

RDF basics

Descrivere i dati
con RDF

Semantic Web: foto di famiglia



- RDF: permette di descrivere risorse.
 - Una risorsa può essere qualsiasi cosa: un documento, una persona, un oggetto fisico, un concetto astratto
- RDFS: permette di descrivere relazioni tra risorse
 - È un linguaggio “property-centric”
 - Propriamente, estensione di RDF
- OWL: permette di descrivere ontologie computazionali
 - Si colloca a un livello superiore di RDF
 - Permette di descrivere relazioni ricche e complesse tra entità (“complex knowledge about things, groups of things, and relations between things”)

Semantic Web: gli standard

- <https://www.w3.org/standards/semanticweb> (parte del più ampio archivio di recommendations: <https://www.w3.org/standards>)
- Oggi: “The term “Semantic Web” refers to W3C’s vision of the Web of linked data.”
- Semantic Web technologies enable people to:
 - create **data stores** on the Web
 - build **vocabularies**
 - write **rules** for handling data.
 - Linked data are empowered by **technologies** such as RDF, SPARQL, OWL, and SKOS.

Specifiche di RDF



- Le specifiche di RDF consistono in una suite di documenti:
- **RDF 1.1 Primer**
- RDF 1.1 Concepts and Abstract Syntax
- RDF 1.1 XML Syntax
- RDF 1.1 Semantics
- RDF Schema 1.1
- RDF 1.1 Test Cases

Carta d'identità di RDF

- Nato per descrivere risorse digitali
- Il suo data model è basato sul concetto di grafo
- I nodi rappresentano entità e gli archi relazioni tra di esse
- Possiede diverse serializzazioni

RDF Data Model

Elementi del data model di RDF:

- Triple
- IRI
- Letterali
- Blank nodes
- Grafi

L'unità di base di RDF è una **tripla**, composta di soggetto, predicato e oggetto. Il soggetto e oggetto possono essere **IRI** o **blank nodes**, l'oggetto può essere anche un **letterale**, il predicato può essere solo un IRI

