

Semantic Web and JRuby

EdgeCase} @carinmeier
SOFTWARE ARTISANS

DUCK

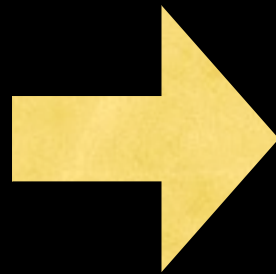
Semantic = Meaning

Welcome to my website!

My name is Johanna

but my friends call me Jo

I have a really cute kitten



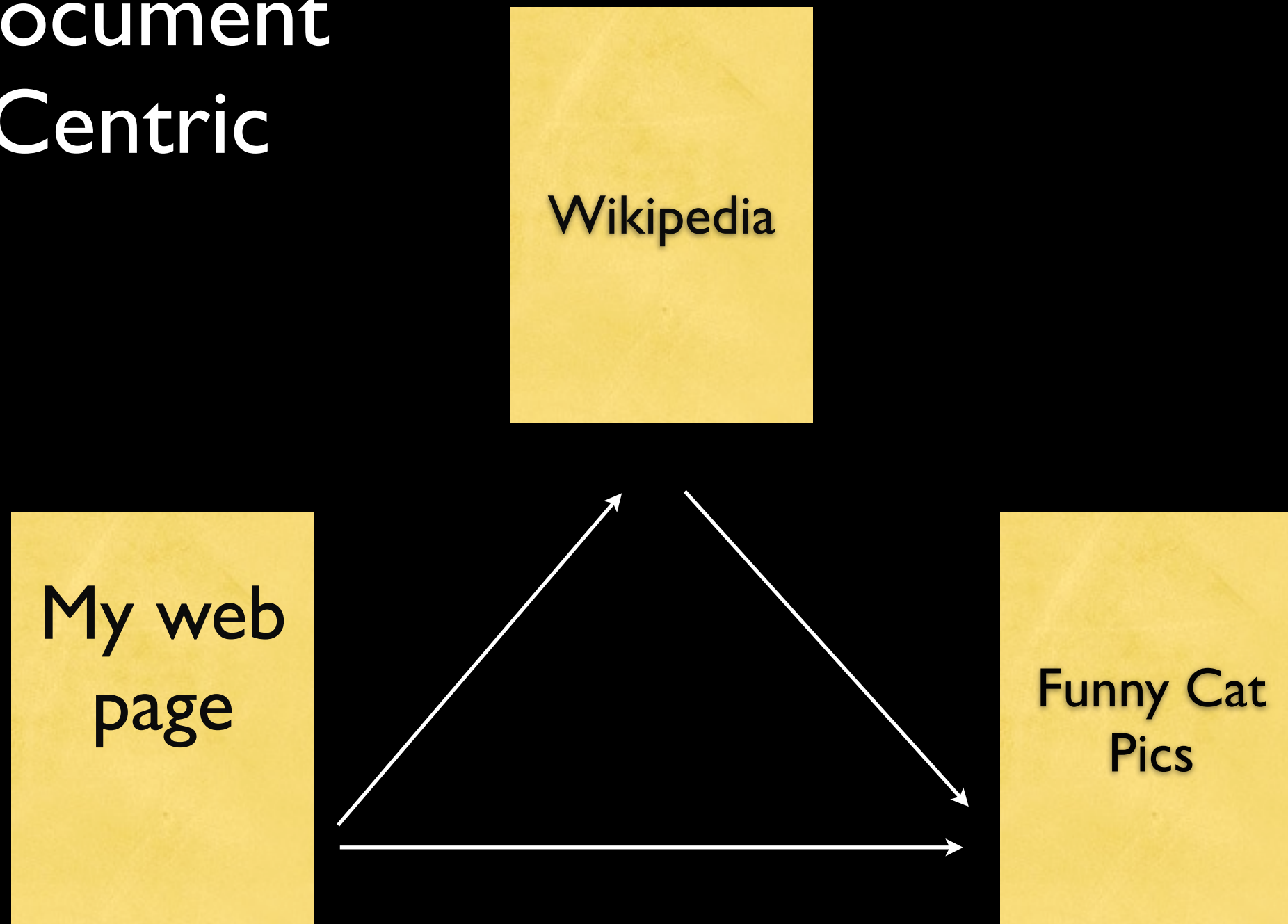
<h1>blah blah</h1>

<h2>blah blah</h2>

<h2> blah blah</h2>

<h2> blah blah<h2>

HTTP Document Centric



DATA

Semantic Web Data Centric







“The Semantic Web is not a separate Web but an extension of the current one in which information is given well-defined meaning, better enabling computers and people to work in cooperation

- Tim Berners-Lee
(Inventor of Web & Director W3C)

Benefits



Context + Meaning =
Better Understanding



Thursday, August 4, 2011



Semantic markup =>
30% increase traffic

Good Relations Vocabulary

```
<xmlns="http://www.w3.org/1999/xhtml"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:gr="http://purl.org/goodrelations/v1#"
  xmlns:vcard="http://www.w3.org/2006/vcard/ns#">

  <typeof="gr:BusinessEntity" about="#company">
    <property="gr:legalName" content="Gourmet Beef Jerkey"></>
    <property="vcard:tel" content="111-222-3333"></>
  </>
</>
```



