

# Previously on Introduction to Linked Data...

- You have learned how the internet and the web works
- On the web, URIs are used to identify resources
- With HTTP you can access representations of resources in various formats
- You can identify information resources; information resources represent “files”
- In the following lecture, you will learn how to
  - describe resources using the Resource Description Framework (RDF) and
  - publish and access descriptions of resources in RDF as Linked Data

# C02 The Linked Data Principles

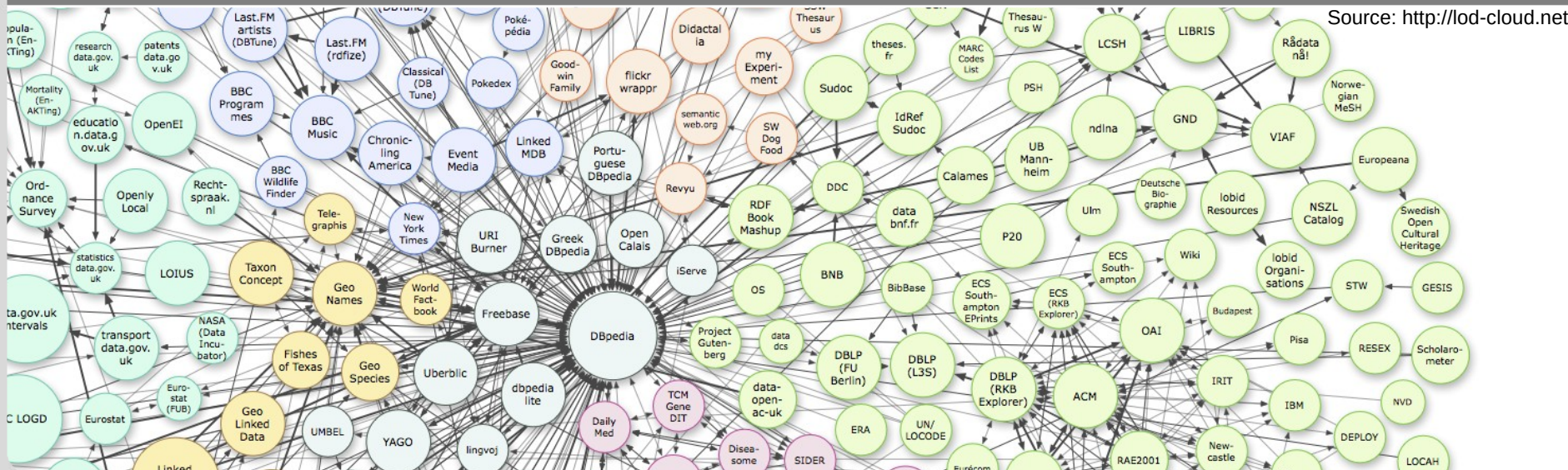
## How to identify and describe things, and access descriptions about things?

Version 2020-05-04

Lecturer: Prof. Dr. Andreas Harth

CHAIR OF TECHNICAL INFORMATION SYSTEMS

Source: <http://lod-cloud.net>



# CC - Creative Commons Licensing

- This set of slides is part of the lecture „Semantic Web Technologies“ held at Karlsruhe Institute of Technology
  - The content of the lecture was prepared by PD Dr. Andreas Harth based on his book „Introduction to Linked Data“
  - The slides were prepared by Benedikt Köhler and Andreas Harth
- 
- **This content is licensed under a Creative Commons Attribution 4.0 International license (CC BY 4.0):**  
<http://creativecommons.org/licenses/by/4.0/>

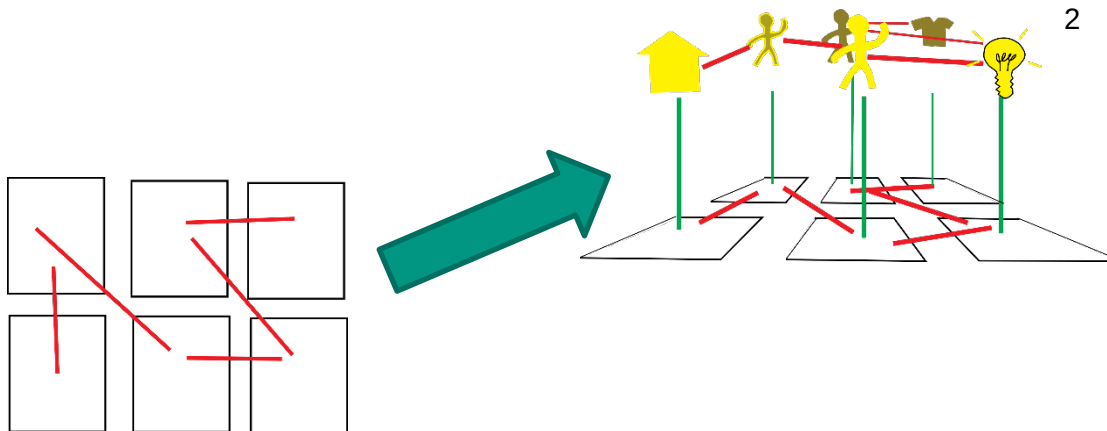


# Agenda

1. **Semantics on the Web**
2. The Resource Description Framework (RDF)
3. Syntaxes for RDF Graphs
4. Turtle Syntax for RDF Graphs
5. Publishing and Accessing Linked Data

# Get the Idea?

- In 2001, Tim Berners-Lee, James Hendler and Ora Lassila published an article in Scientific American entitled “The Semantic Web”
- The article describes the vision of evolving the World Wide Web from a Web of Documents to a Web of Data



<sup>1</sup> <http://www.sciam.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21>

<sup>2</sup> <http://www.w3.org/Talks/WWW94Tim/>

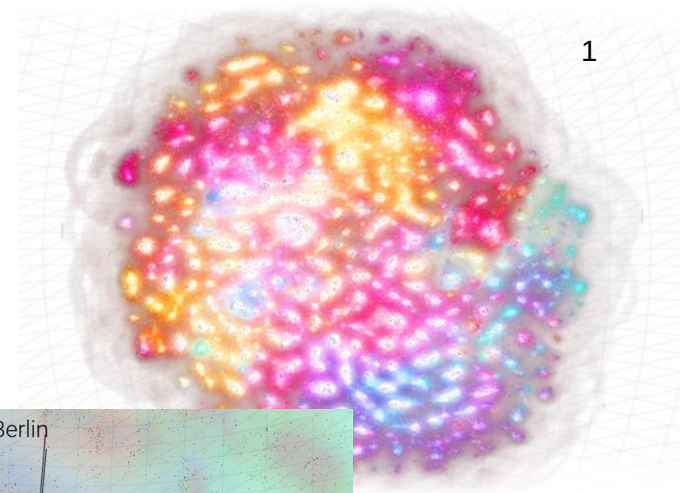


# The Web of Wikipedia

- Consider a graphical representation of pages and hyperlinks within Wikipedia
  - each dot stands for one article
  - articles are ordered thematically through colors
- Page of Berlin is located somewhere in this web
- Many inlinks and outlinks
- Immense amount of information
  - ▢ how can we use that information?

1 <http://wiki.polyfra.me/>

2 <https://en.wikipedia.org/wiki/Berlin/>



1



## Berlin

From Wikipedia, the free encyclopedia

Coordinates: 52°31′N 13°23′E

*This article is about the capital of Germany. For other uses, see Berlin (disambiguation).*

**Berlin** (/bərˈlɪn/, German: [bɛʁˈliːn]  (ⓘ)  (ⓘ)) is the capital of Germany. Berlin is also one of the 16 states of Germany. With a population of 3.5 million people,<sup>[4]</sup> Berlin is Germany's largest city. It is the second most populous city proper and the seventh most populous urban area in the European Union.<sup>[5]</sup> Located in northeastern Germany on the banks of Rivers Spree and Havel, it is the centre of the Berlin-Brandenburg Metropolitan Region, which has about six million residents from over 180 nations.<sup>[9]</sup><sup>[7]</sup><sup>[8]</sup><sup>[9]</sup> Due to its location in the European Plain, Berlin is influenced by a temperate seasonal climate. Around one-third of the city's area is composed of forests, parks, gardens, rivers and lakes.<sup>[10]</sup>

First documented in the 13th century, Berlin became the capital of the Margraviate of Brandenburg (1417–1701), the Kingdom of Prussia (1701–1918), the German Empire (1871–1918), the Weimar Republic (1919–1933) and the Third Reich (1933–1945).<sup>[11]</sup> Berlin in the 1920s was the third largest municipality in the world.<sup>[12]</sup> After World War II, the city was divided; East Berlin became the capital of East Germany while West Berlin became a *de facto* West German exclave, surrounded by the Berlin Wall (1961–1989).<sup>[13]</sup> Following German reunification in 1990, the city was once more designated as the capital of all Germany, hosting 158 foreign embassies.<sup>[14]</sup>

Berlin is a world city of culture, politics, media, and science.<sup>[15]</sup><sup>[16]</sup><sup>[17]</sup><sup>[18]</sup> Its economy is based on high-tech firms and the service sector, encompassing a diverse range of creative industries, research facilities, media corporations, and convention venues.<sup>[19]</sup><sup>[20]</sup> Berlin serves as a continental hub for air and rail traffic and has a highly complex public transportation network. The metropolis is a popular tourist destination.<sup>[21]</sup> Significant industries also include



2

# Accessing Information

- Consider the following simple question:

*What is Berlin's largest borough by population?*

- Intuitive human approach
  - Deduct relevant questions from main question
    1. What are Berlin's boroughs?
    2. What is their population?
  - Combine answers, sort them, chose first borough
- Approach for the web of HTML documents
  - String matching algorithms performed on continuous text

# Limitations of String Matching

- Slight change of query string (district = borough) having same meaning to human reader changes a lot
- Attempt of string matching gets messy really fast

Help: Searching

1

population berlin largest district by population

Search

Results 1 - 20 of 1,972

Content pages Multimedia Everything Advanced

Germany (section Immigrant population)

847 sq mi) with a largely temperate seasonal climate. Its capital and largest city is Berlin. With 81 million inhabitants, Germany is the most populous member 178 KB (17,044 words) - 11:43, 16 August 2015

Berlin

also one of the 16 states of Germany. With a population of 3.5 million people, Berlin is Germany's largest city. It is the second most populous city proper 120 KB (11,948 words) - 09:35, 14 August 2015

Berlin/Brandenburg Metropolitan Region

Potsdam is the only one with a population greater than 100,000 – and 14 districts (Landkreise). The inhabitants of Berlin and Potsdam account for more than 9 KB (573 words) - 21:22, 21 June 2015

? Indonesia

economies. The Indonesian economy is the world's 16th largest by nominal GDP and the 8th largest by GDP at PPP. The Indonesian archipelago has been an important 102 KB (9,938 words) - 22:47, 9 August 2015

Greater Berlin Act

estate district in its own right. The act increased the area of Berlin 13-fold from 66 km² (25.5 mi²) to 883 km² (341 mi²) and the population doubled 4 KB (439 words) - 05:44, 31 May 2015

New Berlin, Wisconsin

New Berlin is a city in Waukesha County, Wisconsin, United States. The population

<sup>1</sup> <https://en.wikipedia.org/>



# Representing Information in Text



Berlin is the capital of Germany. Berlin is also one of the 16 states of Germany. With a population of 3.5 million people, [...]. Located [...] on the banks of Rivers Spree and Havel [...]....

