

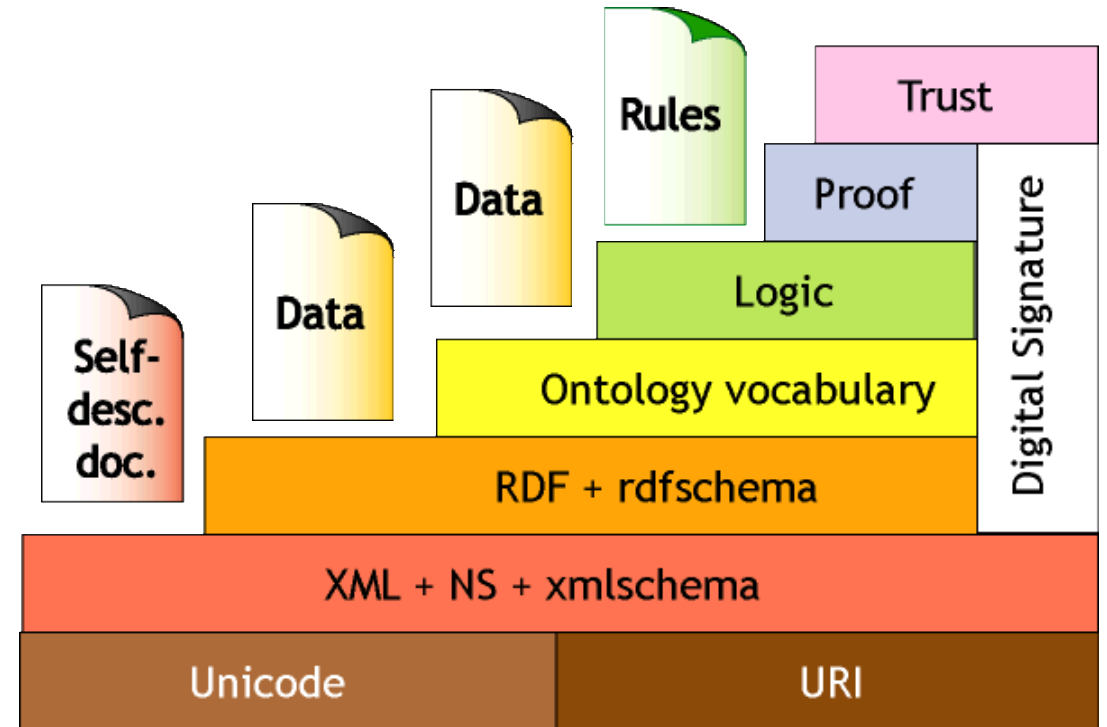
Semantic Web Technologies

Semantic web technologies

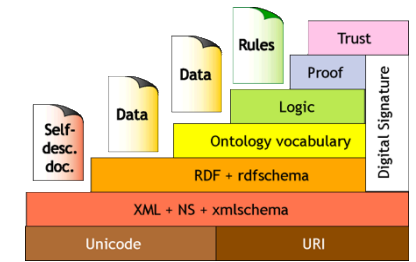
- In order to achieve the objective of web of data (linked data), W3C provides a set of technologies
- These range from using IRIs to identify entities to higher level tools for proof and verification
- Layers, each building on the layers below
 - Addressing and coding
 - Format
 - Data representation
 - Model representation
 - Inference

Semantic web stack of technologies

- **Formatting language:** XML (eXtensible Markup Language): markup language where user can define own tags
- **Basic data model:** RDF Schema, defining RDF vocabularies
- **Ontology modelling:** OWL (Web Ontology Language)
- **Logic layer:** application-specific declarative knowledge; rules that enable inferences
- **Proof layer:** to allow the explanation of given answers by automated software agents (**not covered**)
- **Trust Layer:** use digital signatures to authenticate services and users, use encryption for transfer of sensitive data, etc. (**not covered**)



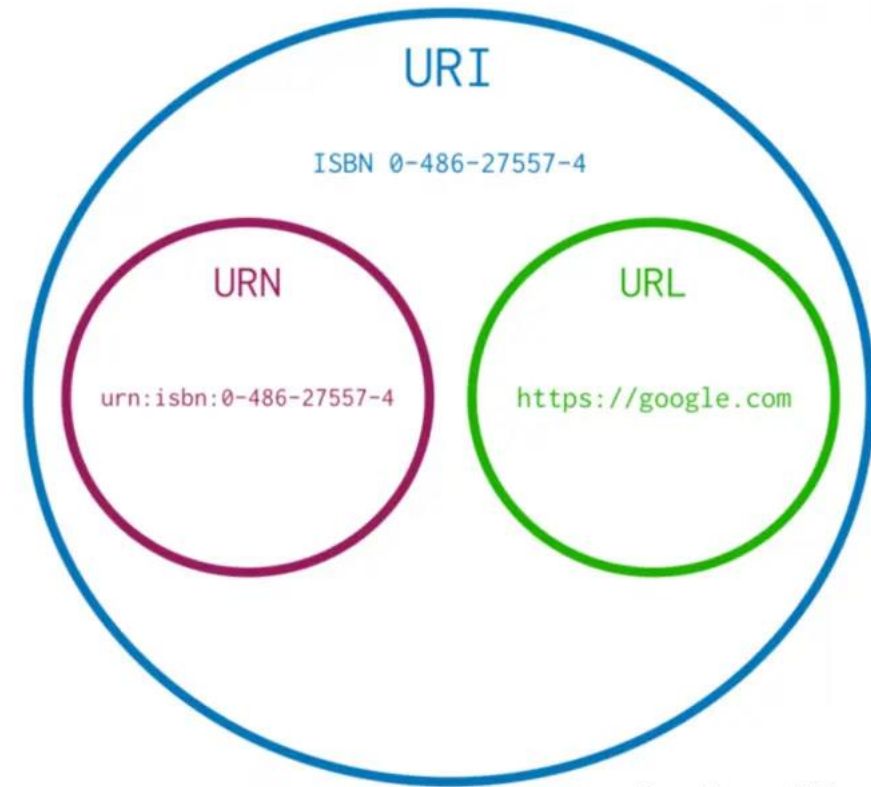
Namespace (IRIs)



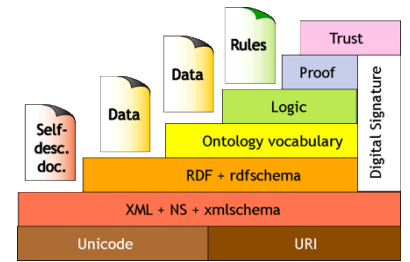
- The abbreviation IRI is short for "International Resource Identifier". An IRI identifies a resource.
- Naming things at different organisation may lead to collisions
 - Developer vs. developer (in business and software)
- There is a need to define namespaces
- Use IRIs to identify things
 - <http://myonto.hw.ac.uk/business#developer>
 - <http://myonto.hw.ac.uk/software#developer>
 - For example, DBpedia uses IRIs of the form `http://dbpedia.org/resource/Name` to denote the thing described by the corresponding Wikipedia article.

Identifier schemes

- IRI: Internationalized Resource Identifier, e.g.
 - Chinese Symbols
`http://例子.卷筒纸` or
 - Japanese Characters
`http://example.com/引き割り.html`
- CURIE: Compact URI
 - Abbreviated syntax for IRIs
 - Often based on IRI fragment
 - Examples:
 - `isbn:0393315703`
 - `doi:10.1038/sdata.2016.18`

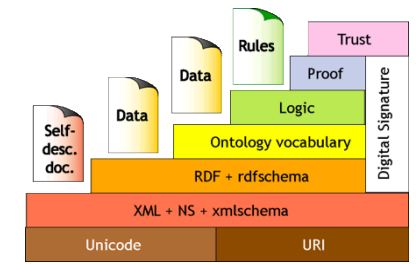


RDF (The Resource Description Framework)



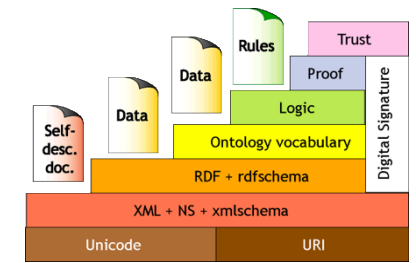
- A language (used with RDF schema) to model data
- It is a set of globally defined concepts
- Allows describing a knowledge graph: graph that links entities(resources) through relationships (properties)
- RDF defines **triples**
 - A labelled connection between two resources

OWL



- Web Ontology Language (W3C)
 - More sophisticated data modelling language
 - Builds on RDF
 - Allows defining and using own entities (classes) and relationships (properties)
 - Models defined by OWL are called ontologies
- **Folksonomy**: community chooses terms (no relationships)
 - **Controlled Vocabulary**: list of predefined terms (no relationships)
 - **Taxonomy**: CV + hierarchy
 - **Thesaurus**: Taxonomy + other relationships, e.g., synonyms
 - **Ontology**: List of terms, with relationships... and the ability to define your own terms AND relationships

Logic layer



- Enables definition of rules for inference and validation
 - Rules come from the ontologies
 - Languages such as SWRL (semantic web rule language) can be used
- Inference example:
 - Socrates is a human
 - All humans are mortal
 - What can we infer as non-stated relationship
 - Socrates is mortal
 - Now, what would be the answer to: are you mortal?

