RDF Data Model

Angelica Lo Duca
IIT-CNR
angelica.loduca@iit.cnr.it

Linked Open Data:
a paradigm for the
Semantic Web

Motivation

How to encode the following piece of knowledge?

The theory of relativity was discovered by Albert Einstein.

```
<theory>
    <name>
       Theory of Relativity
    </name>
    <discoverer>
        AlbertEinstein
    </discoverer>
</theory>
```

```
<person>
    <name>
    Albert Einstein
    </name>
    <discovered>
    Theory of Relativity
    </discovered>
</person>
```

```
<person name="Albert
Einstein">
   <discovered>
     Theory of Relativity
   </discovered>
</person>
```

In XML there is not a single way to represent knowledge

Resource Description Framework (RDF)

- RDF gives a standard model to represent knowledge
  - RDF is W3C Recommendation
- RDF is a data model
  - Originally used for metadata for web resources, then generalized
  - Encodes structured information
  - Universal, machine readable exchange format
- Data structured in graphs
  - Nodes, Arcs
- RDF Exploits XML

The Resource Description Framework (RDF) is a general framework to describe any Internet **resource** such as a Web site and its content.

RDF is a set of rules (a sort of language) for creating descriptions of **resources**.

A **resource** is any object that is uniquely identifiable by an Uniform Resource Identifier (URI).

