

《XML 原理与应用》

# 实 验 指 导 书

版 本：201401

设计者：夏 天

中国人民大学 信息资源管理学院

\_\_\_\_\_年\_\_\_\_\_月

## 目 录

实 验 指 导 书.....	1
目 录.....	2
实验 1 选择与安装 XML 编辑工具.....	3
实验 2 创建格式良好的 XML 文档.....	6
实验 3 利用 DTD 实现 XML 文档约束.....	9
实验 4 利用 XML Schema 实现 XML 文档约束.....	12
实验 5 利用 CSS 格式化 XML.....	15
实验 6 利用 XPath 进行定位.....	18
实验 7 利用 XSLT 和 CSS 对 XML 文档进行转换 .....	21
实验 8 创建基于 JavaScript 的网页时钟 .....	25
实验 9 利用 JavaScript 操纵 DOM.....	29
实验 10 利用 SVG 绘制散点图.....	33

## 实验1 选择与安装 XML 编辑工具

### 1.1 实验目的

- [1] 理解 XML 的基本特点和 XML 编辑工具的作用；
- [2] 能够自己安装主流的文本编辑工具；
- [3] 能够安装配置基本的 Java 环境，并安装 oXygen XML Editor。

### 1.2 实验准备

- [1] 下载任一文本编辑工具到本地计算机，如：Notepad++、Sublime Text 3；
- [2] 下载 oXygen XML Editor 到本地计算机；
- [3] 下载最新版本的 Java (J2SDK)，请根据个人操作系统的情况，选择 32 位或 64 位 J2SDK。

### 1.3 实验要求

- [1] 在系统环境变量中增加 JAVA\_HOME，其值设为 Java 的安装目录；
- [2] 在系统环境变量中能够把 java 可执行文件的目录加入到 path 变量中；
- [3] 对于非 Windows 操作系统，需要达到要求[1]和[2]，并找到或安装支持语法高亮显示的编辑器。

### 1.4 预期结果

- [1] 能够在命令行中任意目录下，输出 `java -version`，显示正常 Java 版本信息；
- [2] 能够使用编辑工具打开 XML 文档，支持语法高亮；
- [3] 能够成功运行 oXygen XML Editor。

### 1.5 实验步骤

说明：以下实验步骤以 WIN7 操作系统为例，其他操作系统可在满足实验要求和预期结果的前提下，调整实验步骤。

- [1] 安装测试文本编辑工具  
双击下载的文本编辑工具，如 Sublime Text 3，根据提示进行安装，能够成功运行。
- [2] 通过文本编辑工具编辑 XML 文档，保存为"E1.xml"，在编辑过程中，能够支

持语法高亮，文字内容如下：

1	<?xml version="1.0" encoding="UTF-8"?>
2	
3	<book isbn="978-7-115-28282-8" id="b2">
4	<title lang="CHN">数学之美</title>
5	<price>45.00</price>
6	<authors>
7	<author>吴军</author>
8	</authors>
9	<press>人民邮电出版社</press>
10	<pages>304</pages>
11	<description>读了“数学之美”，才发现大学时学的数学知识，比如马尔科夫链、矩阵计
12	算，甚至余弦函数原来都如此亲切，并且栩栩如生，才发现自然语言和信息处理这么有趣，
13	才真正明白“数学是科学的皇后”这句名言。</description>
14	<cover>book-math.jpg</cover>
15	</book>

### [3] 安装 Java 虚拟机

(A) 假设下载的 J2SDK 文件为"jdk-7u45-windows-i586"，双击安装，默认安装目录为"C:\Program Files\Java\jdk1.7.0\_45"。

(B) 在 Windows 中依次进入控制面板→系统和安全→系统→高级系统设置，在系统属性对话框中的"高级"标签页中，点击底部的“环境变量”按钮，弹出“环境变量”对话框。

(C) 点击在“用户环境变量”下面的“新建”按钮，变量名称输入"JAVA\_HOME"，不包括引号，下同；变量值输入"C:\Program Files\Java\jdk1.7.0\_45"。

(D) 双击“用户环境变量”中已有的 Path 变量，如没有 Path 变量，则按照上一步骤新建 Path 变量，在 Path 变量的变量值的最前面，加入"%JAVA\_HOME%\bin;"。

(E) 点击“确定”按钮，关闭系统属性对话框。

[4] 在开始菜单→运行对话框中输入 cmd，进入命令行窗口，在该窗口中输入："java -version"，正常输出如下内容：

```
java version "1.7.0_45"
Java(TM) SE Runtime Environment (build 1.7.0_45-b18)
Java HotSpot(TM) Client VM (build 24.45-b08, mixed mode, sharing)
```

### [5] 安装使用 oXygen XML Editor

双击下载的 oXygen XML Editor 安装文件，根据提示安装，安装运行后，通过 File 菜单中的 Open 菜单项，选择新建的"E1.xml"，能够成功打开该文件，并且文件中没有红色的波浪底线错误提示。

## 1.6 思考分析

- [1] "%JAVA\_HOME%"代表什么含义?
- [2] 编辑 XML 文档时, 要注意中英文标点符合的差异, 防止因中文标点符号导致语法异常。

RUC IRM 复习

## 实验2 创建格式良好的 XML 文档

### 2.1 实验目的

- [1] 熟悉 XML 的基本语法;
- [2] 掌握 XML 的内置实体;
- [3] 掌握 CDATA 的使用;
- [4] 掌握 XML 格式良好与否的验证方法。

### 2.2 实验准备

- [1] 计算机中已安装好 XML 编辑工具;
- [2] 第一个 XML 文档"E2-1.xml"的源代码清单:

1	<?xml version="1.0" encoding="UTF-8"?>
2	
3	<blacklist>
4	<item>
5	<code>001</code>
6	<content>
7	<!-- 待嵌入的文档内容 -->
8	</content>
9	</item>
10	</blacklist>

- [3] 第二个 XML 文档"E2-2.xml"的源代码清单:

1	<?xml version="1.0" encoding="UTF-8"?>
2	
3	<contact>
4	<name>段玉</name>
5	<age>30</age>
6	<cellphone>123-4567-8901</cellphone>
7	<address>大理</address>
8	<spouse>
9	<name>王语嫣</name>
10	<age>28</age>
11	</spouse>
12	</contact>

## 2.3 实验要求

- [1] 根据以上代码清单，输入 XML 文档内容，并保存为"E2-1.xml"和"E2-2.xml"；
- [2] 在"E2-1.xml"的 content 部分，把"E2-2.xml"的内容以文本方式嵌入，"E2-2.xml"的尖括号、引号等作为普通的字符嵌入。