Inference assumptions

Open world assumption

- Assume our knowledge is incomplete, so everything not listed may be true OR false. i.e., its validity is unknown
- If a statement is not in the database, or cannot be inferred, then its validity is unknown
- Example
 - Socrates is a human
 - All humans are mortal
 - \Rightarrow Socrates is mortal
- Truth
 - Socrates is mortal : true
 - John is mortal: unknown

Closed world assumption

- Assume our knowledge is complete, so everything not listed is false
- Truth
 - Socrates is mortal : true
 - John is mortal: false
- *Relational databases apply the closed world assumption*

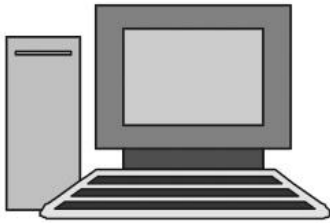
Solution to the inference question

- Given the Socrates example, *Are you mortal?*
- We consider an Open World Assumption
- We don't know if interlocutor a human or a chatbot?
- Therefore, we don't know if you are mortal

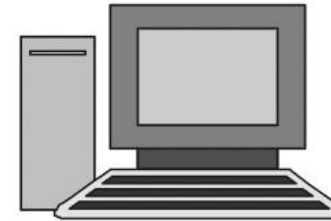
RDF: Resource Description Framework

ID	Title	Author	Medium	Year
1	<i>As You Like It</i>	Shakespeare	Play	1599
2	<i>Hamlet</i>	Shakespeare	Play	1604
3	<i>Othello</i>	Shakespeare	Play	1603
4	"Sonnet 78"	Shakespeare	Poem	1609
5	<i>Astrophil and Stella</i>	Sir Phillip Sidney	Poem	1590
6	<i>Edward II</i>	Christopher Marlowe	Play	1592
7	<i>Hero and Leander</i>	Christopher Marlowe	Poem	1593
8	<i>Greensleeves</i>	Henry VIII Rex	Song	1525

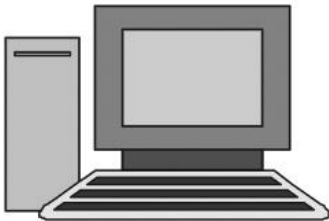
RDF: Resource Description Framework



1	As You Like It	Shakespeare	Play	1599
---	----------------	-------------	------	------



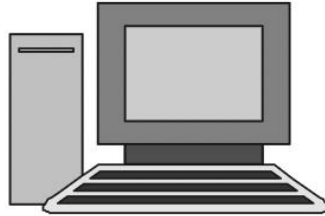
Needs common schema—which column is which?



4	Sonnet 78	Shakespeare	Poem	1609
6	Edward II	Christopher Marlowe	Play	1592

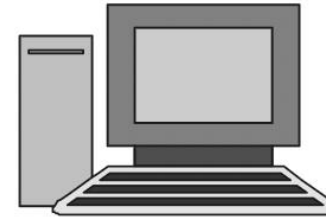
7	Hero and Leander	Christopher Marlowe	Poem	1593
3	Othello	Shakespeare	Play	1603

RDF: Resource Description Framework



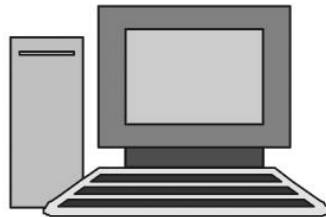
Year	Medium
1599	Play
1604	Play
1603	Play
1609	Poem
1590	Poem
1592	Play
1593	Poem
1525	Song

Needs to reference entities—which thing are we talking about?

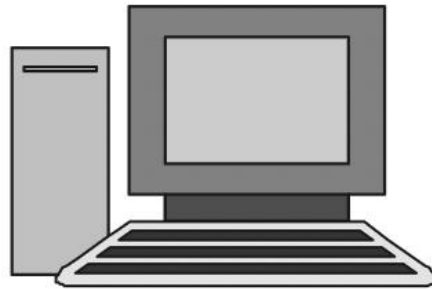


Author
Shakespeare
Shakespeare
Shakespeare
Shakespeare
Sir Phillip Sidney
Christopher Marlowe
Christopher Marlowe
Henry VIII Rex

Title
<i>As You Like It</i>
<i>Hamlet</i>
<i>Othello</i>
"Sonnet 78"
<i>Astrophil and Stella</i>
<i>Edward II</i>
<i>Hero and Leander</i>
<i>Greensleeves</i>



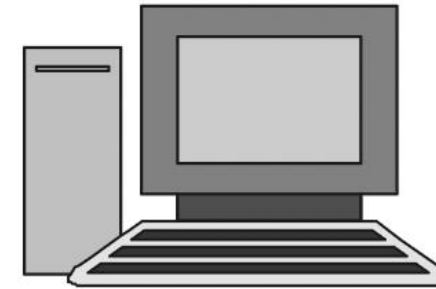
RDF: Resource Description Framework



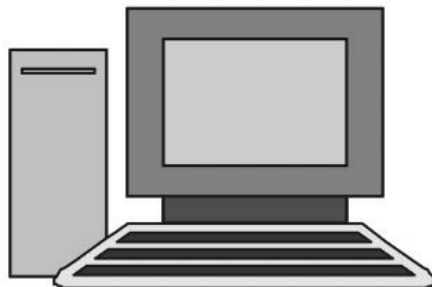
	Medium
Row 7	Poem

	Title
Row 2	Hamlet

*Needs to reference both
schema and entities*



	Author
Row 4	Shakespeare

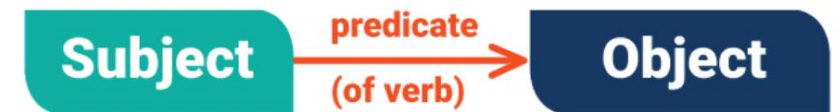
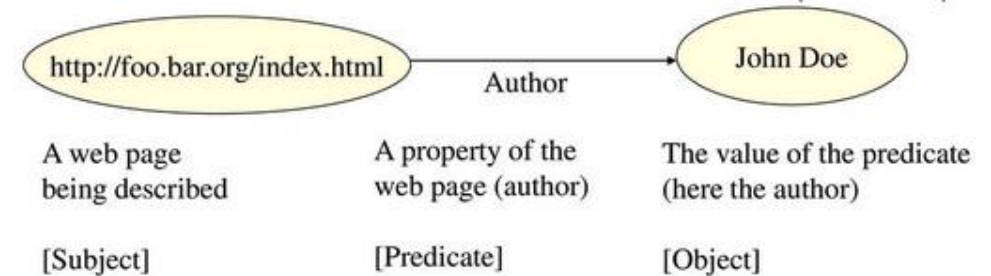


	Year
Row 2	1604
	Medium
Row 6	Play

RDF: Resource Description Framework

- W3C standard <http://www.w3.org/TR/rdf11-primer/>
- A data model rather than a language: we can say that an object possesses a **particular property**, or that it has a **named relationship** with another object.
- RDF defines **triples**
 - A labelled connection between two resources
- A triple (s,p,o) – (subject, predicate, object)
 - s: subject resource (an IRI)
 - p: predicate defining a relationship (an IRI)
 - o: object can be resource (IRI) or literal (text, number, date...)
- Uses IRIs
- Represents **directed labelled graph**
- It is schema-free: it supports multiple vocabularies

An RDF statement (or rule)



RDF vs XML vs HTML

- HTML

- `<p>HTML` merely tells the computer how to present text`</p>`
- The computer has no understanding of what the text means

