JACOBS
UNIVERSITY

XML

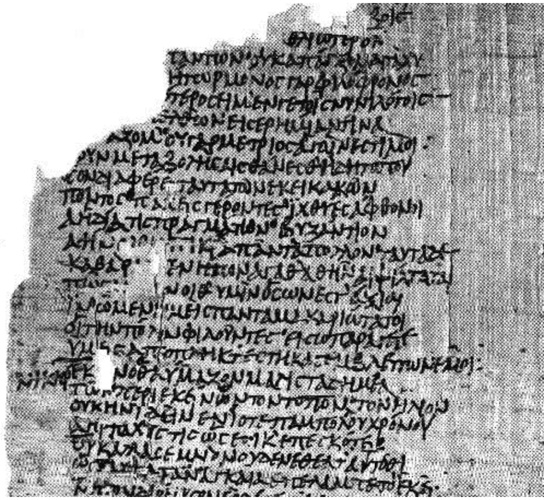
Ramakrishnan & Gehrke, Chapter 20
w3schools.com

Instructors: Peter Baumann

email: p.baumann@jacobs-university.de
tel: -3178
office: room 88, Research 1

So What's Wrong with HTML?

- "Web services" ultimately means: programs communicate via Web!
- Assume end user is not a human, but a program (ex: automated orders) – what can it recognize in HTML?
- Actually, HTML does not help:
  - Freedom to hide semantics in layout conventions (bold-face for name...)
  - No semantic document structure – how to locate address in letter? Check validity?
  - Only one fixed HTML definition – cannot define document types
  - No support for reuse – cannot identify address field across documents
  - Navigation cumbersome – e.g., link into document requires modification of this doc!



- XML to catch semantics of different document types – "semantic Web"

XML

- XML = eXtensible Markup Language
  - is not a protocol (uses HTTP!), is not a database
  - but is a flexible mechanism for defining domain-specific data exchange formats
  - Designed to allow easy implementation

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

- "Extensible": meta language for defining new markup languages
  - Each language defines aka "document type", still leaving large degree of variability to single document instances
    - *DTD = Document Type Definition*
  - "application of XML" = use XML to define such a new markup language
    - *Example: XHTML = redefinition of HTML in XML*
  - Automatic validity checking against definition
  - All ASCII, only references to binary data

Comparison: Degrees of Freedom

	SGML	XML	HTML
SGML declaration	variable	fixed	fixed
DTD	variable	variable	fixed
Document	variable	variable	variable

