Ontologies and Knowledge bases

Nicolò Pratelli

Istituto di Scienza e Tecnologie dell'Informazione -
Consiglio Nazionale delle Ricerche – Pisa

A.A. 2024-2025

In seminar 1

In the last seminar you learned how to:

- write RDF using the **Turtle** syntax
- write **triples**
- use **prefixed names** and **relative IRIs**
- use literals with several **datatypes** (strings, dates, numbers, booleans)
- use **language tags**
- express **blank nodes**
- deal with **predicate lists** and **object lists**

In the last lectures

- You learned how to cope with:
  - RDF Schema Vocabulary (RDFS)
    - Classes and subclasses
    - Properties and subproperties
  - Specialization and generalization in RDFS
  - Property typing in RDFS (domain and range)
  - Controlled Vocabularies
    - Thesauri
    - Taxonomies
    - Knowledge Organization Systems (KOS)
    - Ontologies
  - Tbox (Terminological Box) and Abox (Assertional Box)

Notation recap

Class names are written with an **initial uppercase** letter,

while **property** and **individual** names are written with an **initial lowercase** letter

Example	Resource Type	Correct?
ex:Book	Class	✓ Correct
~~ex:person~~ ex:Person	Class	✗ Wrong
~~ex:HasAuthor~~ ex:hasAuthor	Property	✗ Wrong
ex:danteAlighieri	Individual	✓ Correct

RDF Vocabulary

Today we will use only one property from RDF vocabulary:

`rdf:type`

an individual (IRI) is instance of a class (IRI)

Remember that **all resources should have a type**.

For instance, to state that someone is a person, you could write:

```
@prefix : <http://example.org/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .

:sherlock rdf:type :Person .
```

The Special Predicate “a”

In Turtle, you can save space when declare the type of a resource using the special predicate `a`.

The previous triple:

```
:sherlock rdf:type :Person .
```

can be simply written as:

```
:sherlock a :Person .
```

In this case, you don’t need to declare the `rdf:` prefix.

RDFS Vocabulary

Two useful properties from RDFS vocabulary are:

`rdfs:label`
a resource (IRI) has a name (xsd:string)

`rdfs:comment`
a resource (IRI) has a description (xsd:string)

In general, you should **always** give your resources a **label** (name), and it is often a good idea to add a **comment** (description) as well

```
:sherlock rdfs:label "Sherlock Holmes" ;
          rdfs:comment "Sherlock Holmes is a
          fictional detective created
          by Sir Arthur Conan Doyle" .
```

RDFS Vocabulary: other properties

Other useful properties from RDFS Vocabulary are:

`rdfs:subClassOf`
this class (IRI) is a subclass of another class (IRI)

`rdfs:subPropertyOf`
this property (IRI) is a subproperty of another property (IRI)

`rdfs:domain`
this property (IRI) has this class (IRI) as its domain

`rdfs:range`
this property (IRI) has this class (IRI) as its range

Ontologies and KBs

Today we will use classes and properties from two ontologies:

- FOAF (http://xmlns.com/foaf/spec/)
- Dublin Core (https://www.dublincore.org/specifications/dublin-core/dcmi-terms/)

We will also use individuals from one knowledge base:

- Wikidata (https://wikidata.org)

FOAF Ontology

FOAF (Friend of a Friend) is an ontology to linking people and information using the Web. Regardless of whether information is in people's heads, in physical or digital documents, or in the form of factual data, it can be linked. FOAF integrates three kinds of network: *social networks* of human collaboration, friendship and association.

http://xmlns.com/foaf/spec/

You can declare the FOAF namespace like this:

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

Some Terms from FOAF

Today we will use the following FOAF terms:

Classes

`foaf:Person`, *the class of people*
`foaf:Organization`, *the class of organizations*

Properties

`foaf:knows`, *a person (IRI) knows another person (IRI)*
`foaf:name`, *a resource (IRI) has a name (xsd:string)*
`foaf:givenName`, *a person (IRI) has a given name (xsd:string)*
`foaf:familyName`, *a person (IRI) has a family name (xsd:string)*
`foaf:homepage`, *a resource (IRI) has a homepage (IRI)*

Dublin Core Ontology

Dublin Core (DC) is an ontology developed by the Dublin Core™ Metadata Initiative (DCMI) to represent metadata about **digital** and **physical resources**, such as held by libraries and archives.