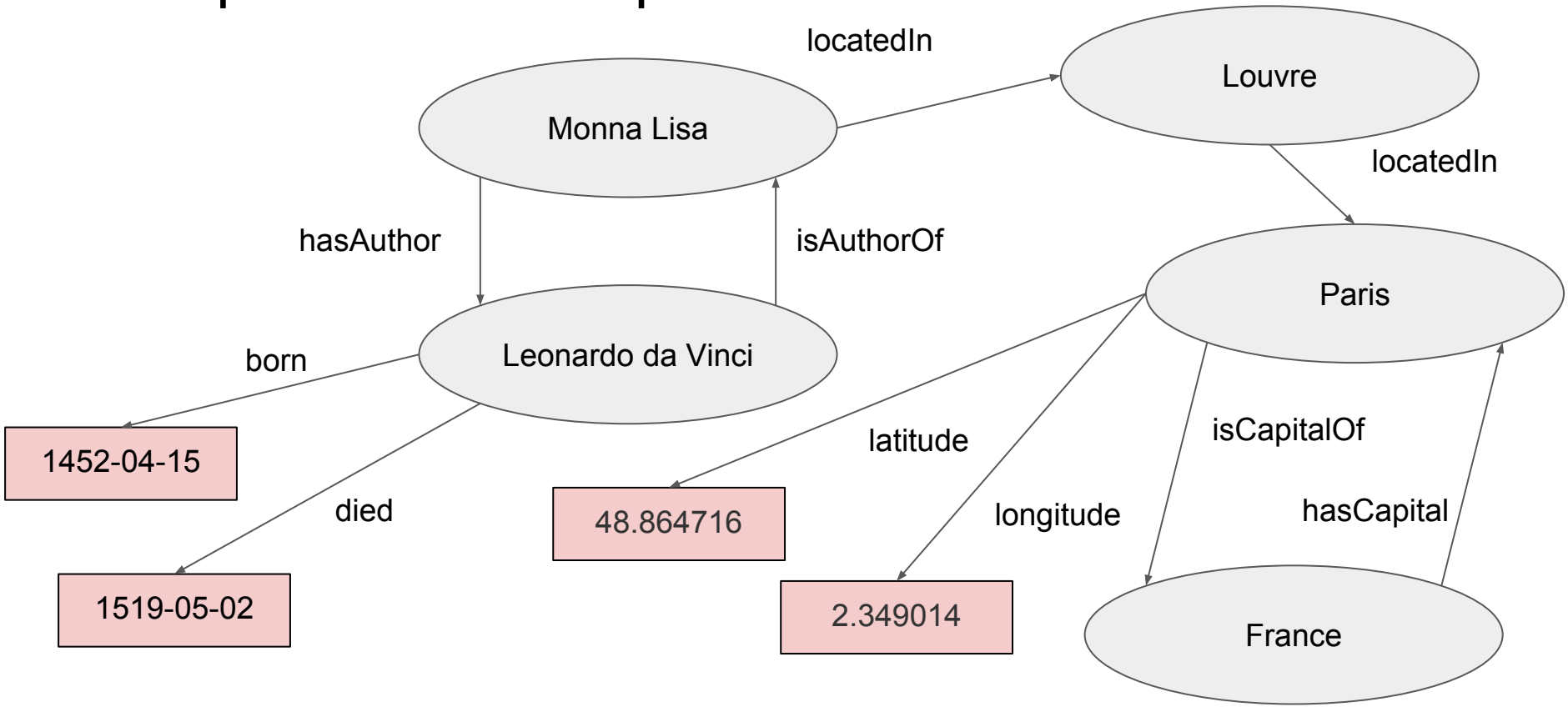
Resources

- Resources have properties (attributes or characteristics)
- RDF provides a mechanism for associating properties with resources
- Each property has a value
- RDF represents the relationship among resources, property and values in a direct labeled graph

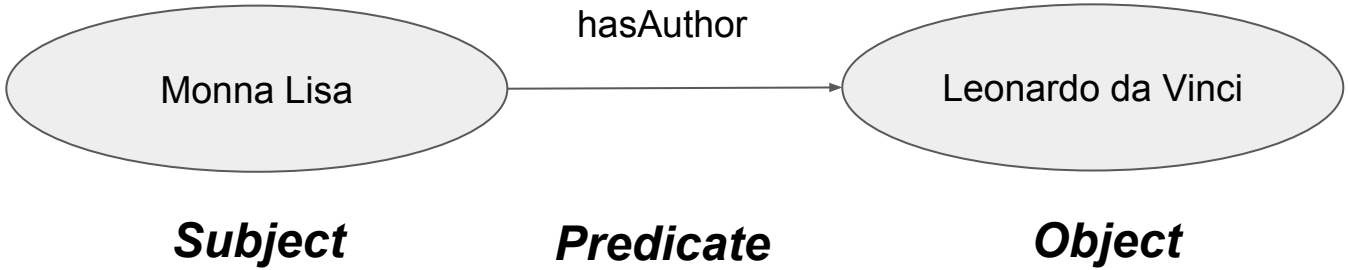
RDF Graph

- **Nodes**
  - URIs - used to reference resources unambiguously
  - Literals - describe data values
  - Blank Nodes - anonymous resources
- **Labeled (direct) arcs**
  - Each arc links a pair of nodes

Example of RDF Graph



RDF Triples



- Allowed assignments:
  - Subject: URI or blank node
  - Predicate: URI
  - Object: URI, blank node or literal
- Node and arcs labels *should be* unambiguous

Uniform Resource Identifier (URI)

- Used to create globally unique names for resources
- Every object with a clear identity can be a resource
  - Books, places, organizations ...

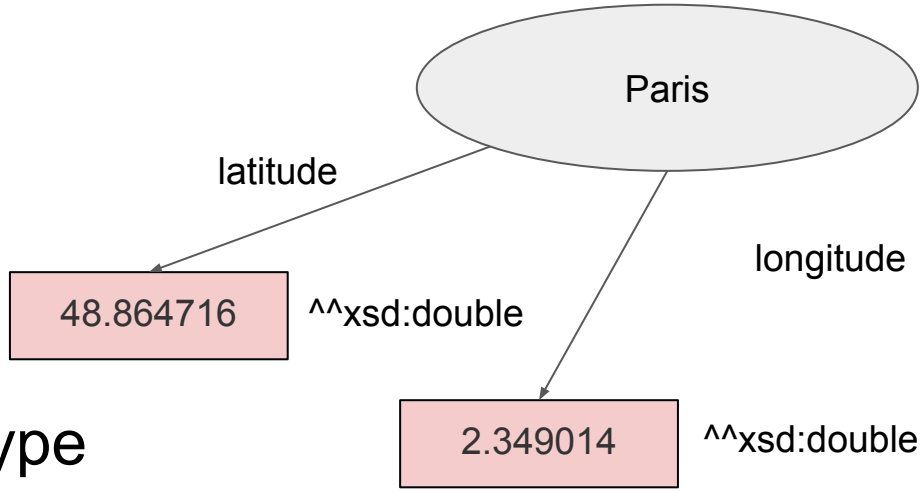
- In books domain the ISBN serves the same purpose

