

软件体系结构原理、方法与实践

❁ XML引入

```
<today>
<temperature>6°C ~ 7°C</temperature>
<weather>晴转阴</weather>
<weather_id>
<fa>00</fa>
<fb>01</fb>
</weather_id>
<wind>南风微风</wind>
<weak>星期二</weak>
<city>乌鲁木齐</city>
<date_y>2022年10月18日</date_y>
<dressing_index>较舒适</dressing_index>
<dressing_advice>建议穿薄外套</dressing_advice>
<dre></dre>
</today>
```

大家认为XML可用于以下哪些用途？

- ☒ A 用于数据传输
- ☒ B 用作配置文件
- ☒ C 用于存储数据
- ☐ D 用于描述体系结构

提交

✿ XML概述

概念：Extensible Markup Language 可扩展标记语言

- XML是一种元标记语言
- XML和HTML一样，都来自SGML
- XML是一套定义语义标记的规则

✿ XML概述

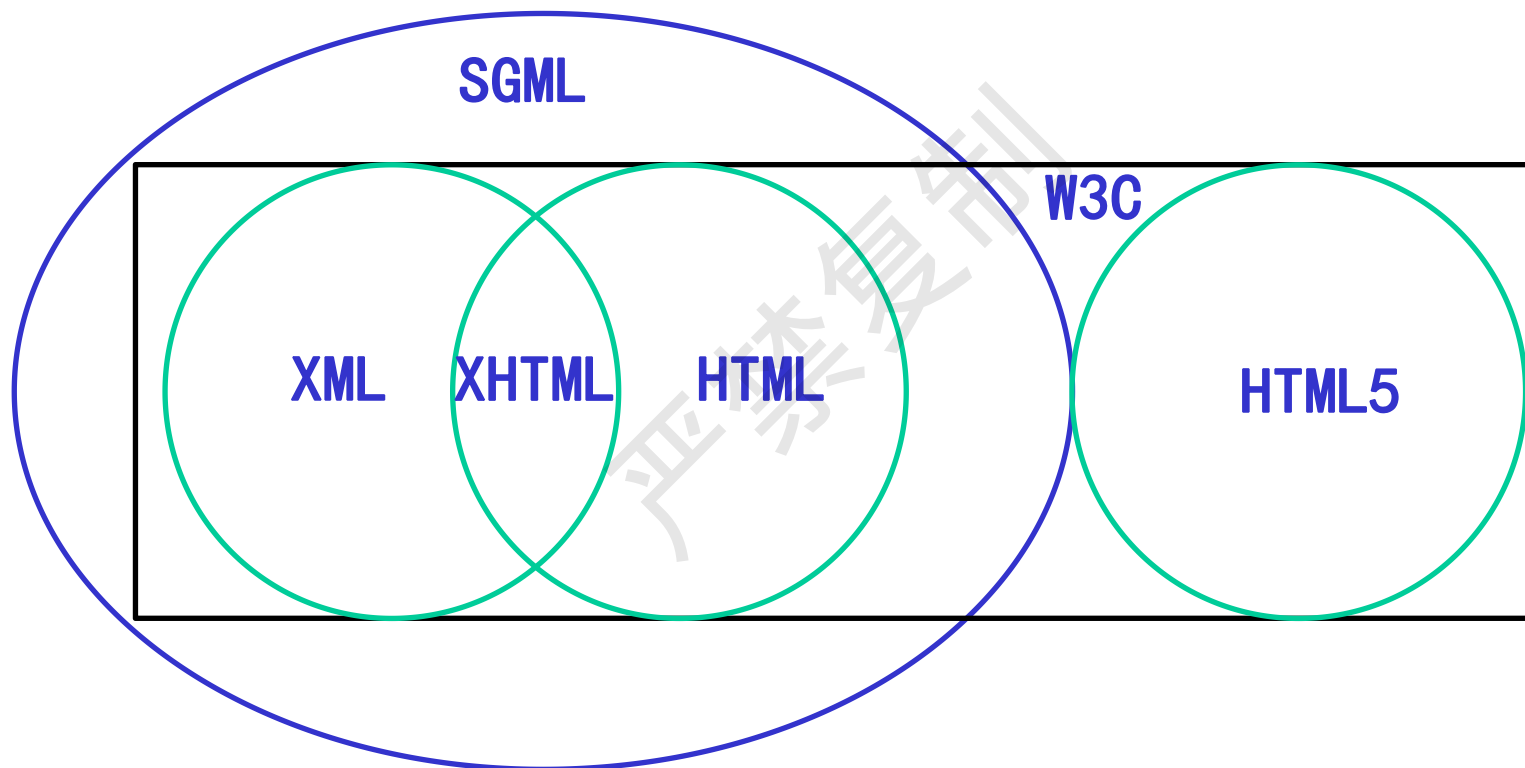
XML的功能：

- 1、数据存储：(各类配置文件)
- 2、数据传输：(各类接口)

XML与html的区别：

- 1、XML标签都是自定的，HTML标签是预定义的
- 2、XML语法严格，HTML语法松散
- 3、XML主要用于数据存储和传输，HTML用于数据显示

XML概述



XML概述

```
<!SGML "版本壹"  
  -- 范例一。 --  
  APPINFO NONE -- 没特定的信息。 --  
>
```

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/TR/xhtml1" xml:lang="en" lang="en">  
  <head>  
    <title>每个文件都必须有一个标题-w3cschool.cn</title>  
  </head>  
  <body>  
    ...你的内容在这里...  
  </body>  
</html>
```

```
<!DOCTYPE html>  
<html>  
<body>  
  
  <video width="420" controls>  
    <source src="mov_bbb.mp4" type="video/mp4">  
    <source src="mov_bbb.ogv" type="video/ogg">  
    Your browser does not support the video tag.  
  </video>  
  
