
  -> Returns the count of TEI files in the Punch collection
- ```
  declare namespace tei = "http://www.tei-c.org/ns/1.0";
  collection('/db/punch/data')//tei:name
  ```
  -> Returns all TEI name elements in the Punch collection

Step 4: Move to oXygen for turning your XQueries into web pages

The Sandbox is a powerful tool for individual exploration. Once you've found queries that you want to turn into webpages, open oXygen to create *XQuery modules*.

- Select File > New > XQuery, and paste in your Sandbox query
- Notice how oXygen colors the XQuery and XML syntax appropriately
- Save the valid XQuery via File > Save to URL.
  - Enter "admin" for User and your eXist-db admin password; for Server URL, enter http://localhost:8080/exist/webdav/db/. Click on Browse to log in and browse eXist-db's collections. Click on the "punch" collection, causing the File URL to read: http://admin@localhost:8080/exist/webdav/db/punch/Untitled1.xquer Change "Untitled1.xquery" to "myquery.xq". Click OK.
- Open the query in your web browser at http://localhost:8080/exist/rest/db/punch/myquery.xq

Exercises: Before You Start

- Install eXist-db (from the course's H: drive), or download from exist-db.org
- Set up WebDAV and oXygen (as detailed above)
- Copy course files into eXist-db:
  - Copy the *index configuration files* in "files/db/system/config" into their corresponding location in eXist-db's database, in the "db/system/config" directory
  - Copy the "punch" directory in "files/db/punch" into eXist-db's root collection "/db", so you have "/db/punch"
- Now you're ready to begin the exercises.

Exercises

- Query your TEI files from eXist-db's sandbox, http://localhost:8080/exist/sandbox
  - Try querying Punch, and querying for elements you have worked with. See "Step 3" above for some examples.
  - Use predicates to filter your results, with the functions contains(), starts-with(), and distinct-values().
  - Use FLWOR expressions to order your results.
- Copy your queries into oXygen, save them to eXist-db, and call them from your web browser, e.g. save 'myquery.xq' into /db/punch/myquery.xq, and point your web browser to http://localhost:8080/exist/rest/db/punch/myquery.xq
- When you're ready to create a full website around the Punch data, open the sample Punch website, http://localhost:8080/exist/rest/db/punch/index.xq

Sample Punch Website

To understand how a website is assembled with XQuery in eXist-db, go to the sample Punch website in http://localhost:8080/exist/rest/db/punch/index.xq.

The XQuery files (.xq, .xqm files) themselves are extensively commented, so please open each file to read the comments and understand.

The sample actually contains 4 versions of a Punch website — the first very simple, and the last polished.

Each "version" of the site improves the presentation and usefulness of the site.

index.xq - Landing Page

TEI@Oxford: Punch eXist Tutorial Website

Welcome to the Punch eXist Tutorial Website, developed for the TEI@Oxford Summer Course in July 2010.

The tutorial demonstrates the development of a website around the Punch materials. Each "version" of the site improves the presentation and usefulness of the site:

- Punch v1: We will list the issues of Punch and create a link out of each issue. Click on the link to select an issue to view. The TEI is transformed into HTML using an imported XQuery module, "tei-to-html." But since we transform an entire issue, the page is very long and slow to load.
- Punch v2: To overcome the problem of slow loading and seeing too much on a page, we add a basic table of contents for each issue. We add a link to each entry in the table of contents, so we can view just the section we're interested in. But the table of contents is very basic, only showing the top-level divs.
- Punch v3: We improve on the table of contents, showing sub-entries. We do this by moving our table of contents code into an XQuery function, and calling it recursively when there are sub-entries.
- Punch v4: We build on these small steps and take a huge leap forward. Most noticeably, we improve the visual appearance of the site, by adding CSS. Under the hood, we have, centralizing all of our HTML into a template function, stored in a style module; this offloads much of the repetitive HTML code from each page, and allows us to add new pages quickly. Also, we add a full text search page, and add the search box to the header of every page. Because the search page needs to use much of the same code as the table of contents page, we store this code in a new "punch" module.