Un insieme di triple forma un **grafo**

Per raggruppare le triple è possibile identificare singoli grafi

<https://www.w3.org/TR/rdf11-concepts/>

RDF: triple

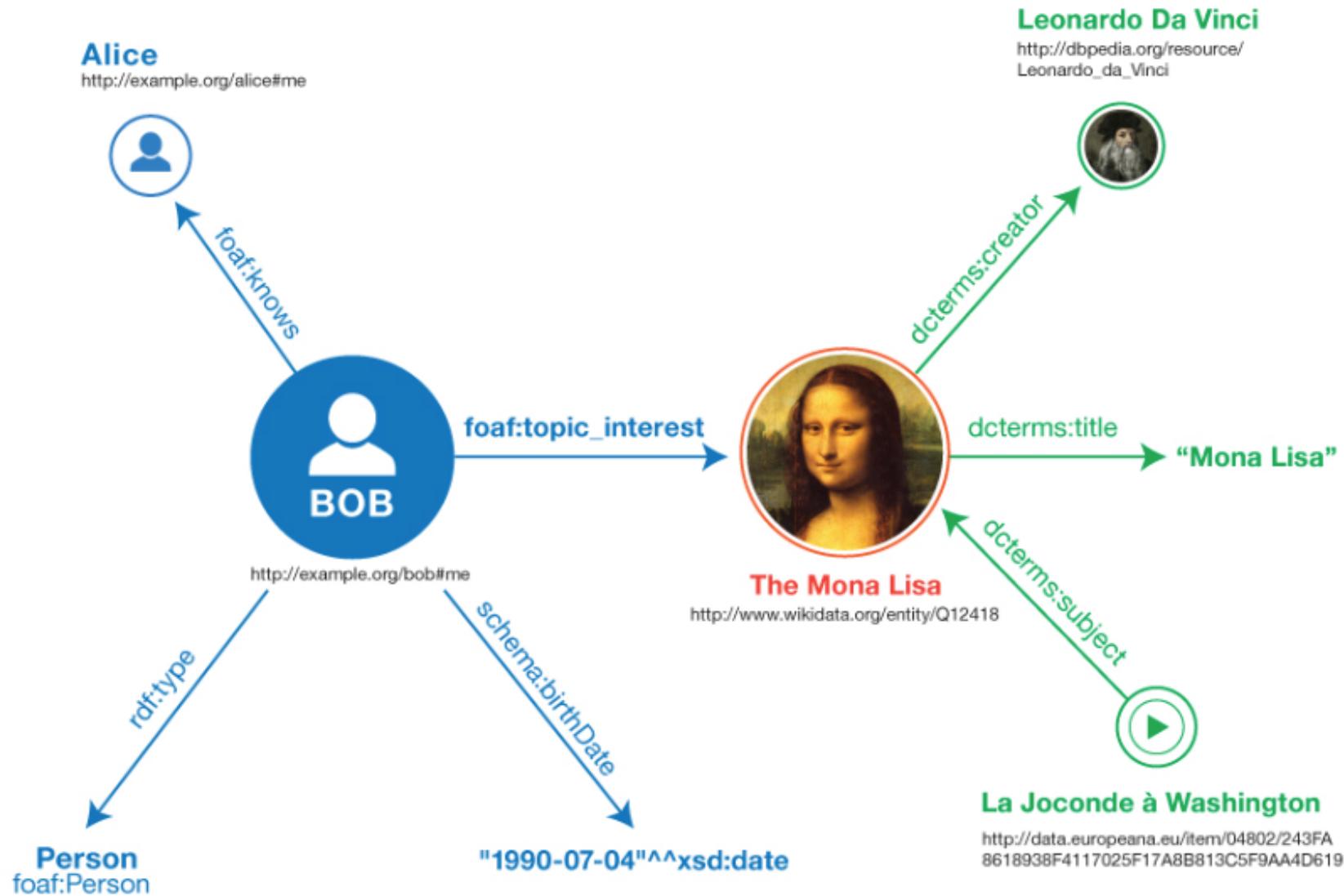
- RDF è formato di **triple**, che hanno la forma
Soggetto – Predicato – Oggetto
- Esempio:
 - Soggetto: Monna Lisa
 - Predicato: Fu creata da
 - Oggetto: Leonardo da Vinci

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

Esempio: un insieme di triple (in pseudocodice)

```
<Bob> <is a> <person>.  
<Bob> <is a friend of> <Alice>.  
<Bob> <is born on> <the 4th of July 1990>.  
<Bob> <is interested in> <the Mona Lisa>.  
<the Mona Lisa> <was created by> <Leonardo da Vinci>.  
<the Mona Lisa> <is entitled> <Mona Lisa>.  
<the video 'La Joconde à Washington'> <is about> <the Mona Lisa>.
```

Dalla tripla al grafo



URI e IRI

- Gli elementi della tripla devono essere riconducibili a entità presenti nel web
- Lo strumento scelto da RDF per rappresentare le entità è dato dal sistema degli IRI
 - IRI, cioè URI Internazionalizzato (RFC 3987)
 - Cosa rappresenti nella realtà un determinato IRI non è rilevante: l'importante è che sia *de-referenziabile*
- Il predicato è sempre rappresentato da un IRI
 - Soggetto e oggetto possono essere costituiti da altri elementi

Letterali

- Qualsiasi tipo di dato che non sia un IRI

- Stringhe
- Numeri in vari formati
- Booleani
- ...

<Bob> <is born on> <the 4th of July 1990>.

<the Mona Lisa> <is entitled> <Mona Lisa>.