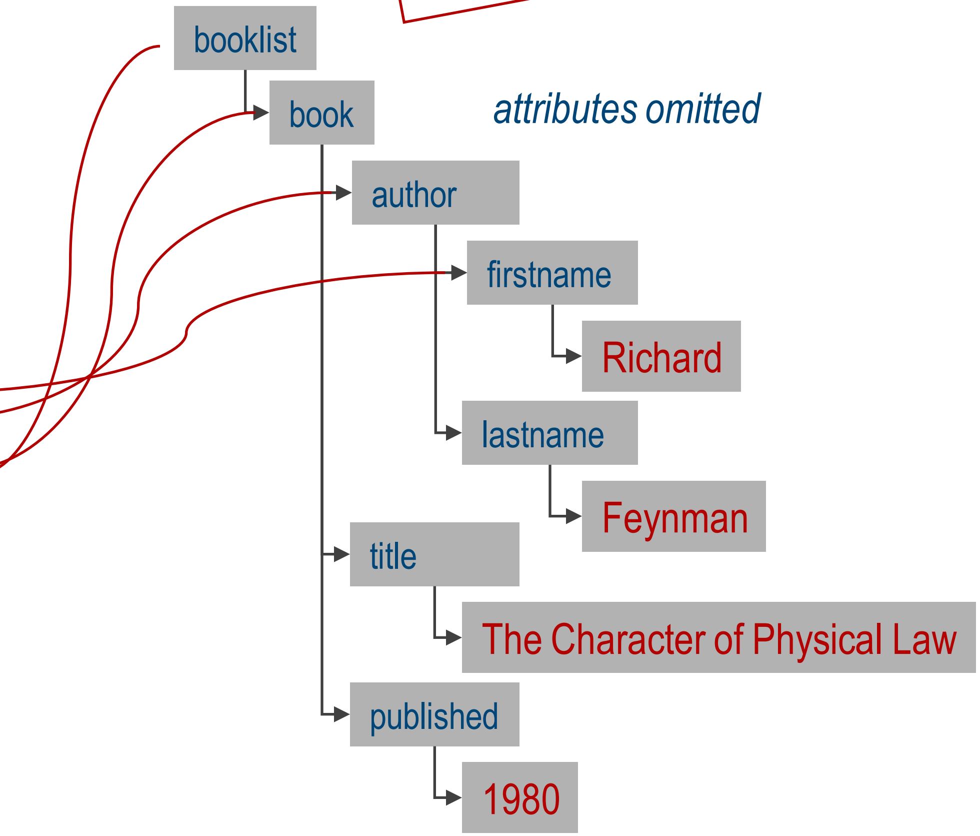
XML Documents Describe Trees

- Tree nodes

  = [element | attribute | text] infoset items

www.w3schools.com

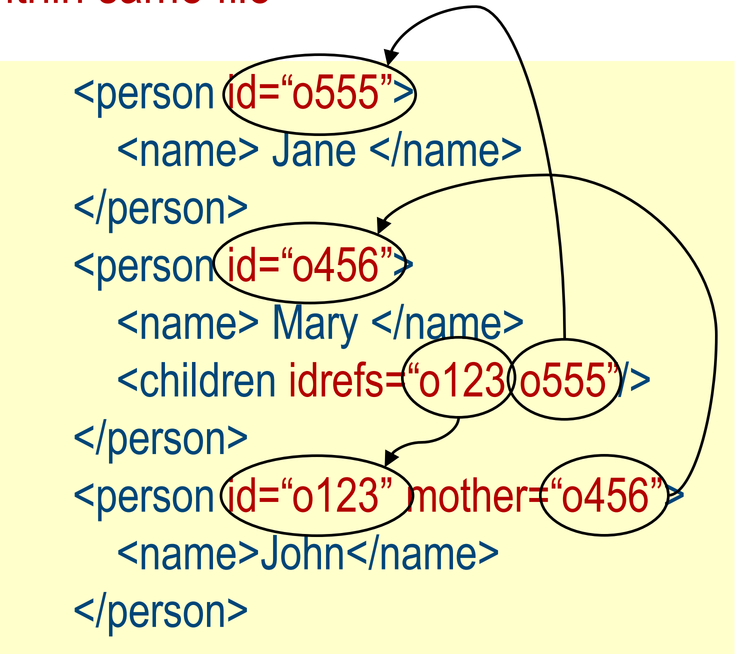
```
<?xml version="1.0" encoding="UTF-8">
<!DOCTYPE booklist SYSTEM "library.std">
<booklist>
  <book genre="Science" format="Hardcover">
    <author>
      <firstname>Richard</firstname>
      <lastname>Feynman</lastname>
    </author>
    <title>The Character of Physical Law</title>
    <published>1980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



attributes omitted

OIDs and References

- OID = object identifier
  - Symbolic name for elements within same file



```
<person id=“o555”>
   <name> Jane </name>
</person>
<person id=“o456”>
   <name> Mary </name>
   <children idrefs=“o123 o555”/>
</person>
<person id=“o123” mother=“o456”>
   <name>John</name>
</person>
```

- XML just syntax, referential integrity in no way guaranteed!

XML Schema

- W3C Recommendation (ie: std), 2012
  - Schema for XML document instances, expressed in XML
  - extensible, built-in data type support, modular through namespaces
- Ex:

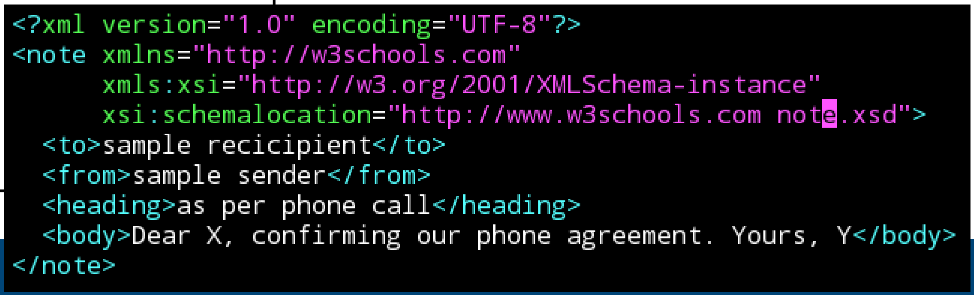
```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.w3schools.com"
xmlns="http://www.w3schools.com"
elementFormDefault="qualified">

