

Microdata and schema.org

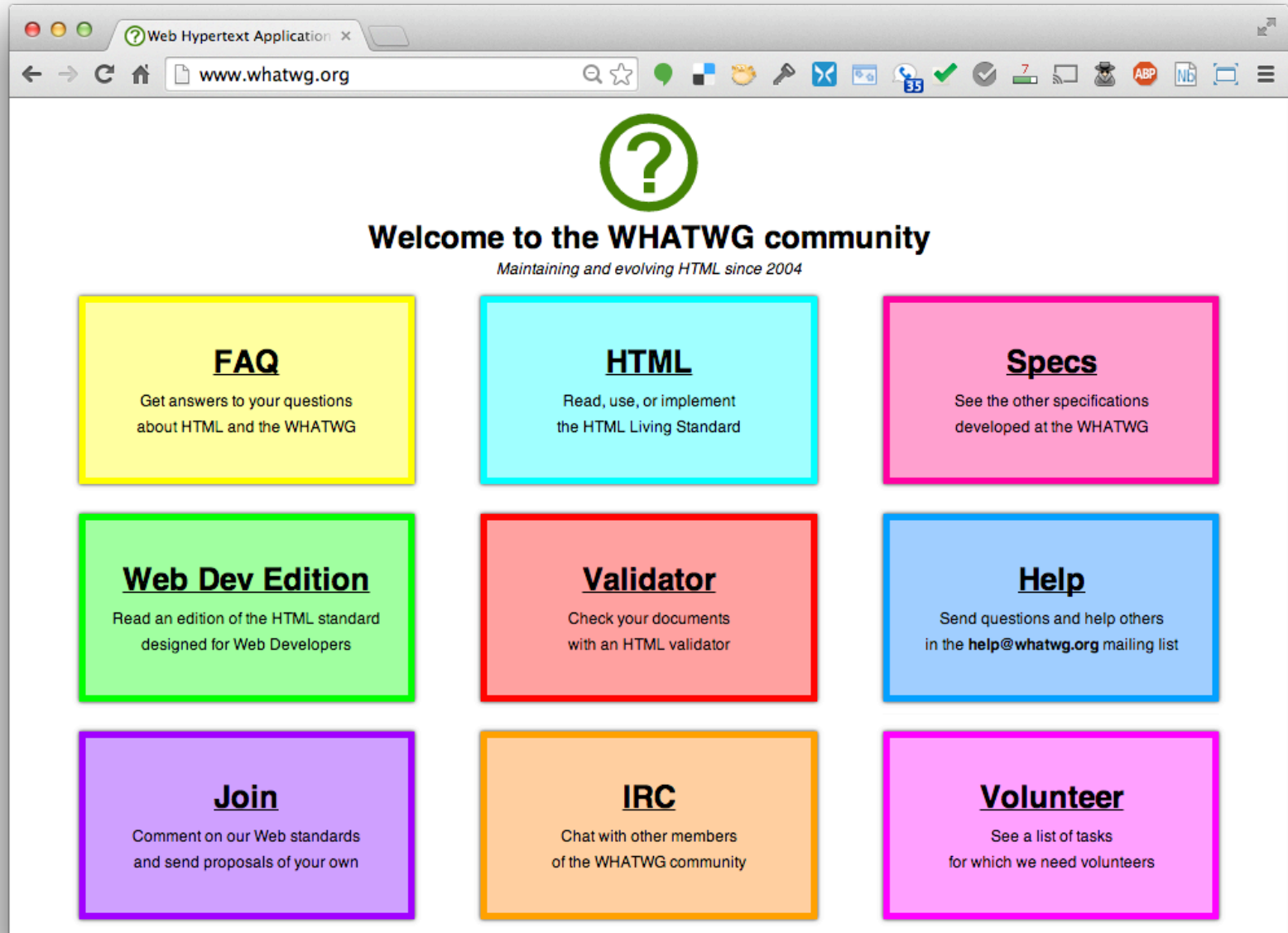
Basics

- Microdata is a simple semantic markup scheme that's an alternative to RDFa
- Developed by WHATWG and supported by major search companies (Google, Microsoft, Yahoo, Yandex)
- 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/



