Knowledge Graphs
Organization
Hello

- Prof. Dr. Heiko Paulheim
  - Chair for Data Science

- Research Interests:
  - Knowledge Graphs on the Web and their Applications
  - Data Quality and Data Cleaning on Knowledge Graphs
  - Using Knowledge Graphs in Data Mining
  - Societal Impact of Artificial Intelligence

- Room: B6 26, B0.22

- Consultation: Tuesdays 9-10
  - Please make an appointment with Bianca Lermer upfront

- Heiko will teach the lectures
Hello

- M.Sc. Sven Hertling
- Graduate Research Associate
- Research Interests:
  - Semantic Technologies / Semantic Web
  - Linked Data
  - Knowledge Graphs
- eMail: sven@informatik.uni-mannheim.de
- Sven will teach the exercises and co-supervise the projects.
Introduction and Course Outline

- Administration
- Introduction to Knowledge Graphs
- History of Knowledge Graphs
  - Vision of the Semantic Web
  - Building blocks of the Semantic Web
  - Technical foundations
Course Organization

• Lecture
  – Knowledge Graph standards and languages
  – Using public knowledge graphs
  – Creating knowledge graphs

• Exercise
  – Understand knowledge graphs and their principles, play with real data

• Project Work
  – teams of 3-4 students build a Knowledge Graph application
  – teams may choose their own data sets and tasks
    (in addition, we will propose some pointers for ideas)
  – write summary about project, present project results
  – **not graded, but mandatory**

• Final exam
  – **final grades are only based on written exam**
Course Organization

• Registration
  – you have registered via Portal2
  – you should have access to ILIAS
  – the course is fully booked with a waiting list
    • if you decide not to attend, please deregister in Portal2

• Important: course replacement
  – This course replaces IE 650 Semantic Web Technologies
  – You cannot get credits for both courses
<table>
<thead>
<tr>
<th>Week</th>
<th>Tuesday</th>
<th>Friday</th>
</tr>
</thead>
<tbody>
<tr>
<td>19.09.2022</td>
<td>Lecture: RDF</td>
<td>Exercise: RDF</td>
</tr>
<tr>
<td>26.09.2022</td>
<td>Lecture: RDFS</td>
<td>Exercise: RDFS</td>
</tr>
<tr>
<td>03.10.2022</td>
<td>Lecture: Linked Data, Semantic Web Programming</td>
<td>Exercise: Linked Data, Semantic Web Programming</td>
</tr>
<tr>
<td>10.10.2022</td>
<td>Lecture: SPARQL and other Query Paradigms</td>
<td>Exercise: SPARQL, Kick off Group Projects</td>
</tr>
<tr>
<td>17.10.2022</td>
<td>Lecture: Public Knowledge Graphs</td>
<td>Exercise: Public Knowledge Graphs</td>
</tr>
<tr>
<td>24.10.2022</td>
<td>Lecture: Labeled Property Graphs</td>
<td>Exercise: Labeled Property Graphs</td>
</tr>
<tr>
<td>31.12.2022</td>
<td>Public holiday</td>
<td>No exercise</td>
</tr>
<tr>
<td>07.11.2022</td>
<td>Lecture: OWL Part 1</td>
<td>Exercise: OWL Part 1</td>
</tr>
<tr>
<td>21.11.2022</td>
<td>Lecture: Knowledge Modeling</td>
<td>Exercise: Knowledge Modeling</td>
</tr>
<tr>
<td>28.11.2022</td>
<td>Lecture: Data Quality and Interlinking</td>
<td>Exercise: Data Quality and Interlinking</td>
</tr>
<tr>
<td>05.12.2022</td>
<td>Project Presentations</td>
<td></td>
</tr>
</tbody>
</table>
Deadlines

• Submission of project work proposal
  – Sunday, October 16th 23:59

• Submission of final project work report
  – Friday, December 9th, 23:59
Course Organization

