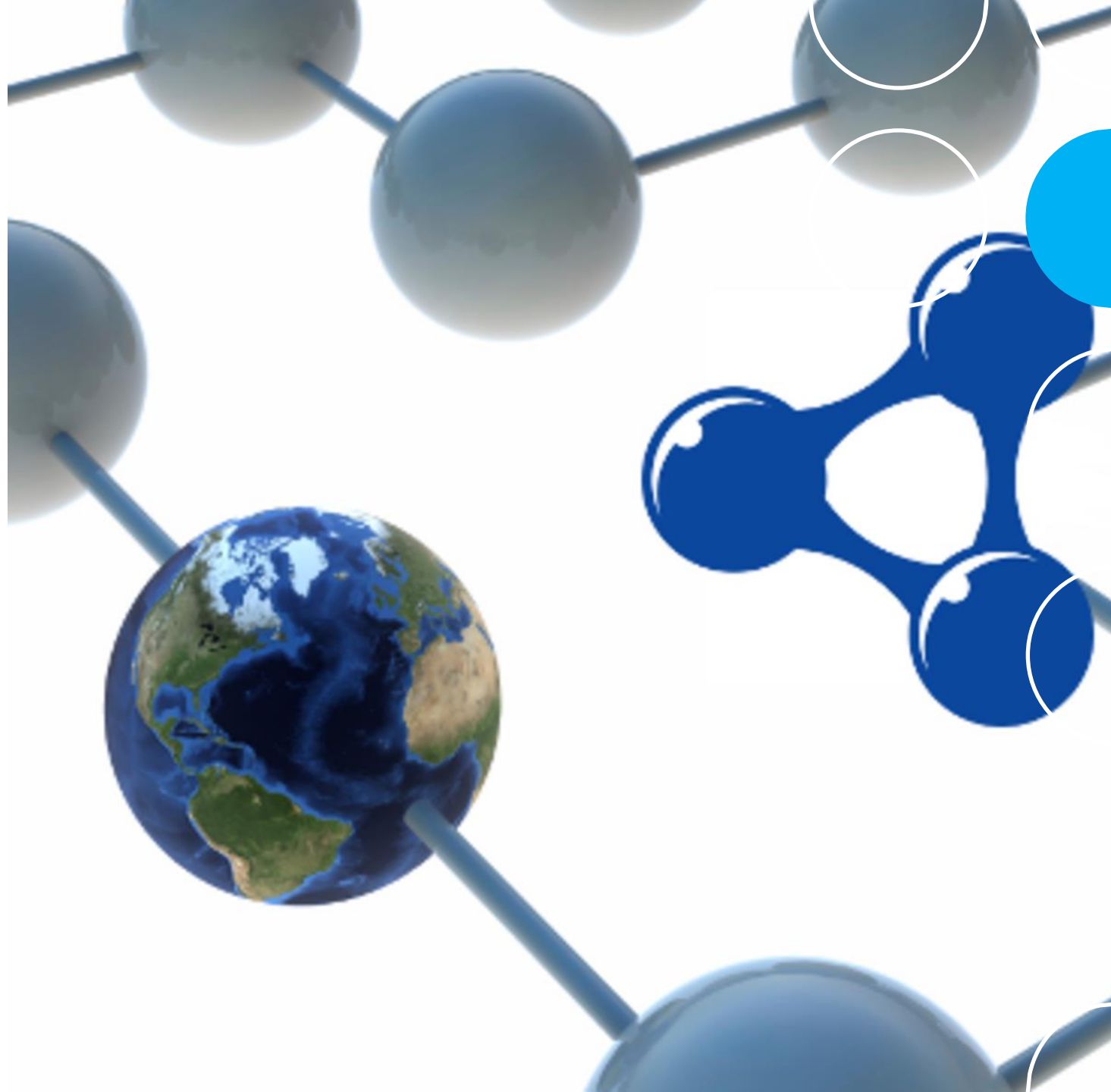


# Semantic Web and Linked Data

Liliana Ferreira

2023/24

---





# Class 4: Learning Objectives

- Review RDF principles;
  - RDF Schema.
  
