



RDF 教程

RDF(资源描述框架)是描述网络资源的 W3C 标准, , 比如网页的标题、作者、修改日期、内容以及版权信息等。
现在开始学习 **RDF(资源描述框架)** !

RDF 文档实例

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:si="http://www.runoob.com/rdf/">

  <rdf:Description rdf:about="http://www.runoob.com">
    <si:title>RUNOOB</si:title>
    <si:author>Jan Egil Refsnes</si:author>
  </rdf:Description>

</rdf:RDF>
```

[点我分享笔记](#)

反馈/建议



RDF 简介

资源描述框架（RDF）是用于描述网络资源的 W3C 标准, 比如网页的标题、作者、修改日期、内容以及版权信息。

您应当具备的基础知识

在继续学习之前, 您需要对下面的知识有基本的了解:

[HTML](#)

XHTML

XML

XML 命名空间

如果您希望首先学习这些项目，请访问我们的 [首页](#)。

什么是 RDF?

RDF 指资源描述框架 (Resource Description Framework)

RDF 是一个用于描述 Web 上的资源的框架

RDF 提供了针对数据的模型以及语法，这样独立的团体们就可以交换和使用它

RDF 被设计为可被计算机阅读和理解

RDF 被设计的目的不是为了向人们显示出来

RDF 使用 XML 编写

RDF 是 W3C 语义网络活动的组成部分

RDF 是一个 W3C 推荐标准

RDF - 应用举例

描述购物项目的属性，比如价格以及可用性

描述 Web 事件的时间表

描述有关网页的信息，比如内容、作者以及被创建和修改的日期

描述网络图片的内容和等级

描述针对搜索引擎的内容

描述电子图书馆

RDF 被设计为可被计算机读取

RDF 被设计为提供一种描述信息的通用方法，这样就可以被计算机应用程序读取并理解。

RDF 描述不是被设计用来在网络上显示的。

RDF 使用 XML 编写

RDF 文档使用 XML 编写。被 RDF 使用的 XML 语言被称为 RDF/XML。

通过使用 XML，RDF 信息可以轻易地在使用不同类型的操作系统和应用语言的计算机之间进行交换。

RDF 和语义网

RDF 语言是 W3C 的语义网活动的组成部分。W3C 的"语义网远景 (Semantic Web Vision)"的目标是：

Web 信息拥有确切的含义

Web 信息可被计算机理解并处理

计算机可从 Web 上整合信息

RDF 是 W3C 标准

RDF 在 2004 年 2 月成为 W3C 标准。

W3C 推荐 (标准) 被业界以及 web 团体奉为 web 标准。W3C 推荐标准 是由 W3C 工作组开发并经 W3C 成员评审的稳定规范。

可以通过此链接找到官方的 W3C 推荐标准。

<http://www.w3.org/RDF/>



RDF 规则

RDF 使用 Web 标识符 (URIs) 来标识资源。

RDF 使用属性和属性值来描述资源。

RDF 资源、属性和属性值

RDF 使用 Web 标识符来标识事物，并通过属性和属性值来描述资源。

对资源、属性和属性值的解释：

资源是可拥有 URI 的任何事物，比如 "http://www.runoob.com/rdf"

属性是拥有名称的资源，比如 "author" 或 "homepage"

属性值是某个属性的值，比如 "David" 或 "http://www.runoob.com"（请注意一个属性值可以是另外一个资源）

下面的 RDF 文档可描述资源 "http://www.runoob.com/rdf"：

