Microdata and schema.org

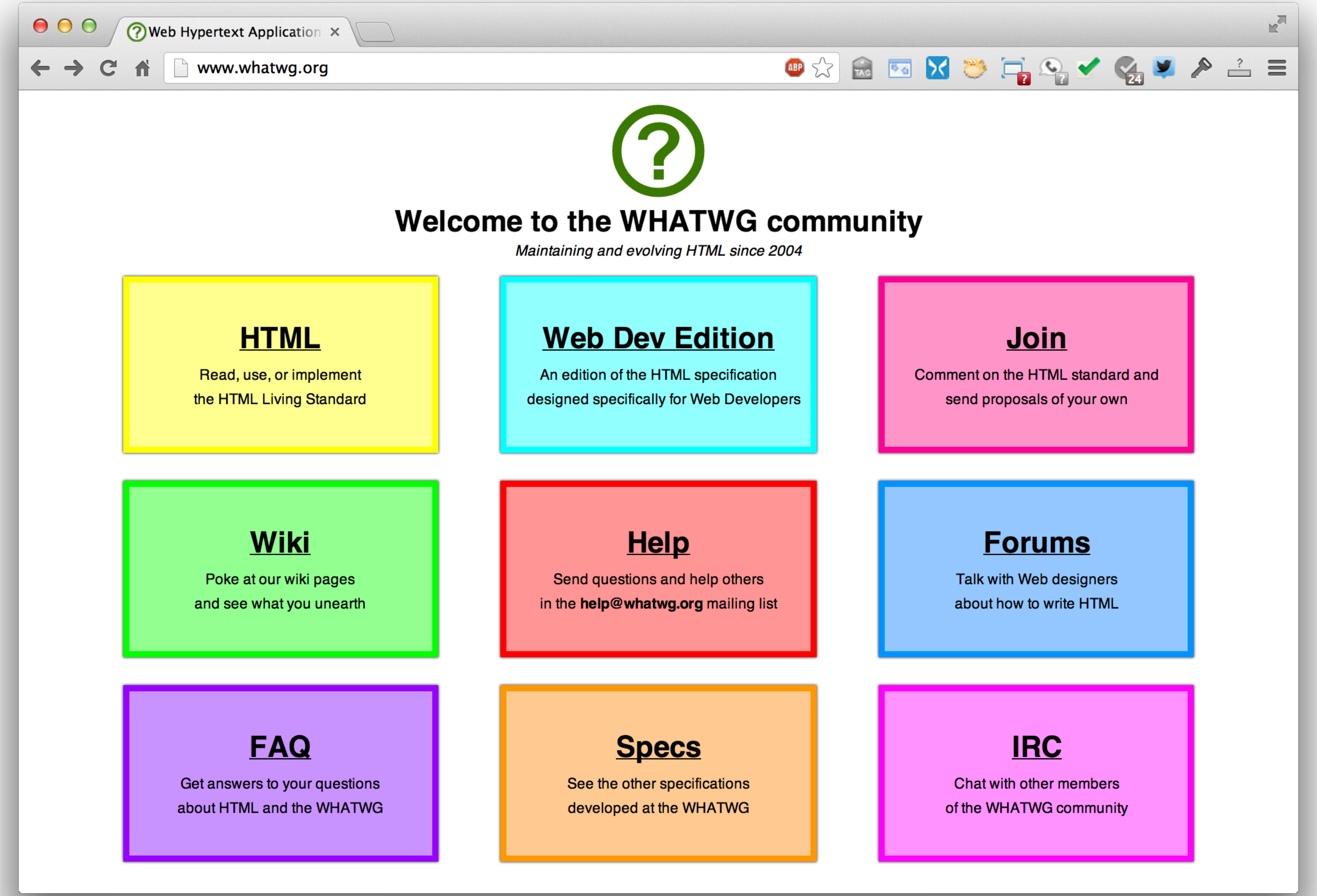
Basics

- Microdata is a simple semantic markup scheme that’s an alternative to RDFa
- Developed by WHATWG and supported by major search companies (Goog,e, MSFT, Yahoo)
- Like RDFa, it uses HTML tag attributes to host metadata
- Vocabularies are controlled and hosted at schema.org

What is WHATWG?

