UNIVERSITÄT MANNHEIM

# Semantic Web Technologies
# Introduction

Heiko Paulheim

# What is the Semantic Web?

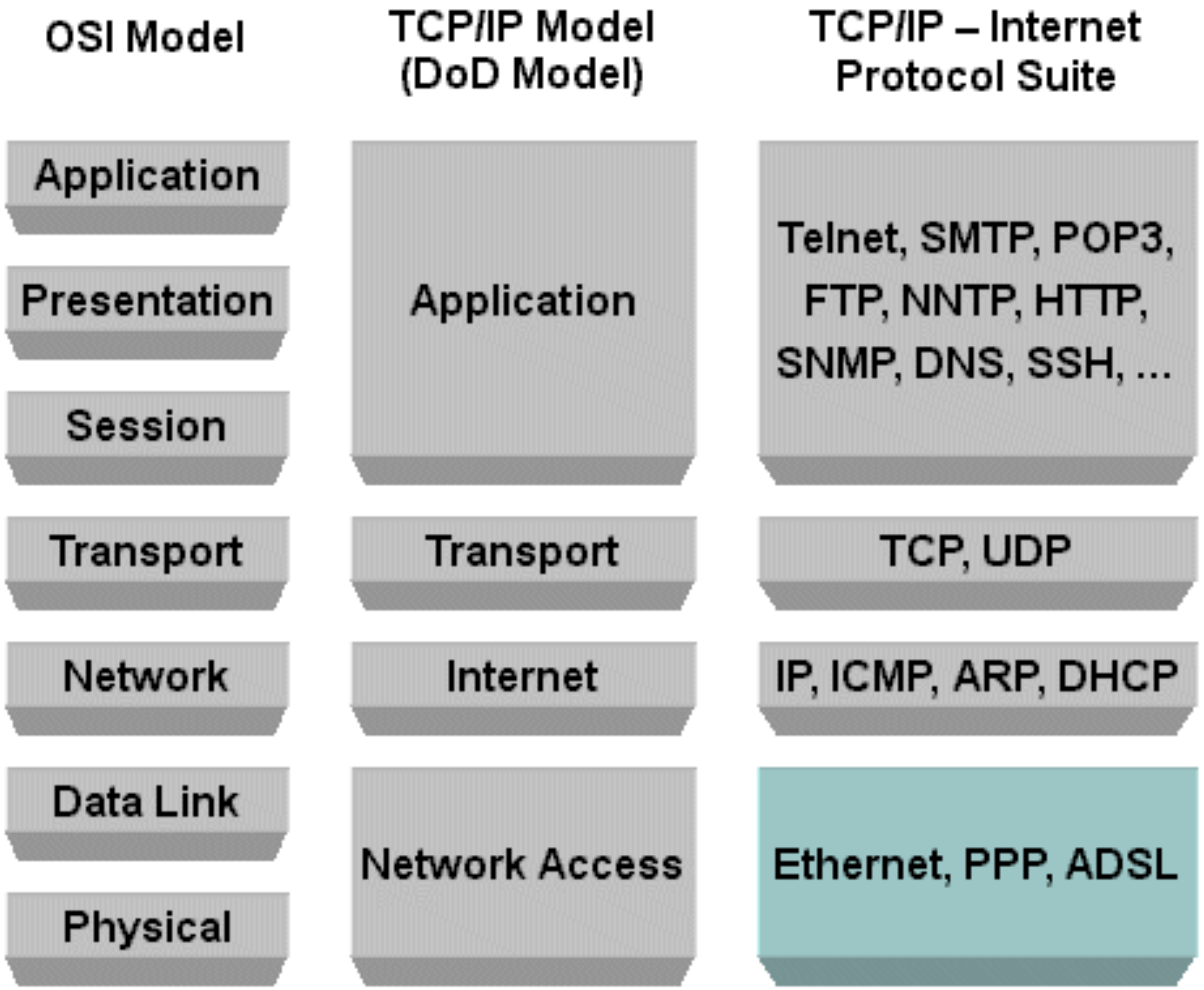
- 2001 article by Tim Berners-Lee, Jim Hendler, and Ora Lassila:

„The Web is the killer app of the Internet.

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


Berners-Lee et al. (2001): *The Semantic Web.* In: Scientific American, Mai 2001.

# Web vs. Internet?




Chin-Shiuh Shieh (2000): *TCP/IP - Internet Protocol Suite and Ethernet.* http://bit.kuas.edu.tw/~csshieh/teach/np/tcpip/index.html

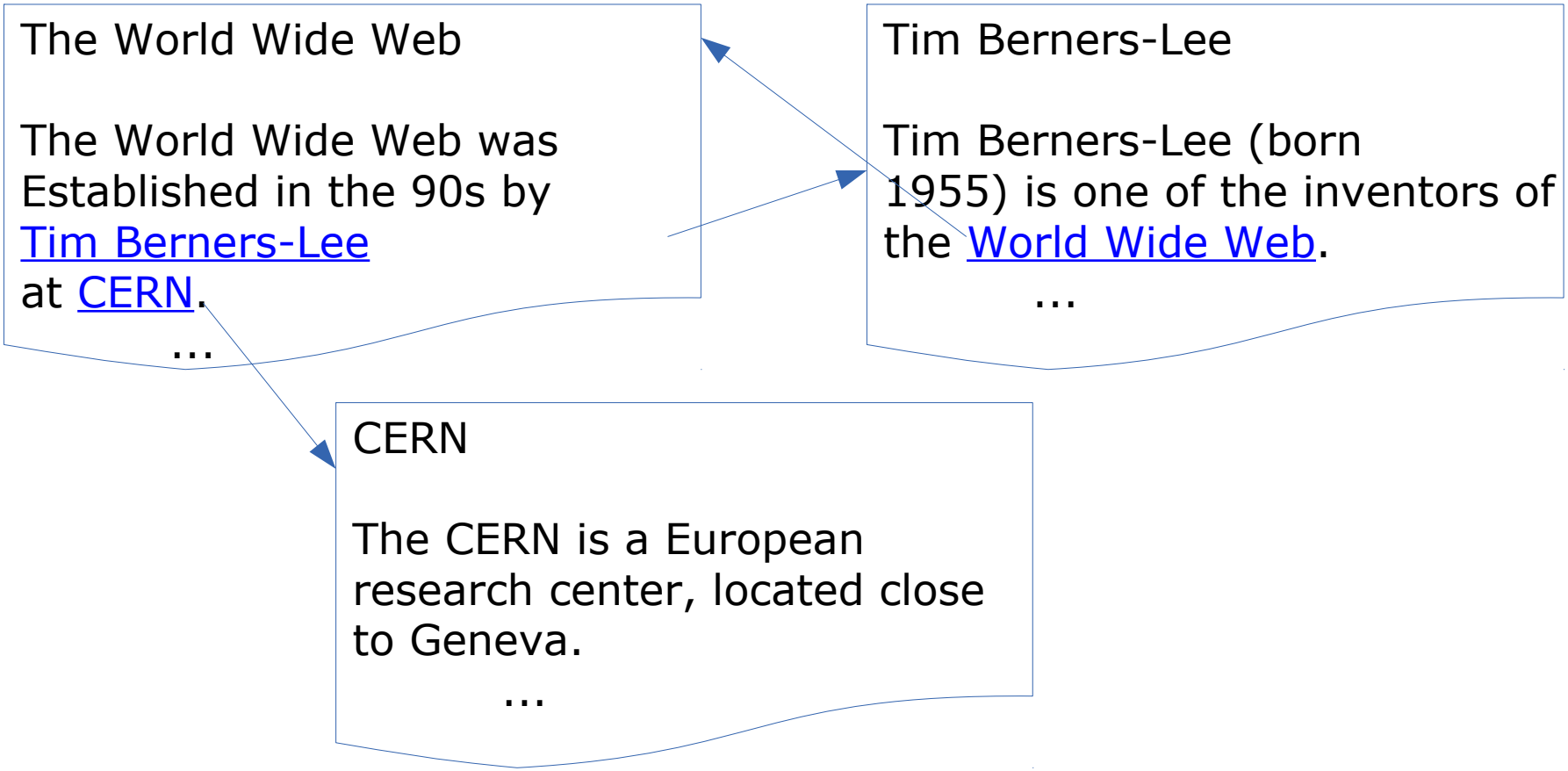
# The “Classic” Web

- a.k.a. “World Wide Web”, “Document 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