  - Exercises
  - Practical Work
  
  - Introduction to SPARQL
-

# RDF Syntax

- The RDF data model provides an abstract, conceptual framework for defining and using metadata.
  - A concrete syntax is also needed for the purposes of creating and exchanging this metadata.
-

# RDF Vocabulary

- RDF defines a number of resources and properties;
- RDF vocabulary is defined in the namespace:
  - <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

The vocabulary defined by the RDF specification is as follows:

- **Classes:**
  - `rdf:Property`, `rdf:Statement`, `rdf:XMLLiteral`
  - `rdf:Seq`, `rdf:Bag`, `rdf:Alt`, `rdf>List`

# RDF Vocabulary

- **Properties:**
  - `rdf:type`, `rdf:subject`, `rdf:predicate`, `rdf:object`,
  - `rdf:first`, `rdf:rest`, `rdf:_n`
  - `rdf:value`
- **Resources:**
  - `rdf:nil`

# RDF Vocabulary

## Classes & Resources

- `rdf:XMLLiteral` - the class of XML literal values,
  - `rdf:Property` - the class of properties,
  - `rdf:Statement` - the class of RDF statements,
  - `rdf:Alt`, `rdf:Bag`, `rdf:Seq` - containers of alternatives, unordered containers, and ordered containers (`rdfs:Container` is a super-class of the three),
  - `rdf:List` - the class of RDF Lists,
  - `rdf:nil` - an instance of `rdf:List` representing the empty list.
-

# RDF Vocabulary

## Properties

- `rdf:type` - an instance of `rdf:Property` used to state that a resource is an instance of a class,
  - `rdf:first` - the first item in the subject RDF list,
  - `rdf:rest` - the rest of the subject RDF list after `rdf:first`,
  - `rdf:value` - idiomatic property used for structured values,
  - `rdf:subject` - the subject of the RDF statement,
  - `rdf:predicate` - the predicate of the RDF statement,
  - `rdf:object` - the object of the RDF statement.
-

# RDF Vocabulary

- Typing using `rdf:type`:

`<A, rdf:type, B>`

“A belongs to class B”

- All properties belong to class `rdf:Property`:

`<P, rdf:type, rdf:Property>`

“P is a property”

`<rdf:type, rdf:type, rdf:Property>`

“`rdf:type` is a property”