- Berlin is the capital of Germany.
- Berlin is a state of Germany.
- Berlin has a population of 3.5 Million.
- Berlin is located on the bank of the Spree.
- Berlin is located on the bank of the Havel.
- ...
- Pankow is a borough of Berlin.
- Neukölln is a borough of Berlin.

# Representing Information in Triples

- Simplified representation of information follows a continuous **Subject** - **Predicate** - **Object** - structure called triple
- Triple: Resource and its relation to other resources or data values



- Berlin is the capital of Germany.
- Berlin is a state of Germany.
- Berlin has a population of 3.5 Million.
- Berlin is located on the bank of the Spree.
- Berlin is located on the bank of the Havel.
- ...
- Pankow is a borough of Berlin.
- Neukölln is a borough of Berlin.

# Querying Triples

- Given data represented as triples, we can query that data
- A query for the largest borough of Berlin could look like the following:

get every ?x that:

?x [ is a borough of ] [ Berlin ] .

get every ?population that:

?x [ has a population of ] ?population .

sort result by ?population .

Return first result .

- The result of the query over structured Wikipedia data would be a table

?x	?population
<b>Pankow</b>	<b>361602</b>

- SPARQL, a query language for triples, is topic of a following lecture

# Linked Data Principles

- Postulated by Tim Berners-Lee in 2006.

*“The Semantic Web isn't just about putting data on the web. It is about **making links**, so that a person or machine can explore the web of data. With linked data, when you have some of it, you can find other, related, data.”<sup>1</sup>*



- Collection of best practices governing the publication and consumption of data on the web
- Aim: unified method for describing and accessing resources
- Later we will also see how to manipulate resource state

<sup>1</sup> <http://www.w3.org/DesignIssues/LinkedData.html>

# Linked Data Principles<sup>1</sup>

1. Use URIs to **name *things***.
  - Things are not only documents, but also people, locations, concepts, etc.
2. Use HTTP URIs so that users can **look up** those names.
  - Users refer to humans and machine agents alike.
3. When someone looks up a URI, **provide *useful* information**, using the standards (**RDF, RDFS, SPARQL**).
  - What “useful” means depends on the data publisher (but the data publisher should return the “useful” data in RDF).
4. Include **links to other URIs**, so that they can **discover more *things***.

<sup>1</sup> <http://www.w3.org/DesignIssues/LinkedData.html>

# Principle 1: Use URIs as Names for Things

- Point on a distinct resource when you share information
- Linked Data follows a resources-centered view of data modelling
- Resources are the basic concept of web architecture
- Example
  - Assume we would identify a book via its ISBN (9-781497-364783)
  - Using the ISBN scheme from RFC 3187<sup>1</sup> we can use `urn:isbn:9-781497-364783` as resource name for the book



<sup>1</sup> <http://ietf.org/rfc/rfc3187.txt>



## Principle 2: Use HTTP URIs to Allow for Lookup

- Given an identifier for a thing (URI), use HTTP as a mechanism to retrieve more information about that thing
- That is, we require some form of mapping between a
  - **URI as name** (identifying a book, a person, a place or a chemical element) and a
  - **URI as location** (identifying a machine-readable description about the book, the person, the place or the chemical element).

## Principle 2: Use HTTP URIs to Allow for Lookup

- Assume we want to know more about a URI-defined resources, say for our book having the URI `urn:isbn:9-781497-364783`
- With the ISBN you can go to your local bookstore, and a clerk there can look up the ISBN in their catalogue
- Or you type the ISBN into a search box of an online bookstore or of a library, to get more information about the book
- Ultimately, there will be a query to a database of things identified via an ISBN, maintained by some organisation



## Principle 2: Use HTTP URIs to Allow for Lookup

- HTTP URIs provide an inherent mechanism for lookup and unites logical and physical address
- You can type an identifier into your browser and immediately get some information back ⇨ tight connection between identifier and source
- E.g. <http://www.w3.org/People/Berners-Lee/card> is the URI of Tim Berners-Lee's machine-readable homepage
- No additional information or mediator is needed to access information
- Just type HTTP URI into browser and access HTML, JPEG, PNG, GIF, MP4 files – any content that can be serialised into bytes

## Principle 3: Provide Useful Information

- When somebody looks up a URI, provide useful information using the standards
- **RDF is the data model** for both Semantic Web and Linked Data, providing content meaningful to computational users
- Serialisations (machine-readable representation) of RDF:
  - RDF/XML
    - Still most widely used and supported serialization
    - Syntax is tedious to read and write
  - JSON LD
  - Notation 3
  - Turtle
    - Slowly replaces RDF/XML

## Principle 4: Include Links to Other URIs

- Associating things from one source to things from another source creates the mesh we will later use to perform algorithms on
- Links are required to be able to connect the separate data graphs together
- The graph-structured data model and the re-use of URIs across graphs allows for an easy merging of multiple graphs
- Central points on the web provide URIs for frequently used resources (e.g., DBpedia). Using these allows for a common understanding of descriptions and fast merging of multiple graphs



# Agenda

1. Semantics on the Web
- 2. The Resource Description Framework (RDF)**
3. Syntaxes for RDF Graphs
4. Turtle Syntax for RDF Graphs
5. Publishing and Accessing Linked Data

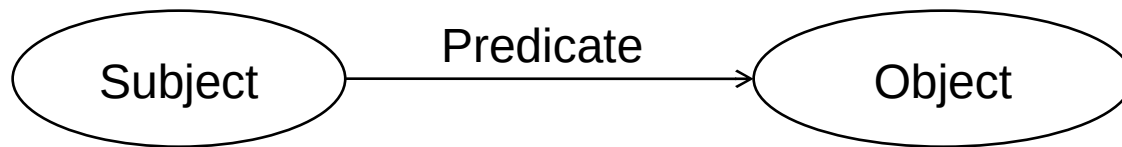


# Resource Description Framework (RDF)



1

- RDF is the foundational data format for both Semantic Web and Linked Data
- An RDF triple is the basic RDF concept describing information as a subject-property-object structure
- Property (or predicate) specifies relation between subject and object
- Triples can be viewed graphically:



- RDF graphs can be presented as directed labelled graph

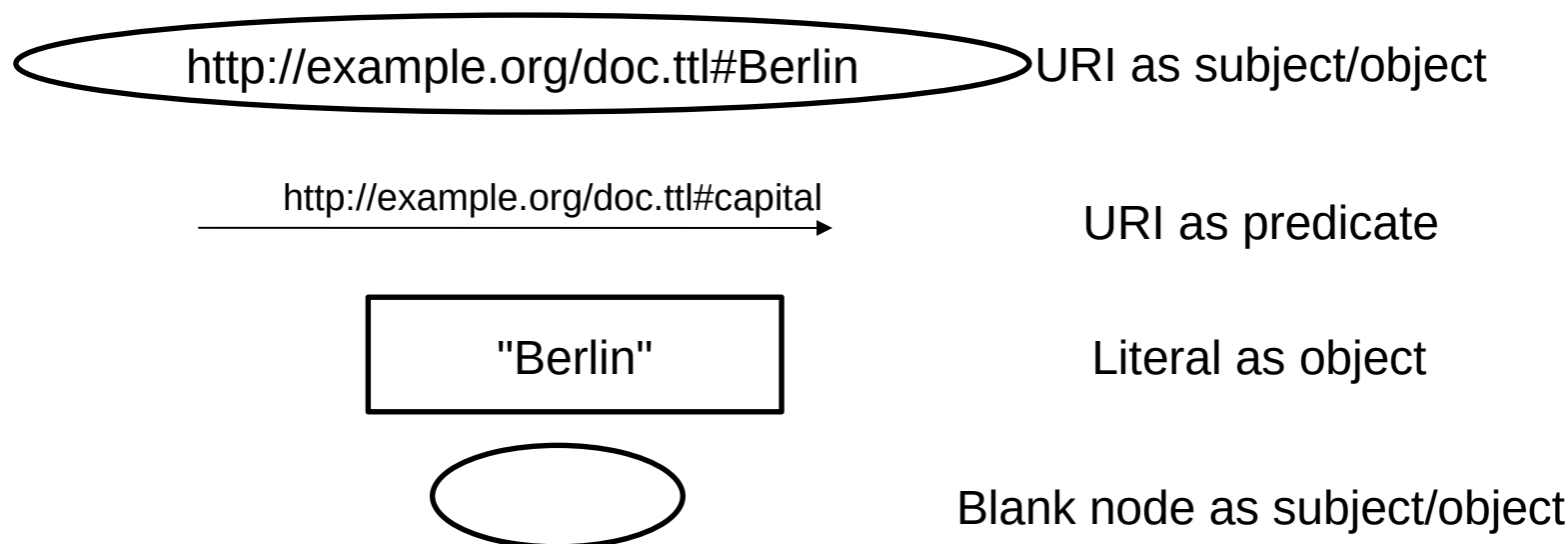
<sup>1</sup> <http://www.w3.org/RDF/icons/>

# RDF Terms: URIs - Blank Nodes - Literals

- URIs are used to globally identify resources
- Blank nodes refer to resources, too, but these resources can only be identified within a file and are not globally addressable (later more)
- Literals refer to concrete data values such as strings, integers, floats or dates. In RDF, we can use the datatypes defined as part of the XML Schema recommendation