## 2.4 预期结果

- [1] 在浏览器中打开"E2-1.xml"，嵌入的内容被当作普通文本而非标记。以 Firefox 为例，"E2-1.xml"的显示效果如下：

```
-<blacklist>
- <item>
  <code>001</code>
  -<content>
    <!-- 待嵌入的文档内容 -->
    <?xml version="1.0" encoding="UTF-8"?> <contact> <name>段玉</name> <age>30</age>
    <cellphone>123-4567-8901</cellphone> <address>大理</address> <spouse> <name>王语
    嫣</name> <age>28</age> </spouse> </contact>
  </content>
</item>
</blacklist>
```

## 2.5 实验步骤

- [1] 通过 XML 编辑工具，如 oXygen XML Editor，根据实验准备中的数据，输入"E2-1.xml"；
- [2] 输入"E2-2.xml"；
- [3] 修改"E2-1.xml"，内容更改如下：

1	<?xml version="1.0" encoding="UTF-8"?>
2	
3	<blacklist>
4	<item>
5	<code>001</code>
6	<content>
7	<!-- 待嵌入的文档位置 -->
8	<![CDATA[
9	<?xml version="1.0" encoding="UTF-8"?>
10	<contact>
11	<name>段玉</name>
12	<age>30</age>
13	<cellphone>123-4567-8901</cellphone>
14	<address>大理</address>
15	<spouse>
16	<name>王语嫣</name>
17	<age>28</age>

18	<code>&lt;/spouse&gt;</code>
19	<code>&lt;/contact&gt;</code>
20	<code>]]&gt;</code>
21	<code>&lt;/content&gt;</code>
22	<code>&lt;/item&gt;</code>
23	<code>&lt;/blacklist&gt;</code>

- [4] 通过浏览器打开文件"E2-1.xml", 查看在浏览器中的显示效果, 并进行元素的折叠、展开操作, 理解 XML 的元素嵌套概念。

## 2.6 思考分析

- [1] "CDATA"在 XML 文档中的作用是什么? 如果不用 CDATA, 如何把"E2-2.xml"嵌入到"E2-1.xml"中?
- [2] XML 是否允许元素的相互交叉? 请举例并借助于 XML 专业工具进行测试。



## 实验3 利用 DTD 实现 XML 文档约束

### 3.1 实验目的

- [1] 了解 DTD 的作用；
- [2] 掌握 DTD 元素、属性、实体声明的基本语法；
- [3] 理解 XML 文档的有效性概念；
- [4] 学习 XML 专业工具的 DTD 功能。

### 3.2 实验准备

- [1] 给定原始图书文档内容，保存在"E3-book.xml"中，代码清单如下：

```
1  <?xml version="1.0" encoding="UTF-8"?>
2
3  <book isbn="978-7-115-28282-8" id="b2">
4      <title lang="CHN">数学之美</title>
5      <price>45.00</price>
6      <authors>
7          <author>吴军</author>
8      </authors>
9      <press>人民邮电出版社</press>
10     <pages>304</pages>
11     <description>读了“数学之美”，才发现大学时学的数学知识，比如马尔科夫链、矩阵计
12 算，甚至余弦函数原来都如此亲切，并且栩栩如生，才发现自然语言和信息处理这么有趣，
13 才真正明白“数学是科学的皇后”这句名言。</description>
14     <cover>book-math.jpg</cover>
15 </book>
```

### 3.3 实验要求

- [1] 根据"E3-book.xml"，编写对应的 DTD 文档"E3-book.dtd"；
- [2] book 元素的 id 为 ID 类型；
- [3] 定义实体 press，表示“人民邮电出版社”；
- [4] 修改"E3-book.xml"，加入与"E3-book.dtd"的关联，并把 press 的内容用自定义的 press 实体替代；
- [5] 修改"E3-book.xml"，用英文的大小于号为 title 加上书名号，即变为<<数学之

美>>;

[6] 保证修改后的"E3-book.xml"能够通过"E3-book.dtd"的有效性验证。

### 3.4 预期结果

- [1] "E3-book.xml"文档中的书名号进行了自定义实体转义;
- [2] "E3-book.xml"文档中的 `press` 元素的内容采用"`&press;`"表示;
- [3] 借助于 XML 专业工具, 如 oXygen XML Editor, 验证"E3-book.xml", 使其通过有效性验证。

### 3.5 实验步骤

[1] 新建"E3-book.dtd", 内容如下:

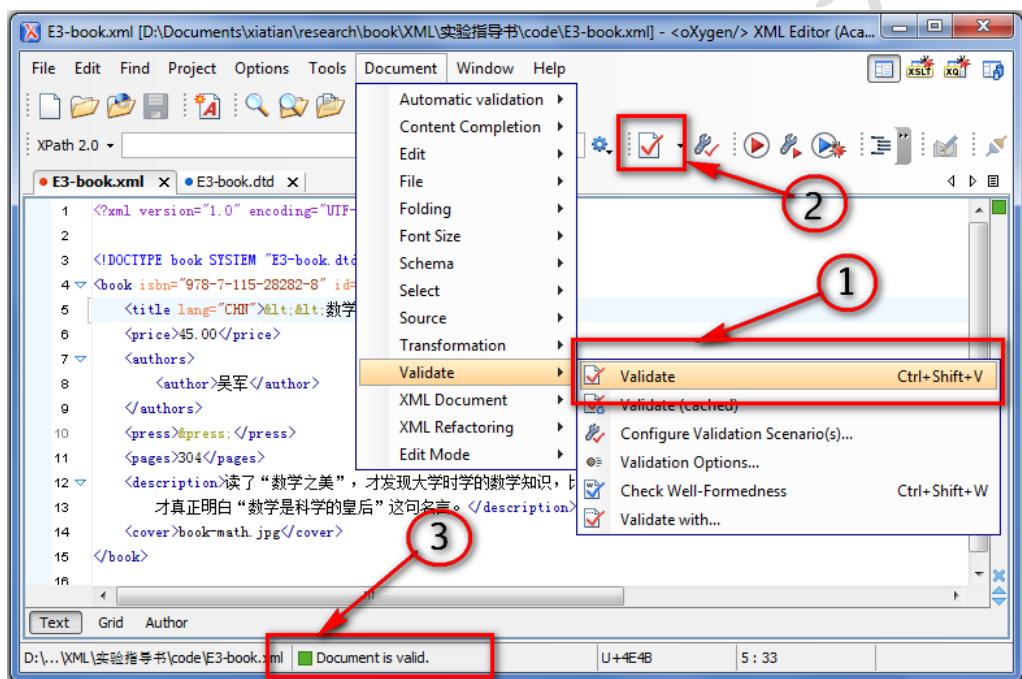
1	<?xml encoding="UTF-8"?>
2	
3	<!ELEMENT book (title,price,authors,press,pages,description,cover)>
4	<!ATTLIST book
5	id ID #REQUIRED
6	isbn CDATA #REQUIRED>
7	
8	<!ELEMENT title (#PCDATA)>
9	<!ATTLIST title
10	lang CDATA #REQUIRED>
11	
12	<!ELEMENT price (#PCDATA)>
13	<!ELEMENT authors (author)>
14	<!ELEMENT press (#PCDATA)>
15	<!ELEMENT pages (#PCDATA)>
16	<!ELEMENT description (#PCDATA)>
17	<!ELEMENT cover (#PCDATA)>
18	<!ELEMENT author (#PCDATA)>
19	
20	<!ENTITY press "人民邮电出版社">

