

Overview of RDF Data Model

Jose Emilio Labra Gayo

WESO Research group
University of Oviedo, Spain

Short history of RDF

Around 1997 - PICS, Dublin core, Meta Content Framework

1997 1st Working draft <https://www.w3.org/TR/WD-rdf-syntax-971002>

RDF/XML

1999 1st W3c Rec <https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>

First applications RSS, EARL

2004 - RDF Revised <https://www.w3.org/TR/2004/REC-rdf-concepts-20040210/>

Emergence of SPARQL, Turtle, Linked Data

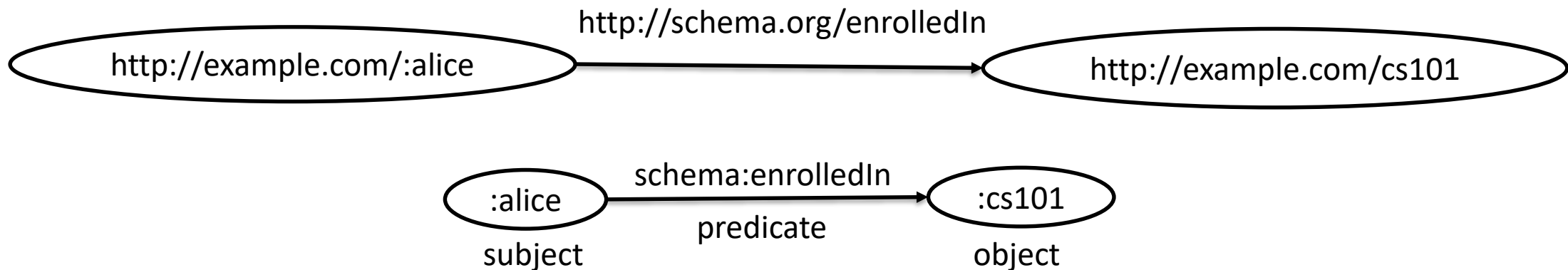
2014 - RDF 1.1 <https://www.w3.org/TR/rdf11-concepts/>

RDF Data Model

RDF Graph = set of triples

A triple = (subject, predicate, object)

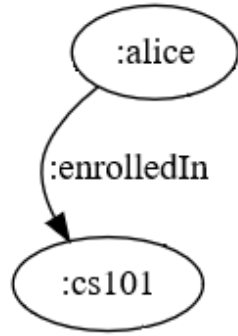
Example:



N-Triples representation

```
<http://example.com/alice> <http://schema.org/knows> <http://example.com/bob> .
```