- Lecture Webpage: Slides, Announcements, Web Links
  - hint: look at version tags of slides!
- Additional Material
  - ILIAS eLearning System, https://ilias.uni-mannheim.de/
- Time and Location
  - Lecture: Tuesday, 3.30 – 5.00, Room B6 30-32, E-F, room 209
  - Exercise: Friday, 12.00 - 13.30, Room B6 26, A1.04
Further Reading and Software

- Follow the links on the website
  - Most material is available online

- Programming environment
  - JENA framework (Java)
  - RDFlib (Python)

- Knowledge graph environment
  - Neo4j

- Ontology engineering environment
  - Protégé
    - [http://protege.stanford.edu/](http://protege.stanford.edu/)
Warning

• This lecture contains
  – cartoons
  – Java and Python code
  – some digressions to philosophy

• Having said that:
  – have fun! :-)
Questions?
Knowledge Graphs
Introduction
The Birth of a Buzzword

Introducing the Knowledge Graph: things, not strings

Sources: Google
Idea of the Google Knowledge Graph

• Web search in the pre-knowledge graph age
  – Documents
  – Keywords (not: disambiguated entities)
Idea of the Google Knowledge Graph

• Web search in the knowledge graph age
  – Backed by structured information
  – All entities are disambiguated

• Linked to other services
  – Ratings and reviews
  – Map information
  – External services (e.g., booking)
  – …
An Example for a Knowledge Graph

https://yashuseth.wordpress.com/2019/10/08/introduction-question-answering-knowledge-graphs-kgqa/
From Google to the World

- Documented list of companies using knowledge graphs
  - Courtesy of Frank van Harmelen, VU Amsterdam
Knowledge Graph Definitions

• Knowledge Graphs are a fairly new technology
  – Hence, there are few universally acclaimed definitions

• Some example definitions from the literature:
  – Knowledge graphs could be envisaged as a network of all kind things which are relevant to a specific domain or to an organization. They are not limited to abstract concepts and relations but can also contain instances of things like documents and datasets. (Blumauer, 2014)
  – Knowledge graphs are large networks of entities, their semantic types, properties, and relationships between entities. (Kroetsch and Weikum, 2016)
Knowledge Graph Definitions (ctd.)

• [...] systems exist, [...] which use a variety of techniques to extract new knowledge, in the form of facts, from the web. These facts are interrelated, and hence, recently this extracted knowledge has been referred to as a knowledge graph. (Pujara et al., 2013)

• A Knowledge Graph (1) mainly describes instances and their relations in a graph, (2) defines possible classes and relations in a schema or ontology, (3) allows for interlinking arbitrary entities with each other, and (4) covers various domains. (Paulheim, 2017)
Knowledge Graph Definitions (ctd.)

• Common ground so far:
  – There are entities and relations that are connected and form a graph
  – There is a set of entity and relation types
    • those are often referred to as a schema or ontology
    • we will get back to this
Rewinding the Time Machine

- Google claim “things, not strings”
  - Entities instead of words in documents
  - Relations between entities explicitly modeled
  - Accessible to humans and machines
    - think: computers cannot read text
The Vision of the Semantic Web (2001)

• 2001 article by Tim Berners-Lee, Jim Hendler, and Ora Lassila:

„The Web is the killer app of the Internet. The Semantic Web is another killer app of that magnitude.“

## Web vs. Internet?

<table>
<thead>
<tr>
<th>OSI Model</th>
<th>TCP/IP Model (DoD Model)</th>
<th>TCP/IP – Internet Protocol Suite</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Application</td>
<td>Telnet, SMTP, POP3, FTP, NNTP, HTTP, SNMP, DNS, SSH, ...</td>
</tr>
<tr>
<td>Presentation</td>
<td>Transport</td>
<td>TCP, UDP</td>
</tr>
<tr>
<td>Session</td>
<td>Internet</td>
<td>IP, ICMP, ARP, DHCP</td>
</tr>
<tr>
<td>Transport</td>
<td>Network Access</td>
<td>Ethernet, PPP, ADSL</td>
</tr>
<tr>
<td>Network</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Link</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physical</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The “Classic” Web

• Uses HTTP protocol and URLs
• HTML as a markup language
  – plus CSS, JavaScript, …
  – plus a few other, more or less standardized formats (GIF, JPEG, Flash, …)
• Browser as a universal client

• Information is accessible to humans, but not to machines
The “Classic” Web

- Hypertext: linked documents

The World Wide Web

The World Wide Web was Established in the 90s by Tim Berners-Lee at CERN.