</body>  
</html>
```

```
<student>  
  <name>jack</name>  
  <age>22</age>  
  <class>2022-01</class>  
</student>  
<student>  
  <name>sari</name>  
  <age>23</age>  
  <class>2022-02</class>  
</student>  
</studentList>
```


✿ XML概述

SGML 是一种用标记来描述文档资料的通用语言,它包含一系列的文档类型定义(Document Type Definition, DTD), DTD 中定义了标记的含义,因而 SGML 的语法是可以扩展的。SGML 十分庞大,既不容易学,又不容易使用,在计算机上实现也十分困难。鉴于这些因素,Web 的发明者(欧洲核子物理研究中心的研究人员)根据当时(1989 年)计算机技术的能力,提出了 HTML。

HTML 只使用 SGML 中很少一部分标记,例如,HTML 3.2 定义了 70 种标记。为了便于在计算机上实现,HTML 规定的标记是固定的,即 HTML 语法是不可扩展的,它不需要包含 DTD。HTML 这种固定的语法使它易学易用,在计算机上开发 HTML 的浏览器也十分容易。正是由于 HTML 的简单性,使 Web 技术从计算机界走向全社会,走向千家万户,Web 的发展如日中天。

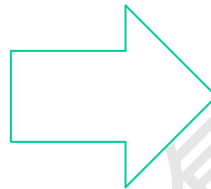
随着 Web 的应用越来越广泛和深入,人们渐渐觉得 HTML 不够用了,HTML 过于简单的语法严重地阻碍了用它来表现复杂的形式。尽管 HTML 推出了一个又一个新版本,已经有了脚本、表格、帧等表达功能,但始终满足不了不断增长的需求。另外,计算机技术的发展也十分迅速,已经可以实现比当初发明 HTML 时复杂得多的 Web 浏览器,所以开发一种新的 Web 页面语言既是必要的,也是可能的。

有人建议直接使用 SGML 作为 Web 语言,这固然能解决 HTML 遇到的困难。但是 SGML 太庞大了,用户学习和使用不方便尚且不说,要全面实现 SGML 的浏览器就非常困难。

✿ XML概述

WEB标准化组织W3C提出了

简化SGML，提供超链接能力，使用简单，文件解析，等几个改进方案，并实现以下10个目标。



- 能够直接应用在Internet上
- 能被各式应用软件使用
- 能与通用标记语言(SGML)兼容
- 能轻易发展XML相关软件
- 能简化通用标记语言(SGML)
- XML文件可读性高
- XML规范能尽完成
- XML规范必须简洁
- XML文件易于建立
- 语法不可模糊不清

以下关于XML与HTML区别说法正确是

- ☐ A HTML是为了解决XML问题而产生的
- ☒ B XML标签是可扩展的
- ☒ C HTML标签是预定义的
- ☒ D HTML语法定义是模糊的
- ☐ E XML语法定义是模糊的

提交

✿ XML的特点*

- 简洁有效
- 易学易用
- 开放的国际化标准
- 高效且可扩充

✿ XML的特点 – 简洁有效+易学易用

- 保留了GMLS的可扩展性，使XML从根本上有别于HTML
- 不再是固定标记，允许定义数量无限的标记描述文档中的资料，允许嵌套的信息结构
- HTML侧重web页面显示格式，XML侧重描述web页面的内容。
- XML中包含可扩展样式语言XSL和可扩展连接语言XLL
