



《XML 设计技术》

随堂测验 2020-12-3

一、单选题

1. XML 表示 ()
A. eXtensible Markup Language B. eXtra Modern Link
C. Example Markup Language D. X-Markup Language
2. 以下哪一项描述是正确的 ()
A. 有效的 XML 文档不一定良构 B. 良构的 XML 文档一定有效
C. 有效性和良构性没有任何关系 D. 良构的 XML 文档不一定有效
3. 下面哪一项关于 XML 的描述是正确的 ()
A. 所有 XML 元素名称都必须是小写的 B. 所有 XML 元素都必须正确地嵌套
C. 所有元素都不能为空 D. 所有 XML 文档都必须有一个 DTD
4. XML 的 () 提供了一种避免元素命名冲突的方法。
A. 命名空间 B. DTD C. XSD D. XSL
5. 在为 XML 文档定义文档模式时, 通常可以使用 ()
A. UML B. Namespaces C. XML Schema D. XSLT
6. DTD 的全称是 ()
A. Document Type Definition B. Direct Type Definition
C. Dynamic Type Definition D. Do The Dance
7. 在 DTD 中定义元素的属性时, #IMPLIED 表示该属性是 ()
A. 必须赋值 B. 可有可无 C. 固定取值 D. 以上都不对
8. 在 XML Schema 的元素声明中, minOccurs 和 maxOccurs 的缺省值分别是 ()
A. 0, 0 B. 1, 1 C. 1, unbounded D. unbounded, unbounded
9. 在 XML Schema 的类型声明中, 可以使用 ()
A. 全局的匿名数据类型 B. 局部的命名数据类型
C. 全局的命名数据类型 D. 以上都可以
10. 对于 XML Schema 中简单数据类型的派生, 不包括以下哪种方式 ()
A. 限制 (Restriction) B. 列表 (List) C. 扩展 (Extension) D. 合并 (Union)
11. 以下哪项不是 XPath 中 step (步) 的组成部分 ()
A. 轴标识符 B. 节点测试 C. 判定谓词 D. 结果排序

12. 以下哪项不是 XPath 中 step（步）的关系轴（ ）

- A. following B. following-sibling C. following-or-self D. self

13. 关于 XPath 中的节点测试，以下描述正确的是（ ）

- A. 节点测试可以测试节点的名称 B. 节点测试可以测试节点的类型
C. 没有名称的节点也能进行节点测试 D. 以上描述都对

14. XSLT 的 XSL 指的是（ ）

- A. eXtra Style Language B. eXpandable Style Language
C. eXtensible Style Listing D. eXtensible Stylesheet Language

15. 有关 XSLT 的内置模板，下面的描述正确的是（ ）

- A. 内置模板在 XML 文档树中各种节点的处理提供了一套缺省的实现，这些实现并没有太大的用处，内置模板可有可无。
B. 在内置模板中，对 XML 文档树的遍历方式是自上而下，从左到右。
C. 如果没有用户定义的模板，内置模板本身不能执行。
D. 上面的描述都是正确的。

二、判断题

➤ 请判断每条 XPath 下面对应的语义描述是否正确，对的写 true，错的写 false。

1. //company/descendant::*

语义描述：所有 company 元素节点的所有子孙节点。（ ）

2. //publisher[@name='Education']

语义描述：所有 name 属性等于 Education 的 publisher 元素节点。（ ）

3. //X

语义描述：等价于 /descendant-or-self::node()/child::X。（ ）

4. //parent::*

语义描述：所有的分支节点和叶节点。（ ）

5. /*/*/color

语义描述：在 XML 文档树第 3 层中的所有 color 元素节点。（ ）

6. //member[not(@*)]

语义描述：第一个不包含任何属性的 member 元素节点。（ ）

➤ 给定 XML 文档如下：

```
<friends>
  <cat>Tom</cat>
  <mouse>Jerry</mouse>
</friends>
```

