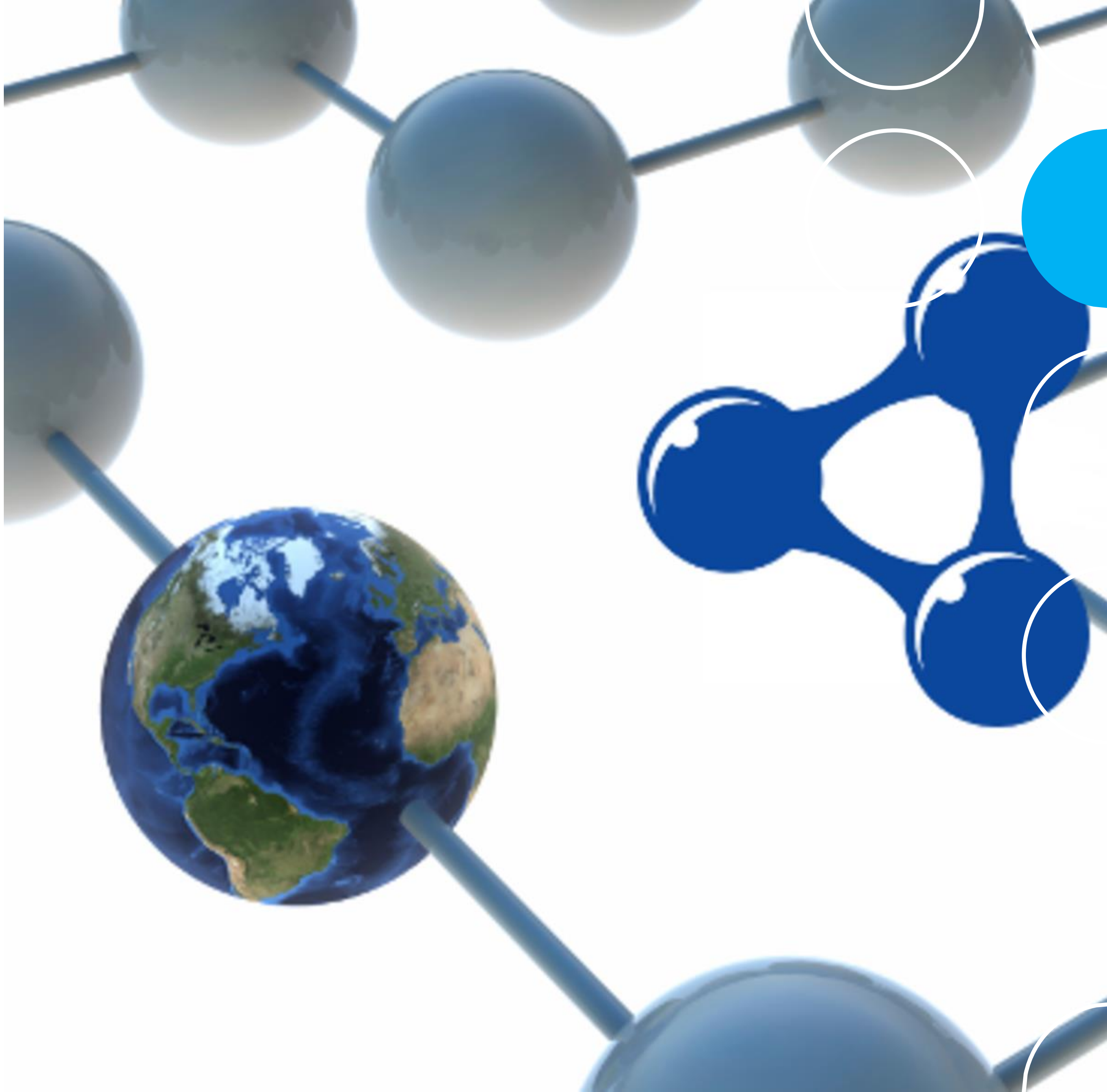
Semantic Web and Linked Data

Liliana Ferreira

2023/24



Class 3: Learning Objectives

- Understand RDF principles and how to provide useful RDF information;
- Familiarise with existing RDF data sets;
- Learn to read and write RDF;
- Train modelling information using a graph-based data model;