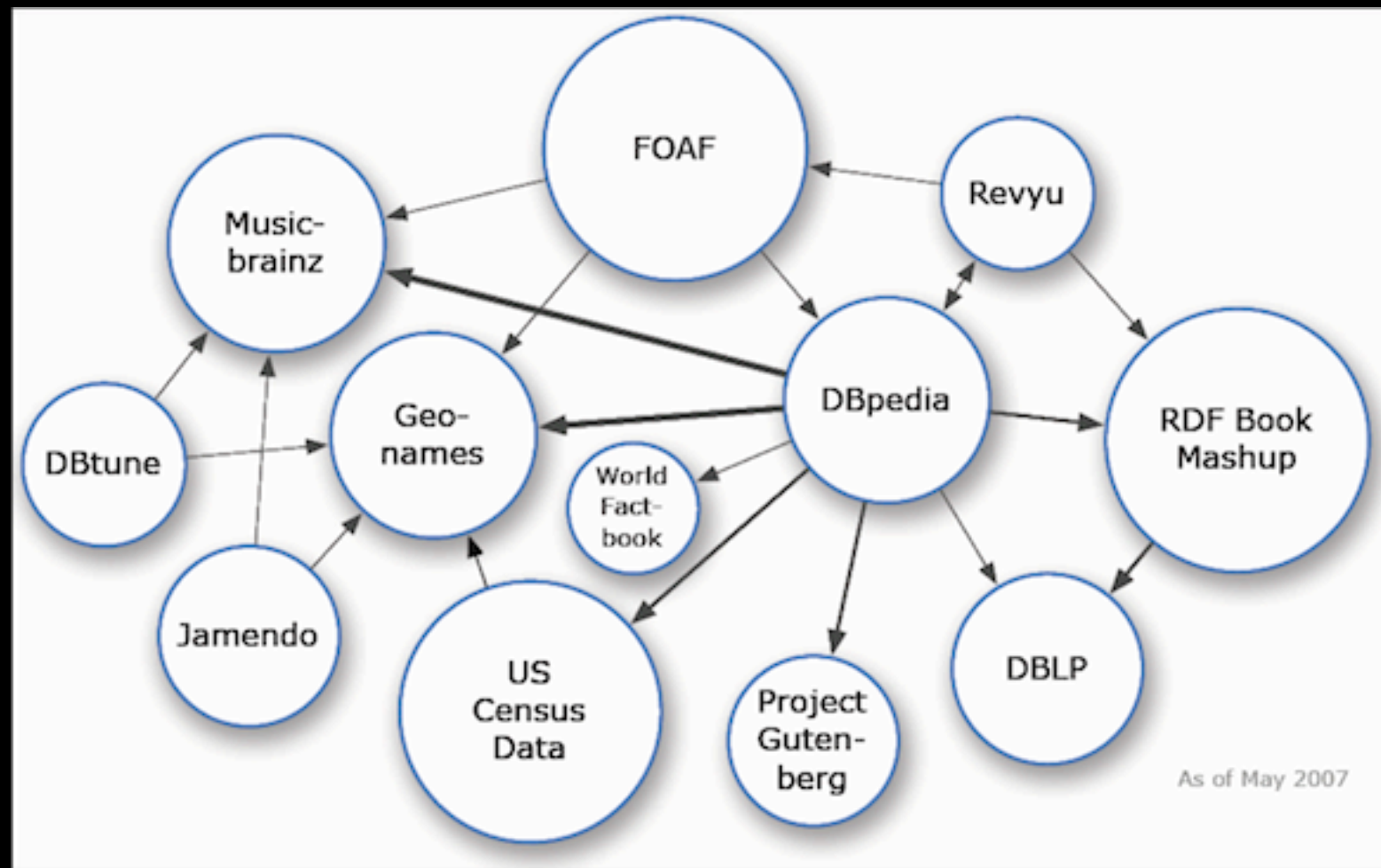

Reasoning of Data =
Advanced Analytics



Oh Yes - Stock Market Investment Firms are on it.



FUTURE



2007

**Now more
Awesome
with
JRuby!**

JENA

Semantic Web Framework

Why JRuby ?

The Jena Library is better and more comprehensive than Ruby alternatives (Java Interop)

Why JRuby ?

The Jena Library is better and more comprehensive than Ruby alternatives (Java Interop)

Ruby is dynamic and more expressive and concise than Java

Why JRuby ?

The Jena Library is better and more comprehensive than Ruby alternatives (Java Interop)

Ruby is dynamic and more expressive and concise than Java

Less code and let's you focus on the problem without being lost in boilerplate.

Resource

URI/ URL

Unique and Resolvable Identifier for the Web

Resource

URI/ URL

Unique and Resolvable Identifier for the Web

~~CAT~~



Resource

URI/ URL

Unique and Resolvable Identifier for the Web

<http://dbpedia.org/resource/Cat>



Create RDF Model

Resource Description Framework

Data Model with basic unit of information is a triple

Create RDF Model

Resource Description Framework

Data Model with basic unit of information is a triple

SUBJECT PREDICATE **OBJECT**

Create RDF Model

Resource Description Framework

Data Model with basic unit of information is a triple

RESOURCE
PROPERTY NAME
PROPERTY

Create RDF Model

Resource Description Framework

```
java_import 'com.hp.hpl.jena.rdf.model.ModelFactory'  
  
model = ModelFactory.create_default_model  
person_uri = "http://somewhere/JohnSmith"  
  
john_smith = model.create_resource(person_uri)
```

Create RDF Model

Resource Description Framework

```
java_import 'com.hp.hpl.jena.vocabulary.VCARD'  
  
full_name = "John Smith"  
  
john_smith.add_property(VCARD::FN, full_name)
```

Create RDF Model

Resource Description Framework

```
model.write(java.lang.System::out)
```

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#" >
  <rdf:Description rdf:about="http://somewhere/JohnSmith">
    <vcard:FN>John Smith</vcard:FN>
  </rdf:Description>
</rdf:RDF>
```

Create RDF Model

Resource Description Framework

```
model.write(java.lang.System::out, "N-TRIPLE")
```

```
<http://somewhere/JohnSmith>
```

```
<http://www.w3.org/2001/vcard-rdf/3.0#FN> "John Smith" .
```



SPARQL

Query DBPedia for places near the White House



```
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84\_pos#>
```

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

