

XML, XPath, XQuery

Ramakrishnan & Gehrke, Chapter 24 / 27

XML

- XML = eXtensible Markup Language

- flexible mechanism for defining domain-specific data exchange formats
- ASCII

```
<molecule>
  <weight>234.5</weight>
  <spectra>...</spectra>
  <figures>...</figures>
</molecule>
```

- "Extensible":

meta language for defining new markup languages

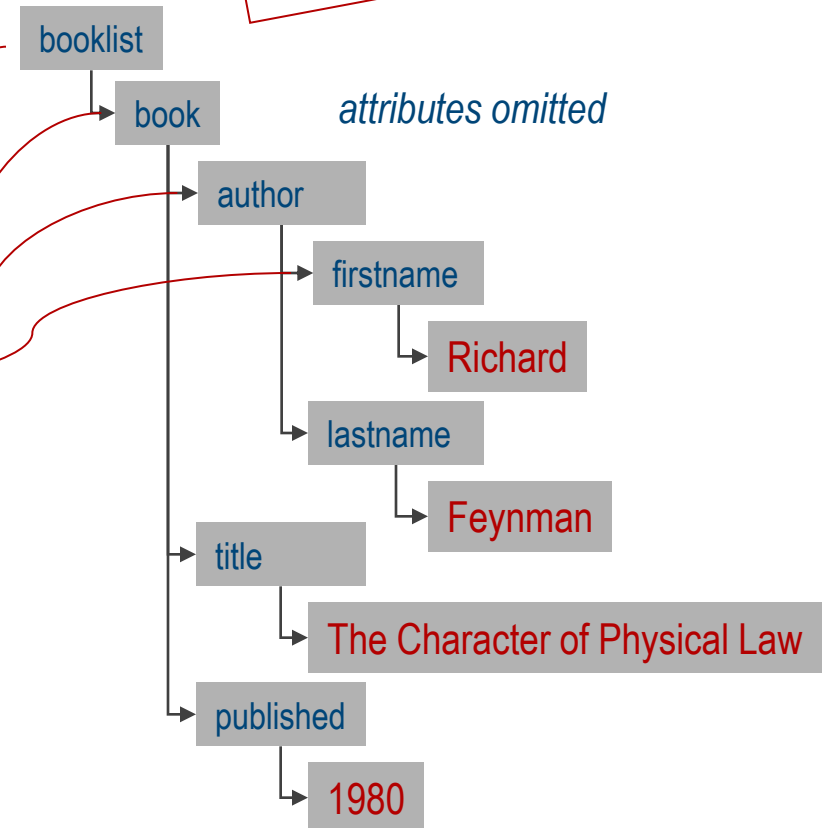
- Each language defines aka "document type", expressed in XML Schema
- Ex: XHTML = HTML in XML; SensorML; MathML; MusicML; ...
- Automatic validity checking against schema

XML Document Tree

- Tree nodes = [element | attribute | text] info: set items

www.w3schools.com

```
<?xml version="1.0" encoding="UTF-8">
<!DOCTYPE booklist SYSTEM "library.std">
<booklist>
  <book genre="Science" format="Hardcover">
    <author>
      <firstname>Richard</firstname>
      <lastname>Feynman</lastname>
    </author>
    <title>The Character of Physical Law</title>
    <published>1980</published>
  </book>
</booklist>
```



Pattern Expressions

■ Walk document, return (sub)tree

- „CD prices in catalog“:
`/catalog/cd/price`
- „all titles, artists“:
`/catalog/cd/title | /catalog/cd/artist`
- „all titles and artists“:
`//title | //artist`
- „all CDs in catalog with price 10.90“:
`/catalog/cd[ price=10.90 ]`
- „all CD countries“:
`//cd/@country`

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<catalog>
  <cd country="USA">
    <title>Empire Burlesque</title>
    <artist>Bob Dylan</artist>
    <price>10.90</price>
  </cd>
  <cd country="UK">
    <title>Hide your heart</title>
    <artist>Bonnie Tyler</artist>
    <price>9.90</price>
  </cd>
  <cd country="USA">
    <title>Greatest Hits</title>
    <artist>Dolly Parton</artist>
    <price>9.90</price>
  </cd>
</catalog>
```

XQuery

- XQuery – retrieving information from XML data
 - XQuery is to XML what SQL is to tables
 - XPath: **extract** from DOM tree; XQuery: **derive new** structure
 - Stored in files or in database
- FOR-LET-WHERE-ORDERBY-RETURN = **FLWOR** ("*flower*")
- Ex: "*all book titles published after 1995*"

```
FOR $x IN document("bib.xml")/bib/book
WHERE $x/year > 1995
RETURN $x/title
```

```
<title> abc </title>
<title> def </title>
<title> ghi </title>
```