The image shows a browser window displaying the homepage of the WHATWG community. The browser's address bar shows the URL www.whatwg.org. The page features a large green question mark icon at the top center, followed by the heading "Welcome to the WHATWG community" and the tagline "Maintaining and evolving HTML since 2004". Below this, there is a 3x3 grid of colorful boxes, each containing a link and a brief description:

<u>FAQ</u> Get answers to your questions about HTML and the WHATWG	<u>HTML</u> Read, use, or implement the HTML Living Standard	<u>Specs</u> See the other specifications developed at the WHATWG
<u>Web Dev Edition</u> Read an edition of the HTML standard designed for Web Developers	<u>Validator</u> Check your documents with an HTML validator	<u>Help</u> Send questions and help others in the help@whatwg.org mailing list
<u>Join</u> Comment on our Web standards and send proposals of your own	<u>IRC</u> Chat with other members of the WHATWG community	<u>Volunteer</u> See a list of tasks for which we need volunteers



HTML5

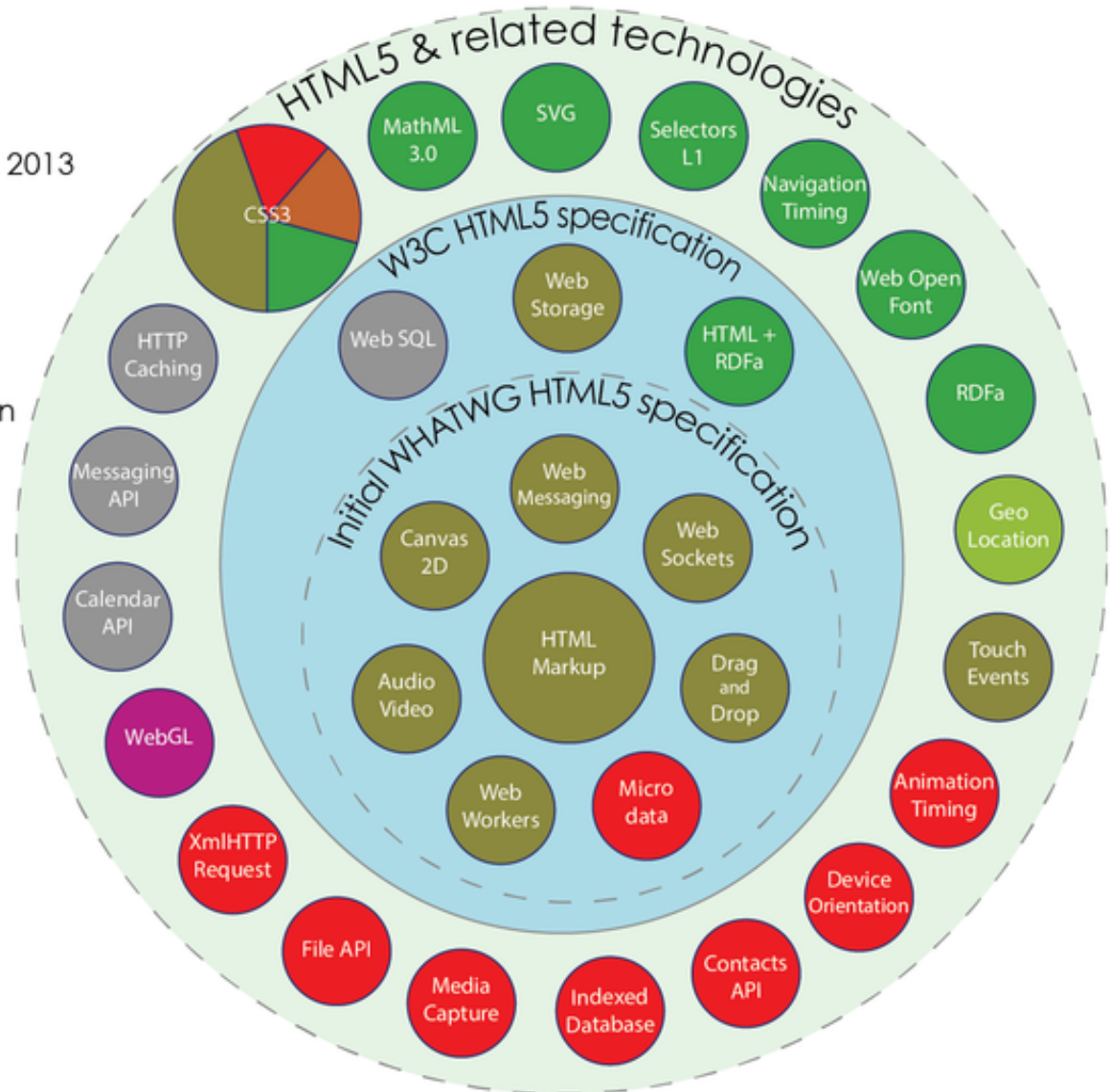
- Started by WHATWG as an alternative to XHTML, joined by W3C
 - A W3C candidate recommendation in 2012 ([draft](#))
 - WHATWG will evolve it as a “living standard”
- HTML5 \approx HTML + CSS + js
- Native support for graphics, video, audio, speech, semantic markup, ...
- Current partial support in major 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