```
SELECT ?subject ?label ?lat ?long
```

```
WHERE {
```

```
  <http://dbpedia.org/resource/White\_House> geo:lat ?whLat.
```

```
  <http://dbpedia.org/resource/White\_House> geo:long ?whLong.
```

```
  ?subject geo:lat ?lat.
```

```
  ?subject geo:long ?long.
```

```
  ?subject rdfs:label ?label.
```

```
  FILTER(?lat - ?whLat <= 0.05 && ?whLat - ?lat <= 0.05 &&
```

```
    ?long - ?whLong <= 0.05 && ?whLong - ?long <= 0.05 &&
```

```
    lang(?label) = "en"
```

```
  ).} LIMIT 20
```



```
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84\_pos#>
```

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

```
SELECT ?subject ?label ?lat ?long
```

```
WHERE {
```

```
  <http://dbpedia.org/resource/White\_House> geo:lat ?whLat.
```

```
  <http://dbpedia.org/resource/White\_House> geo:long ?whLong.
```

```
  ?subject geo:lat ?lat.
```

```
  ?subject geo:long ?long.
```

```
  ?subject rdfs:label ?label.
```

```
  FILTER(?lat - ?whLat <= 0.05 && ?whLat - ?lat <= 0.05 &&
```

```
    ?long - ?whLong <= 0.05 && ?whLong - ?long <= 0.05 &&
```

```
    lang(?label) = "en"
```

```
  ).}
```

```
LIMIT 20
```

```
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84\_pos#>
```

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

```
SELECT ?subject ?label ?lat ?long
```

```
WHERE {
```

```
  <http://dbpedia.org/resource/White\_House> geo:lat ?whLat.
```

```
  <http://dbpedia.org/resource/White\_House> geo:long ?whLong.
```

```
  ?subject geo:lat ?lat.
```

```
  ?subject geo:long ?long.
```

```
  ?subject rdfs:label ?label.
```

```
  FILTER(?lat - ?whLat <= 0.05 && ?whLat - ?lat <= 0.05 &&
```

```
    ?long - ?whLong <= 0.05 && ?whLong - ?long <= 0.05 &&
```

```
    lang(?label) = "en"
```

```
  ).} LIMIT 20
```

```
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84\_pos#>
```

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

```
SELECT ?subject ?label ?lat ?long
```

```
WHERE {
```

```
  <http://dbpedia.org/resource/White\_House> geo:lat ?whLat.
```

```
  <http://dbpedia.org/resource/White\_House> geo:long ?whLong.
```

```
  ?subject geo:lat ?lat.
```

```
  ?subject geo:long ?long.
```

```
  ?subject rdfs:label ?label.
```

```
  FILTER(?lat - ?whLat <= 0.05 && ?whLat - ?lat <= 0.05 &&
```

```
    ?long - ?whLong <= 0.05 && ?whLong - ?long <= 0.05 &&
```

```
    lang(?label) = "en"
```

```
  ).} LIMIT 20
```

```
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84\_pos#>
```

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

```
SELECT ?subject ?label ?lat ?long
```

```
WHERE {
```

```
  <http://dbpedia.org/resource/White\_House> geo:lat ?whLat.
```

```
  <http://dbpedia.org/resource/White\_House> geo:long ?whLong.
```

```
  ?subject geo:lat ?lat.
```

```
  ?subject geo:long ?long.
```

```
  ?subject rdfs:label ?label.
```

```
  FILTER(?lat - ?whLat <= 0.05 && ?whLat - ?lat <= 0.05 &&
```

```
    ?long - ?whLong <= 0.05 && ?whLong - ?long <= 0.05 &&
```

```
    lang(?label) = "en"
```

```
  ).} LIMIT 20
```

```
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84\_pos#>
```

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

```
SELECT ?subject ?label ?lat ?long
```

```
WHERE {
```

```
  <http://dbpedia.org/resource/White\_House> geo:lat ?whLat.
```

```
  <http://dbpedia.org/resource/White\_House> geo:long ?whLong.
```

```
  ?subject geo:lat ?lat.
```

```
  ?subject geo:long ?long.
```

```
  ?subject rdfs:label ?label.
```

```
  FILTER(?lat - ?whLat <= 0.05 && ?whLat - ?lat <= 0.05 &&
```

```
    ?long - ?whLong <= 0.05 && ?whLong - ?long <= 0.05 &&
```

```
    lang(?label) = "en"
```

```
  ).} LIMIT 20
```

```
PREFIX geo: <http://www.w3.org/2003/01/geo/wgs84\_pos#>
```

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