# The “Classic” Web

- Hypertext: linked documents



# A Short History of the Web

- Let's see what you know...
- Try to find the correct chronological ordering of the following events:

1. First version of HTML
2. Wikipedia goes online
3. Foundation of Skype
4. First Web catalogue
5. Foundation of the W3C
6. First Search Engine
7. Foundation of Twitter
8. HTTP Standard
9. 500 Servers online
10. Foundation of Facebook
11. Dotcom bubble and stock market crash
12. First version of Internet explorer
13. Foundation of Google
14. First domain registered
15. First version of Firefox
16. TCP/IP Standard
17. 1,000 computers online
18. 1,000,000 computers online
19. 1,000,000,000 computers online
20. First multi user online game

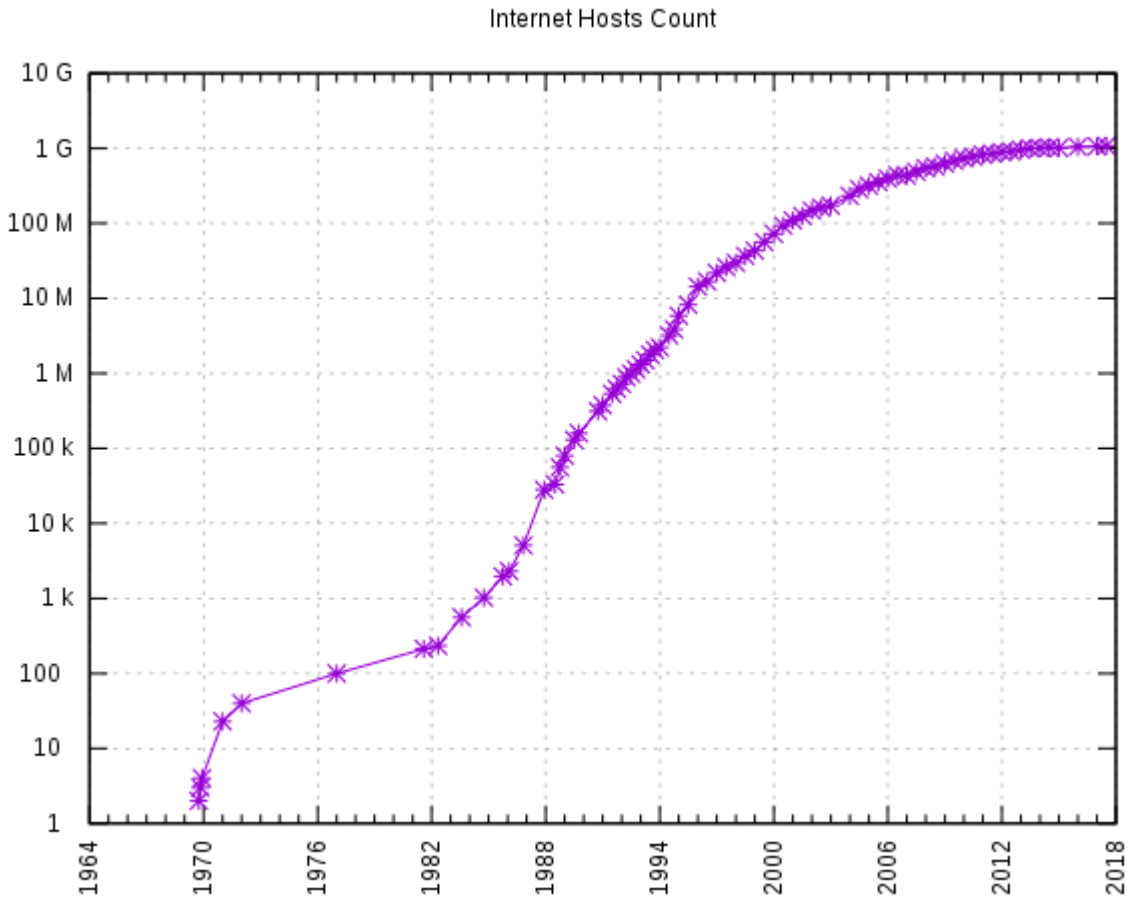
# A Short History of the Web

- 1974: TCP/IP Standard
- 1979: First Multi User Game
- 1985: First domain registered
  1,000 computers online
- 1989: Hypertext concept
  by Tim Berners-Lee
- 1991: First HTML version (20 elements)
- 1992: ~1,000,000 computers online
- 1993: Mosaic-Browser,
  around 500 web servers world wide
- 1994: Full text search engines (WebCrawler, Lycos)
  Web catalogues (Yahoo!, AltaVista)
  Foundation of the W3C

# A Short History of the Web

- 1995: Internet Explorer
- 1996: HTTP Standard
- 1998: Foundation of Google
- 2000: Dotcom Bubble, Stock Market Crash
- 2001: Foundation of Wikipedia
- 2003: Foundation of Skype
- 2004: Foundation of Facebook,
  First version of firefox
- 2006: Foundation of Twitter and WikiLeaks
- 2014: 1,000,000,000 computers online

# Growth of the Web

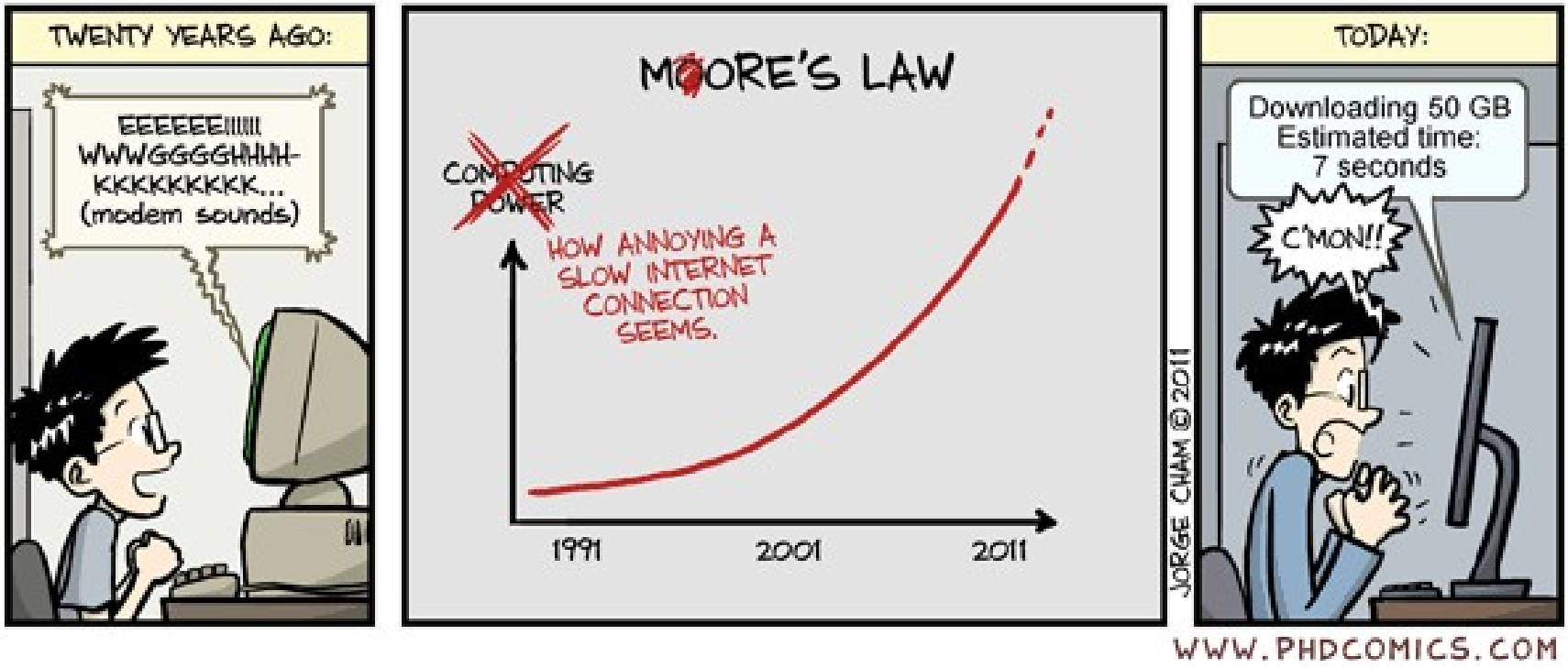



https://commons.wikimedia.org/wiki/File:Internet_Hosts_Count_log.svg

# The Dotcom Bubble and Stock Market Crash


http://de.wikipedia.org/w/index.php?title=Datei:NASDAQ_IXIC_-_dot-com_bubble.png&filetimestamp=20050426161953