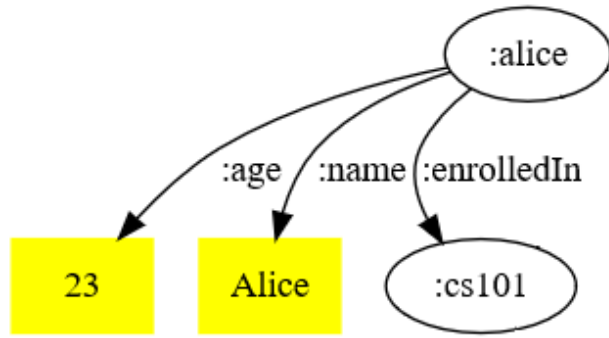
RDF Graph



Basic statement = a simple triple

RDF Graph

...we can add more statements



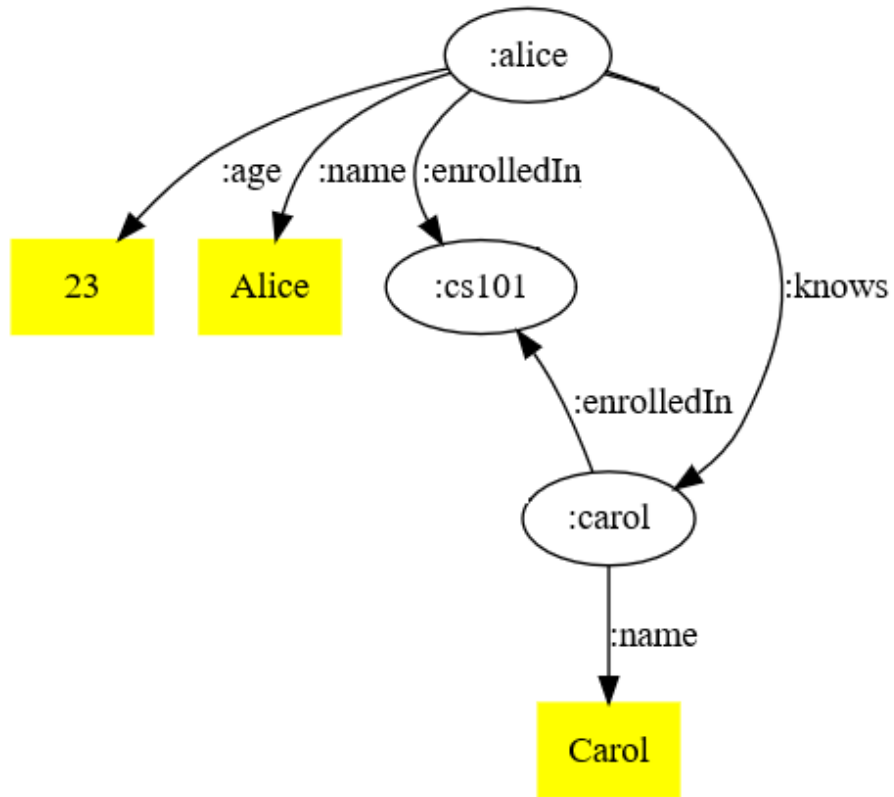
Circled nodes are IRIs



Yellow boxes are literals

RDF Graph

...more statements can be added...