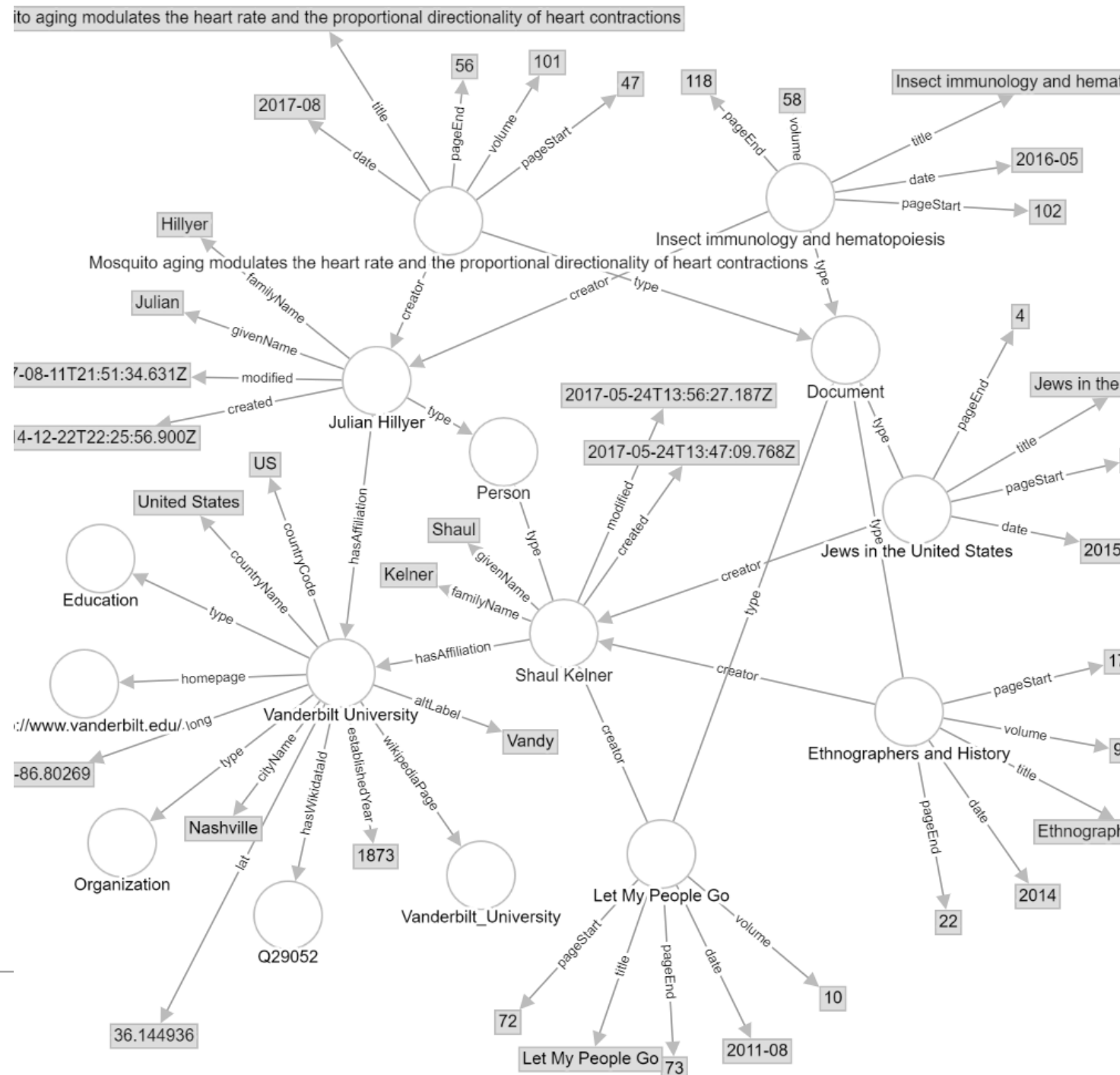


# RDF Serializations and Triplestores

- Since RDF is an abstract model for expressing information about graphs, it can be expressed in a number of concrete ways.
  - One way that is particularly easy for humans to understand is a graphical diagram.
-

# RDF Serializations

- The triples in [this table](#) form a graph that can be represented by this diagram.



# RDF Serializations

- However, it is generally not possible for machines to interpret graphs that are expressed as diagrams.
  - Machines need an RDF *serialization*, a method of transmitting or storing the information about the triples in the graph as a file.
-

# RDF graphs as files

- In WSLD, 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
-

# The Turtle RDF syntax

- Turtle stands for “**Terse RDF Triple Language**”.
  - N-Triples is a subset of the RDF Turtle serialization, meaning that any file that is valid N-Triples is also valid Turtle serialization.
  - However, Turtle allows compact URIs (CURIes) and also allows shortcuts to prevent repeating parts of triples.
-

# The Turtle RDF syntax

- For example, if several triples share the same subject, the predicates and objects can be listed, separated by semicolons.
-

# 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 RDF syntax

- The namespace prefixes that are used in the triples must be listed in a prolog at the start of the document.
  - Notice that URIs aren't required to be abbreviated.
-



# 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

- If two triples share both the same subject and predicate, the two objects can be separated by commas. For example:

```
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 object

```
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 .
```

---

# The Turtle syntax

- Turtle also allows a special abbreviation for the important predicate `rdf:type`. It can be replaced with `a`.
- Hence, the triple:

```
<http://dbpedia.org/resource/Bob_Marley> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
```

can be shortened in Turtle to:

```
dbr:Bob_Marley a foaf:Person
```

# The Turtle syntax

- RDF text files in Turtle serialization are usually given the file extension **.ttl**
  - Let us learn to read and write Turtle in an online editor. Go to:  
<https://perfectkb.github.io/yate/>
-

# REMEMBER!

- RDF is not an ontology language but a data model (!!!)
    - RDF is a W3C Recommendation
    - RDF is designed to be read by computers
    - RDF is for describing resources on the Web
    - RDF uses URIs to identify and reference resources on the Web
  - RDF/XML is just one way of serializing RDF. Other serializations format include TURTLE and N3.
  - NQuads and Trig even support (named) graphs.
-

# Suitable Text Editors

- Remember, **RDF is a data model.**
  - Ontologies require ontology languages (such as the Web Ontology Language described later on) for which adequate tools exist.
  - For RDF, however, most adequate tooling performs syntactic checks rather than semantic checks.
-

# Suitable Text Editors

- Two fairly known editors with support for RDF (via plugins) are
    - [Visual Studio Code](#) and
    - [Atom](#).
  - The former might be more lightweight, easier to install, and proposes the installation of plugins upon or saving files of a particular type.
-

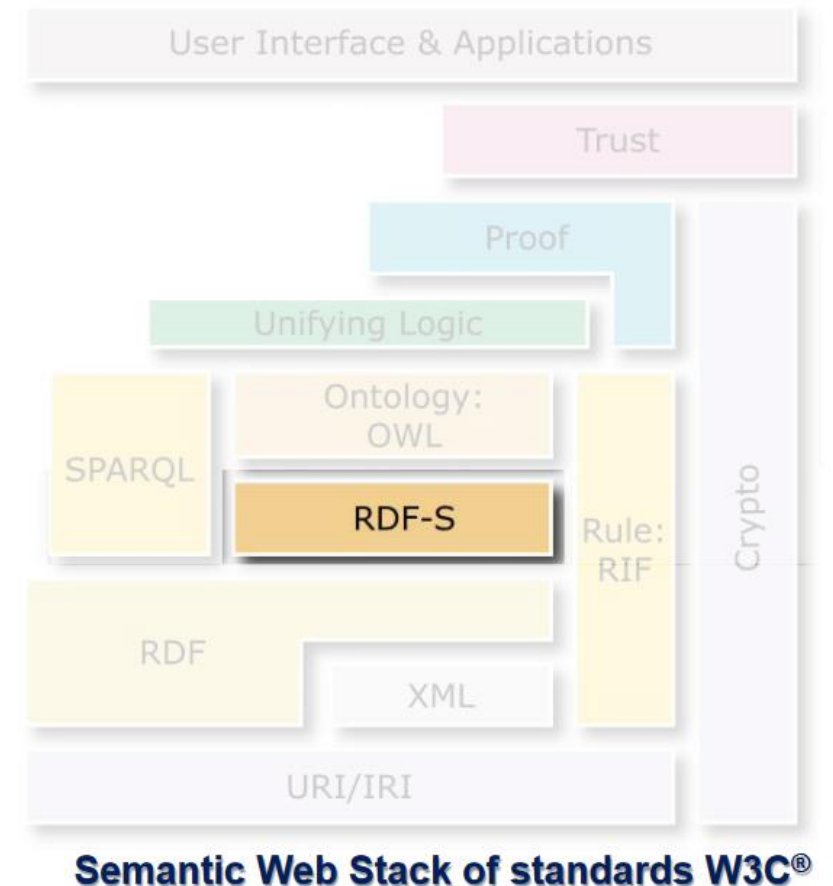


# Class 4: Learning Objectives

- Review RDF principles;
  - **RDF Schema.**
  
  - Exercises
  - Practical Work
  
  - Introduction to SPARQL
-

# RDF Schema

- To represent light-weight ontologies in RDF;
- RDFS provides standard vocabulary to declare *in* RDF vocabularies to be used in RDF descriptions;
- RDFS reuses the vocabulary of RDF and introduces additional constructs;
- An RDF vocabulary is a set of property declarations and class declarations.



# RDF Schema

- RDF(S) is a W3C Recommendation
  - RDF(S) is an extension of RDF
  - RDF(S) provides a framework to describe vocabularies
  - RDF(S) describe resources with classes, properties and values
-

# Associating a Namespace to a Vocabulary

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

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
@base <http://fe.up.pt/2021/students.rdfs>
```

```
(...)
```

---

# RDF(S) Classes

<b>rdfs:Resource</b>	RDF(S) top element, all other classes are derived from this
<b>rdfs:Class</b>	The Class class
<b>rdf:Property</b>	Base class for properties
<b>rdfs:Literal</b>	The base class for literal values. Allows literal values such as strings and integers
<b>rdfs:Datatype</b>	The base class of data types