index.xq

http://localhost:8080/exist/rest/db/punch/index.xq

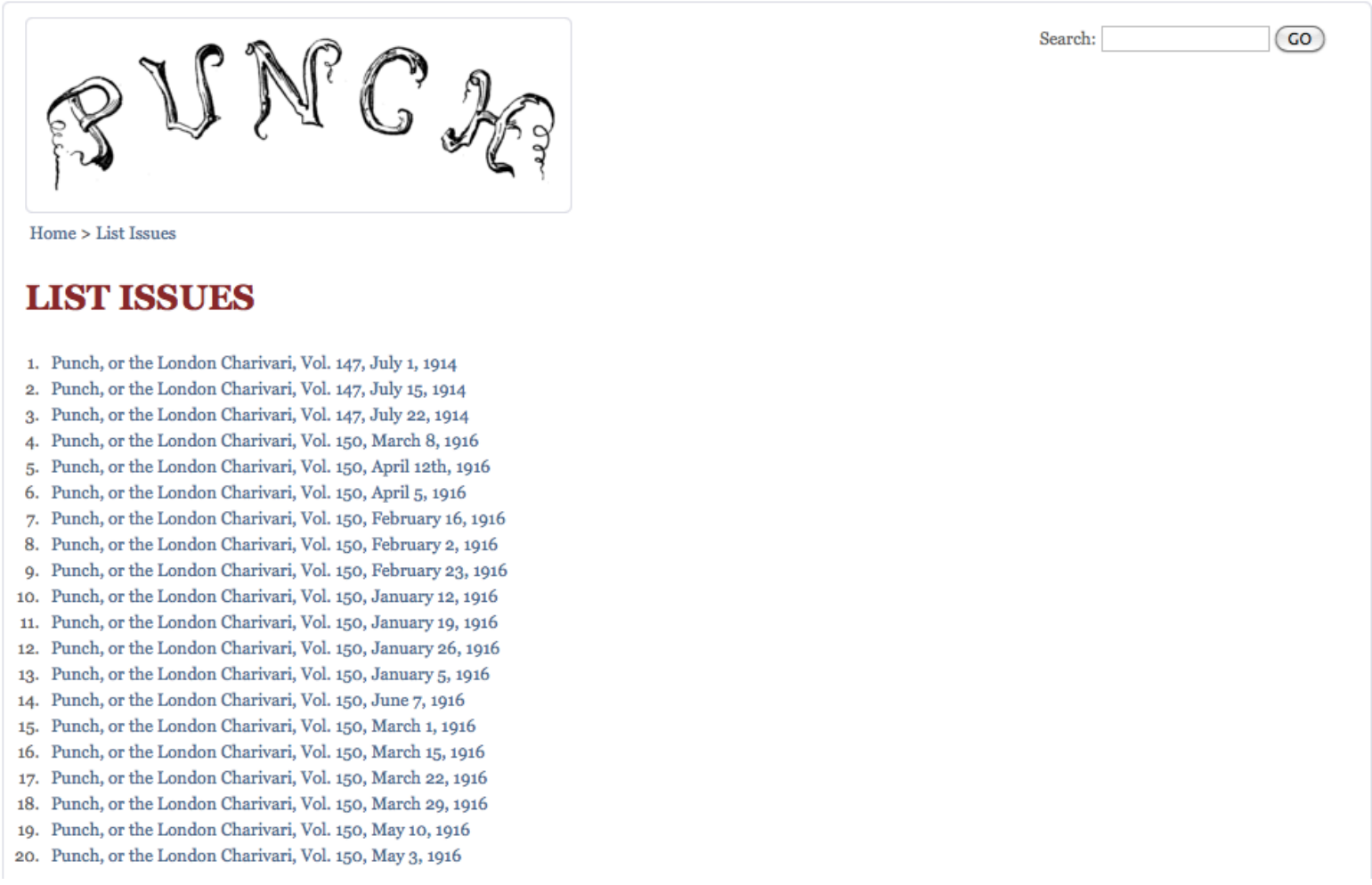
Version 1: List issues

Punch: Browse by Issue

1. Punch, or the London Charivari, Vol. 147, July 1, 1914
2. Punch, or the London Charivari, Vol. 147, July 15, 1914
3. Punch, or the London Charivari, Vol. 147, July 22, 1914
4. Punch, or the London Charivari, Vol. 150, March 8, 1916
5. Punch, or the London Charivari, Vol. 150, April 12th, 1916
6. Punch, or the London Charivari, Vol. 150, April 5, 1916
7. Punch, or the London Charivari, Vol. 150, February 16, 1916
8. Punch, or the London Charivari, Vol. 150, February 2, 1916
9. Punch, or the London Charivari, Vol. 150, February 23, 1916
10. Punch, or the London Charivari, Vol. 150, January 12, 1916
11. Punch, or the London Charivari, Vol. 150, January 19, 1916
12. Punch, or the London Charivari, Vol. 150, January 26, 1916
13. Punch, or the London Charivari, Vol. 150, January 5, 1916
14. Punch, or the London Charivari, Vol. 150, June 7, 1916
15. Punch, or the London Charivari, Vol. 150, March 1, 1916
16. Punch, or the London Charivari, Vol. 150, March 15, 1916
17. Punch, or the London Charivari, Vol. 150, March 22, 1916
18. Punch, or the London Charivari, Vol. 150, March 29, 1916
19. Punch, or the London Charivari, Vol. 150, May 10, 1916
20. Punch, or the London Charivari, Vol. 150, May 3, 1916

