

### Exercise #1: Creating a native-XML database under BaseX

Let us consider a book catalog stored in an XML document whose DTD is provided below.

```
<!-- books.dtd -->
<!ELEMENT catalog (book)*>
  <!ELEMENT book (author, title, genre, price, publish_date, description,
    onsale?)>
    <!ATTLIST book id ID #REQUIRED>
    <!ELEMENT author (#PCDATA)>
    <!ELEMENT title (#PCDATA)>
    <!ELEMENT genre (#PCDATA)>
    <!ELEMENT price (#PCDATA)>
    <!ELEMENT publish_date (#PCDATA)>
    <!ELEMENT description (#PCDATA)>
    <!ELEMENT onsale (EMPTY)>
```

1. Download the following XML document and save it locally.  
<http://eric.univ-lyon2.fr/~jdarmont/docs/books.xml>
2. Start BaseX from the Windows start menu or the JAR file available at the following URI.  
<http://basex.org/products/download/all-downloads/>
3. Load document `books.xml` into BaseX from the right-hand side window.

### Exercise #2: XPath queries

Create a new file in BaseX's right-hand side window to write a query. Execute the query by CTRL+ENTER or by clicking on the green arrow. Formulate the following queries with the help of path expressions (only! no FLWOR yet).

1. `book` elements of document `books.xml`. Specify the whole path to `book` elements.
2. Same question without explicitly stating the whole path to `book` elements.
3. Titles of all books.
4. Identifiers of all books.
5. Characteristics of the 4<sup>th</sup> book.
6. Titles of the 5 first books.
7. Titles of books belonging to genre "Computer".
8. Books of genre "Computer" costing less than 40 €.
9. Books that are currently on sale (i.e., that possess an `onsale` element).
10. Title and description of books by Eva Corets.
11. Average price of books.
12. Title of books published in 2001.

13. Most expensive book in the list.

### Exercise #3: More XPath queries!

Download the excerpt of a real-life auction dataset at the URI below, insert it in your collection and visualize its XML structure.

<http://eric.univ-lyon2.fr/~jdarmont/docs/auctions.xml>

Then, formulate the following queries with XPath.

1. Number of auctions.
2. Number of bidders.
3. Number of distinct bidders.
4. Name and rating of all sellers.
5. Lowest seller rating (discard new sellers<sup>1</sup>).
6. Names of sellers with the lowest rating.
7. Items with a Celeron CPU.
8. Average number of bids.
9. Information about items with no memory element specified.
10. Highest bid.

<sup>1</sup> You may want to use the `normalize-space()` function.