- XML

- `<info>XML encodes information in user defined tags</info>`
- Tags can be nested, thus creating a structure, which may have meaning

```
<name>
```

```
  <given>Ken</given>
```

```
  <family>McLeod</family>
```

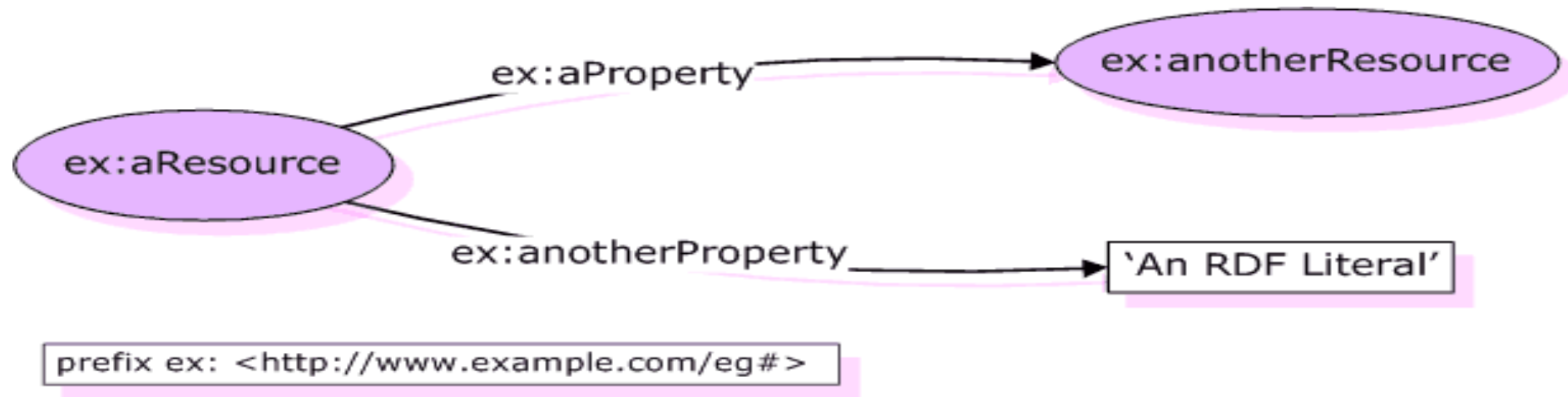
```
</name>
```

- No built mechanism to convey the meaning of the user's new tags to other users.

- RDF

- Provides extra information (relationships between resources) that the computer can interpret and use for inference

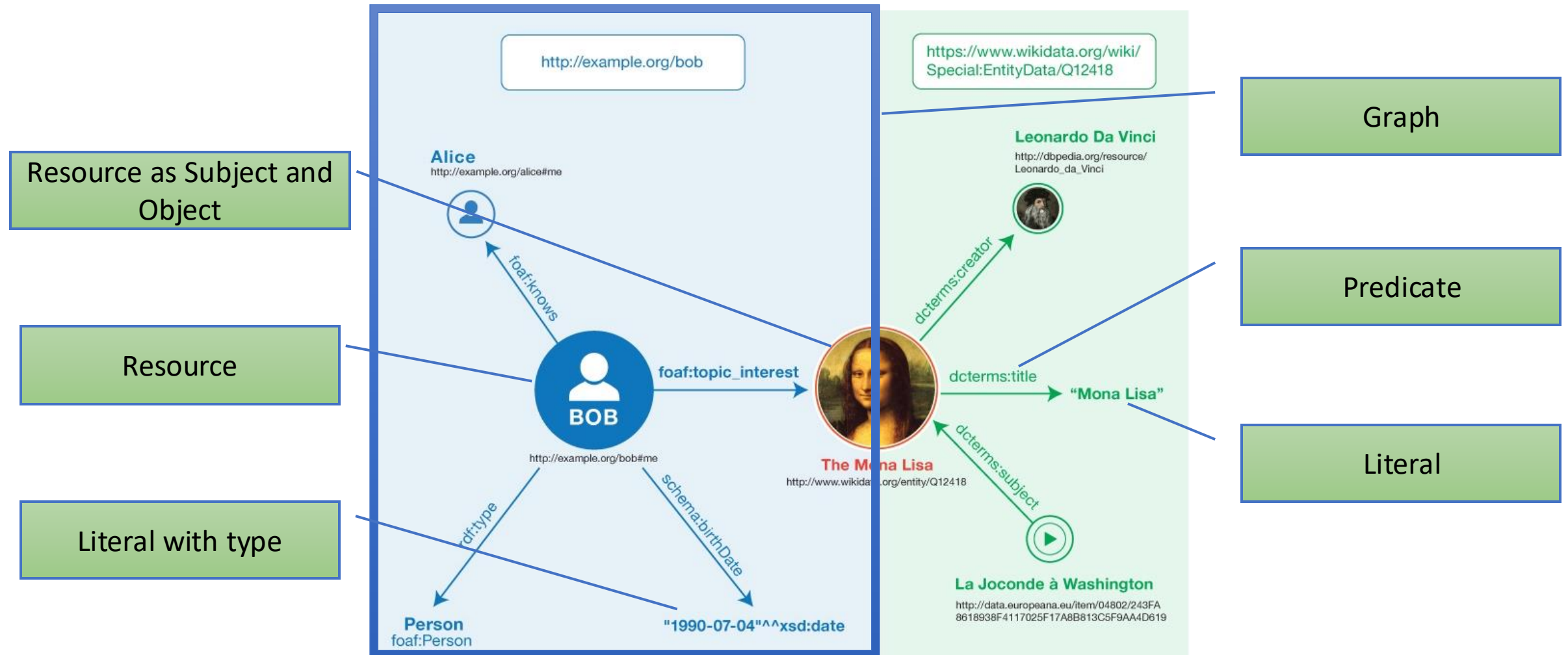
RDF components



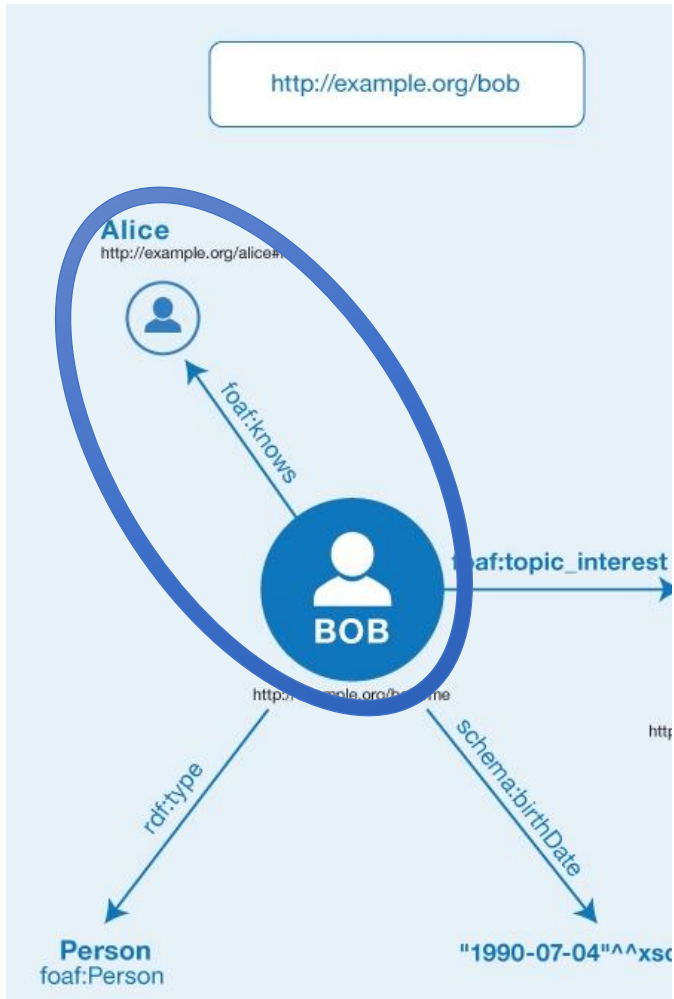
Compared to object model:

- A Resource is an Object
- A Property is an Attribute
- The attribute value can be another resource or a primitive type
 - value such as number, text, date

RDF example



RDF example (contd.)



