

# 复习 第七章

---

元数据

题目

JSON：列表和字典结构

基本概念

特征

JSON结构

注意

XML：树状结构

基本概念

HTML、XML和SGML

为什么XML需要演变

XML和HTML的区别

XML结构

XML引用规则

XML元素

XML属性

XML命名空间

XML Encoding

XML验证

展示XML文件

JSON与XML对比

注意

RDF：图结构

RDF概念

RDF序列化格式

RDF空白节点

类型文字值

SUMMARY

## 元数据

- 描述、解释、定位信息资源，使得检索、使用、管理信息资源变得更加容易的结构化信息
- 常见结构：集合、列表、字典、树、图
- json和xml：对信息进行结构化的元模型、文本化地表示信息的编写系统；指定了描述的抽象结构，又指定了如何将其表示出来
- rdf：仅仅是元模型，而不是编码书写系统；rdf模型本身有多种序列化表示方式

## 题目

- 不是元数据：通过调查问卷获得的受众针对某一问题的问卷数据
- 对资源进行描述时，需要考虑：确定资源描述（元数据）的结构；确定如何对描述结构采用文本化地编码进行表示

## JSON：列表和字典结构

### 基本概念

- 轻量级的基于文本的开放标准，被设计用于人类可读的数据交换
- 使用JavaScript的语法，用于数据交换的文本格式
- 用于在网络上序列化和传输数据：Web服务器和Web应用程序之间传输数据

### 特征

- 不是编程语言
- 可扩展，轻量级，易于读写，基于文本，人类可读

### JSON结构

- 列表：数组，方括号[]

- 一组值；值之间用逗号分隔
- 一组值的有序集合：双引号中的字符串\数值\布尔值\null\对象\数组，这些结构可以嵌套

▼ 数组 JSON | 复制代码

```
1  ▼  [  
2  ▼  {  
3      "firstname":"Alex",  
4      "lastname":"ref"  
5  },  
6  ▼  {  
7      "firstname":"Black",  
8      "lastname":"Wenny"  
9  }  
10 ]
```

- 字典：对象，花括号{}
  - 标签-值；标签-值对之间逗号分隔；标签后用冒号
  - 名称/值对的无序集合

▼ 对象；字典 JSON | 复制代码

```
1  ▼  {  
2      "username":"Alex",  
3      "userage":26  
4  }
```

## 注意

1. 一种轻量级的文本数据交换格式
2. 不是只能用于JavaScript语言
3. 容易读写
4. 不是一种编程语言；不是一种超文本标记语言
5. 用于数据传输和数据交换

## XML：树状结构

# 基本概念

1. XML：扩展标记语言；一种标记语言
2. 面向互联网应用，一套定义和表示信息的结构化文档的规则
3. 标准通用语言SGML的一种严格形式
4. 用于传输数据而非显示数据
5. XML的标签不是预定义的，需要自己定义；自描述

## HTML、XML和SGML

1. SGML：一种定义文档通用标记语言的技术，始于Web之前
2. XML：SGML的一个子集；XHTML、SOAP、RDF、OWL、SKOS、RSS等等
3. HTML是SGML的一个应用

## 为什么XML需要演变

1. 简单通用的规则
2. 能够描述网络资源及其关系
3. 能够定义信息结构
4. 格式要足够形式化，便于计算机处理，同时人可读
5. 规则要足够简单，便于构建软件
6. 支持各种自然语言

## XML和HTML的区别

1. XML不是HTML的替代品而是其补充
2. 二者为不同目的而设计，HTML是关于显示信息，XML是关于传输信息
  - a. XML被设计用来传输和存储数据，重点是数据是什么
  - b. HTML被设计用来显示数据，重点是数据的外观
  - c. 在Web应用中，XML用于传输数据，HTML用于显示数据
3. HTML文档只能使用预定义的好的标签；XML允许作者自定义标签和文档结构
4. XML是一种独立于软件和硬件的各种应用程序之间最常用的信息传输工具

