

RDF/XML: Encoding RDF into XML

Werner Nutt

Acknowledgment

These slides are essentially identical with those by Sebastian Rudolph for his course on Semantic Web Technologies at TU Dresden

- Brief Introduction to XML
- Overview of XML/RDF

- Brief Introduction to XML
- Overview of XML/RDF

Annotation with Mark-up Languages

- Idea of mark-up:
enrich (unstructured) **text**
with **additional information** (or structure)
- Synonym: *annotate* text

text = data

additional information = metadata

Annotation with Mark-up Languages

- Common strategy: include the text to be annotated in so-called tags:

`<tag name> ... Text ... </tag name>`

↑ opening tag

↑ closing tag

- The additional information is read and interpreted by processing software (e.g., a Web browser)

Annotation with Mark-up Languages

Example: HTML tags encode

- visual presentation information
- links

```
<h2>Ian Horrocks FRS</h2>  
<h3>Professor of Computer Science and  
  Fellow of  
  <a href="http://www.oriel.ox.ac.uk/">Oriel College</a>  
</h3>
```

Output of web browser

Ian Horrocks FRS

Professor of Computer Science and Fellow of [Oriel College](http://www.oriel.ox.ac.uk/)

INTERESTS

My research interests include knowledge representation, [ontologies](#) and [ontology languages](#), modal and [description logics](#), [automated reasoning](#), implementation and optimisation of [reasoning systems](#), and applications in areas such as [e-Science](#) and the [Semantic Web](#).

RECENT PUBLICATIONS

My [recent publications](#) can be found [here](#), and a [complete list](#) of my publications [here](#).



Annotation with Mark-up Languages

Alternative: annotations describing content type

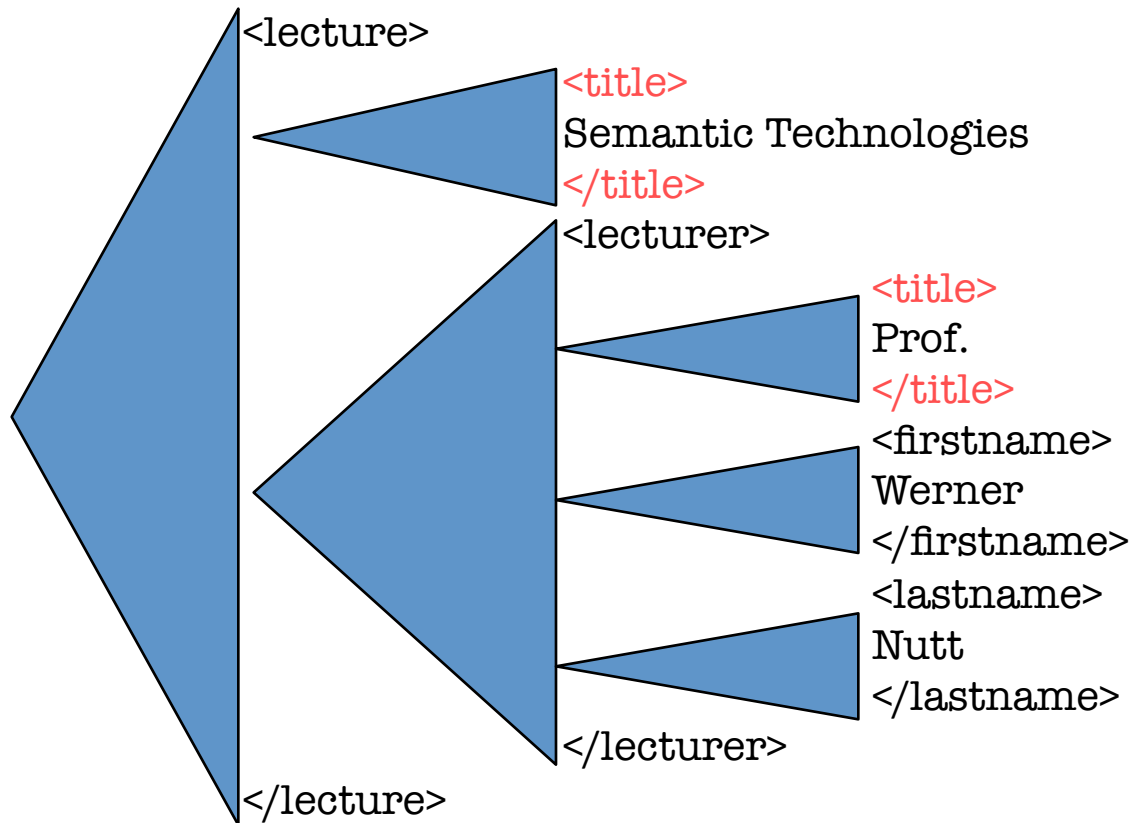
```
<academic>
  <firstname>Ian</firstname>
  <lastname>Horrocks </lastname>
  <fellowship>FRS</fellowship>
  <position>Professor of Computer Science</position>
  and
  <position>Fellow of <college>Oriel College</college>
  </position>
</academic>
```