Prefixes

- As abbreviations of URIs, prefixes can be used
  - use ex: instead of http://www.example.com/

Literals

- Used to model data values
- Representation as strings
- Interpretation through datatype
- Literals without datatype are treated as strings



Literals may **never be the origin of a node** of an RDF graph

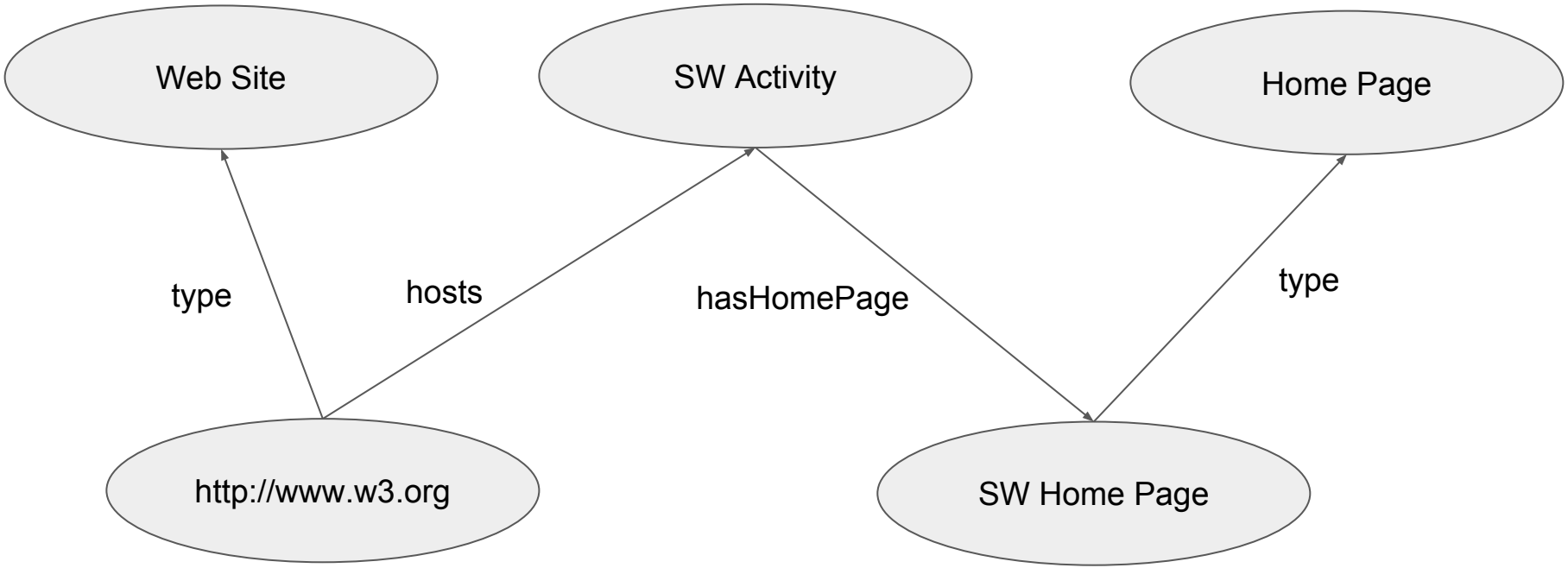
Arcs may **never be labeled** with literals

