Why, What, How
Practical introduction to SPARQL for biologists and informaticians

Using the real world UniProt and neXtProt databases as illustrative examples

Jerven Bolleman (Swiss-Prot)
Daniel Teixeira (CALIPHO)
Pierre-André Michel (CALIPHO)

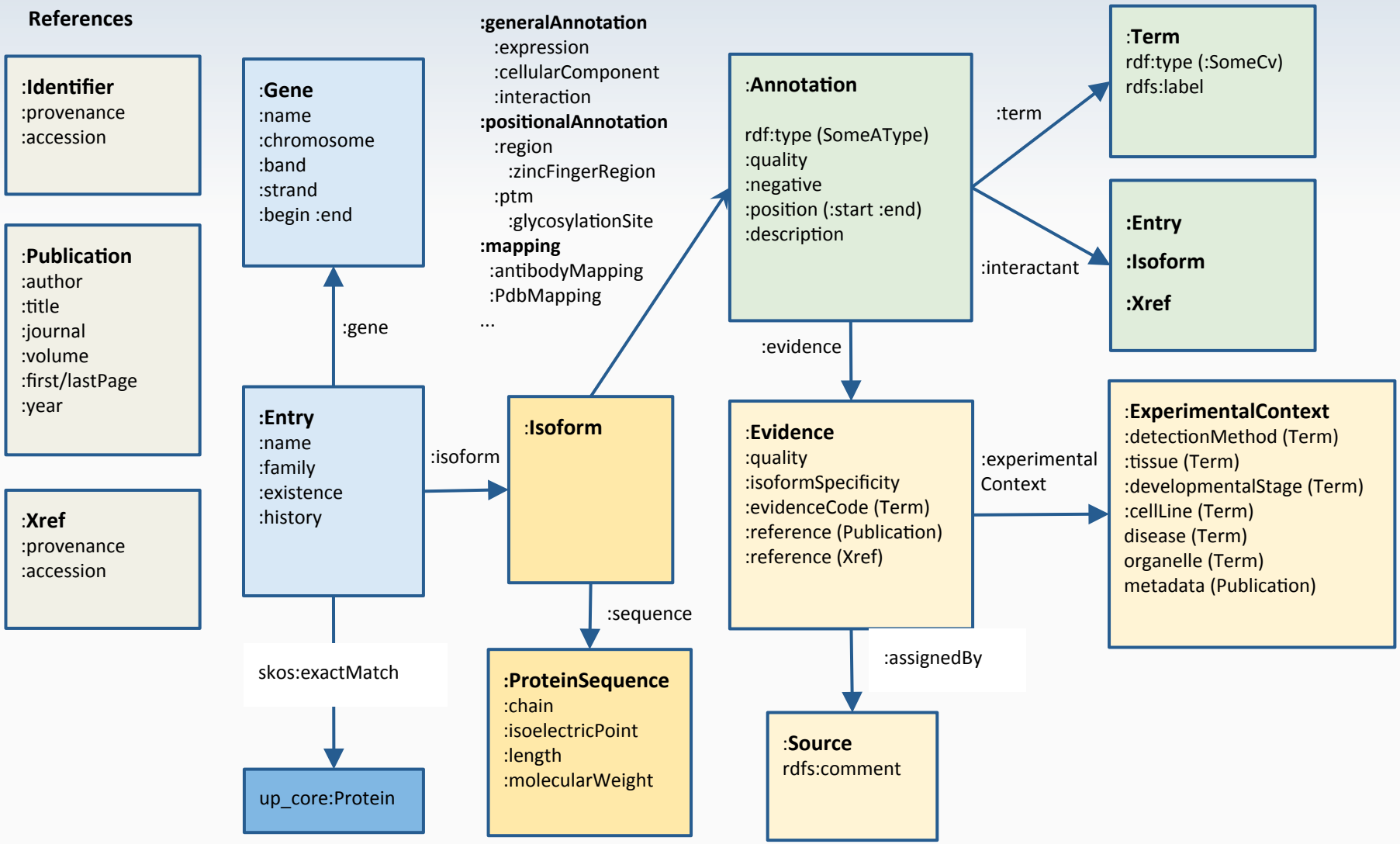
SIB
Swiss Institute of Bioinformatics

elixir SWITZERLAND

Additional slides

Example with UniProt

Documentation – neXtProt data model



Harrison Ford in wikipedia

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikipedia store

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages

Harrison Ford

From Wikipedia, the free encyclopedia

For the unrelated silent film actor, see Harrison Ford (silent film actor).