Microdata

- The microdata effort has two parts:
 - A markup scheme
 - A set of vocabularies/ontologies
- The markup is similar to RDFa in providing ways to identify subjects, types, properties & objects
 - There's also a standard way to encode microdata as RDFa
- 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>
```

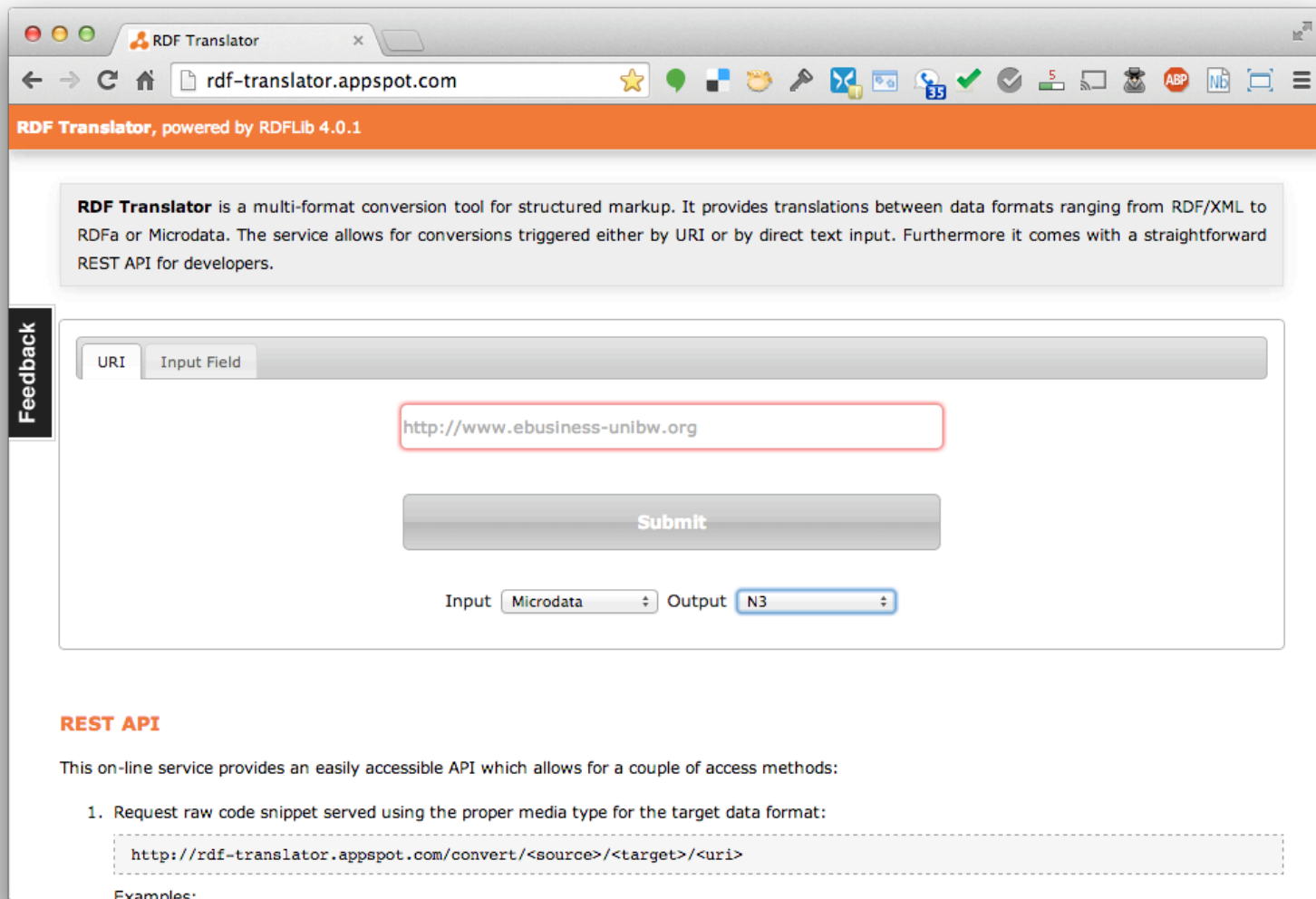
An example: itemtype

- 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>
```