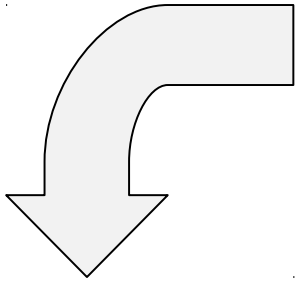
# Evolution of the Web




http://www.phdcomics.com/comics.php?n=1456

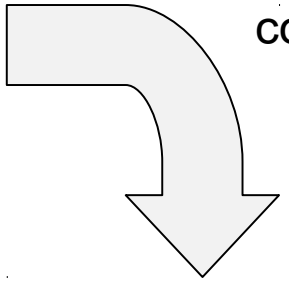
# The “Classic” Web

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:** „hmf298hmmhudsa“
**Print in italics:** „mj2i9ji0“
**Print normal:** „fdsah
02hfadsh0um2m0adsmf0ihm
asdfjköfdsa298ndsfmij32mio
lk2mjpoimjiofdpmsajiomjm“

# 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“

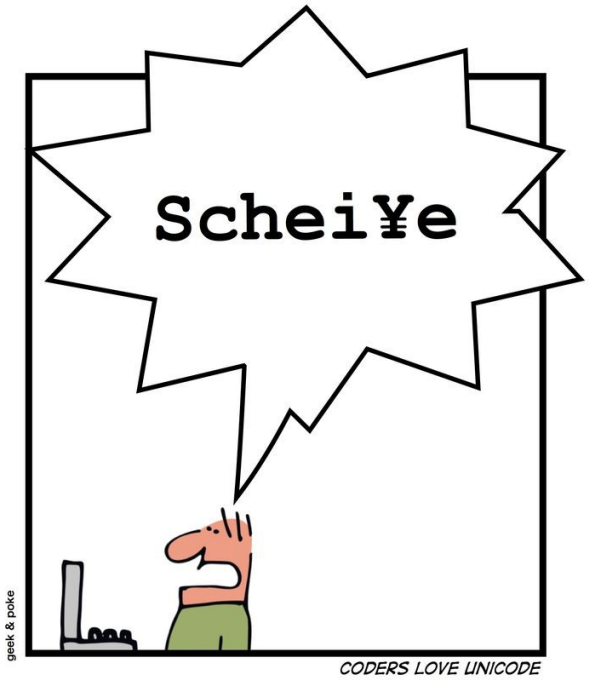
```
<html>
  …
  <b>Dr. Mark Smith</b>
  <i>Physician</i>
  Main St. 14
  Smalltown
  Mon-Fri 9-11 am
  Wed 3-6 pm
  …
</html>
```

→ “classic” Web is too inflexible for useful search

→ hard to use for intelligent agents

# 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



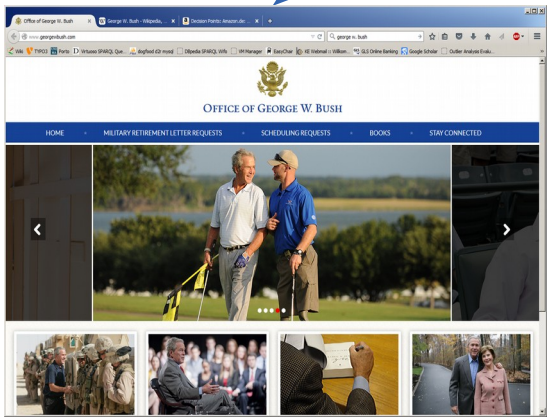

http://geekandpoke.typepad.com/geekandpoke/2011/08/coders-love-unicode.html

# Homonyms and Polysemous Words

# Untyped Links

Bush Era Law Could Get You 20 Years in Prison For Clearing Your Browser History

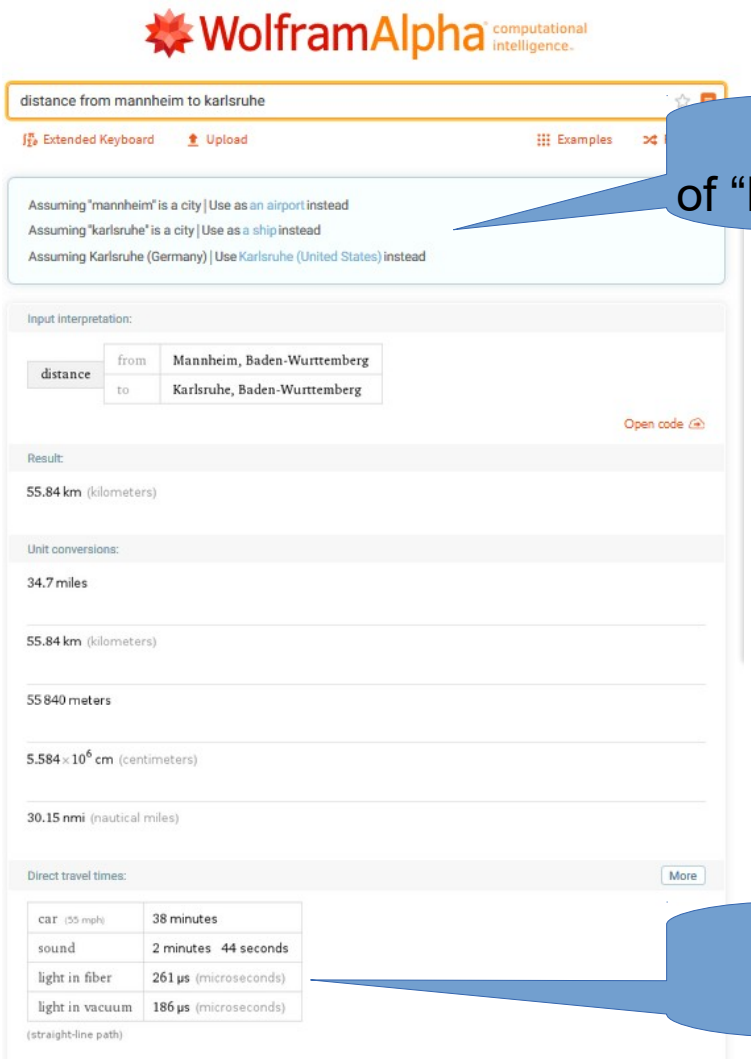
?




?


?