Tim Berners-Lee

Tim Berners-Lee (born 1955) is one of the inventors of the World Wide Web.

CERN

The CERN is a European research center, located close to Geneva.
The “Classic” Web

In the eyes of a human

<html>
...  
<b>Dr. Mark Smith</b><i>Physician</i>
Main St. 14  
Smalltown  
Mon-Fri 9-11 am  
Wed 3-6 pm  
...
</html>

in the eyes of a computer

Dr. Mark Smith  
Physician  
Main St. 14  
Smalltown  
Mon-Fri 9-11 am  
Wed 3-6 pm

Print in bold: „hmf298hmhuds“  
Print in italics: „mj2i9ji0“  
Print normal: „fdsah  
02hfadsh0um2m0adsmf0i0hm  
asdfjköfdsa298ndsfmi032mio  
lk2mjpoimji0fdpmaj0mjm“
Searching for Information on the Web

Full text search by keywords (e.g., Google):

• „Mark Smith“
• „Physician in Smalltown“
• „Doctor in Smalltown“
• „Doctor in Smalltown with opening hours on Wednesday afternoon“
• „Somebody in Smalltown who can fix a broken leg“

→ “classic” Web is too inflexible for useful search
→ hard to use for intelligent agents

<html>
...<br/>
<b>Dr. Mark Smith</b><br/>
<i>Physician</i><br/>
Main St. 14<br/>
Smalltown<br/>
Mon-Fri 9-11 am<br/>
Wed 3-6 pm<br/>
...</html>
Problems of the “Classic” Web

• Finding information
  – Keyword based search instead of natural language questions
  – Different natural languages
  – Synonyms, homonyms and polysemous words
  – Ambiguity of natural language

• Processing information
  – Formats and encodings

• Making use of information
  – Distributed across pages
  – e.g., a book's author on the publishers site, address on his/her personal page

Homonyms and Polysemous Words
Bullet points:

- **Bush Era Law** Could Get You 20 Years in Prison For Clearing Your Browser History
Example: Wolfram Alpha

Multiple interpretations of “Mannheim” and “Karlsruhe”

Multiple interpretations of “distance”
Example: Wolfram Alpha
Solutions

Lectures:
Web Mining, Information Extraction

Extract information from the Web

Lecture:
Knowledge Graphs

Create machine interpretable information

WWW
Semantic Web Vision

• Provide information in machine interpretable form
• Make (semantic) links between (data) documents usable
• Facilitate useful (!) complex queries
• Allow logical reasoning
(Enterprise) Knowledge Graph Vision

- Integrate data from different sources
- Make connections between entities in those sources
- Facilitate cross data source queries
- Overcome data silos

Semantic Web – Architecture

here be dragons...

Knowledge Graph Technologies (This lecture)

Technical Foundations

User Interface and Applications

Trust

Proof

Unifying Logic

Query: SPARQL

Ontology: OWL

Rules: RIF

Schema: RDF-S

Data Interchange: RDF

Data Interchange: XML

URI

Unicode

Berners-Lee (2009): Semantic Web and Linked Data
Data Interoperability with Knowledge Graphs

Berners-Lee (2009): Semantic Web and Linked Data
Syntactic Interoperability: Character Sets

- ASCII („American Standard Code for Information Interchange“) ISO 646 (1963), 127 characters, 95 of which are printable:
  
  `!"#$%&'()*)+,-./0123456789:;<=?>@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
  
  `abcdefghijklmnopqrstuvwxyz{|}~`

- Extension to 8 Bit: ISO 8859-1 to -16 (1998)
  - covers major European languages
  - most well known: 8859-1 („Latin-1“)

- The Web, however, speaks many more languages...
Syntactic Interoperability: Multilinguality

TOWER OF BABEL

HOW DARE YOU!!! FROM NOW ON YOU’LL TALK....