<http://www.w3.org/TR/rdf11-primer/>

Statement (informal)
<Bob> <Knows> <Alice>

Namespace `ex`: defined to state
where properties have come from.
Eliminates need to write out full IRI

Triple (formal)
PREFIX `ex`: <<http://example.org/>>
PREFIX `foaf`:
 <<http://xmlns.com/foaf/0.1/>>
`ex:bob#me foaf:knows ex:alice#me .`

Namespaces

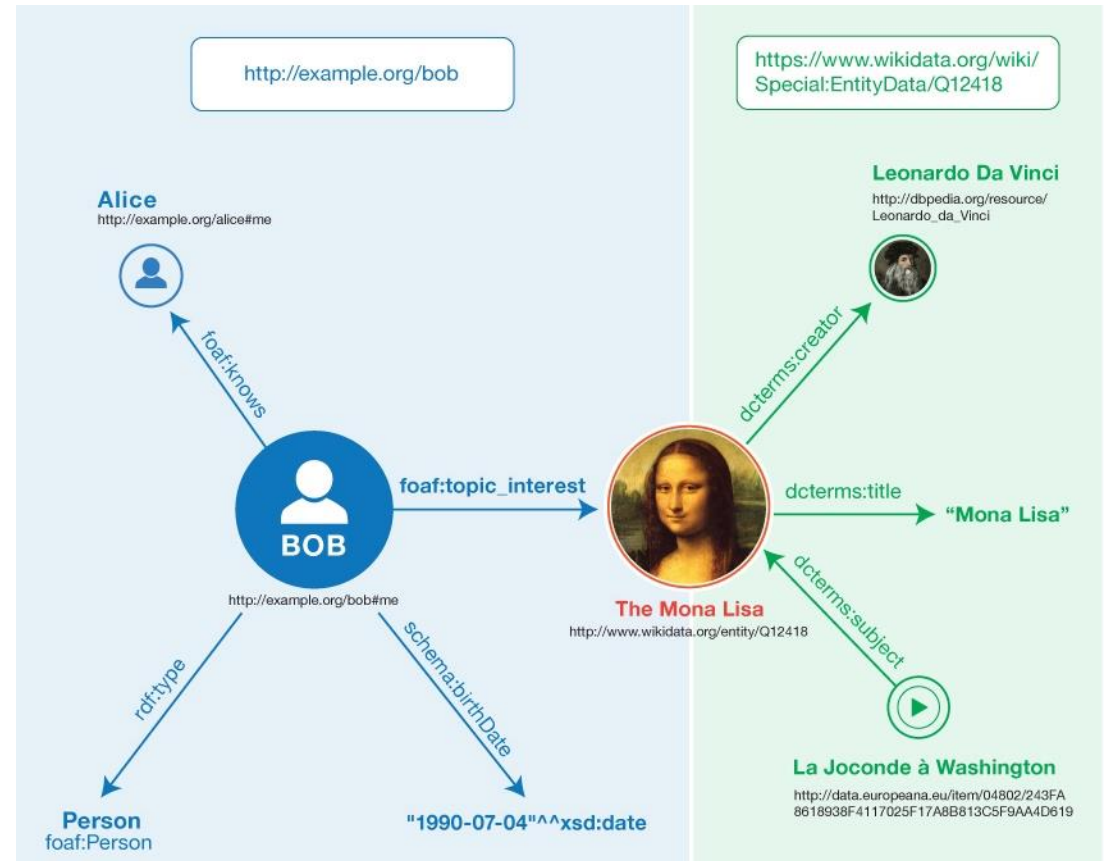
- Defines abbreviations for properties

```
PREFIX ns1: <http://purl.org/pav/>
```

- Common namespaces
 - rdf: RDF vocabulary – Knowledge representation
 - rdfs: RDF Schema – Knowledge representation
 - owl: OWL – Knowledge representation
 - foaf: friend of a friend – people and organizations
 - dct/dcterms: Dublin Core – Document metadata
- Namespace lookup: <http://prefix.cc/>

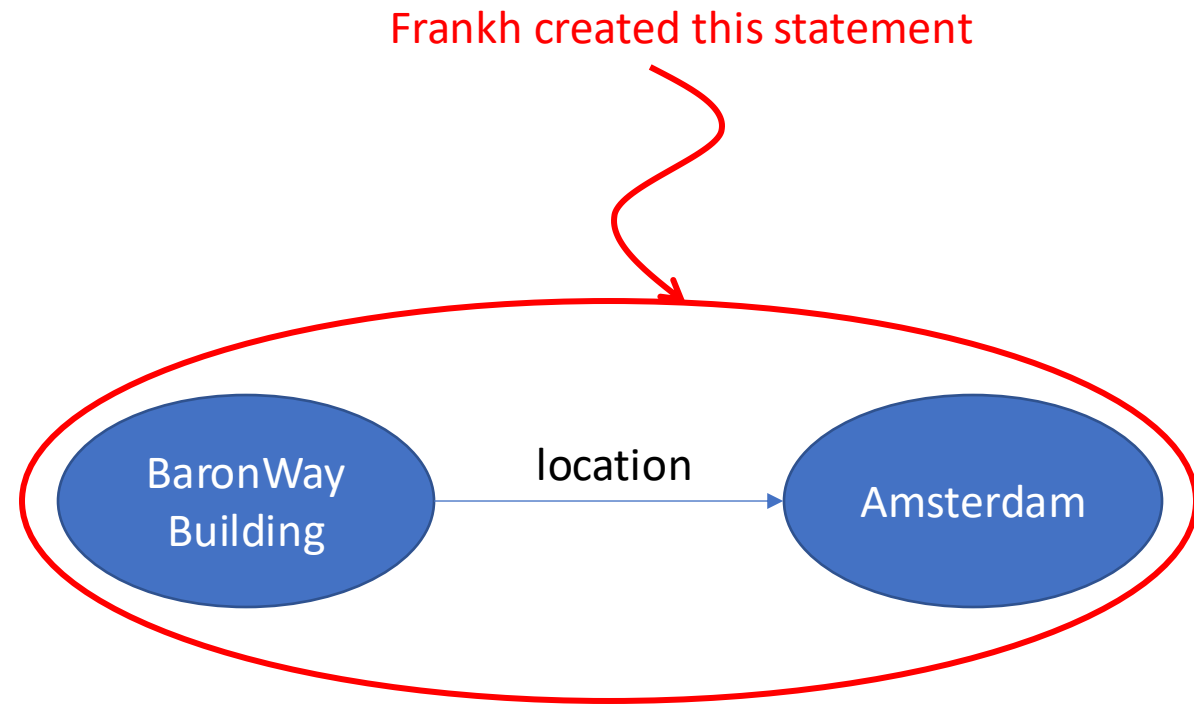
Integrating graphs

- Two graphs from different sources
- Merge identical nodes
 - Same IRI means identical nodes
 - In this case 'The Mona Lisa' is [wd:Q12418](http://www.wikidata.org/wiki/Special:EntityData/Q12418) in both sources
- If there exists no identical IRIs, then the two graphs will remain distinct in a single repository (file, store)



