

```
Soluzione DTD

- <!ELEMENT binario( zero*,uno*)*>
<!ELEMENT zero(#PCDATA)>
<!ELEMENT uno(#PCDATA)>

- <!ELEMENT binario( zero|uno)*>
<!ELEMENT zero(#PCDATA)>
<!ELEMENT uno(#PCDATA)>
```

```
Soluzione XML Schema
```

```
Documenti istanza validi
<br/>
<br/>
dinario>
                                 <hinario>
   <uno>1</uno>
                                    <uno>0</uno>
  <zero>0</zero>
                                    <zero>1</zero>
  <uno>1</uno>
                                    <uno>0</uno>
  <uno>1</uno>
                                    <uno>1</uno>
  <uno>1</uno>
                                    <uno>0</uno>
   <zero>0</zero>
                                    <zero>0</zero>
</binario>
                                 </binario>
```

```
Documenti istanza validi (per DTD)

<br/>
```

```
Documenti istanza validi
                                  <br/>
<br/>
dinario>
  <uno>1</uno>
                                     <uno>1</uno>
  <zero>0</zero>
                                     <7ern/>
  <uno>1</uno>
                                     <uno/>
  <uno>1</uno>
                                     <uno/>
                                     <uno/>
  <uno>1</uno>
  <zero>0</zero>
                                     <zero>0</zero>
</binario>
                                  </binario>
```

```
Esercizio 2 - soluzione - 1
<xsd:schema xmlns... >
<xsd:element name="lettera" type="Tlettera"/>
<xsd:element name="cliente" type="xsd:string"/>
<xsd:element name="ordine" type="Tordine";</pre>
<xsd:element name="articolo" type="Tarticolo" />
<xsd:element name="codice" type="xsd:string"/>
<xsd:element name="descr" type="xsd:string"/>
<xsd:element name="responsabile" type="Tresponsabile"/>
<xsd:element name="tit" type="xsd:string" />
<xsd:attribute name="num" type="xsd:integer"/>
<xsd:complexType name="Tlettera" mixed="true"><xsd:sequence>
       <xsd:element ref="cliente" />
       <xsd:element ref="ordine"/>
       <xsd:element ref="responsabile"/>
</xsd:sequence></xsd:complexType>
```

```
Esercizio 2 - soluzione tende alla veneziana - 1
<xsd:schema xmlns... >
<xsd:element name="lettera" type="Tlettera"/>
<xsd:complexType name="Tlettera" mixed="true">
  <xsd:sequence><xsd:element name="cliente" type="xsd:string"/>
       <xsd:element name="ordine" type="Tordine"/>
       <xsd:element name="responsabile" type="Tresponsabile"/>
  </xsd:sequence>
</xsd:complexType>
<xsd:complexType name="Tordine" >
  <xsd:sequence>
       <xsd:element name="articolo" type="Tarticolo"</pre>
                    maxOccurs="unbounded"/>
  </xsd:sequence>
   <xsd:attribute name="num" type="xsd:integer"/>
</xsd:complexType>
Tecnologie Web - 12
```

```
Problema?

E' possibile modellare questa struttura?