# Graphical Representation

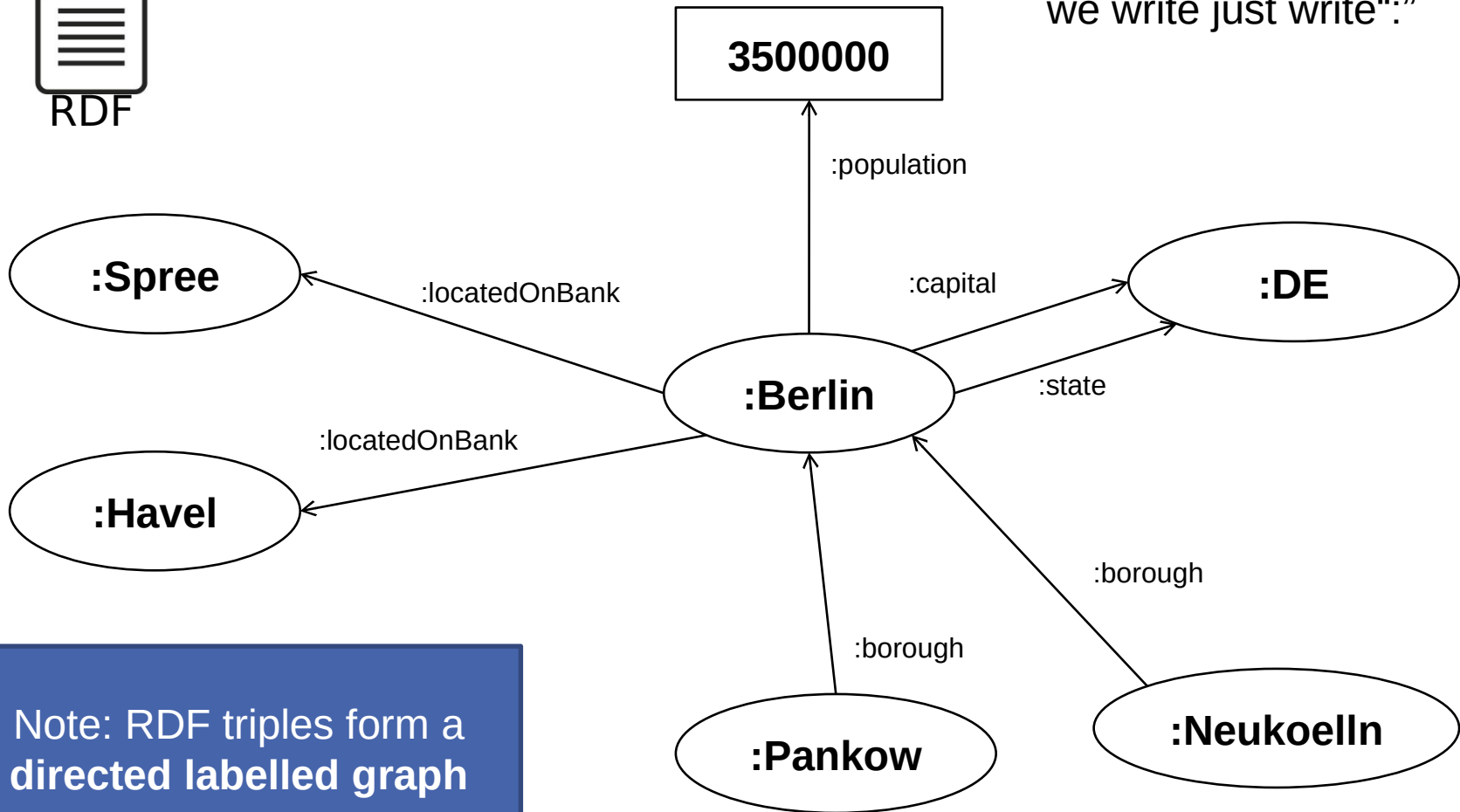
- RDF triples can have URIs or blank nodes on subject position, only URIs on property position, and URIs, blank nodes or literals on object position
- The RDF terms have different shapes in the graphical representation of an RDF graph



# Example RDF Graph



Instead of  
<http://example.org/doc.ttl#>  
we write just write ":"



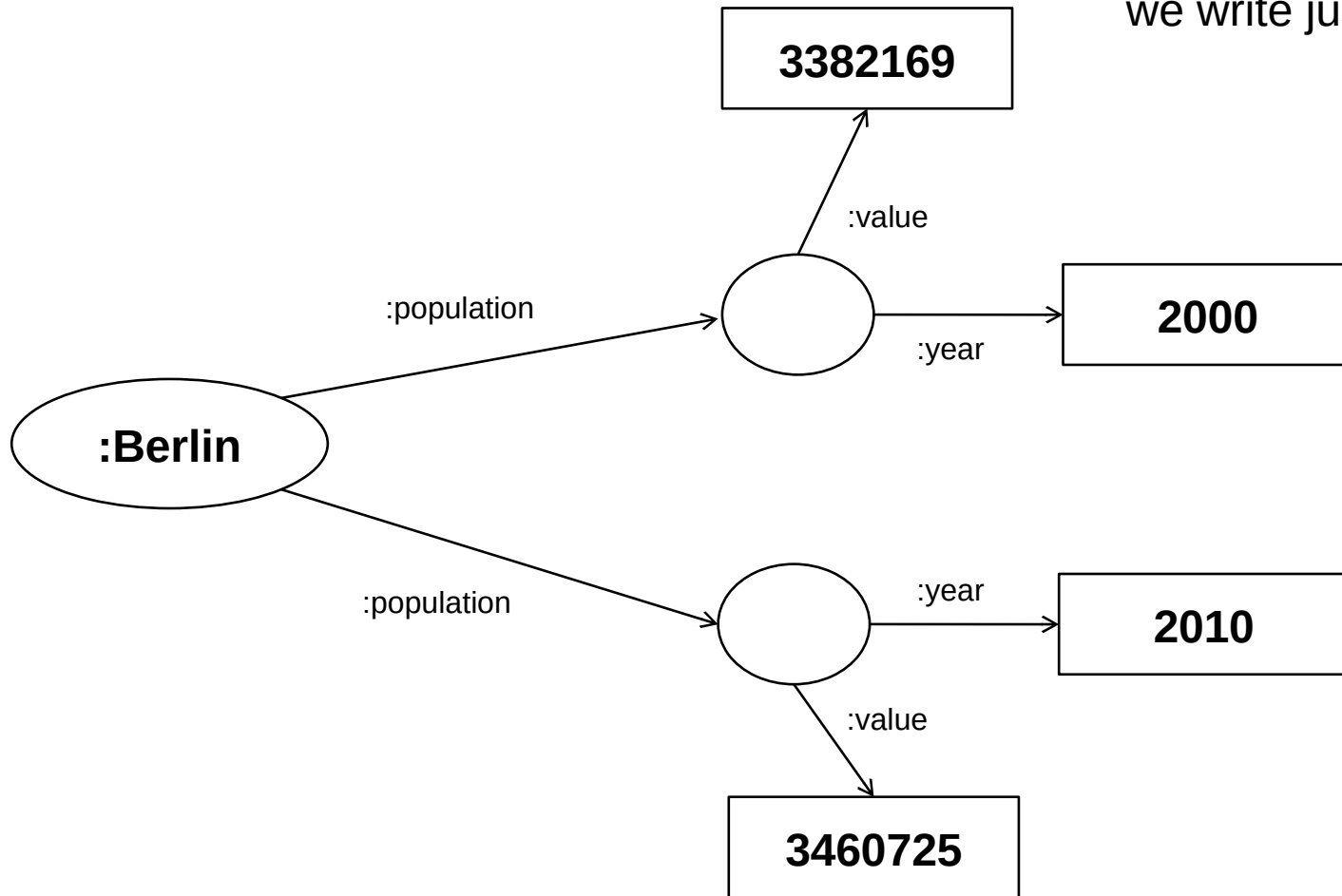
Note: RDF triples form a  
**directed labelled graph**

# N-ary Relations

- An RDF property represents a binary relation between resources
- But there are cases where we want to model relations between more than two resources
- So-called n-ary relations can be modelled as binary relations, if we think of the relation itself as a resource
- Often, we use blank nodes to identify the relation (as resource)

# Example: N-ary Relation in RDF

Instead of  
<http://example.org/doc.ttl#>  
we write just write ":"



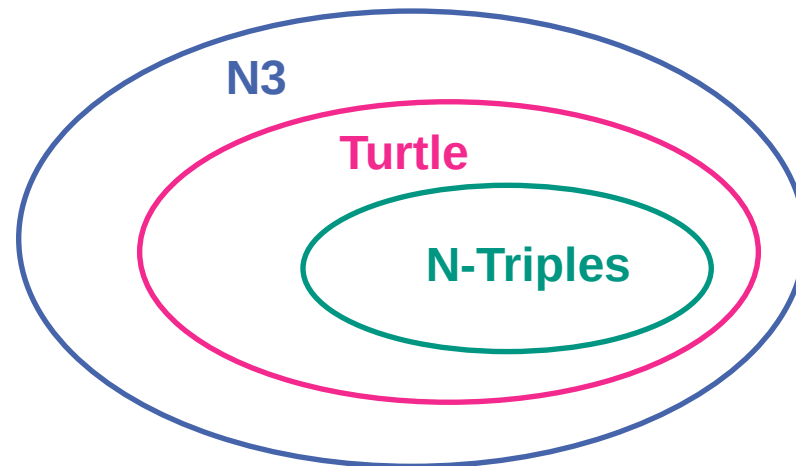


# Agenda

1. Semantics on the Web
2. The Resource Description Framework (RDF)
- 3. Syntaxes for RDF Graphs**
4. Turtle Syntax for RDF Graphs
5. Publishing and Accessing Linked Data

# RDF Syntaxes based on Notation3

- Notation3 (N3) is an expressive textual syntax for a superset of the RDF data model
- Turtle: Terse RDF Triple Language, is a subset of Notation3, specified in an official W3C recommendation<sup>1</sup>
- N-Triples is a very simple syntax, in which one triple is written in one line
- With N-Triples and Turtle we can express the RDF graphs we have seen so far in textual form



<sup>1</sup> <http://www.w3.org/TR/turtle/>

# A Simple Syntax for RDF Triples

- N-Triples provides a very straight-forward way to write down RDF triples



- The basic structure consists of subject property object triples, followed by a dot and a newline
- URIs are enclosed in angle brackets (“<>”)
- Blank nodes are prefixed with an underscore and a colon (“\_:”)
- Literals are enclosed in quotation marks (“””)
- Comments are marked with a hash character (“#”)

```
<http://example.org/doc.ttl#Berlin> <http://example.org/doc.ttl#capital> <http://example.org/doc.ttl#DE> .
```

```
<http://example.org/doc.ttl#Berlin> <http://example.org/doc.ttl#label> "Berlin" .
```

```
<http://example.org/doc.ttl#Berlin> <http://example.org/doc.ttl#population> _:bn . # part of n-ary rel.
```