- Solo in posizione di **oggetto** nella tripla
- Sostanzialmente coincidono con i data types di XML Schema, anche se il supporto può essere parziale in alcuni software

Blank node

- Permettono di denotare risorse senza utilizzare un IRI
- Un blank node è come una variabile e può comparire in posizione di soggetto e oggetto nella tripla

```
<> <has_x_coord> <12.5>.  
<> <is a> <tree>.
```

- Possono essere associati a un identificativo nella rappresentazione delle triple in un determinato store, ma non hanno significato al di fuori di esso
 - In alcune implementazioni, uno speciale IRI unico può essere generato per ogni blank node, usando la keyword genid:
 - <http://example.com/.well-known/genid/d26677abc1f6>

Usare vocabolari

- Per descrivere una certa entità, la tripla si basa sull'utilizzo di vocabolari condivisi
 - Invece di «inventare» ogni volta i termini per descrivere le entità di cui parla la tripla, meglio usare tutti gli stessi termini!

<Bob><likes><The Mona Lisa>.

<Bob><loves><The Mona Lisa>.

(0 meglio:)

<Bob><http://www.my_vocab#like><The Mona Lisa>.

<Bob><<http://www.likesanddislikes/loves/adores>><The Mona Lisa>.

Meglio concordare in anticipo su un unico termine:

<Bob><http://xmlns.com/foaf/0.1/topic_interest><The Mona Lisa>.

Vocabolari e prefissi

- In molte serializzazioni di RDF, il vocabolario è identificato da un prefisso, che corrisponde a un namespace
- Per esempio:
 - Friend of Friend (FOAF) ha il namespace <http://xmlns.com/foaf/0.1/> a cui corrisponde convenzionalmente il prefisso foaf:
 - Il vocabolario Dublin Core ha come namespace <http://purl.org/dc/elements/1.1/> e come prefisso dc:

Relazioni *instance of* in RDF

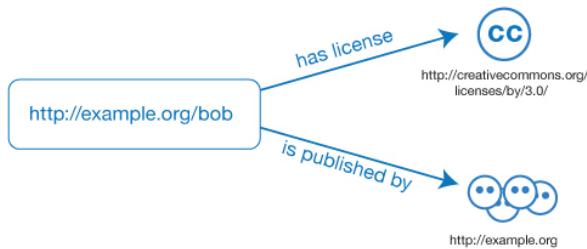
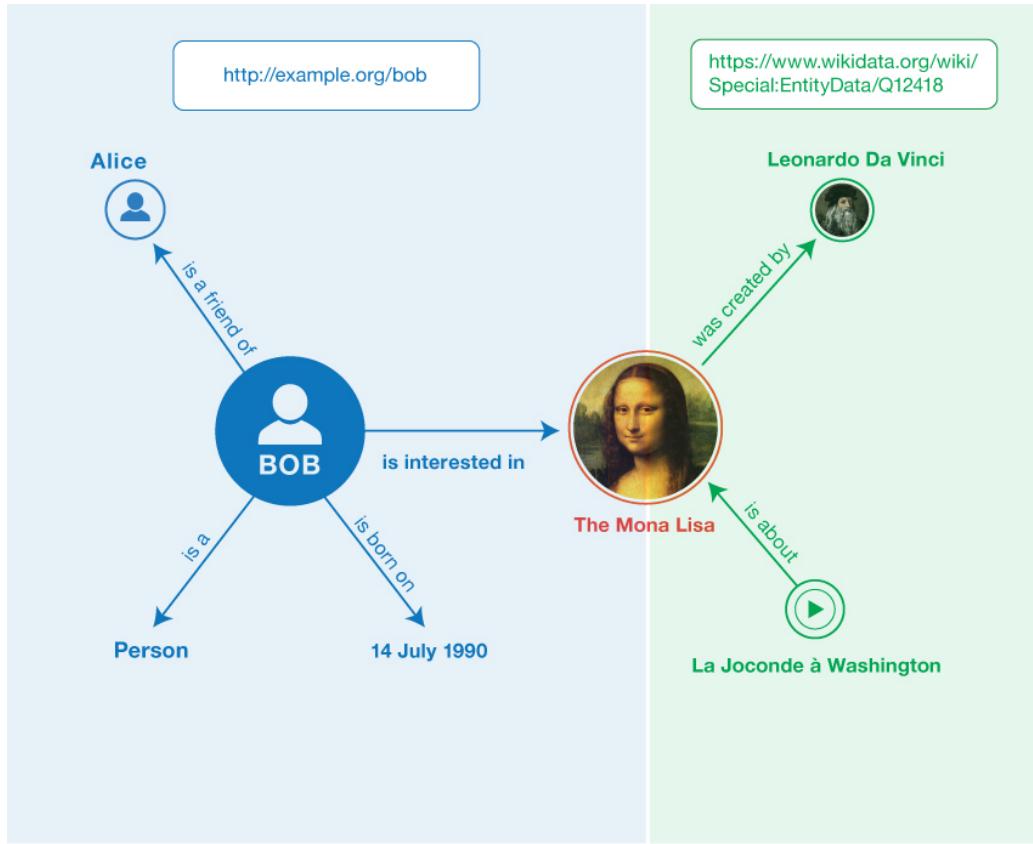
- RDF non permette di specificare il significato dei predicati che mettono in relazione soggetto e oggetto
- Tuttavia, con RDF si può asserire che una risorsa appartiene a una certa tipologia (**classe**) utilizzando il predicato **type**