- XSL可解释数量无限的标记，是web页面样式更丰富多彩
- XLL支持可扩展连接和多方向连接，可不受文档制约，完全按用户要求指定和管理
- XML抛弃了SGML中不常用的功能，SGML中80%的功能不常用，XML语法说明只有30页，而SGML具有500页。
- XML设计中也考虑的易用性，编写web页面方便，实现XML浏览器也不太困难

❁ XML的特点 – 开放的国际化标准

- XML标准。这是W3C正式批准的，这意味着这个标准是稳定的，完全可用于Web和工具的开发。
- XML名域标准。用来描述名域的句法，支持能识别名域的XML解析器，如：
`<book> <title>网页研究</title> </book>`
`<job> <title>技术编辑</title> </job>`
- DOM (Document Object Model, 文档对象模型) 标准。为给结构化的数据编写脚本提供标准，这样，开发人员就能够与计算机在基于XML的数据上进行交互。
- XSL标准。XSL有两个模块：XSL转换语言和XSL格式化对象。其中转换语言可用来转换XML以满足显示要求。由于XSL的两部分是模块化的，因此，转换语言能够独立地用来进行多用途的转换，包括把XML转换成结构完整的HTML。
- (可扩展连接语言)XLL标准和XML指针语言 (XPointer) 标准。XLL提供类似与HTML的链接，但功能更强大。例如，链接可以是多方向的，可以存在于对象上而不仅仅是页面上。

新的开始，新的起点，让我们一起为梦想而努力。

✿ XML的特点 – 高效可扩充

- XML支持复用文档片段，用户可以发明和使用自定义的标签
- XML可以定义无限量的一组标注
- XML提供了一个表示格式化的体系结构
- 可通过DTD验证接收到的数据是否有效
也可以验证自己的数据是否有效

PCDATA: 被解析的字符数据

CDATA: 不被解析的文本 (< >)

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE persons[
3   <!ELEMENT persons (person+)>
4   <!ELEMENT person (name,age,contact,br?)>
5   <!ELEMENT name (#PCDATA)>
6   <!ELEMENT age (#PCDATA)>
7   <!ELEMENT contact (phone?,email?)>
8   <!ELEMENT br EMPTY>
9 ]>
10 ]>
11 <persons>
12   <person pid="p1" sex="男" qq="aaa" parent="p2">
13     <name>张小明</name>
14     <age>10</age>
15     <contact>
16       <phone>1234567</phone>
17     </contact>
18     <br/>
19   </person>
20   <person pid="p2">
21     <name>张大明</name>
22     <age>35</age>
23     <contact>
24       <email>123@qq.com</email>
25     </contact>
26   </person>
27 </persons>
```