...UNICODE!

HE WAS NOT AMUSED

Syntactic Interoperability: Unicode

- ISO 10646
  - first version 1991 (Europe, Near East, India)
  - Unicode 14.0 (September 2021)
  - defines ~144,000 characters
  - covers even very exotic languages
  - Plus: currency symbols, emojis, sign language, music notation...

Klingon is still missing!!!
Syntactic Interoperability: Unicode

Source: Wikimedia Commons
Information Representation in XML

XML (eXtensible Markup Language)

- A W3C standard since 1998
- Universal format for data exchange and integration

```xml
<physician>
    <name>Dr. Mark Smith</name>
    <address>
        <street>Main St.</street>
        <number>14</number>
        <city>Smalltown</city>
    </address>
    <telephone>
        <number>+44 123 456789</number>
    </telephone>
    <hours>
        <monday>9-11 am</monday>
        <tuesday>9-11 am</tuesday>
        ...
    </hours>
</physician>
```
XML: Basic Concepts

• Tags (arbitrarily definable):
  – Form pairs:
    <physician> … </physician>
  – ...or empty element tags
    <young />

• Attributes:
  <physician location="Smalltown"> … </physician>

• Tags are nested (with exactly one root element):
  <physician>
   <address> … </address>
  </physician>
XML: Well-formed Documents

<title>
<physician>
  <name>Dr. Mark Smith</name>
  <address>
    <street>Main St.</street>
    <number>14</number>
    <city>Smalltown</city>
  </address>
  <telephone>
    <number>+44 123 456789</number>
  </telephone>
  <hours>
    <monday>9-11 am</monday>
    <tuesday>9-11 am</tuesday>
    ...
  </hours>
</physician>
</title>
HTML and XML

• HTML documents look like XML documents
  – ...but they are usually not well-formed!

  <p>Look at this! <img src="smiley.gif"/> <br/>

• XHTML: HTML as well-formed XML documents
• A W3C standard since 2000

  <p>Look at this! <img src="smiley.gif"/> <br/></p>
XPath: Accessing Information in XML

- Query language for XML
- A W3C standard since 1999 (Version 2.0: 2010)

/physician[name='Dr. Mark Smith']/telephone/number

```xml
<physician>
  <name>Dr. Mark Smith</name>
  <address>
    <street>Main St.</street>
    <number>14</number>
    <city>Smalltown</city>
  </address>
  <telephone>
    <number>+44 123 456789</number>
  </telephone>
  <hours>
    <monday>9-11 am</monday>
    <tuesday>9-11 am</tuesday>
    ...
  </hours>
</physician>
```
Namespaces in XML

- Elements with the same name can occur in different places
  - …but the contents and semantics may differ
- How can we tell them apart?

```
<physician>
  <name>Dr. Mark Smith</name>
  <address>
    <street>Main St.</street>
    <number>14</number>
    <city>Smalltown</city>
  </address>
  <telephone>
    <number>+44 123 456789</number>
  </telephone>
  <hours>
    <monday>9-11 am</monday>
    <tuesday>9-11 am</tuesday>
    ...
  </hours>
</physician>
```
Namespaces in XML

- Namespace definition using prefixes (Notation: `prefix:name`)
- Each namespace itself is a URI
- Default namespaces may be defined

```xml
<physician xmlns="http://www.med.com/physician"
            xmlns:addr="http://www.med.com/addr">
  <name>Dr. Mark Smith</name>
  <addr:address>
    <addr:street>Main St.</addr:street>
    <addr:number>14</addr:number>
    <addr:city>Smalltown</addr:city>
  </addr:address>
  <telephone>
    <number>+44 123 456789</number>
  </telephone>
  <hours>
    <monday>9-11 am</monday>
    <tuesday>9-11 am</tuesday>
    ...
  </hours>
</physician>
```
XML: Document Type Definition (DTD)

• Defines valid elements for a class of XML documents
  – Names
  – allowed attributes
  – allowed nested child elements