```
<?xml version="1.0"?>

<RDF>
  <Description about="http://www.runoob.com/rdf">
    <author>Jan Egil Refsnes</author>
    <homepage>http://www.runoob.com</homepage>
  </Description>
</RDF>
```

☐ 上面是一个简化的例子。命名空间被忽略了。

RDF 陈述

资源、属性和属性值的组合可形成一个陈述（被称为陈述的主体、谓语和客体）。

请看一些陈述的具体例子，来加深理解：

陈述: "The author of http://www.runoob.com/rdf is David."

陈述的主体是: http://www.runoob.com/rdf

谓语是: author

客体是: David

陈述: "The homepage of http://www.runoob.com/rdf is http://www.runoob.com".

陈述的主体是: http://www.runoob.com/rdf

谓语是: homepage

客体是: http://www.runoob.com

RDF 实例

RDF 实例

这是一个 CD 列表的其中几行:

| 标题 | 艺术家 | 国家 | 公司 | 价格 | 年份 |
|------------------|--------------|-----|-------------|-------|------|
| Empire Burlesque | Bob Dylan | USA | Columbia | 10.90 | 1985 |
| Hide your heart | Bonnie Tyler | UK | CBS Records | 9.90 | 1988 |

这是一个 RDF 文档的其中几行:

```
<?xml version="1.0"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cd="http://www.recshop.fake/cd#">

  <rdf:Description
    rdf:about="http://www.recshop.fake/cd/Empire Burlesque">
    <cd:artist>Bob Dylan</cd:artist>
    <cd:country>USA</cd:country>
    <cd:company>Columbia</cd:company>
    <cd:price>10.90</cd:price>
    <cd:year>1985</cd:year>
  </rdf:Description>

  <rdf:Description
    rdf:about="http://www.recshop.fake/cd/Hide your heart">
    <cd:artist>Bonnie Tyler</cd:artist>
    <cd:country>UK</cd:country>
    <cd:company>CBS Records</cd:company>
    <cd:price>9.90</cd:price>
    <cd:year>1988</cd:year>
  </rdf:Description>
  .
  .
  .
</rdf:RDF>
```

此 RDF 文档的第一行是 XML 声明。这个 XML 声明之后是 RDF 文档的根元素: `<rdf:RDF>`。

`xmlns:rdf` 命名空间, 规定了带有前缀 `rdf` 的元素来自命名空间 `"http://www.w3.org/1999/02/22-rdf-syntax-ns#"`。

`xmlns:cd` 命名空间, 规定了带有前缀 `cd` 的元素来自命名空间 `"http://www.recshop.fake/cd#"`。

`<rdf:Description>` 元素包含了对被 `rdf:about` 属性标识的资源的描述。

元素: `<cd:artist>`、`<cd:country>`、`<cd:company>` 等是此资源的属性。

RDF 在线验证器

[W3C 的 RDF 验证服务](#)在您学习 RDF 时是很有帮助的。在此您可对 RDF 文件进行试验。

RDF 在线验证器可解析您的 RDF 文档，检查其中的语法，并为您的 RDF 文档生成表格和图形视图。

把下面这个例子拷贝粘贴到 W3C 的 RDF 验证器：

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:si="http://www.runoob.com/rdf/">
<rdf:Description rdf:about="http://www.runoob.com">
  <si:title>runoob.com</si:title>
  <si:author>Jan Egil Refsnes</si:author>
</rdf:Description>
</rdf:RDF>
```

在您对上面的例子进行解析后，结果将是类似这样的。

RDF 主要 元素

RDF 的主要元素是 **<RDF>** 以及可表示某个资源的 **<Description>** 元素。

<rdf:RDF> 元素

<rdf:RDF> 是 RDF 文档的根元素。它把 XML 文档定义为一个 RDF 文档。它也包含了对 RDF 命名空间的引用：

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
...Description goes here...
</rdf:RDF>
```

<rdf:Description> 元素

<rdf:Description> 元素可通过 **about** 属性标识一个资源。

<rdf:Description> 元素可包含描述资源的那些元素：

```
<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cd="http://www.recshop.fake/cd#">

<rdf:Description
rdf:about="http://www.recshop.fake/cd/Empire Burlesque">
  <cd:artist>Bob Dylan</cd:artist>
  <cd:country>USA</cd:country>
```

```
<cd:company>Columbia</cd:company>
<cd:price>10.90</cd:price>
<cd:year>1985</cd:year>
</rdf:Description>