https://www.dublincore.org/specifications/dublin-core/dcmi-terms/

Dublin Core has 3 namespaces:

```
@prefix dcterms: <http://purl.org/dc/terms/>
@prefix dctypes: <http://purl.org/dc/dcmitype/>
# We will NOT use the third namespace:
@prefix dc: <http://purl.org/dc/elements/>
```

Some Terms from Dublin Core

Today we will use the following DC terms:

Classes

`dctypes:Text`, *the class of texts (books, poems, letters...)*
`dctypes:Event`, *the class of events*

Properties

`dcterms:creator`, *a resource (IRI) has a creator (IRI)*
`dcterms:title`, *a resource (IRI) has a title (xsd:string)*
`dcterms:date`, *an event (IRI) happened on a date (xsd:date)*
`dcterms:issued`, *a resource (IRI) was published on a date (xsd:date)*

`dcterms:language`, *a resource (IRI) has a language (IRI)*

Wikidata Knowledge Base

Wikidata is a giant general-purpose **knowledge base** (> 89 million resources and > 1 billion statements).
It is a **free** and **open** project (similar to Wikipedia), and its data can be read and edited by both humans and machines.

Wikidata acts as central storage for the **structured data** of its Wikimedia sister projects including Wikipedia, Wikivoyage, Wiktionary, Wikisource, and others.

Today, we will use Wikidata as a source of IRIs for **individuals** (specific people, objects, events...)

https://www.wikidata.org/

Wikidata IRIs

The Wikidata repository consists mainly of items, each one having a label, a description and any number of aliases. Items are uniquely identified by a **Q** followed by a number, such as *Sherlock Holmes* (Q4653). Properties in Wikidata have a **P** followed by a number, such as with *educated at* (P69)

Wikidata IRIs have the following form:

<http://wikidata.org/entity/Q4653>
(this is the IRI for the resource “Sherlock Holmes”)

The following IRI should **not be used** in your Turtle files:

<http://wikidata.org/wiki/Q4653>

This IRI points to the Wikidata page, not to the RDF data

prefix.cc

The **prefix.cc** website is a useful resource when you don’t remember the correct IRI of a certain prefix.

By loading the website and searching for a prefix name, (such as rdfs or foaf), you will receive the full IRI

https://prefix.cc/

Instructions for Exercises

- For individuals, use Wikidata IRIs with the following prefix:
  `@prefix wd: <http://wikidata.org/entity/>`
- Only for the underlined words, define your own class or property using the default prefix with IRI <http://example.org/> and, if possible, state that it is **subclass** or **subproperty** of a term from FOAF or DC
- For all other classes and properties, use directly one of the terms from RDF, RDFS, FOAF, or DC that we learned today
- For literals, use the appropriate XSD datatypes
- Validate your code (http://ttl.summerofcode.be), then post the solutions on Moodle

Example Exercise

- Use the Turtle syntax to represent a set of facts extracted from the following text: “A Study in Scarlet is a book (subclass of text) written by Arthur Conan Doyle (a person), published in 1887.

```
@prefix : <http://example.org/> .
@prefix wd: <http://wikidata.org/entity/> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix dctypes: <http://purl.org/dc/dcmitype/> .
wd:Q223131 a :Book ;
           dcterms:title "A Study in Scarlet" ;
           dcterms:creator wd:Q35610 ;
           dcterms:issued "1887-01-01"^^xsd:date .
:Book rdfs:subClassOf dctypes:Text .
wd:Q35610 a foaf:Person .
```

Exercises

Use the **Turtle syntax** and the appropriate classes and properties from the ontologies presented in this seminar to represent 2 separate sets of facts extracted from the following texts:

- Sherlock Holmes is a detective (subclass of person); he has a given name, a family name, and a label. He is best friend of (subproperty of knows) Doctor Watson (a doctor, subclass of person) and is worst enemy of (subproperty of knows) Professor Moriarty (a professor, subclass of person). Doctor Watson has a given name (John), a family name, and a label. Professor Moriarty has a given name (James), a family name, and a label.
- Dante is a writer (subclass of person) and a poet (subclass of person). He is the creator of the "Divina Commedia", a text published in 1309 in Italian language; the "Convivio", a text published in 1304 in Italian language; the "Vita Nova", a text published in 1293 in Italian language; the "Monarchia", a text published in 1312 in Latin language. Each work and each language have a label. Dante has a given name, a family name, and a label.