<https://blog.csdn.net/m0>

以下关于XML说法错误的是？

- ☐ A HTML侧重web页面显示格式，XML侧重描述web页面的内容
- ☒ B XML使用了SGML中的全部功能和语法
- ☐ C XML不仅能在不同的计算机系统之间交换信息，而且能跨国界和超越不同文化疆界交换信息
- ☐ D XML支持复用文档片段，用户可以发明和使用自定义的标签

提交



✿ XML的作用

使得搜索更加有意义：数据可被XML唯一的标识。例如搜索书籍,没有XML，搜索软件必须了解每个数据库是如何构建的。有了XML，就可以很容易以标准的方式按照作者、标题、ISBN序号或其他的标准分类。搜索书就变得十分方便。

开发灵活的Web应用软件：数据一旦建立，XML能被发送到其他应用软件、对象或者中间层服务器做进一步地处理。或者它可以发送到桌面用浏览器浏览。XML和HTML、脚本、公共对象模式一起为灵活的三层Web应用软件的开发提供了所需的技术。

实现不同数据的集成：XML能够使不同来源的结构化的数据结合在一起。软件代理商可以在中间层对从后端数据库和其他应用处来的数据进行集成。然后，数据就能被发送到客户或其他服务器做进一步的集合、处理和分发。

使用于多种应用环境：XML的扩展性和灵活性允许它描述不同种类应用软件中的数据，从描述搜集的Web页到数据记录。同时，由于基于XML的数据是自我描述的，数据不需要有内部描述就能被交换和处理。

客户端数据处理与计算：XML格式的数据发送给客户后，客户可以用应用软件解析数据并对数据进行编辑和处理。如使用脚本或其他编程语言处理数据。数据计算不需要回到服务器就能进行。

新的开始，新的起点，让我们一起为梦想而努力。

✿ XML的作用

数据显示多样化：数据发到桌面后，能够用多种方式显示。XML补充了HTML，被广泛的用来描述使用者界面。HTML描述数据的外观，而XML描述数据本身。CSS和XSL为数据的显示提供了公布的机制。

局部数据更新：通过XML，数据可以局部更新。每当一部分数据变化后，不需要重发整个结构化的数据。变化的元素必须从服务器发送给客户，变化的数据不需要刷新整个使用者的界面就能够显示出来。

与现有Web发布机制相兼容：由于XML是一个开放的基于文本的格式，它可以和HTML一样使用HTTP进行传送，不需要对现存的网络进行变化。

可升级性：由于XML彻底把标识的概念同显示分开，处理者能够在结构化的数据中嵌套程序化的描述以表明如何显示数据。使得客户计算机同使用者间的交互作用尽可能的减少了，同时减少了服务器的数据交换量和浏览器的响应时间。

压缩性能高：XML压缩性能很好，因为用于描述数据结构的标签可以重复使用。XML数据是否要压缩要根据应用来定，还取决于服务器与客户间数据的传递量。XML能够使用HTTP1.1中的压缩标准。

✿ XML的应用

➤ 应用于客户需要与不同的数据源进行交互时

将来自不同数据库不同格式的数据统一转换成XML交由用户处理

➤ 应用于将大量运算负荷分布在客户端

对服务器推送的XML格式数据，用户可根据自己的需求制定不同的程序处理数据。

➤ 应用于将同一数据以不同的面貌展现给不同的用户

XML可将同一个数据按照不同形式展现出来，为网络用户界面个性化，风格化铺平道路

➤ 应用于网络代理对所取得的信息进行编辑、增减以适应个人用户的需要

将网络上的XML根据自己的需要进行过滤，组织成自己的数据库。

❁ 解析XML

假设这是在某地方发现的，经过时间冲刷而保存下来的部分XML代码，我们依然可以读懂此人的姓名，出生日期和逝世日期，但是对于有格式的电子表格或字处理程序就不行了。

```
<PERSON ID="p1100" SEX="M">
  <NAME>
    <GIVEN>Judson</GIVEN>
    <SURNAME>McDaniel</SURNAME>
  </NAME>
  <BIRTH>
    <DATE>2 Feb 1934</DATE>
  </BIRTH>
  <DEATH>
    <DATE>9 Dec 2005</DATE>
  </DEATH>
</PERSON>
```