- The s p o . representation is also called simple triple form

# Typed Literals

- For some information, it is not useful to link to a URI
- Think about numerical values: we would need an infinite number of URIs to map every resource
- We introduce Literals that can only be in the object position of a triple
- Literals can be typed using the XML Schema Datatypes<sup>1</sup>

<sup>1</sup> <http://www.w3.org/TR/xmlschema-2/>

# Commonly Used Primitive XSD Datatypes

Datatype	Description	Example
xsd:string	Character string	Tim
xsd:boolean	Boolean	true
xsd:integer	Arbitrary-size integer numbers	5
xsd:int	32-bit integer numbers	5
xsd:decimal	Real numbers which can be represented by decimal numbers	3.14
xsd:float	IEEE single-precision 32-bit floating point	3.14
xsd:double	IEEE double-precision 64-bit floating point	3.14
xsd:dateTime	Date and time inspired by ISO 8601	2016-06-18T23:59:59.5+00:00

The prefix `xsd` expands to `http://www.w3.org/2001/XMLSchema#`

# Literals with Datatypes in N-Triples

- Literals with are enclosed in quotation marks (""") and use two caret characters ("^^") to specify the datatype URI
- For example, the literal denoting the integer value 1 is written as

`"1"^^<http://www.w3.org/2001/XMLSchema#integer>`



# RDF/XML Example

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns="http://example.org/doc.ttl#">
  <rdf:Description rdf:about="http://example.org/doc.ttl#Berlin">
    <capital rdf:resource="http://example.org/doc.ttl#DE"/>
    <locatedOnBank rdf:resource="http://example.org/doc.ttl#Havel"/>
    <locatedOnBank rdf:resource="http://example.org/doc.ttl#Spree"/>
    <population
rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">3500000</population>
    <state rdf:resource="http://example.org/doc.ttl#DE"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://example.org/doc.ttl#Neukoelln">
    <borough rdf:resource="http://example.org/doc.ttl#Berlin"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://example.org/doc.ttl#Pankow">
    <borough rdf:resource="http://example.org/doc.ttl#Berlin"/>
  </rdf:Description>
</rdf:RDF>
```

# JSON-LD Example

```
{
  "@context": {
    "rdf": "http://www.w3.org/1999/02/22-
rdf-syntax-ns#",
    "rdfs": "http://www.w3.org/2000/01/rdf-
schema#",
    "xsd":
"http://www.w3.org/2001/XMLSchema#"
  },
  "@graph": [
    {
      "@id":
"http://example.org/doc.ttl#Pankow",
      "http://example.org/doc.ttl#borough":
    {
      "@id":
"http://example.org/doc.ttl#Berlin"
    }
  },
    {
      "@id":
"http://example.org/doc.ttl#Berlin",
      "http://example.org/doc.ttl#capital":
    {
      "@id":
"http://example.org/doc.ttl#DE"
    }
  },
    {
      "@id":
"http://example.org/doc.ttl#DE"
    }
  ]
}
```

```
"http://example.org/doc.ttl#locatedOnBank":
[
  {
    "@id":
"http://example.org/doc.ttl#Havel"
  },
  {
    "@id":
"http://example.org/doc.ttl#Spree"
  }
],

"http://example.org/doc.ttl#population":
3500000,
"http://example.org/doc.ttl#state": {
  "@id":
"http://example.org/doc.ttl#DE"
},
{
  "@id":
"http://example.org/doc.ttl#Neukoelln",
  "http://example.org/doc.ttl#borough":
    {
      "@id":
"http://example.org/doc.ttl#Berlin"
    }
}
```

# Think-Pair-Share

- Express the fact that you are attending the Linked Data lecture in simple triples form (N-Triples).

# Agenda

1. Semantics on the Web
2. The Resource Description Framework (RDF)
3. Syntaxes for RDF Graphs
- 4. Turtle Syntax for RDF Graphs**
5. Publishing and Accessing Linked Data

# CURIEs

- Turtle supports CURIE syntax<sup>1</sup> for compact URIs
- The Turtle keyword `@prefix` introduces prefixes used for CURIEs:  
`@prefix : <http://example.org/doc.ttl#> .`
- With the prefix `:` defined, the CURIE `:Berlin` expands to  
`http://example.org/doc.ttl#Berlin`

<sup>1</sup> <http://www.w3.org/TR/curie/>

# Turtle Document with CURIEs

```
@prefix : <http://example.org/doc.ttl#> .  
  
:Berlin :capital :DE .  
  
:Berlin :state :DE .  
  
:Berlin :locatedOnBank :Spree .  
  
:Berlin :locatedOnBank :Havel .  
  
# this is just a comment line  
  
:Pankow :borough :Berlin .  
  
:Neukoelln :borough :Berlin .
```

The prefix has to be defined in each file

CURIEs consist of  
prefix:reference  
pairs

CURIEs written in the form  
prefix:reference  
forgo the angle brackets

We'll take care of the  
population in a minute

# CURIEs – Relative Links

- As you might have noticed, all of last slide's triples refer to resources located in the document itself
- If the document is moved, all of those URIs are invalid as they are absolute
- Therefore, we introduce the relative (default) prefix, always pointing at the location of the document itself. We can assume, for example, that the documents is available at `http://example.org/doc.ttl`:

```
@prefix : <#> .
```

```
:Berlin :capital :DE .  
:Berlin :state :DE .  
:Berlin :locatedOnBank :Spree .  
:Berlin :locatedOnBank :Havel .  
...
```

# Specifying Literals

- To specify the datatype for a literal, we use the double caret ( ^^ )
- For example, `"1"^^xsd:integer` denotes the datatype “integer” of a literal “1”, assuming the prefix mapping from `xsd` to `http://www.w3.org/2001/XMLSchema#`.
- The Turtle specification introduces several abbreviations for datatypes:
  - boolean: `true` is short for `"true"^^xsd:boolean`
  - integer: `-3` is short for `"-3"^^xsd:integer`
  - decimal: `-5.3` is short for `"-5.3"^^xsd:decimal`
  - double: `4.2E9` is short for `"4.2E9"^^xsd:double`<sup>1</sup>
- Since RDF 1.1, all literals that do not contain a datatype are assumed to be of datatype `xsd:string`

<sup>1</sup> Scientific notation (“E”) is reserved for `xsd:float` and `xsd:double`.



# Typed Literal Example

- Expressing the population and the yearly average temperature of Karlsruhe:

```
@prefix : <http://example.org/doc.ttl#> .  
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .  
  
:Karlsruhe :population "300000"^^xsd:integer .  
:Karlsruhe :yearlyAverageTemp "10.5"^^xsd:decimal .
```

- Or using the Turtle abbreviations for literals with datatype `xsd:integer` and `xsd:decimal`:

```
@prefix : <http://example.org/doc.ttl#> .  
  
:Karlsruhe :population 300000 .  
:Karlsruhe :yearlyAverageTemp 10.5 .
```

# Repeating Subject

- We can even be more „terse“
- Several predicate/object pairs relating to one subject can be put in one statement by using a semicolon ( ; )

```
@prefix : <http://example.org/doc.ttl#> .
```

```
:Berlin :capital :DE ; :locatedOnBank :Spree .
```

- Only use one dot ( . ) to finish the statement containing multiple triples

# Repeating Subject and Predicate

- It's also possible that several objects have the same relation to a subject
- In this case, use a comma ( , ) to denote repeated subject and predicate, so only objects have to be written

```
@prefix : <http://example.org/doc.ttl#> .
```

```
:Berlin :locatedOnBank :Spree , :Havel .
```

- Again: only use one dot ( . ) to finish the statement!

# Turtle Document using Shortcuts

- The information about Berlin can be expressed in the following Turtle document (accessible at <http://example.org/doc.ttl>):

```
@prefix : <#> .
```

```
:Berlin :capital :DE ; :state :DE ;  
:locatedOnBank :Spree , :Havel ;  
      :population 3500000 .
```

```
:Pankow :borough :Berlin .
```

```
:Neukoelln :borough :Berlin .
```

Finish longer  
statements  
with one dot

There is no  
possibility to  
shorten the last  
two  
expressions

# Language Tags

- Untyped literals can include language tags with an “@” character ( @) to denote the language of the text in the literal
- Language tags are, for example en for English, fr for French or pt for Portuguese

```
@prefix : <http://example.org/doc.ttl#> .
```

```
:DE :label "Germany"@en .
```

```
:DE :label "Deutschland"@de .
```

```
:DE :label "Alemania"@es .
```

- Language tags are listed in RFC 5646<sup>1</sup>

<sup>1</sup> <http://tools.ietf.org/rfc/rfc5646.txt>

# Blank Nodes

- Blank nodes are treated as simply indicating the existence of a thing, without using, or saying anything about, the name of that thing<sup>1</sup>
- Blank nodes start with an underscore and colon character ( `_:` ), followed by their label
- For example, the label of the blank node `_:bn` is `bn`
- Used in a triple in Turtle syntax:

```
@prefix eg: <http://example.com/eg#> .
```

```
_:bn eg:knows eg:bob .
```

- The triple states that „there exists a resource that is connected to the resource `eg:bob` by the predicate `eg:knows`
- Or, more intuitively, there is someone who knows Bob

<sup>1</sup> <http://www.w3.org/TR/2002/WD-rdf-mt-20020429/#unlabel>

# Using Blank Nodes

- Blank nodes can be used several times in a document

```
@prefix eg: <http://example.com/eg#> .
```

```
_:bn1 eg:knows _:bn2 .
```

```
_:bn2 eg:name "Bob" .
```

- Someone knows someone else who has the name „Bob“
- Blank nodes can only appear in subject or object position
- Blank nodes can **not** be referenced by an external source

# Nesting Blank Nodes

- Turtle provides a shortcut for unlabelled blank nodes via `[]`, which establishes a unique unlabelled blank node
- To be able to re-use unlabelled blank nodes, Turtle allows for nesting
- With the nesting syntax, a fresh blank node is used in the subject and object position of the surrounding predicate/object and subject/predicate pairs:

```
@prefix eg: <http://example.com/eg#> .
```

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

- The parser creates blank node labels (a, b) and expands to the following simple triples:

```
@prefix eg: <http://example.com/eg#> .
```

```
_:a <http://example.com/eg#knows> _:b .
```

```
_:b <http://example.com/eg#name> "Bob" .
```



## Example: N-ary Relation in Turtle Syntax

- Expressing the population of Berlin in 1990:

```
@prefix : <http://example.org/doc.ttl#> .  
:Berlin :population [ :inhabitants 3433695 ;  
                      :year 1990 ] .
```

- Expressing the population of Berlin in 2000:

```
@prefix : <http://example.org/doc.ttl#> .  
:Berlin :population [ :inhabitants 3382169 ;  
                      :year 2000 ] .
```

- Expressing the population of Berlin in 2010:

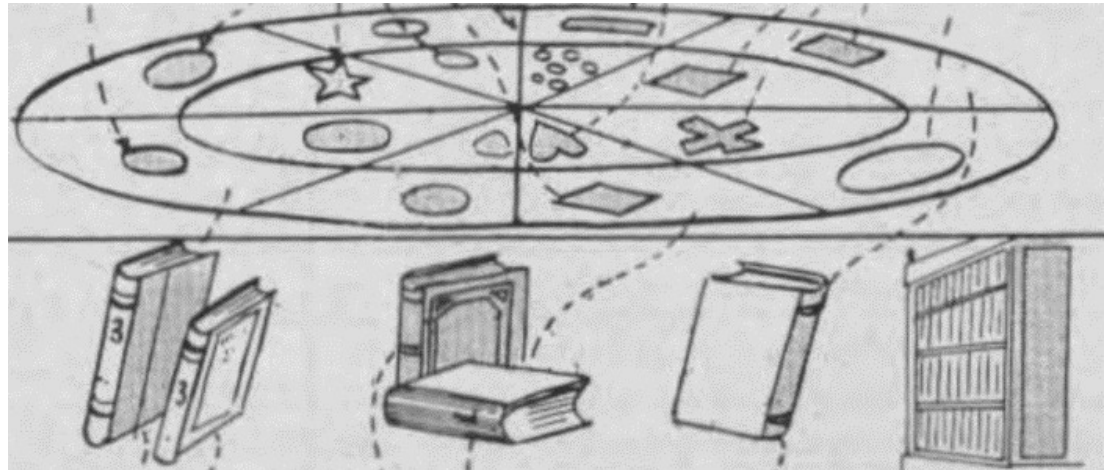
```
@prefix : <http://example.org/doc.ttl#> .  
:Berlin :population [ :inhabitants 3460725 ;  
                      :year 2010 ] .
```

# Agenda

1. Semantics on the Web
2. The Resource Description Framework (RDF)
3. Syntaxes for RDF Graphs
4. Turtle Syntax for RDF Graphs
- 5. Publishing and Accessing Linked Data**

# Referencing a Resource, Dereferencing a URI

- Referencing a resource is easy: just write the URI
- But what about dereferencing?
- How do you get the referenced resource?
- What do you get?



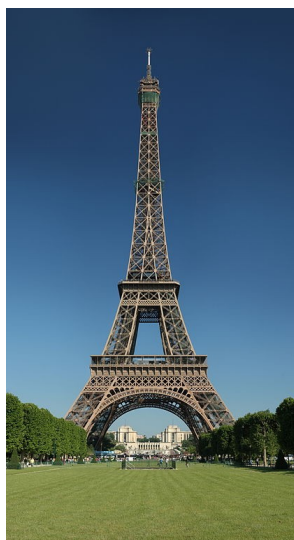
# Referencing a Resource, Dereferencing a URI<sup>1</sup>

- The act of retrieving a representation of a resource identified by a URI is known as **dereferencing** that URI
- Applications, such as browsers, render the retrieved representation for the user
- Most web users do not distinguish between a resource and the rendered representation they receive by accessing it
- **Information resources** associated with a **resource** need to have their own URIs
- They are themselves distinct resources and provide representations

<sup>1</sup> <https://www.w3.org/2001/tag/doc/httpRange-14/2007-05-31/HttpRange-14>

# Referencing a Resource, Dereferencing a URI

- It is important to differentiate between a resource and an informational document **about** that resource
- As you cannot retrieve the resource via your browser, a representation is needed



1



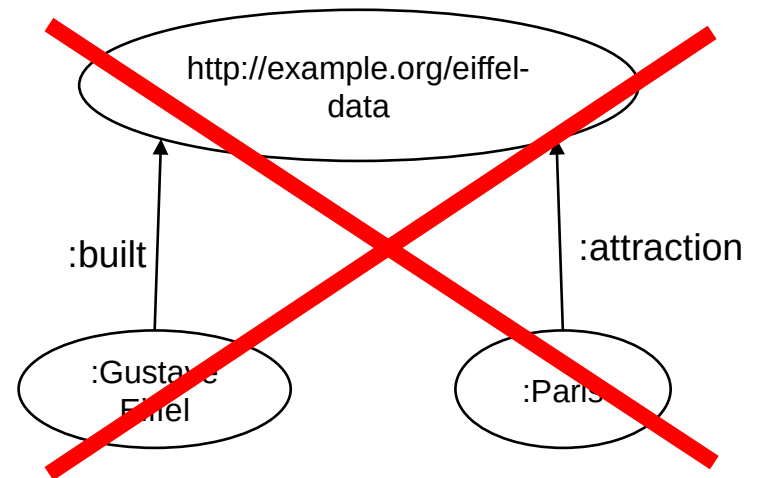
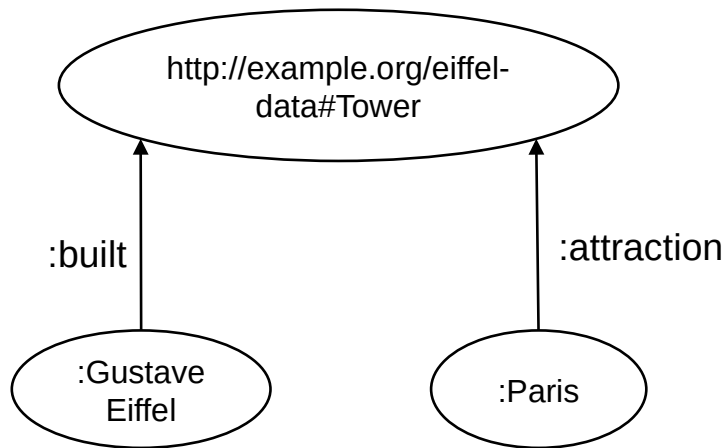
<http://example.org/eiffel-data#Tower>

<http://example.org/eiffel-data>

<sup>1</sup> Talking about differentiation: this is not the Eiffel Tower. It is a picture of the Eiffel Tower and the picture's URI is [https://upload.wikimedia.org/wikipedia/commons/thumb/8/85/Tour\\_Eiffel\\_Wikimedia\\_Commons\\_\(cropped\).jpg/360px-Tour\\_Eiffel\\_Wikimedia\\_Commons\\_\(cropped\).jpg](https://upload.wikimedia.org/wikipedia/commons/thumb/8/85/Tour_Eiffel_Wikimedia_Commons_(cropped).jpg/360px-Tour_Eiffel_Wikimedia_Commons_(cropped).jpg)

# Referencing a Resource, Dereferencing a URI

- As the document about the resource is also a resource itself, it needs its own URI (Information Resource)
- To reference the „Eiffel Tower“, only the URI of the “**resource**” is used:



# Referencing a Resource, Dereferencing a URI

- A user that wants information about a given resource might not know the URI of the describing document (the associated information resource)
- In the Semantic Web, two possibilities for providing the information resource of a resource are used: “hash URIs” and “slash URIs”