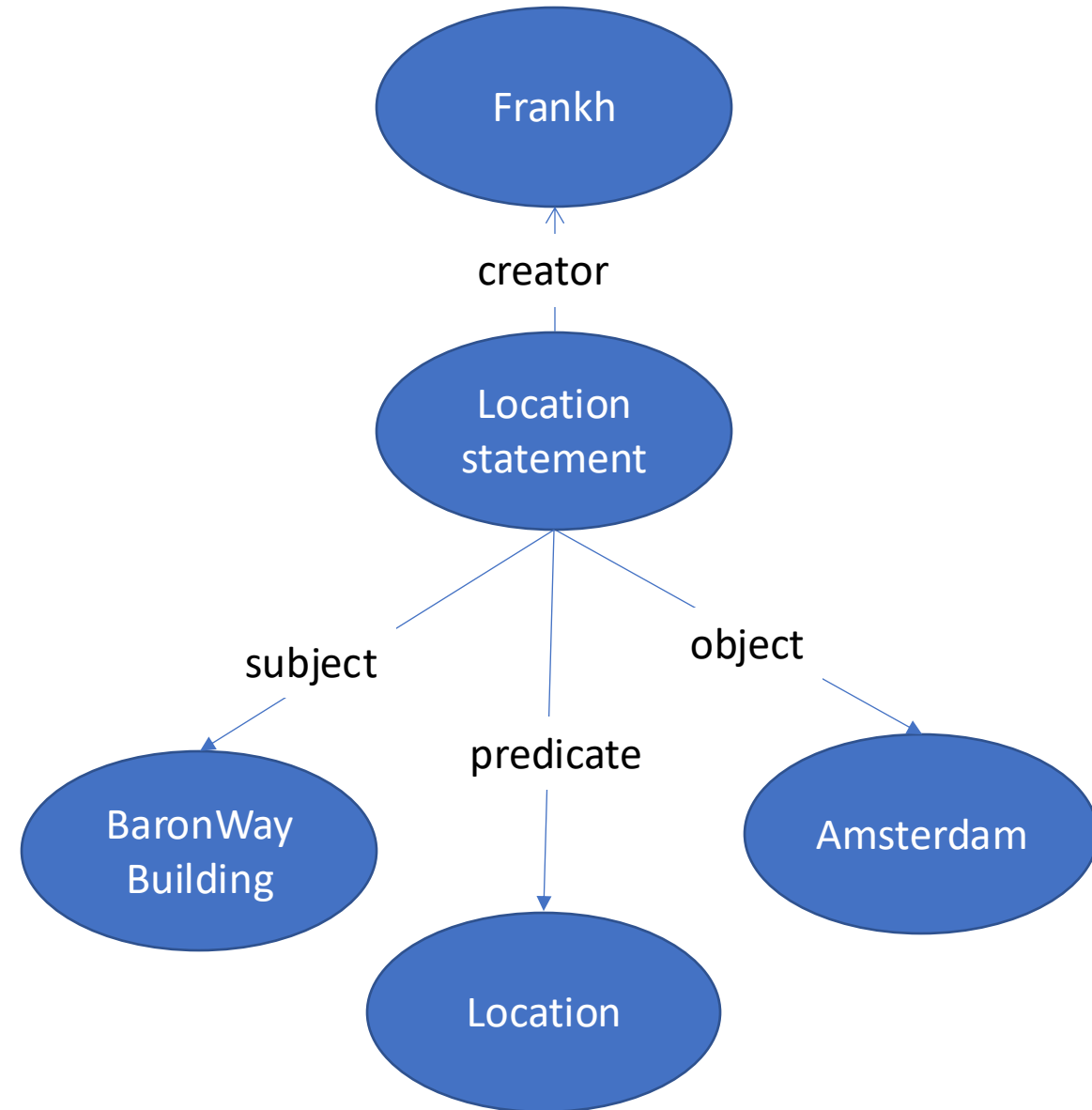
Complex RDF modelling

- How can we model that fact that
 - “Frankh created the statement about the location of the BaronWay Building”?
- Solution
 - Reification
 - Named graph



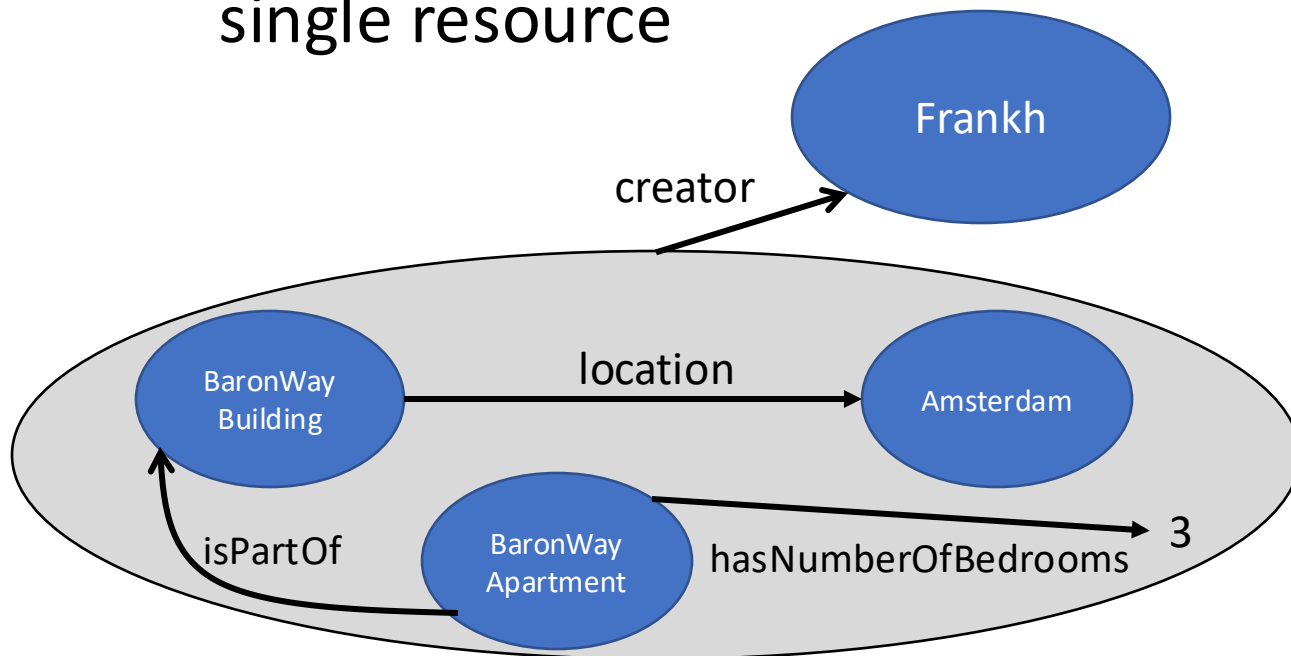
Reification

- Create auxiliary statement that relate to each component of the original statement
- Quadruples the size (4 triples for a statement)



Named graph

- We give an explicit identifier to the statement (or a set of statements)
- It would be considered as a single resource



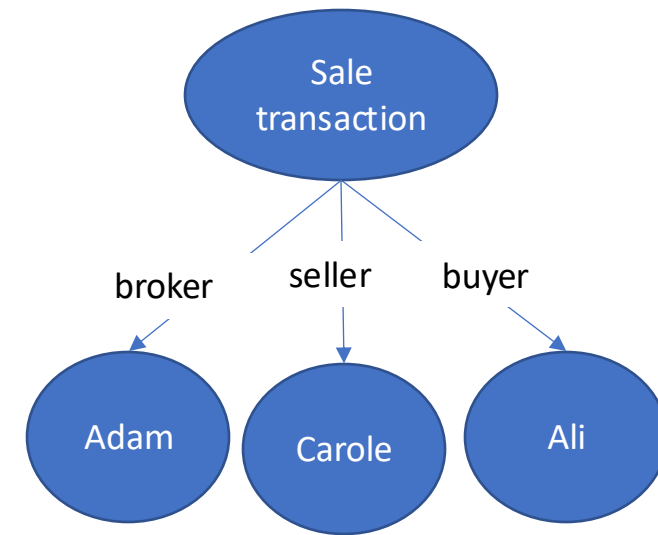
```
@prefix swp: <http://www.semanticwebprimer.org/ontology/apartments.ttl#>.
@prefix dbpedia: <http://dbpedia.org/resource/>.
@prefix dbpedia-owl: <http://dbpedia.org/ontology/>.
@prefix dc: <http://purl.org/dc/terms/>.

{
  <http://www.semanticwebprimer.org/ontology/apartments.ttl#>
  dc:creator <http://www.cs.vu.nl/ frankh>
}

<http://www.semanticwebprimer.org/ontology/apartments.ttl#>
{
  swp:BaronWayApartment swp:hasNumberOfBedrooms 3;
  swp:isPartOf swp:BaronWayBuilding.
  swp:BaronWayBuilding dbpedia-owl:location dbpedia:Amsterdam,
  dbpedia:Netherlands.
}
```


RDF and n-ary relationships

- RDF offers binary predicates only
- To model n-ary relationships, we adopt the following pattern
 - Introduce an auxiliary resource
 - Add pairwise relationships to the different resources



RDF serialisation formats

- **RDF** is an abstract **direct labelled graph** model
- To store it in a file, we use a concrete syntax
- There are many valid formats
 - RDF/XML
 - **N-triples**
 - **Turtle/Trig**
 - JSON-LD
 - RDF-a

RDF/XML example

- Original standardised serialisation
- Syntax processable by XML tools
- Content not interpretable by XML tools