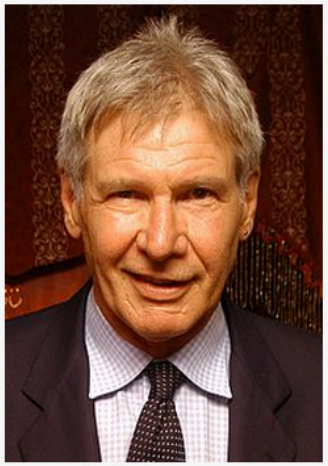
Harrison Ford (born July 13, 1942) is an American actor and film producer. He gained worldwide fame for his starring roles as Han Solo in the original *Star Wars* epic space opera trilogy and the title character of the *Indiana Jones* film series. Ford is also known for his roles as Rick Deckard in the 1982 neo-noir dystopian science fiction film *Blade Runner*, John Book in the 1985 thriller *Witness* and Jack Ryan in the 1992 action-suspense film *Patriot Games* and the 1994 spy action thriller film *Clear and Present Danger*.

His career has spanned six decades and includes roles in several Hollywood blockbusters; including the epic war film *Apocalypse Now* (1979), the legal drama *Presumed Innocent* (1990), the action film *The Fugitive* (1993), the political action thriller *Air Force One* (1997) and the psychological thriller *What Lies Beneath* (2000). At one point, four of the top six box-office hits of all time included one of his roles.[1] Seven of his films have been inducted into the National Film Registry: *American Graffiti* (1973), *The Conversation* (1974), *Star Wars* (1977), *Apocalypse Now* (1979), *The Empire Strikes Back* (1980), *Raiders of the Lost Ark* (1981) and *Blade Runner* (1982).

In 1997, Ford was ranked No. 1 in *Empire*'s "The Top 100 Movie Stars of All Time" list. As of July 2008, the US domestic box office grosses of Ford's films total over US$3.5 billion, with worldwide grosses surpassing $6 billion, making Ford the 4th highest grossing U.S. domestic box-office star.[2] Ford is married to actress Calista Flockhart, who is known for playing the title role in the comedy-drama series *Ally McBeal*.

Contents [hide]

Harrison Ford



Ford in January 2002

Born	July 13, 1942 (age 72) Chicago, Illinois, U.S.
Occupation	Actor · producer
Years active	1966–present
Spouse(s)	Mary Marquardt (m. 1964–79) Melissa Mathison (m. 1983–2004)

Description of http://dbpedia.org/resource/Harrison_Ford subpart 1

About: Harrison Ford Goto Sponge NotDistinct Permalink

An Entity of Type : yago:LivingPeople, within Data Space : dbpedia.org associated with source document(s)

Type: yago:LivingPeople New Facet based on Instances of this Class

Harrison Ford (born July 13, 1942) is an American film actor and producer. He is famous for his
in the original Star Wars trilogy and the title character of the Indiana Jones film series. Ford is als
Rick Deckard in Blade Runner, John Book in Witness and Jack Ryan in Patriot Games and Clear

Attributes	Values
type	Person
	http://schema.org/Person
	person
	Thing
	Body
	»more»
sameAs	fbase:m.0c0k1
	http://fr.dbpedia.org/resource/Harrison_Ford

Description of http://dbpedia.org/resource/Harrison_Ford subpart 2

About: Harrison Ford Goto Sponge NotDistinct Permalink

An Entity of Type : yago:LivingPeople, within Data Space : dbpedia.org associated with source document(s)

Type: yago:LivingPeople New Facet based on Instances of this Class

Harrison Ford (born July 13, 1942) is an American film actor and producer. He is famous for h in the original Star Wars trilogy and the title character of the Indiana Jones film series. Ford is Rick Deckard in Blade Runner, John Book in Witness and Jack Ryan in Patriot Games and Clea