Annotation with XML Tags

```
<lecture>
  <title>
    Semantic Technologies
  </title>
  <lecturer>
    <title>
      Prof.
    </title>
    <firstname>
      Werner
    </firstname>
    <lastname>
      Nutt
    </lastname>
  </lecturer>
</lecture>
```

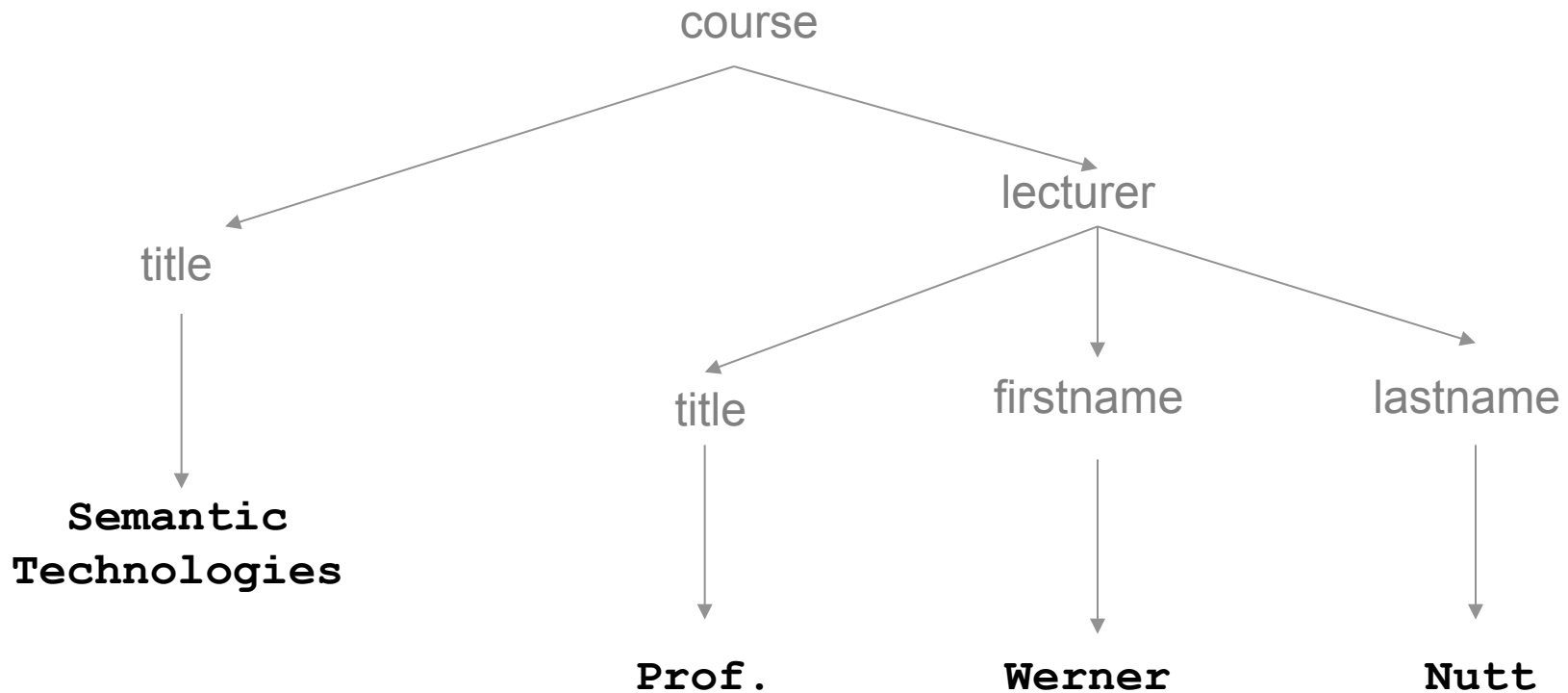
- Nesting of tags is permitted
- **Multiple usage** of tags is permitted

Annotation with XML Tags



- Nesting of tags is permitted
- **Multiple usage** of tags is permitted
- XML tags constitute a tree structure

XML Documents are Node-labeled Trees



Two (main) kinds of nodes:

- element nodes (constituted by the tags)
- content nodes

There are also attributes, comments, ...