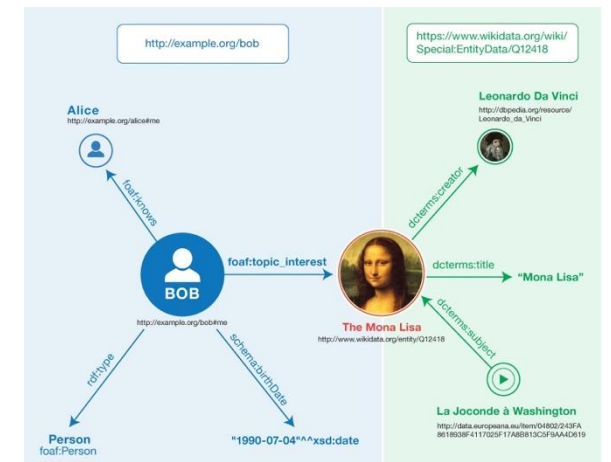
```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:dcterms="http://purl.org/dc/terms/"
  xmlns:foaf="http://xmlns.com/foaf/0.1/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:schema="http://schema.org/">

  <rdf:Description rdf:about="http://example.org/bob#me">
    <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
    <schema:birthDate rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1990-07-04</schema:birthDate>
    <foaf:knows rdf:resource="http://example.org/alice#me"/>
    <foaf:topic_interest rdf:resource="http://www.wikidata.org/entity/Q12418"/>
  </rdf:Description>

  <rdf:Description rdf:about="http://www.wikidata.org/entity/Q12418">
    <dcterms:title>Mona Lisa</dcterms:title>
    <dcterms:creator rdf:resource="http://dbpedia.org/resource/Leonardo_da_Vinci"/>
  </rdf:Description>

  <rdf:Description
    rdf:about="http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619">
    <dcterms:subject rdf:resource="http://www.wikidata.org/entity/Q12418"/>
  </rdf:Description>

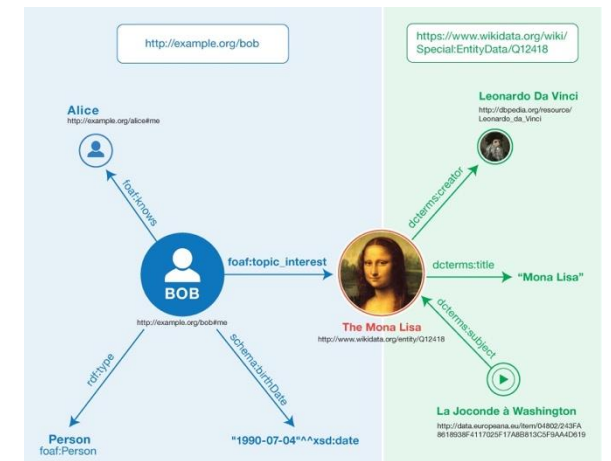
</rdf:RDF>
```



N-triples

- Every triple is complete (No namespaces)
- Cannot represent graphs
(Extended to N-Quads to capture graph)
- Good for mass insertion to triplestores
(Each triple is complete)

<http://example.org/bob#me> <http://www.w3.org/1999/02/22-rdf-syntax-ns#**type**> <http://xmlns.com/foaf/0.1/Person> .
<http://example.org/bob#me> <http://xmlns.com/foaf/0.1/**knows**> <http://example.org/alice#me> .
<http://example.org/bob#me> <http://schema.org/**birthDate**> "1990-07-04"^^<http://www.w3.org/2001/XMLSchema#date> .
<http://example.org/bob#me> <http://xmlns.com/foaf/0.1/**topic_interest**> <http://www.wikidata.org/entity/Q12418> .
<http://www.wikidata.org/entity/Q12418> <http://purl.org/dc/terms/**title**> "Mona Lisa" .
<http://www.wikidata.org/entity/Q12418> <http://purl.org/dc/terms/**creator**> <http://dbpedia.org/resource/Leonardo_da_Vinci> .
<http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619> <http://purl.org/dc/terms/**subject**>
<http://www.wikidata.org/entity/Q12418> .



A simple line-based, plain-text way for serializing RDF graphs
Each line represents a triple. Full IRIs are enclosed in angle brackets “<>”
The period “.” at the end of the line signals the end of the triple.

N-Quads (generalise N-triples)

- Every quad is complete (No namespaces)
- Triple + graph IRI (Graph IRI can be omitted)
- Good for mass insertion to triplestores (Each quad is complete, Graphs go into separate repository)

```

<http://example.org/bob#me> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person>
<http://example.org/bob> .

<http://example.org/bob#me> <http://xmlns.com/foaf/0.1/knows> <http://example.org/alice#me> <http://example.org/bob> .

<http://example.org/bob#me> <http://schema.org/birthDate> "1990-07-04"^^<http://www.w3.org/2001/XMLSchema#date>
<http://example.org/bob> .

<http://example.org/bob#me> <http://xmlns.com/foaf/0.1/topic_interest> <http://www.wikidata.org/entity/Q12418>
<http://example.org/bob> .

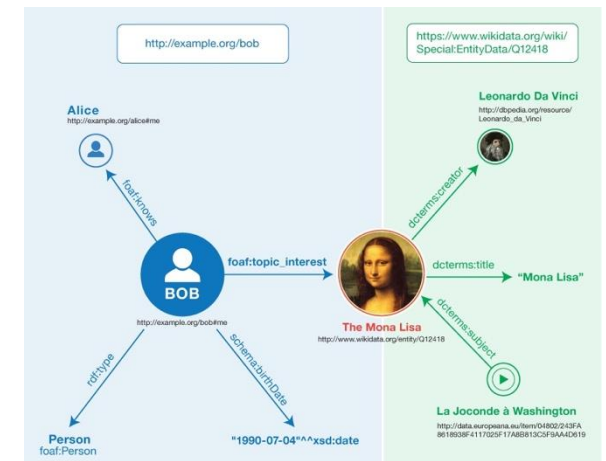
<http://www.wikidata.org/entity/Q12418> <http://purl.org/dc/terms/title> "Mona Lisa"
<https://www.wikidata.org/wiki/Special:EntityData/Q12418> .

<http://www.wikidata.org/entity/Q12418> <http://purl.org/dc/terms/creator> <http://dbpedia.org/resource/Leonardo_da_Vinci>
<https://www.wikidata.org/wiki/Special:EntityData/Q12418> .

<http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619> <http://purl.org/dc/terms/subject>
<http://www.wikidata.org/entity/Q12418> <https://www.wikidata.org/wiki/Special:EntityData/Q12418> .

<http://example.org/bob> <http://purl.org/dc/terms/publisher> <http://example.org> .

<http://example.org/bob> <http://purl.org/dc/terms/rights> <http://creativecommons.org/licenses/by/3.0/> .
    
```



N-Quads allows one to add a fourth element to a line, capturing the graph IRI of the triple described on that line

Turtle

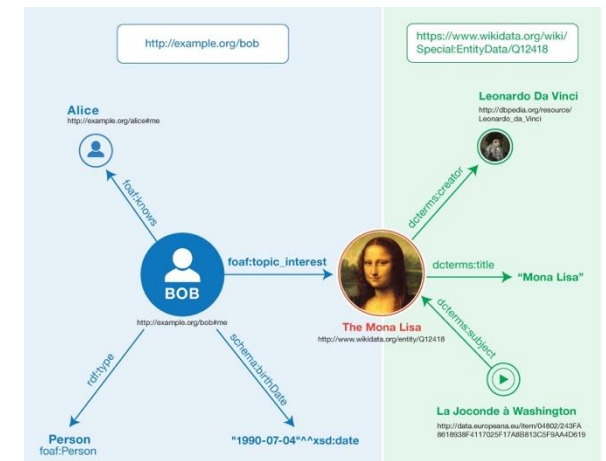
- Human readable serialisation
- Very similar to SPARQL
- Uses namespaces
- Cannot represent graphs

```
BASE    <http://example.org/>
PREFIX  foaf: <http://xmlns.com/foaf/0.1/>
PREFIX  xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX  schema: <http://schema.org/>
PREFIX  dcterms: <http://purl.org/dc/terms/>
PREFIX  wd: <http://www.wikidata.org/entity/>
```

```
<bob#me> a foaf:Person ;
         foaf:knows <alice#me> ;
         schema:birthDate "1990-07-04"^^xsd:date ;
         foaf:topic_interest wd:Q12418 .
```

```
wd:Q12418 dcterms:title "Mona Lisa" ;
         dcterms:creator
           <http://dbpedia.org/resource/Leonardo_da_Vinci> .
```

```
<http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619>
         dcterms:subject wd:Q12418 .
```



Turtle introduces a number of syntactic shortcuts, such as support for **namespace prefixes**, lists and shorthands for datatyped literals.