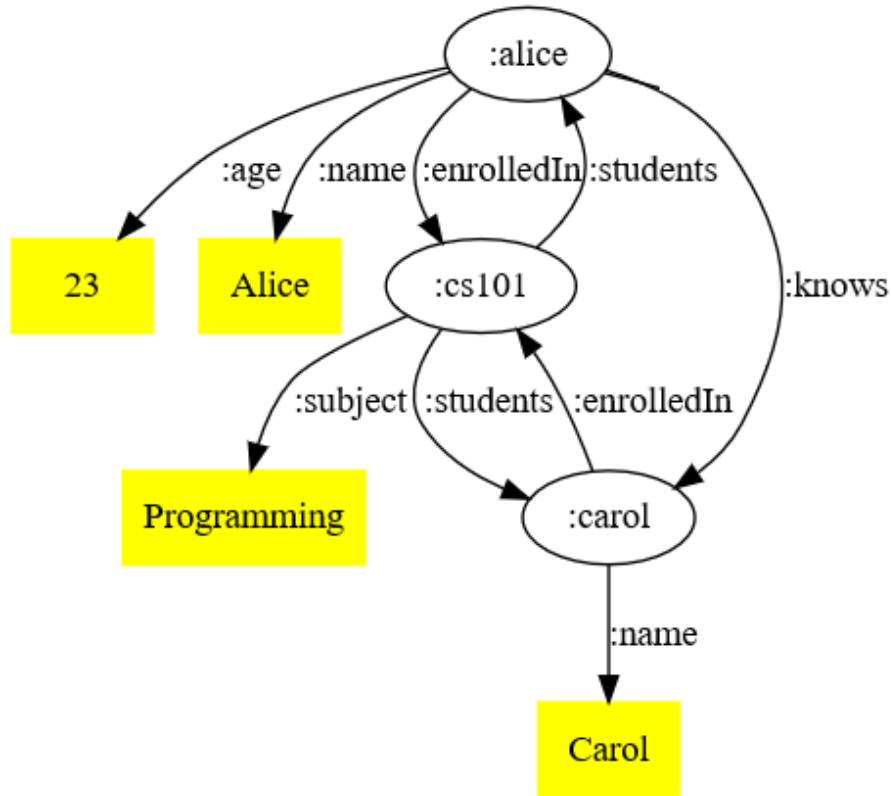


RDF Graph

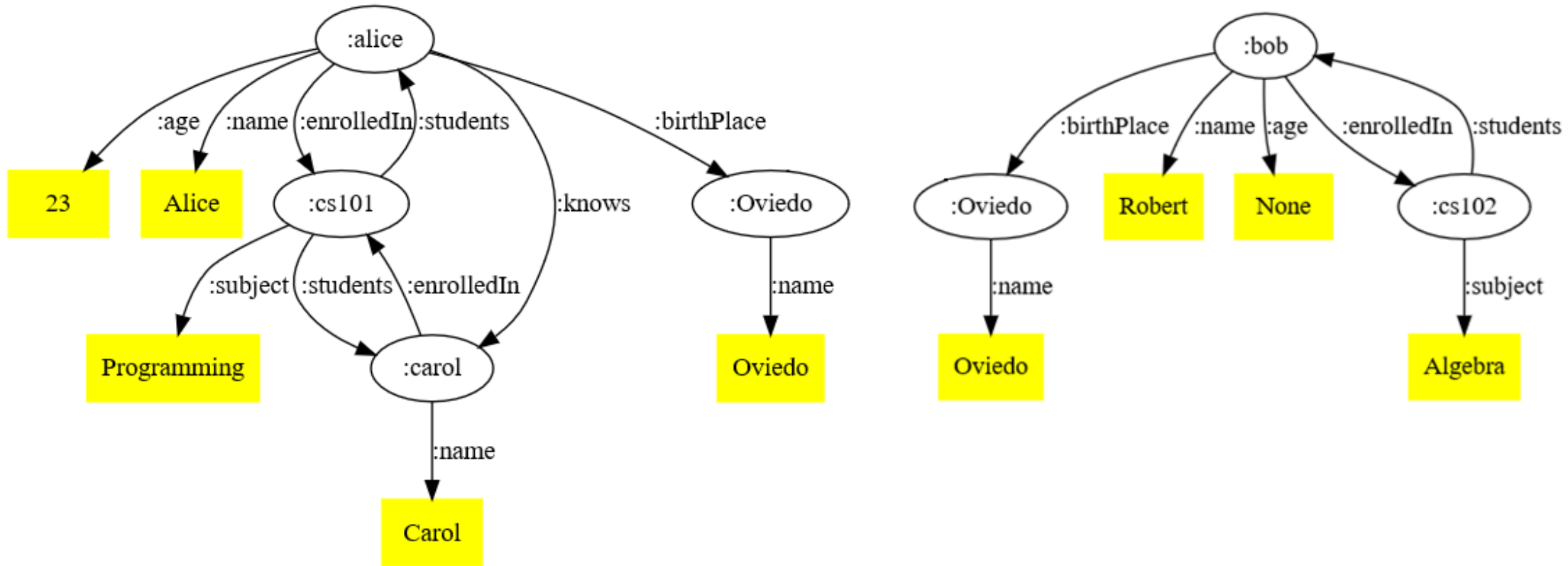
...and more...