- Web Hypertext Application Technology Working Group
  - Community interested in evolving the Web with focus on HTML and Web API development
  - Ian Hickson is a key person, now at Google
- Founded in 2004 by individuals from Apple, Mozilla and Opera after a W3C workshop
  - Concern about W3C's embrace of XHTML
- Current work on HTML5
- Developed Microdata spec

http://whatwg.org/



Welcome to the WHATWG community

Maintaining and evolving HTML since 2004

HTML
Read, use, or implement the HTML Living Standard

Web Dev Edition
An edition of the HTML specification designed specifically for Web Developers

Join
Comment on the HTML standard and send proposals of your own

Wiki
Poke at our wiki pages and see what you unearth

Help
Send questions and help others in the **help@whatwg.org** mailing list

Forums
Talk with Web designers about how to write HTML

FAQ
Get answers to your questions about HTML and the WHATWG

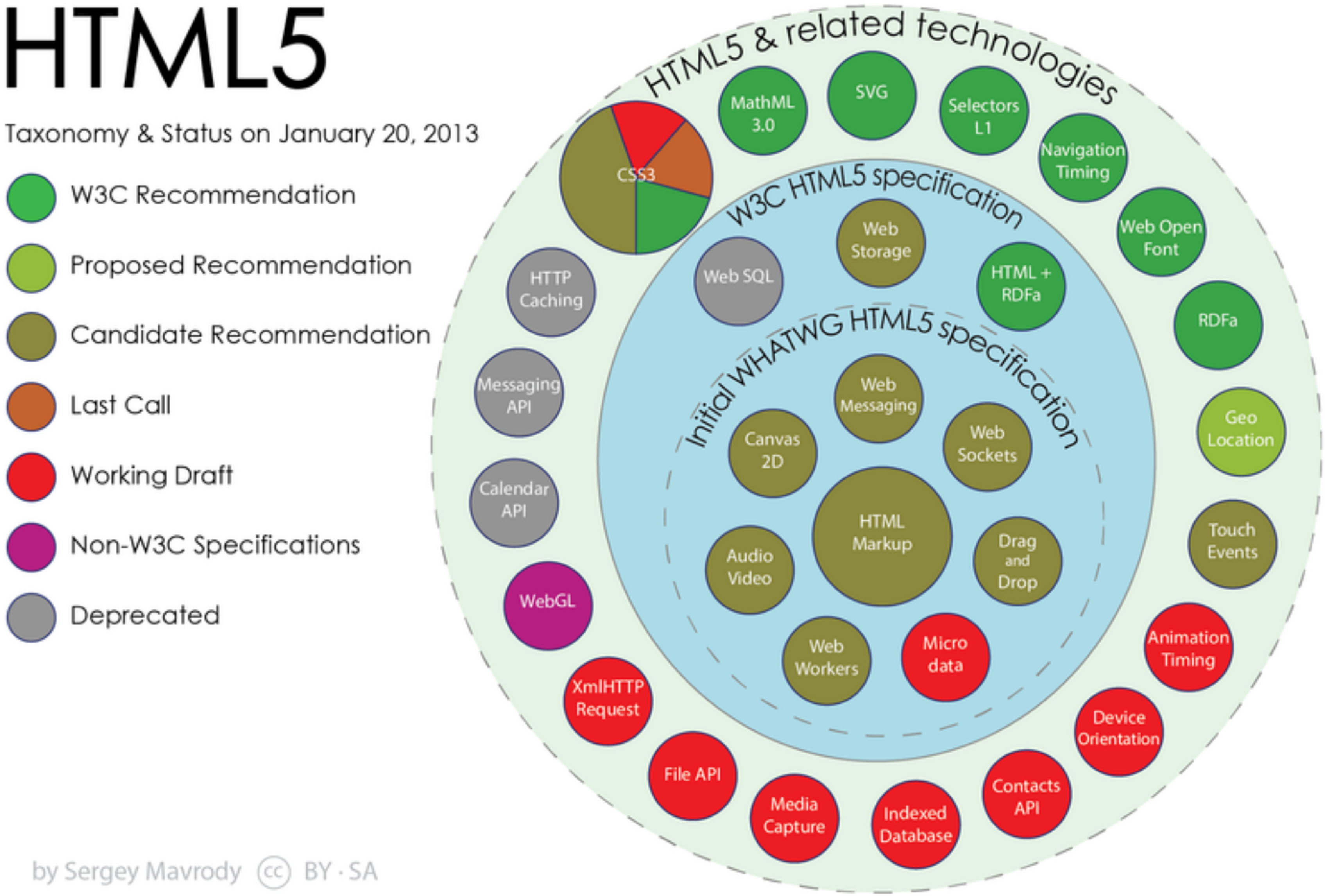
Specs
See the other specifications developed at the WHATWG