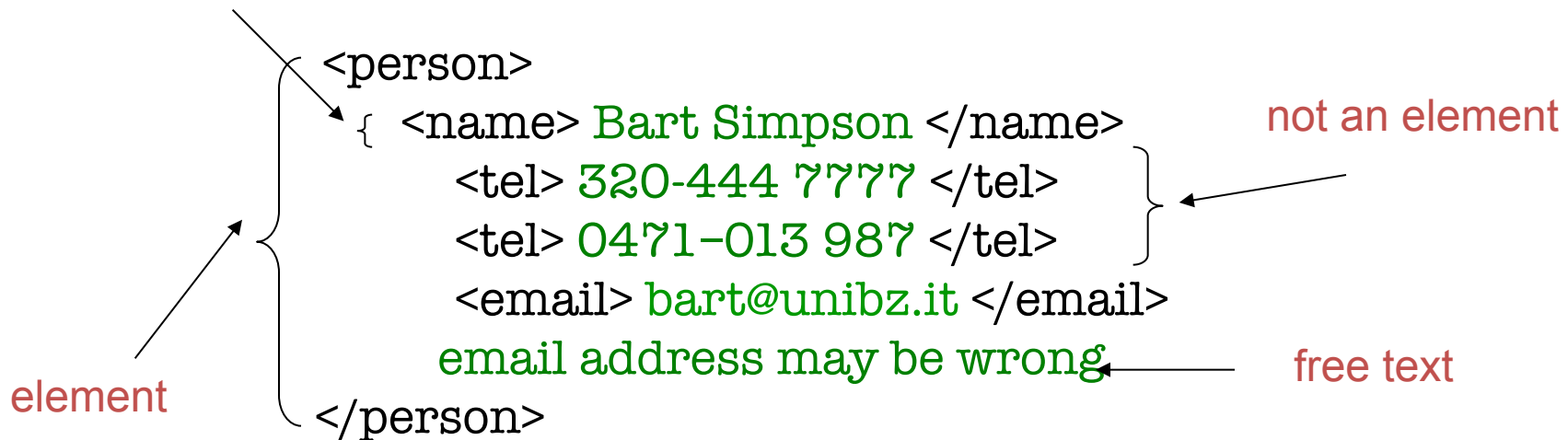
XML

- eXtensible Markup Language
- Origin: document markup ($\text{HTML4.0} \in \text{XML} \subset \text{SGML}$)
- Web standard (W3C) for data exchange:
 - applications receive XML input and produce XML output
 - industry only has to agree on standardized tag names (the vocabulary) and their usage (the schema)
- Complementary language for HTML:
 - HTML describes presentation
 - XML describes content
- Database perspective: XML as a data model for semi-structured data

XML-Syntax: XML Element

- Description of an object enclosed by matching tags
- Content of an elements: text and/or further elements (arbitrary nesting possible)
- Empty elements: `<year></year>` short: `<year/>`
- “Outermost” element is called the top element (and there can be only one per document)

element,
a sub-element of the first



XML-Syntax: XML Attribute

- Pair of name and string-value in start tag or in self-closing tag
- Associated with one XML element
- Alternative option for describing data

attribute email



```
<person email="bart@unibz.it">  
  <name> Bart Simpson </name>  
  <tel> 320-444 7777 </tel>  
  <tel> 0471-013 987 </tel>  
</person>
```

- An element can have several child elements with the same tag, but only one attribute

```
<person name="Bart Simpson" email="bart@unibz.it">  
  <tel>320-444 7777</tel><tel>0471-013 987</tel>  
</person>
```

HTML vs. XML

- HTML: fixed vocabulary (set of tags) and semantics (visual presentation of text)
- XML: free choice of names for describing application-specific syntax and semantics
- XML $\subset$ SGML
- The info in one XML document can be translated in different HTML documents