Forming a graph

...which can contain cycles

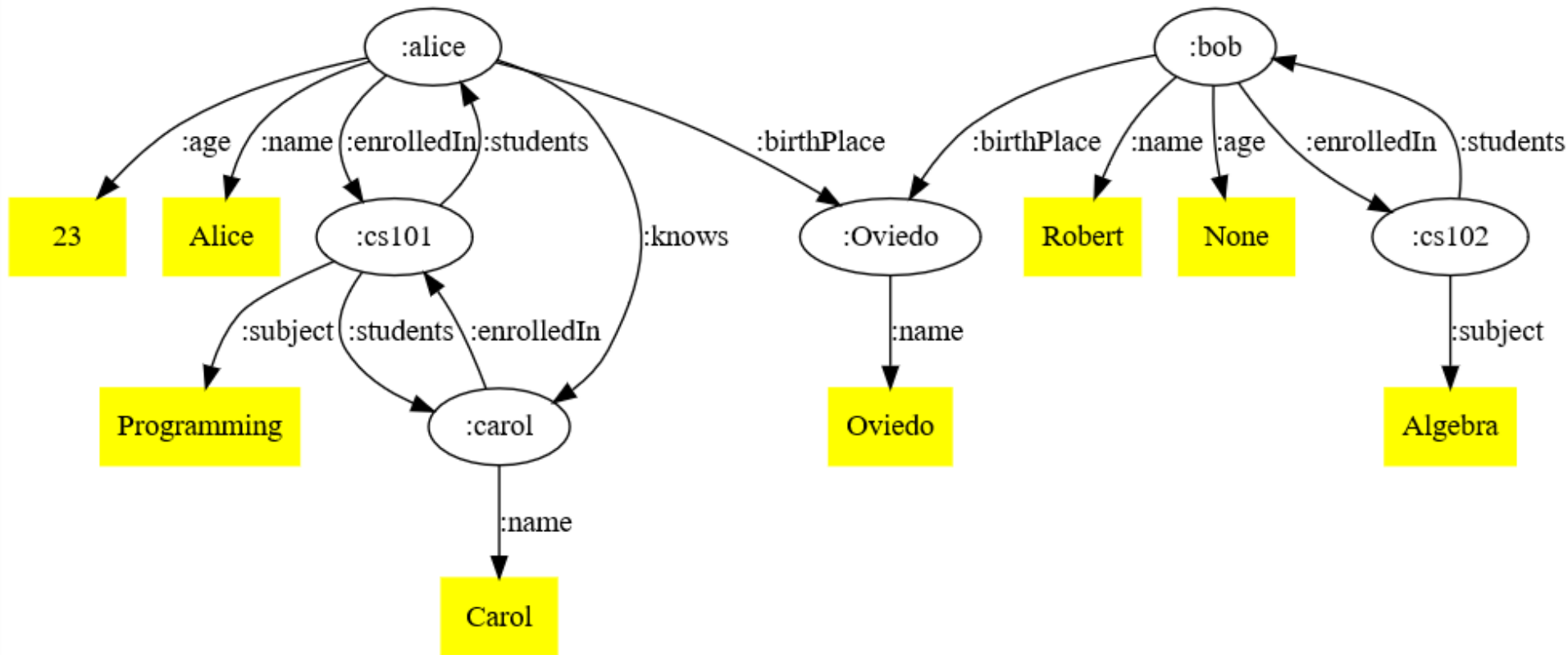


RDF Graph



Graphs can be created independently

RDF Graph

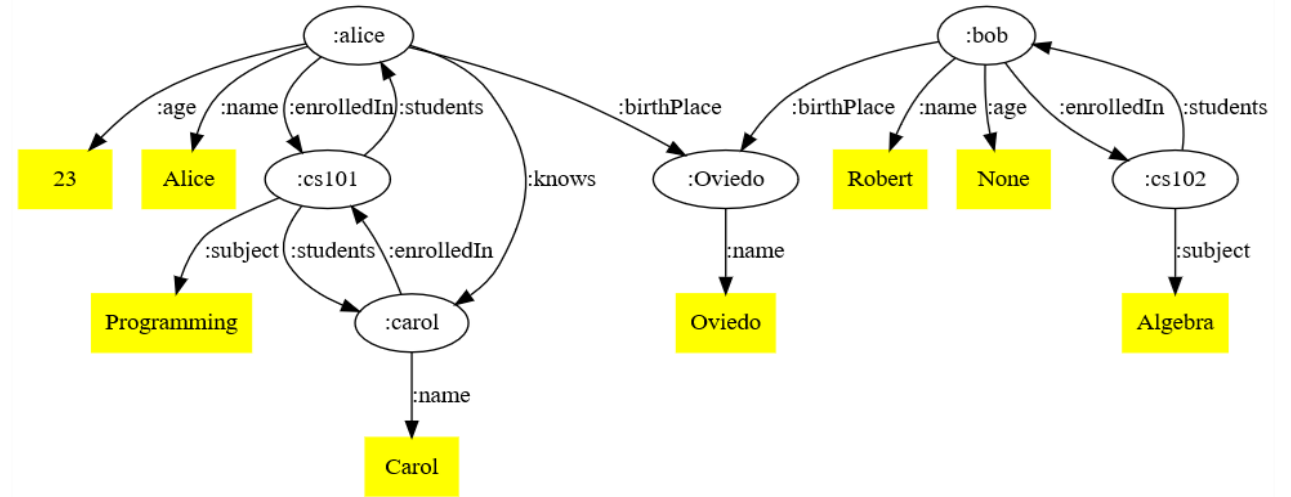


...and automatically merged

RDF helps information integration

RDF syntax

Basic syntax: N-triples



```
<http://example.org/alice> <http://example.org/name> "Alice" .
<http://example.org/alice> <http://example.org/age> "23"^^<http://www.w3.org/2001/XMLSchema#integer> .
<http://example.org/alice> <http://example.org/enrolledIn> <http://example.org/cs101> .
<http://example.org/alice> <http://example.org/knows> <http://example.org/carol> .
<http://example.org/alice> <http://example.org/birthPlace> <http://example.org/Oviedo> .
<http://example.org/bob> <http://example.org/name> "Robert" .
<http://example.org/bob> <http://example.org/age> "None" .
<http://example.org/bob> <http://example.org/birthPlace> <http://example.org/Oviedo> .
<http://example.org/bob> <http://example.org/enrolledIn> <http://example.org/cs102> .