[2] 修改"E3-book.xml", 内容如下:

1	<?xml version="1.0" encoding="UTF-8"?>
2	
3	<!DOCTYPE book SYSTEM "E3-book.dtd">
4	<book isbn="978-7-115-28282-8" id="b2">
5	<title lang="CHN">&lt;&lt;数学之美&gt;&gt;</title>
6	<price>45.00</price>
7	<authors>
8	<author>吴军</author>

9	</authors>
10	<press>&press;</press>
11	<pages>304</pages>
12	<description>读了“数学之美”，才发现大学时学的数学知识，比如马尔科夫链、矩阵计
13	算，甚至余弦函数原来都如此亲切，并且栩栩如生，才发现自然语言和信息处理这么有趣，
14	才真正明白“数学是科学的皇后”这句名言。</description>
15	<cover>book-math.jpg</cover>
16	</book>

- [3] 以 oXygen XML Editor 为例，点击菜单 Document→Validate→Validate，如下图中的位置 1，或者直接点击 Validate 工具栏按钮，如下图中的位置 2，根据提示确定后，文档能够通过有效性验证，给出"Document is Valid"的提示，如下图中的位置 3 所示。



### 3.6 思考分析

- [1] 引入 DTD 的目的是什么？
- [2] 如果把本实验的 DTD 直接嵌入到 XML 文档中，如何编写？并思考内外部 DTD 各有什么优缺点？

## 实验4 利用 XML Schema 实现 XML 文档约束

### 4.1 实验目的

- [1] 了解 XML Schema 的特点
- [2] 掌握 XMLSchema 元素、属性的作用及使用方式
- [3] 掌握 XML Schema 的数据类型
- [4] 学习 XML 专业工具所提供的 XML Schema 相关功能

### 4.2 实验准备

- [1] 给定原始图书文档内容，保存在"E4-book.xml"中，代码清单如下：

1	<?xml version="1.0" encoding="UTF-8"?>
2	
3	<book isbn="978-7-115-28282-8" id="b2">
4	<title lang="CHN">数学之美</title>
5	<price>45.00</price>
6	<authors>
7	<author>吴军</author>
8	</authors>
9	<press>人民邮电出版社</press>
10	<pages>304</pages>
11	<description>读了“数学之美”，才发现大学时学的数学知识，比如马尔科夫链、矩阵计
12	算，甚至余弦函数原来都如此亲切，并且栩栩如生，才发现自然语言和信息处理这么有趣，
13	才真正明白“数学是科学的皇后”这句名言。</description>
14	<cover>book-math.jpg</cover>
15	</book>

### 4.3 实验要求

- [1] 根据"E4-book.xml"，编写对应的 XML Schema 文档"E4-book.xsd"；
- [2] book 元素的 id 属性为 ID 类型；isbn 属性满足正则表达式的限制；二者必须都出现；
- [3] authors 中 author 元素的数量最少为一个，最多不受限制；
- [4] price 为 decimal 类型；
- [5] pages 为 integer 类型；

[6] 在"E4-book.xml"中加入与"E4-book.xsd"的关联，并能通过有效性验证。

#### 4.4 预期结果

- [1] "E4-book.xml"和"E4-book.xsd"满足实验要求;
- [2] 借助于 XML 专业工具, 如 oXygen XML Editor, 验证"E4-book.xml", 使其通过有效性验证。

#### 4.5 实验步骤

[1] 新建"E4-book.xsd", 内容如下:

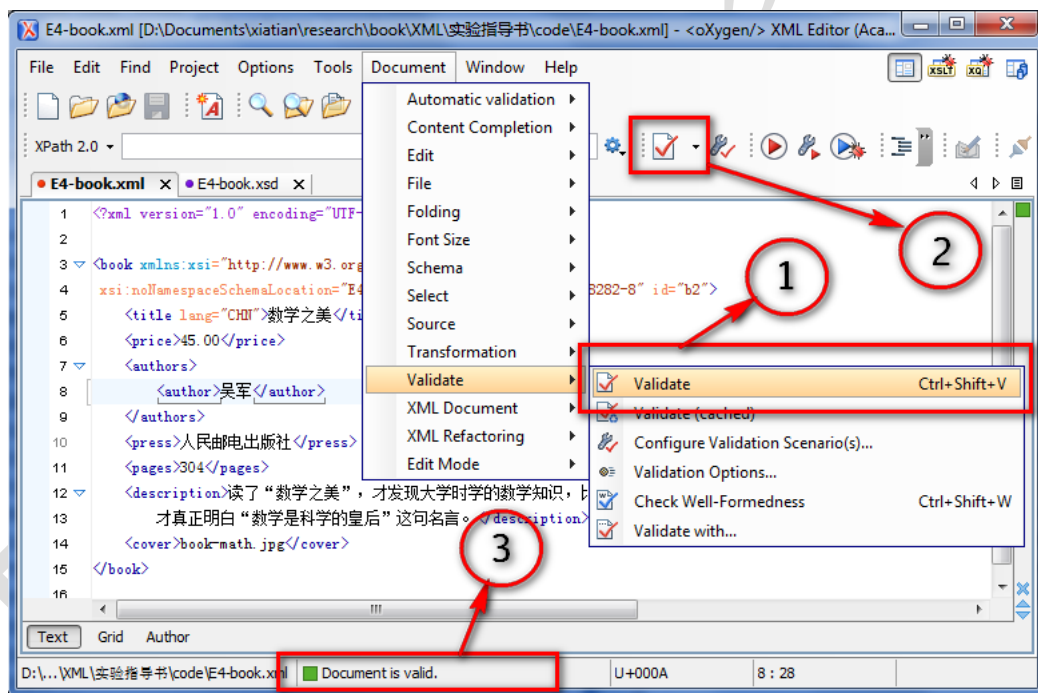
1	<?xml version="1.0" encoding="UTF-8"?>
2	<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
3	version="1.1">
4	<xs:element name="book">
5	<xs:complexType>
6	<xs:sequence>
7	<xs:element ref="title"/>
8	<xs:element name="price" type="xs:decimal"/>
9	<xs:element ref="authors"/>
10	<xs:element name="press" type="xs:string"/>
11	<xs:element name="pages" type="xs:integer"/>
12	<xs:element name="description" type="xs:string"/>
13	<xs:element name="cover" type="xs:string"/>
14	</xs:sequence>
15	<xs:attribute name="id" use="required" type="xs:ID"/>
16	<xs:attribute name="isbn" use="required">
17	<xs:simpleType>
18	<xs:restriction base="xs:string">
19	<xs:pattern value="\d{3}\-\d\-\d{3}\-\d{5}\-\d"/>
20	</xs:restriction>
21	</xs:simpleType>
22	</xs:attribute>
23	</xs:complexType>
24	</xs:element>
25	
26	<xs:element name="title">
27	<xs:complexType mixed="true">
28	<xs:attribute name="lang" use="required" type="xs:string"/>
29	</xs:complexType>
30	</xs:element>
31	
32	<xs:element name="authors">