<the Mona Lisa> <type> <Painting>.

Grafi

- Per gestire le triple è meglio raggrupparle in grafi diversi
 - Novità di SPARQL accolta in RDF 1.1
- Un grafo ha un **identificativo** dato da un IRI (o un blank node)
 - grafo = IRI + triple
- Un **data set RDF** è costituito da un insieme di grafi
 - un grafo anonimo (**default graph**)
 - zero o più grafi con un nome (**named graphs**)

Esempio (dal Primer)



Le triple

BOB is a Person

BOB is born on 1990-07-04

BOB likes The Mona Lisa

Mona Lisa was created by Leonardo_da_Vinci> .

La Joconde à Washington is about Mona Lisa

I grafi

Grafo: <http://example.org/bob>

BOB	is a	Person
BON	is born on	1990-07-04
BOB	likes	The Mona Lisa

Grafo: <http://www.wikidata.org/wiki/SpecialEntity/Data/Q121418>

Mona Lisa	was created by	Leonardo_da_Vinci> .
La Joconde à Washington	is about	Mona Lisa

Anche i grafi possono essere nodi di un grafo :

<http://example.org/bob> *has licence* CC

Serializzazioni di RDF

Formati più comuni:

- RDF/XML
- Turtle
- N-triples
- Json LD
- N3
- RDFA

Turtle

- Storicamente il formato di serializzazione è stato XML (RDF/XML), tuttavia XML si è imposto come standard per la condivisione di documenti
- Preferito a RDF/XML per la maggiore coincisione e leggibilità
- Validatori online:
 - <http://ttl.summerofcode.be>
 - <http://www.easyrdf.org/converter>

Turtle: come si presenta

```
@base <http://example.org/> .  
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .  
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .  
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
@prefix rel: <http://www.perceive.net/schemas/relationship/> .
```

```
<...#green-goblin>  
  rel:enemyOf <#spiderman> ;  
  a foaf:Person ; # in the context of the Marvel universe  
  foaf:name "Green Goblin".
```