IRC
Chat with other members of the WHATWG community

HTML5

- Started by WHATWG as an alternative to XHTML, joined by W3C
  - A W3C candidate recommendation in 2012
  - WHATWG will evolve it as a "living standard"
- HTML5 ≈ HTML + CSS + js
- Native support for graphics, video, audio, speech, semantic markup, …
- Partial support in current browsers + extensions

HTML taxonomy and status



HTML5

Taxonomy & Status on January 20, 2013

- ● W3C Recommendation
- ● Proposed Recommendation
- ● Candidate Recommendation
- ● Last Call
- ● Working Draft
- ● Non-W3C Specifications
- ● Deprecated

by Sergey Mavrody (cc) BY · SA

Microdata

- The microdata effort has two parts: markup and a set of vocabularies
- The markup is similar to RDFa in that it provides a way to identify subjects, types, properties and objects
- The sanctioned vocabularies are found at schema.org and include a small number of very useful ones: people, movies, etc.

An example

```
<div>
 <h1>Avatar</h1>
 <span>Director: James Cameron (born 1954) </span>
 <span>Science fiction</span>
 <a href=”avatar-trailer.html">Trailer</a>
</div>
```

An example: itemscope

- An *itemscope* attribute identifies a content subtree that is the subject about which we want to say something

```
<div itemscope >
 <h1>Avatar</h1>
 <span>Director: James Cameron (born 1954) </span>
 <span>Science fiction</span>
 <a href=”avatar-trailer.html">Trailer</a>
</div>
```

An example: itemtype

- An *itemscope* attribute identifies a content subtree that is the subject about which we want to say something
- The *itemtype* attribute specifies the subject’s type

```
<div itemscope itemtype="http://schema.org/Movie">
 <h1>Avatar</h1>
 <span>Director: James Cameron (born 1954) </span>
 <span>Science fiction</span>
 <a href=”avatar-trailer.html">Trailer</a>
</div>
```

An example: itemprop

- An *itemscope* attribute identifies a content subtree that is the subject about which we want to say something
- The *itemtype* attribute specifies the subject’s type
- An *itemprop* attribute gives a property of that type

```
<div itemscope itemtype="http://schema.org/Movie">
 <h1 itemprop="name">Avatar</h1>
 <span>Director: James Cameron (born 1954) </span>
 <span itemprop="genre">Science fiction</span>
 <a href=”avatar-trailer.html” itemprop="trailer">Trailer</a>
</div>
```

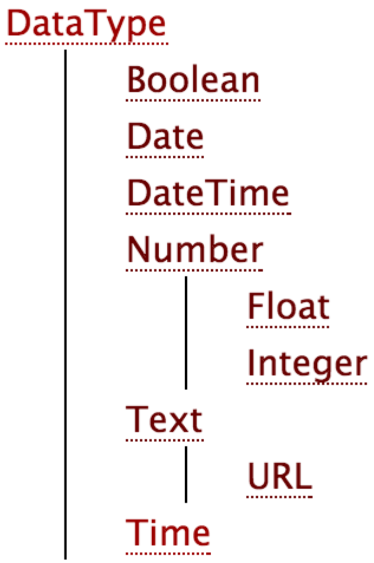
An example: embedded items

- An itemprop immediately followed by another itemcope makes the value an object

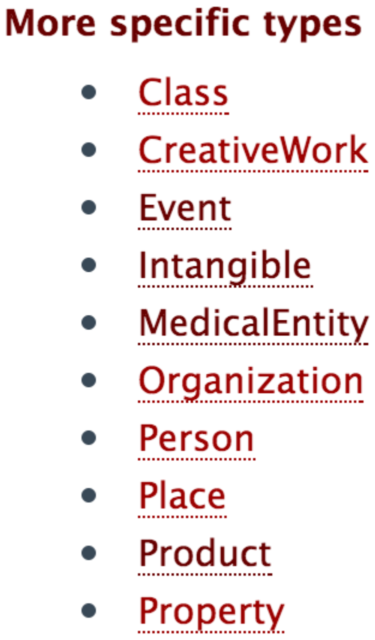
```
<div itemscope itemtype="http://schema.org/Movie">
 <h1 itemprop=“name”>Avatar</h1>
<div itemprop="director”
        itemscope itemtype="http://schema.org/Person">
   Director: <span itemprop="name">James Cameron</span>
(born <span itemprop="birthDate">1954</span>) </div>
 <span itemprop=“genre”>Science fiction</span>
 <a href=”avatar-trailer.html” itemprop="trailer">Trailer</a>
</div>
```