33	<xs:complexType>
34	<xs:sequence minOccurs="1" maxOccurs="unbounded">
35	<xs:element name="author" type="xs:string"/>
36	</xs:sequence>
37	</xs:complexType>
38	</xs:element>
39	
40	</xs:schema>

- [2] 修改"E4-book.xml", 加入与"E4-book.xsd"的关联, 把 book 元素更改为如下形式:

```
<book xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:noNamespaceSchemaLocation="E4-book.xsd"
      isbn="978-7-115-28282-8" id="b2">
```

- [3] 以 oXygen XML Editor 为例, 点击菜单 Document→Validate→Validate, 如下图中的位置 1, 或者直接点击 Validate 工具栏按钮, 如下图中的位置 2, 根据提示确定后, 文档能够通过有效性验证, 给出"Document is Valid"的提示, 如下图中的位置 3 所示。



## 4.6 思考分析

- [1] 结合 DTD 实验部分, 思考 XML Schema 与 DTD 的显著差异有哪些?
- [2] 借助于工具, 可以直接从 XML 文档生成对应的 XML Schema, 请尝试并分析自动生成的模式文档是否能够满足实验要求?

## 实验5 利用 CSS 格式化 XML

### 5.1 实验目的

- [1] 掌握 CSS 与 XML 的关联方式
- [2] 掌握 CSS 的常见属性定义方式
- [3] 熟悉利用编辑工具实现对 CSS 的手工编写

### 5.2 实验准备

- [1] 给定原始图书文档内容，保存在"E5-book.xml"中，代码清单如下：

1	<?xml version="1.0" encoding="UTF-8"?>
2	
3	<book isbn="978-7-115-28282-8" id="b2">
4	<title lang="CHN">数学之美</title>
5	<price>45.00</price>
6	<authors>
7	<author>吴军</author>
8	</authors>
9	<press>人民邮电出版社</press>
10	<pages>304</pages>
11	<description>读了“数学之美”，才发现大学时学的数学知识，比如马尔科夫链、矩阵计
12	算，甚至余弦函数原来都如此亲切，并且栩栩如生，才发现自然语言和信息处理这么有趣，
13	才真正明白“数学是科学的皇后”这句名言。</description>
14	<cover>book-math.jpg</cover>
15	</book>

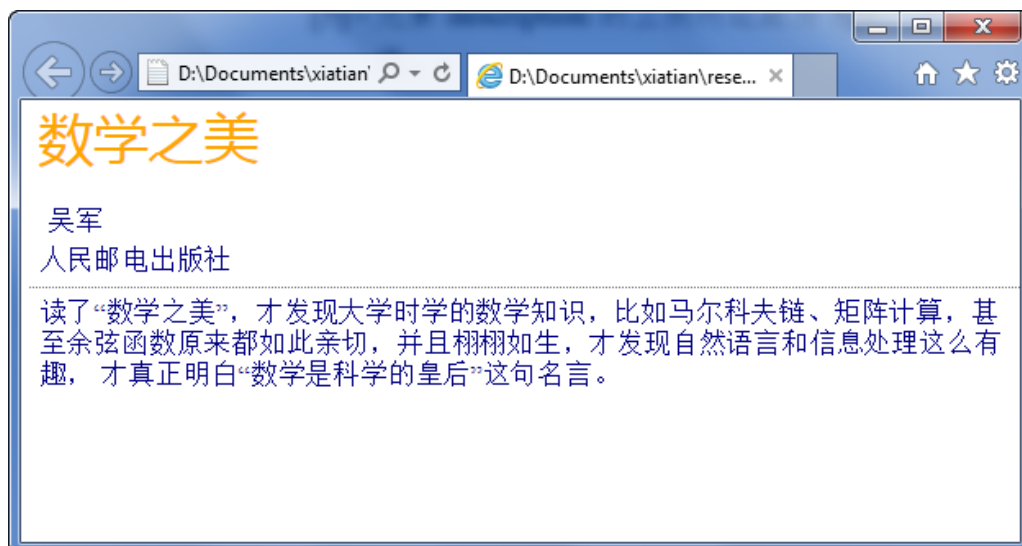
### 5.3 实验要求

- [1] 编写样式文件"E5-book.css"，与"E5-book.xml"进行关联以对其呈现样式进行控制；
- [2] 每个元素默认占据一行，字体缺省为微软雅黑、海军蓝色 navy，下部外边距为 5px，左侧内边距为 5px；
- [3] 元素 title 的字体大小为 24pt，下部外边距为 20px，颜色为橙色 orange；
- [4] 隐藏 price、pages 和 cover 元素；
- [5] 元素 description 的上侧内边距为 5px，上边框为 1 个像素宽的灰色（gray）虚

线。

## 5.4 预期结果

[1] "E5-book.xml"在浏览器中的呈现效果如下：

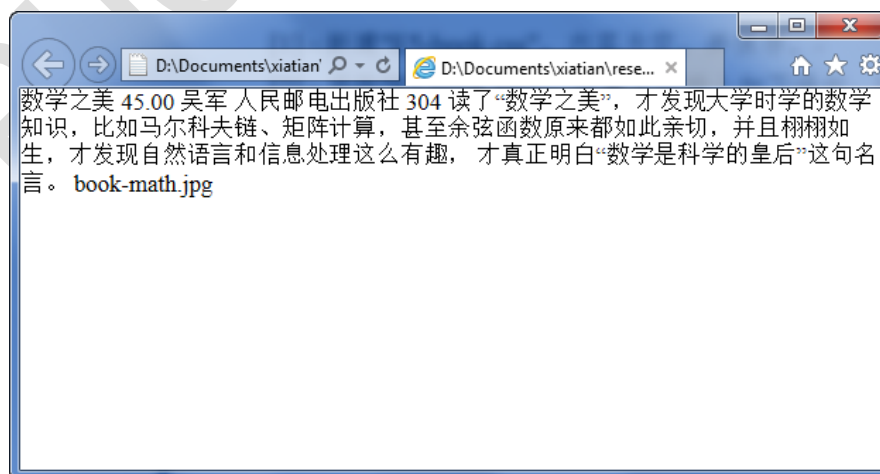


## 5.5 实验步骤

- [1] 新建"E5-book.css", 内容为空, 并保存。
- [2] 修改"E5-book.xml", 在第 2 行插入如下语句, 实现与"E5-book.css"样式文件的关联:

```
<?xml-stylesheet type="text/css" href="E5-book.css"?>
```