</rdf:RDF>
```

`artist`、`country`、`company`、`price` 以及 `year` 这些元素被定义在命名空间 `http://www.recshop.fake/cd#` 中。此命名空间在 `RDF` 之外（并非 `RDF` 的组成部分）。`RDF` 仅仅定义了这个框架。而 `artist`、`country`、`company`、`price` 以及 `year` 这些元素必须被其他人（公司、组织或个人等）进行定义。

属性（**property**）来定义属性（**attribute**）

属性元素（**property elements**）也可作为属性（**attributes**）来被定义（取代元素）：

```
<?xml version="1.0"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cd="http://www.recshop.fake/cd#">

  <rdf:Description
    rdf:about="http://www.recshop.fake/cd/Empire Burlesque"
    cd:artist="Bob Dylan" cd:country="USA"
    cd:company="Columbia" cd:price="10.90"
    cd:year="1985" />

</rdf:RDF>
```

属性（**property**）来定义属性（**attribute**）

属性元素（**property elements**）也可作为属性（**attributes**）来被定义（取代元素）：

```
<?xml version="1.0"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cd="http://www.recshop.fake/cd#">

  <rdf:Description
    rdf:about="http://www.recshop.fake/cd/Empire Burlesque">
    <cd:artist rdf:resource="http://www.recshop.fake/cd/dylan" />
    ...
    ...
  </rdf:Description>

</rdf:RDF>
```

上面的例子中，属性 `artist` 没有值，但是却引用了一个对包含有关艺术家的信息的资源。

[☐ RDF 实例](#)

[RDF 容器 ☐](#)

[☐ 点我分享笔记](#)

反馈/建议



[☐ RDF 主要元素](#)

[RDF 集合 ☐](#)

RDF 容器元素

RDF 容器用于描述一组事物。举个例子，把某本书的作者列在一起。

下面的 **RDF** 元素用于描述这些的组：**<Bag>**、**<Seq>** 以及 **<Alt>**。

<rdf:Bag> 元素

<rdf:Bag> 元素用于描述一个规定为无序的值的列表。

<rdf:Bag> 元素可包含重复的值。

实例

```
<?xml version="1.0"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cd="http://www.recshop.fake/cd#">

  <rdf:Description
    rdf:about="http://www.recshop.fake/cd/Beatles">
    <cd:artist>
      <rdf:Bag>
        <rdf:li>John</rdf:li>
        <rdf:li>Paul</rdf:li>
        <rdf:li>George</rdf:li>
        <rdf:li>Ringo</rdf:li>
      </rdf:Bag>
    </cd:artist>
  </rdf:Description>

</rdf:RDF>
```

<rdf:Seq> 元素

<rdf:Seq> 元素用于描述一个规定为有序的值的列表（比如一个字母顺序的排序）。

<rdf:Bag> 元素可包含重复的值。

实例

```
<?xml version="1.0"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cd="http://www.recshop.fake/cd#">

  <rdf:Description
    rdf:about="http://www.recshop.fake/cd/Beatles">
    <cd:artist>
      <rdf:Seq>
        <rdf:li>George</rdf:li>
        <rdf:li>John</rdf:li>
        <rdf:li>Paul</rdf:li>
        <rdf:li>Ringo</rdf:li>
      </rdf:Seq>
    </cd:artist>
  </rdf:Description>

</rdf:RDF>
```

<rdf:Alt> 元素

<rdf:Alt> 元素用于一个可替换的值的列表（用户仅可选择这些值的其中之一）。

实例

```
<?xml version="1.0"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cd="http://www.recshop.fake/cd#">

  <rdf:Descriptio
    rdf:about="http://www.recshop.fake/cd/Beatles">
    <cd:format>
      <rdf:Alt>
        <rdf:li>CD</rdf:li>
```

```
<rdf:li>Record</rdf:li>
<rdf:li>Tape</rdf:li>
</rdf:Alt>
</cd:format>
</rdf:Description>