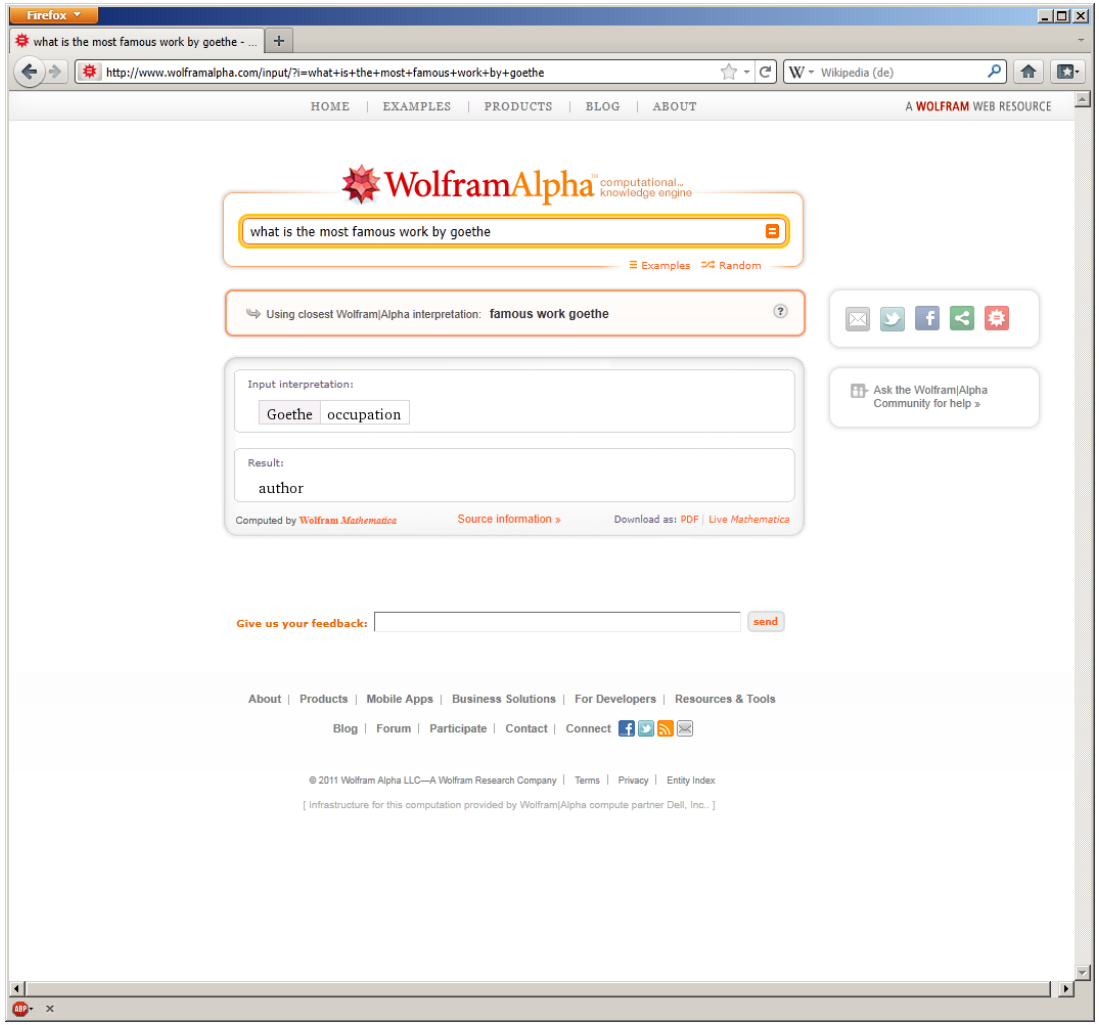
# Example: Wolfram Alpha




Multiple interpretations of “Mannheim” and “Karlsruhe”

Multiple interpretations of “distance”

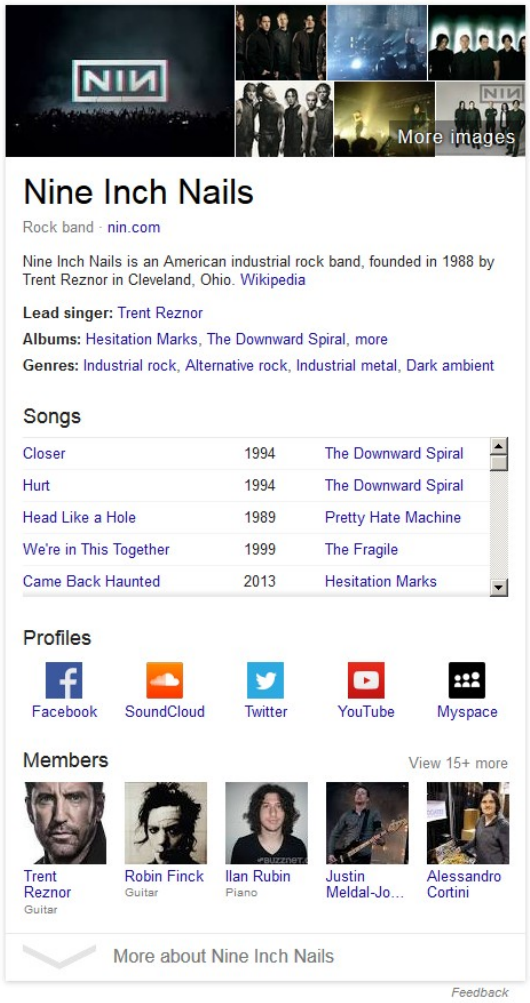
# Example: Wolfram Alpha



# Example: Google Knowledge Graph

- Paradigm shift in Web Search
  - "Things, not strings"
- Contains structured data for many entities
- Displayed to the user in a uniform way
- Connect entities via named links

note: these are *typed* links!

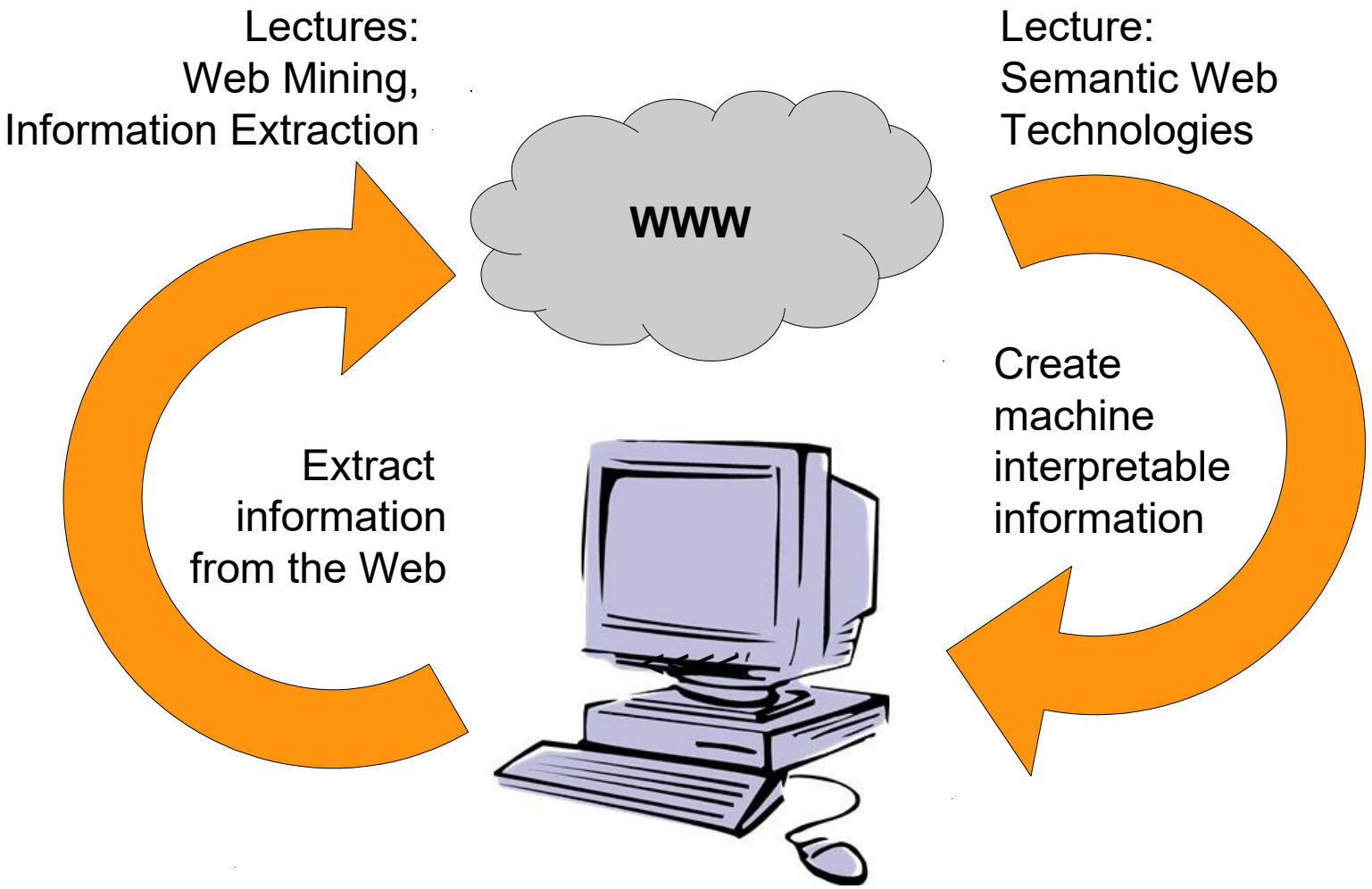


# Example: Enterprise Knowledge Graphs

- Many companies use knowledge graphs
  - As a unified access point to their data
  - To allow joint reasoning over different data sources


https://dl.acm.org/citation.cfm?id=3332266

# Solutions


Lectures:
Web Mining,
Information Extraction
Lecture:
Semantic Web
Technologies
WWW
Create
machine
interpretable
information
Extract
information
from the Web