Microdata <-> RDF

<http://rdf-translator.appspot.com/>



The screenshot shows a web browser window with the title "RDF Translator" and the URL "rdf-translator.appspot.com". The page has an orange header bar with the text "RDF Translator, powered by RDFLib 4.0.1". Below the header, there is a descriptive paragraph: "RDF Translator is a multi-format conversion tool for structured markup. It provides translations between data formats ranging from RDF/XML to RDFa or Microdata. The service allows for conversions triggered either by URI or by direct text input. Furthermore it comes with a straightforward REST API for developers." The main interface features a "URI" tab and an "Input Field" containing the text "http://www.ebusiness-unibw.org". A "Submit" button is located below the input field. At the bottom of the form, there are two dropdown menus: "Input" set to "Microdata" and "Output" set to "N3". A vertical "Feedback" button is on the left side of the page. Below the form, there is a section titled "REST API" with the text: "This on-line service provides an easily accessible API which allows for a couple of access methods:" followed by a numbered list: "1. Request raw code snippet served using the proper media type for the target data format:" and a code block:

```
http://rdf-translator.appspot.com/convert/<source>/<target>/<uri>
```

 The word "Examples:" is partially visible at the bottom left.

Microdata <-> RDF

<http://rdf-translator.appspot.com/>

The screenshot shows a web browser window titled "RDF Translator" with the URL "rdf-translator.appspot.com". The page features a large text input area at the top. Below it, there are examples of supported formats: "RDFa - Microdata - RDF/XML - N3 - N-Triples - RDF/JSON - JSON-LD". A "Submit" button is centered below the examples. Underneath the button, there are two dropdown menus: "Input" set to "Microdata" and "Output" set to "N3". A "Copy To Clipboard..." button is located below the dropdowns. The main content area displays a list of RDF prefixes in a monospaced font, enclosed in a dashed box. The prefixes include hcalendar, hcard, md, rdf, rdfa, rdfs, schema, xml, and xsd. Below the prefixes, there is a line of code: "<> rdfs:usesVocabulary schema: ." and another line: "[] a schema:Movie .". At the bottom of the page, there is a "REST API" section with the text: "This on-line service provides an easily accessible API which allows for a couple of access methods:".