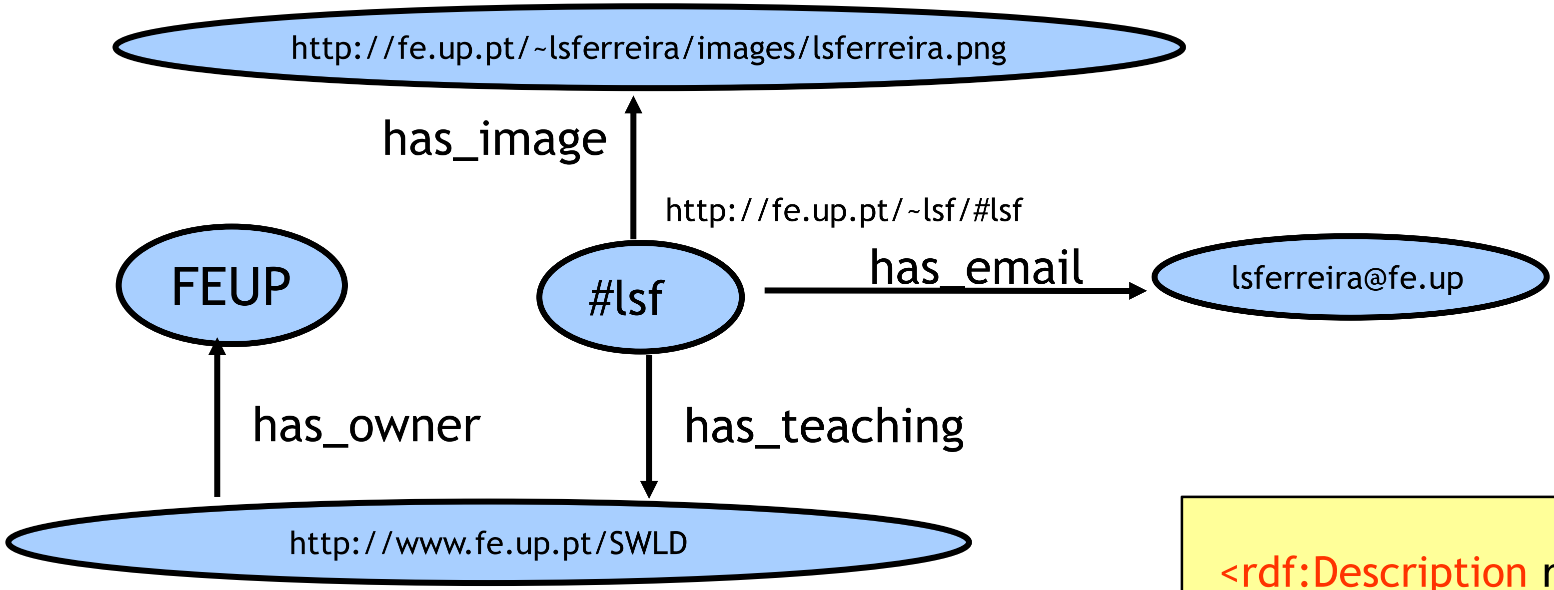
RDF basics

- The core features of RDF are:
  - **Identify** things (resources)
  - Express **relations** between things (properties)
- Additional important features are:
  - Assign **data values** to things (literals)
  - Organize things in **categories** (i.e., classes or types)
  - Add **simple knowledge** about categories and relations

Composition Rules for RDF Triples

1. The subject is always a resource (and not a literal)
2. The type of the binary property is identified by a URI
3. The value is a resource or a literal

RDF triples form graphs




```
<rdf:Description rdf:about="#lsf">
   <has_email>lsferreira@fe.up</has_email>
</rdf:Description>
```

Identify things

Resources

- RDF is used to **describe resources**
- A ***resource*** can be anything (real or imaginary entity, abstract or concrete)
- Every resource has a URI (Universal Resource Identifier) or IRI (Internationalized Resource Identifier)
- A URI/IRI **identify** things but *may* be used as **locators** (URL, a web address) at the same time;
- An identifier does not necessarily enable access to a resource;
- To describe a resource, it must be named or identified
- On the Web, the identification mechanism must be uniform at Web scale: an identifier must identify the same thing **everywhere** on the Web

Examples:

- ✓ `urn:ietf:rfc:3987`
- ✓ `svn://yadiyada.foo.bar/`
- ✓ `mailto:lsferreira@fe.up.pt`
- ✓ `ftp://ftp.up.pt/#meta`
- ✓ `http://en.wikipedia.org/wiki/User:Wikiuser100`

Note: to shorten notations, we use namespace **prefixes**, e.g.,
`rdf:` is for `http://www.w3.org/1999/02/22-rdf-syntax-ns#`

How to choose an IRI?