<lettera xmlns=...>
Gentile <cliente>Tal dei Tali</cliente>,
la informiamo che i seguenti articoli da lei ordinati non sono più disponibili a magazzino:
<ordine num="1234">
<articolo><codice>1</codice>
<descr>articolo 1</descr></articolo></descr>articolo></ordine>
Cordiali saluti,
<responsabile><tit>dr.</tit> Mario Rossi</responsabile>
</lettera>
Tecnologie Web - 14
```

```
Creazione della lettera
 <ordine num="1234"> <cliente>Tal dei Tali</cliente>,
 <articolo><codice>1</codice><descr>articolo 1</descr></articolo>
 <articolo><codice>5</codice><descr>articolo 5</descr></articolo>
<lettera xmlns=...>
            Gentile <cli>entile <cli>entil
                                la informiamo che i seguenti articoli da lei ordinati non sono più
            disponibili a magazzino:
            Numero ordine: 1234
             <articolo><codice>1</codice>
                                                                                                    <descr>articolo 1</descr></articolo>
             <articolo><codice>5</codice><descr>articolo 5</descr></articolo>
             </ordine>
            Cordiali saluti, <responsabile > <tit>dr. </tit> Mario Rossi </responsabile >
 </lettera>
 Tecnologie Web - 17
```

```
Foglio di trasformazione
<xsl:stylesheet ...>
<xsl:output method="xml" version='1.0' indent='yes'/>
<xsl:template match="/">
  <?xml version="1.0" encoding="UTF-8"?>
   <lettera>
    Gentile <xsl:copy-of select="ordine/cliente" />,
     la informiamo che i seguenti articoli da lei ordinati non sono
    piú disponibili a magazzino:
     Numero ordine: <xsl:value-of select="ordine/@num"/>
     <xsl:for-each select="ordine/articolo">
       <xsl:copy-of select="." />
     </xsl:for-each>
       Cordiali saluti,<responsabile><tit>dr.</tit> Mario
     Rossi</responsabile> </lettera> </xsl:template>
</xsl:stylesheet>
```

```
Soluzione DTD

<!ELEMENT montagne(catenaMontuosa+)>
<!ELEMENT catenaMontuosa(monte+)>
<!ELEMENT monte(nome,regione,altezza)>

<!ELEMENT monte(nome,regione?,altezza?)>

<!ELEMENT nome(#PCDATA)>
<!ELEMENT regione(#PCDATA)>
<!ELEMENT altezza(#PCDATA)>

<!ATTLIST catenaMontuosa nome CDATA>
<!ATTLIST altezza unitaMisura CDATA>

Tecnologie Web - 20
```

```
Soluzione XML Schema - 1

<pre
```

```
Soluzione XML Schema - 4

<xsd:element name="nome" type="xsd:string"/>
<xsd:element name="regione" type="xsd:string"/>
<xsd:element name="altezza" >

<xsd:complexType>

<xsd:simpleContent>

<xsd:extension base="xsd:integer">

<xsd:attribute name="unitaMisura" type="xsd:string"/>

</xsd:extension>

</xsd:simpleContent>

</xsd:complexType>

</xsd:complexType>

</xsd:element>

Modello Fette di Salame
```

```
Soluzione Modello Bambole Russe - 1
```

```
Esercizio 3 - raffinamento

Definire uno schema che permetta di validare il seguente documento:

<montagne><catenaMontuosa nome="Alpi">

<monte>

<nome>Monte Bianco</nome>

<regione>Valle d'Aosta</regione>

<altezza unitaMisura="metri">4810</altezza>

</monte>

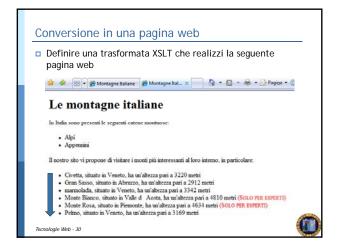
</catenaMontuosa></montagne>

...

e che permetta all'elemento catenaMontuosa di contenere altri elementi definiti in altri namespace.
```

```
Soluzione Esercizio 3 - raffinamento

<pre
```



```
Il file XML

</xml version="1.0" encoding="UTF-8"?>

</xml-stylesheet type="text/xsl" href="montagne.xsl"?>

<montagne>

<catenaMontuosa nome="Alpi">

<monte>

<nome>Monte Bianco</nome><regione>Valle d'Aosta</regione>

<altezza unitaMisura="metri">4810</altezza>

</monte>

<monte><nome>Monte Rosa</nome>...</monte>

</catenaMontuosa>

<catenaMontuosa nome="Appennini">

<monte> <nome>Gran Sasso</nome>...</monte>

</catenaMontuosa>

</catenaMontuosa>

</montagne>

/montagne>

Tecnologie Web - 31
```

```
La trasformata - 1

<p
```

```
La trasformata - 2

<br/>
<br/>
<h1>Le montagne italiane</h1>
 In Italia sono presenti le seguenti catene montuose: 

<br/>
<sl:for-each select="montagne/catenaMontuosa">
<br/>
<sl:oxsl:value-of select="@nome"/>
<br/>
</sl:for-each>

>p>Il nostro sito vi propone di visitare i monti pi8#250; interessanti al loro interno, in particolare: 

<ssl:for-each select="montagne/catenaMontuosa/monte">
<br/>
...</ssl:for-each select="montagne/catenaMontuosa/monte">
<br/>
...