# The Semantic Web Idea

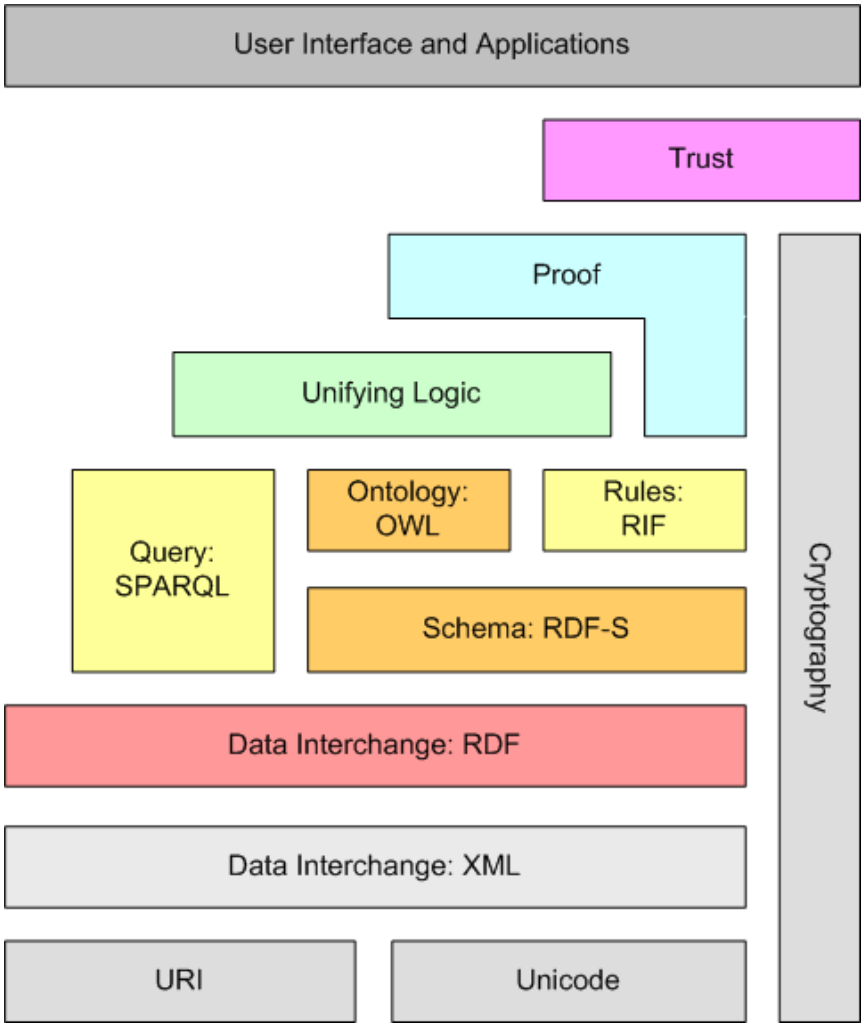
- Provide information in machine interpretable form
- Make (semantic) links between (data) documents usable
- Allow reasoning
- Facilitate useful (!) complex queries

# Semantic Web – Architecture

here be dragons...

Semantic Web Technologies (This lecture)

Technical Foundations



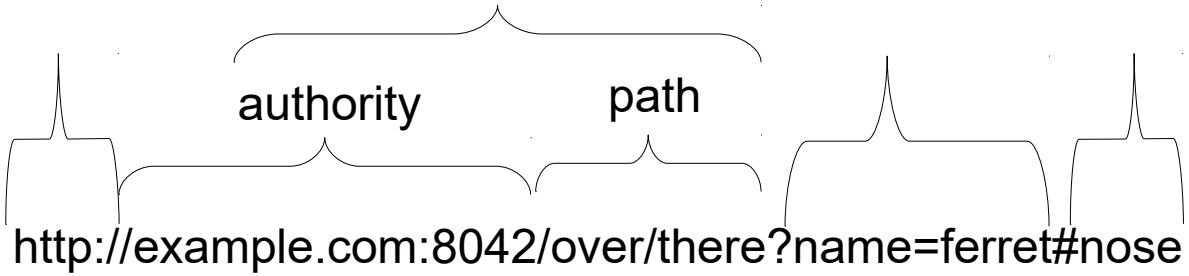

Berners-Lee (2009): *Semantic Web and Linked Data*
http://www.w3.org/2009/Talks/0120-campus-party-tbl/

# Uniform Resource Identifiers (URIs)

- Proposed by Tim-Berners-Lee as „Universal Resource Identifier“ (IETF RFC 1630)
- Standardized: IETF RFC 3986 (2005)

- Used for naming and finding resources on the Web

URI = scheme ":" hier-part [ "?" query ][ "#" fragment ]



# URIs vs. URLs

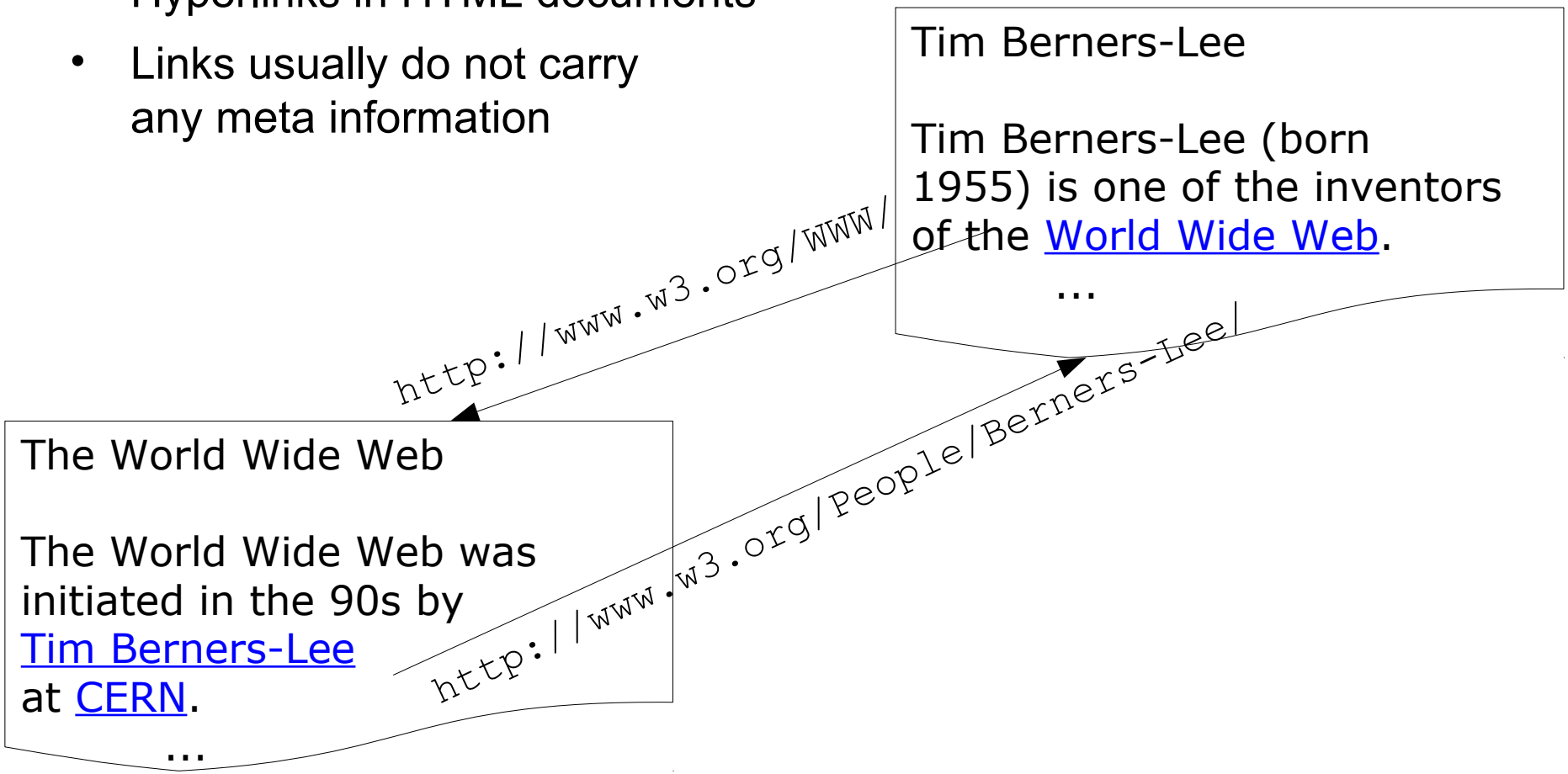
- Uniform Resource Locators (IETF RFC 1738, 1994) are a *subset* of URIs