Attributes	Values
spouse	Melissa Mathison
birth date	1942-07-13(xsd:dateTime)
birth place	Chicago, Illinois, U.S.
Caption	Harrison Ford with his Jules Verne Award.
children	5(xsd:integer)
DATE OF BIRTH	1942-07-13(xsd:dateTime)

Description of *rdf:type* property
http://www.w3.org/1999/02/22-rdf-syntax-ns#rdf:type

About: type Goto Sponge NotDistinct Permalink

An Entity of Type : rdf:Property, within Data Space : dbpedia.org associated with source document(s)

Type: Property | New Facet based on Instances of this Class

The subject is an instance of a class.

Attributes	Values
type	Property
label	type
comment	The subject is an instance of a class.
domain	Resource
range	Class
isDefinedBy	The RDF Concepts Vocabulary (RDF)

Description of *rdfs:domain* property
http://www.w3.org/2000/01/rdf-schema#rdfs:domain

About: domain Goto Sponge NotDistinct Permalink

An Entity of Type : rdf:Property, within Data Space : dbpedia.org associated with source docur

Type: Property | New Facet based on Instances of this Class

A domain of the subject property.

Attributes	Values
type	Property
label	domain
comment	A domain of the subject property.
domain	Property
range	Class
isDefinedBy	The RDF Schema vocabulary (RDFS)

Description of http://dbpedia.org/property/spouse

About: spouse Goto Sponge NotDistinct Permalink

An Entity of Type : owl:ObjectProperty, within Data Space : dbpedia.org associated with source document(s)

Type: ObjectProperty | New Facet based on Instances of this Class

the person they are married to

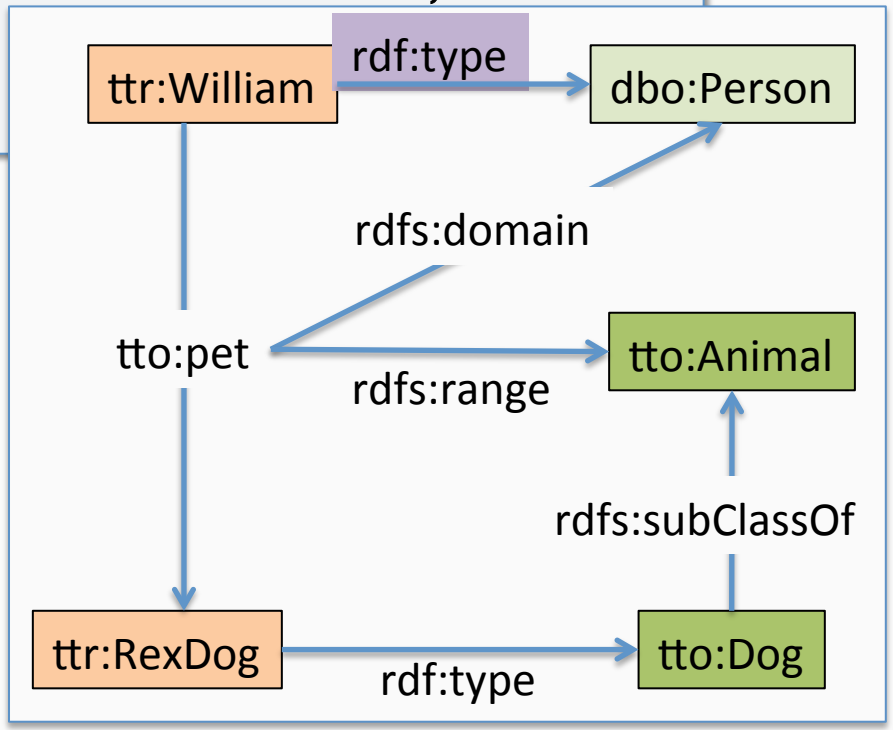
Attributes	Values
type	Property ObjectProperty
subPropertyOf	dul:sameSettingAs
equivalentProperty	http://schema.org/spouse
label	spouse
prov:wasDerivedFrom	http://mappings.dbpedia.org/index.php/OntologyProperty:spouse
comment	the person they are married to
domain	person
range	person