• DTD is a part of the W3C's XML specification

• XML documents matching a DTD are called “valid”
XML: Document Type Definition (DTD)

```xml
<!DOCTYPE physician SYSTEM "physician.dtd">

<physician>
  <name>Dr. Mark Smith</name>
  <address>
    <street>Main St.</street>
    <number>14</number>
    <city>Smalltown</city>
  </address>
  <telephone>
    <number>+44 123 456789</number>
  </telephone>
  <hours>
    <monday>9-11 am</monday>
    <tuesday>9-11 am</tuesday>
    ...
  </hours>
</physician>
```
XML: Document Type Definition (DTD)

• Definition of child elements and their order

```xml
<!ELEMENT address (street, no, line*, zip, city, state?)>
```
- ?, + and * mark optional and possible multiple elements

• Definition of attribute lists

```xml
<!ATTLIST person title CDATA>
```
- Allowed modifiers: #REQUIRED, #FIXED, #IMPLIED, “...“
- Enumerating allowed values: (dr|prof)

• Definition of entities:

```xml
<!ENTITY sw "Semantic Web"> 
```
- May be used as shortcuts in the XML document: &sw;
XML Schema

- W3C-Standard (since 2004)
- XML schemas are XML files themselves

- More flexible than DTDs:
  - Minimum and maximum number of elements
  - Combinations of elements (either/or, combinations w/out fixed order, …)
  - Data types (Numbers, dates, …), own types may be defined
  - Support for namespaces
  - Possibility to create modular schemas
XML Schema

```xml
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="physician">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name" type="xs:string">
        </xs:element>
        <xs:element name="address">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="street" type="xs:string">
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

<physician xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="physician.xsd">
  <name>Dr. Mark Smith</name>
  <address>
    <street>Main St.</street>
    <number>14</number>
    <city>Smalltown</city>
  </address>
  <telephone>
    <number>+44 123 456789</number>
  </telephone>
  <hours>
    <monday>9-11 am</monday>
    <tuesday>9-11 am</tuesday>
  </hours>
</physician>
```
XML Schema – Modular Schemas

```xml
<xs:schema elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:addr="http://www.address.com/">

  <xs:import
      namespace="http://www.address.com/"
schemaLocation="address.xsd"/>

  <xs:element name="physician">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="name"
type="xs:string">
        </xs:element>
        <xs:element ref="addr:address" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```xml
<xs:element name="address">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="street"
type="xs:string">
      </xs:element>
      ...  
    </xs:sequence>
  </xs:complexType>
</xs:element>
</xs:schema>
```
Example: Modular Schemas in XHTML

```html
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:svg="http://www.w3.org/2000/svg">
<body>
<h1>SVG embedded inline in XHTML:</h1>
<svg:svg width="300px" height="200px">
  <svg:circle cx="150" cy="100" r="50" fill="#ff0000"/>
</svg:svg>
</body>
</html>
```

https://developer.mozilla.org/En/SVG:Namespaces_Crash_Course
So, what does a DTD/Schema Define?

- **Syntax** – σύνταξις ("together" + "order")
  - Which elements are there?
  - How are they arranged?
  - Which combinations are allowed?

- ...as opposed to: **Semantics** - σημαίνειν ("denote")
  - How to interpret the contents of an element?
  - What is their relation?
Syntax and Semantics: The Linguists' View

• Syntax: how are correct sentences formed?
  „This sentence no verb.“
  „The dreaming lamp give gives a freshly cut juices juice to the tire tired sink."

• Semantics: what does a word and sentence mean?

• Notes
  – syntactic correctness does not guarantee semantic interpretability
  – semantic interpretability does not require syntactic correctness (for humans)
Syntax and Semantics: The Linguists' View

Definition of knowledge noun from the Oxford Advanced Learner's Dictionary

knowledge noun

BrE /'nɒlɪdʒ/ NAmE /'nɑːlɪdʒ/

[uncountable, singular] the information, understanding and skills that you gain through education or experience

• practical/medical/scientific knowledge
• knowledge of/about something He has a wide knowledge of painting and music.
• There is a lack of knowledge about the tax system.

