

# Semantic Univer

WEBINARS (/CATEGORY/WEBINARS/)

BLOG (HTTP://BLOG.CAMBRIDGESEMANTICS.C

WHITE PAPERS (/CATEGORY/WHITE-PAPERS/)

ARTICLES (/CATEGORY/ARTICLES/)

SEMANTIC UNIVERSITY (/BLOG/SEMANTIC-UN

## RDF NUTS & BOLTS

([HTTPS://WWW.CAMBRIDGESEMANTICS.COM/BLOG/SEMANTIC-UNIVERSITY/LEARN-RDF/RDF-NUTS-BOLTS/](https://www.cambridgesemantics.com/blog/semantic-university/learn-rdf/rdf-nuts-bolts/))

## XSD DATATYPE CHEAT SHEET

([HTTPS://WWW.CAMBRIDGESEMANTICS.COM/BLOG/SEMANTIC-UNIVERSITY/LEARN-RDF/XSD-DATATYPE-CHEAT-SHEET/](https://www.cambridgesemantics.com/blog/semantic-university/learn-rdf/xsd-datatype-cheat-sheet/))

## RDF VS. XML

([HTTPS://WWW.CAMBRIDGESEMANTICS.COM/BLOG/SEMANTIC-UNIVERSITY/LEARN-RDF/RDF-VS-XML/](https://www.cambridgesemantics.com/blog/semantic-university/learn-rdf/rdf-vs-xml/))

## WHAT IS JSON-LD?

([HTTPS://WWW.CAMBRIDGESEMANTICS.COM/BLOG/SEMANTIC-UNIVERSITY/LEARN-RDF/RDF-NUTS-BOLTS-2/](https://www.cambridgesemantics.com/blog/semantic-university/learn-rdf/rdf-nuts-bolts-2/))

## Introduction

This set of lessons is an introduction to RDF, the core data Semantic Web technologies. The lessons introduce RDF, p technologies such as XML and JSON. Coverage is meant fo help you follow the material.

If you haven't already completed the lessons in Semantic T context for this set of lessons.

## RDF 101

RDF (Resource Description Framework) is one of the thre SPARQL and OWL.

In particular, RDF is the data model of the Semantic Web. If you store Semantic Web data, it's ir RDF data. If you send Semantic Web data to your friend, it

In this lesson we will introduce RDF.

# Objectives

In this lesson you will learn:

- What RDF is and how it fundamentally differs from XML
- What is meant by a “graph data model”
- How RDF is typically represented visually
- The importance of the URI, and the significance (or lack thereof)

## Today's Lesson

RDF is the foundation of the Semantic Web and what provides the structure for data represented in RDF, including schema describing RDF data.

RDF is not like the tabular data model of relational databases.

### RDF Graphs

In particular, it's a labeled, directed graph. I don't mean “graph” as in a social network.

Therefore you can think of RDF as a bunch of nodes (the data) and edges (the relationships). The nodes and edges have labels.

The term labeled, directed graph will mean a lot to the mathematicians. Here's a simple example here.

This is a complete, valid, visual representation of a small RDF graph. It's immediately obvious to her what it represents.

The nodes of the graph are the ovals and rectangles (ovals represent resources, rectangles represent literals). The labeled arrows that connect nodes to each other. The labels are the property names (in detail in a bit).

Note: the graph nature of RDF is why the logos of Semantic Web and the Semantic University are graph-like. See if you can spot the graph in the logos of Revelytix, or not to mention the RDF logo, pictured at the top of this lesson.

There are three kinds of nodes in an RDF directed graph:

- **Resource nodes.** A resource is anything that can have things associated with it. In a visual representation, resources are represented by ovals.
- **Literal nodes.** The term literal is a fancy word for value. In a visual representation, literals are represented by rectangles. For example, the URI `http://www.cambridgesemantics.com/people/about/rob-gonzalez` is a resource, while the string “Rob Gonzalez” is a literal. In a visual representation, literals are represented by rectangles.
- **Blank nodes.** A blank node is a resource without a URI. Blank nodes are usually recommended to be avoided in general, especially if they are used to represent a specific entity.

Edges can go from any resource to any other resource, or from a resource to a literal, or from a literal to anything at all.

Think about this for a second.