将下列 XPath 分别运行于上述 XML 文档，请判断其返回值是 true 或 false。

（提示：运算符 << 和 >> 用于比较两个节点的文档序）

7. /friends/cat >> /friends/mouse （ ）

8. /friends >> /friends/cat （ ）

9. /friends/cat << /friends/cat/text() （ ）

10. /friends/cat << /friends/mouse/text() （ ）

三、设计题

1. 根据下面的 department.xml 文档，设计合理的 XML Schema 模式，要求：
 - (1) department 只有“软件工程”和“数字媒体技术”两个名称；
 - (2) 每个 department 下的学生个数最多为 300 人；
 - (3) 学生必须有“sID”子元素；
 - (4) 学生的“birthday”格式为：xxxx 年 xx 月 xx 日，其中 x 为数字；
 - (5) 学生的“email”格式为 xxx@xmu.edu.cn，其中@之前必须有三个及以上数字或字母的字符，且必须以字母开头；@之后固定为“xmu.edu.cn”；
 - (6) 学生的课程必须有 cID 属性；
 - (7) 每位学生的课程数最多为 40 门；
 - (8) 学生的 Score 的值范围为 0-100（含）；

```
<?xml version="1.0" encoding="UTF-8"?>
<departments>
  <department>
    <name>软件工程</name>
    <student>
      <sID>24320172203000</sID>
      <name>张三</name>
      <birthday>2000年01月01日</birthday>
      <email>abc@xmu.edu.cn</email>
      <scores>
        <course cID="001">82</course>
        <course cID="002">00</course>
      </scores>
    </student>
    <student>
      <sID>24320172203001</sID>
      <name>李四</name>
      <birthday>2000年01月01日</birthday>
      <email>a123456@xmu.edu.cn</email>
      <scores>
        <course cID="001">90</course>
        <course cID="002">60</course>
      </scores>
    </student>
  </department>
  <department>
    <name>数字媒体技术</name>
    <student>
      <sID>24320172203002</sID>
      <name>王五</name>
      <birthday>2000年01月01日</birthday>
      <email>ab8c@xmu.edu.cn</email>
```

```

        <scores>
            <course cID="001">88</course>
            <course cID="002">77</course>
        </scores>
    </student>
    <student>
        <sID>24320172203003</sID>
        <name>郑明</name>
        <birthday>2000年01月01日</birthday>
        <email>a12346@xmu.edu.cn</email>
        <scores>
            <course cID="001">93</course>
            <course cID="002">69</course>
        </scores>
    </student>
</department>
</departments>

```

department.xml

2. 现将下面的 summary.xslt 运行 book.xml 之上，请写出转换后的结果。

```

<?xml version="1.0" encoding="UTF-8"?>
<book>
    <title>The Grand Design</title>
    <author>Stephen Hawking</author>
    <author>Leonard Mlodinow</author>
    <section id="1">
        <title>Existence</title>
        <p>Text ... </p>
        <figure height="200" width="400">
            <title>World Map</title>
            <image source="map.jpg"/>
        </figure>
        <p>Text ... </p>
        <figure height="100" width="100">
            <title>Philosophy</title>
            <image source="philosophy.gif"/>
        </figure>
        <p>Text ... </p>
    </section>
    <section id="2">
        <title>History</title>
        <p>Text ... </p>
        <figure height="250" width="400">
            <title>What is light</title>

```

```

        <image source="light.gif"/>
      </figure>
    </section>
    <section id="3">
      <title>Universe</title>
      <p>Text ... </p>
    </section>
  </book>

```

book.xml

```

<xsl:stylesheet version="2.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="xml" version="1.0" encoding="UTF-8" indent="yes"/>
<xsl:template match="/">
  <toc>
    <xsl:for-each select="//book">
      <xsl:call-template name="section-summary"/>
    </xsl:for-each>
  </toc>
</xsl:template>
<xsl:template name="section-summary">
  <xsl:for-each select="section">
    <section>
      <xsl:attribute name="id" select="./@id"/>
      <xsl:copy-of select="./title"/>
      <figcount>
        <xsl:value-of select="count(figure)"/>
      </figcount>
    </section>
  </xsl:for-each>
</xsl:template>
</xsl:stylesheet>

```

summary.xslt