Assignment 1

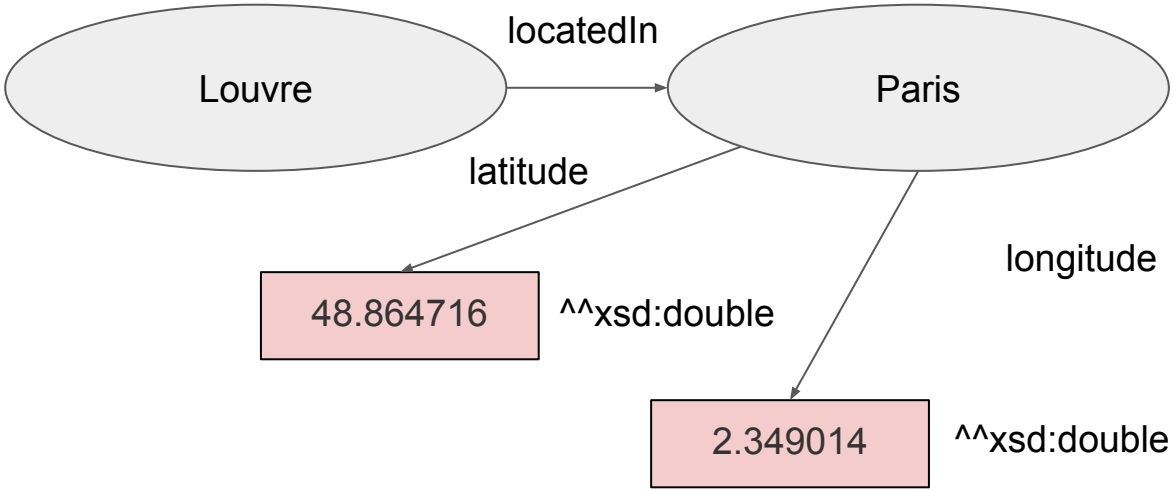
Map the following statement to an RDF Graph:

The Web site http://www.w3.org contains the home page of W3C activity about Semantic Web.

A possible solution



RDF/XML Syntax




```
<rdf:Description>
    <ex:locatedIn>
        <rdf:Description>
            <ex:latitude rdf:datatype=xsd:double>48.864716</ex:latitude>
            <ex:longitude rdf:datatype=xsd:double>2.349014>/ex:longitude>
        </rdf:Description>
    </ex:locatedIn>
</rdf:Description>
```

- `<rdf:Description>` to represent a node
- `<predicate_name>` to represent a predicate

RDF/XML Syntax (2)

```
<rdf:Description>
    <ex:locatedIn>
        <rdf:Description ex:latitude="48.864716" ex:longitude="2.349014">
        </rdf:Description>
    </ex:locatedIn>
</rdf:Description>
```

- As an alternative, a literal node can be set as a XML attribute element on the containing node
- In the example latitude and longitude loose datatype

RDF/XML Syntax (3)

```
<rdf:Description rdf:about="http://ex.com/Louvre">
    <ex:locatedIn>
        <rdf:Description
            rdf:about="http://ex.com/Paris"
            ex:latitude="48.864716"
            ex:longitude="2.349014">
        </rdf:Description>
    </ex:locatedIn>
</rdf:Description>
```

The URI of a node can be specified through the attribute **rdf:about**

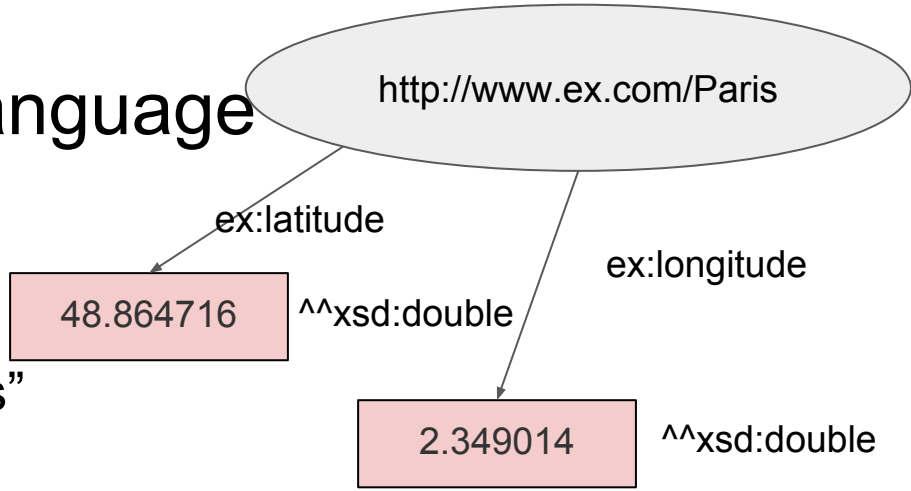
RDF/XML Syntax (4)