This means that anything in RDF can be connected to anyt

This idea is key. When we talk about Semantic Web technoc (XML, relational databases, BI cubes, etc.), this is the reaso Moreover, creating a new thing is as easy as drawing an ov

If you compare this mentally to the model you might know for basic relationships, such as many-to-many relationship end up adding extra tables and columns (think foreign keys system.

The ability to connect anything together, any time you wan you have! The following video was in Introduction to the S point.

This linking between things is the fundamental capability o

## The Central Importance of the URI

If you want to connect two things in a relational database y many relationship, create join tables), etc. If you want to lir like Informatica. It's just not easily done.

If you consider the XML world, the same thing is true. Coni Connecting things between XML documents requires real not doing that very often.

The fundamental value and differentiating capability of the

The URI is what makes this possible.

URI stands for Universal Resource Identifier. The universa single database (think primary keys), in the Semantic Web databases. This enables us to create linkages between all t

In RDF, resources and edges are URIs. Literals are not; the means in the name). Everything else is, including the edges

If you look at our example above, there are several exampl

- <http://www.cambridgesemantics.com/>
- <http://www.cambridgesemantics.com/people/about/rob>
- `foaf:member` (this is shorthand for <http://xmlns.com/foaf/>)
- `foaf:name` (again, shorthand for <http://xmlns.com/foaf/0.1>)

The first one is the URI for the company Cambridge Semar other two are URIs for the edges that connect the resourc



(<https://www.cambridgesemantics.com>)

You should notice that a couple of the URIs above are URIs, but not all. The ones that are URIs make them a URI?

In short, all URIs are URIs, but not all URIs are URIs. Practitioners stick to using URIs for URIs, and not for other things. Differences if you're curious.

Back to the concept of universality. A URI <http://www.cambridgesemantics.com> that has other information about me, you would also use it to query databases about me, we could query

Easy, right?

Like all theoretically simple models, things are different in practice.

## On the Limits of Universality and RDF Schema Design

There is a major problem with the concept of universality in RDF.

It's impossible to get everyone everywhere to agree on a single vocabulary.

If you read the introductions to Semantic Web technologies, you'll see the importance of the URI. After all, how can you connect things if you don't have a common language for those creating RDF vocabularies.

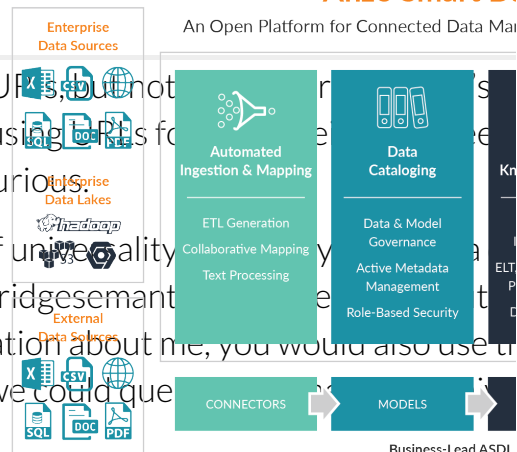
What we mean by an RDF Vocabulary is essentially the set of terms and their relationships that relate the things in graph, and are what give it meaning. The term vocabulary. For example, in order for two Semantic Web applications to share a vocabulary. (Note: we're going to be covering RDF vocabularies in a future post.)

So if two applications have to agree on vocabulary for all concepts ahead of time, right? Fortunately, the a priori existence of a common language for our example, foaf:name is not the first URI ever created that represents the name concept wasn't reused.

Fortunately, as you'll see in the SPARQL tutorials, it is very easy to create a vocabulary. The Semantic Web technologies were built uncoordinatedly for different purposes at different times would create a common language. Therefore there are provisions and methods to make it all work in the XML or relational database worlds.

Said another way, you do not have to agree on all URIs for a single vocabulary when possible and convenient, and don't worry about it.

This same universal identity conundrum also happens for resources. A URI representing me. This is clearly distinct from the URI representing a concept. The Web offers very simple ways to merge identical concepts and



We'll cover the details of how this works in future lessons,

## Statements and Triples

Now that you get the basics, I have to introduce some concepts on the Web about Semantic Web technologies.

Rather than talk in the language of nodes and edges, Semantic Web uses representations of graph edges.

A statement or triple (they are synonymous) refers to a 3-tuple. The linguistic, sentential form is why RDF schemas are often called statements.