Feedback

Examples: [RDFa](#) - [Microdata](#) - [RDF/XML](#) - [N3](#) - [N-Triples](#) - [RDF/JSON](#) - [JSON-LD](#)

Submit

Input Output

Copy To Clipboard...

```
@prefix hcalendar: <http://microformats.org/profile/hcalendar#> .
@prefix hcard: <http://microformats.org/profile/hcard#> .
@prefix md: <http://www.w3.org/ns/md#> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfa: <http://www.w3.org/ns/rdfa#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix schema: <http://schema.org/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

<> rdfs:usesVocabulary schema: .

[ ] a schema:Movie .
```

REST API

This on-line service provides an easily accessible API which allows for a couple of access methods:

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

[] a schema:Movie .

```
<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: itemprop

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

```
[ ] a schema:Movie ;  
    schema:genre "Science fiction" ;  
    schema:name "Avatar" ;  
    schema:trailer <avatar-trailer.html> .
```

```
<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 itemscope 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>
```


An example: embedded items

- An itemprop immediately follows the value an object

```
[ ] a schema:Movie ;
  schema:director [ a schema:Person ;
    schema:birthDate "1954" ;
    schema:name "James Cameron" ] ;
  schema:genre "Science fiction" ;
  schema:name "Avatar" ;
  schema:trailer <avatar-trailer.html> .
```

```
<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](#)
- 548 classes, 711 properties (5/4/14)
- **Data types:** Boolean, Date, DateTime, Number (Float, Integer) Text (URL), Time
- **Objects:** Rooted at Thing with two 'metaclasses' (Class and Property) and eight subclasses