❁ XML与HTML的区别

- HTML是一种格式化的语言，一个HTML文本可以看作一个格式化的程序；HTML中的标记是固定的，不可自定义的。
- XML是一种元标记语言；它的标签是可以自定义的。可以定义其它的标记语言。如DTD，XSL都是用XML语法描述的。
- XML定义了一套元句法，与特定领域有关的标记语言（例如，音乐扩展标记语言(MusicML)、数学标记语言(MathML)和化学标记语言等(CML))都必须遵守

编辑方程

```
<math>
  <mrow>
    <mi>y</mi>
    <mi>=</mi>
    <mi>mx</mi>
    <mi>+</mi>
    <mi>b</mi>
  </mrow>
</math>
```

$$y=mx+b$$

XML文档

```
<?xml version="1.0"?>
<visit>
  <to>alluser</to>
  <from>educity.cn</from>
  <heading>welcome</heading>
  <body>Welcome to the best online education website!</body>
</visit>
```

- (1) 文档以XML定义`<?xml version="1.0"?>`开始。
- (2) 有一个包含所有其它内容的根元素，如上面例子中的`<visit>`和`</visit>`标记符。
- (3) 所有元素必须合理地嵌套，不允许交叉嵌套。

✿ XML模式-定义XML合法模块(DTD)

DTD用于对XML中数据内容做出规定;如下图:

```
<?xml version="1.0"?>
<!DOCTYPE visit [
    <!ELEMENT visit (to,from,heading,body)>
    <!ELEMENT to (#PCDATA)>
    <!ELEMENT from (#PCDATA)>
    <!ELEMENT heading (#PCDATA)>
    <!ELEMENT body (#PCDATA)>
]>
```

! DOCTYPE visit 第二行的结点是 XML 文档中的 visit 类型;第三行说明元素 visit 有 4 个元素 to、from、heading、body;第四行定义了 to 元素的类型为“#PCDATA”;第五行定义了 from 元素的类型为“#PCDATA”,等等。

DTD 可以看作对 XML 数据的语法结构的描述。使用 DTD,不同的人就能够使用共同的 DTD 来交换数据。有专门的小程序对 XML 文档是否组织良好,以及是否符合 DTD 定义的有效性进行扫描、判断和报告。

✿ XML模式-XML

与DTD相比，XML模式的几个优势：



- XML模式使用XML语法
- XML模式支持数据类型
- XML模式是可扩展的
- XML模式有更强的表达能力

```
<!-- address.dtd -->
<!ELEMENT address (name, street, city, state, postal-code)>
<!ELEMENT name (title? first-name, last-name)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT first-name (#PCDATA)>
<!ELEMENT last-name (#PCDATA)>
<!ELEMENT street (#PCDATA)>
<!ELEMENT city (#PCDATA)>
<!ELEMENT state (#PCDATA)>
<!ELEMENT postal-code (#PCDATA)>

<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <xsd:element name="postal-code">
    <xsd:simpleType>
      <xsd:restriction base="xsd:string">
        <xsd:pattern value="[0-9]{5}(-[0-9]{4})?" />
      </xsd:restriction>
    </xsd:simpleType>
  </xsd:element>
</xsd:schema>
```

CSS

CSS 之于 HTML 文档的作用类似于 MS Word 中的“样式”的作用,可以在某种程度上把 HTML 文档中的排版格式信息与其他数据信息分离开。

一个 CSS 样式单就是一组规则(rule),样式再根据特定的一套规则级联起来。每个规则给出此规则所适用的元素的名称,以及此规则要应用于哪些元素的样式。例如,下列清单是一首诗的 CSS 样式单:

```
POEM { display: block }
```

```
TITLE { display: block; font-size: 16pt; font-weight: bold }
```

```
POET { display: block; margin-bottom: 10px }
```

```
STANZA { display: block; margin-bottom: 10px }
```

```
VERSE { display: block }
```