<http://example.org/carol> <http://example.org/name> "Carol" .
<http://example.org/carol> <http://example.org/enrolledIn> <http://example.org/cs101> .
<http://example.org/cs101> <http://example.org/subject> "Programming" .
<http://example.org/cs101> <http://example.org/students> <http://example.org/alice> .
<http://example.org/cs101> <http://example.org/students> <http://example.org/carol> .
<http://example.org/cs102> <http://example.org/subject> "Algebra" .
<http://example.org/cs102> <http://example.org/students> <http://example.org/bob> .
```

Turtle Syntax

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

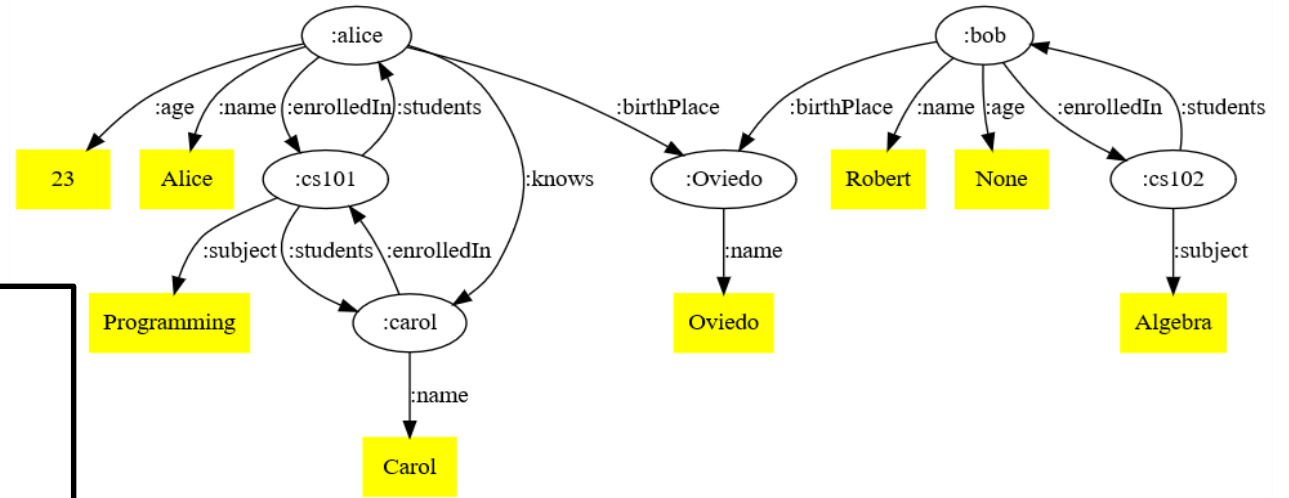
:alice  :name      "Alice" ;
        :age       23 ;
        :enrolledIn :cs101 ;
        :knows     :carol ;
        :birthPlace :Oviedo .

:carol  :name      "Carol" ;
        :enrolledIn :cs101 .

:cs101  :students  :alice , :carol ;
        :subject   "Programming" .