```

```
La trasformata - 3

Inostro sito vi propone di visitare i monti piú interessanti al loro interno, in particolare: 

<xsl:for-each select="montagne/catenaMontuosa/monte">
<xsl:sort select="nome"/>
xml:space="preserve"><xsl:value-of select="nome"/>,
ha un'altezza pari a <xsl:value-of select="altezza"/>
<xsl:value-of select="altezza/@unitaMisura"/>
<xsl:if test="altezza &gt; 4000">
<span class="attenzione"> (Solo per esperti) </span><//xsl:if>

</xsl:for-each>

Tecnologie Web - 34
```

```
Namespace - la trasformata - 2

<br/>
<br/
```

```
Esercizio 4

Si scriva un possibile schema per il seguente file XML:
<alberghi>
<albergo categoria="****">
<nome>Hotel Astoria</nome>
<citta>Roma</citta><tel>06237813</tel>
<cel|>353134125</cel|>
<prezzo stanza="singola">120</prezzo>
<prezzo stanza="doppia">200</prezzo>
</albergo>
<albergo categoria=""">
<nome>Pensione Scatafascio</nome>
<citta>Roma</citta><tel>068872382</tel>
<prezzo stanza="doppia">500</prezzo>
</albergo categoria=""">
<nome>Pensione Scatafascio</nome>
<citta>Roma</citta><tel>068872382</tel>
<prezzo stanza="doppia">50</prezzo>
</albergo>
</albergo>
</albergo>
</alberghi>

</alberghi>

</alberghi>
</prezzo stanza="singola">40</prezzo>
</albergo>
</alberghi>

</alberghi>
```

```
Soluzione DTD

<!ELEMENT alberghi (albergo)+>
<!ELEMENT albergo (nome, citta, tel, cell?, prezzo+) >
<!ATTLIST albergo categoria CDATA #REQUIRED>
<!ELEMENT nome (#PCDATA) >
<!ELEMENT citta (#PCDATA) >
<!ELEMENT tel (#PCDATA) >
<!ELEMENT cell (#PCDATA) >
<!ELEMENT cell (#PCDATA) >
<!ELEMENT prezzo (#PCDATA) >
<!ELEMENT prezzo (#PCDATA) >
<!ATTLIST prezzo stanza CDATA #REQUIRED>
```

```
Soluzione DTD: raffinamento

<!ATTLIST prezzo stanza (singola | doppia ) #REQUIRED>

<!ATTLIST albergo categoria (* | ** | *** | ...) #REQUIRED>

NON CORRETTA!
```

```
Soluzione XMLSchema: elemento albergo

<xs:element name="albergo" maxOccurs="unbounded">

<xs:complexType><xs:sequence>

<xs:element name="nome" type="xs:string"/>

<xs:element name="citta" type="xs:string"/>

<xs:element name="tell" type="xs:string" minOccurs="0"/>

<xs:element name="prezzo" maxOccurs="unbounded">

...

</xs:element>

</xs:element>

</xs:sequence>

<xs:attribute name="categoria"> ... </xs:attribute>

</xs:complexType>

</xs:element>
```

```
Soluzione XMLSchema: elemento prezzo

<pre
```

```
Le nostre offerte

ALBERGO MIRALAGO, GARDA (TELEFONO 0458872382)
categoria ****** stanza doppia: 160 stanza singola: 100
PENSIONE UNA STELLA. MILANO (TELEFONO 028872382)
categoria * stanza doppia: 40 stanza singola: 30
ALBERGO CARLOT, MILANO (TELEFONO 028872382)
categoria: **** stanza doppia: 160 stanza singola: 100
HOTEL ASTORIA, ROMA (TELEFONO 06237813)
categoria: **** stanza singola: 120 stanza doppia: 200
PENSIONE SCATAFASCIO, ROMA (TELEFONO 068872382)
categoria: * stanza doppia: 50 stanza singola: 40
```



```
Codice - 1
```

```
Codice - 3
<xsl:for-each select="a:alberghi/a:albergo">
  <xsl:sort select="a:citta"/>
   <xsl:if test="not(a:citta=preceding-sibling::a:albergo/a:citta)">
   Queste sono le nostre offerte per alberghi a <xsl:value-of</p>
       select="a:citta"/> : 
   </xsl:if>
   <dl><dt> <xsl:value-of select="a:nome"/> (telefono <xsl:value-of
   select="a:tel"/>) </dt>
   <dd> categoria: <xsl:value-of select="@categoria"/>