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns# "
         xmlns:ex=”http:/ex.com#”>

    <rdf:Description rdf:about=”http://ex.com/Louvre”>

        …

    </rdf:Description>

</rdf:RDF>
```

- Complete Document with Document Element and XML declaration
- Declare all the employed prefixes

Turtle - Terse RDF Triple Language

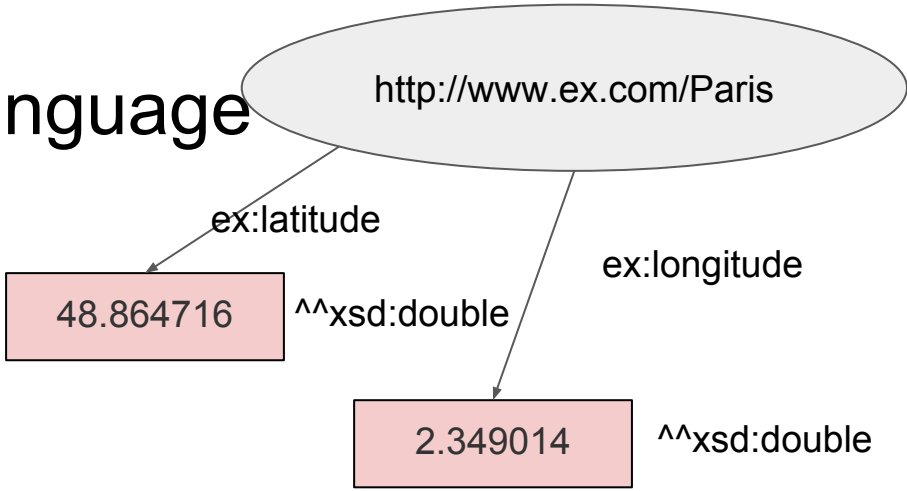
- simple syntax for RDF
  - URIs are in <angle brackets>
  - Literals are “enclosed in quotes”
  - triples end with a full-stop .
  - whitespace is ignored




```
@prefix ex: <http://www.ex.com> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
.
<http://www.ex.com/Paris> ex:latitude “48.864716”^^xsd:double .
<http://www.ex.com/Paris> ex:longitude “2.349014”^^xsd:double .
```

Turtle - Terse RDF Triple Language

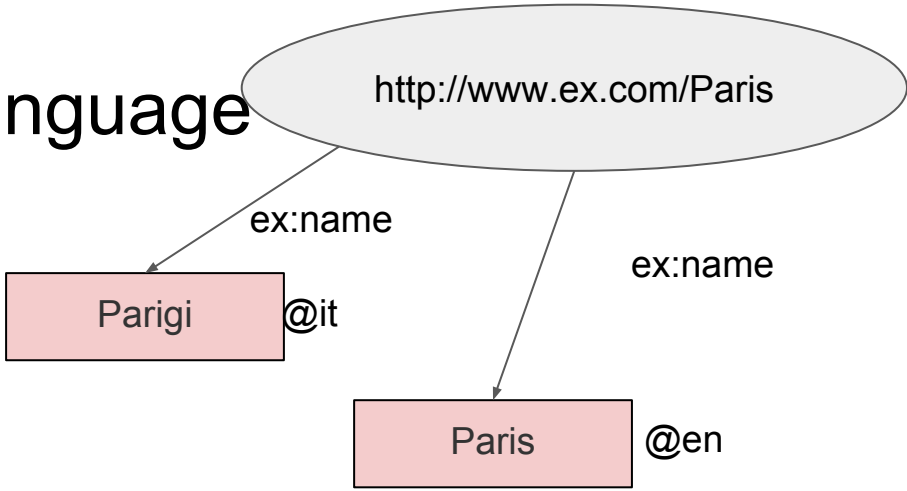
- URIs can be made more readable through prefixes
- use semicolon to group triples having the same subject




```
@prefix ex: <http://www.ex.com> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
.
ex:Paris    ex:latitude “48.864716”^^xsd:double ;
            ex:longitude “2.349014”^^xsd:double .
```

Turtle - Terse RDF Triple Language

- use colon to group triples having the same subject and predicate
- us @ to specify the language of a literal