</rdf:RDF>
```

RDF 术语

在上面的例子中，我们在描述容器元素时已经讨论了"值的列表"。在 **RDF** 中，这些"值的列表"被称为成员（**members**）。

因此，我们可以这么说：

一个容器是一个包含事物的资源

被包含的事物被称为成员（不能称为"值的列表"）。

[☐ RDF 主要元素](#)

[RDF 集合 ☐](#)

[☐ 点我分享笔记](#)

[反馈/建议](#)



[☐ RDF 容器](#)

[RDF Schema ☐](#)

RDF 集合

RDF 集合用于描述仅包含指定成员的组。

rdf:parseType="Collection" 属性

正如在前面的章节所看到的，我们无法关闭一个容器。容器规定了所包含的资源为成员 - 它没有规定其他的成员是不被允许的。

RDF 集合用于描述仅包含指定成员的组。

集合是通过属性 **rdf:parseType="Collection"** 来描述的。

实例

```
<?xml version="1.0"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:cd="http://recshop.fake/cd#">

  <rdf:Description
    rdf:about="http://recshop.fake/cd/Beatles">
    <cd:artist rdf:parseType="Collection">
      <rdf:Description rdf:about="http://recshop.fake/cd/Beatles/George"/>
      <rdf:Description rdf:about="http://recshop.fake/cd/Beatles/John"/>
      <rdf:Description rdf:about="http://recshop.fake/cd/Beatles/Paul"/>
      <rdf:Description rdf:about="http://recshop.fake/cd/Beatles/Ringo"/>
    </cd:artist>
  </rdf:Description>

</rdf:RDF>
```


RDF Schema (RDFS)

RDF Schema (RDFS) 是对 RDF 的一种扩展。

RDF Schema 和 应用程序的类

RDF 通过类、属性和值来描述资源。

此外, RDF 还需要一种定义应用程序专业的类和属性的方法。应用程序专用的类和属性必须使用对 RDF 的扩展来定义。

RDF Schema 就是这样一种扩展。

RDF Schema (RDFS)

RDF Schema 不提供实际的应用程序专用的类和属性, 而是提供了描述应用程序专用的类和属性的框架。

RDF Schema 中的类与面向对象编程语言中的类非常相似。这就使得资源能够作为类的实例和类的子类来被定义。

RDFS 实例

下面的实例演示了 RDFS 的能力的某些方面:

```
<?xml version="1.0"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xml:base="http://www.animals.fake/animals#">

  <rdf:Description rdf:ID="animal">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
  </rdf:Description>

  <rdf:Description rdf:ID="horse">
    <rdf:type rdf:resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
    <rdfs:subClassOf rdf:resource="#animal"/>
  </rdf:Description>

</rdf:RDF>
```

在上面的例子中, 资源 "horse" 是类 "animal" 的子类。

简写的例子

由于一个 RDFS 类就是一个 RDF 资源, 我们可以通过使用 `rdfs:Class` 取代 `rdf:Description`, 并去掉 `rdf:type` 信息, 来把上面的例子简写一下:

```
<?xml version="1.0"?>

<rdf:RDF
```

```
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xml:base="http://www.animals.fake/animals#">

<rdfs:Class rdf:ID="animal" />

<rdfs:Class rdf:ID="horse">
  <rdfs:subClassOf rdf:resource="#animal"/>
</rdfs:Class>