Data Type

Boolean

Date

DateTime

Number

Float

Integer

Text

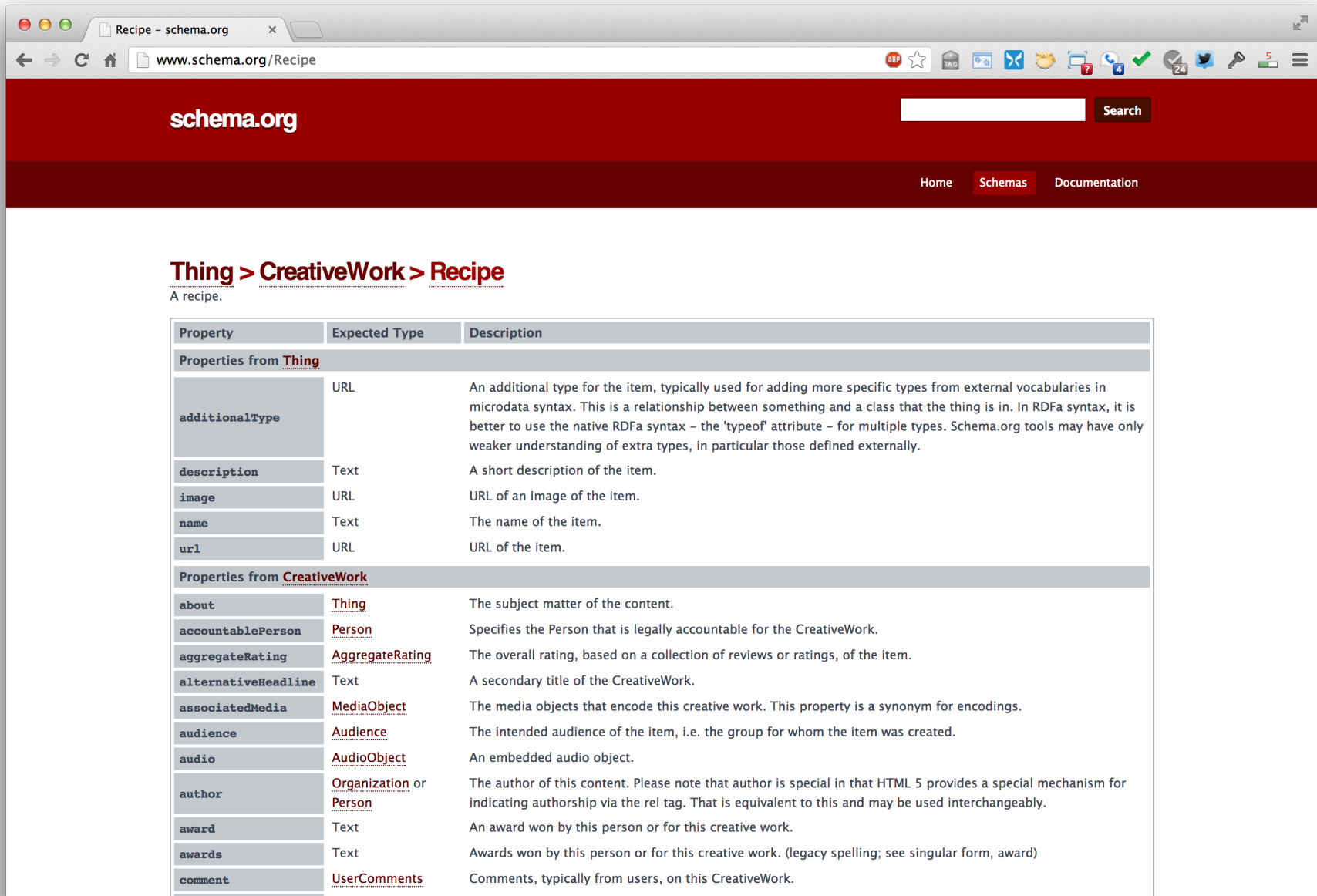
URL

Time

More specific types

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

<http://www.schema.org/Recipe>



The screenshot shows a web browser window with the URL www.schema.org/Recipe. The page features a dark red header with the "schema.org" logo on the left, a search bar in the center, and navigation links for "Home", "Schemas", and "Documentation" on the right. The main content area is white and displays the breadcrumb "Thing > CreativeWork > Recipe" in red, followed by the text "A recipe." Below this is a table with three columns: "Property", "Expected Type", and "Description". The table is divided into two sections: "Properties from Thing" and "Properties from CreativeWork".