# Resource vs. Information Resource

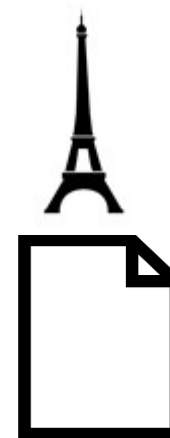
## Hash URIs

- Retrieving the document's URI by stripping off the hash of a hash URI

<http://example.org/eiffel-data#Tower>



<http://example.org/eiffel-data>

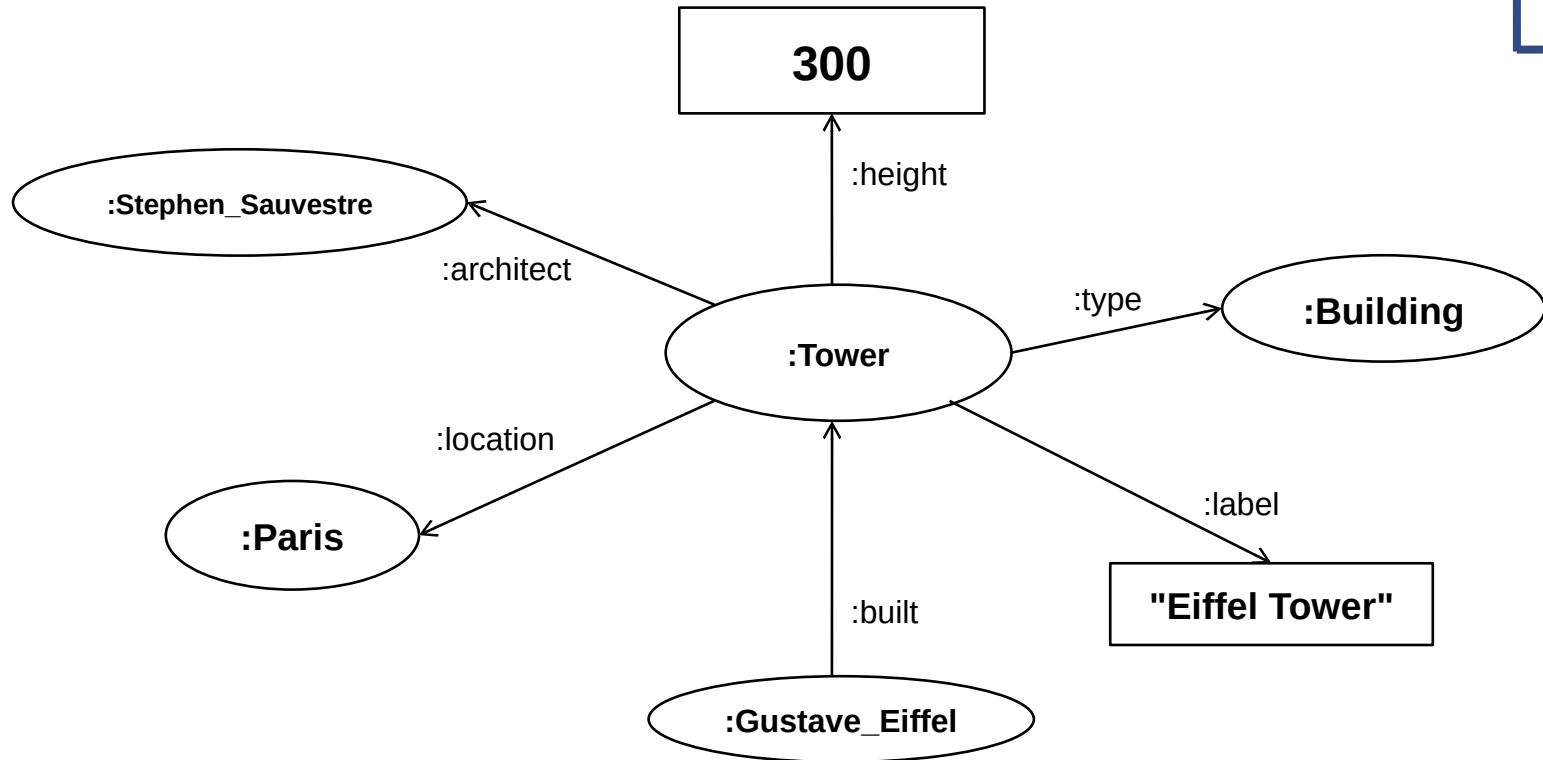




# Eiffel Tower Hash URI Example

<http://example.org/eiffel-data>

@prefix : <#> .



The URI of Eiffel Tower is <http://example.org/eiffel-data#Tower>, the URI of the information resource (associated document) is <http://example.org/eiffel-data>

# Resource vs. Information Resource

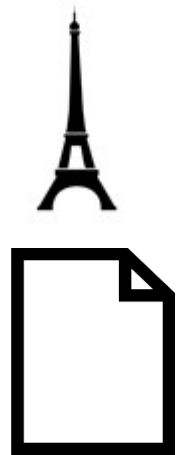
## Slash URIs

- Retrieving the document's URI by an automated HTTP redirect (303)

[http://dbpedia.org/resource/Eiffel\\_Tower](http://dbpedia.org/resource/Eiffel_Tower)



[http://dbpedia.org/data/Eiffel\\_Tower](http://dbpedia.org/data/Eiffel_Tower)

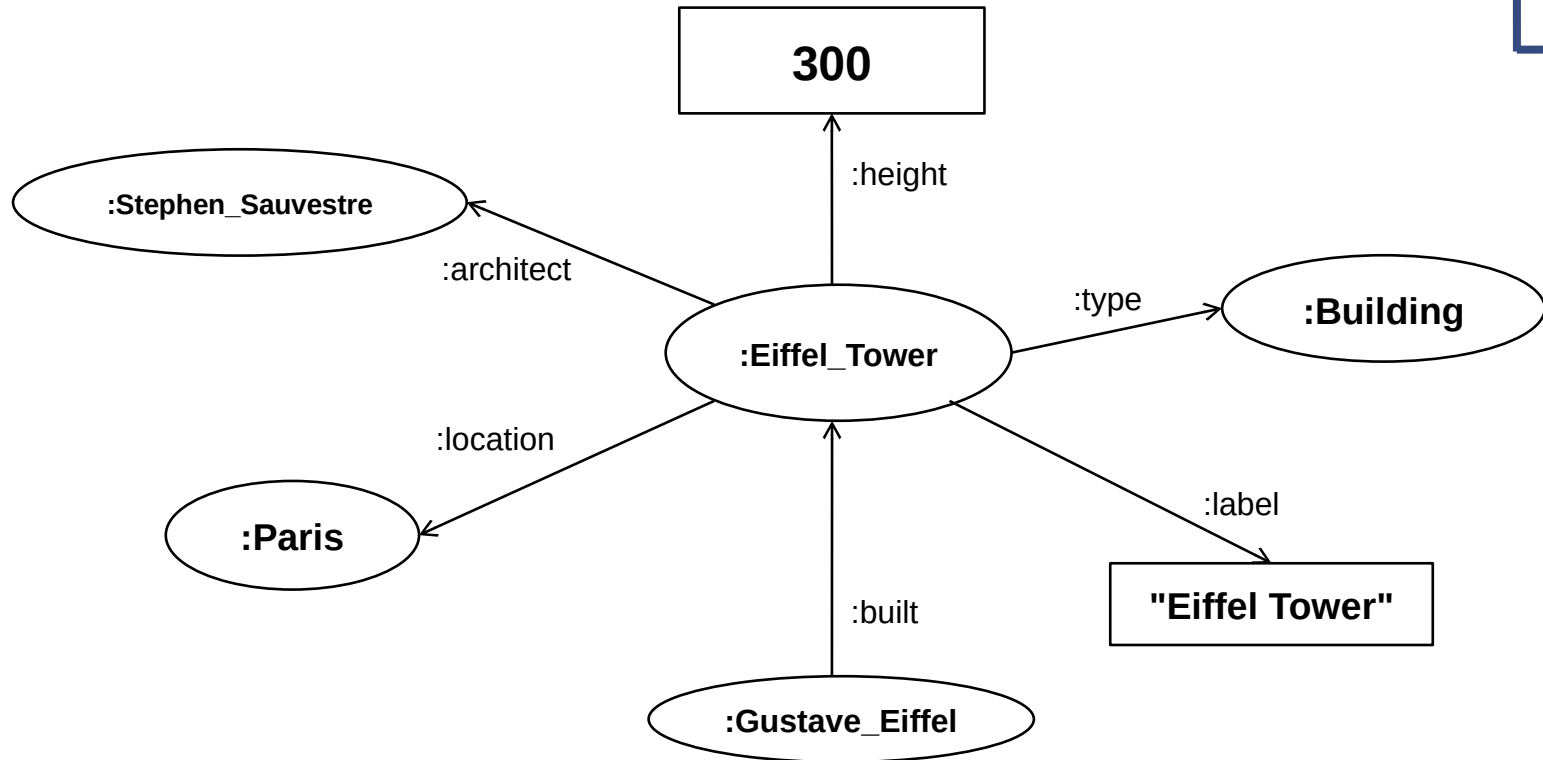


- This method also allows for content negotiation: the user can specify his preferred content format in the initial HTTP request (directing to the resource) and the server redirects him to the requested document

# Eiffel Tower Slash URI Example

[http://dbpedia.org/data/Eiffel\\_Tower](http://dbpedia.org/data/Eiffel_Tower)

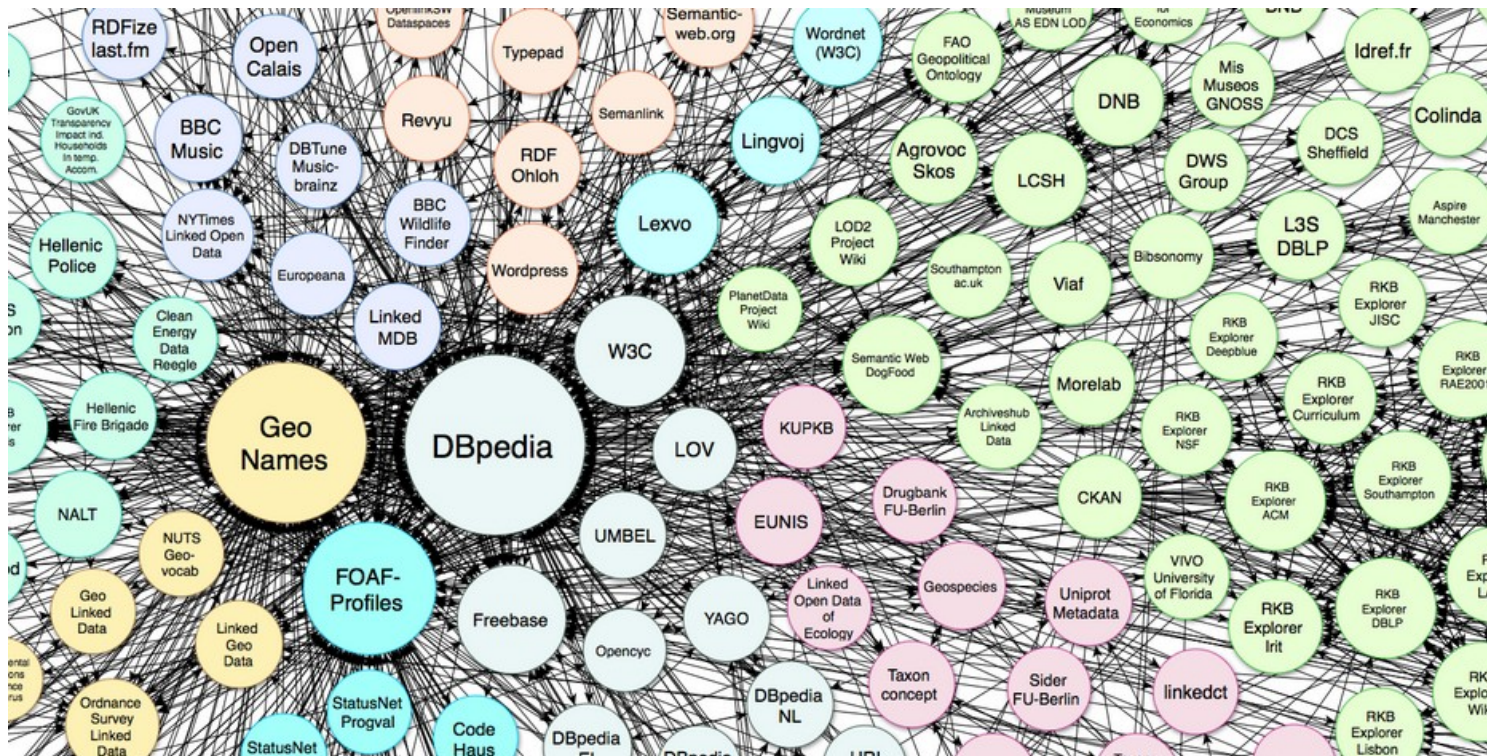
@prefix : <<http://dbpedia.org/resource/>> .



The URI of Eiffel Tower is [http://dbpedia.org/resource/Eiffel\\_Tower](http://dbpedia.org/resource/Eiffel_Tower), the URI of the information resource is [http://dbpedia.org/data/Eiffel\\_Tower](http://dbpedia.org/data/Eiffel_Tower).

# Linking Open Data Cloud

- The Linking Open Data cloud diagram contains hundreds of datasets from many domains<sup>1</sup>
- This is a good starting point to get an idea of what data is available



2

<sup>1</sup> <http://lod-cloud.net/>

<sup>2</sup> [http://lod-cloud.net/versions/2014-08-30/lod-cloud\\_colored.png](http://lod-cloud.net/versions/2014-08-30/lod-cloud_colored.png)

# DBpedia



- DBpedia contains data extracted from Wikipedia infoboxes via extraction scripts<sup>2</sup>
- The resulting RDF dataset is made available as Linked Data
- DBpedia is a community project, and the extraction scripts and the data are available under open licenses<sup>3</sup>
- DBpedia is a good starting point for exploring Linked Data, as DBpedia links to many other Linked Data sites


<sup>1</sup> <https://en.wikipedia.org/wiki/DBpedia>

<sup>2</sup> <http://dbpedia.org/>

<sup>3</sup> Christian Bizer, Jens Lehmann, Georgi Kobilarov, Sören Auer, Christian Becker, Richard Cyganiak, and Sebastian Hellmann. Dbpedia – a crystallization point for the web of data. Web Semant., 7 ( 3 ): 154 – 165 , September 2009 . ISSN 1570 - 8268 . doi : 10 . 1016 /j.websem. 2009 . 07 . 002 . <http://dx.doi.org/10.1016/j.websem.2009.07.002>

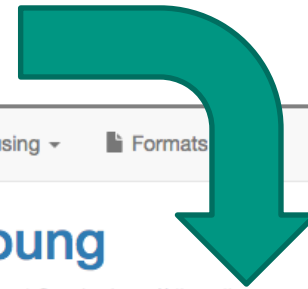
# DBpedia - Example



## Editing Neil Young

**B** *I*  [Advanced](#) [Special characters](#) [Help](#) [Cite](#)

