

Correzione Compito 22.03.2011

Ombretta Gaggi
Università di Padova

DTD del tag film

```
<!ELEMENT film (  
    (regia, genere, durata, (cast|vociOriginali), riassunto,  
                                locandina, note?)|  
    (orario+)  
)>  
  
<!ATTLIST film titolo CDATA #REQUIRED>
```



XMLSchema per il tag spettacoli/film

```
<xs:element name="film">  
  <xs:complexType>  
    <xs:sequence>  
      <xs:element name="regia" type="xs:string" />  
      <xs:element name="genere" type="Tgenere" />  
      <xs:element name="durata" type="xs:string" />  
      <xs:choice>  
        <xs:element name="cast" type="xs:string"/>  
        <xs:element name="vociOriginali" type="xs:string" />  
      </xs:choice>  
      <xs:element name="riassunto" type="xs:string" />  
    </xs:sequence>  
  </xs:complexType>  
</xs:element>
```



XMLSchema per il tag spettacoli/film

```
...  
<xs:element name="locandina" type="Tlocandina" />  
<xs:element name="note" type="Tnote" minOccurs="0"/>  
</xs:sequence>  
<xs:attribute name="titolo" type="xs:string" />  
</xs:complexType>  
</xs:element>
```



Domande V/F documento XML

- | | | |
|---|------------------------------------|------------------------------------|
| Il tag locandina può essere definito in XMLSchema con un tipo semplice | V | <input checked="" type="radio"/> F |
| L'espressione XPath count(//film/@titolo) restituisce il numero di film in visione | V | <input checked="" type="radio"/> F |
| Il tipo del tag programmazione si può ottenere per estensione dal tipo del tag spettacoli | <input checked="" type="radio"/> V | F |
| Lo schema completo che valida il documento può contenere due tipi derivati per estensione | <input checked="" type="radio"/> V | F |
| Utilizzando XMLSchema l'elemento <i>durata</i> può avere tipo xs:duration | V | <input checked="" type="radio"/> F |



Espressioni XPath

- Dato il documento XML in allegato 1, scrivere la funzione XPath che ritorna tutti i titoli di film non vietati ai minori:
 - //spettacoli/film[(note/@vietatoMinori="no") or not(note)]/@titolo
 - (//film[note/@vietatoMinori="no"]/@titolo | //spettacoli/film[not(note)]/@titolo)
- Dato il documento XML in allegato 1, scrivere la funzione XPath che ritorna tutti i titoli di film proiettati in sala 1:
 - /cinema/programmazione/film[orario/@sala="1"]/@titolo
 - //film[orario/@sala="1"]/@titolo
 - //film[orario[@sala="1"]]/@titolo



Foglio di trasformazione di stile: testo

- Scrivere un template che, inserito in un foglio di stile completo, permetta di stampare a video (utilizzando il linguaggio XHTML) il numero di film in visione, e la lista dei film. La lista deve contenere le seguenti informazioni: titolo, regia, durata, cast e riassunto. Accanto al titolo di ogni film deve essere indicate eventuali limitazioni (film vietato ai minori di 14, ai minori di 18, ecc.)



Foglio di trasformazione di stile - soluzione

```
<xsl:template match="/cinema">
```

```
...
```

```
<p> Nel cinema di <xsl:value-of select="@sede" /> ci sono  
<xsl:value-of select="count(spettacoli/film)" /> film in  
visione: </p>
```

```
<dl>
```

```
<xsl:for-each select="spettacoli/film">
```

```
<dt><xsl:value-of select="@titolo" />
```

```
<xsl:if test="note/@vietatoMinori and  
not(note/@vietatoMinori='no')">
```

```
(vietato ai minori di
```

```
<xsl:value-of select="note/@vietatoMinori"/>)
```

```
</xsl:if>
```

```
</dt>
```



Foglio di trasformazione di stile - soluzione

```
<dd>Regia: <xsl:value-of select="regia" /> </dd>
<dd>Durata: <xsl:value-of select="durata" /> </dd>
<dd>Cast: <xsl:value-of select="cast|vociOriginali" /> </dd>
<dd>Riassunto: <xsl:value-of select="riassunto" /> </dd>
</xsl:for-each>
</dl></body></html>
</xsl:template>
```



Foglio di trasformazione di stile: testo

- ❑ Scrivere un template che, inserito in un foglio di stile completo, permetta di stampare a video (utilizzando il linguaggio XHTML) il numero di film in programmazione il giorno 22-03-2011, e la lista dei titoli dei film, con accanto la lista degli orari di inizio spettacolo (attenzione al carattere di divisione).



Foglio di trasformazione di stile - soluzione

```
<xsl:template match="/cinema"> ...
<p> Nel cinema di oggi (2011-03-22) ci sono in programmazione
<xsl:value-of select="count(programmazione[@giorno='2011-03-22']/film)" /> film: </p>
<ul>
  <xsl:for-each select="programmazione[@giorno='2011-03-22']/film">
    <li><xsl:value-of select="@titolo" />
    <ul> <xsl:for-each select="orario">
      <li><xsl:value-of select="." /></li>
    </xsl:for-each></ul></li>
  </xsl:for-each>
</ul>...
</xsl:template>
```



Domande V/F

- La divisione tra contenuto e comportamento aiuta a migliorare il posizionamento nelle risposte dei motori di ricerca ☒ V ☐ F
- La divisione tra contenuto e comportamento contribuisce ad abbassare al dimensione totale di un sito web ☒ V ☐ F
- Una struttura organizzativa gerarchica, per essere efficiente, deve essere ampia e poco profonda ☒ V ☐ F
- In generale si può dire che il linguaggio XMLSchema è maggiormente adatto per lo scambio di dati tra applicazioni, mentre il linguaggio DTD per lo scambio di dati tra utenti. ☒ V ☐ F