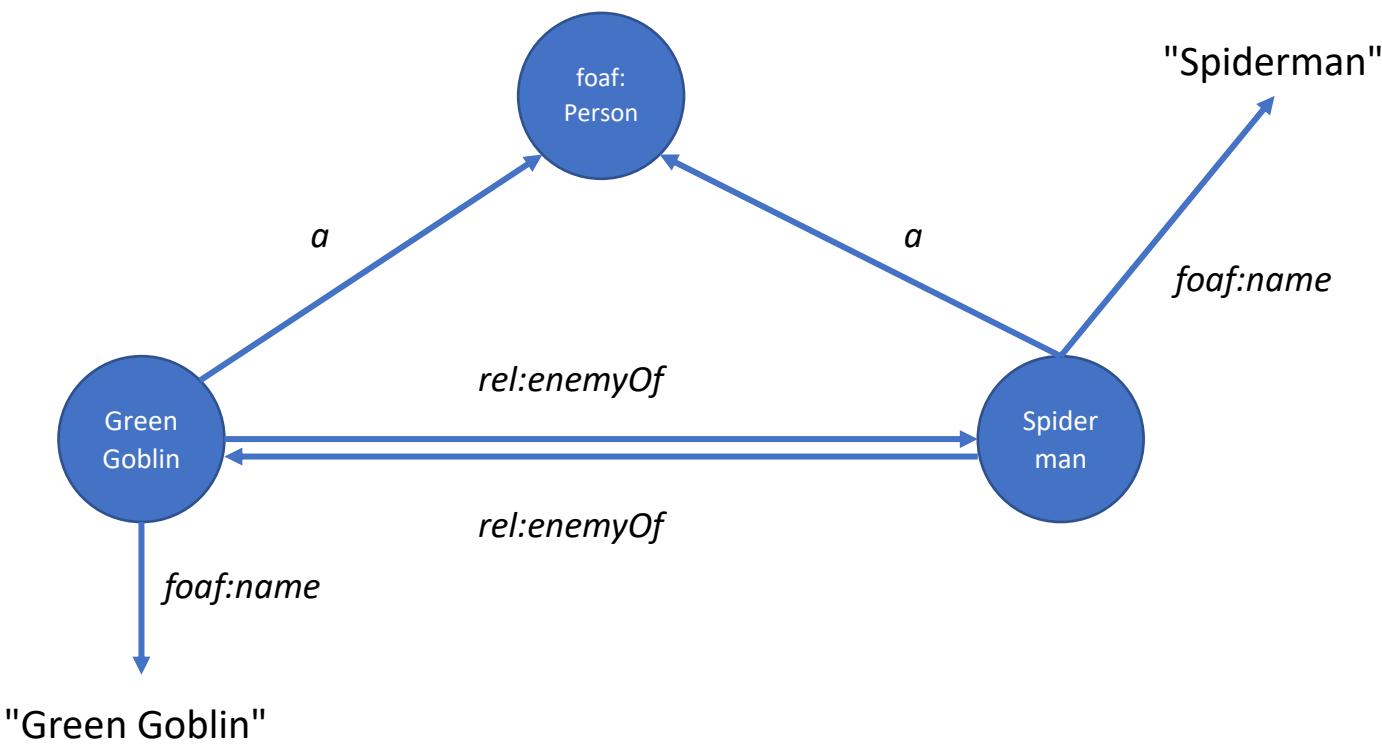
```
<...#spiderman>  
  rel:enemyOf <#green-goblin> ;  
  a foaf:Person ;  
  foaf:name "Spiderman",
```

prefissi

commento

triple

Il grafo



IRI

- Gli IRI sono racchiusi tra '<' and '>'
 1. @prefix foaf: <<http://xmlns.com/foaf/0.1/>> .
 2. <<http://example.org/#green-goblin>> foaf:name "Green Goblin" .
 3. <<http://example.org/#spiderman>> foaf:name "Spiderman" .

La parte finale che segue # identifica un frammento (fragment identifier)

IRI: RDF e Turtle

- In Turtle, tutti gli elementi della tripla possono essere costituiti da un IRI, incluso il predicato

```
:spiderman "http://www.marvel.net/enemyOf" "GreenGoblin"
```

- In RDF/XML, invece, il predicato non può essere costituito da un IRI perché, data la struttura della serializzazione in XML, il predicato viene a essere anch'esso un tag (Per abbreviare quindi si usa il prefisso del namespace)

```
<owl:NamedIndividual rdf:about="http://www.miaontologia#ciccio">
    <rdf:type rdf:resource="http://www.miaontologia#classe1"/>
</owl:NamedIndividual>
```