## XML结构

1. 第一行是XML声明，说明XML的版本和使用的编码
2. XML文档中的元素形成一个文档树，文档必须包含一个根元素，所有元素都可以有子元素
3. 所有元素都可以有文本内容和属性

## XML引用规则

语言	符号
&lt;	<
&gt;	>
&amp;	&
&apos;	'
&quot;	"

## XML元素

1. 一个XML元素包括从起始标签到闭合标签的所有内容
2. 一个元素可以包含其他元素，简单文本或者两种的混合
3. 元素可扩展
4. 元素名称可以包含字母、数字和其他字符
5. 元素名称不能以数字或标点符号开始
6. 元素名称不能以xml、XML或Xml开始
7. 元素名称不能含有空格

## XML属性

1. 属性通常提供额外信息，但并不是数据的一部分
2. 属性值必须使用单引号或双引号
3. 没有规则规定，什么时候用属性，什么时候用元素
4. 问题

- a. 属性不能包含多个值
  - b. 属性不能包含树结构
  - c. 属性不易扩展
  - d. 属性难以读取和维护
5. 将数据相关的信息作为元素，将与数据无关的信息作为属性
  6. 数据本身存储为元素，数据的数据存储为属性

#### ▼ 属性

XML | 复制代码

```
1 <note date="10/01/2008">
2   <to>Tove</to>
3   <from>Jani</from>
4   <heading>Reminder</heading>
5   <body>Don't forget me this weekend!</body>
6 </note>
```

## XML命名空间

1. XML命名空间用于在XML文档中唯一命名元素与属性
2. 提供了一种避免元素名冲突的方法
3. 在XML中使用前缀，必须定义命名空间
4. 命名空间采用“xmlns”属性在起始元素中定义

#### ▼ 命名空间

XML | 复制代码

```
1 <?xml version="1.0"encoding="ISO-8859-1"?>
2 <x:coach xmlns:x="http://www.nju.edu.cn/sports">
3   <x:name>Zhang San</name>
4   <x:age>45</age>
5   <x:gender>male</gender>
6   <x:sports>football</sports>
7 </x:coach>
```

## XML Encoding

1. just an indicator, does not make effect

2. 总是使用encoding 属性
3. 采用支持编码的编辑器
4. 在 encoding属性中使用同一编码

## XML验证

1. 具有正确XML语法的文档
  - a. 具有根元素，元素都有闭合标签，大小写敏感，元素正确嵌套，属性值采用引号
2. 符合DTD或者XML Schema 中的定义
  - a. DTD的目的是采用合法的元素定义XML的结构
  - b. XML Schema是DTD的替代品，其本身就是XML文档
  - c. 外部DTD声明： DTD文件；内部DTD声明

内部DTD声明

XML | 复制代码

```
1  <?xml version="1.0" encoding="ISO-8859-1"?>
2  <!DOCTYPE note [
3      <!ELEMENT note (to,from,heading,body)>
4      <!ELEMENT to      (#PCDATA)>
5      <!ELEMENT from      (#PCDATA)>
6      <!ELEMENT heading    (#PCDATA)>
7      <!ELEMENT body      (#PCDATA)>
8  ]>
9  <note>
10 <to>Tove</to>
11 <from>Jani</from>
12 <heading>Reminder</heading>
13 <body>Don't forget me this weekend!</body>
14 </note>
```

## 展示XML文件

1. 大多数浏览器都可显示原始XML文档，也只是展示原本的XML文档
2. XML文档中不包含如何显示数据的信息，不要期望像显示HTML网页一样显示XML文档

## JSON与XML对比

1. XML比JSON表达能力强
2. XML有时也会反复使用标签，因此JSON更简洁，XML更复杂
3. XML需要一些规范，如DTD和XSD, 来定义XML的模式（即元素与结构），JSON模式虽然也有，但是并不广泛使用
4. XML是一种标记语言，更灵活，不支持数组，更安全
5. JSON更快捷、更简单、自描述