- [3] 通过浏览器打开"E5-book.xml"查看效果, 以 IE10.0 为例, 默认效果如下:



- [4] 修改"E5-book.css", 使其内容如下:

1	* {
2	display: block;



3	color:navy;
4	margin-bottom:5px;
5	padding-left:5px;}
6	title {
7	font-family:"微软雅黑";
8	font-size:24pt;
9	margin-bottom:20px;
10	color:orange;
11	}
12	price, pages, cover {
13	display:none;
14	}
15	description {
16	padding-top:5px;
17	border-top:1px dotted gray;
18	}

[5] 在浏览器中打开"E5-book.xml"，展示效果与预期效果图保持一致。

## 5.6 思考分析

- [1] 能否根据 XML 文档的 cover 元素，通过 CSS 在展示时显示图书的封面图片？
- [2] 能否通过 CSS 调整元素的显示顺序？
- [3] 请考虑 CSS 在格式化 XML 时，存在哪些局限性？

## 实验6 利用 XPath 进行定位

### 6.1 实验目的

- [1] 理解 XPath 定位 XML 文档的基本原理
- [2] 掌握 XPath 的定位路径表达式
- [3] 了解 XPath 的函数功能

### 6.2 实验准备

- [1] 给定原始图书文档内容，保存在"E6-books.xml"中，代码清单如下：

```
1  <?xml version="1.0"?>
2
3  <books>
4      <book isbn="978-1-449-31979-3" id="b1">
5          <title lang="EN">Python for Data Analysis</title>
6          <price currency="dollar">25.40</price>
7          <authors>
8              <author>Wes McKinney</author>
9          </authors>
10         <press> O'Reilly Media</press>
11         <pages>470</pages>
12         <description>Python for Data Analysis is concerned with the nuts and bolts of
13 manipulating, processing, cleaning, and crunching data in Python. It is also a practical, modern
14 introduction to scientific computing in Python, tailored for data-intensive applications. This is a
15 book about the parts of the Python language and libraries you&#8217;ll need to effectively
16 solve a broad set of data analysis problems. This book is not an exposition on analytical methods
17 using Python as the implementation language.</description>
18         <cover>book-python.jpg</cover>
19     </book>
20     <book isbn="978-7-115-28282-8" id="b2">
21         <title lang="CHN">数学之美</title>
22         <price>45.00</price>
23         <authors>
24             <author>吴军</author>
25         </authors>
26         <press>人民邮电出版社</press>
27         <pages>304</pages>
```

	<pre>&lt;description&gt;读了“数学之美”，才发现大学时学的数学知识，比如马尔科夫链、矩阵计算，甚至余弦函数原来都如此亲切，并且栩栩如生，才发现自然语言和信息处理这么有趣，才真正明白“数学是科学的皇后”这句名言。&lt;/description&gt;  &lt;cover&gt;book-math.jpg&lt;/cover&gt;  &lt;/book&gt;  &lt;/books&gt;</pre>
--	--

### 6.3 实验要求

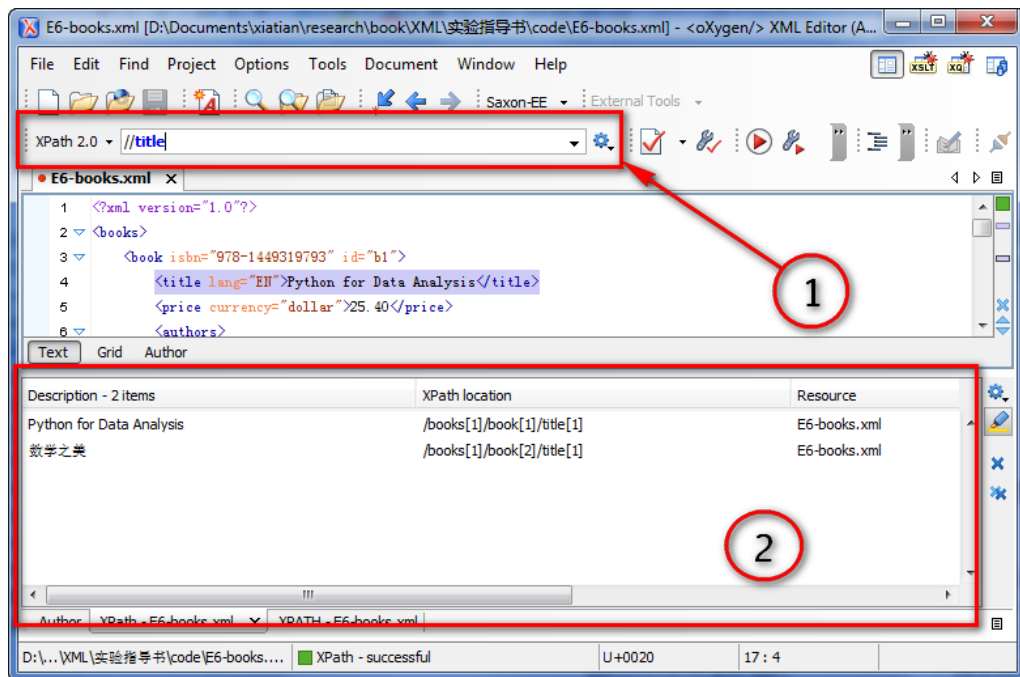
- [1] 通过 XPath 定位所有的 title;
- [2] 通过 XPath 定位第 2 个 title;
- [3] 输出所有图书的价格总和和平均价格;
- [4] 输出页码大于 400 页的图书标题;
- [5] 输出图书 id 为"b2"的图书标题的语言。

### 6.4 预期结果

- [1] 借助于 XML 专业工具所提供的 XPath 功能，输出定位的所有 title 内容和第 2 个 title 内容;
- [2] 借助于 XML 专业工具所提供的 XPath 功能，输出所有图书的价格总和和平均价格，分别为：70.4 和 35.2;
- [3] 借助于 XML 专业工具所提供的 XPath 功能 h，输出页码大于 400 页的图书标题：Python for Data Analysis;
- [4] 借助于 XML 专业工具所提供的 XPath 功能，输出 id 为"b2"的图书标题的语言：CHN。