- 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



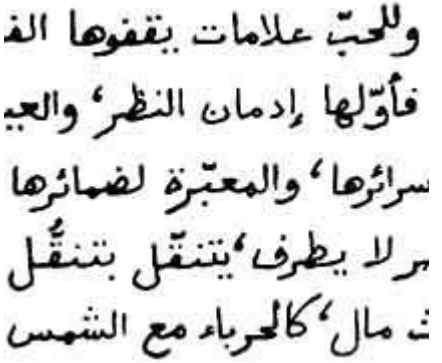
# Character Sets on the Web

- ASCII („American Standard Code for Information Interchange“)
  ISO 646 (1963), 127 characters, 95 of which are printable:

```
 !"#$%&'()*+,-./0123456789:;<=>?
@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_
`abcdefghijklmnopqrstuvwxyz{|}~
```

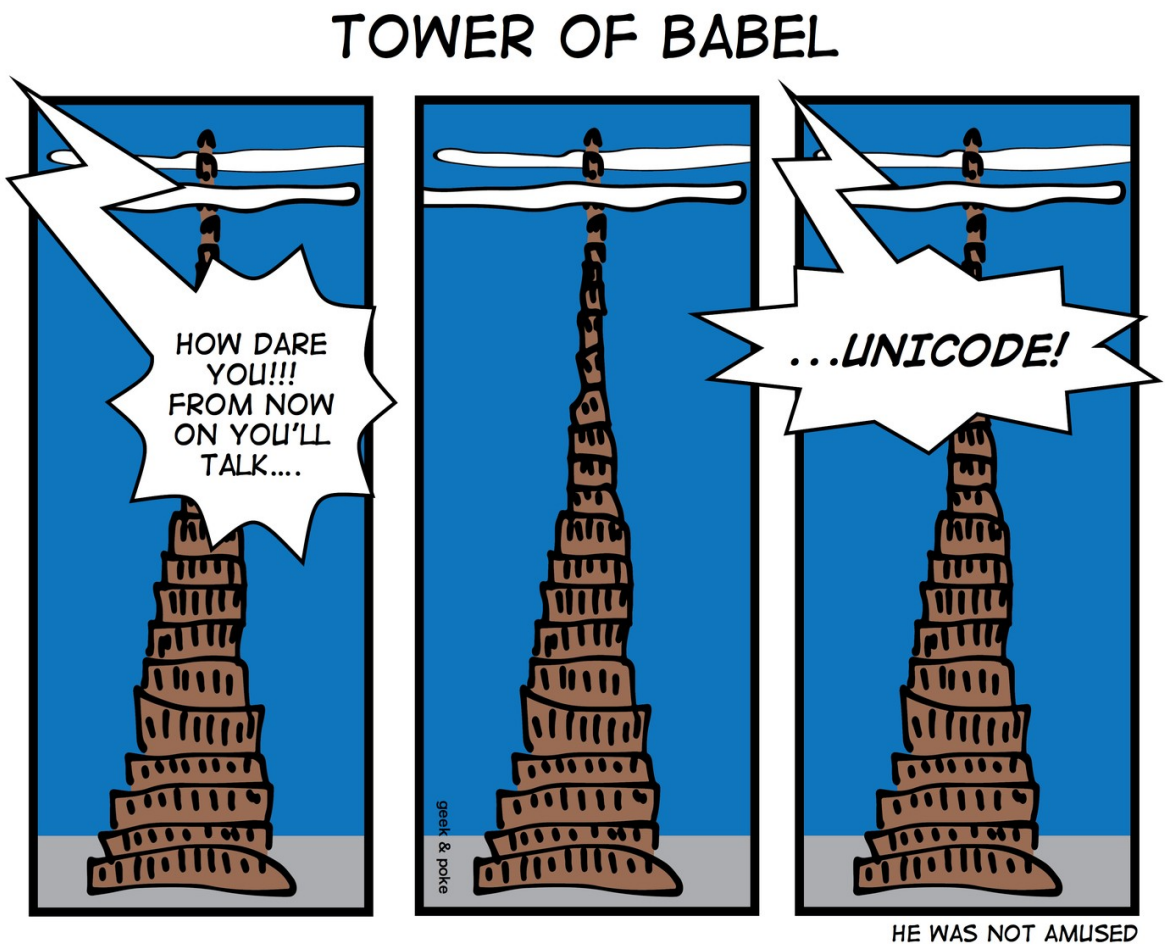
- Extension to 8 Bit: ISO 8859-1 to -16 (1998)
  - covers major European languages
  - most well known: 8859-1 („Latin-1“)

- The Web, however, speaks many more languages...



我爱中国
国中爱我

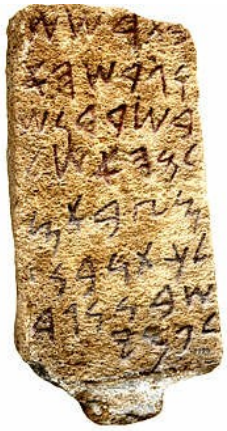
# The Multilingual Web



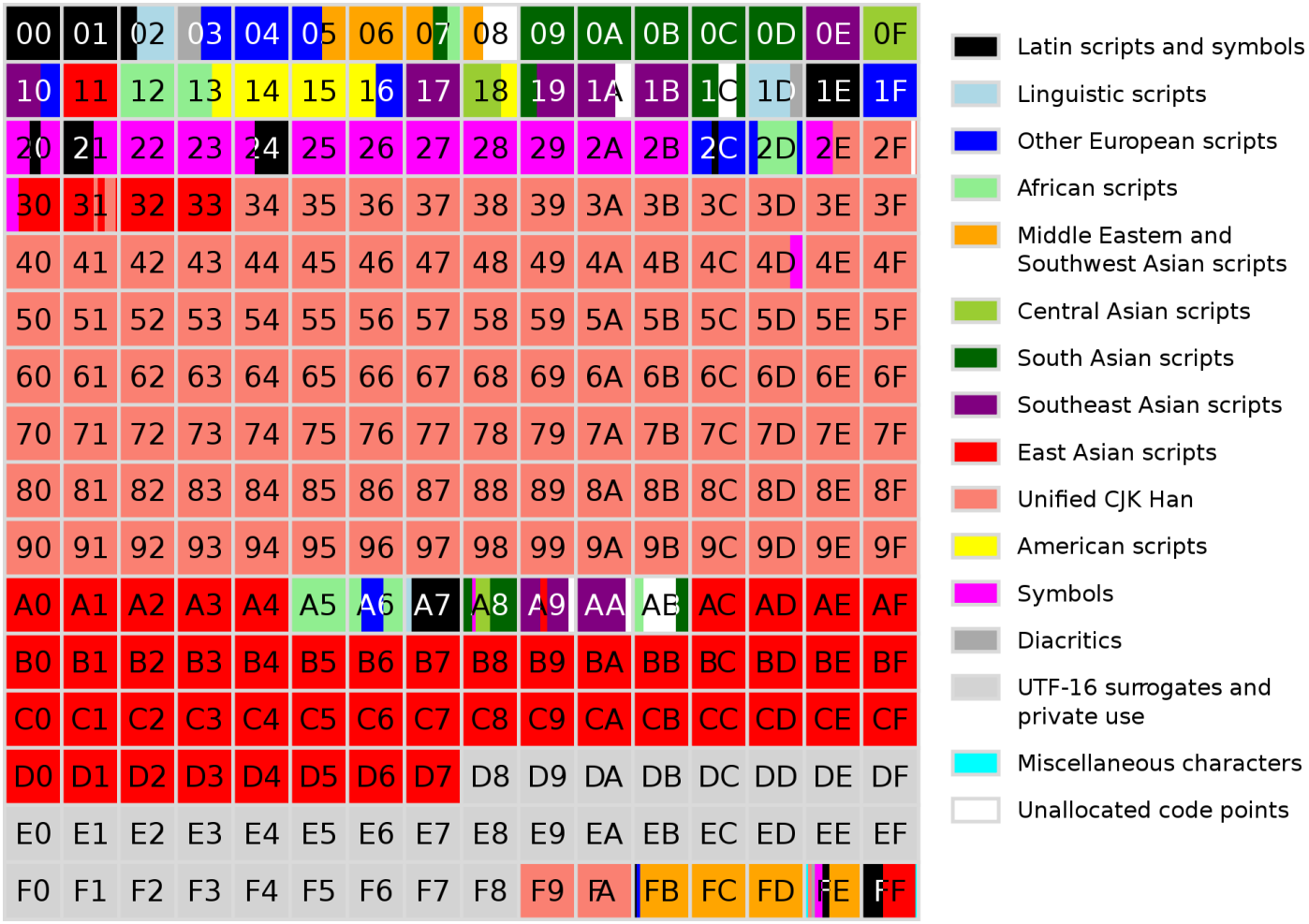

http://geek-and-poke.com/geekandpoke/2013/8/29/when-it-all-began

# Unicode

- ISO 10646
  - first version 1991 (Europe, Near East, India)
  - Unicode 12.1 (May 2019)
  - defines ~138,000 characters
  - covers even very exotic languages
  - Plus: currency symbols, emojis, sign language, ...