Stesso soggetto, più predicati

```
:spiderman :enemyOf :green-goblin ;  
           :name      "Spiderman" .
```

“;” indica che allo stesso soggetto si applicano più predicati (con diverso oggetto), equivale a ripetere il soggetto (:spiderman)

:

è il prefisso vuoto

Un predicato con più oggetti

```
:spiderman :name "Spiderman",  
"Uomo Ragno".
```

“,” equivale a ripetere soggetto e predicato

Prefissi

- Turtle supporta notazioni abbreviate per gli IRI, di cui la principale è data dai prefissi:
- Il prefisso sostituisce l'IRI (spesso termina con un fragment identifier #):

```
@prefix p: <http://two.example/> .  
p:subject3 p:predicate3 p:object3 .
```
- La dichiarazione di una base permette di rimuovere il prefisso di fronte agli IRI:

```
@base <http://one.example/> .  
<subject2> <predicate2> <object2> .  
<http://one.example/subject2> <http://one.example/predicate2> ...
```

- La sintassi concreta dei blank node in Turtle è data dalle parentesi quadre []
- [] da solo indica un soggetto o oggetto non specificato
- E' possibile raggruppare nelle parentesi un insieme di triple che hanno il blank node come *soggetto*

```
[  
 :predicate1 :object;  
 :predicate2 value  
 ]
```

Rappresentazione dei blank node

Blank nodes in Turtle

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

Someone knows someone else, who has the name "Bob".

```
[] foaf:knows [ foaf:name "Bob" ] .
```

```
@prefix : <http://example.org/foo> .
```

```
:subject :predicate ( :a :b :c ) .
```

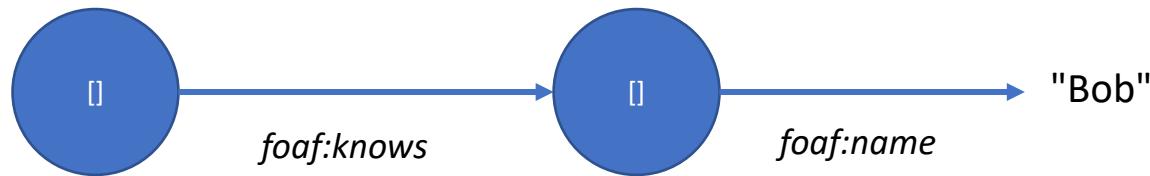
the object of this triple is the collection blank node

In alternativa, si possono rappresentare i blank node con il formato `_: node_label`

Blank nodes nel grafo

Someone knows *someone else*, who has the name "Bob".

[] foaf:knows [foaf:name "Bob"] .



Blank nodes: rappresentazioni a confronto

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .  
[ foaf:name "Alice" ] foaf:knows [ foaf:name "Bob" ;  
    foaf:knows [foaf:name "Eve" ] ;  
    foaf:mbox <bob@example.com>  
] .
```

```
_:a <http://xmlns.com/foaf/0.1/name> "Alice" .  
_:a <http://xmlns.com/foaf/0.1/knows> _:b .  
_:b <http://xmlns.com/foaf/0.1/name> "Bob" .  
_:b <http://xmlns.com/foaf/0.1/knows> _:c .  
_:c <http://xmlns.com/foaf/0.1/name> "Eve" .  
_:b <http://xmlns.com/foaf/0.1/mbox> <bob@example.com> .
```

esercizio: disegnare il grafo

Letterali