---

# RDF(S) Properties

<b>rdfs:subClassOf</b>	Indicates the subject is a subclass of the object in a statement.
<b>rdfs:subPropertyOf</b>	The subject is a sub-property of the property.
<b>rdfs:comment</b> <b>rdfs:label</b>	Simple properties that take string literals as values. Labels refer to <i>human-readable</i> versions of a resource's <i>name</i> and a comment provides a human-readable <i>description</i> of a resource.
<b>rdfs:domain</b>	Used to state that any resource that has a given property is an instance of one or more classes.
<b>rdfs:range</b>	Used to state that the values of a property are instances of one or more classes.
<b>rdfs:isDefinedBy</b>	Points to the human readable definition of a class, usually a URL.

# Declaring Classes of Resources

- Naming classes;
- Organizing them into hierarchies.

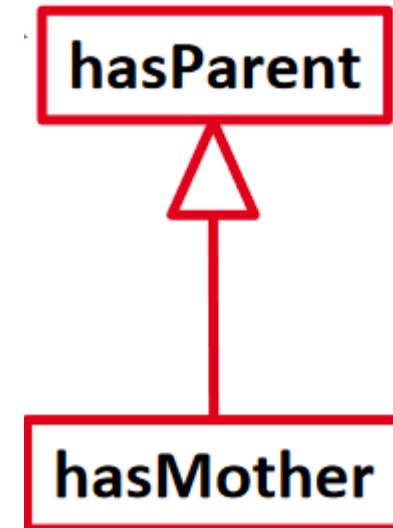
```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
@base <http://fe.up.pt/2021/students.rdfs>  
<Woman> a rdfs:Class ;  
      rdfs:subClassOf <Person>, <Female> .
```



# Declaring Types of Properties

- Naming types of properties
- Organizing them into hierarchies

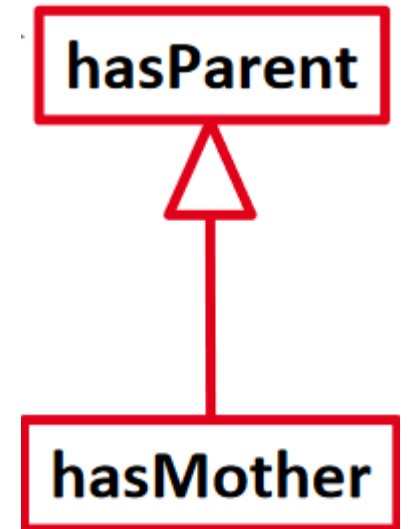
```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>  
@base <http://fe.up.pt/2021/students.rdfs>  
<hasMother> a rdf:Property ;  
rdfs:subPropertyOf <hasParent> .
```





# Declaring Property Signatures

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
@base <http://fe.up.pt/2021/students.rdfs>
<hasMother> a rdf:Property ;
    rdfs:subPropertyOf <hasParent> ;
    rdfs:domain <Person> ;
    rdfs:range <Woman> .
```



# Documenting Class and Property Declarations

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

```
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>
```

```
@base <http://fe.up.pt/2021/students.rdfs>
```

```
<Woman> a rdfs:Class ;  
rdfs:label "woman"@en ;  
rdfs:comment "an adult female person"@en .
```

```
<hasMother> a rdf:Property ;  
rdfs:label "has for mother"@en ;  
rdfs:comment "to have a woman for mother"@en .
```

---

# Referencing and Using Schemas

in the description of a resource

```
@prefix h: <http://fe.up.pt/2021/students.rdfs#>  
@base < http://fe.up.pt/2021/students.rdfs-instances >  
<Alice> a h:Woman; h:hasMother <Laura> .
```

---

# Further reading RDFS

- [RDF Schema 1.1](#)
  - [RDF Schema on Wikipedia](#)
  - To choose/find a schema, use the [Linked Open Vocabularies](#) (LOV) service
  - To find a schema with a prefix frequently used for it, use the [prefix.cc](#) service
-

# Class 4: Learning Objectives

- Review RDF principles;
  - RDF Schema.
  
  - **Exercises**
  - **Practical Work**
  
  - Introduction to SPARQL
-