</rdf:RDF>
```

就是这样！

[☐ RDF 集合](#)

[RDF 都柏林核心](#) ☐

[☐ 点我分享笔记](#)

反馈/建议



[☐ RDF Schema](#)

[OWL 简介](#) ☐

RDF 都柏林核心元数据倡议

都柏林核心元数据倡议 (DCMI) 已创建了一些供描述文档的预定义属性。

Dublin 核心

RDF 是元数据（关于数据的数据）。RDF 被用于描述信息资源。

都柏林核心是一套供描述文档的预定义属性。

第一份都柏林核心属性是于1995年在俄亥俄州的都柏林的元数据工作组被定义的，目前由都柏林元数据倡议来维护。

| 属性 | 定义 |
|-------------|------------------------|
| Contributor | 一个负责为资源内容作出贡献的实体(如作者)。 |
| Coverage | 资源内容的氛围或作用域 |
| Creator | 一个主要负责创建资源内容的实体。 |
| Format | 物理或数字的资源表现形式。 |
| Date | 在资源生命周期中某事件的日期。 |
| Description | 对资源内容的说明。 |
| Identifier | 一个对在给定上下文中的资源的明确引用 |
| Language | 资源智力内容所用的语言。 |
| Publisher | 一个负责使得资源内容可用的实体 |

| | |
|----------|--------------------|
| Relation | 一个对某个相关资源的引用 |
| Rights | 有关保留在资源之内和之上的权利的信息 |
| Source | 一个对作为目前资源的来源的资源引用。 |
| Subject | 一个资源内容的主题 |
| Title | 一个给资源起的名称 |
| Type | 资源内容的种类或类型。 |

通过浏览上面这个表格，我们可以发现 **RDF** 是非常适合表示都柏林核心信息的。

RDF 实例

下面的例子演示了都柏林核心属性在一个 **RDF** 文档中的使用：

```
<?xml version="1.0"?>

<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:dc= "http://purl.org/dc/elements/1.1/">

  <rdf:Description rdf:about="http://www.runoob.com">
    <dc:description>Run Noob - 奔跑吧！菜鸟</dc:description>
    <dc:publisher>Refsnes Data as</dc:publisher>
    <dc:date>2008-09-01</dc:date>
    <dc:type>Web Development</dc:type>
    <dc:format>text/html</dc:format>
    <dc:language>en</dc:language>
  </rdf:Description>