```
1. @prefix : <http://example.org/elements> .  
2. <http://en.wikipedia.org/wiki/Helium>  
3.   :name "Helium" ;  
4.   :atomicNumber "2"^^<http://www.w3.org/2001/XMLSchema#integer> ; # xsd:integer  
5.   :specificGravity "1.663E-4"^^<http://www.w3.org/2001/XMLSchema#double> . # xsd:double
```

- Il datatype segue il letterale (tra apici doppi) dopo il deliminatore ^^
- Dato che le stringhe sono molto comuni, si può *omettere* il datatype con i letterali di tipo stringa (è implicito).
→ "Mona Lisa" è uguale a "Mona Lisa"^^xsd:string.
- Dopo una stringa, è possibile apporre il language tag: @it, @en, ecc

Notazione abbreviata per numeri

I valori numerici possono essere scritti in modo esteso (forma lessicale seguita da datatype) oppure in modo **abbreviato**:

- xsd:integer: "-5"^^xsd:integer → -5
- xsd:decimal: "-5.0"^^xsd:decimal → -5.0
- xsd:double: "4.2E9"^^xsd:double → 4.2E9
- Lo stesso vale per i booleani (true/false)

Monna Lisa in Turtle

```
01 BASE <http://example.org/>
02 PREFIX foaf: <http://xmlns.com/foaf/0.1/>
03 PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
04 PREFIX schema: <http://schema.org/>
05 PREFIX dcterms: <http://purl.org/dc/terms/>
06 PREFIX wd: <http://www.wikidata.org/entity/>
07
08 <bob#me>
09   a foaf:Person ;
10   foaf:knows <alice#me> ;
11   schema:birthDate "1990-07-04"^^xsd:date ;
12   foaf:topic_interest wd:Q12418 .
13
14 wd:Q12418
15   dcterms:title "Mona Lisa" ;
16   dcterms:creator <http://dbpedia.org/resource/Leonardo_da_Vinci> .
17
18 <http://data.europeana.eu/item/04802/243FA8618938F4117025F17A8B813C5F9AA4D619>
19   dcterms:subject wd:Q12418 .
```

N-triples

- Sottoinsieme di Turtle **senza abbreviazioni e senza prefissi**
- Verboso e meno leggibile per la mancanza di abbreviazioni
- Migliore per il trasferimento dei data set tra applicazioni
- N-Quads è la sua estensione che aggiunge alla tripla un quarto elemento, il graph IRI

Monna Lisa in N-Triples

```
<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/04802/243FA86...> <http://purl.org/dc/terms/subject> <http://www.wikidata.org/entity/Q12418> .
```

Formati RDF: trade-off

- RDF/XML → strumenti per XML
- Turtle → leggibilità e compatibilità con SPARQL
- N-triples → portabilità dei dati
- Json LD → comunicazione client server via API

<https://ontola.io/blog/rdf-serialization-formats/>

Codifica RDF/XML

```
01 <?xml version="1.0" encoding="utf-8"?>
02 <rdf:RDF
03     xmlns:dcterms="http://purl.org/dc/terms/"
04     xmlns:foaf="http://xmlns.com/foaf/0.1/"
05     xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
06     xmlns:schema="http://schema.org/">
07     <rdf:Description rdf:about="http://example.org/bob#me">
08         <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
09         <schema:birthDate rdf:datatype="http://www.w3.org/2001/XMLSchema#date">1990-07-04</schema:birthDate>
10         <foaf:knows rdf:resource="http://example.org/alice#me"/>
11         <foaf:topic_interest rdf:resource="http://www.wikidata.org/entity/Q12418"/>
12     </rdf:Description>
13     <rdf:Description rdf:about="http://www.wikidata.org/entity/Q12418">
14         <dcterms:title>Mona Lisa</dcterms:title>
15         <dcterms:creator rdf:resource="http://dbpedia.org/resource/Leonardo_da_Vinci"/>
16     </rdf:Description>
17     <rdf:Description rdf:about="http://data.europeana.eu/item/
18                     04802/243FA8618938F4117025F17A8B813C5F9AA4D619">
19         <dcterms:subject rdf:resource="http://www.wikidata.org/entity/Q12418"/>
20     </rdf:Description>
21 </rdf:RDF>
```

Dal documento: <https://www.w3.org/TR/rdf11-primer/>