schema.org vocabulary

- Full type hierarchy in one file
- As of 4/23/13: 419 classes, 756 properties
- **Data types:** Boolean, Date, DateTime, Number (Float, Integer, Text (URL), Time
- **Objects:** Rooted at Thing with two ‘metaclasses’ (Class and Property) and eight subclasses



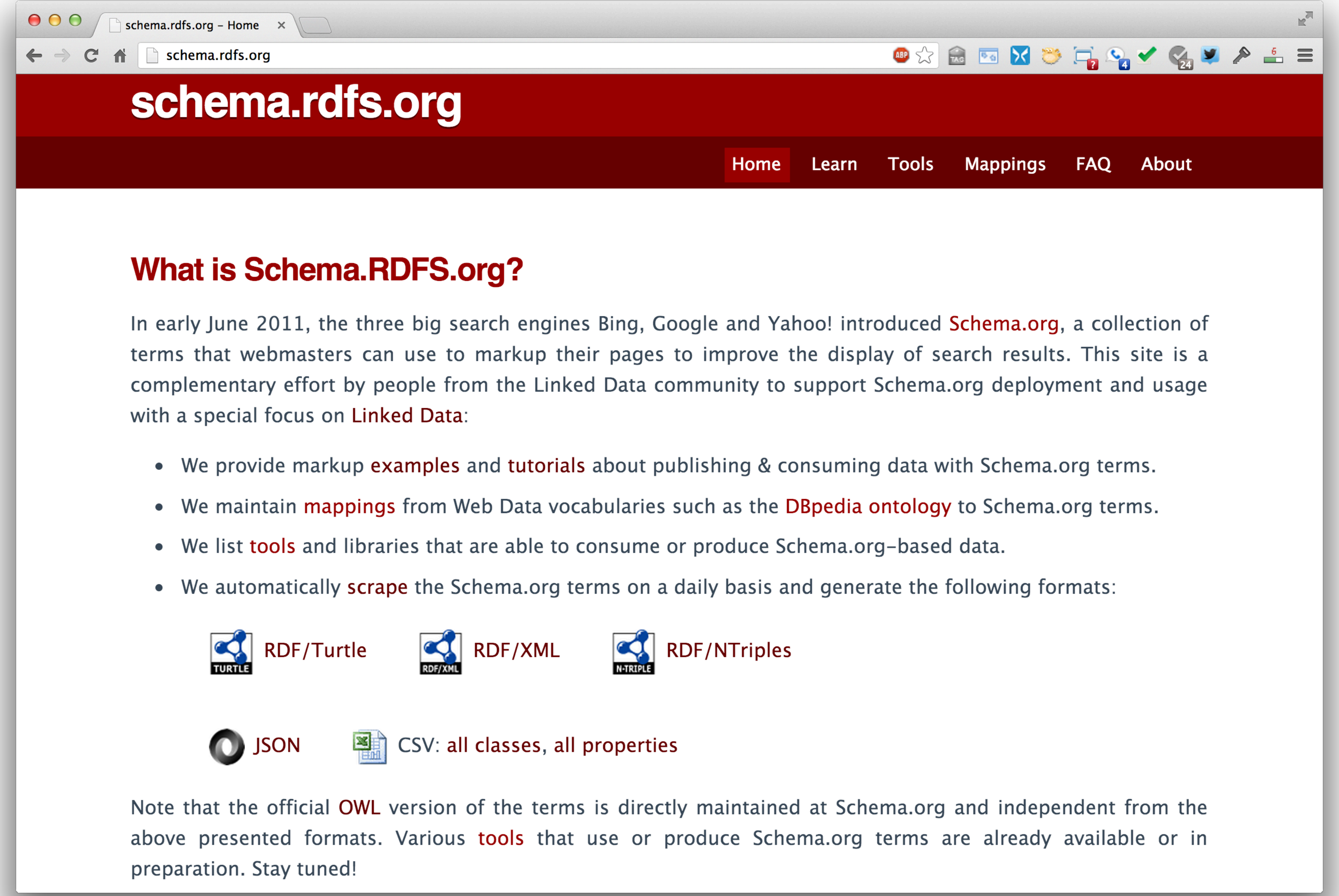
- DataType
  - Boolean
  - Date
  - DateTime
  - Number
    - Float
    - Integer
  - Text
    - URL
  - Time