- If possible, reuse an existing IRI from an authoritative source, e.g.:
  - from a national library
  - from a government website, a ministry
- If not, make your own IRI:
  - use HTTP IRIs
  - use a namespace under your control
  - Cool URIs don’t change
  - Refer to the guide on Cool URIs for the Semantic Web

Relate things

Predicate

- A ***predicate*** is a specific aspect of a resource.
- It can be a characteristic that belongs to a resource, or a relationship that links one resource with another: "João **studies** at FEUP"
- Predicates describe relations between resources;
  - For example: "written by", "composed by", "title", "topic", etc;
- The predicate in RDF is also identified by IRIs. This provides a global, unique naming scheme.

Example:

http://example.org/data/Joao , subject
http://social.relations.com/knows ,predicate
http://example.org/data/Francisco , object

Data values

- As everything else, a data value (number, string, date) is a resource
- A specific data value can be identified with a **literal**, a character string that represents the value
- Every literal is typed such that its string representation can be interpreted as the correct value
- Usually, we use standard data type IRIs from the xsd: namespace (XML Schema Datatypes) and the rdf: namespace

Example:

- E.g., "42" represents the number fourty two if this is of type decimal integer, but represents sixty six if it is an hexadecimal integer

Resource Description Framework (RDF)

Statements

- A ***statement*** is a piece of description about a particular resource in the RDF format: an object-attribute-value triple;
- It consists of a resources, a property, and a value (subject, predicate, object)

http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=10140

Resource Description Framework (RDF)

- A statement about a resource instance has:
  - the resource's identifier
  - one of the resource's property (defined in an RDF schema)
  - the value for that property (can be either a literal, or a resource)

```
<rdf:RDF
  xmlns:wc=”http://www.lapd.fe.up.pt/~exRDF/wc/schema”>
  <rdf:Description about=”http://www.cnn.com/2000/HEALTH/cancer/12/06/
            colon.cancer.ap/index.html”>
  <wc:Title>Cigarette smoking linked to colorectal cancer </wc:Title>
  </rdf:Description>
</rdf:RDF>
```

RDF: triples form graph edges

(subject, predicate, object)

->

(node, edge, node)

- An RDF graph is a set of RDF triples
- RDF graphs can be drawn as directed, edge-labelled multigraphs

RDF is an oriented labeled multigraph model

- RDF is an oriented labeled multigraph model
    1. Several edges can connect the same two nodes;
    2. Edges are oriented: the head is the object, the tail is the subject;
    3. Edges and nodes are labeled.

RDF graphs as files

- An RDF graph (a set of RDF triples) can be **serialised** to multiple syntaxes (similarly to tabular data that can be written to CSV, Excel, HTML tables, SQL…)
- In WSLD 2023-24, we will mostly use [Turtle]
- Others:
  - There is an XML-based syntax: RDF/XML
  - There is a JSON-based syntax: JSON-LD
  - There is an easy to parse, line-based triple syntax: N-Triples
  - There is a syntax to embed RDF in HTML and XML documents: RDFa
- Let us learn to read and write Turtle in an online editor. Go to: https://perfectkb.github.io/yate/

The Turtle syntax

- Full IRIs:

```
<http://www.example.com/test#this>
```

- A simple triple:

```
    <http://www.example.com/test#this>
              <http://relations.example.com/in>
                             <http://www.example.com/test#box> .
```

- Abbreviated IRIs (declare prefixes at the beginning of the file):

```
# This is a comment
@prefix ex: <http://www.example.com/test#> . # end dot!
@prefix rel: <http://relations.example.com/> .
ex:this rel:in ex:box . # Another comment
```

The Turtle syntax

- Literals:

```
ex:this rel:date "2019-09-13"^^xsd:date . # normal literal
ex:this rel:name "this"@en . # language-tagged literal
ex:this rel:code "TX32" . # xsd:string can be omitted
ex:this rel:number 42 . # xsd:integer (no quotes)
ex:this rel:sizeInMeters 3.75 . # xsd:decimal (use a dot)
```

The Turtle syntax

- Repeat the same subject and predicate:

```
ex:box rel:contains ex:this .
ex:box rel:contains ex:that .
# can be written
ex:box rel:contains ex:this, ex:that . # comma
```

The Turtle syntax

- Repeat subject

```
ex:this rel:date "2019-09-13"^^xsd:date;
rel:name "this"@en; # new lines are optional
rel:code "TX32";
rel:nextTo ex:that, ex:thoot, ex:thus .
```

Further reading

- Semantic Web Stack
- RDF 1.1 Primer
- RDF 1.1 Concepts and Abstract
- https://www.w3.org/TR/turtle/

- https://www.w3.org/RDF/Validator/