XML Namespaces: Motivation

- Element tags and attributes should have an unambiguous meaning
- Within one vocabulary, tag and attribute names can well be defined (like within a Java package)
- If vocabularies with overlapping terms are merged, names can clash
 - Example: title of a lecture vs. title of a person
- Namespaces resolve such conflicts

XML Namespaces

- ... are collections of element tags and attribute names
- ... are identified by a URI
 - the URI may resolve to a resource with information about the namespace, but need not
- ... can be defined by an XML schema
- ... allow for disambiguation of tag names through usage of different “prefixes”
 - a prefix is separated from a local name by a colon (:), thereby prefix:name tags come into being

Namespace Bindings

- How are prefixes associated with name space URIs?
 - ➔ by inserting an attribute `xmlns:prefix` into
 - the relevant element, or
 - some of its predecessor elements:
`prefix:name1 , ... , prefix:namen`
- The attribute `xmlns:prefix` has as value a URI
 - the URI may point to a description of the namespace
- An XML element can contain bindings for several (different) name spaces:
 - use separate attributes
`xmlns:prefix1, ..., xmlns:prefixm`

Example: Without Namespaces

```
<course>  
  <title>Semantic Technologies</title>  
  <lecturer>  
    <title>Prof.</title>  
    <firstname>Werner</firstname>  
    <lastname>Nutt</lastname>  
  </lecturer>  
</course>
```

title is an ambiguous tag

Two Distinct Namespaces

```
<lec:lecture xmlns:lec="http://www.example.org/lectures/"  
             xmlns:per="http://www.example.org/person/">  
<lec:title>Semantic Technologies</lec:title>  
  <lec:lecturer>  
    <per:title>Prof.</per:title>  
    <per:firstname>Werner</per:firstname>  
    <per:lastname>Nutt</per:lastname>  
  </lec:lecturer>  
</lec:lecture>
```

title has been disambiguated by the prefixes **lec:** and **per:**, each of which indicates a different namespace

- Brief Introduction to XML
- Overview of XML/RDF

XML Syntax of RDF

We show how to represent the following three triples in RDF/XML:

```
@prefix db: http://dbpedia.org/resource/ .
```

```
@prefix dbo: http://example.org/terms/ .
```

```
db:Massachusetts dbo:capital db:Boston .
```

```
db:Massachusetts dbo:nickname "The Bay State" .
```

```
db:Boston dbo:nickname "Beantown" .
```

RDF/XML

- W3C Standard since 1999, revised in 2004
- Used to be the only standard until 2014,
when the Turtle standard was adopted
- Encodes RDF into standard XML

XML Syntax of RDF

- Like in XML, name spaces are used in order to disambiguate tag names
- RDF-specific tags have a predefined name space, by convention abbreviated with 'rdf'

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<rdf:RDF
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:db="http://dbpedia.org/resource/"
  xmlns:dbo="http://example.org/terms/"
```

```
<rdf:Description rdf:about="http://dbpedia.org/resource/Massachusetts">
```

```
  <dbo:capital>
```

```
    <rdf:Description rdf:about="http://dbpedia.org/resource/Boston"/>
```

```
  </dbo:capital>
```

```
</rdf:Description>
```

```
</rdf:RDF>
```

XML Syntax of RDF

- The `rdf:Description` element encodes the **subject** (the URI of which is stated as the value of the associated `rdf:about` attribute)
- Every **element directly nested** into an `rdf:Description` element denotes a **predicate** (the URI of which is the element name)
- Predicate **elements in turn contain the triple's object as** `rdf:Description` element

...

```
<rdf:Description rdf:about="http://dbpedia.org/resource/Massachusetts">  
  <dbo:capital>  
    <rdf:Description rdf:about="http://dbpedia.org/resource/Boston"/>  
  </dbo:capital>  
</rdf:Description>
```

...

XML Syntax of RDF

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<rdf:RDF
```

```
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:db="http://dbpedia.org/resource/"
```

```
  xmlns:dbo="http://example.org/terms/"
```

```
  <rdf:Description rdf:about="http://dbpedia.org/resource/Massachusetts">
```