### 6.5 实验步骤

- [1] 以 oXygen XML Editor 为 XML 编辑工具，输入"E6-books.xml"。
- [2] 在 XPath 输入框中输入"//title"，如下图位置 1 所示，然后按回车键，XPath 的选择结果如位置 2 所示，选中两个条目，分别为 Python for Data Analysis 和数学之美。



- [3] 在 XPath 输入框中输入 "//book[2]/title", 执行结果为选中一个条目: 数学之美。
- [4] 在 XPath 输入框中输入 "sum(//price)", 执行结果为 70.4。
- [5] 在 XPath 输入框中输入 "sum(//price) div count(//price)", 执行结果为 35.2。
- [6] 在 XPath 输入框中输入 "//book[pages > 400]/title", 执行结果选中一个条目: Python for Data Analysis
- [7] 在 XPath 输入框中输入 "//book[@id='b2']/title/@lang", 执行结果选中一个条目: CHN

## 6.6 思考分析

- [1] 请考虑 "//title/@lang" 和 "//title[@lang]" 有什么区别?
- [2] 实验中哪些步骤涉及的是属性结点选择, 选择属性结点和选择元素结点的显著区别在哪里?

## 实验7 利用 XSLT 和 CSS 对 XML 文档进行转换

### 7.1 实验目的

- [1] 理解 XSLT 的作用和转换流程;
- [2] 掌握把 XML 转换为 HTML 的样式编写规则;
- [3] 掌握如何把 XSL 和 CSS 配合使用, 实现对 XML 文档的格式化。

### 7.2 实验准备

- [1] 给定图书 XML 文档"E7-books.xml", 内容和"E6-books.xml"相同;
- [2] 给定图片"book-python.jpg"和"book-math.jpg", 与"E7-books.xml"放置在一起;
- [3] 参考教材, 在系统中配置好 Saxon XSLT 转换器。

### 7.3 实验要求

- [1] 通过 XSLT 把 XML 文档转换为 HTML;
- [2] 转换效果的图书标题、图片大小、字体等显示效果由 CSS 控制;
- [3] 默认字体为 Times New Roman、微软雅黑;
- [4] 图书标题的颜色为橙色、20pt 大小, 下侧外边距为 20px;
- [5] 如图书的价格大于 40, 则以红色显示价格。

### 7.4 预期结果

- [1] 通过 XSLT 对"E7-books.xml"进行转换, 转换结果在浏览器中的呈现效果如下:



#### Python for Data Analysis

ISBN:978-1-449-31979-3

作者: Wes McKinney

出版社: O'Reilly Media

价格: 25.40

Python for Data Analysis is concerned with the nuts and bolts of manipulating, processing, cleaning up, and analyzing data from various sources, and using an increasing number of powerful Python libraries to efficiently address complicated programming problems that can only be solved with a computer.



#### 数学之美

ISBN:978-7-115-28282-8

作者: 吴军

出版社: 人民邮电出版社

价格: 45.00

读了“数学之美”, 才发现大学时学的数学知识, 比如马尔科夫链、矩阵计算, 甚至勾股定理。

## 7.5 实验步骤

- [1] 以 oXygen XML Editor 为 XML 编辑工具，新建样式转换文件"E7-books.xsl", 内容如下：

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
3   version="1.0">
4   <xsl:template match="/">
5     <html>
6       <head>
7         <meta charset="utf-8"/>
8         <title>图书列表</title>
9         <link type="text/css" rel="stylesheet" href="E7-books.css"/>
10      </head>
11      <body>
12        <table>
13          <xsl:apply-templates select="books/book"/>
14        </table>
15      </body>
16    </html>
17  </xsl:template>
18
19  <xsl:template match="book">
20    <tr>
21      <td>
22        <img class="cover">
23          <xsl:attribute name="src">
24            <xsl:value-of select="cover"/>
25          </xsl:attribute>
26        </img>
27      </td>
28      <td>
29        <ul>
30          <li class="title"><xsl:value-of select="title"/></li>
31          <li>ISBN:<xsl:value-of select="@isbn"/></li>
32          <li>作者:<xsl:value-of select="authors"/></li>
33          <li>出版社:<xsl:value-of select="press"/></li>
34          <li>价格:
35            <xsl:choose>
36              <xsl:when test="price > 40">
37                <span class="highPrice">
```

38	<xsl:value-of select="price"/>
39	</span>
40	</xsl:when>
41	<xsl:otherwise>
42	<xsl:value-of select="price"/>
43	</xsl:otherwise>
44	</xsl:choose>
45	</li>
46	<li><xsl:value-of select="description"/></li>
47	</ul>
48	</td>
49	</tr>
50	</xsl:template>
51	</xsl:stylesheet>

[2] 新建"E7-books.css", 内容如下:

1	body {
2	font-family:"Times New Roman","微软雅黑";
3	}
4	ul li {
5	list-style-type:none;
6	}
7	.cover {
8	width:150px;
9	height:200px;
10	}
11	.title {
12	font-size:20pt;
13	margin-bottom:20px;
14	color:orange;
15	}
16	.highPrice {
17	color:red;
18	}

[3] 修改"E7-books.xml", 在第 2 行插入如下语句, 实现与"E7-books.xml"样式文件的关联:

<?xml-stylesheet type="text/xsl" href="E7-books.xml"?>:

[4] 通过浏览器打开"E7-books.xml", 显示效果应与预期结果保持一致;

[5] 通过 saxon 进行转换, 把转换结果输出到"E7-books.html"中, 参考命令如下:

```
java -jar c:\saxon9he.jar -s:E7-books.xml -xsl:E7-books.xsl -o:E7-books.html
```

[6] 通过浏览器和文本编辑工具分别打开"E7-books.html", 观察 Saxon 转换结果。

## 7.6 思考分析

- [1] 浏览器和 Saxon 专用 XSLT 处理器处理样式转换时的具体步骤有何不同？
- [2] XSLT 和 CSS 各有何优缺点？

RUC I RM 复习



## 实验8 创建基于 JavaScript 的网页时钟

### 8.1 实验目的

- [1] 掌握 JavaScript 的编写和测试方法;
- [2] 熟悉 JavaScript 的基本语法;
- [3] 熟悉 jQuery 的基本用法;
- [4] 理解浏览器对象模型 BOM;
- [5] 熟悉 JavaScript 的定时器功能。

### 8.2 实验准备

- [1] 下载 jQuery, 并把 jquery.js 放置在实验目录中。

### 8.3 实验要求

- [1] 在浏览器中显示当前时钟, 并根据系统时间自动更新;
- [2] 时钟的分、秒, 如不足两位数, 自动补 0;
- [3] 时钟背景颜色为灰色 gray, 边界颜色为银色 silver, 数字颜色为黄色 yellow;
- [4] 通过 CSS 控制时钟的显示样式;
- [5] 所有代码保存到一个单一的 HTML 文件"E8-clock.html"之中。

### 8.4 预期结果

- [1] 在浏览器中打开"E8-clock.html", 效果如下:



### 8.5 实验步骤

[1] 根据教材中定时器一节的描述，编写文件"E8-clock.html"，内容如下：

1	<!DOCTYPE html>
2	<html>
3	<head>
4	<meta charset="utf-8" />
5	<title>Clock</title>
6	<style>
7	#clock {
8	border:30px inset silver;
9	width:980px;
10	font-family:"Times New Roman";
11	font-size:16em;
12	background-color:gray;
13	color:yellow;
14	text-align:center;
15	}
16	</style>
17	</head>
18	<body>
19	<center>
20	<div id="clock">00:00:00</div>
21	</center>
22	<script type="text/javascript">
23	function format(n) {
24	return (n < 10) ? ('0'+n):n;
25	}
26	function showTime( ){
27	var clock = document.getElementById('clock');
28	var d = new Date();
29	var time = d.getHours() + ':'
30	+ format(d.getMinutes()) + ':'
31	+ format(d.getSeconds());
32	clock.innerText = time;
33	};
34	var timer = setInterval(showTime, 1000);
35	</script>
36	</body>
37	</html>

[2] 通过浏览器打开"E8-clock.html"，显示效果应与预期结果保持一致；

[3] 根据教材讲解，采用 jQuery 实现相同功能，保存到"E8-clock2.html"中，代码清单如下：

1	<!DOCTYPE html>
2	<html>

3	<head>
4	<meta charset="utf-8" />
5	<title>Clock</title>
6	<script src="jquery.js"></script>
7	<style>
8	#clock {
9	border:30px inset silver;
10	width:980px;
11	font-family:"Times New Roman";
12	font-size:16em;
13	background-color:gray;
14	color:yellow;
15	text-align:center;
16	}
17	</style>
18	</head>
19	<body>
20	<center>
21	<div id="clock">00:00:00</div>
22	</center>
23	<script type="text/javascript">
24	function format(n) {
25	return (n < 10) ? ('0'+n):n;
26	}
27	function showTime(){
28	var clock = \$('#clock');
29	var d = new Date();
30	var time = d.getHours() + ':'
31	+ format(d.getMinutes()) + ':'
32	+ format(d.getSeconds());
33	clock.text(time);
34	};
35	var timer = setInterval(showTime, 1000);
36	</script>
37	</body>
38	</html>

[4] 通过浏览器打开"E8-clock2.html", 显示效果应与预期结果保持一致。

## 8.6 思考分析

[1] 分析"E8-clock.html"和"E8-clock2.html"有何不同?

[2] "E8-clock.html"中第 27 行的 document 变量, 在源文件中并没有定义, 为什么

能够直接使用？

- [3] 打开 Web 浏览器所提供的控制台，利用控制台所提供的语法提示功能，查看 Date()对象还提供了哪些方法？

## 实验9 利用 JavaScript 操纵 DOM

### 9.1 实验目的

- [1] 理解 DOM 的主要作用；
- [2] 掌握 JavaScript 利用 XMLHttpRequest 访问 XML 文件的方法；
- [3] 掌握 DOM 所提供的操作 XML 的常见方法；
- [4] 掌握 DOM、JavaScript、CSS 和 HTML 技术组合使用，用于网页呈现的方法。

### 9.2 实验准备

- [1] 根据教材内容，在个人计算机上搭建 Mongoose Web 服务器，并把 Web 服务器的端口设为 80；
- [2] 把 jquery.js 放置在 Mongoose Web 服务器的主目录中，实验中其他相关文件也放置在主目录下；
- [3] 给定图书 XML 文档"E7-books.xml"。

### 9.3 实验要求

- [1] 通过 DOM 读取"E7-books.xml"的内容，并逆序按预期效果输出；
- [2] 输出结果的字体、颜色等基本样式采用 CSS 控制；
- [3] 每个元素默认占据一行，字体缺省为"Times New Roman"、微软雅黑；
- [4] 元素 title 的字体大小为 20pt，下部外边距为 20px，颜色为红色 red；
- [5] 封面图片的宽度和高度分别为 150px 和 200px；
- [6] 图书描述 description 的字体颜色为海军色 navy，大小为 0.6em，占据的最大宽度为 600px。
- [7] 网页代码保存到"E9-books.html"中，操纵 XML 的脚本保存到"E9-books.js"中，CSS 样式代码保存到"E9-books.css"中。

### 9.4 预期结果

- [1] 在浏览器中打开"E9-books.html"，效果如下：



## 数学之美

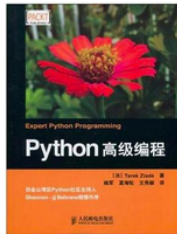
ISBN: 978-7-115-28282-8

作者: 吴军

出版社: 人民邮电出版社

价格: 45.00

读了“数学之美”，才发现大学时学的数学知识，比如马尔科夫链、矩阵计算，甚至余弦函数原来都如此亲切，并且栩栩如生，才发现自然语言和信息处理这么有趣，才真正明白“数学是科学的皇后”这句名言。



## Python for Data Analysis

ISBN: 978-1-449-31979-3

作者: Wes McKinney

出版社: O'Reilly Media

价格: 25.40

Python for Data Analysis is concerned with the nuts and bolts of manipulating, processing, cleaning, and crunching data in Python. It is also a practical, modern introduction to scientific computing in Python, tailored for data-intensive applications. This is a book about the parts of the Python language and libraries you'll need to effectively solve a broad set of data analysis problems. This book is not an exposition on analytical methods using Python as the implementation language.

## 9.5 实验步骤

[1] 编写文件"E9-books.html", 内容如下:

```
1 <html>
2   <head>
3       <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
4       <meta charset="utf-8"/>
5       <title>图书列表</title>
6       <script type="text/javascript" src="jquery.js"></script>
7       <script type="text/javascript" src="E9-books.js"></script>
8       <link type="text/css" rel="stylesheet" href="E9-books.css"/>
9   </head>
10  <body>
11      <table id="container">
12      </table>
13  </body>
14 </html>
```

[2] 编写 CSS 文件"E9-books.css", 内容如下:

```
1 body {
2     font-family:"Times New Roman","微软雅黑";
3 }
4 ul li {
5     list-style-type:none;
6 }
7 .cover {
8     width:150px;
9     height:200px;
10 }
```

