# Server Framework – 3

## (configuration file)

Peter Levinsky IT, Roskilde

03.02.2025

# XML (and HTML)

- **XML** stands for E**X**tensible **M**arkup **L**anguage
- **XML is not a replacement for HTML**.
- **XML** and HTML were designed with different goals:
  - **XML** was designed to **transport** and **store** data
  - HTML was designed to **display** data
- **XML** is a **W3C Recommendation**

# XML and JSON

- **Both for carrying information (share data).**
- **Json is shorter in bytes**
- **XML can be validated**
- **Json often used in REST-services**
- **XML often used in configuration**

# XML Example

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this Weekend!</body>
</note>
```

**Json equivalent**

```
{"Note":{"To":"Tove","From":"Jani","Heading":"Reminder","Body":
"Don\u0027t forget me this Weekend!"}}
```

# XML Documents Form a Tree Structure

- XML documents must contain a **root element**.
  This element is "the parent" of all other elements.
  NB! Only **one** root element are allowed

- The elements in an XML document form a document tree.

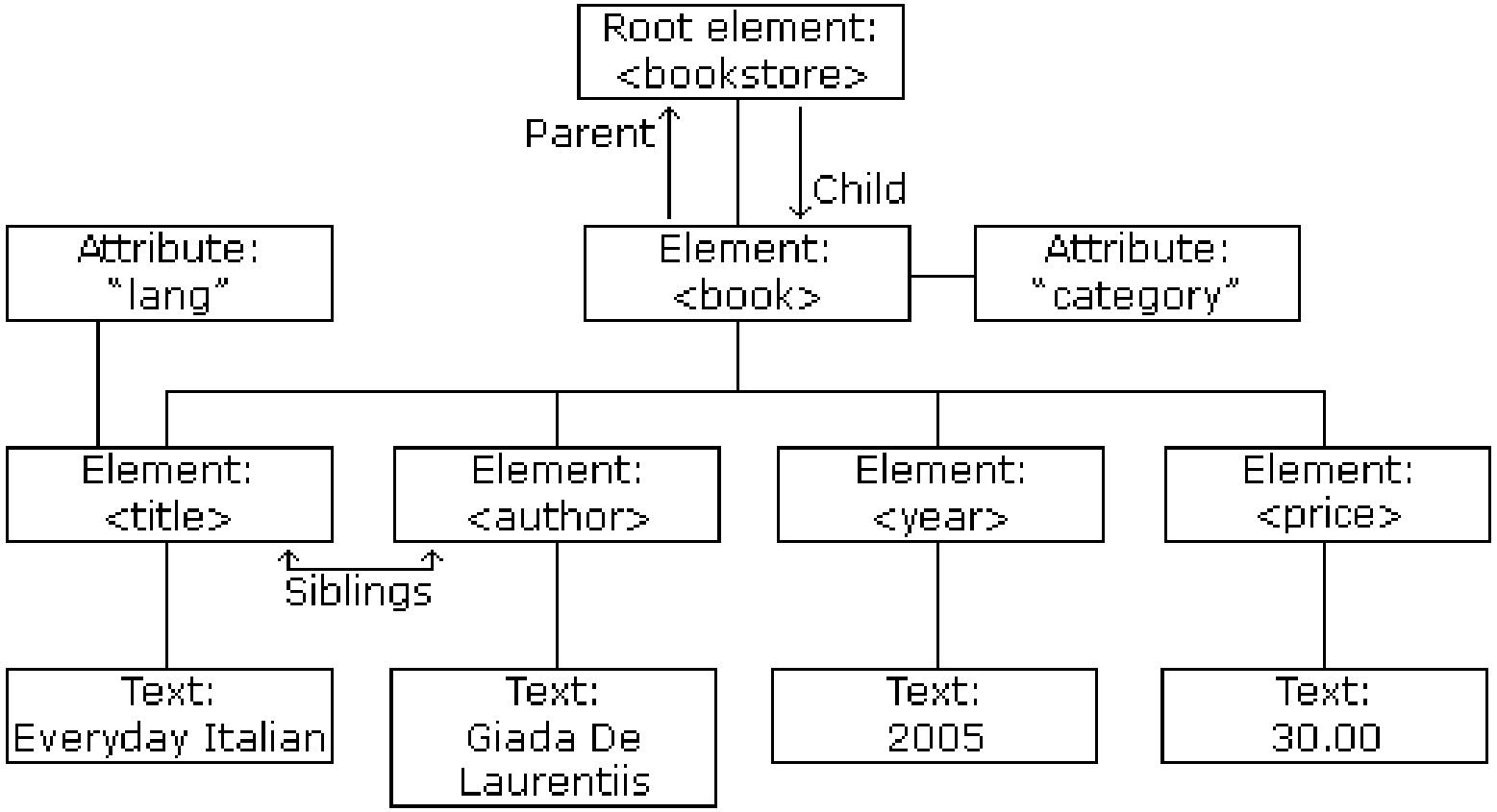
- The tree starts at the root.

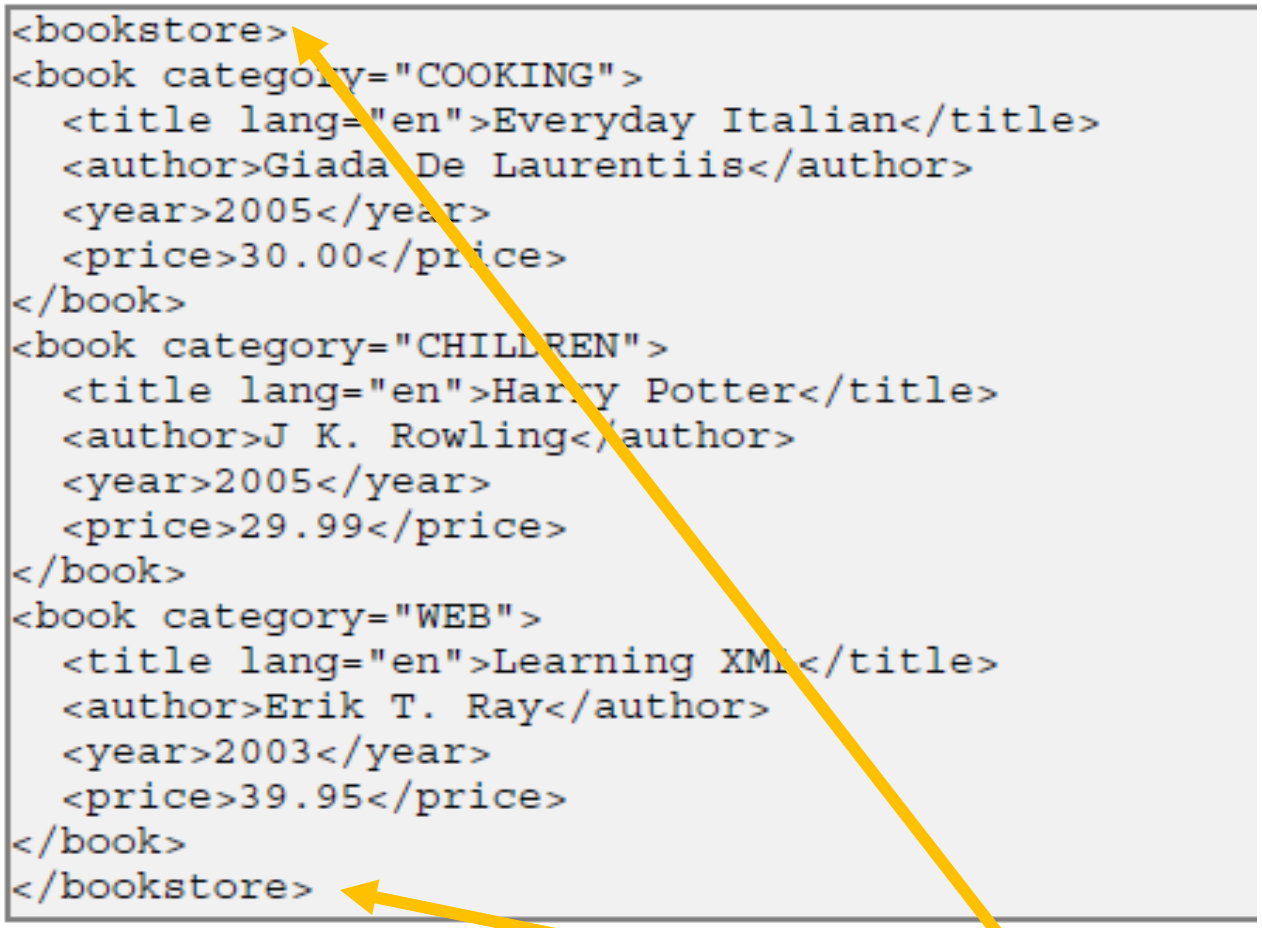
# XML Documents – General structure

- All elements can have sub elements (child elements):