Trig

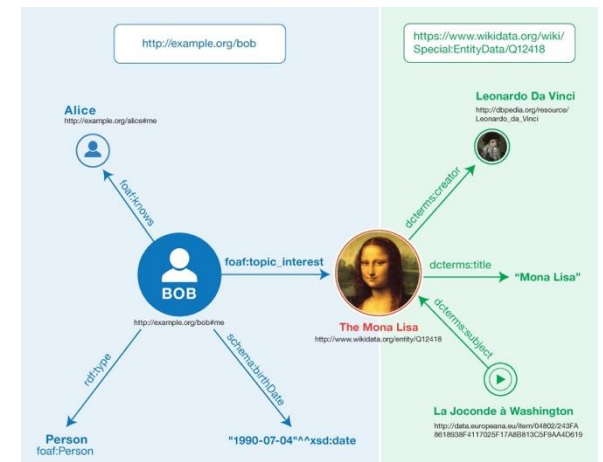
- Extension of turtle to represent graph context

```
BASE    <http://example.org/>
PREFIX  foaf: <http://xmlns.com/foaf/0.1/>
PREFIX  xsd:  <http://www.w3.org/2001/XMLSchema#>
PREFIX  schema: <http://schema.org/>
PREFIX  dcterms: <http://purl.org/dc/terms/>
PREFIX  wd: <http://www.wikidata.org/entity/>

GRAPH <http://example.org/bob> {
    <bob#me> a foaf:Person ;
    foaf:knows <alice#me> ;
    schema:birthDate "1990-07-04"^^xsd:date ;
    foaf:topic_interest wd:Q12418 .
}

GRAPH <https://www.wikidata.org/wiki/Special:EntityData/Q12418> {
    wd:Q12418 dcterms:title "Mona Lisa" ;
    dcterms:creator <http://dbpedia.org/resource/Leonardo_da_Vinci> .

    <http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619>
        dcterms:subject wd:Q12418 .
}
```



JSON-LD

- JSON extension for RDF representation
- Favoured by Google for web page markup
 - Include as script at start or end of web page

```
<script type="application/ld+json">
{
  "@context": "http://schema.org",
  "@type": "Movie",
  "name": "Jason Bourne",
  "url": "http://www.jasonbourne.com/",
  "actor": {
    "@type": "Person",
    "name": "Matt Damon",
    "sameAs": "http://www.imdb.com/name/nm0000354/"
  },
  "director": {
    "@type": "Person",
    "name": "Paul Greengrass",
    "sameAs": "http://www.imdb.com/name/nm0339030/"
  },
  "image": {
    "@type": "ImageObject",
    "url": "http://themoderatevoice.com/wp-content/uploads/2016/08/maxresdefault.jpg"
  }
}
</script>
```



Tips: Indent nested elements.



<https://developers.google.com/search/docs/data-types/tv-movies>

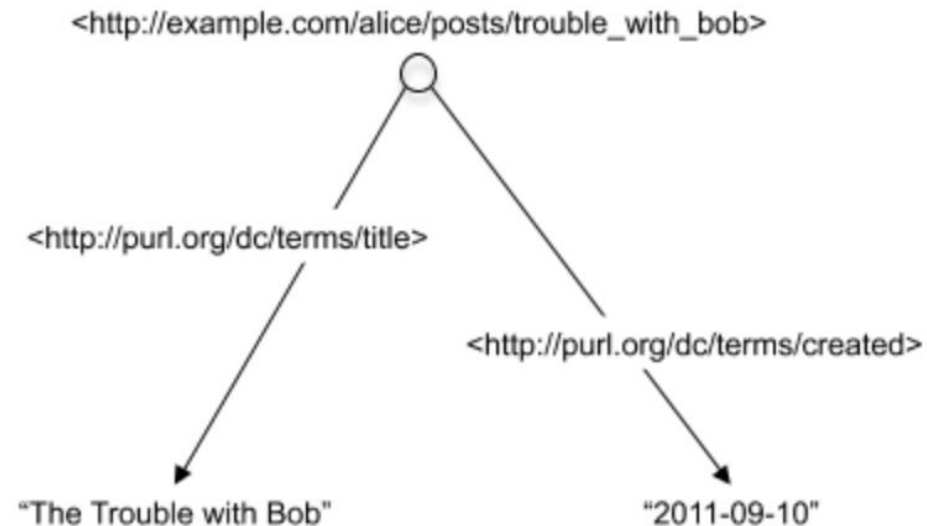
RDF-a

- The essence of RDFa is to provide a **set of attributes** that can be used to carry metadata in an XML language (hence the 'a' in RDFa).
- **about** - a URI specifying the resource
- **rel** and **rev** specifying a relationship and reverse-relationship with another resource,
- **src**, **href** and **resource** - specifying the partner resource
- **property** - specifying a property for the content of an element or the partner resource
- **content** - optional attribute that overrides the content of the element when using the property attribute
- **datatype** - optional attribute that specifies the datatype of text specified for use with the property attribute
- **typeof** - optional attribute that specifies the RDF type(s) of the subject or the partner resource (the resource that the metadata is about).

RDF-a

- Embeds RDF markup within HTML representation
- More concise web pages than JSON-LD
- More difficult to maintain than JSON-LD

```
<html>
<head>
  ...
</head>
<body>
  ...
  <h2 property="http://purl.org/dc/terms/title">The Trouble with Bob</h2>
  <p>Date: <span property="http://purl.org/dc/terms/created">2011-09-10</span></p>
  ...
</body>
```

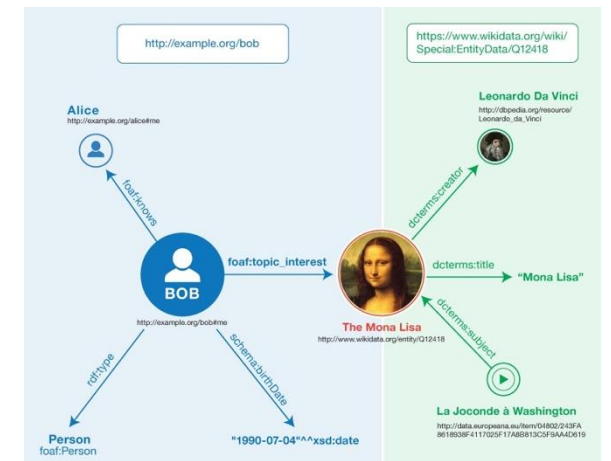


```
<p>All content on this site is licensed under
  <a property="http://creativecommons.org/ns#license" href="http://creativecommons.org/licenses/by/3.0/">
    a Creative Commons License</a>. ©2011 Alice Birpemschwang.</p>
```