Full description of *rdf:type* property

http://www.w3.org/1999/02/22-rdf-syntax-ns#

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>

# Properties (ontology)
rdf:type a rdf:Property ;
     rdfs:isDefinedBy <http://www.w3.org/1999/02/22-rdf-syntax-ns#> ;
     rdfs:label "type" ;
     rdfs:comment "The subject is an instance of a class." ;
     rdfs:range rdfs:Class ;
     rdfs:domain rdfs:Resource .
```

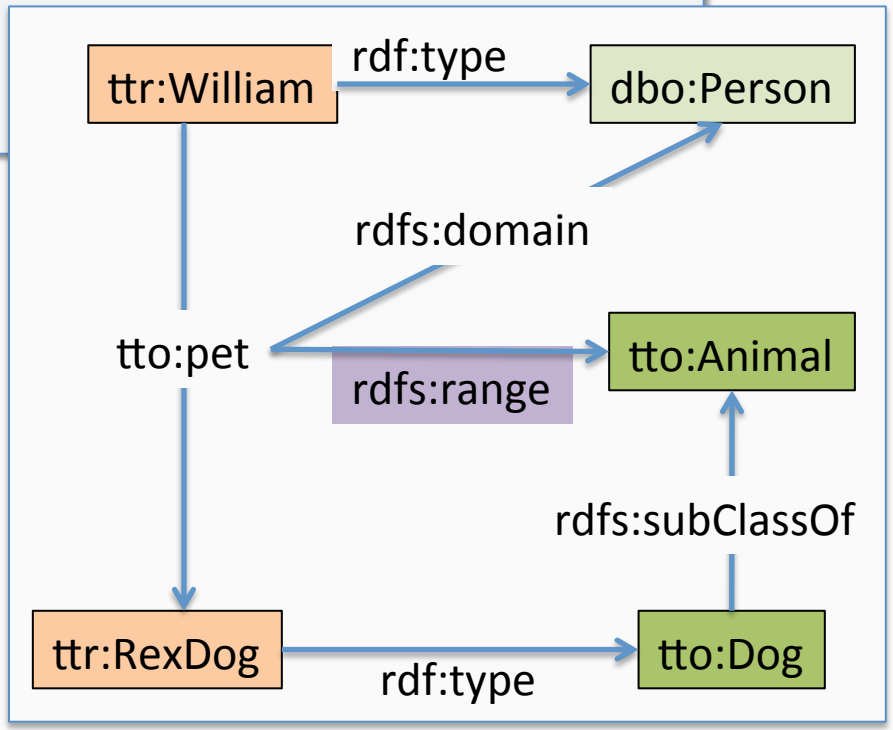


Full description of *rdfs:range* property

http://www.w3.org/2000/01/rdf-schema#