:bob    :name      "Robert" ;
        :age       "None" ;
        :enrolledIn :cs102 ;
        :birthPlace :Oviedo .

:cs102  :students  :bob ;
        :subject   "Algebra" .
```



Some simplifications

prefix declarations

- ; when triples share the subject
- , when triples share subject and object

Try it: <https://goo.gl/pK3Csh>

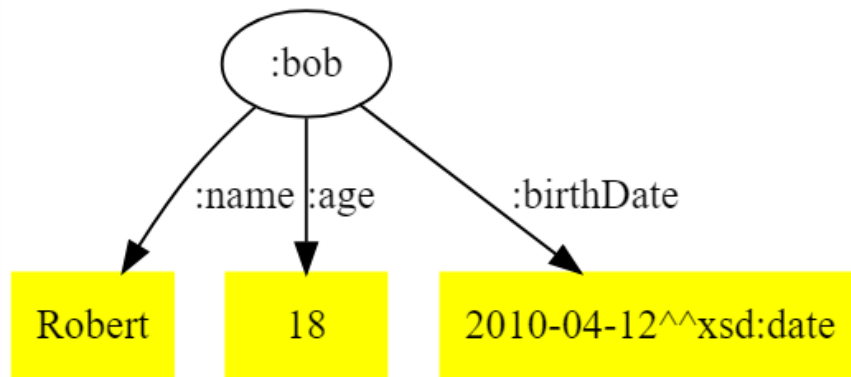
Literals

Objects can also be literals

Literals contain a lexical form and a datatype

Common datatypes: XML Schema primitive datatypes

If not specified, a literal has type xsd:string



```
:bob      :name      "Robert"      ;  
          :age       18              ;  
          :birthDate "2010-04-12"^^xsd:date .
```

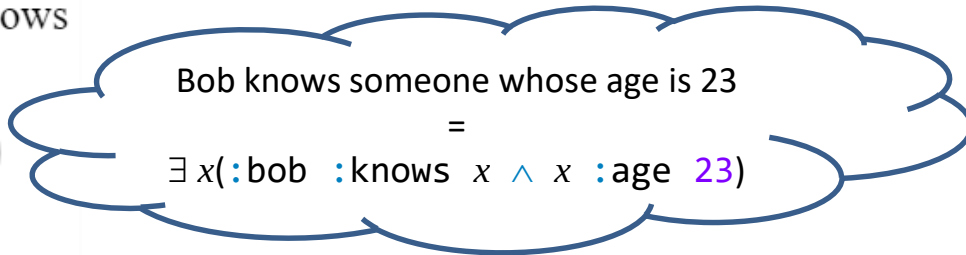
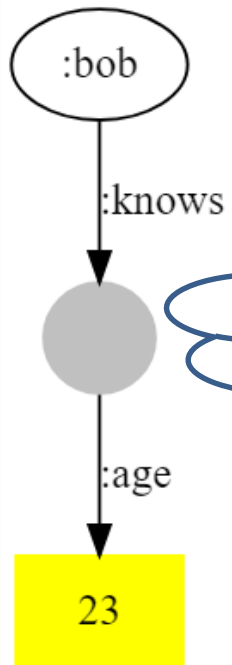


```
:bob      :name      "Robert"^^xsd:string ;  
          :age       "18"^^xsd:integer  ;  
          :birthDate "2010-04-12"^^xsd:date .
```

Blank nodes

Subjects and objects can also be Blank nodes

Blank nodes can have local identifiers



```
:bob :knows _:1 .  
_:1 :age 23 .
```

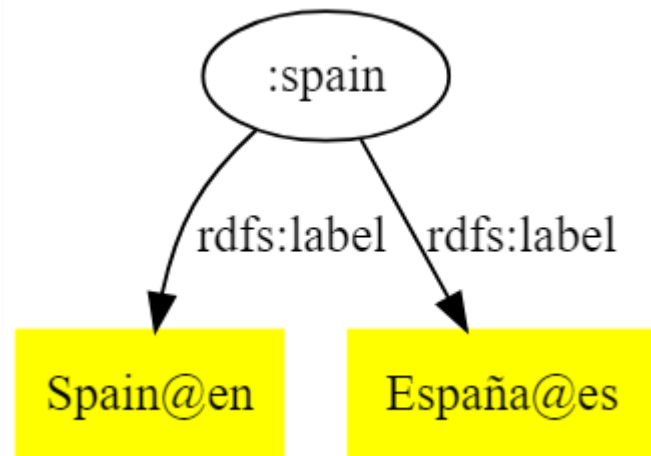
or