RDF-a

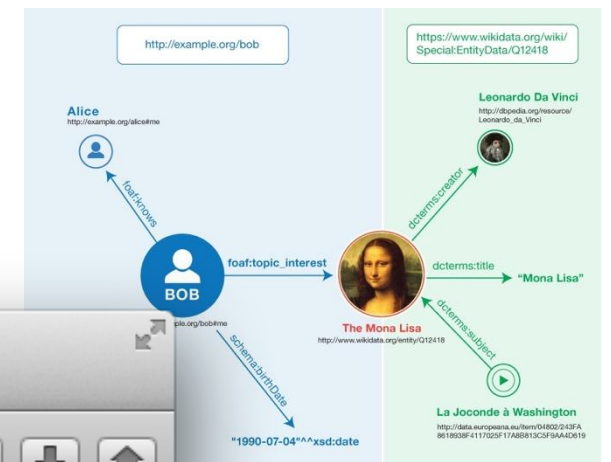
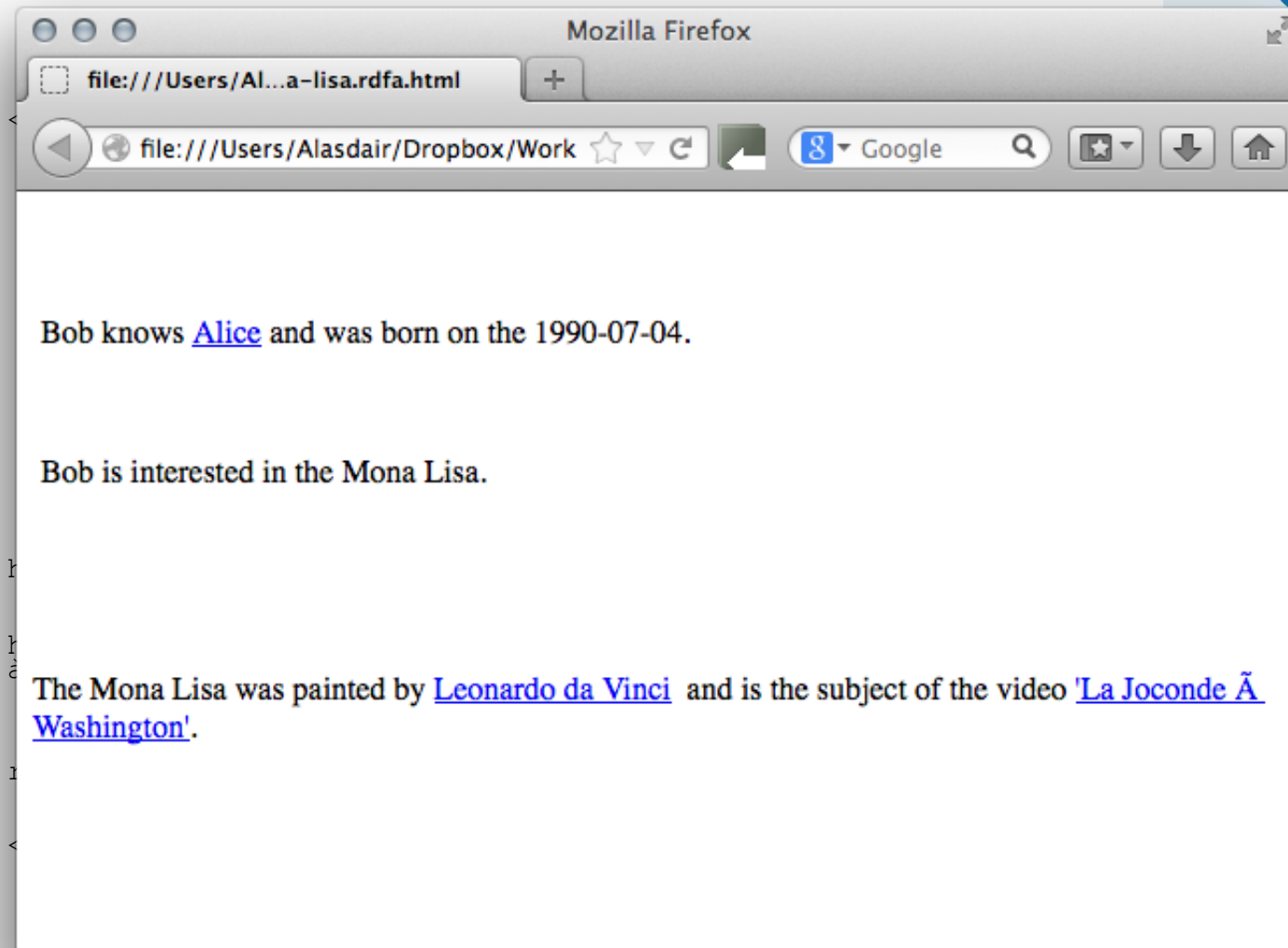
- Embeds RDF markup within HTML representation
- More concise web pages than JSON-LD
- More difficult to maintain than JSON-LD

```
<body prefix="foaf: http://xmlns.com/foaf/0.1/
          schema: http://schema.org/
          dcterms: http://purl.org/dc/terms/">
  <div resource="http://example.org/bob#me" typeof="foaf:Person">
    <p>
      Bob knows <a property="foaf:knows"
        href="http://example.org/alice#me">Alice</a>
      and was born on the <time property="schema:birthDate">1990-07-04</time>.
    </p>
    <p>
      Bob is interested in <span property="foaf:topic_interest"
        resource="http://www.wikidata.org/entity/Q12418">the Mona Lisa</span>.
    </p>
  </div>
  <div resource="http://www.wikidata.org/entity/Q12418">
    <p>
      The <span property="dcterms:title">Mona Lisa</span> was painted by
      <a property="dcterms:creator"
        href="http://dbpedia.org/resource/Leonardo_da_Vinci">Leonardo da Vinci</a>
      and is the subject of the video
      <a
        href="http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619">'La Joconde
        à Washington'</a>.
    </p>
  </div>
  <div
    resource="http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619">
    <link property="dcterms:subject" href="http://www.wikidata.org/entity/Q12418"/>
  </div>
</body>
```



RDF-a: screenshot

- Embeds RDF markup within HTML representation
- More concise web pages than JSON-LD
- More difficult to maintain than JSON-LD



7-04</time>.

sa.

ted by

a>

5F9AA4D619">'La Joconde

813C5F9AA4D619">
ntity/Q12418"/>

Exercise 1: draw the graph

```
BASE <http://www.recshop.fake/cd#>
PREFIX cd:
<http://www.recshop.fake/cd#>
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX mo:
<http://purl.org/ontology/mo/>
PREFIX rdf:
<http://www.w3.org/1999/02/22-rdf-
syntax-ns#>
PREFIX xsd:
<http://www.w3.org/2001/XMLSchema
#>
```

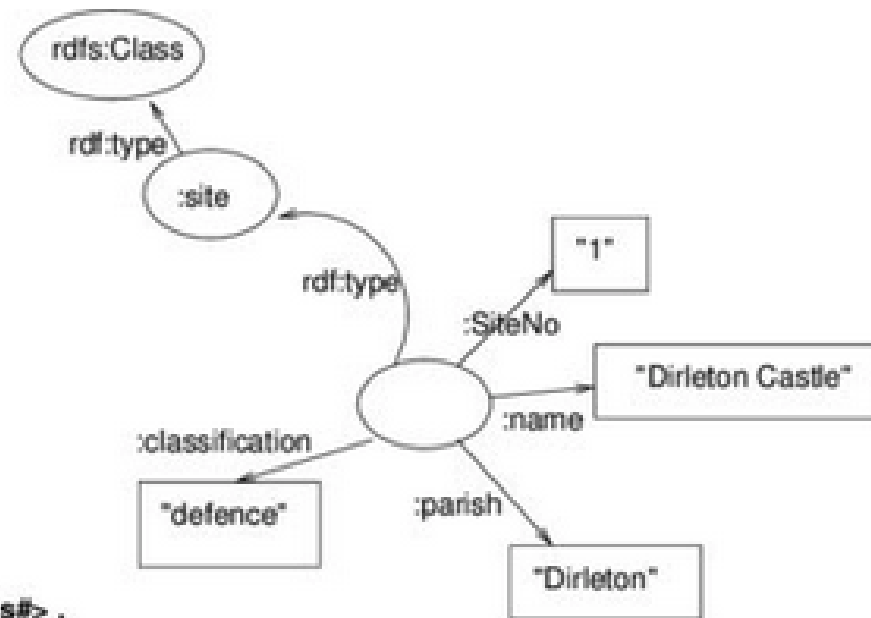
```
cd:Empire_Burlesque a mo:CD;
  dct:title "Empire Burlesque";
  cd:artist "BobDylan";
  cd:company "Columbia";
  cd:year "1985"^^xsd:year.
cd:Hide_your_heart a mo:CD;
  dct:title "Hide your heart";
  cd:artist "BonnieTyler";
  cd:company "Columbia";
  cd:year "1988"^^xsd:year.
```

Exercise2: write RDF for the following graph

SITE

siteNo	name	parish	classification
1	Dirleton Castle	Dirleton	defence
2	Dirleton Cottage	Dirleton	residential
3	Drem Airfield	Dirleton	military
4	Jamie's Neuk	Dirleton	military

@prefix : <http://www.ltg.ed.ac.uk/tether/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .



Summary

- Web semantic technology stack made of layers
 - RDFS, OWL to write ontologies to model the domain (classes, relationships)
- Data written in RDF, where classes and relationships are from ontologies (ours or existing open ones)
- Inferencing is used to generate new data (applying the ontology rules)
- RDF uses subject, predicate, object, with many serialisations
- RDF is machine interpretable
- Globally defined identifiers and properties
- If we need our own vocabulary, we can create it using OWL (**next week**)

The most important take away

The *meaning of the data* is encoded alongside the data

Machine readable and globally defined

i.e., how to use/interpret the data is now located alongside the data in a standardised data model that anyone can read and process.

This makes integration/interoperability much easier

RDF visualisers (many online)

- RDF graph visualiser
- <https://www.lda.fi/service/rdf-grapher> –
- <https://issemantic.net/rdf-visualizer>
- <https://visgraph3.github.io/>
- <https://wiki.gephi.org/index.php/SemanticWebImport>