```
SELECT ?subject ?label ?lat ?long
```

```
WHERE {
```

```
  <http://dbpedia.org/resource/White\_House> geo:lat ?whLat.
```

```
  <http://dbpedia.org/resource/White\_House> geo:long ?whLong.
```

```
  ?subject geo:lat ?lat.
```

```
  ?subject geo:long ?long.
```

```
  ?subject rdfs:label ?label.
```

```
  FILTER(?lat - ?whLat <= 0.05 && ?whLat - ?lat <= 0.05 &&
```

```
    ?long - ?whLong <= 0.05 && ?whLong - ?long <= 0.05 &&
```

```
    lang(?label) = "en"
```

```
  ).} LIMIT 20
```

Creating the Query Object

```
query = QueryFactory.create(query)
qexec = QueryExecutionFactory.sparql_service(
    "http://dbpedia.org/sparql", query)
```

Iterate Through the Results

```
results = qexec.exec_select
```

```
while results.has_next
```

```
    result = results.next
```

```
    place = result.get("subject").uri
```

```
    label = result.get("label").string
```

```
    latitude = result.get("lat").string
```

```
    longitude = result.get("long").string
```

```
    puts "#{place}"
```

```
    puts "#{label}"
```

```
    puts "lat:#{latitude} long:#{longitude}\n\n"
```

```
end
```

```
qexec.close
```


Iterate Through the Results

```
results = qexec.exec_select

while results.has_next
  result = results.next
  place = result.get("subject").uri
  label = result.get("label").string
  latitude = result.get("lat").string
  longitude = result.get("long").string

  puts "#{place}"
  puts "#{label}"
  puts "lat:#{latitude} long:#{longitude}\n\n"
end

qexec.close
```

Iterate Through the Results

```
results = qexec.exec_select

while results.has_next
  result = results.next
  place = result.get("subject").uri
  label = result.get("label").string
  latitude = result.get("lat").string
  longitude = result.get("long").string

  puts "#{place}"
  puts "#{label}"
  puts "lat:#{latitude} long:#{longitude}\n\n"
end

qexec.close
```

http://dbpedia.org/resource/Nationals_Park

Nationals Park

lat:38.87277603149414

long:-77.00749969482422



http://dbpedia.org/resource/Supreme_Court_of_the_United_States

Supreme Court of the United States

lat:38.89070892333984

long:-77.00434112548828

http://dbpedia.org/resource/White_House

White House

lat:38.89767074584961

long:-77.03655242919922



Let's Find all the DBPedia
Musical Artist Sopranos
with the
Home Town of the United States


```
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX dbowl: <http://dbpedia.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
SELECT ?person ?voiceType ?hometown
```

```
WHERE {
```

```
    ?person rdf:type <http://dbpedia.org/ontology/MusicalArtist>.
```

```
    ?person dbpedia2:voiceType ?voiceType .
```

```
    ?person dbowl:hometown ?hometown .
```

```
    ?person dbowl:hometown <http://dbpedia.org/resource/United\_States>
```

```
    FILTER regex(?voiceType, "soprano", "i")
}
```

```
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX dbowl: <http://dbpedia.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
SELECT ?person ?voiceType ?hometown
```

```
WHERE {
    ?person rdf:type <http://dbpedia.org/ontology/MusicalArtist>.
    ?person dbpedia2:voiceType ?voiceType .
    ?person dbowl:hometown ?hometown .
    ?person dbowl:hometown <http://dbpedia.org/resource/United\_States>
```

```
FILTER regex(?voiceType, "soprano", "i")
}
```

```
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX dbowl: <http://dbpedia.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
SELECT ?person ?voiceType ?hometown
```

