

Correzione Compito 16.03.2010

Ombretta Gaggi
Università di Padova

Domande V/F : DTD

Dato seguente frammento di codice XML e il frammento di DTD:
(2 punti per ogni risposta corretta, -1 per ogni risposta sbagliata, 0 se lasciato in bianco)

<code><prodotto id='01' sconto=''></code>	<code><!ELEMENT prodotto (descr tipologia quantita marca?) ></code>
<code> <descr>LCD 17 pollici</descr></code>	<code><!ATTLIST prodotto id CDATA #REQUIRED</code>
<code> <tipologia>Monitor</tipologia></code>	<code> sconto CDATA #IMPLIED></code>
<code> <quantita>3</quantita></code>	<code><!ELEMENT quantita (#PCDATA) ></code>
<code></prodotto></code>	<code><!ELEMENT descr (#PCDATA) ></code>

indicare quali delle seguenti affermazioni sono vere (V) o false (F)

Il frammento XML è ben formato

Se ben formato, il frammento XML è valido rispetto al frammento DTD

Il tag `<quantita>` è valido rispetto la definizione XMLSchema `<element name="quantita" type="xs:string"/>`

V F
V F
V F



Domande V/F : DTD

Dato seguente frammento di codice XML e il frammento di DTD:
(2 punti per ogni risposta corretta, -1 per ogni risposta sbagliata, 0 se lasciato in bianco)

<code><prodotto id='01' sconto=''></code>	<code><!ELEMENT prodotto ((tipologia, descr)+, dataArrivo, sconto?) ></code>
<code> <descr>LCD 17 pollici</descr></code>	<code><!ATTLIST prodotto id CDATA #REQUIRED</code>
<code> <tipologia>Monitor</tipologia></code>	<code> sconto CDATA #IMPLIED></code>
<code> <dataArrivo>16/03/2010</dataArrivo></code>	<code><!ELEMENT dataArrivo (#PCDATA) ></code>
<code></prodotto></code>	<code><!ELEMENT sconto (#PCDATA) ></code>

Il frammento XML è ben formato

Se ben formato, il frammento XML è valido rispetto al frammento DTD

Il tag `<dataArrivo>` è valido rispetto la definizione XMLSchema `<element name="dataArrivo" type="xs:date"/>`

V F
V F
V F



Domande V/F a carattere generale

Indicare quali delle seguenti affermazioni sono vere (V) o false (F)
(2 punti per ogni risposta corretta, -1 per ogni risposta sbagliata, 0 se lasciato in bianco)

Il tag XHTML input può comparire direttamente all'interno del tag form

Un test esaustivo dell'accessibilità di una pagina web non può essere fatto in modo automatico

Permettere la fruibilità di una pagina attraverso gli screenreader è condizione necessaria e sufficiente per definire una pagina accessibile

V F
V F
V F



XPath

- Dato il file XML in allegato, scrivere la funzione XPath che ritorna tutti i prodotti venduti ai quali è stato applicato uno sconto:
 - //prodotto[sconto]
 - /documento/riga/prodotto[sconto]
 - /documento//prodotto[sconto]
- Dato il file XML in allegato, scrivere la funzione XPath che ritorna tutti i monitor venduti:
 - //prodotto[tipologia="Monitor"]
 - /documento/riga/prodotto[tipologia="Monitor"]
 - /documento//prodotto[tipologia="Monitor"]



Frammento di DTD

- Scrivere il frammento DTD che descrive prodotto:

```
<!ELEMENT prodotto ( #PCDATA | descr | quantita | sconto |  
tipologia )+ >  
<!ATTLIST prodotto id CDATA #REQUIRED >
```

- Scrivere il frammento DTD che descrive radice:

```
<!ELEMENT documento ( #PCDATA | data | riga | tipologia )+ >  
<!ATTLIST documento numero CDATA #REQUIRED >
```



Frammento di schema per il tag radice

```
<xs:element name="documento">  
  <xs:complexType mixed="true" >  
    <xs:sequence>  
      <xs:element name="tipologia" type="xs:string" />  
      <xs:element name="data" type="xs:string" />  
      <xs:element name="riga" type="Triga"  
maxOccurs="unbounded" />  
    </xs:sequence>  
    <xs:attribute name="numero" type="xs:string" />  
  </xs:complexType>  
</xs:element>
```



Frammento XMLSchema per il tag prodotto

```
<xs:element name="prodotto">  
  <xs:complexType mixed="true" >  
    <xs:sequence>  
      <xs:element name="descr" type="xs:string" />  
      <xs:element name="tipologia" type="xs:string" />  
      <xs:element name="quantita" type="xs:positiveInteger" />  
      <xs:element name="sconto" type="xs:string"  
minOccurs="0"/>  
    </xs:sequence>  
    <xs:attribute name="id" type="xs:string" />  
  </xs:complexType>  
</xs:element>
```



Domande V/F su frammento di schema

```
<xs:element name="documento">
  <xs:complexType><xs:sequence>
    <xs:element name="tipologia" type="xs:string" />
    <xs:element name="data" type="xs:string" />
    <xs:element name="riga" type="Triga" maxOccurs="unbounded" />
    <xs:attribute name="numero" type="xs:string" />
  </xs:sequence></xs:complexType>
</xs:element>
```

Il frammento di schema è valido rispetto lo standard XMLSchema

L'espressione XPath `//tipologia` ha come risultato tutte le tipologie dei prodotti del documento

Il file XML è valido anche rispetto uno schema che contiene un elemento `<data>` con tipo `xs:date`

V ☐ F ☒
V ☐ F ☒
V ☐ F ☒



Domande V/F su frammento di schema

```
<xs:complexType name="Tprodotto">
  <xs:sequence>
    <xs:element name="descr" type="xs:string" />
    <xs:element name="tipologia" type="xs:string" />
    <xs:element name="quantita" type="xs:positiveInteger" />
    <xs:element name="sconto" type="xs:string" minOccurs="0"/>
    <xs:attribute name="id" type="xs:string" />
  </xs:sequence>
</xs:complexType>
```

Il frammento di schema è valido rispetto lo standard XMLSchema

Se il frammento XMLSchema è corretto, utilizza il modello Tende alla Veneziana

L'espressione XPath `//tipologia` ha come risultato tutte le tipologie dei prodotti del documento

V ☐ F ☒
V ☐ F ☒
V ☐ F ☒



Template

- ❑ Scrivere un template che, inserito in un foglio di stile completo, permetta di stampare a video (utilizzando il linguaggio XHTML) la lista dei prodotti inseriti nell'ordine indicando lo sconto ottenuto (la stringa con lo sconto deve comparire solo se questo è presente):



Template

```
<xsl:template match="/">
  <html><body><ul>
    <xsl:for-each select="documento/riga/prodotto">
      <li><xsl:value-of select="descr" />
        <xsl:if test="sconto">
          <em>Sconto applicato:<xsl:value-of
            select="sconto"/></em>
        </xsl:if></li>
    </xsl:for-each>
  </ul></body></html>
</xsl:template>
```



Template ordinato

- ❑ Scrivere un template che, inserito in un foglio di stile completo, permetta di stampare a video (utilizzando il linguaggio XHTML) la lista dei prodotti, ordinati per tipologia in ordine alfabetico inverso:

```
<xsl:template match="/">
  <html><body><ul>
    <xsl:for-each select="documento/riga/prodotto">
      <xsl:sort select="tipologia" order="descending"/>
      <li><xsl:value-of select="tipologia" />
        <xsl:value-of select="descr" /> </li>
    </xsl:for-each>
  <ul></body></html>
</xsl:template>
```