</rdf:RDF>
```



OWL 简介

OWL 是一门供处理 **web** 信息的语言。

在学习之前应具备的基础知识

在您学习 **OWL** 之前，应当对 **XML**、**XML** 命名空间以及 **RDF** 有基本的了解。

如果首先学习这些项目，请访问：

菜鸟教程的 [XML 教程](#) 和 [RDF 教程](#)。

什么是 **OWL**?

OWL 指的是 **web** 本体语言

OWL 构建在 **RDF** 的顶端之上

OWL 用于处理 **web** 上的信息

OWL 被设计为供计算机进行解释

OWL 不是被设计为供人类进行阅读的

OWL 由 **XML** 来编写

OWL 拥有三种子语言

OWL 是一项 **web** 标准

什么是本体?

本体"这个术语来自于哲学，它是研究世界上的各种实体以及他们是怎么关联的科学。

对于 **web**，本体则关于对 **web** 信息及 **web** 信息之间的关系的精确描述。

为什么使用 **OWL**?

OWL 是"语义网远景"的组成部分 - 目标是：

Web 信息拥有确切的含义

Web 信息可被计算机理解并处理

计算机可从 **Web** 上整合信息

OWL 被设计为供计算机来处理信息

OWL 被设计为提供一种通用的处理 **Web** 信息的内容的方法（而不是把它显示出来）。

OWL 被设计为由计算机应用程序来读取（而不是被人类）。

OWL 与 **RDF** 不同

OWL 与 **RDF** 有很多相似之处，但是较之 **RDF**，**OWL** 是一门具有更强机器解释能力的更强大的语言。

与 **RDF** 相比，**OWL** 拥有更大的词汇表以及更强大的语言。

OWL 子语言

OWL 有三门子语言：

OWL Lite

OWL DL (包含 OWL Lite)

OWL Full (包含 OWL DL)

OWL 使用 **XML** 编写

通过使用 **XML**，**OWL** 信息可在使用不同类型的操作系统和应用语言的不同类型的计算机之间进行交换。

OWL 是一个 **Web** 标准

OWL 于 2004 年 2 月成为一项 W3C 的推荐标准。

W3C 推荐（标准）被业界以及 **web** 团体奉为 **web** 标准。W3C推荐标准 是由 W3C 工作组开发并经 W3C 成员评审的稳定规范。

在 w3c 有关 **OWL** 的文档：<http://www.w3.org/2004/OWL/>

[首页](#) [HTML](#) [CSS](#) [JS](#) [本地书签](#)[□ OWL 简介](#)

RDF 参考手册

RDF 命名空间

RDF 命名空间(xmlns:rdf):: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

RDF 命名空间(xmlns:rdfs):: <http://www.w3.org/2000/01/rdf-schema#>

RDF 扩展名和MIME 类型

RDF 文件的推荐扩展名为**.rdf**,然而, 扩展名**.XML**是经常被用来兼容旧的**XML**解析器。

MIME 类型: **"application/rdf+xml"**。

RDFS / RDF 类

| 元素 | 类 | 子类 |
|----------------|-----------------------------------|-----------|
| rdfs:Class | All classes | |
| | | |
| rdfs:Datatype | Data types | Class |
| rdfs:Resource | All resources | Class |
| | | |
| rdfs:Container | Containers | Resource |
| rdfs:Literal | Literal values (text and numbers) | Resource |
| | | |
| rdf:List | Lists | Resource |
| rdf:Property | Properties | Resource |
| rdf:Statement | Statements | Resource |
| | | |
| rdf:Alt | Containers of alternatives | Container |
| rdf:Bag | Unordered containers | Container |
| rdf:Seq | Ordered containers | Container |

| | | |
|----------------------------------|---------------------------------|----------|
| | | |
| rdfs:ContainerMembershipProperty | Container membership properties | Property |
| rdf:XMLLiteral | XML literal values | Literal |

RDFS / RDF 属性

| 元素 | 领域 | 范围 | 描述 |
|--------------------|-----------|----------|----------------|
| rdfs:domain | Property | Class | 资源域 |
| rdfs:range | Property | Class | 资源的范围 |
| rdfs:subPropertyOf | Property | Property | 该属性是一个属性的子属性 |
| | | | |
| rdfs:subClassOf | Class | Class | 资源是一个类的子类 |
| rdfs:comment | Resource | Literal | 人类可读的资源描述 |
| rdfs:label | Resource | Literal | 人类可读的资源标签（名称） |
| rdfs:isDefinedBy | Resource | Resource | 资源的定义 |
| rdfs:seeAlso | Resource | Resource | 关于资源的其他信息 |
| rdfs:member | Resource | Resource | 资源的成员 |
| | | | |
| rdf:first | List | Resource | |
| rdf:rest | List | List | |
| rdf:subject | Statement | Resource | 一个RDF陈述的资源主体 |
| rdf:predicate | Statement | Resource | 在一个RDF陈述的资源的谓词 |
| rdf:object | Statement | Resource | 一个RDF陈述的资源客体 |
| rdf:value | Resource | Resource | value属性 |
| rdf:type | Resource | Class | 资源是一个类的实例 |

RDF 属性

| 元素 | 领域 | 范围 | 描述 |
|-----------------|----|----|--------------|
| | | | |
| rdf:about | | | 定义所描述的资源 |
| rdf:Description | | | 资源描述的容器 |
| rdf:resource | | | 定义资源，以确定一个属性 |
| rdf:datatype | | | 定义一个元素的数据类型 |
| rdf:ID | | | 定义元素的ID |

| | | | |
|---------------------|--|--|-----------|
| rdf:li | | | 定义列表 |
| rdf:_n | | | 定义一个节点 |
| rdf:nodeID | | | 定义元素节点的ID |
| rdf:parseType | | | 定义元素应如何解析 |
| rdf:RDF | | | 一个RDF文档的根 |
| xml:base | | | 定义了XML基础 |
| xml:lang | | | 定义元素内容的语言 |
| | | | |
| rdf:aboutEach | | | (删除) |
| rdf:aboutEachPrefix | | | (删除) |
| rdf:bagID | | | (删除) |

描述为"删除" 的为最近从RDF标准删除元素。

[❏ OWL 简介](#)

[❏ 点我分享笔记](#)

反馈/建议