```

11 .title {
12     font-size:20pt;
13     margin-bottom:20px;
14     color:red;
15 }
16 .description {
17     color:navy;
18     font-size:0.6em;
19     max-width:600px;
20 }

```

[3] 编写脚本"E9-books.js", 实现从"E7-books.xml"文件中读取内容, 并插入到网页中, 脚本内容如下:

```

1  function loadXml(xmlFile) {
2      var xhttp = new XMLHttpRequest();
3      xhttp.open("GET", xmlFile, false);
4      xhttp.send();
5      return xhttp.responseXML;
6  }
7
8  $(document).ready(function(){
9      var xmlDoc = loadXml('E7-books.xml');
10     var bookList = xmlDoc.getElementsByTagName("book");
11     for(var i=bookList.length; i> 0; i--) {
12         var bookNode = bookList[i-1];
13         var cover = bookNode.getElementsByTagName('cover')[0].textContent;
14         var title = bookNode.getElementsByTagName('title')[0].textContent;
15         var isbn = bookNode.attributes['isbn'].value;
16         var author = bookNode.getElementsByTagName('authors')[0].textContent;
17         var press = bookNode.getElementsByTagName('press')[0].textContent;
18         var price = bookNode.getElementsByTagName('price')[0].textContent;
19         var description = bookNode.getElementsByTagName('description')[0].textContent;
20         var row = '<tr>\
21             <td><img class="cover" src="" + cover + ""></td>\
22             <td>\
23                 <ul>\
24                     <li class="title">' + title + '</li>\
25                     <li>ISBN: ' + isbn + ' </li>\
26                     <li>作者: ' + author + '</li>\
27                     <li>出版社: ' + press + '</li>\
28                     <li>价格: ' + price + '</li>\
29                     <li class="description">' + description + '</li>\
30                 </ul>\
31             </td>\

```

32	<code>&lt;/tr&gt;;</code>
33	<code>\$('#container').append(row);</code>
34	<code>}</code>
35	<code>});</code>

- [4] 启动 Mongoose，然后打开浏览器，在地址栏输入以下地址，网页显示效果应与预期结果一致。

`http://localhost/E9-books.html`

## 9.6 思考分析

- [1] JavaScript 所提供的 DOM 功能有什么不便之处？
- [2] 实验中不通过 Mongoose，而是直接利用浏览器打开"E9-books.html"，能够正常显示预期效果？为什么？



## 实验10 利用 SVG 绘制散点图

### 10.1 实验目的

- [1] 熟悉 SVG 的功能特点
- [2] 熟悉 d3.js 的基本语法
- [3] 掌握 HTML 和 JavaScript 脚本编程的基本步骤

### 10.2 实验准备

散点图的实验数据如下：

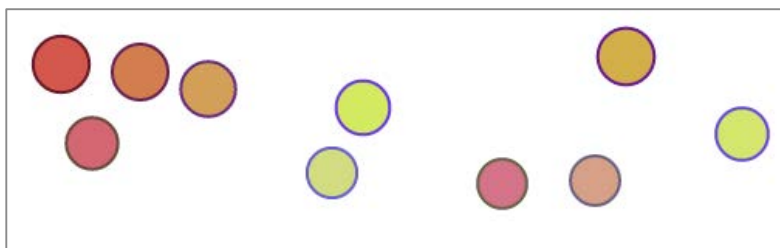
```
var dataset = [  
    [ 15, 20 ],  
    [ 380, 95 ],  
    [ 230, 48 ],  
    [ 100, 36 ],  
    [ 320, 97 ],  
    [ 400, 15 ],  
    [ 475, 55 ],  
    [ 35, 71 ],  
    [ 76, 25 ],  
    [ 210, 90 ]  
];
```

### 10.3 实验要求

- [1] SVG 画布大小为 600×350;
- [2] dataset 中的每一对数据代表圆心位置，在图上显示时，纵坐标位置下移 10 个像素；
- [3] 每个圆的半径为  $\text{Math.sqrt}(h - d[1])$ ，其中 h 表示画布的高度，d[1]表示圆心的纵坐标；
- [4] 每个圆的填充颜色为 `rgba(200, d[0]%250, d[1]%250, 0.75)`，其中 d[0], d[1]分别代表坐标的横纵坐标值；
- [5] 每个圆的 `stroke-width` 属性为 2；
- [6] 每个圆的 `stroke` 颜色为 `rgba(100, d[1]%250, d[0]%250, 0.95)`

## 10.4 预期结果

在浏览器中的预期显示效果如下：



## 10.5 实验步骤

- [1] 参考教材讲述的内容，下载并在 HTML 代码中引用 d3.js, HTML 文件保存为“E10-scatter.html”；
- [2] 编写代码，利用 d3.js 实现预期显示结果；参考代码如下：

```
1  <!DOCTYPE html>
2  <html>
3    <head>
4      <meta charset="utf-8" />
5      <title>Scatterplot</title>
6      <script type="text/javascript" src="d3/d3.v3.js"></script>
7    </head>
8    <body>
9      <script type="text/javascript">
10         var dataset = [
11           [ 35, 20 ],
12           [ 380, 95 ],
13           [ 230, 48 ],
14           [ 130, 36 ],
15           [ 320, 97 ],
16           [ 400, 15 ],
17           [ 475, 65 ],
18           [ 55, 71 ],
19           [ 86, 25 ],
20           [ 210, 90 ]
21         ];
22
23         var h = 350;
24         var w = 600;
25         var svg = d3.select("body")
26           .append("svg")
```

27	.attr("width", w)
28	.attr("height", h);
29	svg.selectAll("circle")
30	.data(dataset)
31	.enter()
32	.append("circle")
33	.attr("cx", function(d) {
34	return d[0];
35	})
36	.attr("cy", function(d) {
37	return d[1]+10;
38	})
39	.attr("r", function(d){
40	return Math.sqrt(h - d[1]);
41	})
42	.attr("fill", function(d){
43	return "rgba(200, " + d[0]%250 + ", " + d[1]%250 + ", 0.75)";
44	})
45	.attr("stroke", function(d){
46	return "rgba(100, " + d[1]%250 + ", " + d[0]%250 + ", 0.95)";
47	})
48	.attr("stroke-width", 2);
49	
50	</script>
51	</body>
52	</html>

[3] 在浏览器中,借助于 Web Developer 控制台,查看图形对应的 SVG DOM 结果。

## 10.6 思考分析

- [1] 不使用 d3.js, 直接利用 SVG 的语法, 也可以绘制目标结果的图形, 为什么还要使用 d3.js?
- [2] SVG 与 PNG、JPG、GIF 图像格式的显著不同有哪些?