<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>

</xs:schema>
```

„complex“ = contains other elements

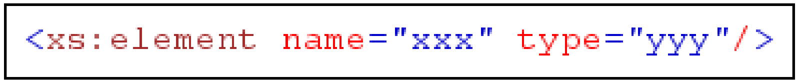
„simple“ = no sub-elements



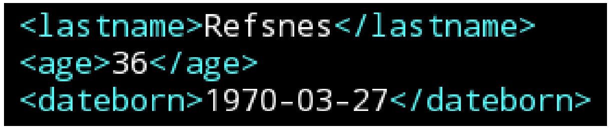
```
<?xml version="1.0" encoding="UTF-8"?>
<note xmlns="http://w3schools.com"
      xmls:xsi="http://w3.org/2001/XMLSchema-instance"
      xsi:schemalocation="http://www.w3schools.com note.xsd">
  <to>sample recicipient</to>
  <from>sample sender</from>
  <heading>as per phone call</heading>
  <body>Dear X, confirming our phone agreement. Yours, Y</body>
</note>
```

XML Schema: Some Details

- Simple element of name xxx and type yyy:



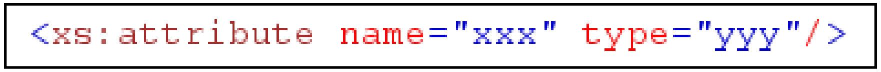
```
<xs:element name="xxx" type="yyy"/>
```



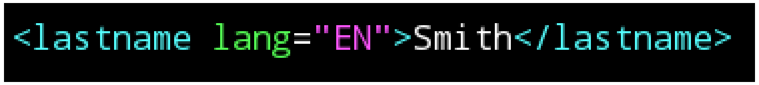
```
<lastname>Refsnes</lastname>
<age>36</age>
<dateborn>1970-03-27</dateborn>
```

  - xs:string, xs:integer, xs:boolean, xs:date, xs:time, ...

- Attribute xxx of type yyy:

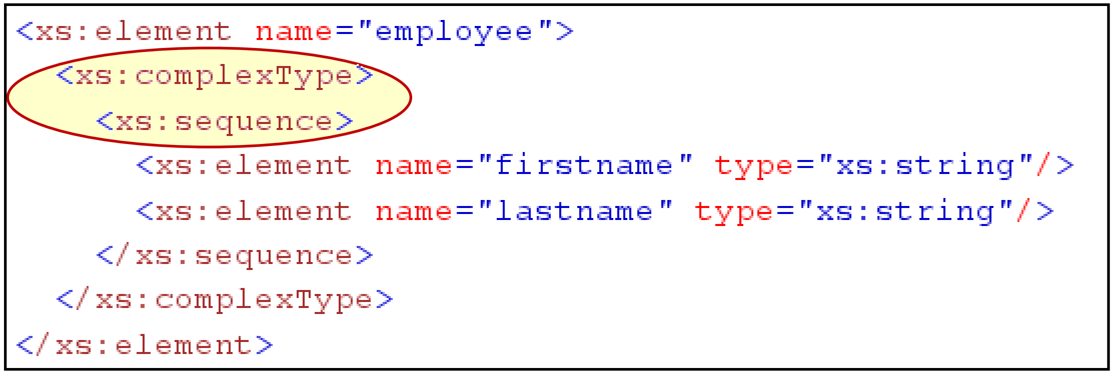


```
<xs:attribute name="xxx" type="yyy"/>
```

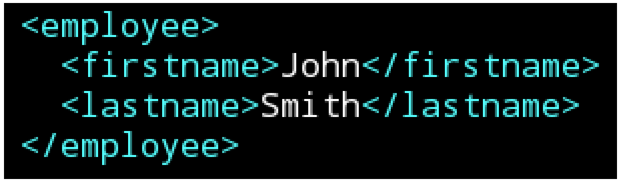


```
<lastname lang="EN">Smith</lastname>
```

- Complex element employee with firstname, lastname:



```
<xs:element name="employee">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="firstname" type="xs:string"/>
      <xs:element name="lastname" type="xs:string"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
```

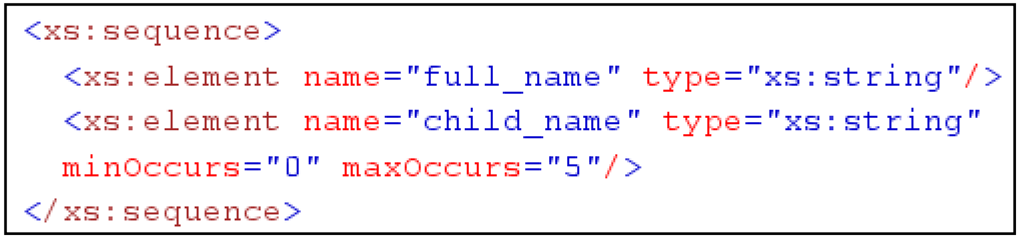