## 注意

1. XML和HTML一样，都是超文本标记语言，都使用了树结构
2. XML元素的值可以为空
3. JSON和XML都是一种适合人类阅读和理解的数据表示格式

## RDF：图结构

### RDF概念

1. RDF代表资源描述框架, 1999年由W3C创建，是对元数据进行编码的标准，是语义网的基石
2. RDF是描述Web资源的通用框架，Web资源是指能够被标识的任何事物 无论是数字的、物理的、具体的或抽象的
3. RDF是领域无关的，用户可以使用RDF Schema语言定义自己所需的术语
4. 本质上RDF是一个数据模型，RDF需要具体的句法来被表示和传输
  - a. RDF抽象模型的关键思想是将信息切分成小的碎片，每个碎片具有清晰定义的语义，使得机器能够理解并能够处理
  - b. 关键组件
    - i. 陈述（三元组）
    - ii. 资源：名称必须是全局的，并采用URI标识符标识
    - iii. 属性：一种特殊的资源，描述了资源间的关系；属性名也必须是全局的，也采用URI标识
    - iv. 属性值
  - c. RDF的基本基石是“对象-属性-值”三元组，被称为一个陈述
  - d. 知识（或信息）被表示为一组陈述，每个陈述采用主-谓-宾的形式，且这一顺序不能改变
  - e. 每个陈述（三元组）表示一个事实，一组陈述表示给定的信息或知识，一组陈述被称为RDF图



f. 一个RDF图是一组RDF三元组，节点的集合就是三元组中主语和宾语的集合

#### ▼ RDF例子

XML | 复制代码

```
1  <?xml version="1.0"?>
2  <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
3      xmlns:dc="http://purl.org/dc/elements/1.1/">
4  <rdf:Description
5      rdf:about="http://www.example.com.cn/book/isbn83425566">
6      <dc:title>Information Retrieval</dc:title>
7  </rdf:Description>
8  <rdf:Description
9      rdf:about="http://www.example.com.cn/book/isbn83425566">
10     <dc:creator rdf:resource="http://www.nju.edu.cn/staff/ShenGuchao"/>
11 </rdf:Description>
12 </rdf:RDF>
```

## RDF序列化格式

1. 采用XML编码，面向机器阅读与处理
2. N3：纯文本，易于阅读，面向人类阅读
3. JSON-based：与JSON兼容，即使没有听说过语义网的程序员也能立即使用它
4. Embedded：将RDF三元组嵌入到Web页面中，主要为人读，其次为机器

## RDF空白节点

1. 一个空节点是一个没有URI表述的节点
2. 空节点不需要在特定的RDF图之外被提及
3. 在RDF图内部，使用空节点标识符表示空节点
4. RDF直接表示的仅是二元关系，对于多维关系，采用空节点将其分解为一组独立的二元关系

## 类型文字值

1. 采用类型文字值指定数据值（即文字）的数据类型
2. 一个RDF类型文字值由一个字符串及标识其特定数据类型的URI标识构成
3. RDF数据类型概念基于XML Schema数据类型的概念框架

# SUMMARY

1. RDF使用URI对资源进行声明
2. 每个RDF陈述包括一个主语、一个谓语和一个宾语
3. 一个RDF陈述可被表示为一个RDF图，其中，一个节点表示主语，一个节点表示宾语，连接弧表示谓语，从主语节点指向宾语节点

## RDF元素

1. Bag：用于描述一个无序的值列表，可能包含重复的值
2. Seq：用于描述一个有序的值列表，可能包含重复的值
3. Alt：用于描述可选值的列表(用户只能选择其中一个值)
4. `rdf:parseType="Collection"`：RDF集合被用来描述只包含指定成员的组

## 注意

1. RDF是一种抽象的数据模型，一种图模型，一种超文本标记语言
2. RDF的数据模型与其序列化表示是分开的
3. 与当前Web相互兼容
4. RDF表示方式更多样，计算机可理解，可表示数据之间的关联联系
5. 与采用关系型数据描述元数据相比，JSON、XML和RDF与平台无关，灵活性高，可扩展性好