        <xsl:for-each select="a:prezzo">
               stanza <xsl:value-of select="@stanza"/>: <xsl:value-of
                select="."/>
        </xsl:for-each></dd>
   </dl></xsl:for-each>
   ologie Web - 54
```

```
Esercizio 5
```

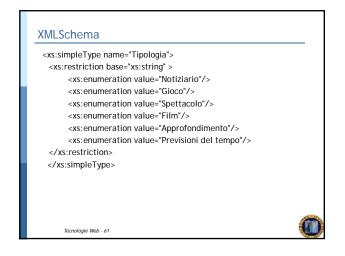
```
XMLSchema

<
```

```
XMLSchema

<
```

```
XMLSchema
<xs:complexType name="Tprogrammi">
   <xs:sequence>
       <xs:element name="titolo" type="xs:string"/>
       <xs:element name="tipologia" type="Tipologia"/>
       <xs:element name="puntata" type="xs:string" minOccurs="0"/>
       <xs:element name="descrizione" type="xs:string"</pre>
                                      minOccurs="0"/>
       <xs:choice>
         <xs:sequence>
               <xs:element name="dalle" type="xs:string"/>
               <xs:element name="alle" type="xs:string"/>
         </xs:sequence>
         <xs:element name="inizio" type="xs:string"/>
       </xs:choice></xs:sequence>
  </xs:complexType>
```



Definire un linguaggio XML che permetta di descrivere un documento di trasporto per ordini di libri identificato da un numero seriale. Ogni documento deve contenere i dati relativi all'acquirente (nome), l'indirizzo a cui spedire (nome, indirizzo, città, nazione) e di dati relativi ai libri ordinati (titolo, note[opzionale], quantità, prezzo)

Esercizio 5 - Documento XML

```
Che modello usare?

Modello Bambole Russe:
Tutti i tipi sono anonimi
Tutti gli elementi sono locali
Modello Fette di Salame:
Tutti i tipi sono anonimi
Tutti gli elementi sono globali
Modello Giardino dell'Eden
Tutti i tipi (e gli attributi) sono nominali
Tutti gli elementi sono globali
Modello Tende alla Veneziana
Tutti i tipi sono nominali
Tutti gli elementi sono locali
```

```
Esercizio 5 - Schema XML - Bambole Russe - 1

<
```

```
Esercizio 5 - Schema XML - Bambole Russe - 2

<p
```

```
Esercizio 5 - Schema XML - Fette di Salame - 1
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
   <xs:element name="shiporder">
       <xs:complexType> <xs:sequence>
               <xs:element ref="orderperson"/>
<xs:element ref="shipto"/>
               <xs:element ref="item" maxOccurs="unbounded"/>
       </xs:sequence> <xs:attribute ref="orderid" use="required"/>
       </xs:complexType>
   </xs:element>
   <xs:element name="orderperson" type="xs:string"/>
   <xs:element name="shipto">
       <xs:complexType> <xs:sequence>
               <xs:element ref="name"/> <xs:element ref="address"/>
               <xs:element ref="city"/> <xs:element ref="country"/>
       </xs:sequence> </xs:complexType>
   </xs:element>
Tecnologie Web - 67
```



```
Esercizio 5 - Schema XML - Giardino dell'Eden - 2

<xs:complexType name="Torder">
<xs:sequence>
<xs:element ref="orderperson"/> <xs:element ref="shipto"/>
<xs:selement ref="item" maxOccurs="unbounded"/>
</xs:sequence>
<xs:attribute ref="orderid" use="required"/>
</xs:complexType>
<xs:complexType name="Tshipto"> <xs:sequence>
<xs:element ref="name"/> <xs:element ref="address"/>
<xs:element ref="city"/> <xs:element ref="country"/>
</xs:sequence></xs:complexType>
```

```
Esercizio 5 - Namespace

La definizione corretta del namespace è:

<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns="http://www.dominio.com" targetNamespace="http://www.dominio.com" > ...