→ SEE RELATED ENTRIES: Teaching and learning

2 [uncountable] the state of knowing about a particular fact or situation

• She sent the letter without my knowledge.
• The film was made with the Prince's full knowledge and approval.
• She was impatient in the knowledge that time was limited.
• I went to sleep secure in the knowledge that I was not alone in the house.
• They could relax safe in the knowledge that they had the funding for the project.
• He denied all knowledge of the affair.

knowledge economy industry worker working with information rather than producing goods

• the emergence of consultancy as a knowledge industry
• the shift toward a knowledge economy
So, what does a DTD/Schema Define?

Employee catalog of the hospital

```xml
<physician>
  <name>Dr. Mark Smith</name>
  <address>
    <street>Main St.</street>
    <number>14</number>
    <city>Smalltown</city>
  </address>
  <telephone>
    <number>+44 123 456789</number>
  </telephone>
  <hours>
    <monday>9-11 am</monday>
    <tuesday>9-11 am</tuesday>
    ...
  </hours>
</physician>
```

(probably) the private address

Yellow Pages

```xml
<physician>
  <name>Dr. Mark Smith</name>
  <address>
    <street>Main St.</street>
    <number>14</number>
    <city>Smalltown</city>
  </address>
  <telephone>
    <number>+44 123 456789</number>
  </telephone>
  <hours>
    <monday>9-11 am</monday>
    <tuesday>9-11 am</tuesday>
    ...
  </hours>
</physician>
```

(probably) the work address
So, what does a DTD/Schema Define?

- XML Schema / DTD defines the syntax of an XML document, but no its semantics

- Tag names are not interpretable by machines
  - i.e., they do not ease the information retrieval process...
  - Semantics of the data is hidden – usually hard wired in the application

- The Semantic Web is meant as a remedy to that problem
  - *Semantic Web is/can do more than XML!*

```xml
<2nf3oiü*>  
  <34f0>Dr. Mark Smith</34f0>  
  <rmd4935r>  
    <e2m4>Main St.</e2m4>  
    <dur3>14</dur3>  
    <jfa34>Smalltown</jfa34>  
  </rmd4935r>  
  <d24r3fmö>  
    <deß5>+44 123 456789</deß5>  
  </d24r3fmö>  
  <vsfif>  
    <f02>9-11 am</f02>  
    <fj9>9-11 am</fj9>  
    ...  
  </vsfif>  
</2nf3oiü*>  
```
Uniform Resource Identifiers (URIs)

- “Things, not strings” requires identifiers for things
  - URIs: Proposed by Tim-Berners-Lee as „Universal Resource Identifier“ (IETF RFC 1630)

- Used for naming and finding resources on the Web

\[
\text{URI} = \text{scheme "":" } \text{hier-part} \text{[ "?" query ][ "#" fragment ]}
\]

\[
\text{http://example.com:8042/over/there?name=ferret#nose}
\]
URIs vs. URLs

• Uniform Resource Locators (IETF RFC 1738, 1994) are a *subset* of URIs

• URIs can refer to *arbitrary* things
• A URL refers to a resource on the Web

• Typical URL prefixes
  – http
  – ftp
  – mailto
  – telnet
  – file
  – ...


URLs on the Web

- Most common usage: Hyperlinks in HTML documents
- Links usually do not carry any meta information

Tim Berners-Lee

Tim Berners-Lee (born 1955) is one of the inventors of the World Wide Web.

Most common usage:

Hyperlinks in HTML documents

Links usually do not carry any meta information
Wrap Up

• Knowledge Graphs
  – Facilitate syntactic and semantic data interoperability

• Today, we have seen syntactic interoperability
  – Unicode: a character set for all languages
  – XML: a universal data exchange format
    • XPath
    • DTD
    • XML Schema
  – URIs
    • Unique identifiers for things (entities, resources, ...)
    • On the Web, URLs are dereferencable
Data Interoperability with Knowledge Graphs

Semantic Interoperability
- Query: SPARQL
- Ontology: OWL
- Data Interchange: RDF

Syntactic Interoperability
- Schema: RDF-S
- Rules: RIF
- Data Interchange: XML

Berners-Lee (2009): Semantic Web and Linked Data
Questions?