More specific types

- Class
- CreativeWork
- Event
- Intangible
- MedicalEntity
- Organization
- Person
- Place
- Product
- Property

http://schema.rdf.org



schema.rdfs.org

Home Learn Tools Mappings FAQ About

What is Schema.RDFS.org?

In early June 2011, the three big search engines Bing, Google and Yahoo! introduced Schema.org, a collection of terms that webmasters can use to markup their pages to improve the display of search results. This site is a complementary effort by people from the Linked Data community to support Schema.org deployment and usage with a special focus on Linked Data:

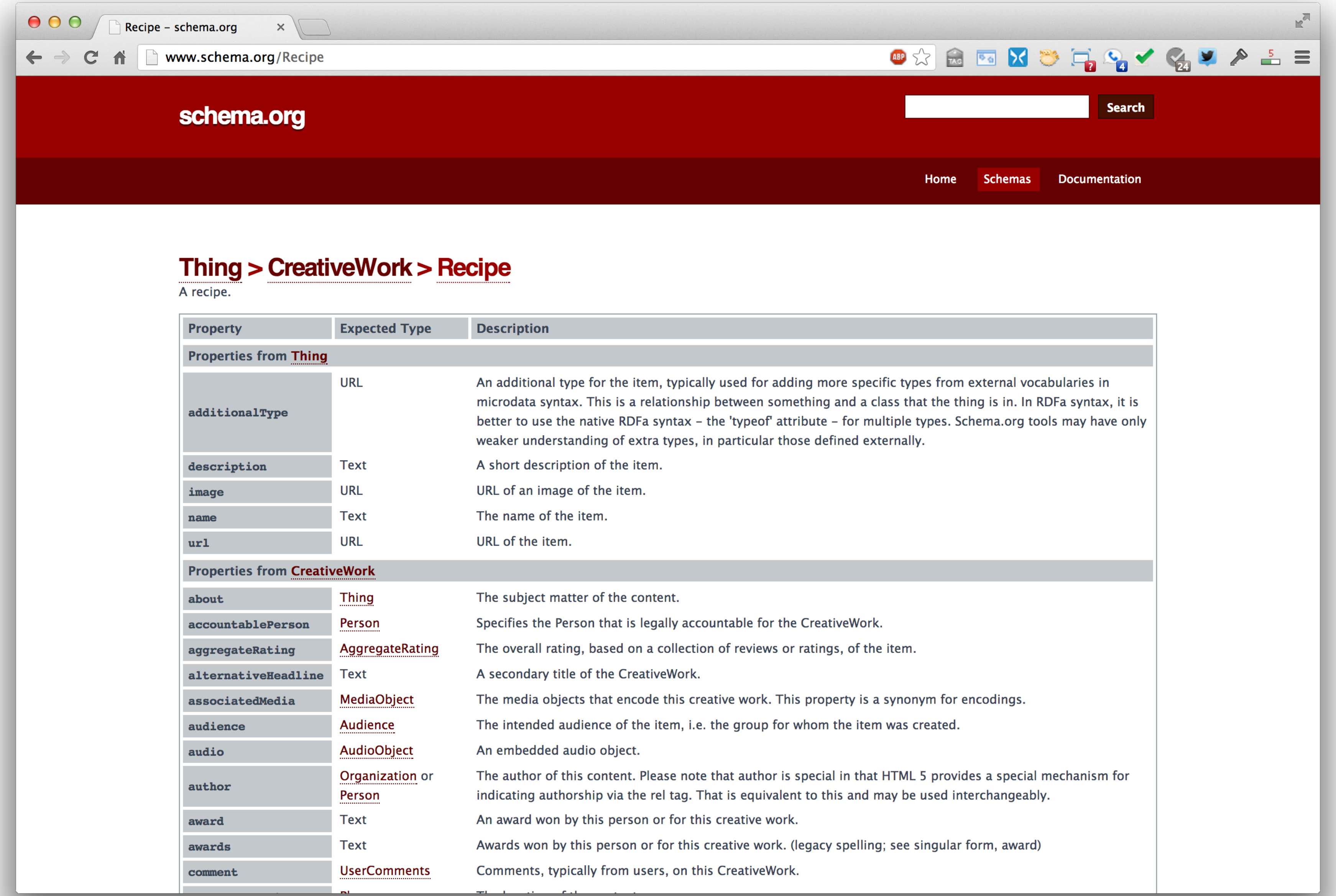
- We provide markup examples and tutorials about publishing & consuming data with Schema.org terms.
- We maintain mappings from Web Data vocabularies such as the DBpedia ontology to Schema.org terms.
- We list tools and libraries that are able to consume or produce Schema.org-based data.
- We automatically scrape the Schema.org terms on a daily basis and generate the following formats:

TURTLE RDF/Turtle   RDF/XML RDF/XML   N-TRIPLE RDF/NTriples

JSON   CSV: all classes, all properties

Note that the official OWL version of the terms is directly maintained at Schema.org and independent from the above presented formats. Various tools that use or produce Schema.org terms are already available or in preparation. Stay tuned!

http://www.schema.org/Recipe



Thing > CreativeWork > Recipe

A recipe.

Property	Expected Type	Description
Properties from Thing		
additionalType	URL	An additional type for the item, typically used for adding more specific types from external vocabularies in microdata syntax. This is a relationship between something and a class that the thing is in. In RDFa syntax, it is better to use the native RDFa syntax – the 'typeof' attribute – for multiple types. Schema.org tools may have only weaker understanding of extra types, in particular those defined externally.
description	Text	A short description of the item.
image	URL	URL of an image of the item.
name	Text	The name of the item.
url	URL	URL of the item.
Properties from CreativeWork		
about	Thing	The subject matter of the content.
accountablePerson	Person	Specifies the Person that is legally accountable for the CreativeWork.
aggregateRating	AggregateRating	The overall rating, based on a collection of reviews or ratings, of the item.
alternativeHeadline	Text	A secondary title of the CreativeWork.
associatedMedia	MediaObject	The media objects that encode this creative work. This property is a synonym for encodings.
audience	Audience	The intended audience of the item, i.e. the group for whom the item was created.
audio	AudioObject	An embedded audio object.
author	Organization or Person	The author of this content. Please note that author is special in that HTML 5 provides a special mechanism for indicating authorship via the rel tag. That is equivalent to this and may be used interchangeably.
award	Text	An award won by this person or for this creative work.
awards	Text	Awards won by this person or for this creative work. (legacy spelling; see singular form, award)
comment	UserComments	Comments, typically from users, on this CreativeWork.

Microdata as a KR language

- More than RDF, less than RDFS
- Properties have an *expected* type (range)
  - Might be a string
  - A list of types, any of which are OK
- Properties attached to one or more types (domain)
- Classes can have multiple parents and inherit (properties) from all of them
- No axioms (e.g., disjointness, cardinality, etc.)

Mixing markup from other vocabularies

- Microdata is intended to work with one vocabulary – the one at schema.org
- Advantages
  - Simple, organized, well designed
  - Controlled by the schema.org people
- Disadvantages: too simple, controlled
  - Too simple, narrow, mono-lingual
  - Controlled by the schema.org people

Extending the schema.org ontology

- http://www.schema.org/docs/extension.html
- You can subclass existing classes
  - Person/Engineer
  - Person/Engineer/ElectricalEngineer
- Subclass exisiting properties
  - musicGroupMember/leadVocalist
  - musicGroupMember/leadGuitar1
  - musicGroupMember/leadGuitar2

Extension Problems

- Do agreed upon meaning
  - Through axioms supported by the language (e.g., equivalence, disjointness, etc.)
  - No place for documentation (annotations, labels, comments)
- Without a namespace mechanism, your Person/Engineer and mine can be confused and might mean different things

Conclusions

- Microdata is a good effort by the search companies to experiment with a simple semantic language
- It’s not a great standard
- RDFa has a more powerful encoding and works with the RDF stack
- There’s a bit of infighting in the WEB community
- RDFa Lite is maybe a good solution