```
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>

# Properties (ontology)
rdfs:range a rdf:Property ;
     rdfs:isDefinedBy <http://www.w3.org/2000/01/rdf-schema#> ;
     rdfs:comment "A range of the subject property." ;
     rdfs:label "range" ;
     rdfs:range rdfs:Class ;
     rdfs:domain rdf:Property .
```



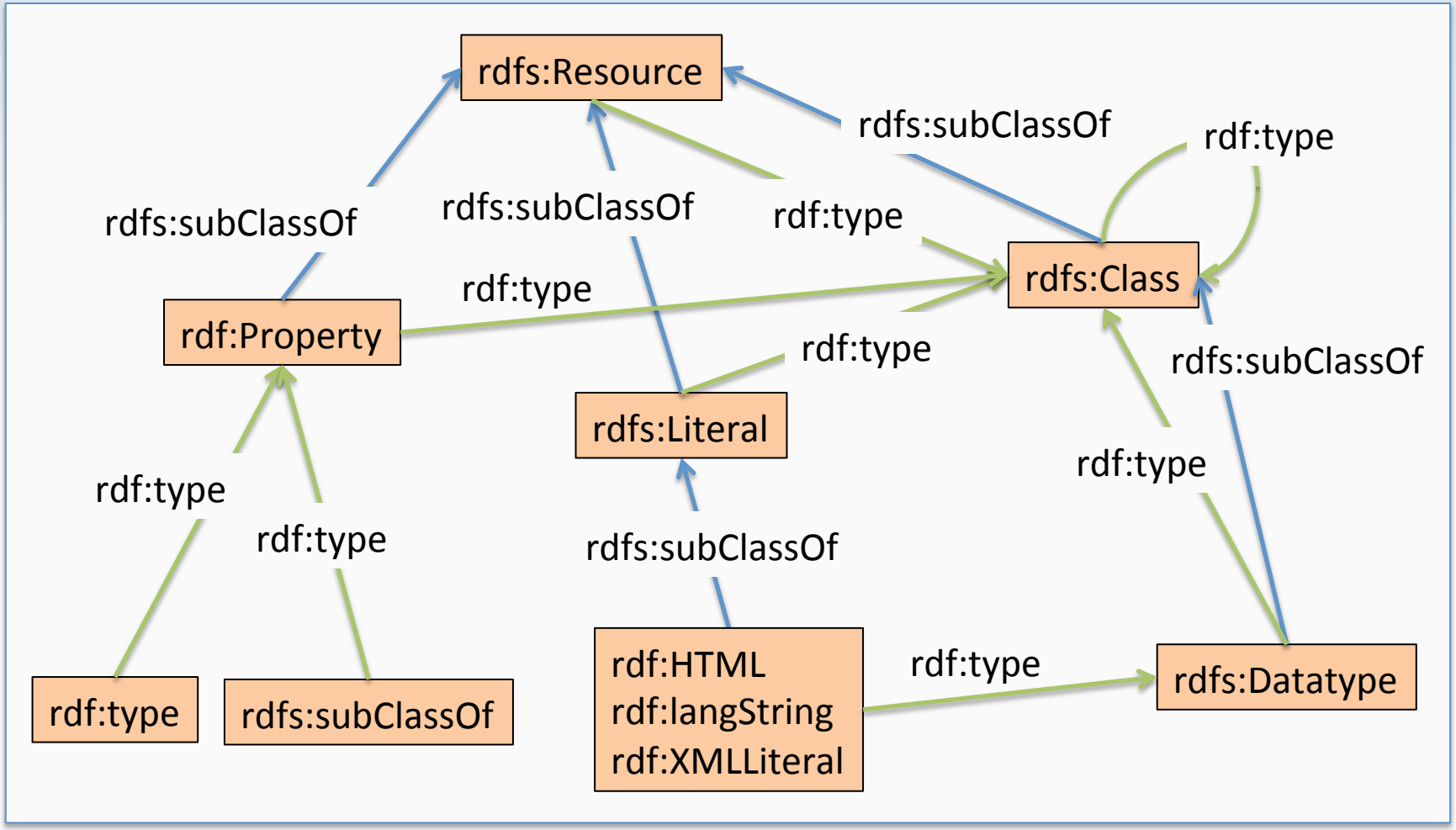
Full RDF Schema description of *tto:Dog* class and *tto:pet* property

```
@prefix tto: <http://example.org/tuto/ontology#> .
@prefix ttr: <http://example.org/tuto/resource#> .
@prefix dbo: <http://dbpedia.org/ontology/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:<http://www.w3.org/2000/01/rdf-schema#>

# Classes (ontology)
tto:Dog a rdfs:Class ;
  rdfs:definedBy <http://example.org/tuto/ontology#> ;
  rdfs:seeAlso <http://example.org/tuto/documentation> ;
  rdfs:label “dog”^^xsd:string ;
  rdfs:comment “the class of dogs”^^xsd:string ;
  rdfs:subClassOf tto:Animal .
...
# Properties (ontology)
tto:pet a rdf:Property ;
  rdfs:definedBy <http://example.org/tuto/ontology#> ;
  rdfs:seeAlso <http://example.org/tuto/documentation> ;
  rdfs:label “pet”^^xsd:string ;
  rdfs:comment “the subject has the object as a pet”^^xsd:string ;
  rdfs:domain dbo:Person ;
  rdfs:range tto:Animal .
...
```

Semantics of RDF Schema terms

(only *rdf:type* & *rdfs:subClassOf* properties)



Datatypes of literal values

XML Schema Built-in Datatypes

- PREFIX xsd: http://www.w3.org/2001/XMLSchema#
- defines xsd:string, xsd:decimal, xsd:integer, xsd:boolean, xsd:date, ...