```
:bob foaf:knows [  
    foaf:age 23  
] .
```

Language tagged strings

String literals can be qualified by a language tag

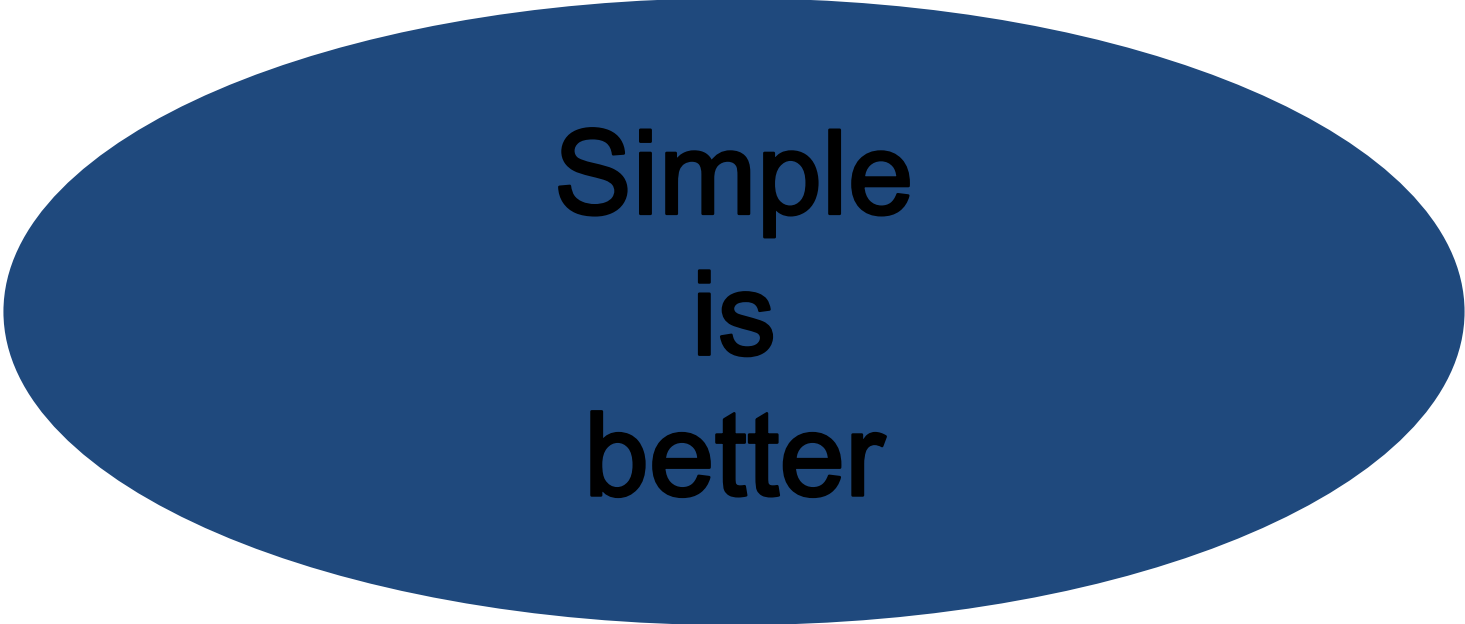
They have datatype `rdfs:langString`