XSL

```
portfolio.xml
<?xml version="1.0"?>
<portfolio>
  <stock exchange="nyse">
    <name>zacx corp</name>
    <symbol>ZCXM</symbol>
    <price>28.875</price>
  </stock>
  <stock exchange="nasdaq">
    <name>zaffymat inc</name>
    <symbol>ZFFX</symbol>
    <price>92.250</price>
  </stock>
</portfolio>
```

专用于XML的样式语言，
能够筛选和排序XML中的数据对xml中的数据进行样式控制。

```
portfolio.xsl
<?xml version='1.0'?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="/">
<html>
<body>
  <table border="2" bgcolor="yellow">
    <tr>
      <th>Symbol</th>
      <th>Name</th>
      <th>Price</th>
    </tr>
    <xsl:for-each select="portfolio/stock">
      <tr>
        <td><xsl:value-of select="symbol"/></td>
        <td><xsl:value-of select="name"/></td>
        <td><xsl:value-of select="price"/></td>
      </tr>
    </xsl:for-each>
  </table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

✿ CSS与XSL的比较

- CSS只能改变特定元素的格式，也只能以元素为基础。但XSL样式单可以重新排列元素并对元素进行重排序[转换方式]。
- CSS的优越性在于具有广泛的浏览器支持。但是XSL更为灵活和强大，可更好地适用于XML文档。而且，带XSL样式单的XML文档可以很容易地转换为带CSS样式单的HTML文档。
- 如果只是要对一些固定数据进行排版，可以使用“HTML + CSS”方式；如果这些数据是与某些应用程序相关，并且独立于程序存在的，并且要独立于程序来使用，则应该充分使用XML技术，采用“HTML + XML + XSL”。

❁ API接口 – DOM

- 文档对象模型为XML文档的已解析版本定义了一组接口。解析器读入整个文档，然后构建一个驻留内存的树结构，然后代码就可以使用DOM接口来操作这个树结构。用户可以遍历树以了解原始文档包含了什么，可以删除树的几个部分，还可以重新排列树和添加新的分支等等。
- DOM提供了一组丰富的功能，用户可以用这些功能来解释和操作XML文档，但使用它们是有代价的。

❁ API接口 – DOM

- DOM 构建整个文档驻留内存的树。如果文档很大，就会要求有极大的内存。
- DOM 创建表示原始文档中每个东西的对象，包括元素、文本、属性和空格。如果用户只需关注原始文档的一小部分，那么创建那些永远不被使用的对象是极其浪费的。
- DOM 解析器必须在代码取得控制权之前读取整个文档。对于非常大的文档，这会引起显著的延迟。

❁ API接口 – SAX

SAX的几个特征解决DOM存在的问题：

- SAX 解析器向代码发送事件：解析器发现元素开始/结束，文本开始/结束时通知用户，用户决定是否保存来自这些事件的数据，如果不被保存则数据会被丢弃。
- SAX解析器根本不创建任何对象，它只是将事件传递给应用程序，开发人员决定基于哪些事件创建对象。
- SAX解析器在解析开始的时候就开始发送事件，解析器发现文档开始或文本开始时，代码会收到一个事件，应用程序立即解析生成结果，不必等到整个文档被解析完毕；如果查找文档中的某些内容，一旦找到就会抛出异常事件停止SAX解析器，然后代码用找到的数据做它需要的事情。

❁ API接口 – SAX

- **SAX事件是无状态的。**当SAX解析器在XML文档中发现文本时，它就向代码发送一个事件。该事件仅仅给用户发现的文本，它不告诉用户什么元素包含那个文本。如果想知道这一点，则用户必须自己编写状态管理代码。
- **SAX事件不是持久的。**如果应用程序需要一个数据结构来对XML文档建模，则必须自己编写那样的代码。如果需要从SAX事件访问数据，并且没有把那个数据存储在代码中，那么用户不得不再次解析该文档。

✿ API接口-DOM/SAX小结

DOM解析: DOM(Document Object Model), 也即文档对象模型, DOM 是以层次结构组织的节点或信息片断的集合。这个层次结构允许开发人员在树中寻找特定信息。分析该结构通常需要加载整个文档和构造层次结构, 然后才能做任何工作。由于它是基于信息层次的, 因而 DOM 被认为是基于树或基于对象的。

优: 整个文件都加载到内存, 可随意访问任意节点, 允许随机对文档结构增删。

缺: 必须全部解析, 占内存。

应用场景: 适用于文档较小, 需要修改文档内容的场景