# Unicode




Source: Wikimedia Commons

# Information Representation in XML

XML (eXtensible Markup Language)

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


```
<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">

- Tags are nested (with *exactly one* root element):

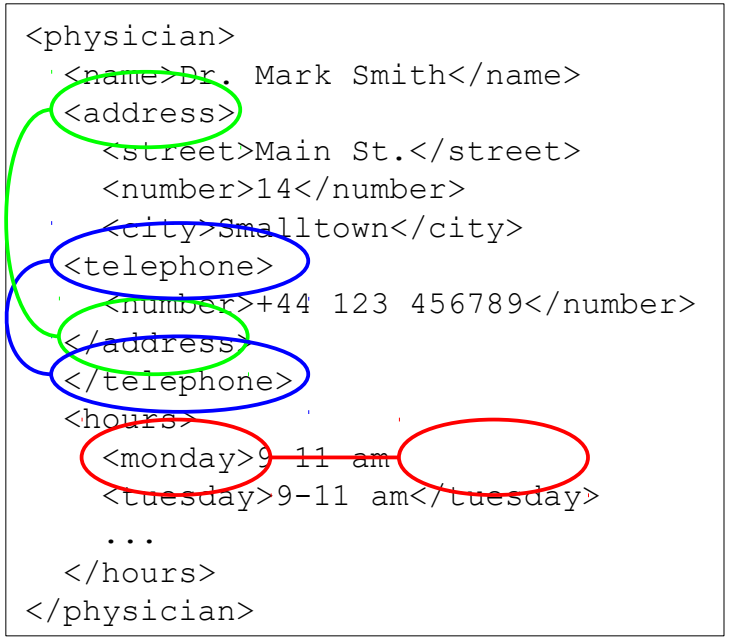
  <physician>

    <address> … </address>

  </physician>

# XML: Well-formed Documents

```
<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>
```



# 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)

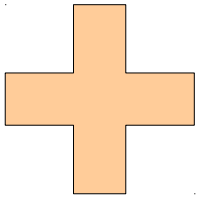
```
/physician[name='Dr. Mark Smith']/telephone/number
```

```
<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>
```

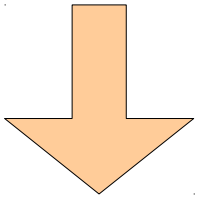
# XSLT: Transformation of XML Documents

- Stylesheet based processing of XML documents
- A W3C standard since 1999
- Uses XPath

```
<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>
```



```
<xsl:template match="/pyhsician">
  <b>
    <xsl:value-of select="name"/>
  </b>
<xsl:template/>
```



```
<b>Dr. Mark Smith</b>
```

# 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

```
<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)

```
<!DOCTYPE physician [

<!ELEMENT physician (
  name,
  address*,
  telephone?,
  fax?,
  hours)>

<!ELEMENT address (
  street,
  number,
  city)>

<!ELEMENT street (#PCDATA)>

  ...

]>
```

```
<!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

```
<!ELEMENT address(street,no,line*,zip,city,state?)>
```

  - ?, + and * mark optional and possible multiple elements

- Definition of attribute lists

```
<!ATTLIST person title CDATA>
```

  - Allowed modifiers: #REQUIRED, #FIXED, #IMPLIED, "..."
  - Enumerating allowed values: (dr|prof)

- Definition of entities:

```
<!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 without fixed order, …)
  - Data types (Numbers, dates, …), own types may be defined
  - Support for namespaces
  - Possibility to create modular schemas

# XML Schema

```
<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 name="address">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="street"
               type="xs:string">
               ...
            </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>
```

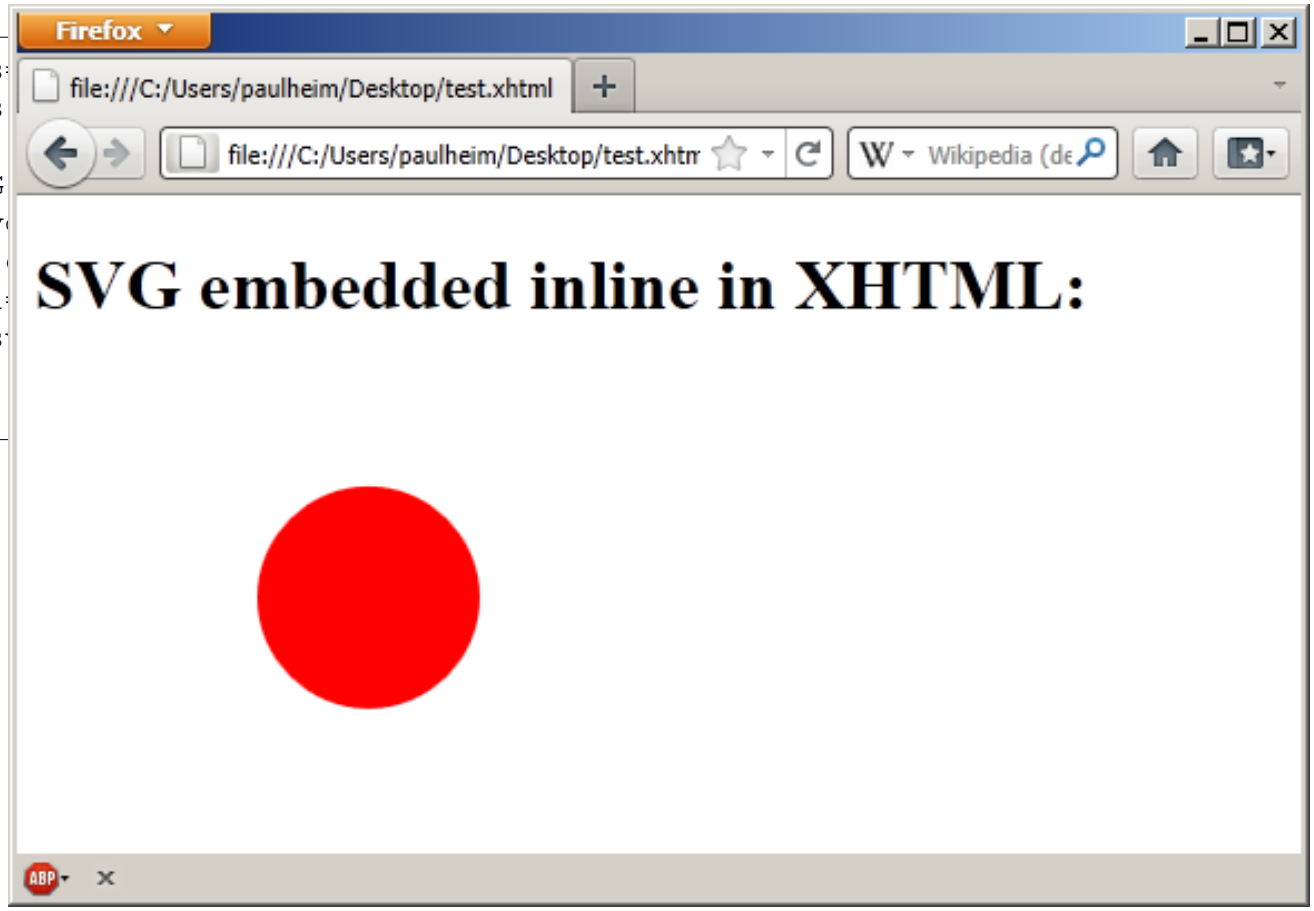
# XML Schema – Modular Schemas

```
<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 ref="addr:address" />
        ...
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

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

  <xs:element name="address">
     <xs:complexType>
        <xs:sequence>
           <xs:element name="street"
            type="xs:string">
            ...
         </xs:sequence>
      </xs:complexType>
   </xs:element>
</xs:schema>
```