```
ttr:TomCat tto:weight "18.4"^^xsd:decimal .
ttr:TomCat dbp:name "Tom"^^xsd:string .
```

RDF datatypes

- PREFIX rdf: http://www.w3.org/2000/01/rdf-schema#
- defines rdf:HTML, rdf:XMLLiteral, rdf:langString

```
ttr:TomCat rfs:comment "a cat named Tom"@en .
ttr:TomCat rfs:comment "un chat nommé Tom"@fr .
```

How to provide a SPARQL end point

SPARQL triple store implementations

- OpenLink Virtuoso
- Blazegraph (formerly BigData)
- Jena fuseki
- Sesame
- ...

More tools at http://www.w3.org/2001/sw/wiki/Category:Triple_Store

Connecting to a SPARQL end point

using the *wget* command line utility

```
wget --header="Accept: application/sparql-results-xml" \
"http://localhost:8080/sparql?query=SELECT ?s ?p ?o \
WHERE {?s ?p ?o} LIMIT 10" \
-q -O -
```

HTTP header “Accept”

- application/sparql-results+json
- application/sparql-results+xml
- text/csv
- text/turtle
- application/rdf+xml

HTTP parameters

- query
- default-graph-uri (opt)
- named-graph-uri (opt)
- format (not standard)

Protocol specifications: http://www.w3.org/TR/sparql11-protocol/

Connecting to a SPARQL end point

using the *Jena* API

```
import com.hp.hpl.jena.query.* ;
...
String endpoint = “http://localhost:8080/sparql” ;
String query = ”select * where { ?s ?p ?o } limit 10" ;
QueryExecution qexec =
  QueryExecutionFactory.sparqlService(endpoint, query);
try {
  ResultSet results = qexec.execSelect() ;
  for ( ; results.hasNext() ; )  {
    QuerySolution soln = results.nextSolution() ;
    RDFNode nod = soln.get(”s") ;
    Resource res = soln.getResource(”p") ;
    Literal lit = soln.getLiteral(”o") ;
  }
} finally { qexec.close() ; }
```

Jena API reference:
https://jena.apache.org/documentation/javadoc/arq/overview-summary.html

Connecting to a SPARQL end point

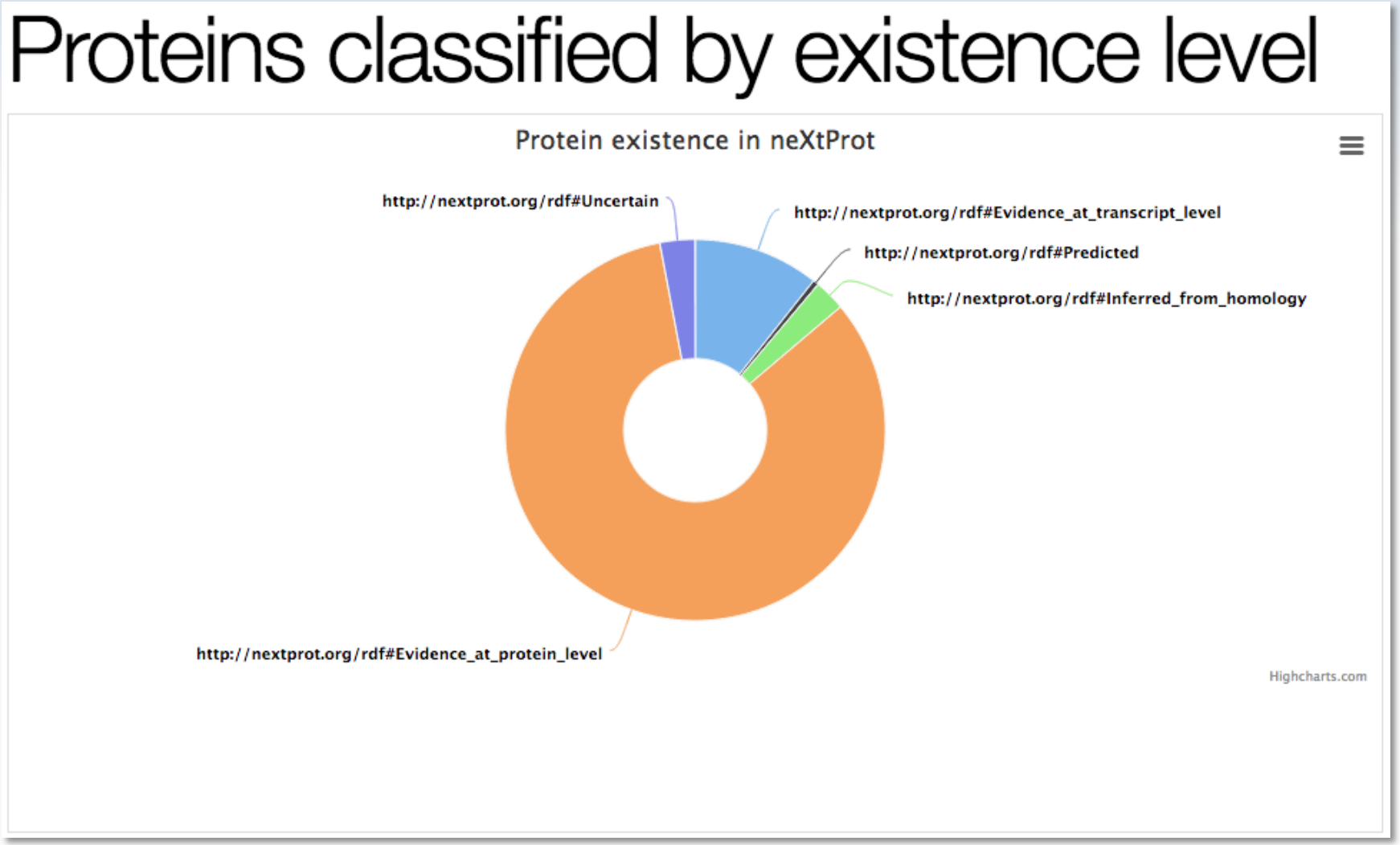
using neXtProt javascript library in an HTML page