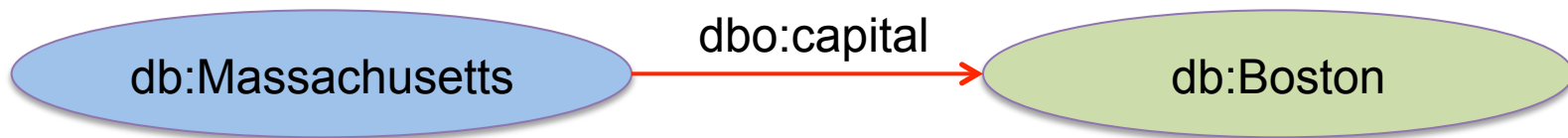
```
    <dbo:capital>
```

```
      <rdf:Description rdf:about="http://dbpedia.org/resource/Boston"/>
```

```
    </dbo:capital>
```

```
  </rdf:Description>
```

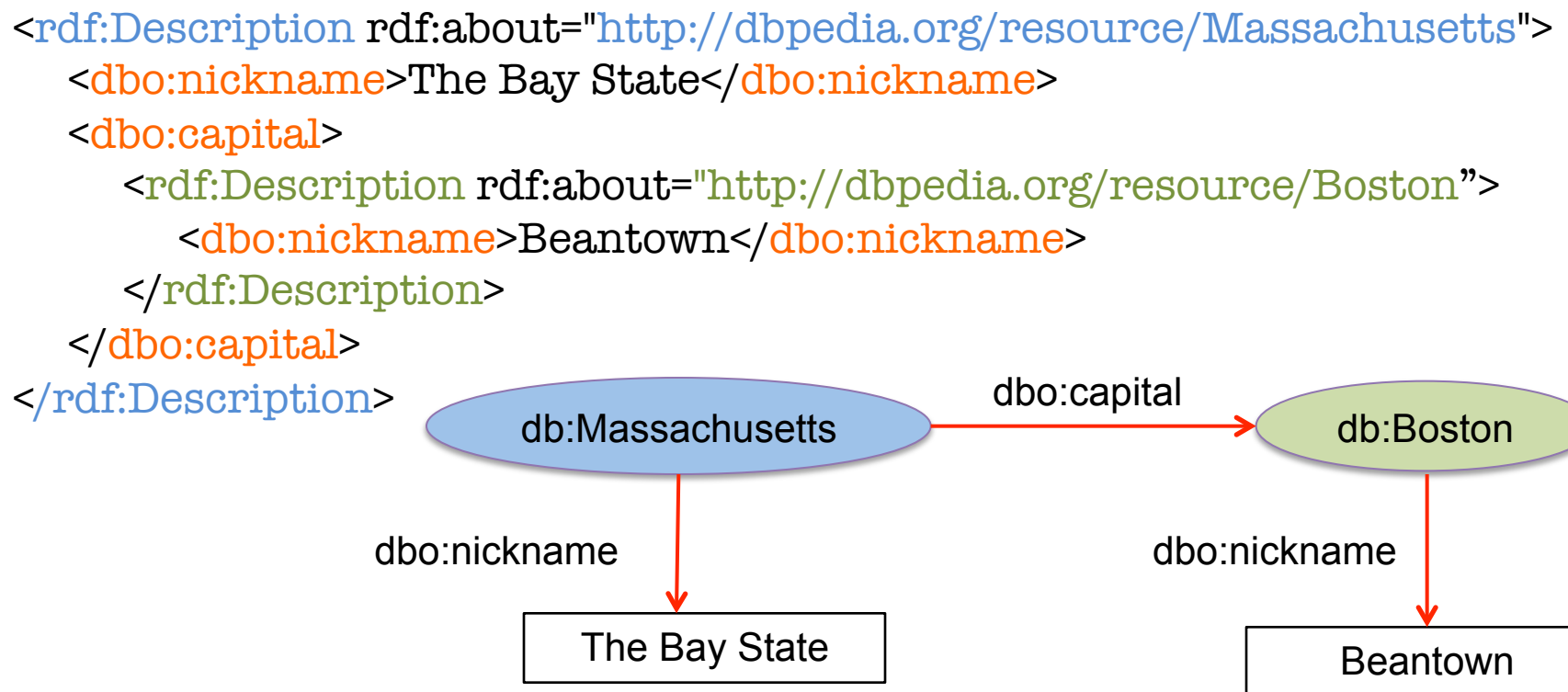
```
</rdf:RDF>
```



XML Syntax of RDF

Untyped literals can be included as free text into the predicate element
Condensed forms admissible:

- one subject containing several property elements
- one object description serves as subject for another triple

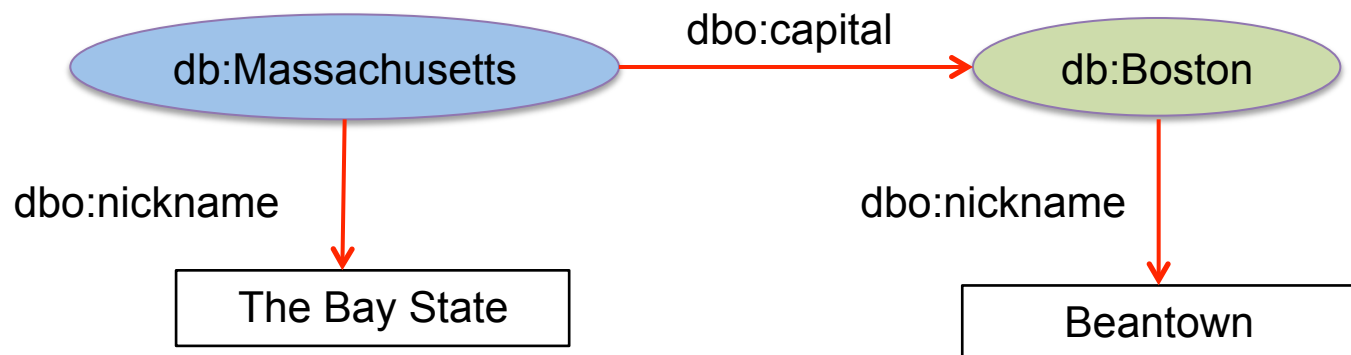


XML Syntax of RDF

Alternative (but equivalent) representation of literals as XML attributes

- property URIs are then used as attribute names
- object URIs can be given as value of the `rdf:resource` attribute inside a property tags

```
<rdf:Description rdf:about="http://dbpedia.org/resource/Massachusetts"  
  dbo:nickname="The Bay State">  
  <dbo:capital rdf:resource="http://dbpedia.org/resource/Boston"/>  
</rdf:Description>  
<rdf:Description rdf:about="http://dbpedia.org/resource/Boston"  
  dbo:nickname="Beantown"/>
```



Exercise

Encode in RDF/XML

```
@prefix ex: <http://example.org/>
```

```
@prefix foaf: <xmlns.com/foaf/0.1/>
```

```
ex:johnSmith foaf:firstName "John"@en ;
```

```
foaf:lastName "Smith"@en ;
```

```
foaf:knows ex:jimmieBrown .
```

RDF/XML Syntax: Complications

- Namespaces are needed
 - for abbreviation reasons,
 - because colons inside XML elements and attributes are always interpreted as namespace delimiters
- Problem: In XML, no namespaces are allowed in attribute values (would be interpreted as URI schema), thus we cannot write:

```
rdf:about="db:Boston"
```

- “Workaround” via XML entities (= XML macros):

Declaration:

```
<!ENTITY dbo 'http://dbpedia.org/resource/'>
```

Usage:

```
rdf:resource="&ex;SemanticWeb"
```

RDF/XML Syntax: Base URIs

- One URI can be declared the base URI of an XML document
- Attributes that expect URIs (like `rdf:about`, `rdf:resource`) can be given “relative URIs”, which have no schema part
- The full URI is then reconstructed by using the base URI as a prefix

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:db="http://dbpedia.org/resource/"
  xmlns:dbo="http://example.org/terms/"
  xmlns:base="http://dbpedia.org/resource"/>

  <rdf:Description rdf:about="Massachusetts">
    <dbo:capital rdf:resource="Boston"/>
  </rdf:Description>
</rdf:RDF>
```

Types in RDF/XML

- Types are indicated by the `rdf:datatype` attribute

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:db="http://dbpedia.org/resource/"
  xmlns:dbo="http://example.org/terms/"
  xmlns:base="http://dbpedia.org/resource/">

  <rdf:Description rdf:about="Massachusetts">
    <dbo:population
      rdf:datatype="http://www.w3.org/2001/XMLSchema#integer"/>
      1788-02-06
    </dbo:population >
  </rdf:Description>
</rdf:RDF>
```