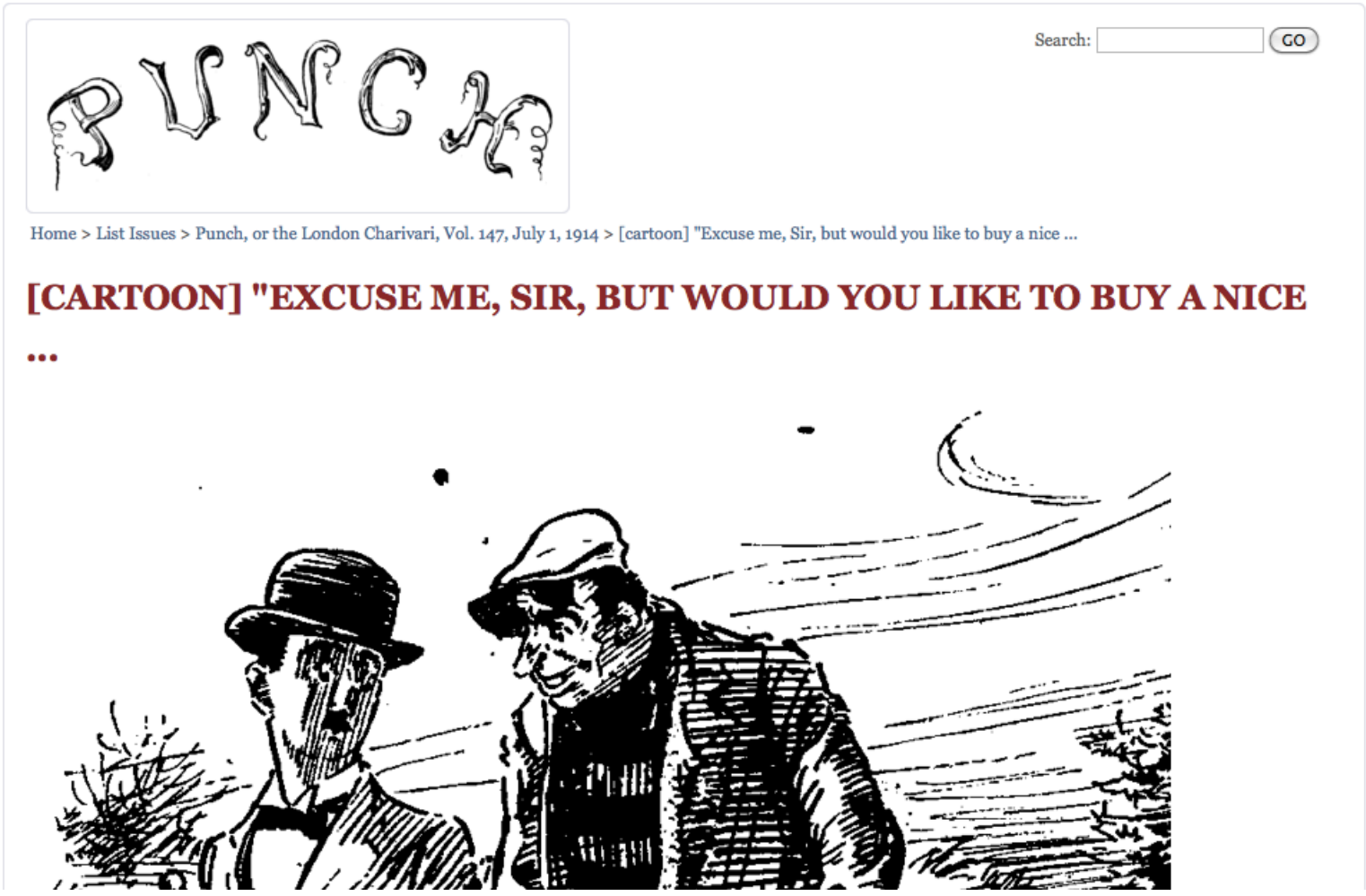
Version 1

Version 4: List issues



Version 4: List issues

Version 4: Show section



Version 4: Show section

Version 4: Search results

Search: GO

Home > Search

88 RESULTS FOR "LONDON"

Search: london GO

1. Punch, or the London Charivari, Vol. 150, March 8, 1916, British Frightfulness.

   *British Frightfulness. "A young woman was fried as a spy in **London** the other day."— Sunday Pictorial .*

2. Punch, or the London Charivari, Vol. 150, April 5, 1916, [snippet] "On the night of February 29th ten thousand women...

   *... February 29th ten thousand women marched through Unter Den **London** crying 'bread' and 'peace.'" Daily Gleaner (Kingston, Jam ...*

3. Punch, or the London Charivari, Vol. 150, January 5, 1916, A Long Turn.

   *... ning Miss Phyllis Bedells makes her final appearance at the **London** Empire, where she has danced without interruption for ni ...*

4. Punch, or the London Charivari, Vol. 150, May 3, 1916, NURSERY RHYMES OF LONDON TOWN.

   *NURSERY RHYMES OF **LONDON** TOWN. IX.— The Poultry and the Borough. The Fox ran to London Starving for his dinner; There he met the We ...*

Version 4: Search results

Resources

There are many resources for learning about eXist-db and XQuery, and for getting answers to your questions:

- All documentation for eXist-db: eXist-db Homepage http://exist-db.org
- Best book about XQuery: XQuery: Search Across a Variety of XML Data, by Priscilla Walmsley (O'Reilly 2007)
- Best website for learning XQuery and eXist-db: XQuery Wikibook http://en.wikibooks.org/wiki/XQuery
- Questions about using eXist-db and TEI - eXist-TEIXML mailing list https://lists.sourceforge.net/lists/listinfo/exist-teixml
- Questions about XQuery in general - XQuery-talk mailing list http://x-query.com/mailman/listinfo/talk
- Questions about eXist-db specificially - eXist-open mailing list https://lists.sourceforge.net/lists/listinfo/exist-open