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

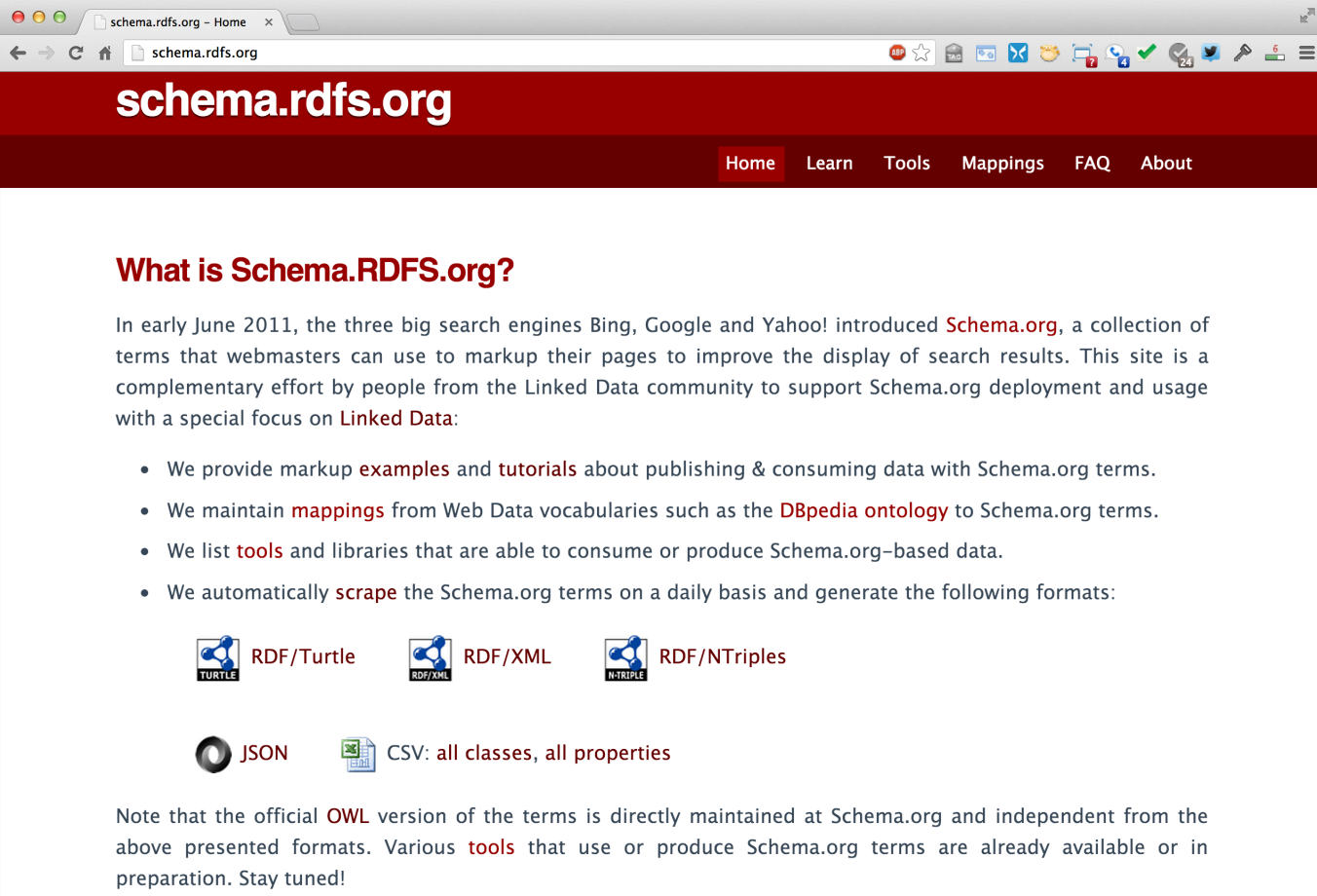
- More than RDF, less than RDFS
- Properties have an *expected* type (range)
 - Might be a string
 - A list of types, any of which are OK
- Properties attached ≥ 1 types (domain)
- Classes can have multiple parents and inherit (properties) from all of them
- No axioms (e.g., disjointness, cardinality, etc.)
- No subPropertyOf like relation

Mixing vocabularies

- Microdata is intended to work with just 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
- Schema.rdfs.org defines mappings between schema.org and popular RDF ontologies

Schema <-> RDF

<http://schema.rdf.org>








The screenshot shows a web browser window with the address bar displaying "schema.rdf.org". The page has a dark red header with the site name "schema.rdf.org" in white. Below the header is a navigation menu with links for "Home", "Learn", "Tools", "Mappings", "FAQ", and "About". The main content area features a section titled "What is Schema.RDFS.org?" followed by a paragraph explaining the site's purpose and a bulleted list of services. At the bottom, there are icons and labels for various data formats: RDF/Turtle, RDF/XML, RDF/NTriples, JSON, and CSV.

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:

 RDF/Turtle  RDF/XML  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!

Extending the schema.org ontology

- <http://www.schema.org/docs/extension.html>
- You can subclass existing classes
 - Person/Engineer
 - Person/Engineer/ElectricalEngineer
- Subclass existing 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

Serialization

- Schema.org has a [data model](#) and serializations
 - Microdata is the original, native serialization
 - RDFa is more expressive and works with the RDF stack
 - Everyone agrees that *RDFa Lite* is a good encoding: as simple as Microdata but more expressive
 - JSON-LD is also an accepted encoding
- Search engines look for Microdata and RDFa encodings and are beginning to look for JSON-LD
- Schema.org considers RDFa to be the “canonical machine representation of schema.org”

Conclusions

- Microdata is a good effort by the search companies to use a simple semantic language
- The semantics is pragmatic
 - e.g., expected types: A string is accepted where a thing is expected – “some data is better than none”
- The real value is in
 - the supported vocabularies and
 - their use by Search companies
- => Immediate motivation for using semantic markup