As mentioned, the subject is a URI, the predicate is a URI, and the object is a URI.

If we represent our graph example as a set of triples, they would be:

- (csipeople:rob, foaf:name, "Rob Gonzalez")
- (csipeople:rob, foaf:member, http://www.cambridgesemantics.com/people/rob)

(Note that for brevity I'm using the namespace alias csipeople: for the namespace <http://www.cambridgesemantics.com/people/>).

RDF graphs therefore are simply collections of triples. An RDF graph is a set of triples.

However, Semantic Web practitioners found it very difficult to manage large RDF graphs.

There are lots of reasons that you would want to segment a graph (e.g., control, simplified updating, trust, etc.), and vanilla RDF makes this difficult.

At first the community tried using reification to solve this (e.g., but the concept is essentially triples about triples), but today we have named graphs.

## Named Graphs and Quads

A named graph is simply a collection of RDF statements that are associated with a URI.

Modern triple stores all support named graphs, and they are defined in the [RDF 1.1 specification](#).

When referring to a triple in a named graph, you would often use the form:

(named graph, subject, predicate, object)

For this reason, a triple store that supports named graphs and quads stores themselves are often quad stores anyway. That is, if a store supports named graphs, the term quad store isn't that important.

Looking at the 4-tuples, it's pretty obvious that the same statement can be used in a very important feature. By organizing the statement into a named graph, you can control, trust, data lineage, and other functionality very clearly.

Exactly the best ways to segment triples in your application large part of the value brought by Semantic Web platforms and segmentation to simplify application development. For segmentation and organizational mechanism of the Semantic

## Conclusion

This is a lot of information to cover in a single lesson, especially a summary that will become second nature to you if you spend

- RDF is a graph data model.
- RDF data are directed, labeled graphs.
- A single edge in an RDF graph is a 3-tuple that is called either
- Triples are organized into named graphs, forming 4-tuples
- RDF resources (nodes), predicates (edges), and named graphs
- Although preferable to reuse URIs when possible, Semantic Web URI conflicts, as we'll see in future lessons.

< PREVIOUS LESSON

([HTTPS://WWW.CAMBRIDGESEMANTICS.COM/BLOG/SEMANTIC-UNIVERSITY/SEMANTIC-TECHNOLOGIES-APPLIED/SEMANTIC-WEB-ENTERPRISE/](https://www.cambridgesemantics.com/blog/semantic-university/semantic-technologies-applied/semantic-web-enterprise/))

## Solutions

(/solutions/)

Semantic Layer for Hadoop

(/solutions/adsl/#semantic-layer-hadoop)

Business Intelligence

(/solutions/adsl/#business-intelligence)

Data Collection (/solutions/adsl/#data-collection)

Data Integration (/solutions/adsl/#data-integration)

Metadata Management

(/solutions/adsl/#metadata-management)

Search & Discovery

(/solutions/adsl/#search-discovery)

Unstructured Text Integration

## About Us

(/about-us/)

Customers (/about-us/customers/)

Partners (/about-us/partners/)

Analyst Coverage (/category/analyst-coverage/)

In The News (/category/news/)

Events (/category/events/)

Press Releases (/category/press-releases/)

Awards & Recognition (/category/awards-recognition/)

Culture & Careers (/about-us/careers/)

Management Team (/about-us/#team)

## Resources

(/blog/)

Webinars (/category/webinars/)

Blog

(<http://blog.cambridgesemantics.com/>)

Case Studies (/category/case-studies/)

White Papers (/category/white-papers/)

Articles (/category/articles/)

Videos (/videos/)

Training (/training/)

Semantic University (/semantic-university/)

Getting Started Technologies (/semantic-university/intro)

(/solutions/adsl/#text-integration)

## Industry Solutions

Financial Services (/solutions/financial/)

Government (/solutions/government-solutions/)

Life Sciences (/solutions/life-sciences/)

Healthcare (/solutions/healthcare/)

Retail (/solutions/retail/)

Material Sciences (/solutions/material-sciences/)

Semantic Techn  
(/blog/semantic  
semantic-techn  
Semantic Techn  
(/blog/semantic  
technologies-ap

500 Boylston St | Suite 1700 | Boston, MA 0

Copyright © 2018, Cambridge Semantics. All Rights R