```
:spain rdfs:label "Spain"@en ;  
      rdfs:label "España"@es .
```

...and that's all?

Yes, the RDF Data model is simple

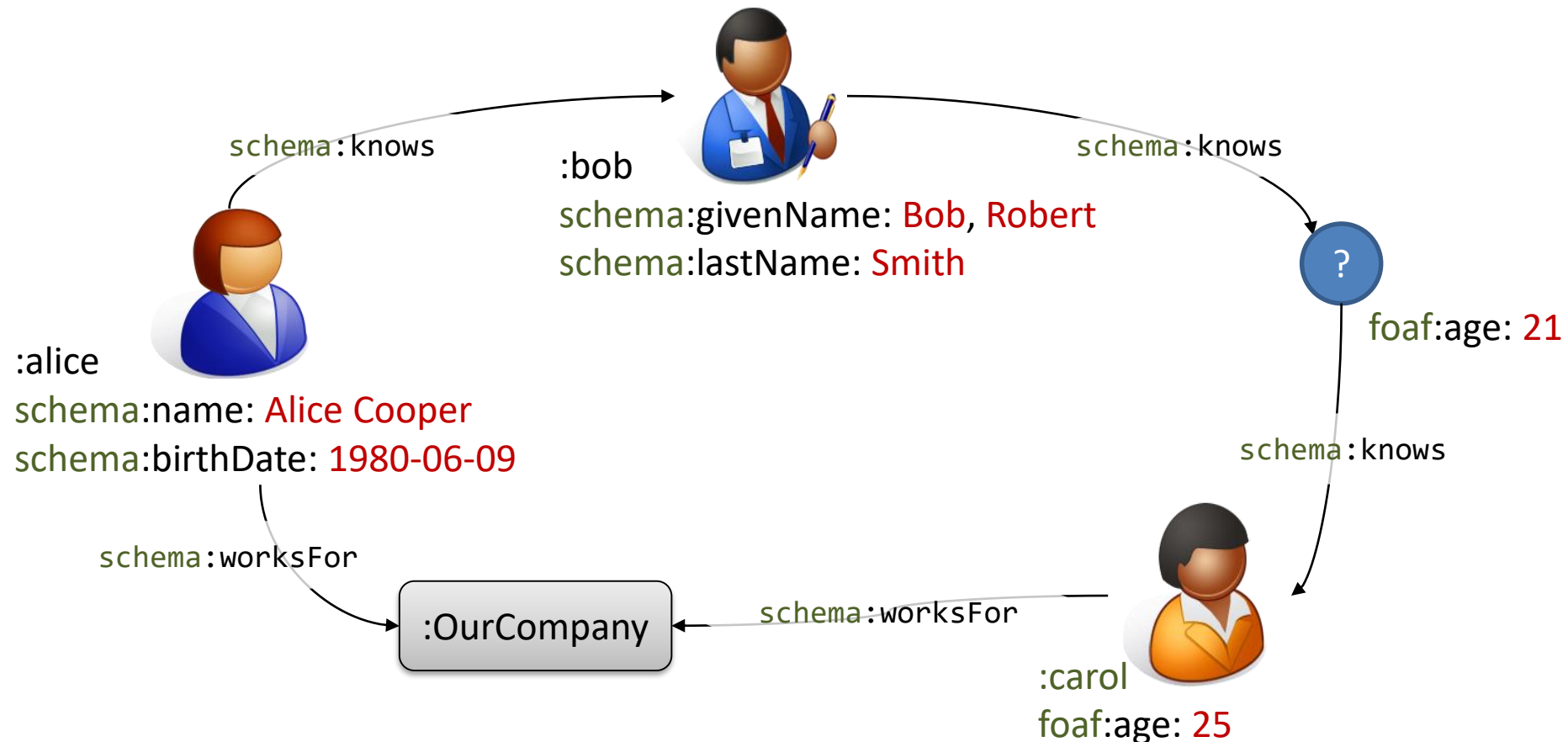


**Simple
is
better**



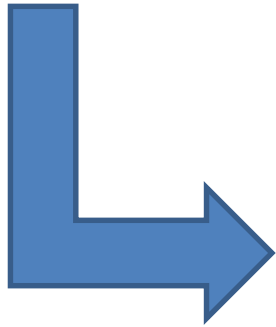
Exercise

Define the following information in RDF



Try it: <https://goo.gl/qzXLp9>

Continue with RDF Validation tutorial



[https://figshare.com/articles/Validating RDF Data/7159802](https://figshare.com/articles/Validating_RDF_Data/7159802)