```
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script src="http://code.highcharts.com/highcharts.js"></script>
<script src="http://code.highcharts.com/modules/exporting.js"></script>
<script src="https://cdn.rawgit.com/calipho-sib/nextprot-js/v0.0.23/dist/nextprot.min.js"></script>
</head>
<body>
  <div id="plot" style="min-width: 310px; height: 400px; margin: 0 auto"></div>
</body>
  <script type="text/javascript">
      // Create an instance of nextprot API
      var Nextprot = window.Nextprot;
      var applicationName = "demo app for using SPARQL with protein existence";
      var clientInformation = "calipho group at SIB";
      var nx = new Nextprot.Client(applicationName, clientInformation);
      //Define your sparql
      var proteinsByExistenceLevel ='SELECT ?pe count(?entry) as ?cnt ' +
                                    'WHERE {?entry :existence ?pe} group by ?pe';
      //Execute the sparql and retrieve result
      nx.executeSparql(proteinsByExistenceLevel).then(function (result){
        var seriesData = [];
        result.results.bindings.map(function (data) {
          seriesData.push([data.pe.value, parseInt(data.cnt.value)]); //gets number of entries
        });
        //Draw the plot
        $('#plot').highcharts({chart: {type: 'pie',  options3d: { enabled: true, alpha: 45 }},
          title: { text: 'Protein existence in neXtProt' },
          plotOptions: {pie: { innerSize: 100,  depth: 45 } },
          series: [{name: 'neXtProt entries count',data: seriesData }]
        });
      });
  </script>
</html>
```

See http://bl.ocks.org/ddtxra/a1fd0e5613ed6b72ff8f

Connecting to a SPARQL end point

using neXtProt javascript library from an HTML page



See http://bl.ocks.org/ddtxra/a1fd0e5613ed6b72ff8f

The end

Many thanks to

Jerven Bolleman

Daniel Teixeira

Alain Gateau

Monique Zahn

Pascale Gaudet