```
WHERE {
```

```
    ?person rdf:type <http://dbpedia.org/ontology/MusicalArtist>.
```

```
    ?person dbpedia2:voiceType ?voiceType .
```

```
    ?person dbowl:hometown ?hometown .
```

```
    ?person dbowl:hometown <http://dbpedia.org/resource/United\_States>
```

```
    FILTER regex(?voiceType, "soprano", "i")
}
```



```
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX dbowl: <http://dbpedia.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
SELECT ?person ?voiceType ?hometown
```

```
WHERE {
    ?person rdf:type <http://dbpedia.org/ontology/MusicalArtist>.
    ?person dbpedia2:voiceType ?voiceType .
    ?person dbowl:hometown ?hometown .
    ?person dbowl:hometown <http://dbpedia.org/resource/United\_States>
```

```
FILTER regex(?voiceType, "soprano", "i")
}
```

```
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX dbowl: <http://dbpedia.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
SELECT ?person ?voiceType ?hometown
```

```
WHERE {
    ?person rdf:type <http://dbpedia.org/ontology/MusicalArtist>.
    ?person dbpedia2:voiceType ?voiceType .
    ?person dbowl:hometown ?hometown .
    ?person dbowl:hometown <http://dbpedia.org/resource/United\_States>
```

```
FILTER regex(?voiceType, "soprano", "i")
}
```

```
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX dbowl: <http://dbpedia.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
SELECT ?person ?voiceType ?hometown
```

```
WHERE {
```

```
    ?person rdf:type <http://dbpedia.org/ontology/MusicalArtist>.
```

```
    ?person dbpedia2:voiceType ?voiceType .
```

```
    ?person dbowl:hometown ?hometown .
```

```
    ?person dbowl:hometown <http://dbpedia.org/resource/United\_States>
```

```
    FILTER regex(?voiceType, "soprano", "i")
}
```

```
PREFIX dbpedia2: <http://dbpedia.org/property/>
PREFIX dbowl: <http://dbpedia.org/ontology/>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
```

```
SELECT ?person ?voiceType ?hometown
```

```
WHERE {
    ?person rdf:type <http://dbpedia.org/ontology/MusicalArtist>.
    ?person dbpedia2:voiceType ?voiceType .
    ?person dbowl:hometown ?hometown .
    ?person dbowl:hometown <http://dbpedia.org/resource/United\_States>
```

```
FILTER regex(?voiceType, "soprano", "i")
}
```


http://dbpedia.org/resource/Vanessa_Bell_Armstrong
Mezzo-soprano http://dbpedia.org/resource/United_States

http://dbpedia.org/resource/Diana_Ross Soprano http://dbpedia.org/resource/United_States

http://dbpedia.org/resource/Florence_Ballard Soprano
http://dbpedia.org/resource/United_States

http://dbpedia.org/resource/Lee_Sissing Soprano http://dbpedia.org/resource/United_States

Summary

Semantic Web is all about the Data.

Summary

Semantic Web is all about the Data.

Sharing Data across the Web >
Sharing Documents across the Web

Summary

Semantic Web is all about the Data.

Sharing Data across the Web >
Sharing Documents across the Web

JRuby lets you use Ruby Awesome Sauce with the
Jena Semantic Web Framework Java library.

Want More?

Github: [gigasquid](#)
[jruby-semantic-web-examples](#)

Credits

http://www.flickr.com/photos/a_ninjamonkey/3294014627/

<http://www.flickr.com/photos/jm2c/3677835375/>

<http://www.flickr.com/photos/catdancing/3584504176/>

<http://www.flickr.com/photos/friarsbalsam/4609210182/>

<http://www.flickr.com/photos/lalunablanca/62556584/>

<http://www.flickr.com/photos/rohdesign/2956057741/>

<http://www.flickr.com/photos/photophonic/3388699703/>

<http://richard.cyganiak.de/2007/10/lod/>

http://www.flickr.com/photos/bc_az/4171769959/

<http://www.flickr.com/photos/smemon/4438763529/>

<http://www.flickr.com/photos/catdancing/3584504176/>

<http://www.flickr.com/photos/rohdesign/2956057741/>