```
<employee>
  <firstname>John</firstname>
  <lastname>Smith</lastname>
</employee>
```

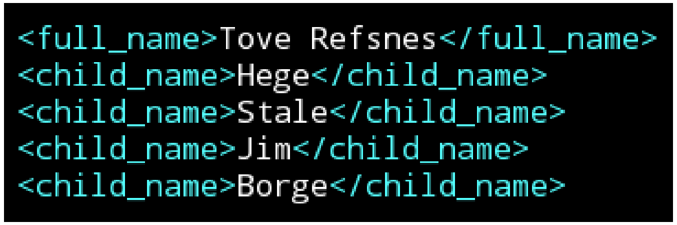
More on Simple & Complex

Yes, confusing

- xs:complexContent → elements & attributes & text outside elements
- xs:simpleContent → no sub-elements, only text & attributes
- xs:complexType → non-atomic structure
- Indicators define use of elements
  - xs:sequence, xs:choice, ...
  - xs:minOccurs, xs:maxOccurs



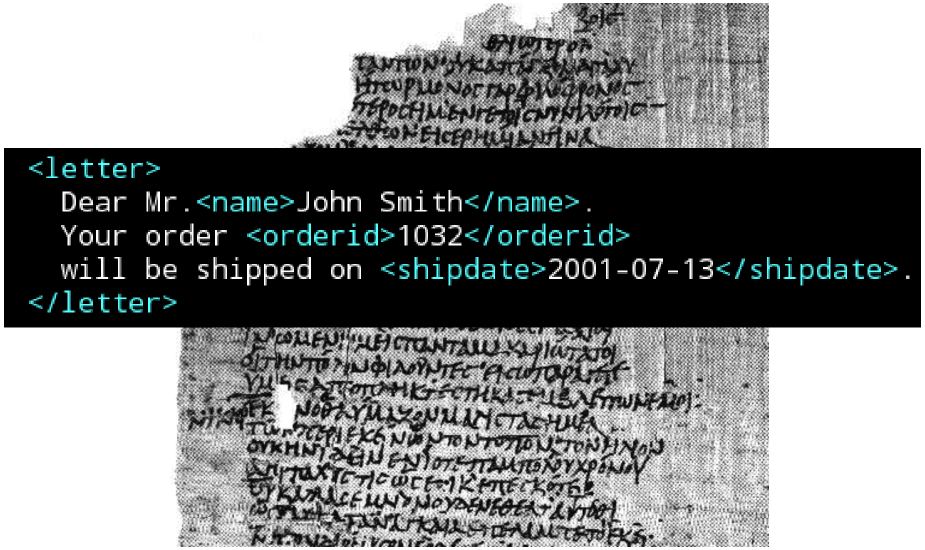
```
<xs:sequence>
  <xs:element name="full_name" type="xs:string"/>
  <xs:element name="child_name" type="xs:string"
  minOccurs="0" maxOccurs="5"/>
</xs:sequence>
```



```
<full_name>Tove Refsnes</full_name>
<child_name>Hege</child_name>
<child_name>Stale</child_name>
<child_name>Jim</child_name>
<child_name>Borge</child_name>
```

XML Assessment

- Document semantics captured
  - Named „markups“ indicate meaning
  - Powerful document structuring, incl. nested elements, attributes, ...
  - Strong typing
  - Automatic validation & processing



```
<letter>
  Dear Mr.<name>John Smith</name>.
  Your order <orderid>1032</orderid>
  will be shipped on <shipdate>2001-07-13</shipdate>.
</letter>
```

- Drawbacks:
  - inefficient (2x tag names!)
  - DTD rudimentary; XML Schema powerful, but sometimes weird

XML Domain Standards

- Many domain-specific schemas existing
- MathML
- Chemical Markup Language
- OpenGIS family of geo service standards: WMS, WFS, WCS, …
- MusicML
- BIPS (Bank Internet Payments System)
- …

Summary

- XML allows to define document types (i.e., data exchange formats)
  - In terms of infoset items: elements, attributes, text, (references), ...
  - Becoming *de facto* standard also for, e.g., configuration files (but is no database!)
  - DTD vs XML Schema vs NG Relax vs Schematron vs ...
- Have seen transformation from ER
  - To generate XML connectors suitable for the miniworld of database applications
- Many facets not covered, such as:
  - XSLT: transforming XML to, e.g., HTML
  - XML DOM

Practising DOM Trees

- Let‘s consider XHTML
- DOM tree?
  - Attributes prefixed with „@“
- References?

```
<html>
<head> <title>Friendly Homepage</title> </head>
<body>
Hello World.
<a name= "picture" > <img src= "world.jpg" /> </a>
Click <a href= "#picture">here</a> for a picture.
</body>
</html>
```