```
{{About|the musician|other persons named Neil Young|Neil Young (disambiguation)|
{{Use Canadian English|date=December 2015}}
{{Use mdy dates|date=December 2015}}
{{Infobox musical artist|<!-- See Wikipedia:WikiProject Musicians
| name
      = Neil Young
| image
      = Neil Young - 1983.jpg
| caption
      = Young in 1983
| image_size
      =
| background
      = solo_singer
| birth_name
      = Neil Percival Young
| alias
      = Bernard Shakey,<ref name="
Canuck, Joe Yankee,<ref name="rose">'[[Charlie Rose]]'
6222184045890256947&ei=eUufS62zDqSIqQP176GXDA&q=ne
Pinecone Young<ref>{{cite web |url=http://www.youtube.com/watch?v=
Young |publisher=YouTube |date=September 14, 2010
| birth_date
      = {{birth date and age|mf=yes|year=1945}}
```

Data extraction from  
Wikipedia to DBpedia  
(find examples yourself!)



 [Browse using](#)  [Formats](#)

## About: Neil Young

An Entity of Type : [person](#), from Named Graph : <http://dbpedia.org>, within Data Space : [dbpedia.org](#)

Neil Percival Young, OC OM (born November 12, 1945) is a Canadian singer-songwriter and screenwriter.

Property	Value
<a href="#">dbo:activeYearsStartYear</a>	<ul style="list-style-type: none"><li>1960-01-01 (xsd:date)</li></ul>
<a href="#">dbo:alias</a>	<ul style="list-style-type: none"><li>Bernard Shakey, Phil Perspective, Shakey Deal, Clyde Coil, Ol' Neil</li></ul>
<a href="#">dbo:associatedBand</a>	<ul style="list-style-type: none"><li><a href="#">dbr:Pearl_Jam</a></li><li><a href="#">dbr:The_Stills-Young_Band</a></li><li><a href="#">dbr:Crosby,_Stills,_Nash_&amp;_Young</a></li><li><a href="#">dbr:Elton_John</a></li></ul>

<sup>1</sup> [https://en.wikipedia.org/wiki/Neil\\_Young](https://en.wikipedia.org/wiki/Neil_Young)

<sup>2</sup> [http://dbpedia.org/page/Neil\\_Young](http://dbpedia.org/page/Neil_Young)



# Wikidata

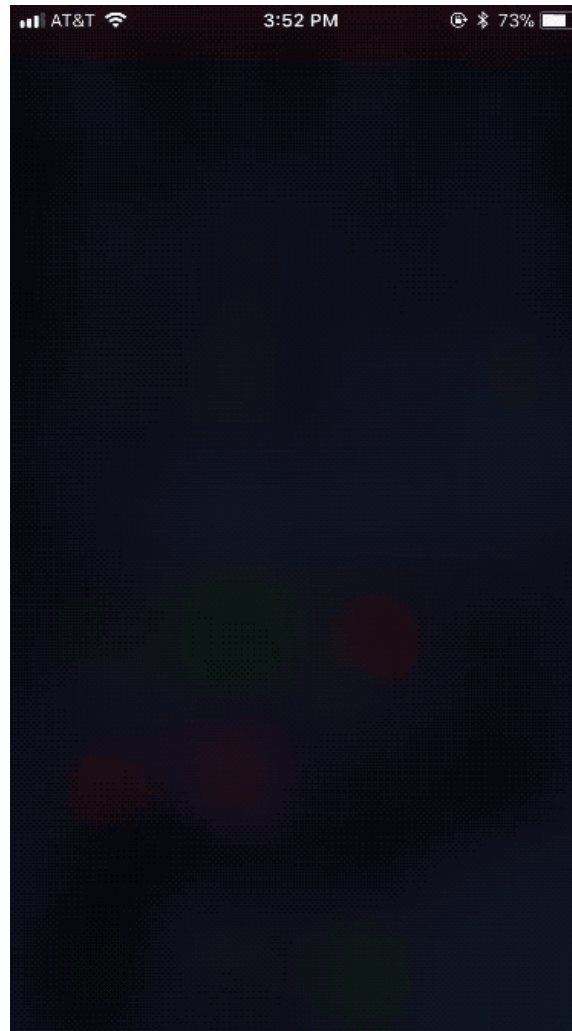


- Wikidata is Wikimedia foundation project, started by the people who created Semantic MediaWiki<sup>2</sup>
- The goal is to provide a common knowledge base for generating fact boxes in different language versions of Wikipedia articles
  - E.g., information about Bulgaria, identified via <http://www.wikidata.org/entity/Q219>, in Turtle file <https://www.wikidata.org/wiki/Special:EntityData/Q219.ttl>
- The data is available under open licenses and is used by Google, Amazon, Apple.

<sup>1</sup> <http://wikidata.org/>

<sup>2</sup> Denny Vrandečić and Markus Krötzsch. Wikidata: A free collaborative knowledgebase. Communications of the ACM, 57 ( 10 ): 78 – 85 , September 2014 . ISSN 0001 - 0782 . <http://doi.acm.org/10.1145/2629489>