SAX解析: SAX的全称是Simple APIs for XML, 也即XML简单应用程序接口。SAX处理的特点是基于事件流的。分析能够立即开始, 而不是等待所有的数据被处理。而且, 由于应用程序只是在读取数据时检查数据, 因此不需要将数据存储在内存在中, sax分析器提供的是一种对xml文档的顺序访问机制, 对于已经分析过的部分, 不能再重新倒回去处理。

优: 1、访问能够立即进行, 不需要等待所有数据被加载; 2、只在读取数据时检查数据, 不需要保存在内存中; 3、不需要将整个数据都加载到内存中, 占用内存少。

缺: 1、需要应用程序自己负责TAG的处理逻辑(例如维护父/子关系等)文档越复杂程序就越复杂; 2、单向导航, 无法定位文档层次, 很难同时访问同一文档的不同部分数据; 3、不能随机访问 xml 文档, 不支持原地修改xml

应用场景: 文档较大, 只需要读取文档数据。

❁ API接口 – JDOM

- 用DOM和SAX模型完成某些任务时的困难使Jason Hunter和Brett McLaughlin感到失望，于是他们创建了JDOM 包。JDOM是基于Java技术的开放源码项目，它试图遵循 80/20 规则：用DOM和SAX的 20%的功能来满足80%的用户需求。JDOM使用SAX和DOM解析器，因此，它是作为一组相对较小的Java类被实现的。
- JDOM 的主要特性是它极大地减少了用户必须编写的代码数量，JDOM应用程序的长度通常是DOM应用程序的三分之一，大约是SAX应用程序的一半。JDOM并不做所有的事，但对于大多数用户要做的解析，它可能正好适合用户的需求。

❁ API接口 – JAXP

- 尽管DOM、SAX和JDOM为大多数常见任务提供了标准接口，但仍有些事情是它们不能解决的。例如，在Java程序中创建DOMParser对象的过程因DOM解析器的不同而不同。为了修正这个问题，SUN发布了JAXP，该API为使用DOM、SAX和XSLT处理XML文档提供了公共接口。
- JAXP提供的诸如DocumentBuilderFactory和DocumentBuilder之类的接口，为不同的解析器提供了一个标准接口。还有一些方法可以允许用户控制底层的解析器是否可以识别名称空间，以及是否使用DTD或模式来验证XML文档。

❁ API接口 – DOM4J

DOM4J: 是一个易用, 开源的库; 它应用于Java平台, 采用了Java集合框架并完全支持DOM、SAX和JAXP。应用非常广泛,

优: 1、大量使用了Java集合类, 方便Java开发人员, 同时提供一些提高性能的替代方法; 2、支持XPath。查找节点特别快; 3、灵活性高

缺: 1、大量的使用了接口, API复杂, 理解难; 2、移植性差。

注: XPath是一门在XML文档中查找信息的语言。

✿ 常用XML解析方法

JDOM解析： JDOM 是一种使用 XML 的独特 Java 工具包，用于快速开发 XML 应用程序。它基于树型结构，利用纯JAVA的技术对XML文档实现解析、生成、序列化以及多种操作。JDOM 直接为JAVA编程服务。它利用更为强有力的JAVA语言的诸多特性（方法重载、集合概念以及映射），把SAX和DOM的功能有效地结合起来。同时JDOM也是用Java语言读、写、操作XML的新API函数，在简单和高效的前提下，这些API函数被最大限度的优化，以弥补DOM及SAX在实际应用当中的不足之处。

优： 使用具体类而不是接口，简化了DOM的API。大量使用了Java集合类，方便了Java开发人员

缺： 不能处理大于内存的文档，API 简单，没有较好的灵活性

DOM4J： 是一个易用，开源的库；它应用于Java平台，采用了Java集合框架并完全支持DOM、SAX和JAXP。应用非常广泛，

优： 1、大量使用了Java集合类，方便Java开发人员，同时提供一些提高性能的替代方法；2、支持XPath。查找节点特别快；3、灵活性高

缺： 1、大量的使用了接口，API复杂，理解难；2、移植性差。

注： XPath是一门在XML文档中查找信息的语言。

❁ API接口 – 接口的选择

- 要用Java编写应用程序吗？JAXP使用DOM、SAX和JDOM；如果用Java编写代码，那么应使用JAXP将代码与各种解析器实现的细节隔离。
- 应用程序将如何部署？如果应用程序将要作为Java applet部署，那么会希望使要下载的代码数量最小，SAX解析器比DOM解析器小，而使用JDOM时，除了SAX或DOM解析器之外还要求编写少量的代码。
- 一旦解析了XML文档，还需要多次访问那些数据吗？如果需要回过头来访问XML文件的已解析版本，DOM可能是正确的选择。而SAX事件被触发时，如果以后需要它，则由（开发人员）自己决定以某种方式保存它。如果需要访问不曾保存的事件，则必须再次解析该文件；而DOM自动保存所有的数据。
- 只需要XML源文件的少量内容吗？如果只需要XML源文件的少量内容，那么SAX可能是正确的选择。SAX不会为源文件中的每个东西创建对象。使用SAX，要检查每个事件以了解它是否与需要有关，然后相应地处理它。
- 正在一台内存很少的机器上工作吗？若是的话，不管可能考虑到的其它因素是什么，SAX都是最佳选择。

XML开发工具

JavaScript语言处理DOM的例子