</xs:schema>
```

Esercizio 5 - Validazione documento XML <shiporder orderid="889923" xmlns="http://www.dominio.com" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance xsi:schemaLocation="http://www.dominio.com shiporder.xsd"> <orderperson>John Smith</orderperson> <shipto> <name>Ola Nordmann</name> <address>Langgt 23</address> <city>4000 Stavanger</city> <country>Norway</country> </shipto> <item> <title>Empire Burlesque</title> Non funziona! <item> <title>Hide your heart</title> <quantity>1</quantity> <price>9.90</price> </item> </shiporder>

```
Esercizio 5 - Validazione documento XML - BR

<so:shiporder orderid="889923" xmlns:so="http://www.dominio.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.dominio.com shiporder.xsd">
  <orderperson>John Smith</orderperson>
  <shipto> <name>Ola Nordmann</name>
  <address>Langgt 23</address>
  <city>4000 Stavanger</city>
  <country>Norway</country>
  </shipto>
  <item> <title>Empire Burlesque</title>
  <note>Special Edition</note>
  <quantity>1</quantity> <price>10.90</price>
  </item>
  <item> <title>Hide your heart</title> <quantity>1</quantity>
  </ri>
  </ri>

    </or>

              </or>

              </or>

              </or>

              </or>

              </or>

              </or>

              </or>

              </or>

              </or>

              </or>

              </or>

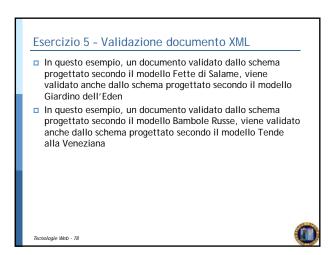
              </or>

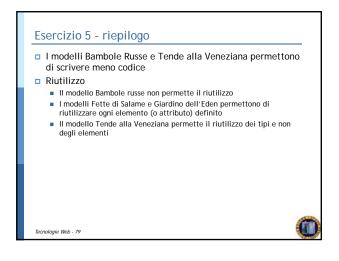
              </or>

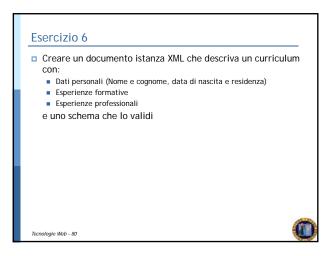
              </or>

              </or>
```

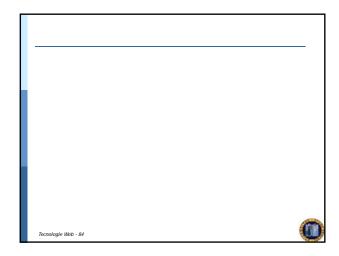
```
Esercizio 5 - Validazione documento XML - FS
<so:shiporder so:orderid="889923" xmlns:so="http://www.dominio.com"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://www.dominio.com shiporder.xsd">
   <so:orderperson>John Smith</so:orderperson>
   <so:shipto> < so:name>Ola Nordmann</so:name>
             <so:address>Langgt 23</so:address>
<so:city>4000 Stavanger</so:city>
              <so:country>Norway</so:country>
   </so:shipto>
   <so:item> <so:title>Empire Burlesque</so:title>
         <so:note>Special Edition<so:note>
<so:quantity>1</so:quantity> <so:price>10.90</so:price>
   </so:item>
   <so:item> <so:title>Hide your heart</so:title>
               <so:quantity>1</so:quantity><so:price>9.90</so:price>
   </so:item>
</so:shiporder>
Tecnologie Web - 77
```







```
Esercizio 6 - schema 2
```



Esercizio 7 Creare uno Schema XML che descriva la struttura di un documento dizionario, che rispetti i seguenti vincoli: L'elemento dizionario deve contenere zero o più elementi termine. L'elemento termine è costituito dagli elementi parola, pronuncia e almeno un'occorrenza dell'elemento significato. L'elemento significato e costituito dagli elementi definizione ed esempio (l'elemento esempio può, comparire zero o più volte). gli elementi parola, pronuncia, definizione e esempio sono di tipo stringa.

```
Esercizio 4 - 2
   <xsd:element name="esercizio" type="Tesercizio" />
   <xsd:complexType name="Tesercizio">
       <xsd:sequence>
               <element ref="car" minOccurs="6" maxOccurs="6"/>
               <element ref="num" minOccurs="2"</pre>
                             maxOccurs="2"/>
               <element ref="car"/>
               <element ref="num" minOccurs="2"
                              maxOccurs="2"/>
                <element ref="car"/>
               <element ref="num" minOccurs="3"</pre>
                             maxOccurs="3"/>
               <element ref="car"/>
        </xsd:sequence>
   </xsd:complexType>
```