```
@prefix ex: <http://www.ex.com> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
.
ex:Paris    ex:name “Parigi”@it, “Paris”@en .
```

Assignment 2

Represent the following statement in Turtle:

The Web site http://www.w3.org contains the home page of W3C activity about Semantic Web.

A possible solution

```
@prefix rdf:
<http://www.w3.org/1999/02/22-rdf-syntax-ns#>.
@prefix : <http://example.com#> .
<http://www.w3.org/> rdf:type :website .
<http://www.w3.org/> :hosts :sw-activity .
:sw-activity :has-homepage :sw-activity-homepage
.
:sw-activity-homepage rdf:type :homepage .
```

N-ary relations

"For the preparation of an apple pie you need 500 gr of flour, 3 eggs, …"

- **First attempt** to model this recipe
  - use strings to model ingredients and their amounts

```
@prefix ex: <http://www.ex.com> .

ex:ApplePie  ex:hasIngredient "500 gr of flour", "3 eggs" .
```

Search for recipes which contain flour is difficult

N-ary relations

"For the preparation of an apple pie you need 500 gr of flour, 3 eggs, …"

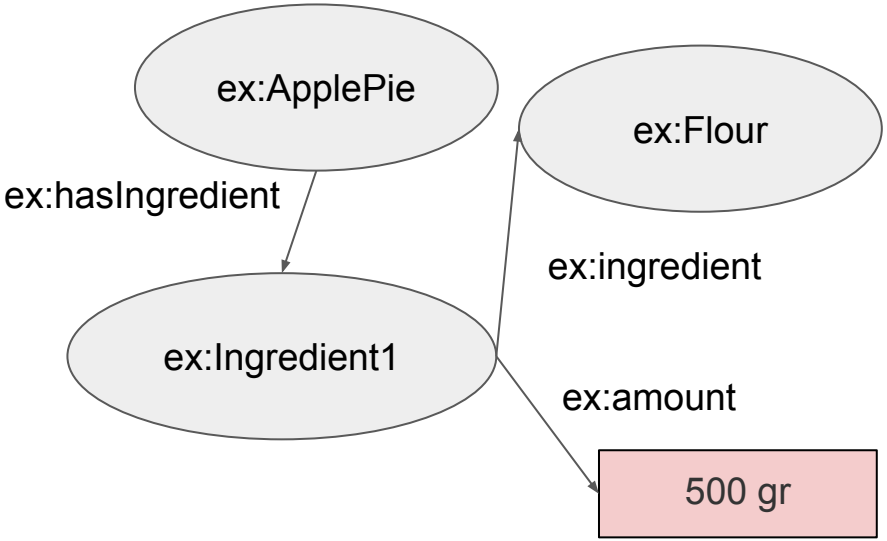
- **Second attempt** to model this recipe
  - ingredients become URIs

```
@prefix ex: <http://www.ex.com> .

ex:ApplePie   ex:ingredient ex:Flour; ex:amount "500 gr";
              ex:ingredient ex:Egg; ex:amount "3" .
```

Unambiguous association between ingredient and amount is not possible

N-ary relations

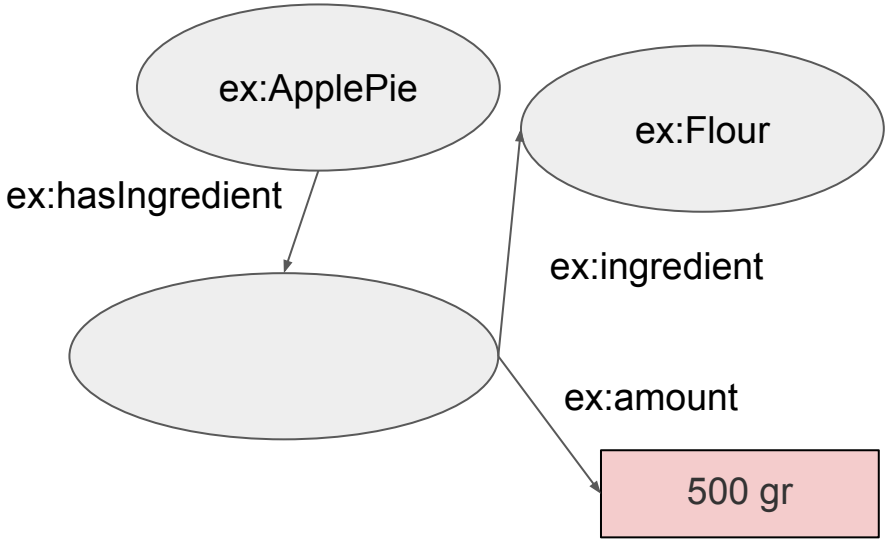
- Solution
  - introduce an helper node for each ingredient