```
<?xml version="1.0"?>
<visit>
  <to>alluser</to>
  <from>educity.cn</from>
  <heading>Welcome</heading>
  <body>Welcome to the best online education website!</body>
</visit>
```

支持 XML 的开发工具比较多,开发语言有脚本语言(例如,JavaScript、VBScript、Perl 等)、编程语言(例如:Java、C++、Object Pascal 等),开发环境有命令行工具(例如,ANT 等)、集成式开发平台(例如,Visual Studio .NET、VisualAge、Inprise JBuilder 等),建模工具具有 IBM Rose 等。

```
<html>
<head>
<script language="JavaScript" for="window" event="onload">
  var xmlDoc = new ActiveXObject("Microsoft.XMLDOM");
  xmlDoc.async="false";
  xmlDoc.load("visit.xml");
  nodes = xmlDoc.documentElement.childNodes;
  to.innerText = nodes.item(0).text;
  from.innerText = nodes.item(1).text;
  header.innerText = nodes.item(2).text;
  body.innerText = nodes.item(3).text;
</script>
<title>HTML using XML data</title>
</head>
<body bgcolor="yellow">
  <h1>Refsnes Data Internal Note</h1>
  <b>To: </b>
  <span id="to"> </span>
  <br>
  <b>From: </b>
  <span id="from"></span>
  <hr>
  <b><span id="header"></span></b>
  <hr>
  <span id="body"></span>
</body>
</html>
```

新的开始,新的起点,让我们一起为梦想而努力。

✿ XML建模

XML建模包括以下方面：

- 描述具体数据:这是XML的经典用途，用文本编写XML文档
- 描述数据结构和模式:用DTD描述XML中的数据结构及模式
- 描述数据的表现:用XSL定义一组元素来描述格式化数据
- 描述数据中的位置:使用XPath描述XML中标签的位置
- 描述数据中的链接关系:用XLink定义将不同资源链接在一起的方法
- 描述数据的应用关系:使用SOAP，WSDL描述有关概念的详细内容。