# Example: Modular Schemas in XHTML

```
<html xmlns=
      xmlns
  <body>
    <h1>SVG
    <svg:sv
      <svg:
       fill
    </svg:s
  </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.“

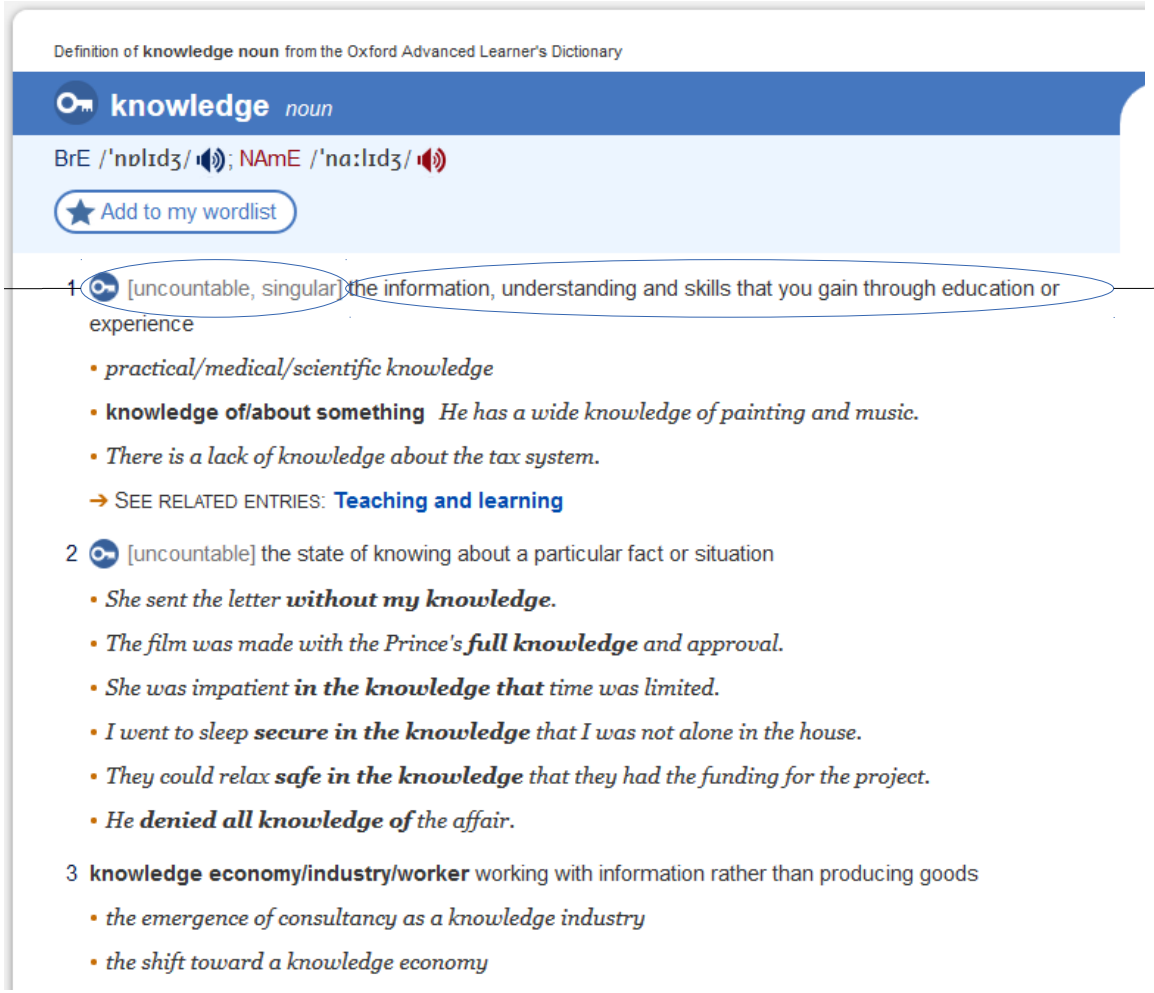
  „The dreaming lamp ~~give~~ gives a freshly cut ~~juices~~ juice to the ~~tire~~ tired sink.

- 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

syntax



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

**knowledge** *noun*

BrE /ˈnɒlɪdʒ/; NAmE /ˈnɑːlɪdʒ/

Add to my wordlist

1 [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.*

3 **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*

semantics

# So, what does a DTD/Schema Define?

Employee catalog
of the hospital

```
<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

```
<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!*

```
<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*>
```

# A Note on Web Services

- Original vision
  - Describe functions of services as XML
  - e.g., stock market ticker, calculator, travel booking…
- ...so that an intelligent agent can combine them
  - and dynamically create a system for a given purpose

- Standards
  - WSDL, UDDI, SOAP, ...

- Problem
  - The semantics is missing!

# Wrap Up

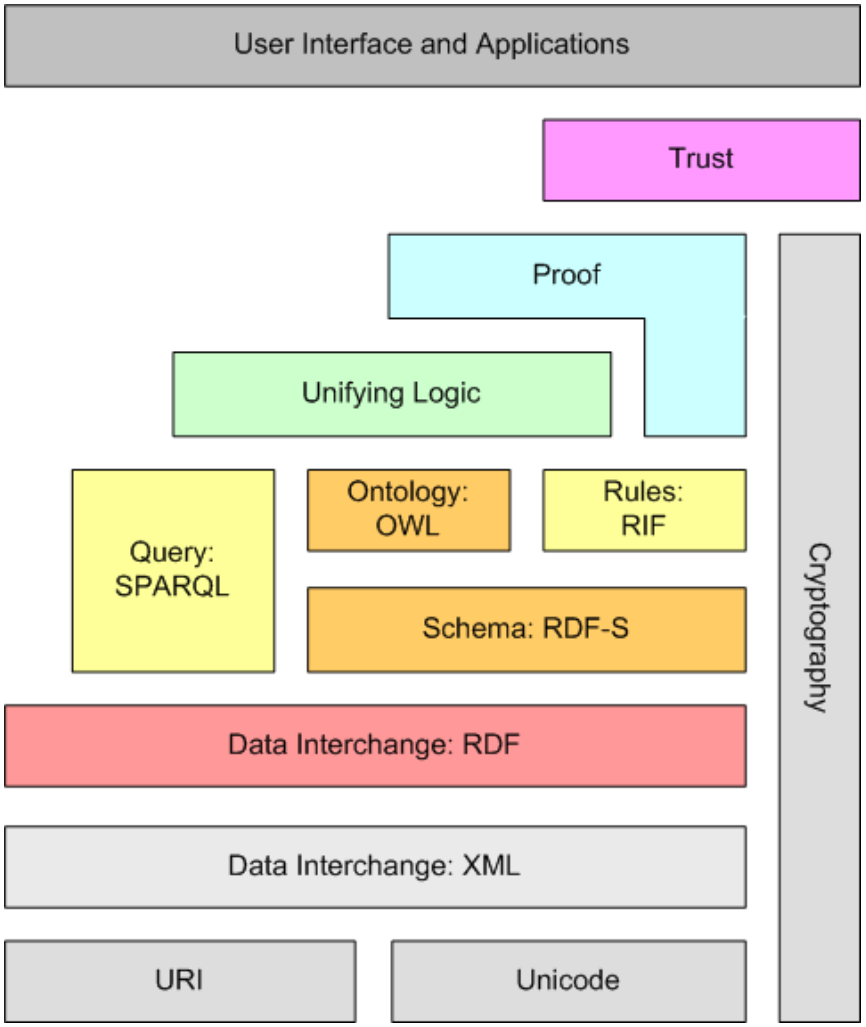
- Problems of the classic web
  - Not usable for machines / intelligent agents
- URIs
  - Unique identifiers for resource
  - URLs are dereferencable on the Web
- Unicode
  - A character set for all languages
- XML
  - XPath
  - XSLT
  - DTD
  - XML Schema

# Semantic Web – Architecture

here be dragons...

Semantic Web Technologies (This lecture)

Technical Foundations




Berners-Lee (2009): *Semantic Web and Linked Data*
http://www.w3.org/2009/Talks/0120-campus-party-tbl/

# Questions?