```
@prefix ex: <http://www.ex.com> .

ex:ApplePie ex:hasIngredient ex:Ingredient1,
                             ex:Ingredient2 .

ex:Ingredient1     ex:ingredient ex:Flour ;
                   ex:amount “500 gr” .

ex:Ingredient2     ex:ingredient ex:Egg ;
                   ex:amount “3” .
```

Blank Nodes

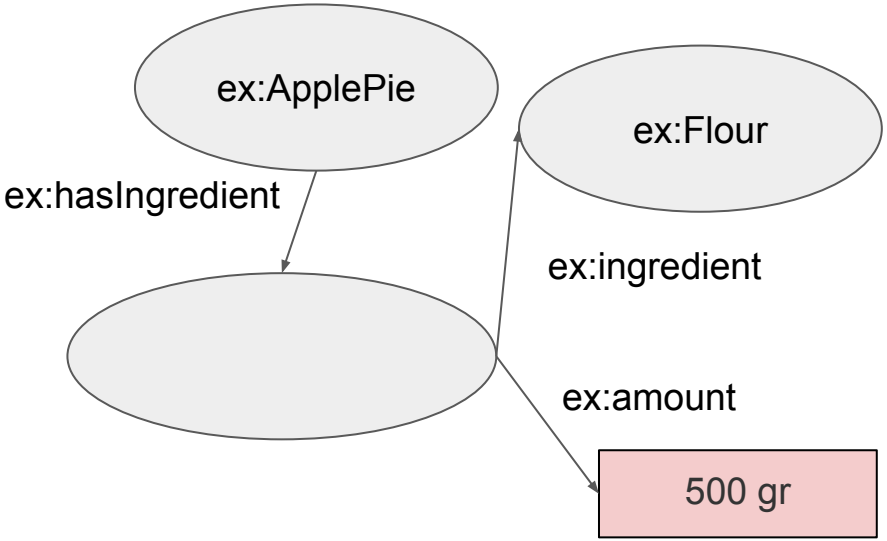
- Blank nodes can be seen as nodes that do not need to be named




```
@prefix ex: <http://www.ex.com> .

ex:ApplePie ex:hasIngredient _:id1,
                             _:id2 .

_:id1     ex:ingredient ex:Flour ;
                        ex:amount “500 gr” .

_:id2     ex:ingredient ex:Egg ;
                        ex:amount “3” .
```

Blank Nodes

- Use the short syntax




```
@prefix ex: <http://www.ex.com> .

ex:ApplePie ex:hasIngredient
[ex:ingredient ex:Flour ; ex:amount "500
gr"],
[ex:ingredient ex:Egg ; ex:amount "3" ] .
```

Assignment 3

Given the following set of triples, build the associated RDF graph.

Subject	Predicate	Object
:P1	rdf:type	:Person
:P1	:name	"Pete"
:P1	:age	"17"
:P1	:email	"pete@abc.com"
:P2	rdf:type	:Person
:P1	:knows	:P2
:P2	:name	"John"
:P2	:email	"john@abc.com"
:P2	:knows	:P1
:P2	:knows	:P3
:P1	:knows	:P3
:P3	rdf:type	:Person
:P3	:name	"Sue"
:P3	:age	“21"

References

- RDF 1.1 W3C Recommendation - http://www.w3.org/TR/rdf-syntax-grammar/
- Turtle - http://www.w3.org/TR/turtle/