# The National Anthem of Bulgaria is...



[https://www.reddit.com/r/softwaregore/comments/74epbw/siri\\_thinks\\_the\\_national\\_anthem\\_of\\_bulgaria\\_is/](https://www.reddit.com/r/softwaregore/comments/74epbw/siri_thinks_the_national_anthem_of_bulgaria_is/)



# Wikidata Change History of Entity “Bulgaria”



Main page  
Community portal  
Project chat  
Create a new item  
Create a new lexeme  
Recent changes  
Random item  
Query Service  
Nearby  
Help  
Donate

Tools  
What links here  
Related changes  
Special pages  
Permanent link  
Page information  
Concept URI  
Cite this page

English Not logged in Talk Contributions Create account Log in

Item Discussion

Read

View history

Search Wikidata



## Difference between revisions of "Bulgaria" (Q219)



Browse history interactively



Revision as of 15:33, 21 September 2017

78.83.94.41 (talk)

(Changed claim: *anthem* (P85): *Despacito* (Q28572509))

(Tag: new editor changing statement)

← Older edit

Revision as of 00:10, 5 October 2017

Chetankale1978 (talk | contribs)

(Changed claim: *population* (P1082): 7,101,859)

Newer edit →

(5 intermediate revisions by 4 users not shown)

property / **anthem**

- **Despacito**

property / **anthem**

+ **Mila Rodino**

property / **population**

**7,101,859**

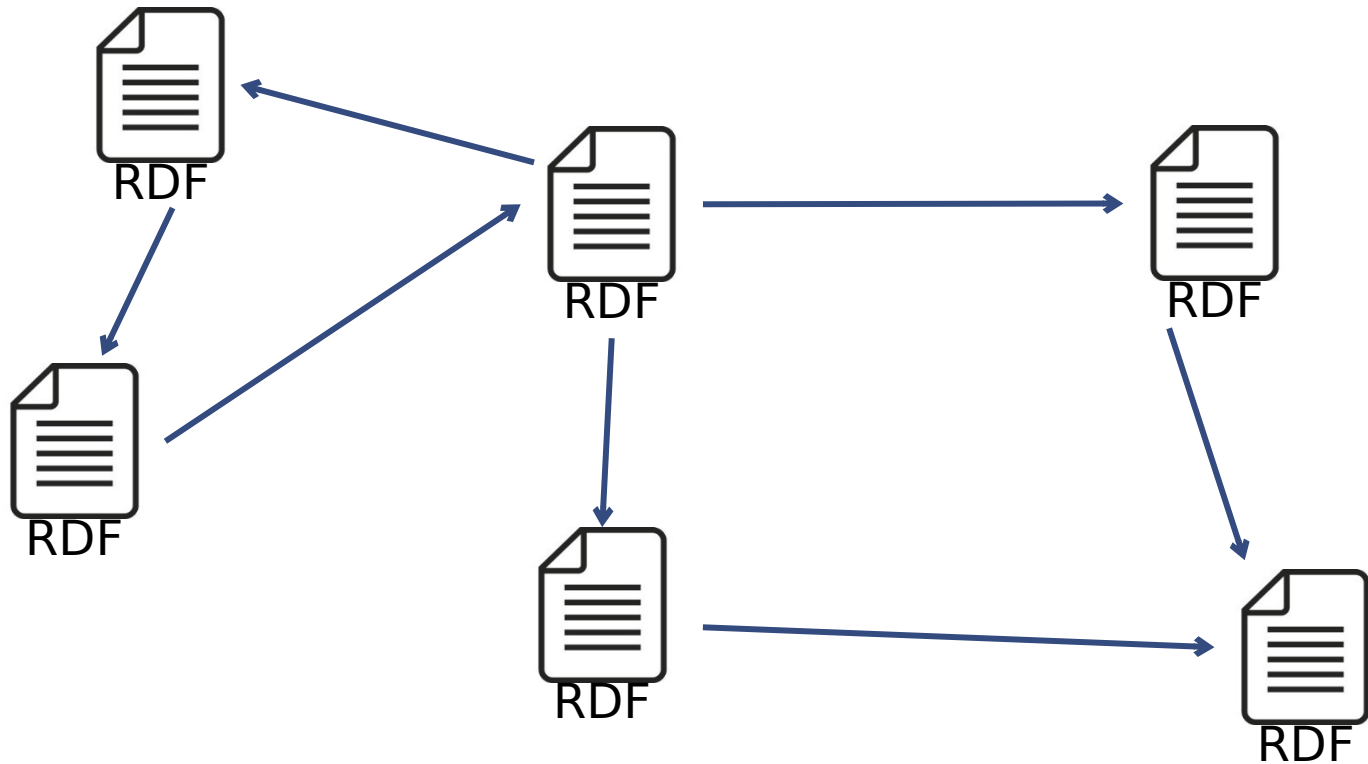
+  
**Amount** 7,101,859  
**Unit** 1

property / **population**: 7,101,859 / rank

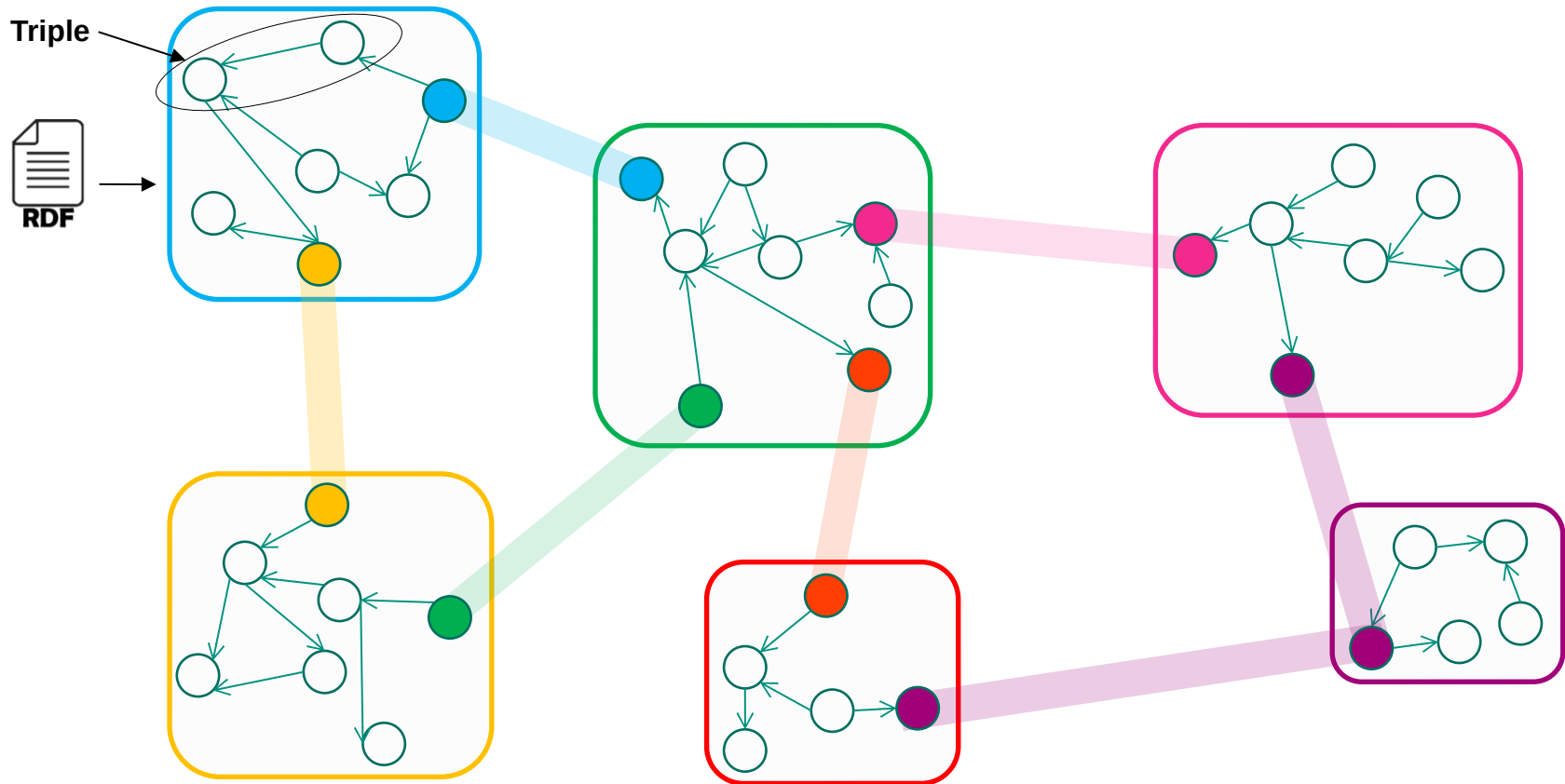
+ **Normal rank**

<https://www.wikidata.org/w/index.php?title=Q219&type=revision&diff=572462872&oldid=563217178>

# A Web of RDF Documents...



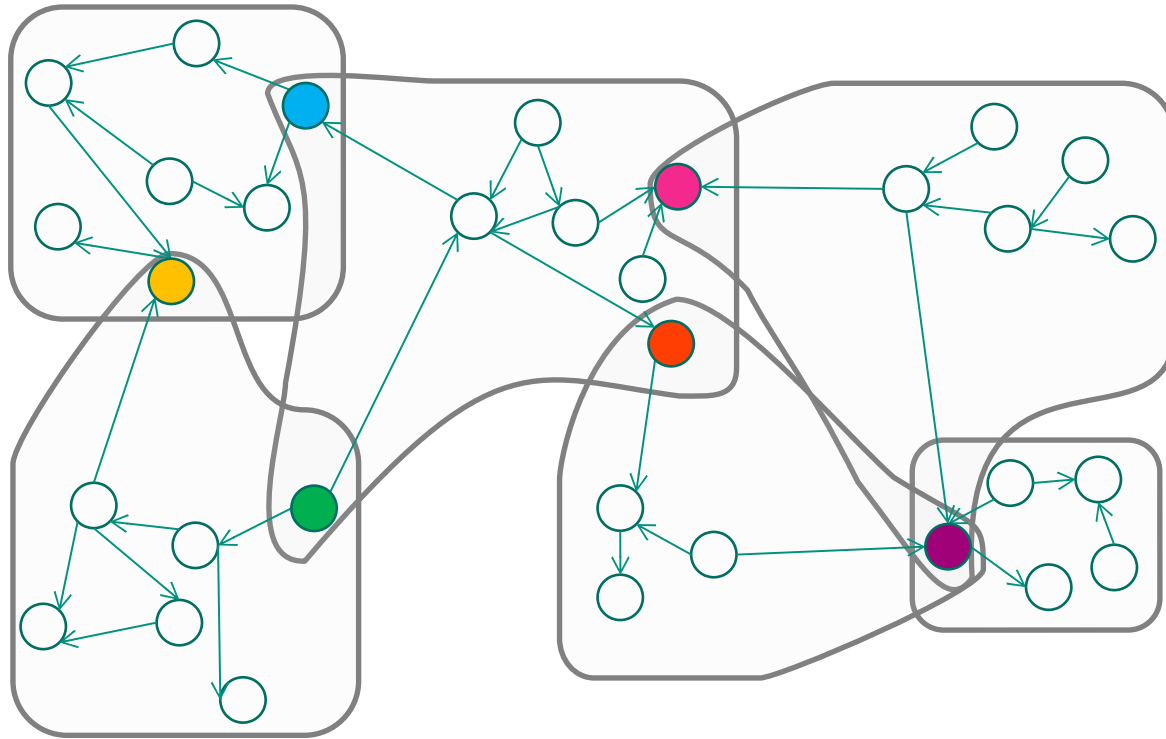
# ...Using URIs...



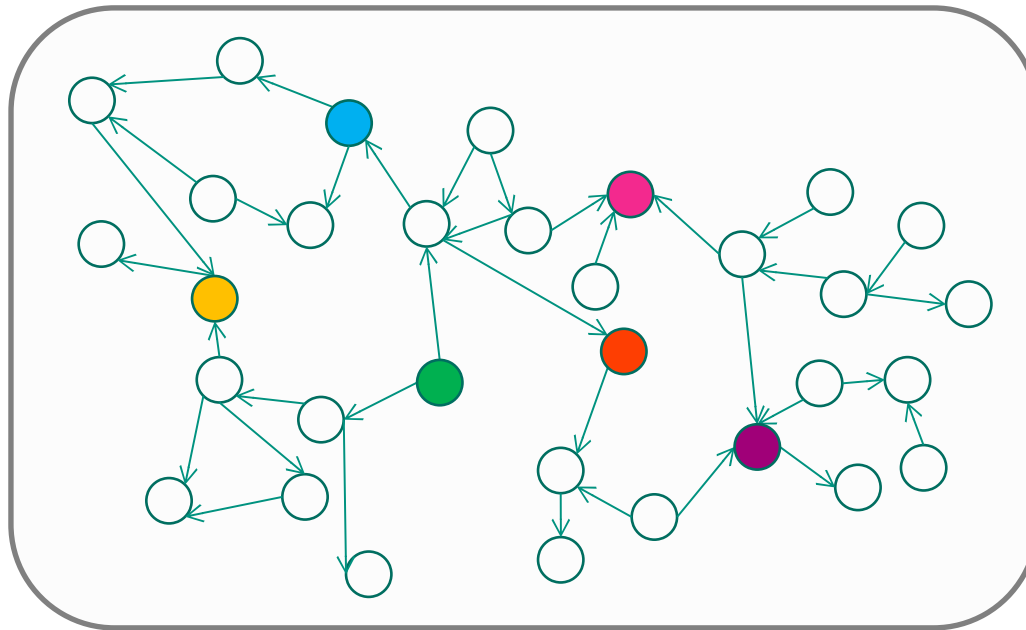
Circles with same color indicates identical resource

Documents and circles in the same color indicate correspondence between resource and information resource

## ...Can Actually Form...



## ...a Web of Data



- Each node is one resource, meaningfully linked to other resources  $\square$   
Linked Data

# Learning Goals

- G 2.1: Distinguish between URIs for things and URIs for documents (“information resources”); tell information resources apart from other resources based on the URI of the resource.
- G 2.2: Transform graphical representations of RDF graphs to Turtle documents (including abbreviations).
- G 2.3: Transform Turtle documents (including abbreviations and the resolution of relative references) to simple triple form (N-Triples), and vice versa.
- G 2.4: Explain the mechanisms behind hash URIs and slash URIs for separating URIs denoting other resources from URIs denoting information resources.
- G 2.5: Decide whether a given RDF document adheres to the four Linked Data principles, and explain why (or why not).