- ```
  <root>
    <child>
      <subchild>.....</subchild>
    </child>
    <child> // sibling
      <subchild>.....</subchild>
    </child>
  </root>
  ```
  ```

# Example of XML-dom-tree






The <book> element itself has 4 children:

<title>,< author>, <year>, <price>.

The **root** element in the example is <bookstore>.
All <book> elements in the document are contained within <bookstore>.

# Valid XML Documents

- A "Valid" XML document is
  - "Well Formed" XML document
  - Conforms to a Document Type Definition (DTD): ***(or schema)***

- ```
  <?xml version="1.0" encoding="ISO-8859-1"?>
  <!DOCTYPE note SYSTEM "Note.dtd">
  <note>
  <to>Tove</to><from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
  </note>
  ```

- The DOCTYPE declaration in the example above, is a reference to an external DTD file.
  ```

# XML DTD (ex: note.dtd)

- The purpose of a DTD is to define the structure of an XML document. It defines the structure with a list of legal elements:
- ```
  <!DOCTYPE note [
  <!ELEMENT note (to,from,heading,body)>  <!ELEMENT to
  (#PCDATA)>
  <!ELEMENT from     (#PCDATA)>
  <!ELEMENT heading (#PCDATA)>
  <!ELEMENT body     (#PCDATA)>
  ]>
  ```
- xxx+ ->1-many  xxx* -> 0-many  xxx? -> 0-1
- , -> and  | -> or
  ```

# XML Schema

- W3C supports an XML based alternative to DTD called XML Schema:
- 
```
<xs:element name="note">
<xs:complexType>
 <xs:sequence>
  <xs:element name="to"    type="xs:string"/>
  <xs:element name="from"    type="xs:string"/>
  <xs:element name="heading" type="xs:string"/>
  <xs:element name="body"    type="xs:string"/>
 </xs:sequence>
</xs:complexType>
</xs:element>
```

# Reading XML files in C#

**Example**:

- To open config-file use:

```
XmlDocument configDoc = new XmlDocument();
configDoc.Load( " << configFileName >> ");
```

- To read a port number:

```
XmlNode xxNode = configDoc.DocumentElement.SelectSingleNode("<NameOfTag>");
if (xxNode != null)
{
    String xxStr = xxNode.InnerText.Trim();
    Int xx = Convert.ToInt32(xxStr);
}
```

# Demo

. . . then exercise.

# Your turn again